Real-time Decision Making in Control and Optimization with Performance and Safety Guarantees

A dissertation presented

by

Yingying Li

to

The John A. Paulson School of Engineering and Applied Sciences

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Applied Mathematics

Harvard University

Cambridge, Massachusetts

July 2021

© 2021 — Yingying Li
All rights reserved.

Dissertation Advisor: Professor Na Li Yingying Li

Real-time Decision Making in Control and Optimization with Performance and Safety Guarantees

Abstract

With the rapid evolution of sensing, communication, computation, and actuation technology, recent years have witnessed a growing interest in real-time decision making, which leverages the data received in real time to improve the decision-making performance. Real-time decision making enjoys great potentials in two major domains: (i) in applications with time-varying environments, harnessing real-time data can allow quick adaptation to the new environments, (ii) when the underlying models are unknown, one can learn the models and improve the decisions by real-time data from the interaction with the systems. However, when applying real-time decision making to these domains, there are several major challenges, three of which are discussed below. Firstly, the real-time data available are usually limited and imperfect, which calls for more understanding on the effective usage and fundamental values of the real-time data. Secondly, the implementability on large-scale systems calls for computation-efficient and communication-efficient algorithm design. Thirdly, safety is crucial for the applications in the real world.

This thesis aims to tackle these challenges by studying real-time decision making in control and optimization with time-varying environments and/or unknown models. Specifically, the thesis consists of three parts.

In Part I, we consider online convex optimization and online optimal control with

time-varying cost functions. The future cost functions are unknown but some predictions on the future are available. We design gradient-based online algorithms to leverage predictions. Further, we consider different models of prediction errors for generality, and study our algorithms' regret guarantees in these models. We also provide fundamental lower bounds, which provide more insights into the fundamental values of the prediction information.

In Part II, we consider optimal control with constraints on the states and actions. We consider two problems, one with time-varying cost functions but known systems, and the other with unknown systems in a time-invariant setting. We design online algorithms for both settings. We provide safety guarantees of our online algorithms, i.e., constraint satisfaction and feasibility. We also provide sublinear regret guarantees for both algorithms, which indicate that our algorithms achieve desirable performance while ensuring safety requirements.

In Part III, we consider a decentralized linear quadratic control problem with unknown systems. We design a distributed policy gradient algorithm that only requires limited communication capacities. We provide a sample complexity bound for reaching a stationary point of the distributed policy optimization. We also provide stability guarantees for the controllers generated along the way.

Contents

Acknowledgments

First and foremost, I would like to express my most sincere gratitude to my advisor Prof. Na (Lina) Li. She has always been extremely supportive, helpful, insightful, caring, and the adjectives can go on and on. There are so many touching moments in my memory during my six years working with her. Lina has been extremely supportive when I explore different research areas. She would provide guidelines and encouragements when I face obstacles and have doubts in myself. I also enjoy listening to Lina's personal experiences when she was a PhD candidate. I have learned so much from these fascinating stories. Further, Lina is always there to remind me when I indulge myself in comfort zones, such as reading papers instead of doing innovative research, or polishing the math instead of starting to write paper. I am truly grateful for these reminders. Last but not the least, Lina is very caring and thoughtful. She would text me or call me when I am sick to make sure I have everything I need. I would really love to say: thank you, Lina, for all the time you spent on me and all the things you have done for me in the past six years!

I am deeply thankful for all the members on my thesis committee, Prof. Florian Dörfler, Prof. David Parkes, and Prof. Yiling Chen for their time and efforts on serving on my committee. I would really love to thank Prof. Florian Dörfler for the discussions and suggestions on the topics in this thesis and for the general suggestions on research and academia life. His questions and suggestions provide guidelines on some future directions mentioned in this thesis, e.g. replacing the canonical form in the online optimal control algorithm with more robust structures, which I hope to address after graduation. I would also love to thank Prof. David Parkes and Prof. Yiling Chen for the meetings and discussions in the past few years. Some questions arising from these meetings, e.g.

noisy predictions, motivate some extensions of my research that are included of this thesis.

I would further like to express my gratitude for Prof. Jeff Shamma, Prof. Lucas Janson, Prof. Flavio Calmon, and my intern mentor Subhro Das at IBM for their helpful discussions and insightful suggestions in the past years. I would especially like to thank Prof. Jeff Shamma for all his time and efforts on our meetings when I work on the safe control design. I would also love to thank Prof. Lucas Janson for his insights from statistics' perspective when I work on the distributed learning design. Further, I would love to thank Prof. Flavio Calmon for offering such a great class on information theory and all our discussions on theoretical lower bounds. Last but not the least, I am truly grateful for the intern opportunity offered by Subhro and I really enjoyed working and discussing research with Subhro.

Further, I would also love to thank Prof. Evelyn Hu, who is my mentor at Harvard Graduate Women in Science and Engineering. Evelyn provides critical support during my ups and downs. I could not thank her enough for her advice, her insights, and her effects as a role model.

I would also love to thank all my lab mates, including Guannan Qu, Yujie Tang, Xin Chen, Qinran Hu, Aoxiao Zhong, Yang Zheng, Runyu Zhang, Sindri Magnusson, Tianpeng Zhang, Zhaolin Ren, Xuan Zhang, etc. They all provide helps and supports during my PhD. Especially, I am very grateful for all the help and suggestions from Guannan, who helps me so such during my PhD that he almost feels like a second advisor to me. Further, I really appreciate all the time and efforts from my collaborators, e.g. Yujie Tang, Xin Chen, Qinran Hu, Aoxiao Zhong, and Runyu Zhang, whose contributions

are important to some of the results included in this thesis.

Beyond our lab, I would also love to thank Hong Hu, Hao Wang, Jiawei Zhou, Hongyao Ma, Linglin Huang, Elizabeth Bondi, Zhiyuan Zhang, and many more. Their friendship and support are very important during my PhD.

Last but not the least, I would like to thank my parents and my boyfriend for their unconditional love and support for my study in the United States, without which this dissertation could not have been possible.

List of Figures

Chapter 1 | Introduction

With rapid evolution of sensing, communication, computation, and actuation technology, real-time data become available and real-time decisions become computable and implementable in a growing number of applications, e.g. power systems [2, 3], transportation systems [4], smart buildings [5, 6], manufacturing [7, 8], robotics [9], etc. These trends give rise to great interests in *real-time decision making* in recent years. Roughly speaking, real-time/online decision making refers to sequential decision making that utilizes the data received in real time to improve the decision-making performance. Compared with the traditional decision making where decisions are determined ahead of time, real-time decison making enjoys great potentials in two major domains:[1]

- **Time-varying environments.** In the applications with time-varying environments and unknown (or not perfectly known) future, the decisions determined beforehand quickly become outdated. A natural hope is to use the data received in real time to adapt to the new environment for better performance.
- **Unknown systems.** In many applications, the underlying systems are unknown or not perfectly known [11–13]. Instead of waiting a long time for learning the true model before the optimal policy design, one can learn the system and improve the performance at the same time by updating the policies based on real-time feedback.

[1] There are other domains for the applications of real-time decision making, e.g. in machine learning with large-scale data [10], but this thesis mainly considers the two domains above.

However, real-time decision making faces several major challenges, some of which are discussed below. Firstly, real-time data are usually limited/incomplete/imperfect and are thus challenging to utilize for performance improvements. Secondly, despite the technology improvements on computing and communication, the real-time operations still impose strong requirements on the computation and communication efficiencies of online algorithms. Thirdly, the balance between safety and performance is inherently challenging, and addressing it in real time causes even more troubles.

This thesis aims to tackle the three challenges above in the two major domains above and focuses on real-time decision making in optimization and control. Specifically, we study three topics in this thesis. Each topic is mainly motivated by one challenge above, though some other challenges are also relevant and required to be addressed together.

(i) **Leveraging predictions in online decision making.** In this part (Chapter 2-3), we focus on the domain of time-varying environments and consider that imperfect predictions on the future environments are accessible. Our major goals are to utilize the imperfect predictions to improve the performance and to study how much improvement can be provided by the imperfect predictions. The computation efficiency is also required.

(ii) **Improving control performance with safety guarantees**. In this part (Chapter 4-7), we consider control problems in either the time-varying domain or the unknown-model domain. The major goal is to ensure safety during the whole process while still trying to optimize the performance. The computation efficiency is also required.

(iii) **Learning to cooperate under limited communication.** In this part (Chapter

8), we consider distributed control with unknown systems. Our major goal is to learn and optimize the systems with limited communication ability in real time. We also aim to address partial state observations and to guarantee stability here.

For the rest of this chapter, we discuss more on each topics and explain our contributions. Then, we provide an outline of the thesis with relevant publications.

1.1 Part I: leveraging predictions in online decision-making

In Part I, we aim to study real-time decision making with *time-varying environments*. One famous research area on time-varying environments is *online convex optimization* (OCO). In the following, we first briefly review OCO and explain its limitations, then introduce our problems studied in this part: smoothed OCO with predictions in Chapter 2, and online optimal control with predictions in Chapter. We also summarize our contributions.

1.1.1 Online convex optimization

OCO enjoys a rich body of literature [14–16]. Classical OCO considers a multi-stage problem with time-varying convex cost functions $f_1, f_2, \ldots, f_T$. At each stage t,

1) an online algorithm selects an action x_t from a feasible set $\mathbb{X}$ without knowing f_t,

2) f_t is (adversarially) generated,

3) the online algorithm suffers the cost $f_t(x_t)$ and receives information on the function f_t. The information will help the algorithm to make decisions in the next stages.

The goal of OCO is to design online algorithms with a small total cost $\sum_{t=1}^{T} f_t(x_t)$. The performance of the online algorithm is usually measured by *regret*, which compares the online total cost with the optimal total cost in hindsight. There are different definitions of regrets, two of which are introduced here, i.e., static regret and dynamic regret.

The static regret is defined as the online total cost minus the optimal time-invariant action's total cost in hindsight.

$$\text{Regret}^s = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathbb{X}} \sum_{t=1}^{T} f_t(x) \tag{1.1}$$

The static regret is probably the most commonly used performance measure in OCO. Sublinear $o(T)$ static regrets are usually achievable under proper conditions. An online algorithm with an $o(T)$ static regret enjoys a diminishing averaged stage regret, i.e. $\frac{1}{T}\text{Regret}^s \to 0$ as T goes to infinity, indicating desirable performance compared with any *time-invariant* action when T is large.

The dynamic regret, also called *competitive difference* in the literature [17–19], is defined as the online total cost minus the optimal total cost in hindsight, where the optimal actions can be time-varying.

$$\text{Regret}^d = \sum_{t=1}^{T} f_t(x_t) - \min_{x_1,\ldots,x_T \in \mathbb{X}} \sum_{t=1}^{T} f_t(x_t) \tag{1.2}$$

The dynamic regret is a stronger notion than the static regret because the optimal total cost by time-varying actions is usually much smaller than the total cost of optimal time-invariant actions in hindsight. It has been shown that the dynamic regrets of online algorithms heavily depend on the variation of the cost functions $\{f_t\}_{t=1}^{T}$, and when the variation is large, no online algorithm can achieve sublinear dynamic regrets in the worst-case scenario [20]. The dynamic regret measure is usually adopted when

the optimal time-invariant action's total cost is not sufficient to characterize desirable performance, for example, when there are shiftings or driftings in the underlying model that generates $\{f_t\}_{t=1}^T$, or when there is a stronger requirement on the online performance.

Despite the rich results, classical OCO does not capture two components that are commonly observed in real-world applications: time couplings and predictions.

- Time couplings. In the OCO formulation above, the current stage action x_t only affects the current stage cost $f_t(x_t)$. However, in many applications, the current action usually influences the future. For example, if a data center decides to switch off a server now, the server requires additional time and energy to be switched on in the future. An important question is how to make online decisions with lasting effects on the future.
- Predictions. In the OCO formulation above, the stage cost function f_t is adversarially generated, but in many applications, the environment is more benign and the future can even be predicted, e.g., weather forecasts, energy price forecasts. An important question is how to leverage the predictions to improve online performance.

To study the questions above, we consider two variant formulations of OCO: smoothed OCO with predictions in Chapter 2 and online optimal control with predictions in Chapter 3.

1.1.2 Smoothed online convex optimization with predictions

Smoothed OCO (SOCO) introduces a switching cost $d(x_t, x_{t-1})$ at each stage t to penalize the change of actions in consecutive stages, an example of which is a quadratic cost $d(x_t, x_{t-1}) = \frac{\beta}{2}\|x_t - x_{t-1}\|^2$. Consequently, the total cost becomes

$\mathsf{C}_\mathsf{T}(x_1, \ldots, x_T) = \sum_{t=1}^{T} (f_t(x_t) + d(x_t, x_{t-1}))$, which is coupled across stages. SOCO enjoys many applications. Further, we consider that some (imperfect) predictions on the future cost functions are accessible. This is motivated by the availability of predictions in many applications, e.g., weather forecast [21], demand forecast [17, 22], etc. The major goals are to design online algorithms to reduce the total costs by taking advantage of the prediction information and to study how much the predictions improve the performance.

However, most online algorithms for SOCO with predictions require solving multi-stage optimization problems, which may be too time-consuming for real-time implementation on large-scale systems. Further, most theoretical analyses on SOCO with predictions rely on prediction models with specific structures, which limits generality. Finally, the fundamental benefits brought by the predictions in SOCO are under-explored.

Our contributions of Chapter 2 We design a gradient-based online algorithm, Receding Horizon Gradient Descent (RHGD) and its variant online algorithms for SOCO with predictions. Our online algorithms only conduct a few gradient updates at each stage, thus being more computationally efficient than existing optimization-based algorithms.

We study two prediction models: (i) limited-accurate-prediction model and (ii) inaccurate-parametric-prediction model. These models include the existing models studied in the SOCO literature as special cases and model (ii) is more general than the existing models. For each model, we analyze the dynamic regret upper bounds of our online algorithms. Further, we provide the fundamental lower bounds on the dynamic regret for model (i). Numerical comparisons with other algorithms in the literature are

also provided.

1.1.3 Online optimal control with predictions

The SOCO problem above can be viewed as a special online optimal control problem with a simple $x_t = x_{t-1} + u_t$ linear system and the switching cost $d(x_t, x_{t-1}) = \frac{\beta}{2}\|x_t - x_{t-1}\|^2$ can be viewed as a quadratic control cost. Hence, a natural question is: can the results above be extended to general linear dynamical systems? The answer is yes and the extension is provided in Chapter 3.

Specifically, Chapter 3 considers a known time-invariant linear dynamical system $x_{t+1} = Ax_t + Bu_t$, and time-varying cost functions $f_t(x_t) + g_t(u_t)$. We only consider limited-accurate predictions in this problem but our results can be extended to other prediction models mentioned above. The online optimal control problem considered in this chapter is summarized below. At each $t = 0, 1, 2, \ldots,$

1) accurate predictions on the cost functions in the next W stages are revealed, i.e. $f_t(x) + g_t(u), \ldots, f_{t+W-1}(x) + g_{t+W-1}(u)$,

2) an online algorithm observes x_t, and selects a control action u_t,

3) the system evolves to a new state $x_{t+1} = Ax_t + Bu_t$, where A and B are considered to be known.

Consider a total of N stages. We measure the performance of our online algorithm by dynamic regret defined by

$$\text{Regret}^d = \sum_{t=0}^{N-1}(f_t(x_t) + g_t(u_t)) - \min_{x_{t+1}=Ax_t+Bu_t} \sum_{t=0}^{N-1}(f_t(x_t) + g_t(u_t)) \tag{1.3}$$

This problem enjoys many applications in, e.g., robotics [9], autonomous driving [4, 23], energy systems [3], manufacturing [7, 8], etc. Hence, there has been a growing interest on the problem, from both control and online learning communities.

In the control community, studies on the above problem focus on economic model predictive control (EMPC), which is a variant of model predictive control (MPC) with a primary goal of optimizing economic costs [24–31]. Recent years have seen a lot of attention on the optimality performance analysis of EMPC, under both time-invariant costs [32–34] and time-varying costs [27, 29, 35–37]. However, most studies focus on asymptotic performance and there is still limited understanding of the non-asymptotic performance, especially under time-varying costs. Moreover, for computationally efficient algorithms, e.g., suboptimal MPC and inexact MPC [38–41], there is limited work on the optimality performance guarantee.

From the online learning community, most studies on the online optimal control focus on the no-prediction case [42–45]. The results on the with-prediction case are mostly limited to OCO or SOCO, including our work in Chapter 2 and [17, 22, 46, 47].

Our contributions. We extend our results on SOCO in Chapter 2 to the online optimal control problem above. Specifically, we design receding horizon gradient control (RHGC) algorithms based on RHGD designed for SOCO. We also provide dynamic regret upper bounds of RHGC and discuss the fundamental lower bounds. We show that a variant of RHGC achieves a near-optimal dynamic regret bound in certain scenarios. Finally, we numerically compare our RHGC with a gradient-based suboptimal MPC algorithm. The major challenge in the extension is how to handle the time couplings introduced by the linear system. We address this challenge by linearly transforming the system to a

canonical form. More details are provided in Chapter 3.

1.2 Part II: improving control performance with real-time data under safety guarantees

In part II, we study optimal control with state constraints and action constraints, which enjoys a lot of applications, especially safety-critical applications [48–50]. For example, consider controlling a robot, the state constraints can describe the safe locations of the robot to avoid collisions, and the action constraints can represent the physical limitations of the actuator. The major challenge of constrained optimal control is to address the potential conflict between performance and safety under uncertainties: the controller should exercise caution in the face of uncertainties to avoid violating the constraints, but over-conservativeness results in degradation of performance. In part II, we tackle this challenge by considering two problems with different uncertainties: (i) Chapter 6 focuses on unknown future **time-varying** cost functions and known system dynamics, (ii) Chapter 7 considers **unknown system dynamics** in a time-invariant setting. Both problems and our contributions are briefly introduced below.

1.2.1 Online optimal control with constraints

In Chapter 6, we study online optimal control with a state constraint set $\mathbb{X}$ and an action constraint set $\mathbb{U}$. We consider a known linear dynamical system $x_{t+1} = Ax_t + Bu_t + w_t$ with random noises $w_t \in \mathbb{W}$. We also consider time-varying cost functions $c_t(x_t, u_t)$ that are unknown before each stage t. Specifically, at each stage $t = 0, 1, 2, \ldots,$

1) an online control algorithm observes the current state x_t and verifies if $x_t \in \mathbb{X}$, then

the algorithm selects an action $u_t \in \mathbb{U}$ without knowing the cost function c_t,

2) the algorithm suffers the current cost $c_t(x_t, u_t)$ and receives the current cost function $c_t(x, u)$,

3) the system evolves to the next state by $x_{t+1} = Ax_t + Bu_t + w_t$, where A and B are known, and w_t is random and bounded by $\mathbb{W}$.

We aim to design safe algorithms that satisfy $x_t \in \mathbb{X}$ and $u_t \in \mathbb{U}$ for any $w_t \in \mathbb{W}$ for all $t \geq 0$. We measure the performance of the online algorithm by *policy regret* defined below, which is a common measure in the online optimal control literature [42, 43, 45].

$$\text{Regret}^p = \mathbb{E}\left[\sum_{t=0}^{T} c_t(x_t, u_t)\right] - \min_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=0}^{T} c_t(x_t^\pi, u_t^\pi)\right] \tag{1.4}$$

where the policy class Π is called a benchmark policy class. For simplicity, this chapter considers linear policies $u_t = Kx_t$ for our benchmark policy class, though the optimal control policy for constrained optimal control can be nonlinear.

Our contributions of Chapter 3. We design an online control algorithm: Online Gradient Descent with Buffer Zones (OGD-BZ). Our OGD-BZ leverages the classic OCO algorithm, Online Gradient Descent, to handle unknown future time-varying cost functions. Besides, OGD-BZ introduces buffer zones to guarantee constraint satisfaction despite random system noises w_t. Theoretically, we provide safety guarantees and feasibility guarantees of OGD-BZ under proper parameters. Further, we provide a $\tilde{O}(\sqrt{T})$ policy regret upper bound for OGD-BZ. We also provide numerical results.

1.2.2 Safe adaptive learning of linear control

Chapter 7 studies constrained linear quadratic regulators with unknown systems and known costs. Specifically, we aim to solve the following problem by online learning.

$$\begin{aligned} \min \ & \lim_{T\to+\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}(x_t^\top Q x_t + u_t^\top R u_t) \\ \text{s.t. } & x_{t+1} = A_* x_t + B_* u_t + w_t, \ \forall\, t \geq 0, \\ & x_t \in \mathbb{X}, \quad u_t \in \mathbb{U}, \qquad \forall\, w_t \in \mathbb{W}. \end{aligned} \tag{1.5}$$

Our goal is to design a *safe adaptive* algorithm to gradually learn the system (A_*, B_*) to optimize (1.5) by interacting with the system without violating the constraints during the process. Further, we aim to analyze the non-asymptotic optimality of the safe adaptive algorithm by discussing its policy regret guarantees.

In the literature, robust model predictive control (RMPC) [51–55] and robust adaptive model predictive control (RAMPC) [55–58] are common methods for control design with constraint satisfaction under uncertainties, but they have limited results on the non-asymptotic optimality performance. In fact, most non-asymptotic optimality analyses in the literature are conducted for unconstrained systems [59–66].

Our contributions of Chapter 7 We design a safe adaptive control algorithm for (1.5). Our algorithm is based on certainty equivalence and utilizes exploration noises to learn the unknown system. Further, we extend the tools developed in Chapter 6 for constraint satisfaction under system uncertainties. We also design an algorithm for safe transitions when the model estimations are updated. We show that our algorithm ensures constraint satisfaction during the learning process under proper conditions. Besides, we provide an $\tilde{O}(T^{2/3})$ policy regret upper bound.

1.3 Part III: learning to cooperate under limited communication and partial observation

Encouraged by the recent success of learning-based centralized LQ control [59–66], this part aims to extend the results and develop scalable learning algorithms for decentralized LQ control in Chapter 8. Specifically, we consider the following decentralized LQ control setting in Chapter 8. Suppose a linear dynamical system, with a global state $x(t) \in \mathbb{R}^n$ and a global control action $u(t)$, is controlled by a group of agents. The global control action is composed of local control actions: $u(t) = \left[u_1(t)^\top, \ldots, u_N(t)^\top\right]^\top$, where $u_i(t)$ is the control input of agent i. At time t, each agent i directly observes a partial state $x_{\mathscr{I}_i}(t)$ and a quadratic local cost $c_i(t)$ that could depend on the global state and action. The dynamical system model is assumed to be unknown, and the agents can only communicate with their neighbors via a communication network. The goal is to design a cooperative distributed learning scheme to find local control policies for the agents to minimize the global cost that is averaged both among all agents and across an infinite horizon. The local control policies are limited to those that only use local observations.

Our contributions of Chapter 8. We propose a distributed adaptive control algorithm ZODPO for the problem above. Our design is based on (distributed) policy gradient and consensus. ZODPO only exchanges a few scalars between every two neighbors on the communication network at each stage, thus being communication efficient. Theoretically, we bound the sample complexity for ZODPO to approach some stationary point of the distributed policy optimization problem. Besides, we provide stability guarantees of all the generated controllers. Numerical results are also provided.

1.4 Structure of the thesis and the relevant publications

Our thesis consists of three main parts, and each part is named after the major topic of the corresponding part. The structure of each part is summarized below.

Part I: Leveraging Predictions in Online Decision Making (Chapter 2-3).

(a) Chapter 2 considers smoothed OCO with predictions. We consider two prediction models: (i) the limited-accurate-prediction model in Section 2.2 and (ii) the inaccurate-parametric-prediction model in Section 2.3.

(i) In Section 2.2, we study model (i) the limited-accurate-prediction model, which also includes the no-prediction case as a special case. We design gradient-based online algorithms: RHGD and its variants. Further, we discuss our algorithms' dynamic regret upper bounds and provide fundamental lower bounds. This section is based on our publication:

Li, Yingying, Guannan Qu, and Na Li. "Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit." *IEEE Transactions on Automatic Control* (2020).

(ii) In Section 2.3, we study (ii) the inaccurate-parametric-prediction model. We revise RHGD to cope with model (ii) and provide dynamic regret upper bounds. We also impose additional stochastic structures to the prediction model and analyze the performance of our algorithms under correlated prediction errors. This section is based on our publication:

Li, Yingying, and Na Li. "Leveraging predictions in smoothed online convex optimization via gradient-based algorithms." *Advances in Neural Information*

Processing Systems 33 (2020): 14520–14531.

(b) Chapter 3 extends the SOCO in Chapter 2 to an online optimal control problem with predictions. We design our online control algorithm RHGC based on RHGD above and provide dynamic regret upper bounds. Further, we consider a linear quadratic tracking problem and provide a fundamental lower bound on the dynamic regret, which is close to our RHGC's upper bound. This section is based on our publication:

Li, Yingying, Xin Chen, and Na Li. "Online Optimal Control with Linear Dynamics and Predictions: Algorithms and Regret Analysis." *Advances in Neural Information Processing Systems* 32 (2019): 14887-14899.

Part II: Improving Control Performance with Safety Guarantees (Chapter 4-7)

(a) Chapter 4 provides general introduction to our results in Chapter 6 and Chapter 7.

(b) Chapter 5 provides preliminary results that will be useful in Chapters 6 and 7.

(c) Chapter 6 considers online optimal control with state constraints and action constraints. We consider time-varying and unknown cost functions. Further, we assume the system is known here. We design online control algorithms with constraint satisfaction and feasibility guarantees. We also investigate the policy regret of our algorithm by comparing it with linear policies. We provide $\tilde{O}(\sqrt{T})$ regret upper bound for convex costs. This chapter is based on our publication:

Li, Yingying, Subhro Das, and Na Li. "Online Optimal Control with Affine Constraints." Proceedings of the AAAI Conference on Artificial Intelligence. Vol.

35. No. 10. 2021.

(d) Chapter 7 considers adaptive control of a constrained linear quadratic regulator. We consider an unknown system but a time-invariant and known cost function. We design a safe adaptive algorithm that learns the system and updates the policies at the same time on a single trajectory. We provide constraint satisfaction and feasibility guarantees. Further, we provide a $\tilde{O}(\sqrt{T})$ regret upper bound for our learning performance. Our major theoretical novelty is a general estimation error bound when learning with general (even nonlinear) policies. This chapter is recently submitted to a conference.

Part III: Learning to Cooperate under Limited Communication and Partial Observation (Chapter 8) This part consists of only one chapter, i.e., Chapter 8. This chapter studies a decentralized linear control system and designs a distributed adaptive control algorithm, which only requires exchanging a few scalars between neighbors on a communication network at each stage, thus being communication efficient. We provide sample complexity and stability guarantees for our algorithm. This chapter is based on:

Li, Yingying*, Yujie Tang*, Runyu Zhang, and Na Li. "Distributed reinforcement learning for decentralized linear quadratic control: A derivative-free policy optimization approach." *Accepted by IEEE Transactions on Automatic Control* (the first two authors contribute equally).

Finally, I also worked on residential demand response by mechanism design [67] and multi-armed bandits [68–70], which are not included in this thesis. For the interested reader, please refer to the papers cited above.

Part I

Leveraging Predictions in Online Decision Making

Chapter 2 | Smoothed Online Convex Optimization with Predictions

This chapter studies online convex optimization (OCO) with stage costs $f_t(x_t)$ and switching costs $d(x_t, x_{t-1})$, where the switching costs penalize the change of actions and enjoys wide applications [17,22,71–73]. Unlike classical OCO with unknown (adversarial) future costs, this chapter considers that some (imperfect) predictions on the future cost functions are accessible, motivated by the available predictions in practice, e.g., weather forecast [21,74], traffic forecast [75], etc. There are two major questions: how to use predictions to improve performance? how much improvement can predictions provide?

Contributions. For the first question, we design a gradient-based online algorithm and its variants. For the second question, we consider two models of predictions for generality: (i) limited-accurate predictions in Section 2.2, (ii) inaccurate-parametric predictions in Section 2.3. For each model, we analyze the dynamic regrets of our algorithms. Further, we provide fundamental lower bounds on the dynamic regrets for model (i). Our lower bounds share the same structure as our upper bounds, reflecting the benefits of the predictions. Numerical results are also provided.

Chapter outline. Section 2.1 provides an introduction. Section 2.2 considers model (i) and provides algorithm design, theoretical results, and numerical results. Section 2.3 focuses on model (ii) and also provides relevant results. Section 2.4 concludes the

chapter.

2.1 Introduction

This chapter studies online convex optimization (OCO) with not only a time-varying stage cost $f_t(x_t)$ and but also a switching cost $d(x_t, x_{t-1})$ at each stage t. The switching costs penalize the changes of actions in consecutive stages, one example of which is the quadratic difference function $d(x_t, x_{t-1}) = \frac{\beta}{2}\|x_t - x_{t-1}\|_2^2$. The problem is also called "smoothed OCO" (SOCO) in the literature [46, 72, 76, 77] and enjoys many applications [17, 22, 72, 73]. Unlike classical OCO with unknown (even adversarial) future costs, this chapter considers that some (imperfect) predictions on the future cost functions are accessible. This is motivated by the availability of predictions in many applications, e.g., weather forecast [21, 74], traffic forecast [75], etc. The goal is to design online algorithms to minimize the total costs by only using the available information. We measure the online algorithms' performance by "dynamic regret" (also known as competitive difference), which is defined as the online algorithm's total cost in T stages subtracted by the optimal total cost in hindsight:

$$\text{Regret}^d = \sum_{t=1}^{T}(f_t(x_t) + d(x_t, x_{t-1})) - \min_{x_1,\ldots,x_T} \sum_{t=1}^{T}(f_t(x_t) + d(x_t, x_{t-1}))$$

Despite the growing interest in SOCO with predictions [17, 19, 22, 46, 76], several important directions remain to be explored, which are discussed below.

- **Gradient-based online algorithm design.** Online decision making problems requires decision selections at every stage, thus calling for computationally efficient online algorithms especially for large-scale applications and/or high-frequency problems. Gradient-based online algorithms are popular choices for online decision

making and achieve desirable performance for classical OCO [14, 16]. However, most existing algorithms for SOCO with predictions require solving multi-stage optimization, e.g., AFHC [17], CHC [76], MPC [54, 76]. Gradient-based algorithm design for SOCO with predictions remains to be explored.

- **Models of imperfect predictions.** The choice of prediction models is crucial for insightful theoretical analysis and the applicability of the results. Several different models have been considered in the SOCO literature, e.g., the fully-adversarial model (no prediction available) [78], the limited-accurate model (perfect/accurate predictions in the next W stages but no further predictions) [17, 22], a stochastic parametric model [46, 76], etc. The first two models above are helpful starting points but are over-simplified. The third model above provides a more detailed characterization but assumes certain stochastic structures.

- **Fundamental limits.** The fundamental limits of SOCO with predictions are largely under-explored. There are a few papers on the fundamental limits of competitive ratio, which is a different performance metric defined by the ratio of the online total cost to the offline optimal cost [22, 79]. However, the fundamental limits of dynamic regrets (competitive differences) remain to be open questions.

2.1.1 Our contributions and chapter structure

Roughly speaking, we contribute to the directions above by designing gradient-based online algorithms for SOCO with predictions, conducting theoretical analysis for general prediction models, and providing fundamental limits for special cases. In the following, we discuss our contributions in more detail according to the structure of this chapter.

1) As a starting point, Section 2.2 considers the limited-accurate-prediction model studied in [17, 22], which considers accurate predictions in the next W stages but no predictions beyond the W stages. The prediction model and the problem formulation are formally provided in Section 2.2.1. This model simplifies the analysis and provides more intuitions, which will be helpful for our study on more general prediction models in Section 2.3. Besides, when $W = 0$, the model reduces to the fully-adversarial/no-prediction case, which is also studied in this Section. The contributions of Section 2.2 are summarized below.

   i) In Section 2.2.2, we design a gradient-based online algorithm, RHGD, and an improved version RHAG, to leverage the predictions in SOCO. Our algorithms are more computationally efficient than the optimization-based algorithms in the literature [17, 46, 76]. When $W = 0$ (no predictions), our algorithms reduce to classical OCO algorithms, such as Online Gradient Descent (OGD) [16].

   ii) In Section 2.2.3, we provide upper bounds on the dynamic regrets of RHGD and RHAG when there are predictions ($W > 0$). The regret upper bounds exponentially decay with W. We also provide an upper bound on the dynamic regret of OGD without predictions ($W = 0$).

   iii) In Section 2.2.4, we provide fundamental lower bounds on the dynamic regrets for both the limited-accurate-prediction case and the no-prediction case. Our results demonstrate the optimality of OGD for SOCO and the near-optimality of RHAG.

   iv) In Section 2.2.5, we conduct numerical experiments to test our algorithms.

2) In Section 2.3, we introduce a general parametric prediction model and analyze it.

By imposing additional stochastic structures, our model becomes the stochastic parametric model studied in [46, 76], which will also be analyzed in this Section. The contributions of Section 2.3 are summarized below.

i) In Section 2.3.1, we formally introduce a general parametric prediction model as well as other necessary problem formulation for this model.

ii) In Section 2.3.2, we explain how to revise RHGD in Section 2.2.2 to suit the prediction model in this Section.

iii) In Section 2.3.3, we provide dynamic regret upper bounds under both the general model and the stochastic model.

iv) In Section 2.2.5, we provide numerical comparisons between our algorithm and the existing algorithms in the literature.

2.1.2 Related work

Online convex optimization (OCO) has a rich body of literature and we refer the reader to [14] for a review. Classic OCO considers fully adversarial future costs. Motivated by the applications where additional predictions are available, [47] study OCO with predictions on the current stage cost function (since cost functions are fully decoupled in OCO, only the current stage cost function is relevant when making decisions).

As a side note, we want to clarify the usage of "predictions" in OCO's literature since the term "predictions" has different meanings in different settings: (i) For the fully adversarial setting, i.e., f_t is adversarial and unknown when deciding x_t, some papers call the decision x_t as the prediction at stage t since x_t can be viewed as a prediction of the optimal decision at stage t without knowing f_t [14], and these papers focus on

making predictions, i.e., selecting x_t, based on history. (ii) In this chapter and in the papers on OCO with predictions [17, 22, 46, 47, 76], at stage t, some (noisy) predictions on f_t are available and such predictions carry additional information. The goal is to select decisions x_t by *using predictions* and history information.

Smoothed online convex optimization (SOCO) is a variant of OCO with coupling switching costs at each stage. We summarize the literature on SOCO based on prediction models. (i) [78] study SOCO in the fully adversarial case. (ii) Several papers assume the current stage function f_t is available at stage t [18, 77]. (iii) [17, 22] study SOCO with first-accurate-then-adversarial predictions in the W steps. (iv) [46, 76] consider a stochastic prediction model when analyzing SOCO.

Other online decision making problems with coupling across stages include OCO with memory [80, 81], online optimal control (with or without predictions) [42, 82–84], OCO with switching constraints [85], online Markov decision processes [86–88], online nonconvex optimization with switching costs and predictions [19], etc. In particular, [82] is an extension of this chapter to the online optimal control and will be presented in the next chapter.

Model predictive control (MPC) is a classic control algorithm that can be applied to SOCO. Our setting is more related with the economic model predictive control (EMPC) with time-varying costs [29, 35, 37, 89, 90], where EMPC is a variant of MPC that aims to reduce the economic costs [29]. Besides, the dynamic regret analysis is related with the optimality analysis of (E)MPC that studies how MPC's cost deviates from the optimal one [29, 35, 37, 89]. However, the optimality guarantees of the fast (E)MPC schemes (see e.g. [38, 39, 54, 91–93]) are relatively under-explored, especially for the time-varying cases.

Time-varying optimization (TVO) is also relevant with this chapter to some extent. TVO considers $\min_x f(x;t)$ for each t [90, 94–97]. For theoretical purposes, most papers on TVO assume that the cost function $f(x;t)$ does not change dramatically with time t, e.g. $f(x;t)$ has certain smoothness properties with respect to t [90, 94–97], which is not assumed in this chapter. It is also worth mentioning the prediction-correction method [94], which computes the predictions of the future costs based on the smoothness of $f(x;t)$ with t; while the predictions in this chapter are not computed by our algorithms but are provided by some external sources. It is our future work to consider designing algorithms to generate predictions.

Notations

$\Pi_{\mathbb{X}}$ denotes the projection onto set $\mathbb{X}$. $\mathbb{X}^T = \mathbb{X} \times \cdots \times \mathbb{X}$ is a Cartesian product. ∇_x denotes the gradient with x. $\sum_{t=0}^{k} a_t = 0$ if $k < 0$. $\|\cdot\|_F$ and $\|\cdot\|$ are Frobenius norm and L_2 norm. $\mathbb{I}_E$ denotes an indicator function on set E. $\|\cdot\|$ denotes the l_2 norm. Denote $\nabla f(x,y)$ as the gradient and $\frac{\partial f}{\partial x}(x,y)$ as the partial gradient with x. For integers a, b, c, $a \equiv b \pmod{c}$ means $a = b + kc$ for some integer k. Let $|\mathbb{J}|$ denote the cardinality of the set $\mathbb{J}$. $\boldsymbol{A}^\top$ denotes the matrix transpose. We write $f(x) = O(g(x))$ ($f(x) = \Omega(g(x))$) if $|f(x)| \leq Mg(x)$ ($|f(x)| \geq Mg(x)$) for $x \geq M$; and we write $f(x) = o(g(x))$ if $\lim_{x\to+\infty} \frac{f(x)}{g(x)} = 0$. $\mathbf{1}_n \in \mathbb{R}^n$ is an all-one vector. $\boldsymbol{I}_n$ is an identity matrix in $\mathbb{R}^{n\times n}$. Denote $[T] = \{1, \ldots, T\}$.

2.2 Limited-accurate-prediction case

2.2.1 Problem formulation

This section considers an online convex optimization (OCO) problem in T stages with a stage cost function $f_t(\cdot)$ and a quadratic switching cost $d(x_t, x_{t-1}) = \frac{\beta}{2}\|x_t - x_{t-1}\|^2$ to penalize the changes in the decisions at each stage t.[1] Formally, we aim to solve

$$\min_{x_1,\dots,x_T \in \mathbb{X}} \mathsf{C}_\mathsf{T}(\boldsymbol{x}) = \sum_{t=1}^{T} \left(f_t(x_t) + \frac{\beta}{2}\|x_t - x_{t-1}\|^2 \right), \tag{2.1}$$

where $\mathbb{X} \subseteq \mathbb{R}^n$ is a convex feasible set, $\boldsymbol{x} = (x_1, \dots, x_T)$, $x_0 \in \mathbb{X}$ is given, $\beta \geq 0$ is a penalty parameter that is known a priori.

To solve (2.1), all the cost functions have to be known a priori, which is not practical in many applications [17]. Nevertheless, some predictions are usually available, especially for the near future. We adopt a simple model to represent the predictions: at each t, the agent receives the cost functions for the next W stages $f_t, \dots, f_{t+W-1}$,[2] but does not know the cost functions beyond the next W stages, that is, $f_{t+W}, f_{t+W+1}, \dots$ may be arbitrary or even adversarial. Though this prediction model is too optimistic in the near future and too pessimistic in the far future, this model captures a commonly observed property in applications, i.e., the short-range predictions are usually much more accurate than the long-range predictions. In addition, this model simplifies the theoretical analysis

[1] For simplicity, we only consider quadratic switching costs in this thesis, but our results can be extended to more general switching cost functions as in our paper [98].

[2] Predicting the complete function can be challenging, but it simplifies the analysis and it is often practical when the cost functions are parametric [17].

and helps generate insightful results that may lay a foundation for future work on more realistic settings, e.g., noisy predictions in Section 2.3.

Protocols of SOCO with limited-accurate predictions. In summary, the SOCO problem in this section is outlined below. At each stage $t = 1, 2, \ldots, T$, an agent

1) receives the predicted cost functions $f_t(\cdot), \ldots, f_{t+W-1}(\cdot)$,

2) computes a stage decision x_t by history and predictions,

3) suffers the cost $f_t(x_t) + \frac{\beta}{2}\|x_t - x_{t-1}\|^2$.

The online information available at each t contains both the predicted and the history cost functions, i.e. $\{f_1, \ldots, f_{t+W-1}\}$. Our goal is to design an online algorithm $\mathscr{A}$ that computes the decision $x_t^{\mathscr{A}}$ based on the online information to minimize the total cost $\mathsf{C}_\mathsf{T}(\boldsymbol{x}^{\mathscr{A}})$. We measure the performance by *dynamic regret* [99], which compares the online algorithm's cost with the optimal cost in hindsight:

$$\text{Regret}^d(\mathscr{A}) = \mathsf{C}_\mathsf{T}(\boldsymbol{x}^{\mathscr{A}}) - \mathsf{C}_\mathsf{T}(\boldsymbol{x}^*) \tag{2.2}$$

where $\boldsymbol{x}^*$ denotes the optimal solution to (2.1) in hindsight.

To ease the theoretical analysis, we list a few assumptions.

Assumption 2.1. *Function $f_t(\cdot)$ is α_t-strongly convex and l_t-smooth in $\mathbb{R}^n$.*[3] *In addition, there exist constants $\alpha, l > 0$ that do not depend on T s.t. $\alpha_t \geq \alpha$ and $l_t \leq l$ for all t.*

Assumption 2.2. *For all $1 \leq t \leq T$, $\sup_{x \in \mathbb{X}} \|\nabla f_t(x)\| \leq G$.*

[3]Here we consider $\mathbb{R}^n$ because we will use Nesterov's accelerated gradient that requires strong convexity and smoothness outside the feasible set $\mathbb{X}$ [100].

Assumption 2.3. $\mathbb{X}$ *is compact with* $D \coloneqq \max_{x,y\in\mathbb{X}} \|x-y\|$.

We assume Assumption 2.1 holds throughout this section. We will explicitly state it when Assumption 2.2 and 2.3 are needed.

Finally, we provide two examples for our problem above.

Example 2.1 (Trajectory Tracking)**.** *Consider a dynamical system* $x_t = x_{t-1} + u_t$*, where* x_t *is the robot's location,* u_t *is the velocity. Let* y_t *be the target's location. The tracking problem can be formulated as* (2.1) *where* $f_t(x_t) = \|x_t - y_t\|^2$ *and the switching cost is the control cost. In reality, a short lookahead window is sometimes available for the target trajectory [71].*

Example 2.2 (Smoothed regression)**.** *Consider a learner solving a sequence of regression tasks without changing the regressors too much between the stages [72]. The problem can be modeled as* (2.1) *where* $f_t(\cdot)$ *is the regression loss, and* β *is the smoothing regularization's parameter. In some cases, a short lookahead window of future tasks are available, e.g., when multiple tasks arrive at the same time but are solved sequentially [73].*

2.2.2 Online Algorithms with Predictions

In the following, we design two online algorithms RHGD and RHAG. Both algorithms only require a few gradient evaluations at each stage, thus being more computationally efficient than the optimization-based online algorithms in the literature [17, 76]. Roughly, RHGD and RHAG adopt OCO (without predictions) algorithms as initialization; then apply gradient descent and Nesterov's accelerated gradient [100] to update the online decisions based on the predictions. The main motivation for RHGD and RHAG is the

structure of the offline optimization on the total cost function $\mathsf{C_T}$. More details are discussed below

2.2.2.1 Offline Optimization and Offline Gradient Descent

Given all cost functions $f_1, \ldots, f_T$, the problem (2.1) becomes a classical optimization problem and can be solved by, e.g., projected gradient descent (GD). The updating rule of GD is the following. For iteration $k = 1, 2, \ldots,$

$$\boldsymbol{x}(k) = \Pi_{\mathbb{X}\times\cdots\times\mathbb{X}} \left(\boldsymbol{x}(k-1) - \eta \nabla \mathsf{C_T}(\boldsymbol{x}(k-1))\right), \tag{2.3}$$

where $\mathbb{X} \times \cdots \times \mathbb{X}$ denotes the joint feasible set of $\boldsymbol{x}$, η is the stepsize, the initial value $\boldsymbol{x}(0)$ is given. The gradient $\nabla \mathsf{C_T}$ can be evaluated by the partial gradient on each variable x_t:

$$\frac{\partial \mathsf{C_T}}{\partial x_t}(\boldsymbol{x}) = \nabla f_t(x_t) + \beta(x_t - x_{t-1}) + \beta(x_t - x_{t+1}),$$

when $1 \le t \le T-1$, and $\frac{\partial \mathsf{C_T}}{\partial x_T}(\boldsymbol{x}) = \nabla f_T(x_T) + \beta(x_T - x_{T-1})$ at stage T. Notice that the partial gradient $\frac{\partial \mathsf{C_T}}{\partial x_t}(\boldsymbol{x})$ only depends on the cost function f_t and the stage variables x_{t-1}, x_t, x_{t+1}. To emphasize this fact, we slightly abuse the notation and write the partial gradient as $\frac{\partial \mathsf{C_T}}{\partial x_t}(x_{t-1:t+1})$. With this notation, the projected gradient descent (2.3) can be written equivalently as follows. For iteration $k = 1, 2, \ldots,$ the updating rule of GD on the stage variable x_t for $1 \le t \le T$ is

$$x_t(k) = \Pi_{\mathbb{X}} \left[x_t(k-1) - \eta \frac{\partial \mathsf{C_T}}{\partial x_t}(x_{t-1:t+1}(k-1)) \right], \tag{2.4}$$

Rule (2.4) shows that, to compute $x_t(k)$ by the offline gradient descent, we only need f_t and $x_{t-1}(k-1), x_t(k-1), x_{t+1}(k-1)$, instead of all the cost functions and all the stage variables. This suggests that it is possible to implement (2.4) for a few iterations

using only W-lookahead information. This is the key insight that motivates our online algorithm design below.

2.2.2.2 Receding Horizon Gradient Descent (RHGD)

Inspired by the offline gradient descent, we design our online RHGD (see Algorithm 1). For ease of notation, we define $f_t(\cdot) := 0$ for $t \leq 0$ or $t > T$ and let $x_t(k) := x_0$ for $t \leq 0$ and $k \geq 0$ when necessary. At stage $t = 1 - W$, RHGD sets $x_1(0) = x_0$. At stage $2 - W \leq t \leq T$, RHGD receives $f_t, \ldots, f_{t+W-1}$ and runs the following two steps.

In Step 1, RHGD initializes the variable $x_{t+W}(0)$ with an initialization oracle ϕ. Notice that ϕ can be any method that only uses the available information at t, i.e. $f_1, \ldots, f_{t+W-1}$ and the stage variables computed before t. For instance, ϕ can be online gradient descent (OGD), which is a well-known OCO algorithm in literature [14] and is provided below.

$$x_{t+W}(0) = \Pi_{\mathbb{X}} \left[x_{t+W-1}(0) - \gamma \nabla f_{t+W-1}(x_{t+W-1}(0)) \right], \tag{2.5}$$

where $\gamma > 0$ is the stepsize and $x_{t+W-1}(0)$ is available from the Step 1 at the previous stage $t - 1$.

In Step 2, RHGD updates the values of $x_{t+W-1}, x_{t+W-2}, \ldots, x_t$ one by one by (2.4). In the following, we show that Step 2 computes the exact values of $x_{t+W-1}(1), \ldots, x_t(W)$ defined in the offline gradient descent (2.4) for the offline optimization (2.1) by only using the available online information at t.

- At first, RHGD computes $x_{t+W-1}(1)$ exactly by (2.4) since f_{t+W-1} is received, $x_{t+W}(0)$ is computed in Step 1, $x_{t+W-1}(0)$ and $x_{t+W-2}(0)$ have been computed in Step 1 of the

Algorithm 1 Receding Horizon Gradient Descent (RHGD)

Inputs: x_0, $\mathbb{X}$, β, W, γ, η, ϕ. Let $x_1(0) = x_0$.

for $t = 2 - W$ **to** T **do**

`Step 1: initialize` x_{t+W}.

compute $x_{t+W}(0)$ by the initialization oracle ϕ.

`Step 2: update` $x_{t+W-1}, x_{t+W-2}, \ldots, x_t$.

for $s = t + W - 1$ **downto** t **do**

compute $x_s(k)$ by projected gradient descent (2.4), where $k = t + W - s$,

$$x_s(k) = \Pi_{\mathbb{X}} \left[x_s(k-1) - \eta \frac{\partial \mathsf{C}_\mathsf{T}}{\partial x_s}(x_{s-1:s+1}(k-1)) \right]$$

Output $x_t(W)$ at stage t.

stages $t-1$ and $t-2$ respectively.

- Next, RHGD computes $x_{t+W-2}(2)$ exactly by (2.4) since f_{t+W-2} is received, $x_{t+W-1}(1)$ is computed above, $x_{t+W-2}(1)$ and $x_{t+W-3}(1)$ have been computed in Step 2 of the stages $t-1$ and $t-2$ respectively.

- Similarly, RHGD computes $x_{t+W-3}(3)$, $\ldots, x_t(W)$ one by one based on the received costs and the values that have been computed at stages $t, t-1, t-2$.

The final output at stage t is $x_t(W)$, which is the same as that in the Wth iteration of the offline gradient descent (2.4) for C_T.

Notice that RHGD (with OGD initialization) only requires $W+1$ projected gradient evaluations at each t. Hence, RHGD is computationally efficient when the projection can be evaluated efficiently, e.g., when $\mathbb{X}$ is a probability simplex, an n-dimensional box, a Euclidean ball, a positive orthant, etc.

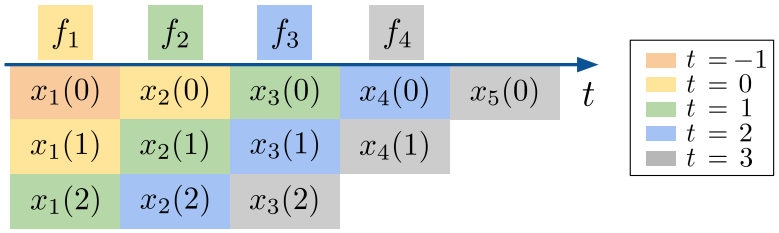


Figure 2.1: An example when $W = 2$. The new cost function received and the new variables computed by RHGD at stage $t = -1, 0, 1, 2, 3$ are marked in different colors.

Example 2.3 (Illustrative example)**.** *Figure 2.1 provides an illustrative example for RHGD when* $W = 2$. *Define* $x_t(k) = x_0$ *for* $t \leq 0$, $k \geq 0$; $f_t = 0$ *for* $t \leq 0$. *At* $t = -1$, *let* $x_1(0) = x_0$.

- *At* $t = 0$, *RHGD receives* f_0 *and* f_1. *Step 1 initializes* $x_2(0)$ *by OGD* (2.5) *with* f_1 *and* $x_1(0)$. *Step 2 computes* $x_1(1)$ *by GD* (2.4) *with* f_1 *and* $x_0(0), x_1(0), x_2(0)$. *RHGD computes* $x_0(2)$ *for ease of notation, which is omitted in Figure 2.1.*

- *At* $t = 1$, *RHGD receives* f_1, f_2. *Step 1 initializes* $x_3(0)$ *by* (2.5) *with* f_2 *and* $x_2(0)$. *Step 2 first computes* $x_2(1)$ *by* (2.4) *with* f_2 *and* $x_1(0), x_2(0), x_3(0)$ *and then computes* $x_1(2)$ *by* (2.4) *with* f_1 *and* $x_0(1), x_1(1), x_2(1)$. *RHGD outputs* $x_1(2)$.

- *Similarly, at* $t = 2$, *RHGD receives* f_2, f_3, *initializes* $x_4(0)$, *computes* $x_3(1)$ *and then* $x_2(2)$ *by* (2.4), *and outputs* $x_2(2)$. *At* $t = 3$, *RHGD initializes* $x_5(0)$, *computes* $x_4(1)$ *and then* $x_3(2)$, *and outputs* $x_3(2)$. *So on and so forth.*

2.2.2.3 Receding Horizon Accelerated Gradient (RHAG)

The design idea of RHGD can be extended to other gradient methods, e.g., Nesterov's Accelerated Gradient (NAG), Triple Momentum, etc. Due to space limits, we only

introduce RHAG (Algorithm 2) based on NAG. NAG updates $x_t(k)$ and auxiliary variable $y_t(k)$ for $1 \le t \le T$ at iteration k by

$$x_t(k) = \Pi_{\mathbb{X}}\left[y_t(k-1) - \eta \frac{\partial \mathsf{C}_{\mathsf{T}}}{\partial y_t}(y_{t-1:t+1}(k-1))\right],\ y_t(k) = (1+\lambda)x_t(k) - \lambda x_t(k-1).$$

Similar to RHGD, RHAG also conducts two steps at each t. The differences are: in Step 1, RHAG initializes not only $x_{t+W}(0)$ but also $y_{t+W}(0)$; and in Step 2, RHAG updates $x_{t+W-1}(1), \ldots, x_t(W)$ by NAG instead of the gradient descent. Nevertheless, RHAG still outputs $x_t(W)$, which is the value of x_t after Wth iterations of NAG. In total, RHAG also only requires $W+1$ projected gradient evaluations.

Algorithm 2 Receding Horizon Accelerated Gradient (RHAG)

Inputs: x_0, $\mathbb{X}$, β, W, γ, η, λ, ϕ. Let $x_1(0) = y_1(0) = x_0$.

for $t = 2 - W$ **to** T **do**

`Step 1: initialize` x_{t+W}, y_{t+W}.

Compute $x_{t+W}(0)$ by the initialization oracle ϕ. Let $y_{t+W}(0) = x_{t+W}(0)$.

`Step 2: update` $(x_{t+W-1}, y_{t+W-1}), \ldots, (x_t, y_t)$.

for $s = t + W - 1$ **downto** t **do**

Compute $(x_s(t+W-s), y_s(t+W-s))$ by NAG, where $k = t+W-s$,

$$x_s(k) = \Pi_{\mathbb{X}}\left[y_s(k-1) - \eta \frac{\partial \mathsf{C}_{\mathsf{T}}}{\partial y_s}(y_{s-1:s+1}(k-1))\right],\ y_s(k) = (1+\lambda)x_s(k) - \lambda x_s(k-1)$$

Output $x_t(W)$ at stage t.

2.2.3 Regret upper bounds

This subsection provides the regret upper bounds of our gradient-based online algorithms RHGD and RHAG. The proofs are provided in Appendix A.1. To establish the bounds, we first prove some properties of C_{T}.

Lemma 2.1. *Given Assumption 2.1,* $\mathsf{C}_\mathsf{T}(\boldsymbol{x})$ *defined in* (2.1) *is* α*-strongly convex and* L*-smooth on* $\mathbb{R}^n$ *for* $L = l + 4\beta$.

Proof. The Hessian of $\sum_{t=1}^{T} \frac{\beta}{2}\|x_t - x_{t-1}\|^2$ has eigenvalues in $[0, 4\beta]$ by the Gershgorin circle theorem. Thus, by Assumption 2.1, $\mathsf{C}_\mathsf{T}(\boldsymbol{x})$ is α strongly convex and $L = l + 4\beta$ smooth. □

Next, we introduce the regret upper bounds under general initialization methods.

Theorem 2.1 (General regret upper bounds)**.** *Under Assumption 2.1, for* $W \geq 0$*, given stepsizes* $\eta = 1/L$ *and* $\lambda = \frac{1-\sqrt{\alpha/L}}{1+\sqrt{\alpha/L}}$,[4] *for any initialization* ϕ *in Step 1, we have*

$$\text{Regret}^d(RHGD) \leq Q_f \left(1 - 1/Q_f\right)^W \text{Reg}(\phi) \tag{2.6}$$

$$\text{Regret}^d(RHAG) \leq 2\left(1 - 1/\sqrt{Q_f}\right)^W \text{Reg}(\phi) \tag{2.7}$$

where $Q_f = \frac{L}{\alpha}$ *and* $\text{Reg}(\phi)$ *is the regret of implementing the initial values* $\{x_t(0)\}_{t=1}^T$ *computed by the initialization* ϕ.

Proof of Theorem 2.1. The proof is straightforward and utilizes the equivalence between RHGD with offline gradient descent on C_T. Notice that $x_t(W)$ is the Wth iterate of (projected) GD for $\mathsf{C}_\mathsf{T}(\boldsymbol{x})$. By GD's convergence rate, we have $\text{Regret}^d(RHGD) = \mathsf{C}_\mathsf{T}(\boldsymbol{x}(W)) - \mathsf{C}_\mathsf{T}(\boldsymbol{x}^*) \leq Q_f(1 - 1/Q_f)^W[\mathsf{C}_\mathsf{T}(\boldsymbol{x}(0)) - \mathsf{C}_\mathsf{T}(\boldsymbol{x}^*)] = Q_f(1 - 1/Q_f)^W \text{Regret}^d(\phi)$. $\text{Regret}^d(RHAG)$ can be bounded similarly by projected NAG's convergence rate [100]. □

Remark 2.1. *The stepsizes of RHGD and RHAG can be different from those in Theorem 2.1. We can use any stepsizes with meaningful convergence rates for GD and NAG.*

[4] The stepsize conditions can be relaxed as in [29], yielding slightly different bounds.

Remark 2.2. *Both RHGD and RHAG's regret upper bounds decay exponentially with the length of accurate lookahead window W, indicating that the online performance quickly improves with additional prediction information. Further, RHAG's decay rate is faster than RHGD's, which is intuitive because RHAG is based on Nesterov's accelerated gradient, which enjoys a faster convergence rate than gradient descent.*

In the following, we provide a regret bound for a specific initialization method: OGD, and then complete our regret bounds of RHGD and RHAG.

Theorem 2.2 (Regret upper bounds with OGD initialization)**.** *When Assumption 2.1-2.3 hold, the regret of OGD (2.5) with stepsize $\gamma = 1/l$ is bounded by:*

$$\text{Regret}^d(OGD) \leq \delta \sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\|,$$

where $\delta = (\beta/l + 1)\frac{G}{(1-\kappa)}$, $\kappa = \sqrt{(1 - \frac{\alpha}{l})}$, $\xi_t = \arg\min_{x \in \mathbb{X}} f_t(x)$ for $t \geq 1$, and $\xi_0 = x_0$.[5] *Consequently, with OGD initialization, under the conditions in Theorem 2.1, we have*

$$\text{Regret}^d(RHGD) \leq \delta Q_f (1 - \frac{1}{Q_f})^W \sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\|,$$

$$\text{Regret}^d(RHAG) \leq 2\delta (1 - \frac{1}{\sqrt{Q_f}})^W \sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\|.$$

Remark 2.3. *The term $\sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\|$ is usually called the "path length" of the cost sequence $\{f_t\}_{t=1}^{T}$ and is a characterization of the variation of the cost sequence [101]. Theorem 2.2 shows that OGD's dynamic regret upper bound is linear with the path length for SOCO without predictions. This suggests that OGD achieves small dynamic regrets when the variation of the cost functions is small. Besides, it is interesting to note that OGD's dynamic regret bound for OCO, i.e., without switching costs, is also linear with the path length.*

[5] Stepsizes $\gamma \leq 1/l$ also work, yielding different constant factors.

Stability. The SOCO problem can be viewed as a simple optimal control problem as indicated in Example 2.1. Hence, the stability of our algorithms may be of interest for some readers. In the following, we show that our RHGD and RHAG enjoy certain stability results.

Corollary 2.1 (Asymptotic stability)**.** *Consider the optimal control problem below*

$$\min \sum_{t=1}^{+\infty} \left[f_t(x_t) + \frac{\beta}{2}\|u_t\|^2 \right], \quad \text{s.t. } x_t = x_{t-1} + u_t, \quad x_t \in \mathbb{X}.$$

The problem is equivalent to problem (2.1) in an infinite horizon. Notice that $\xi_t := \arg\min_{\mathbb{X}} f_t(x)$ *is the optimal steady state for the stage cost at* t*. If* $\xi_t \to \xi_\infty$ *as* $t \to +\infty$ *and* $\sum_{t=1}^{+\infty} \|\xi_t - \xi_{t-1}\| < +\infty$*, then* $x_t(W) \to \xi_\infty$ *as* $t \to +\infty$*, where* $x_t(W)$ *denotes the output of RHGD (or RHAG) and* $W \geq 0$*.*

2.2.4 Fundamental limits

This subsection provides fundamental lower bounds on the dynamic regrets for any online algorithms. We consider both the no-prediction case and the limited-accurate-prediction case. All the proofs are provided in Appendix A.1.

The no-prediction case.

Theorem 2.3 (No predictions)**.** *Consider* $W = 0$*. Given any* $T \geq 0$*,* $\alpha > 0$*,* $\beta \geq 0$*, a convex compact set* $\mathbb{X}$ *with diameter* $D > 0$*, and* $0 \leq L_T \leq DT$*, for any deterministic online algorithm* $\mathscr{A}$*,*[6] *there exist a sequence of quadratic functions* $\{f_t(\cdot)\}_{t=1}^T$ *with* α *strong convexity and* α *smoothness on* $\mathbb{R}^n$*, gradient bound* $(3\alpha + \beta)D$ *on* $\mathbb{X}$*, path length*

[6]For simplicity, this chapter only studies deterministic online algorithms and leave the stochastic online algorithms as future work.

$\sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\| \leq L_T$, *where* $\xi_t = \arg\min_{\mathbb{X}} f_t(x)$, *such that the regret is bounded by*

$$\text{Regret}^d(\mathscr{A}) \geq \zeta D L_T \geq \zeta D \sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\| \tag{2.8}$$

where $\zeta = \frac{\alpha^3(1-\rho)^2}{32(\alpha+\beta)^2}$, $\rho = \frac{\sqrt{Q}_f - 1}{\sqrt{Q}_f + 1}$, *and* $Q_f = \frac{\alpha+4\beta}{\alpha}$.

The proof is deferred to Appendix A.1.2. We provide some discussions below. Theorem 2.3 shows that when the path length is $\Omega(T)$, no online algorithm can achieve $o(T)$ dynamic regret, which is similar to the claims in [20]. Further, notice that the fundamental regret lower bound is linear with the path length, while Theorem 2.2 shows that OGD's regret upper bound is also linear with the path length. This indicates the optimality of OGD for SOCO without predictions. We call the term L_T as the "path length budget" since L_T is an upper bound on the path length $\sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\|$. Finally, when $\beta = 0$, Theorem 2.3 provides a fundamental limit on the dynamic regret for (classical) OCO (without switching costs or predictions), which matches OGD's dynamic regret in [99].

The limited-accurate-prediction case

Theorem 2.4 (W-stage predictions)**.** *Given any* $T > 1$, $1 \leq W \leq T/2$, $\alpha > 0$, $\beta \geq 0$, *a convex compact set* $\mathbb{X}$ *with diameter* $D > 0$, $0 \leq L_T \leq DT$, *for any deterministic online algorithm* $\mathscr{A}$, *there exist a sequence of quadratic functions* $\{f_t(\cdot)\}_{t=1}^{T}$ *with* α *strong convexity and* α *smoothness on* $\mathbb{R}^n$, *gradient bound* αD *on* $\mathbb{X}$, *path length* $\sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\| \leq L_T$, *where* $\xi_t = \arg\min_{\mathbb{X}} f_t(x)$, *such that the regret satisfies*

$$\text{Reg}(\mathscr{A}) \geq \begin{cases} \frac{\zeta D}{3} \rho^{2W} L_T, & \text{if } L_T \geq D, \\ \frac{\zeta}{3} \rho^{2W} L_T^2, & \text{if } L_T < D, \end{cases} \tag{2.9}$$

where $\rho = \frac{\sqrt{Q}_f - 1}{\sqrt{Q}_f + 1}$, $Q_f = \frac{\alpha + 4\beta}{\alpha}$, *and* $\zeta = \frac{\alpha^3(1-\rho)^2}{32(\alpha+\beta)^2}$.

We first provide some discussions and then provide a proof in the following.

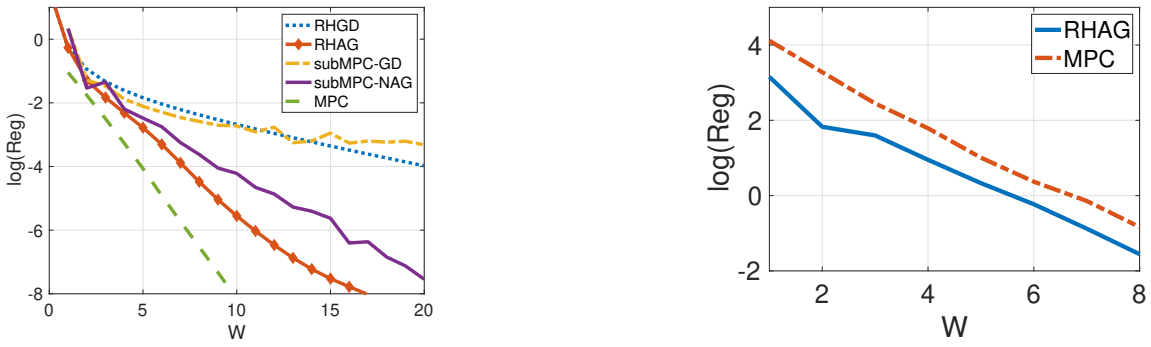
Special-case analysis. Notice that Theorem 2.3 and 2.4 only prove the fundamental lower bounds for the *special/worst cases*. It is possible to achieve smaller regrets in non-worst cases.

Effect of W. Theorem 2.4 shows that given W-lookahead information, the dynamic regret of any online algorithm at most decays exponentially with W. The decay rate of the fundamental limit is close to the decay rate of RHAG in Theorem 2.1 in the sense that for large Q_f, to reach the same regret value R, the lower bound requires at least $W \geq \Omega((\sqrt{Q_f} - 1)\log(L_T/R))$ (by $\rho^{2W} \geq \exp(-4W/(\sqrt{Q_f} - 1))$), while RHAG requires at most $W \leq O(\sqrt{Q_f}\log(L_T/R))$ (by $(1 - 1/\sqrt{Q_f})^W \leq \exp(-W/\sqrt{Q_f})$).

Discussion on path length. Theorem 2.4 shows that when L_T is not close to 0 ($L_T \geq D$), the lower bound is linear with the path length when $W > 0$, which is consistent with our regret upper bounds in Theorem 2.4. Interestingly, when L_T is close to 0 ($L_T < D$), the lower bound in Theorem 2.4 is quadratic on L_T. When $L_T \to 0$, this is smaller than the linear dependence of the upper bounds in Theorem 2.2. Nevertheless, the quadratic dependence on L_T when $W \geq 1$ can be achieved by a simple (optimization-based) online algorithm, that is, let $x_t^{\mathscr{A}} = \xi_t$. Since ξ_t minimizes each $f_t(\cdot)$, the dynamic regret of $x_t^{\mathscr{A}} = \xi_t$ is upper bounded by the switching costs, i.e. $\sum_{t=1}^{T} \frac{\beta}{2}\|\xi_t - \xi_{t-1}\|^2$, which is bounded by L_T^2. However, when there is no prediction, i.e., $W = 0$, this simple online algorithm can not be implemented because $f_t(\cdot)$ is not available at stage t. In fact, the fundamental limit in Theorem 2.3 indicates that no algorithm can achieve $O(L_T^2)$ regret as $L_T \to 0$ when $W = 0$.

2.2.5 Numerical results for limited accurate predictions

This section provides numerical results to complement our theoretical analysis by comparing with MPC and suboptimal MPC. We first briefly explain how to apply MPC and suboptimal MPC to our SOCO problem, then test the algorithms by considering Example 2.2. We also provide a special case to demonstrate the worst-case performance as analyzed in Section 2.2.4.



(a) Smoothed regression

(b) A special example

Figure 2.2: Algorithms' regret comparison for different W.

2.2.5.1 Review of MPC and suboptimal MPC when applied on SOCO.

Consider the following MPC algorithm: at stage t, MPC solves the W-stage optimization

$$\min_{x_t \ldots x_{t+W-1} \in \mathbb{X}} \sum_{\tau=t}^{t+W-1} (f_\tau(x_\tau) + d(x_\tau, x_{\tau-1})\|^2), \tag{2.10}$$

obtains the optimizer $\{x_t^t, \ldots, x_{t+W-1}^t\}$, and implements x_t^t. We solve (2.10) by iterating NAG, which requires W gradient evaluations per iteration since (2.10) is a W-stage optimization problem. As for faster algorithms, though there are many fast MPC methods and various ways to exploit the structures of MPC to speed up the computation,

we limit ourselves to suboptimal MPC, which terminates the NAG iterations before the convergence. We call this method as subMPC-NAG and call suboptimal MPC based on GD as subMPC-GD. As a fair comparison, we let suboptimal MPC use the same number of gradient evaluations, the same stepsizes and initial values as RHGD and RHAG. Further, we let suboptimal MPC warm start the NAG/GD iterations by the previous computation and OGD at each stage t.

2.2.5.2 Smoothed regression.

Consider Example 2.2 with a logistic regression loss [102] with an l_2 regularizer: $f_t(x_t) = \frac{1}{M}\sum_{m=1}^{M}[\log(1+e^{w_{t,m}^\top x_t}) - v_{t,m}w_{t,m}^\top x_t] + \frac{r}{2}\|x_t\|^2$, where M is the number of the samples, $w_{t,m} \in \mathbb{R}^n$ are the features, $v_{t,m} \in \{0,1\}$ are the labels, r is the regularization parameter. Specifically, we let $M = 60$, $n = 3$, $T = 60$, $\beta = 5$, and $r = 0.5$. We generate $\{w_{t,m}\}_{m=1}^M$ as i.i.d. Gaussian vectors with mean $\mu_t \mathbf{1}_n$ and covariance $\sigma_t^2 \boldsymbol{I}_n$ for each t. $\{\mu_t\}_{t=1}^T$ are i.i.d. from Unif$[-1,1]$, and $\{\sigma_t^2\}_{t=1}^T$ are i.i.d. from Unif$[0,1]$. We generate $\{v_{t,m}\}_{m=1}^M$ i.i.d. from Bern(p_t), where $\{p_t\}_{t=1}^T$ are i.i.d. from Unif$[0,1]$. Let $x_0 = 0$, $\mathbb{X} = [-1,1]^3$. We choose OGD initialization and select stepsizes by Theorem 2.2.

Figure 2.2(a) compares the regrets of different algorithms. Firstly, the regrets decay linearly on a log scale, indicating the exponential decay with W. Further, RHAG decays faster than RHGD despite some fluctuations caused by the inherent properties of NAG [100], which is consistent with Theorem 2.2. In addition, RHAG generates smaller regrets than subMPC-GD and subMPC-NAG, indicating that RHAG utilizes the computation more effectively to improve the performance than the other gradient-based methods. Finally, MPC enjoys smaller regrets, suggesting the benefits of computation in

non-worst scenarios. More regret analysis on MPC is left for the future.

Table I compares the algorithms' running time. Notice that RHGD, RHAG, subMPC-GD and subMPC-NAG require similar time, which is much smaller than the time of MPC. This is because our algorithms and suboptimal MPC only evaluate $W+1$ gradients per stage, while MPC involves multiple NAG iterations per stage and each iteration requires W gradients. Nevertheless, there might be faster methods or faster MPC's variants, which is beyond the scope of this chapter.

2.2.5.3 A special example.

As indicated in Section 2.2.5, RHAG is near-optimal among all the deterministic online algorithms at least in the special/worst cases. We construct a special case to partially validate this indication by showing that RHAG slightly outperforms MPC in this case. In particular, consider $f_t(x_t) = 0.5(x_t - \xi_t)^2$, $\mathbb{X} = [0, 6]$, $T = 20$. Let $\{\xi_t\}_{t=1}^T$ be $[6, 0, 6, 0, 6, 6, 0, 6, 6, 0, 6, 6,\ 0, 6, 6, 6, 6, 6, 6, 6]$. Let $\beta = 20$ and $\gamma = 0.4$ and choose η and λ by Theorem 2.2. Figure 2.2(b) shows that our gradient-based RHAG achieves slightly better performance than the optimization-based MPC in this case.

2.3 Inaccurate-parametric-prediction case

2.3.1 Problem formulation

This section also considers the SOCO problem defined in Section 2.2.1, but we restrict the time-varying cost functions $f_t(x_t)$ to parameter-varying functions, i.e., $f(x_t; \theta_t)$, where $\theta_t \in \Theta \subseteq \mathbb{R}^p$ is a time-varying parameter. We let $\mathsf{C}_\mathsf{T}(\boldsymbol{x}; \boldsymbol{\theta})$ denote the total cost (2.1) for

this model, where $\boldsymbol{\theta} = (\theta_1^\top, \ldots, \theta_T^\top)^\top$. Accordingly, the dynamic regret with a fixed $\boldsymbol{\theta}$ is $\text{Regret}^d = \mathsf{C}_\mathsf{T}(\boldsymbol{x}; \boldsymbol{\theta}) - \min_{x_1,\ldots,x_T \in \mathbb{X}} \mathsf{C}_\mathsf{T}(\boldsymbol{x}; \boldsymbol{\theta})$. We only consider parameter-varying cost functions in this model because it is easier to represent prediction errors of parameters than of functions and because it is consistent with most literature [46, 76].

Prediction models. In this section, we denote the prediction of the future parameter θ_τ obtained at the beginning of stage t as $\theta_{\tau|t-1}$ for $t \le \tau \le T$. The initial predictions $\theta_{1|0}, \ldots, \theta_{T|0}$ are usually available before the problem starts. We call $\theta_{t|t-k}$ as k-step-ahead predictions of parameter θ_t and let $\delta_t(k)$ denote the k-step-prediction error, i.e.

$$\delta_t(k) := \theta_t - \theta_{t|t-k}, \quad \forall\, 1 \le k \le t. \tag{2.11}$$

For notation's simplicity, we define $\theta_{t|\tau} := \theta_{t|0}$ for $\tau \le 0$, and thus $\delta_t(k) = \delta_t(t)$ for $k \ge t$. Further, we denote the vector of k-step prediction errors of all stages as follows

$$\boldsymbol{\delta}(k) = (\delta_1(k)^\top, \ldots, \delta_T(k)^\top)^\top \in \mathbb{R}^{pT}, \qquad \forall\, 1 \le k \le T. \tag{2.12}$$

It is commonly observed that the number of lookahead steps heavily influences the prediction accuracy and in most cases long-term prediction errors are usually larger than short-term ones.

We will first consider the general prediction errors without additional assumptions on $\delta_t(k)$. Then, we will carry out a more insightful discussion for the case when the prediction error $\|\delta_t(k)\|$ is non-decreasing with the number of look-ahead steps k. Further, it is also commonly observed that the prediction errors are correlated. To study how the correlation among prediction errors affect the algorithm performance, we adopt the stochastic model in [46] and analyze the performance under this model.

Protocols of SOCO with inaccurate-parametric predictions. We summarize the

protocols of our online problem below. We consider that the agent knows the function form $f(\cdot;\cdot)$ a priori. For each stage $t = 1, 2, \ldots, T$, the agent

1) receives the predictions $\theta_{t|t-1}, \ldots, \theta_{T|t-1}$ at the beginning of stage;[7]

2) selects x_t based on the predictions and the history, i.e. $\theta_1, \ldots, \theta_{t-1}, \theta_{t|t-1}, \ldots, \theta_{T|t-1}$;

3) suffers $f(x_t;\theta_t) + d(x_t, x_{t-1})$ at the end of stage after true θ_t is revealed.

Lastly, we introduce an additional assumption on this model.

Assumption 2.4. *$\nabla_x f(x;\theta)$ is h-Lipschitz continuous with respect to θ for any x, i.e.*

$$\|\nabla_x f(x;\theta_1) - \nabla_x f(x;\theta_2)\| \leq h\|\theta_1 - \theta_2\|, \ \forall\, x \in \mathbb{X}, \ \theta_1, \theta_2 \in \Theta.$$

Assumption 2.4 ensures that a small prediction error on θ only causes a small error in the gradient estimation, otherwise, little can be achieved with noisy predictions. Besides, we note that the assumption is only for theoretical analysis and our algorithm can still be implemented even without this assumption.

2.3.2 Online algorithm design

We revise RHGD in Section 2.2.2 to cope with the inaccurate-parametric-prediction model in this section. We call the revised version as Receding Horizon Inexact Gradient (RHIG), which is presented in Algorithm 3. The main idea behind the revision is quite straightforward. Since we do not know the accurate future parameter θ_τ but only know an inaccurate prediction $\theta_{\tau|t-1}$ in this section, we use the inaccurate prediction $\theta_{\tau|t-1}$ to

[7] If only W-step-ahead predictions are received, we define $\theta_{t+\tau|t-1} := \theta_{t+W-1|t-1}$ for $\tau \geq W$.

estimate the future partial gradients, i.e.,

$$g_\tau(x_{\tau-1:\tau+1};\theta_{\tau|t-1}) = \nabla_{x_\tau} f(x_\tau;\theta_{\tau|t-1}) + \nabla_{x_\tau} d(x_\tau, x_{\tau-1}) + \nabla_{x_\tau} d(x_{\tau+1}, x_\tau)\mathbb{I}_{(\tau\leq T-1)},$$

which will be used in the online gradient updates (Line 7 of Algorithm 3).

Algorithm 3 Receding Horizon Inexact Gradient (RHIG)

1: **Inputs:** The length of the lookahead horizon: $W \geq 0$; initial decision x_0; stepsize η; initialization oracle ϕ. Let $x_1(0) = x_0$.
2: **for** $t = 2 - W$ to T **do**
3: **if** $t + W \leq T$ **then**
4: Compute $x_{t+W}(0)$ by the initialization oracle ϕ with inexact information.
5: **for** $\tau = \min(t + W - 1, T)$ **downto** $\max(t, 1)$ **do**
6: Compute $x_\tau(t + W - \tau)$ by inexact partial gradient, where $k = t + W - \tau$,

$$x_\tau(k) = \Pi_{\mathbb{X}}[x_\tau(k-1) - \eta g_\tau(x_{\tau-1:\tau+1}(k-1);\theta_{\tau|t-1})], \tag{2.13}$$

7: Output the decision $x_t(W)$ when $1 \leq t \leq T$.

Next, we provide more discussions on the differences between RHIG and RHGD.

- (Line 1) Unlike RHGD, the lookahead horizon length $W \geq 0$ is a tunable algorithm parameter in RHIG. When selecting $W = 0$, RHIG does not use any predictions in Line 5-7. When selecting $1 \leq W \leq T$, RHIG utilizes at most W-step-ahead predictions $\{\theta_{\tau|t-1}\}_{\tau=t}^{t+W-1}$ in Line 5-7. Specifically, when $W = T$, RHIG utilizes all the future predictions $\{\theta_{\tau|t-1}\}_{\tau=t}^{T}$. Interestingly, one can also select $W > T$. In this case, RHIG not only utilizes all the predictions but also conducts more computation based on the initial predictions $\{\theta_{\tau|0}\}_{\tau=1}^{T}$ at $t \leq 0$ (recall that $\theta_{\tau|t-1} = \theta_{\tau|0}$ when $t \leq 0$). Notably, when $W \to +\infty$, RHIG essentially solves $\arg\min_{\boldsymbol{x}\in\mathbb{X}^T} C(\boldsymbol{x}; \{\theta_{\tau|0}\}_{\tau=1}^{T})$ at

$t \leq 0$ to serve as warm starts at $t = 1$.[8] The choice of W will be discussed in the next subsection.

- (Line 5) Notice that the initialization oracle ϕ no longer receives θ_{t+W-1} exactly in RHIG, so OCO algorithms need to be modified here. For example, OGD initializes $x_{t+W}(0)$ by prediction $\theta_{t+W-1|t-1}$:

$$x_\tau(0) = \Pi_{\mathbb{X}}[x_{\tau-1}(0) - \xi_\tau \nabla_{x_{\tau-1}} f(x_{\tau-1}(0); \theta_{\tau-1|t-1})], \quad \text{where } \tau = t + W. \tag{2.14}$$

  Besides, we note that since $\theta_{\tau|t-1}$ is available, OGD (2.14) can also use $\theta_{\tau|t-1}$ to update $x_\tau(0)$. Similarly, OCO algorithms with predictions, e.g. (A)OMD [47, 101], DMD [103], can be applied.

- (Line 7) Instead of exact offline GD in RHGD, RHIG can be interpreted as inexact offline GD with prediction errors. Especially, (2.13) can be written as $x_\tau(k) = x_\tau(k-1) - \eta \nabla_{x_\tau} C(\boldsymbol{x}(k-1); \boldsymbol{\theta} - \boldsymbol{\delta}(W-k+1))$ by the definition (2.11). More compactly, we can write RHIG updates as

$$\boldsymbol{x}(k) = \Pi_{\mathbb{X}^T} \left[\boldsymbol{x}(k-1) - \eta \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k-1); \boldsymbol{\theta} - \boldsymbol{\delta}(W-k+1))\right], \ \forall\, 1 \leq k \leq W, \tag{2.15}$$

  where $\nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k-1); \boldsymbol{\theta} - \boldsymbol{\delta}(W-k+1))$ is an inexact version of the gradient $\nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k-1); \boldsymbol{\theta})$.

Remark 2.4. *Though the design of RHIG is rather straightforward, both theoretical analysis and numerical experiments show promising performance of RHIG even under poor long-term predictions (Section 2.3.3-2.3.4). Some intuitions are discussed below. By formula (2.15), as the iteration number k increases, RHIG employs inexact gradients*

[8] For more discussion on $W > T$, we refer the reader to [98].

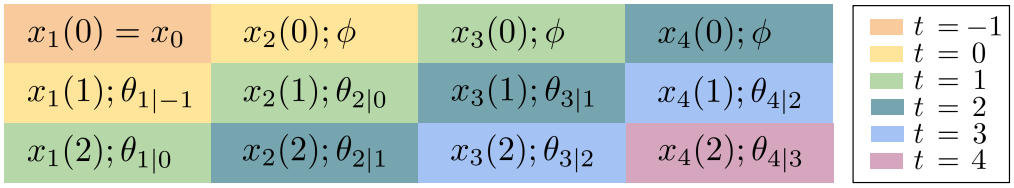


Figure 2.3: Example: RHIG for $W = 2, T = 4$. (Orange) at $t = -1$, let $x_1(0) = x_0$. (Yellow) at $t = 0$, initialize $x_2(0)$ by ϕ, then compute $x_1(1)$ by inexact offline GD (2.13) with prediction $\theta_{1|-1} = \theta_{1|0}$. (Green) At $t = 1$, initialize $x_3(0)$ by ϕ, and update $x_2(1)$ and $x_1(2)$ by (2.13) with $\theta_{2|0}$ and $\theta_{1|0}$ respectively. At $t = 2$, initialize $x_4(0)$ by ϕ, then update $x_3(1)$, $x_2(2)$ by inexact offline GD (2.13) with $\theta_{3|1}$ and $\theta_{2|1}$ respectively. $t = 3, 4$ are similar. Notice that $\boldsymbol{x}(1) = (x_1(1), \ldots, x_4(1))$ is computed by inexact offline gradient with 2-step-ahead predictions, and $\boldsymbol{x}(2)$ by 1-step-ahead predictions.

with shorter-term prediction errors $\boldsymbol{\delta}(W - k + 1)$. Since shorter-term predictions are often more accurate than the longer-term ones, RHIG gradually utilizes more accurate gradient information as iterations go on, reducing the optimality gap caused by inexact gradients. Further, the longer-term prediction errors used at the first several iterations are compressed by later gradient updates, especially for strongly convex costs where GD enjoys certain contraction properties.

2.3.3 Theoretical results

In this section, we provide theoretical results for RHIG under the inaccurate-parametric-prediction model presented in Section 2.2.2. All the proofs are deferred to Appendix A.1. We will first consider the general prediction errors without additional assumptions on $\delta_t(k)$. Then, we will carry out a more insightful discussion for the case when the prediction error $\|\delta_t(k)\|$ is non-decreasing with the number of look-ahead steps k.

Finally, we introduce the stochastic prediction model in [46] to analyze our algorithm's performance under correlated prediction errors.

2.3.3.1 The general parametric model in Section 2.3.1

The next theorem provides a general regret bound for RHIG with any initialization ϕ.

Theorem 2.5 (General Regret Bound). *Under Assumptions 2.1 and2.4, for $W \geq 0$, oracle ϕ, $\eta = \frac{1}{2L}$, we have*

$$\text{Regret}^d(RHIG) \leq \frac{2L}{\alpha}\rho^W \text{Regret}^d(\phi) + \zeta \sum_{k=1}^{\min(W,T)} \rho^{k-1}\|\boldsymbol{\delta}(k)\|^2 + \mathbb{I}_{(W>T)}\frac{\rho^T - \rho^W}{1-\rho}\zeta\|\boldsymbol{\delta}(T)\|^2, \tag{2.16}$$

where $\rho = 1 - \frac{\alpha}{4L}$, $\zeta = \frac{h^2}{\alpha} + \frac{h^2}{2L}$, $\text{Regret}^d(\phi) = C(\boldsymbol{x}(0); \boldsymbol{\theta}) - C(\boldsymbol{x}^*; \boldsymbol{\theta})$ and $\boldsymbol{x}(0)$ is computed by ϕ.

The proof is provided in Appendix A.1. The regret bound (2.16) consists of three terms. The first term $\frac{2L}{\alpha}\rho^W \text{Regret}^d(\phi)$ depends on ϕ. The second term $\zeta \sum_{k=1}^{\min(W,T)} \rho^{k-1}\|\boldsymbol{\delta}(k)\|^2$ and the third term $\mathbb{I}_{(W>T)}\frac{\rho^T-\rho^W}{1-\rho}\zeta\|\boldsymbol{\delta}(T)\|^2$ depend on the errors of the predictions used in Algorithm 3 (Line 5-7). Specifically, when $W \leq T$, at most W-step-ahead predictions are used, so the second term involves at most W-step-ahead prediction errors $\{\boldsymbol{\delta}(k)\}_{k=1}^W$ (the third term is irrelevant). When $W > T$, RHIG uses all predictions, so the second term includes all prediction errors $\{\boldsymbol{\delta}(k)\}_{k=1}^T$. Besides, RHIG conducts more computation by the initial predictions $\{\theta_{t|0}\}_{t=1}^T$ at $t \leq 0$, causing the third term on the initial prediction error $\|\boldsymbol{\delta}(T)\|^2$.

An example of ϕ: restarted OGD [20]. For more concrete discussions on the regret bound, we consider a specific ϕ, restarted OGD [20], as reviewed below. Consider an

epoch size Δ and divide T stages into $\lceil T/\Delta \rceil$ epochs with size Δ. In each epoch k, restart OGD (2.14) and let $\xi_t = \frac{4}{\alpha j}$ at $t = k\Delta + j$ for $1 \leq j \leq \Delta$. Similar to [20], we define the variation of the environment as $V_T = \sum_{t=1}^{T} \sup_{x \in \mathbb{X}} |f(x;\theta_t) - f(x;\theta_{t-1})|$, and consider V_T is known and $1 \leq V_T \leq T$.[9]

Theorem 2.6 (Regret bound of restarted OGD)**.** *Under the conditions in Theorem 2.5 and under Assumption 2.2, consider $T > 2$ and $\Delta = \lceil \sqrt{2T/V_T} \rceil$, the initialization based on restarted OGD described above satisfies the regret bound:*

$$\text{Regret}^d(OGD) \leq C_1 \sqrt{V_T T} \log(1 + \sqrt{T/V_T}) + \frac{h^2}{\alpha} \|\boldsymbol{\delta}(\min(W,T))\|^2, \tag{2.17}$$

where $C_1 = \frac{4\sqrt{2}G^2}{\alpha} + \frac{32\sqrt{2}\beta G^2}{\alpha^2} + 20$.

The proof is in Appendix A.1. Notice that restarted OGD's regret bound (2.17) consists of two terms: the first term $C_1\sqrt{V_T T}\log(1 + \sqrt{T/V_T})$ is consistent with the original regret bound in [20] for strongly convex costs, which increases with the environment's variation V_T; the second term depends on the $\min(W,T)$-step prediction error, which is intuitive since OGD (2.14) in our setting only has access to the inexact gradient $\nabla_{x_{s-1}} f(x_{s-1}(0); \theta_{s-1|s-W-1})$ predicted by the $\min(W,T)$-step-ahead prediction $\theta_{s-1|s-W-1}$.[10]

Corollary 2.2 (RHIG with restarted OGD initialization)**.** *Under the conditions in*

[9] This is without loss of generality. When V_T is unknown, we can use doubling tricks and adaptive stepsizes to generate similar bounds [101]. $1 \leq V_T \leq T$ can be enforced by defining a proper θ_0 and by normalization.

[10] We have this error term because we do not impose the stochastic structures of the gradient errors in [20].

Theorem 2.5 and 2.6, RHIG with ϕ based on restarted OGD satisfies

$$\text{Regret}^d(RHIG) \leq \underbrace{\rho^W \frac{2L}{\alpha} C_1 \sqrt{V_T T} \log(1+\sqrt{T/V_T})}_{\text{Part I}} + \underbrace{\frac{2L}{\alpha}\frac{h^2}{\alpha}\rho^W \|\boldsymbol{\delta}(\min(W,T))\|^2 + \sum_{k=1}^{\min(W,T)} \zeta \rho^{k-1} \|\boldsymbol{\delta}(k)\|^2 + \mathbb{I}_{(W>T)} \frac{\rho^T - \rho^W}{1-\rho} \zeta \|\boldsymbol{\delta}(T)\|^2}_{\text{Part II}}$$

where $\rho = 1 - \frac{\alpha}{4L}$, $\zeta = \frac{h^2}{\alpha} + \frac{h^2}{2L}$, and C_1 is defined in Theorem 2.6.

Next, we discuss the regret bound in Corollary 2.2, which consists of two parts: Part I involves the variation of the environment V_T and Part II involves the prediction errors $\{\boldsymbol{\delta}(k)\}_{k=1}^{\min(W,T)}$.

Impact of V_T. With a fixed V_T, Part I decays exponentially with the lookahead window W. This suggests that the impact of the environment variation V_T on the regret bound decays exponentially as the lookahead window W increases, which is intuitive since long-term thinking/planning allows early preparation for future changes and thus mitigates the negative impact of the environment variation.

Impact of $\boldsymbol{\delta}(k)$. Part II in Corollary 2.2 consists of the prediction error terms in (2.17) and the prediction error terms in Theorem 2.5. Notably, for both $W \leq T$ and $W > T$, the factor in front of $\|\boldsymbol{\delta}(k)\|^2$ is dominated by ρ^{k-1} for $1 \leq k \leq \min(W,T)$, which decays exponentially with k since $0 \leq \rho < 1$. This property suggests that the impact of the total k-step-ahead prediction error $\|\boldsymbol{\delta}(k)\|^2$ on RHIG's regret bound decays exponentially with k, which is intuitive since RHIG (implicitly) focuses more on the shorter-term predictions by using shorter-term predictions in the later iterations of inexact gradient updates. This property also indicates desirable online performance in practice since short-term predictions are usually more accurate and reliable than the long-term ones.

The order of the regret bound. Based on the discussions above, the regret bound in Corollary 2.2 can be summarized as $\tilde{O}(\rho^W\sqrt{V_T T} + \sum_{k=1}^{\min(W,T)} \rho^{k-1}\|\boldsymbol{\delta}(k)\|^2)$. The prediction errors $\|\boldsymbol{\delta}(k)\|^2$ can be either larger or smaller than V_T as mentioned in [101]. When $V_T = o(T)$ and $\|\boldsymbol{\delta}(k)\|^2 = o(T)$ for $k \leq W$, the regret bound of RHIG is sublinear in T.[11]

Choices of W. The optimal choice of W depends on the trade-off between V_T and the prediction errors. For more insightful discussions, we consider non-decreasing k-step-ahead prediction errors, i.e. $\|\boldsymbol{\delta}(k)\| \geq \|\boldsymbol{\delta}(k-1)\|$ for $1 \leq k \leq T$ (in practice, longer-term predictions usually suffer worse quality). It can be shown that Part I increases with V_T and Part II increases with the prediction errors. Further, as W increases, Part I decreases but Part II increases.[12] Thus, when Part I dominates the regret bound, i.e., V_T is large when compared with the prediction errors, selecting a large W reduces the regret bound. On the contrary, when Part II dominates the regret bound, i.e., the prediction errors are large when compared with V_T, a small W is preferred. The choices of W above are quite intuitive: when the environment is drastically changing while the predictions roughly follow the trends, one should use more predictions to prepare for future changes; however, with poor predictions and slowly changing environments, one can ignore most predictions and rely on the understanding of the current environment. Lastly, though we only consider RHIG with restarted OGD, the discussions provide insights for other ϕ.

[11] For example, consider θ_{t-1} as the prediction of θ_{t+k} at time t for $k \geq 0$, then $\|\boldsymbol{\delta}(k)\|^2 = O(\sum_{t=1}^{T} \|\theta_t - \theta_{t-k-1}\|^2)$. If Θ is bounded and $\sum_{t=1}^{T} \|\theta_t - \theta_{t-1}\| = o(T)$, then $\|\boldsymbol{\delta}(k)\|^2 = o(T)$. Further, if $f(x;\theta)$ is Lipschitz continuous with respect to θ, then $V_T = o(T)$. In this case, the regret of RHIG is $o(T)$.

[12] All the monotonicity claims above are verified in [98].

2.3.3.2 The stochastic model in [46, 76]

In many applications, prediction errors are usually correlated. For example, the predicted market price of tomorrow usually relies on the predicted price of today, which also depends on the price predicted yesterday. Motivated by this, we adopt an insightful and general stochastic model on prediction errors, which was originally proposed in [46]:

$$\delta_t(k) = \theta_t - \theta_{t|t-k} = \sum_{s=t-k+1}^{t} P(t-s)e_s, \quad \forall\, 1 \le k \le t \tag{2.18}$$

where $P(s) \in \mathbb{R}^{p\times q}$, $e_1, \ldots, e_T \in \mathbb{R}^q$ are independent with zero mean and covariance R_e. Model (2.18) captures the correlation patterns described above: the errors $\delta_t(k)$ of different predictions on the same parameter θ_t are correlated by sharing common random vectors from $\{e_t, \ldots, e_{t-k+1}\}$; and the prediction errors generated at the same stage, i.e. $\theta_{t+k} - \theta_{t+k|t-1}$ for $k \ge 0$, are correlated by sharing common random vectors from $\{e_t, \ldots, e_{t+k}\}$. Notably, the coefficient matrix $P(k)$ represents the degree of correlation between the $\delta_t(1)$ and $\delta_t(k)$ and between $\theta_t - \theta_{t|t-1}$ and $\theta_{t+k} - \theta_{t+k|t-1}$.

As discussed in [46, 76], the stochastic model (2.18) enjoys many applications, e.g. Wiener filters, Kalman filters [104]. For instance, suppose the parameter follows a stochastic linear system: $\theta_t = \gamma\theta_{t-1} + e_t$ with a given θ_0 and random noise $e_t \sim N(0,1)$. Then $\theta_t = \gamma^k\theta_{t-k} + \sum_{s=t-k+1}^{t} \gamma^{t-s}e_s$, the optimal prediction of θ_t based on θ_{t-k} is $\theta_{t|t-k} = \gamma^k\theta_{t-k}$, the prediction error $\delta_t(k)$ satisfies the model (2.18) with $P(t-s) = \gamma^{t-s}$. A large γ causes strong correlation among prediction errors.

Our next theorem bounds the expected regret of RHIG by the degree of correlation $\|P(k)\|_F$.

Theorem 2.7 (Expected regret bound)**.** *Under the conditions in Theorem 2.5, with*

initialization method ϕ, we have

$$\mathbb{E}[\text{Regret}^d(RHIG)] \leq \frac{2L}{\alpha}\rho^W \mathbb{E}[\text{Reg}(\phi)] + \sum_{t=0}^{\min(W,T)-1} \zeta\|R_e\|(T-t)\|P(t)\|_F^2 \frac{\rho^t - \rho^W}{1-\rho}$$

where the expectation is taken with respect to $\{e_t\}_{t=1}^T$, $\rho = 1 - \frac{\alpha}{4L}$, $\zeta = \frac{h^2}{\alpha} + \frac{h^2}{2L}$.

The first term in Theorem 2.7 represents the influence of ϕ while the second term captures the effects of the correlation. We note that the t-step correlation $\|P(t)\|_F^2$ decays exponentially with t in the regret bound, indicating that RHIG efficiently handles the strong correlation among prediction errors.

Next, we provide a regret bound when RHIG employs the restarted OGD oracle as in Section 2.3.2. Similarly, we consider a known $\mathbb{E}[V_T]$ and $1 \leq V_T \leq T$ for technical simplicity.

Corollary 2.3 (RHIG with restarted OGD)**.** *Under Assumptions 2.1, 2.2, and 2.4, with stepsize $\eta = 1/L$, consider the restarted OGD with $\Delta = \lceil\sqrt{2T/\mathbb{E}[V_T]}\rceil$, we obtain*

$$\mathbb{E}[\text{Regret}^d(RHIG)] \leq \rho^W C_2 \sqrt{\mathbb{E}[V_T]T}\log(1+\sqrt{T/\mathbb{E}[V_T]}) + \sum_{t=0}^{\min(W,T)-1} \zeta\|R_e\|(T-t)\|P(t)\|_F^2 \frac{\rho^t}{1-\rho},$$

where we define $C_2 = \frac{2LC_1}{\alpha}$ and C_1 is defined in Theorem 2.6.

Notice that large W is preferred with a large environment variation and weakly correlated prediction errors, and vice versa.

Next, we discuss the concentration property. For simplicity, we consider Gaussian vectors $\{e_t\}_{t=1}^T$.[13]

[13]Similar results can be obtained for sub-Gaussian random vectors.

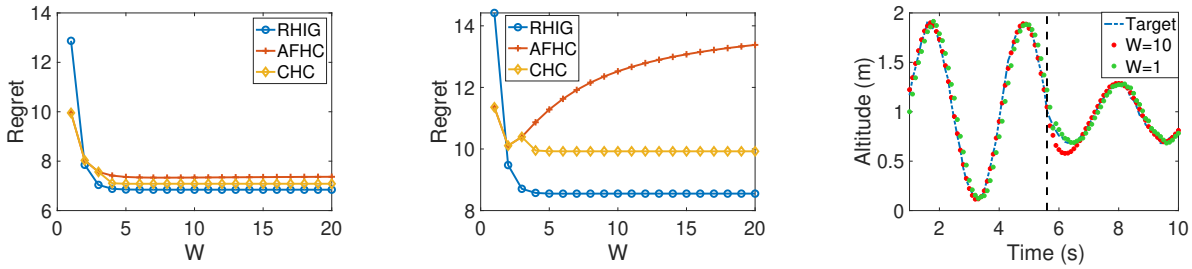
Theorem 2.8 (Concentration bound). *Under the conditions in Corollary 2.3, let* $\mathbb{E}[\text{RegBdd}]$ *denote the expected regret bound in Corollary 2.3 with* $\mathbb{E}[V_T] = T$*, then we have*

$$\mathbb{P}(\text{Regret}^d(RHIG) \geq \mathbb{E}[\text{Regbdd}] + b) \leq \exp\left(-c\min\left(\frac{b^2}{K^2}, \frac{b}{K}\right)\right), \quad \forall\ b > 0,$$

where $K = \zeta \sum_{t=0}^{\min(T,W)-1} \|R_e\|(T-t)\|P(t)\|_F^2 \frac{\rho^t}{1-\rho}$ *and* c *is an absolute constant.*

Theorem 2.8 shows that the probability of the regret being larger than the expected regret by $b > 0$ decays exponentially with b when $\mathbb{E}[V_T] = T$, indicating RHIG's concentration property. Further, the concentration effect is stronger (i.e., a larger $1/K$) with a smaller degree of correlation $\|P(t)\|_F^2$.

2.3.4 Numerical results



(a) Small prediction errors (b) Large prediction errors (c) Quadrotor tracking

Figure 2.4: (a) and (b): the regrets of RHIG, AFHC and CHC. (c): RHIG's tracking trajectories.

(i) A high-level planning problem. We consider SOCO: $\min \sum_{t=1}^{T} \frac{1}{2}(\alpha(x_t - \theta_t)^2 + \beta(x_t - x_{t-1})^2)$, where x_t is quadrotor's altitude, θ_t is target's altitude, and $(x_t - x_{t-1})^2$ penalizes a sudden change in the quadrotor's altitude. The target θ_t follows: $\theta_t = y_t + d_t$, where $y_t = \gamma y_{t-1} + e_t$ is an autoregressive process with noise e_t [105] and $d_t = a\sin(\omega t)$

is a periodic signal. The predictions are the sum of d_t and the optimal predictions of y_t. Notice that a large γ indicates worse long-term predictions. We consider both a small $\gamma = 0.3$ and a large $\gamma = 0.7$ for different error levels. We compare RHIG with AFHC [17, 46] and CHC [76]. The parameters are: $e_t \sim N(0, 1)$ i.i.d., $T = 20$, $\alpha = 1$, $\beta = 0.5$, $x_0 = 10$, $a = 4$, $\omega = 0.5$, $\eta = 0.5$, $\xi_t = 1$, CHC's commitment level $v = 3$. The regret is averaged over 200 iterations

Figure 2.4(a) shows that with small prediction errors, the three algorithms perform similarly well and RHIG is slightly better. Figure 2.4(b) shows that with large prediction errors, RHIG significantly outperforms AFHC and CHC. Some intuitive explanations are provided below. Firstly, AFHC and CHC are optimization-based methods, while our RHIG is based on gradient descent, which is known to be more robust to errors. Secondly, RHIG implicitly reduces the impact of the (poorer-quality) long-term predictions and focuses more on the (better) short-term ones by using long-term predictions in the first several updates and then using short-term ones in later updates to refine the decisions; while AFHC and CHC treat predictions more equally by taking averages of the optimal solutions computed by both long-term and short-term predictions (see [46, 76] for more details). These two intuitive reasons may explain the better numerical performance of our RHIG when compared with AFHC and CHC.

(ii) A physical tracking problem. We consider a simplified second-order model of quadrotor vertical flight: $\ddot{x} = k_1 u - g + k_2$, where $x, \dot{x}, \ddot{x}$ are the altitude, velocity and acceleration respectively, u is the control input (motor thrust command), g is the gravitational acceleration, k_1 and k_2 are physical parameters. We consider cost function $\sum_{t=1}^{T} \frac{1}{2}(\alpha(x_t - \theta_t)^2 + \beta u_t^2)$. The target θ_t follows the process in (i), but with a sudden change in d_t at $t_c = 5.6$s, causing large prediction errors at around

t_c, which is unknown until t_c. We consider a discrete-time version of the system above $\frac{x_{t+1}-2x_t+x_{t-1}}{\Delta^2} = k_1 u_t - g + k_2$. We can write the cost function by states: $\frac{\alpha}{2}(x_t - \theta_t)^2 + \frac{\beta}{2}\frac{1}{k_1^2}(\frac{x_{t+1}-2x_t+x_{t-1}}{\Delta^2} - (-g + k_2))^2$. Notice that the switching cost is not $d(x_t, x_{t-1})$ but $d(x_{t+1}, x_t, x_{t-1})$, but we still have the local coupling property so we can still apply RHIG. The parameters are provided below. Consider horizon 10s and time discretization $\Delta = 0.1$s. Let $k_1 = 1$, $k_2 = 1$, $\alpha = 1$, $\beta = 1 \times 10^{-5}$, $x_0 = 1$m, $g = 9.8\text{m/s}^2$. Let $e_t \sim N(0, 0.5^2)$ i.i.d.. Let $d_t = 0.9\sin(0.2t) + 1$ before $t \leq 5.6$s and $d_t = 0.3\sin(0.2t) + 1$ afterwards. Let $\gamma = 0.6$, $\xi_t = 1$, $\eta = 1/L$, $L \approx 2.6$.

Figure 2.4(c) plots the quadrotor's trajectories generated by RHIG with $W = 1, 10$ and shows RHIG's nice tracking performance even when considering physical dynamics. $W = 10$ performs better first by using more predictions. However, right after t_c, $W = 1$ performs better since the poor prediction quality there degrades the performance. Lastly, the trajectory with $W = 10$ quickly returns to the desired one after t_c, showing the robustness of RHIG to prediction error shocks.

2.4 Conclusion

This chapter studies SOCO with predictions. We propose a gradient-based online algorithm, RHGD, and its variant forms RHAG and RHIG. We analyze our online algorithms based on the following prediction error models: (i) no-prediction case, (ii) limited-accurate prediction case, (iii) inaccurate parametric predictions, (iv) stochastic parametric prediction errors. We provide dynamic regret upper bounds of our online algorithms for the prediction models above. We also discuss the fundamental lower bounds for the model (i)-(ii). Lastly, we test our algorithms with numerical experiments.

Chapter 3 | Online Optimal Control with Predictions

This chapter studies the online optimal control problem with time-varying convex stage costs and a time-invariant linear dynamical system, where a finite lookahead window of accurate predictions of the stage costs are available at each time. This problem can be viewed as an extension to the smoothed online convex optimization problem with limited-accurate predictions considered in Chapter 2.2. We design online algorithms, Receding Horizon Gradient-based Control (RHGC), based on RHGD designed in Chapter 2.2.2. We study the dynamic regret upper bounds of RHGC. Further, we provide a fundamental limit on the dynamic regret by considering linear quadratic tracking. Finally, we numerically test our algorithms.

Chapter outline: In Section 3.1, we motivate our problem and discuss our contributions and related work. Section 3.2 formally presents our problem formulation and introduces useful preliminary results for our algorithm design. Section 3.3 describes our online control algorithm RHGD and an improved version RHTM. Section 3.4 provides dynamic regret upper bounds for RHGD and RHTM. Section 3.5 focuses on a linear quadratic tracking problem, where we provide a fundamental lower bound and compare it with our regret upper bound. Section 3.6 provides numerical results on the comparison between our algorithms with a suboptimal MPC method.

3.1 Introduction

In this chapter, we consider an N-horizon discrete-time sequential decision-making problem. At each time $t = 0, \ldots, N-1$, the decision maker observes a state x_t of a dynamical system, receives a W-step lookahead window of future cost functions of states and control actions, i.e. $f_t(x)+g_t(u), \ldots, f_{t+W-1}(x)+g_{t+W-1}(u)$, then decides the control input u_t which drives the system to a new state x_{t+1} following some known dynamics. For simplicity, we consider a linear time-invariant (LTI) system $x_{t+1} = Ax_t + Bu_t$ with (A, B) known in advance. The goal is to minimize the overall cost over the N time steps. This problem enjoys many applications in, e.g. data center management [106,107], robotics [9], autonomous driving [4, 23], energy systems [3], manufacturing [7, 8]. Hence, there has been a growing interest on the problem, from both control and online optimization communities.

In the control community, studies on the above problem focus on economic model predictive control (EMPC), which is a variant of model predictive control (MPC) with a primary goal on optimizing economic costs [24–31]. Recent years have seen a lot of attention on the optimality performance analysis of EMPC, under both time-invariant costs [32–34] and time-varying costs [27, 29, 35–37]. However, most studies focus on asymptotic performance and there is still limited understanding on the non-asymptotic performance, especially under time-varying costs. Moreover, for computationally efficient algorithms, e.g. suboptimal MPC and inexact MPC [38–41], there is limited work on the optimality performance guarantee.

In online optimization, on the contrary, there are many papers on the non-

asymptotic performance analysis, where the performance is usually measured by regret, e.g., static regrets [14, 15], dynamic regrets [101], etc., but most work does not consider predictions and/or dynamical systems. Further, motivated by the applications with predictions, e.g. predictions of electricity prices in data center management problems [17, 22], there is a growing interest on the effect of predictions on the online problems [17, 22, 46, 47, 76, 85, 108]. However, though some papers consider switching costs which can be viewed as a simple and special dynamical model [72, 108], there is a lack of study on the general dynamical systems and on how predictions affect the online problem with dynamical systems.

In this chapter, we propose gradient-based online control algorithms, receding horizon gradient-based control (RHGC), and provide nonasymptotic optimality guarantees by dynamic regrets. RHGC can be based on many gradient methods, e.g. vanilla gradient descent, Nesterov's accelerated gradient, triple momentum, etc., [100, 109]. Due to the space limit, this chapter only presents receding horizon gradient descent (RHGD) and receding horizon triple momentum (RHTM). For the theoretical analysis, we assume strongly convex and smooth cost functions, whereas applying RHGC does not require these conditions. Specifically, we show that the regret bounds of RHGD and RHTM decay exponentially with the prediction window's size W, demonstrating that our algorithms efficiently utilize the prediction. Besides, our regret bounds decrease when the system is more "agile" in the sense of a controllability index [110]. Further, we provide a fundamental limit for any online control algorithms and show that the fundamental lower bound almost matches the regret upper bound of RHTM. This indicates that RHTM achieves near-optimal performance at least in the worst case. We also provide some discussion on the classic linear quadratic tracking problems, a widely studied control

problem in literature, to provide more insightful interpretations of our results. Finally, we numerically test our algorithms. In addition to linear systems, we also apply RHGC to a nonlinear dynamical system: path tracking by a two-wheeled robot. Results show that RHGC works effectively for nonlinear systems though RHGC is only presented and theoretical analyzed on LTI systems.

Results in this chapter are built on our work on online optimization with switching costs in Chapter 2. Compared with [108], this chapter studies online optimal control with *general linear dynamics*, which includes [108] as a special case; and studies how the system controllability index affects the regrets.

3.1.1 Additional related work

This chapter is based on our work [82]. After the publication of [82], there are several papers studying online optimal control with predictions [111–115]. We would like to mention that [113] provides a more detailed fundamental limit analysis for online linear quadratic tracking. The difference between our lower bound result and [113] is the following: for any controllable linear system, [113] provides a fundamental lower bound; while our result suggests that there always exists a certain system such that our lower bound hold. When considering the same system, our lower bound coincides with [113].

There has been some recent work on online optimal control problems with time-varying costs [42, 72, 116, 117] and/or time-varying disturbances [42], but most papers focus on the no-prediction cases. As we show later in this chapter, these algorithms can be used in our RHGC methods as initialization oracles. Moreover, our regret analysis shows that RHGC can reduce the regret of these no-prediction online algorithms by a factor exponentially decaying with the prediction window's size.

Besides, we would like to mention another related line of work: learning-based control [60, 62, 63, 118]. In some sense, the results in this chapter are orthogonal to that of the learning-based control, because the learning-based control usually considers a time-invariant environment but unknown dynamics, and aims to learn system dynamics or optimal controllers by data; while this chapter considers a time-varying scenario with known dynamics but changing objectives and studies decision making with limited predictions. It is an interesting future direction to combine the two lines of work for designing more applicable algorithms.

Notations.

Consider matrices A and B, $A \geq B$ means $A - B$ is positive semidefinite and $[A, B]$ denotes a block matrix. The norm $\|\cdot\|$ refers to the L_2 norm for both vectors and matrices. Let x^i denote the ith entry of the vector. Consider a set $\mathscr{I} = \{k_1, \ldots, k_m\}$, then $x^{\mathscr{I}} = (x^{k_1}, \ldots, x^{k_m})^\top$, and $A(\mathscr{I}, :)$ denotes the $\mathscr{I}$ rows of matrix A stacked together. Let I_m be an identity matrix in $\mathbb{R}^{m \times m}$.

3.2 Problem formulation and preliminaries

Consider a finite-horizon discrete-time optimal control problem with time-varying cost functions $f_t(x_t) + g_t(u_t)$ and a linear time-invariant (LTI) dynamical system:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{u}} \quad & J(\mathbf{x}, \mathbf{u}) = \sum_{t=0}^{N-1} [f_t(x_t) + g_t(u_t)] + f_N(x_N) \\ \text{s.t.} \quad & x_{t+1} = Ax_t + Bu_t, \quad t \geq 0 \end{aligned} \tag{3.1}$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, $\mathbf{x} = (x_1^\top, \ldots, x_N^\top)^\top$, $\mathbf{u} = (u_0^\top, \ldots, u_{N-1}^\top)^\top$, x_0 is given, $f_N(x_N)$ is the terminal cost.[1] To solve the optimal control problem (3.1), all cost functions from $t = 0$ to $t = N$ are needed. However, at each time t, usually only a finite lookahead window of cost functions are available and the decision maker needs to make an online decision u_t using the available information.

In particular, we consider a simplified prediction model: at each time t, the decision maker obtains accurate predictions for the next W time steps, $f_t, g_t, \ldots, f_{t+W-1}, g_{t+W-1}$, but no further prediction beyond these W steps, meaning that $f_{t+W}, g_{t+W}, \ldots$ can even be adversarially generated. Though this prediction model may be too optimistic in the short term and over pessimistic in the long term, this model i) captures a commonly observed phenomenon in predictions that short-term predictions are usually much more accurate than the long-term predictions; ii) allows researchers to derive insights for the role of predictions and possibly to extend to more complicated cases [17, 22, 119, 120].

Protocols of online optimal control with predictions. At each time $t = 0, 1, \ldots,$

1) the agent observes state x_t and receives prediction $f_t,\ g_t,\ \ldots,\ f_{t+W-1},\ g_{t+W-1}$;

2) the agent decides and implements a control u_t and suffers the cost $f_t(x_t) + g_t(u_t)$;

3) the system evolves to the next state $x_{t+1} = Ax_t + Bu_t$.[2]

An online control algorithm, denoted as $\mathscr{A}$, can be defined as a mapping from the prediction information and the history information to the control action $u_t(\mathscr{A})$:

$$u_t(\mathscr{A}) = \mathscr{A}(x_t(\mathscr{A}), \ldots, x_0(\mathscr{A}), \{f_s, g_s\}_{s=0}^{t+W-1}), \quad t \geq 0, \tag{3.2}$$

[1] The results in this chapter can be extended to cost $c_t(x_t, u_t)$ with proper assumptions.

[2] We assume known A, B, no process noises, state feedback, and leave relaxing assumptions for future.

where $x_t(\mathscr{A})$ is the state generated by implementing $\mathscr{A}$ and $x_0(\mathscr{A}) = x_0$ is given.

This chapter evaluates the performance of online control algorithms by dynamic regret, i.e. comparing it against the optimal control cost J^* in hindsight:

$$\text{Regret}(\mathscr{A}) := J(\mathscr{A}) - J^* = J(\mathbf{x}(\mathscr{A}), \mathbf{u}(\mathscr{A})) - J^*, \tag{3.3}$$

where $J^* \coloneqq \min\{J(\mathbf{x}, \mathbf{u}) \mid x_{t+1} = Ax_t + Bu_t,\ \forall\, t \geq 0\}$.[3]

Example 3.1 (Linear quadratic (LQ) tracking)**.** *Consider a discrete-time tracking problem for a system $x_{t+1} = Ax_t + Bu_t$. The goal is to minimize the quadratic tracking loss of a trajectory $\{\theta_t\}_{t=0}^N$*

$$J(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\sum_{t=0}^{N-1}\left[(x_t - \theta_t)^\top Q_t(x_t - \theta_t) + u_t^\top R_t u_t\right] + \frac{1}{2}(x_N - \theta_N)^\top Q_N(x_N - \theta_N).$$

In practice, it is usually difficult to know the complete trajectory $\{\theta_t\}_{t=0}^N$ a priori, what are revealed are usually the next few steps, making it an online control problem with predictions.

Assumptions and useful concepts. Firstly, we assume controllability, which is standard in control theory and roughly means that the system can be steered to any state by proper control inputs [121].

Assumption 3.1. *The LTI system $x_{t+1} = Ax_t + Bu_t$ is controllable.*

It is well-known that any controllable LTI system can be linearly transformed to a canonical form [110] and the linear transformation can be computed efficiently a priori

[3]The optimality gap depends on initial state x_0 and $\{f_t, g_t\}_{t=0}^N$, but we omit them for simplicity of notation.

using A and B, which can further be used to reformulate the cost functions f_t, g_t. Thus, without loss of generality, this chapter only considers LTI systems in the canonical form, defined as follows.

Definition 3.1 (Canonical form)**.** *A system $x_{t+1} = Ax_t + Bu_t$ is said to be in the canonical form if*

$$A = \begin{bmatrix} 0 & 1 & & & 0 & & & & & & & & & & \\ \vdots & \ddots & \ddots & & & & & & & & & & & & \\ & & 0 & 1 & & & & & & & & & & & \\ * & * & \cdots & * & * & * & \cdots & * & & & \cdots & & & * & \\ & & & & 0 & 1 & & 0 & & & & & & & \\ & & & & \vdots & \ddots & \ddots & & & & & & & & \\ & & & & & & 0 & 1 & & & & & & & \\ * & * & \cdots & * & * & * & \cdots & * & \cdots & * & \cdots & & & * & \\ \cdots & & & & & & & & \cdots & & & & & & \\ & & & & & & & & & 0 & 1 & \cdots & 0 & & \\ & & & & & & & & & \vdots & \ddots & \ddots & & & \\ & & & & & & & & & & & 0 & 1 & & \\ * & * & \cdots & * & * & * & \cdots & * & \cdots & * & * & \cdots & * & & \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & \cdots \\ \vdots & \vdots & \vdots \\ 0 & & \\ 1 & 0 & \cdots \\ 0 & 0 & \\ \vdots & \vdots & \cdots \\ 0 & 1 & \cdots \\ \cdots & & \cdots \\ 0 & \cdots & \cdots \\ \vdots & \ddots & \\ 0 & 0 & \cdots 1 \end{bmatrix},$$

*where each * represents a (possibly) nonzero entry, and the rows of B with 1 are the same rows of A with * and the indices of these rows are denoted as $\{k_1, \ldots, k_m\} =: \mathscr{I}$. Moreover, let $p_i = k_i - k_{i-1}$ for $1 \le i \le m$, where $k_0 = 0$. The* controllability index *of a canonical-form (A, B) is defined as*

$$p = \max\{p_1, \ldots, p_m\}.$$

Next, we introduce assumptions on the cost functions and their optimal solutions.

Assumption 3.2. *Assume f_t is μ_f strongly convex and l_f Lipschitz smooth for $0 \le t \le N$, and g_t is convex and l_g Lipschitz smooth for $0 \le t \le N-1$ for some $\mu_f, l_f, l_g > 0$.*

Assumption 3.3. *Assume the minimizers to f_t, g_t, denoted as $\theta_t = \arg\min_x f_t(x)$, $\xi_t = \arg\min_u g_t(u)$, are uniformly bounded, i.e. there exist $\bar{\theta}, \bar{\xi}$ such that $\|\theta_t\| \le \bar{\theta}$, $\|\xi_t\| \le \bar{\xi}$, $\forall\, t$.*

These assumptions are commonly adopted in convex analysis. The uniform bounds rule out extreme cases. Notice that the LQ tracking problem in Example 1 satisfies Assumption 3.2 and 3.3 if Q_t, R_t are positive definite with uniform bounds on the eigenvalues and if θ_t are uniformly bounded for all t.

3.3 Online control algorithms: receding horizon gradient-based control

This section introduces our online control algorithms RHGC. The design is by first converting the online control problem to an equivalent online optimization problem with *finite temporal-coupling* costs, then designing gradient-based online optimization algorithms by utilizing this finite temporal-coupling property.

3.3.1 Problem transformation

Firstly, we notice that the offline optimal control problem (3.1) can be viewed as an optimization with equality constraints over $\boldsymbol{x}$ and $\boldsymbol{u}$. The individual stage cost $f_t(x_t) + g_t(u_t)$ only depends on the current x_t and u_t but the equality constraints couple x_t, u_t with x_{t+1} for each t. In the following, we will rewrite (3.1) in an equivalent form of an *unconstrained* optimization problem on some entries of x_t for all t, but the new stage cost at each time t will depend on these new entries across a few nearby time steps. We will harness this structure to design our online algorithm.

In particular, the entries of x_t adopted in the reformulation are: $x_t^{k_1}, \dots, x_t^{k_m}$, where $\mathscr{I} = \{k_1, \dots, k_m\}$ is defined in Definition 3.1. For ease of notation, we define

$$z_t \coloneqq (x_t^{k_1}, \dots, x_t^{k_m})^\top, \quad t \geq 0 \tag{3.4}$$

and write $z_t^j = x_t^{k_j}$ where $j = 1, \dots, m$. Let $\boldsymbol{z} \coloneqq (z_1^\top, \dots, \dots, z_N^\top)^\top$. By the canonical-form equality constraint $x_t = Ax_{t-1} + Bu_{t-1}$, we have $x_t^i = x_{t-1}^{i+1}$ for $i \notin \mathscr{I}$, so x_t can be represented by $z_{t-p+1}, \dots, z_t$ in the following way:

$$x_t = (\underbrace{z_{t-p_1+1}^1, \dots, z_t^1}_{p_1}, \underbrace{z_{t-p_2+1}^2, \dots, z_t^2}_{p_2}, \dots, \underbrace{z_{t-p_m+1}^m, \dots, z_t^m}_{p_m})^\top, \quad t \geq 0, \tag{3.5}$$

where z_t for $t \leq 0$ is determined by x_0 in a way to let (3.5) hold for $t = 0$. For ease of exposition and without loss of generality, we consider $x_0 = 0$ in this chapter; then we have $z_t = 0$ for $t \leq 0$. Similarly, u_t can be determined by $z_{t-p+1}, \dots, z_t, z_{t+1}$ by

$$u_t = z_{t+1} - A(\mathscr{I}, :)x_t = z_{t+1} - A(\mathscr{I}, :)(z_{t-p_1+1}^1, \dots, z_t^1, \dots, z_{t-p_m+1}^m, \dots, z_t^m)^\top, \quad t \geq 0 \tag{3.6}$$

where $A(\mathscr{I}, :)$ consists of $k_1, \dots, k_m$ rows of A.

It is straightforward to verify that equations (3.4, 3.5, 3.6) describe a bijective transformation between $\{(\boldsymbol{x}, \boldsymbol{u}) \mid x_{t+1} = Ax_t + Bu_t\}$ and $\boldsymbol{z} \in \mathbb{R}^{mN}$, since the LTI constraint $x_{t+1} = Ax_t + Bu_t$ is naturally embedded in the relation (3.5, 3.6). Therefore, based on the transformation, an optimization problem with respect to $\boldsymbol{z} \in \mathbb{R}^{mN}$ can be designed to be equivalent with (3.1). Notice that the resulting optimization problem has no constraint on $\boldsymbol{z}$. Moreover, the cost functions on $\boldsymbol{z}$ can be obtained by substituting (3.5, 3.6) into $f_t(x_t)$ and $g_t(u_t)$, i.e. $\tilde{f}_t(z_{t-p+1}, \dots, z_t) \coloneqq f_t(x_t)$ and $\tilde{g}_t(z_{t-p+1}, \dots, z_t, z_{t+1}) \coloneqq g_t(u_t)$. Correspondingly, the objective function of the equivalent optimization with respect to $\boldsymbol{z}$ is

$$C(\mathbf{z}) \coloneqq \sum_{t=0}^{N} \tilde{f}_t(z_{t-p+1}, \dots, z_t) + \sum_{t=0}^{N-1} \tilde{g}_t(z_{t-p+1}, \dots, z_{t+1}) \tag{3.7}$$

$C(\boldsymbol{z})$ has many nice properties, some of which are formally stated below.

Lemma 3.1. *The function $C(\boldsymbol{z})$ has the following properties:*

i) $C(\boldsymbol{z})$ *is* $\mu_c = \mu_f$ *strongly convex,* l_c *smooth for* $l_c = pl_f + (p+1)l_g\|[I_m, -A(\mathscr{I}, :)]\|^2$.

ii) For any $(\boldsymbol{x}, \boldsymbol{u})$ *s.t.* $x_{t+1} = Ax_t + Bu_t$, $C(\boldsymbol{z}) = J(\boldsymbol{x}, \boldsymbol{u})$ *where* $\boldsymbol{z}$ *is defined in (3.4). Conversely,* $\forall\, \boldsymbol{z}$, *the* $(\boldsymbol{x}, \boldsymbol{u})$ *determined by (3.5,3.6) satisfies* $x_{t+1} = Ax_t + Bu_t$ *and* $J(\boldsymbol{x}, \boldsymbol{u}) = C(\boldsymbol{z})$;

iii) Each stage cost $\tilde{f}_t + \tilde{g}_t$ *in (3.7) only depends on* $z_{t-p+1}, \ldots, z_{t+1}$.

Property ii) implies that any online algorithm for deciding $\boldsymbol{z}$ can be translated to an online algorithm for $\boldsymbol{x}$ and $\boldsymbol{u}$ by (3.5, 3.6) with the same costs. Property iii) highlights one nice property, finite temporal-coupling, of $C(\boldsymbol{z})$, which serves as a foundation for our online algorithm design.

Example 3.2. *For illustration, consider the following dynamical system with* $n = 2,\ m = 1$*:*

$$\begin{bmatrix} x_{t+1}^1 \\ x_{t+1}^2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ a_1 & a_2 \end{bmatrix} \begin{bmatrix} x_t^1 \\ x_t^2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_t \tag{3.8}$$

Here, $k_1 = 2$, $\mathscr{I} = \{2\}$, $A(\mathscr{I}, :) = (a_1, a_2)$, *and* $z_t = x_t^2$. *By (3.8),* $x_t^1 = x_{t-1}^2$ *and* $x_t = (z_{t-1}, z_t)^\top$. *Similarly,* $u_t = x_{t+1}^2 - A(\mathscr{I}, :)x_t = z_{t+1} - A(\mathscr{I}, :)(z_{t-1}, z_t)^\top$. *Hence,* $\tilde{f}_t(z_{t-1}, z_t) = f_t(x_t) = f_t((z_{t-1}, z_t)^\top)$, $\tilde{g}_t(z_{t-1}, z_t, z_{t+1}) = g_t(u_t) = g_t(z_{t+1} - A(\mathscr{I}, :)(z_{t-1}, z_t)^\top)$.

Remark 3.1. *this chapter considers a reparameterization method with respect to states* $\boldsymbol{x}$ *via the canonical form, and it might be interesting to compare it with the more direct reparameterization with respect to control inputs* $\boldsymbol{u}$. *The control-based reparameterization has been discussed in literature [122]. It has been observed in [122] that when* A *is not stable, the condition number of the cost function derived from the control-based reparameterization goes to infinity as* $W \to +\infty$, *which may result in computation issues*

when W is large. However, the state-based reparameterization considered in this chapter can guarantee bounded condition number for all W even for unstable A, as shown in Lemma 3.1. This is one major advantage of the state-based reparameterization method considered in this chapter.

3.3.2 Online algorithm design: RHGC

This section introduces our RHGC based on the reformulation (3.7) and Chapter 2. As mentioned earlier, any online algorithm for z_t can be translated to an online algorithm for x_t, u_t. Hence, we will focus on designing an online algorithm for z_t in the following. By the finite temporal-coupling property of $C(\boldsymbol{z})$, the partial gradient of the *total cost* $C(\boldsymbol{z})$ only depends on the finite neighboring stage costs $\{\tilde{f}_\tau, \tilde{g}_\tau\}_{\tau=t}^{t+p-1}$ and finite neighboring stage variables $(z_{t-p}, \dots, z_{t+p}) =: z_{t-p:t+p}$.

$$\frac{\partial C}{\partial z_t}(\boldsymbol{z}) = \sum_{\tau=t}^{t+p-1} \frac{\partial \tilde{f}_\tau}{\partial z_t}(z_{\tau-p+1}, \dots, z_\tau) + \sum_{\tau=t-1}^{t+p-1} \frac{\partial \tilde{g}_\tau}{\partial z_t}(z_{\tau-p+1}, \dots, z_{\tau+1})$$

Without causing any confusion, we use $\frac{\partial C}{\partial z_t}(z_{t-p:t+p})$ to denote $\frac{\partial C}{\partial z_t}(\boldsymbol{z})$ for highlighting the local dependence. Thanks to the local dependence, despite the fact that not all the future costs are available, it is still possible to compute the partial gradient of the total cost by using only a finite lookahead window of the cost functions. This observation motivates the design of our receding horizon gradient-based control (RHGC) methods, which are the online implementation of gradient methods, such as vanilla gradient descent, Nesterov's accelerated gradient, triple momentum, etc., [100, 109].

Firstly, we illustrate the main idea of RHGC by receding horizon gradient descent (RHGD) based on vanilla gradient descent. In RHGD (Algorithm 4), index j refers to the iteration number of the corresponding gradient update of $C(\boldsymbol{z})$. There are two

Algorithm 4 Receding Horizon Gradient Descent (RHGD)

inputs: Canonical form (A, B), $W \geq 1$, $K = \lfloor \frac{W-1}{p} \rfloor$, stepsize γ_g, initialization oracle φ.

for $t = 1 - W : N - 1$ **do**

  Step 1: initialize $z_{t+W}(0)$ by oracle φ.

  for $j = 1, \ldots, K$ **do**

    Step 2: update $z_{t+W-jp}(j)$ by gradient descent $z_{t+W-jp}(j) = z_{t+W-jp}(j-1) - \gamma_g \frac{\partial C}{\partial z_{t+W-jp}}(z_{t+W-(j+1)p:t+W-(j-1)p}(j-1))$.

  Step 3: compute u_t by $z_{t+1}(K)$ and the observed state x_t: $u_t = z_{t+1}(K) - A(\mathscr{I}, :)x_t$

major steps to decide z_t. Step 1 is initializing the decision variables $\boldsymbol{z}(0)$. Here, we do not restrict the initialization algorithm φ and allow any oracle/online algorithm without using lookahead information, i.e. $z_{t+W}(0)$ is selected based only on the information up to $t+W-1$: $z_{t+W}(0) = \varphi(\{\tilde{f}_s, \tilde{g}_s\}_{s=0}^{t+W-1})$. One example of φ will be provided in Section 4. Step 2 is using the W-lookahead costs to conduct gradient updates. Notice that the gradient update from $z_\tau(j-1)$ to $z_\tau(j)$ is implemented in a backward order of τ, i.e. from $\tau = t+W$ to $\tau = t$. Moreover, since the partial gradient $\frac{\partial C}{\partial z_t}$ requires the local decision variables $z_{t-p:t+p}$, given W-lookahead information, RHGD can only conduct $K = \lfloor \frac{W-1}{p} \rfloor$ iterations of gradient descent for the total cost $C(\boldsymbol{z})$. For more discussion, we refer the reader to [108] for the $p = 1$ case.

In addition to RHGD, RHGC can also incorporate accelerated gradient methods in the same way, such as Nesterov's accelerated gradient and triple momentum. For the space limit, we only formally present receding horizon triple momentum (RHTM) in Algorithm 5 based on triple momentum [109]. RHTM also consists of two major steps when determining z_t: initialization and gradient updates based on the lookahead

window. The two major differences from RHGD are that the decision variables in RHTM include not only $z(j)$ but also auxiliary variables $\omega(j)$ and $y(j)$, which are adopted in triple momentum to accelerate the convergence, and that the gradient update is by triple momentum instead of gradient descent. Nevertheless, RHTM can also conduct $K = \lfloor \frac{W-1}{p} \rfloor$ iterations of triple momentum for $C(\boldsymbol{z})$ since the triple momentum update requires the same neighboring cost functions.

Though it appears that RHTM does not fully exploit the lookahead information since only a few gradient updates are used, in Section 3.5, we show that RHTM achieves near-optimal performance with respect to W, which means that RHTM successfully extracts and utilizes the prediction information.

Finally, we briefly introduce MPC [123] and suboptimal MPC [38], and compare them with our algorithms. MPC tries to solve a W-stage optimization at each t and implements the first control input. Suboptimal MPC, as a variant of MPC aiming at reducing computation, conducts an optimization method only for a few iterations without solving the optimization completely. Our algorithm's computation time is similar to that of suboptimal MPC with a few gradient iterations. However, the major difference between our algorithm and suboptimal MPC is that suboptimal MPC conducts gradient updates for a truncated W-stage optimal control problem based on W-lookahead information, while our algorithm is able to conduct gradient updates for the complete N-stage optimal control problem based on the same W-lookahead information by utilizing the reformulation (3.4, 3.5, 3.6, 3.7).

3.4 Regret upper bounds

Because our RHTM (RHGD) is designed to exactly implement the triple momentum (gradient descent) of $C(\boldsymbol{z})$ for K iterations, it is straightforward to have the following regret guarantees that connect the regrets of RHTM and RHGD with the regret of the initialization oracle φ,

Algorithm 5 Receding Horizon Triple Momentum (RHTM)

inputs: Canonical form (A, B), $W \geq 1$, $K = \lfloor \frac{W-1}{p} \rfloor$, $\gamma_c, \gamma_z, \gamma_\omega, \gamma_y > 0$, oracle φ.

for $t = 1 - W : N - 1$ **do**

Step 1: initialize $z_{t+W}(0)$ by oracle φ, then let $\omega_{t+W}(-1), \omega_{t+W}(0), y_{t+W}(0)$ be $z_{t+W}(0)$

for $j = 1, \ldots, K$ **do**

Step 2: update $\omega_{t+W-jp}(j), y_{t+W-jp}(j), z_{t+W-jp}(j)$ by triple momentum.

$$\begin{aligned}
\omega_{t+W-jp}(j) &= (1+\gamma_\omega)\omega_{t+W-jp}(j-1) - \gamma_\omega \omega_{t+W-jp}(j-2) \\
&\quad - \gamma_c \frac{\partial C}{\partial y_{t+W-jp}}(y_{t+W-(j+1)p:t+W-(j-1)p}(j-1)) \\
y_{t+W-jp}(j) &= (1+\gamma_y)\omega_{t+W-jp}(j) - \gamma_y \omega_{t+W-jp}(j-1) \\
z_{t+W-jp}(j) &= (1+\gamma_z)\omega_{t+W-jp}(j) - \gamma_z \omega_{t+W-jp}(j-1)
\end{aligned}$$

Step 3: compute u_t by $z_{t+1}(K)$ and the observed state x_t: $u_t = z_{t+1}(K) - A(\mathscr{I}, :)x_t$

Theorem 3.1. *Consider $W \geq 1$ and stepsizes $\gamma_g = \frac{1}{l_c}$, $\gamma_c = \frac{1+\phi}{l_c}$, $\gamma_\omega = \frac{\phi^2}{2-\phi}$, $\gamma_y = \frac{\phi^2}{(1+\phi)(2-\phi)}$, $\gamma_z = \frac{\phi^2}{1-\phi^2}$, $\phi = 1 - 1/\sqrt{\zeta}$, and let $\zeta = l_c/\mu_c$ denote $C(\boldsymbol{z})$'s condition number. For any oracle φ,*

$$\begin{aligned}
\text{Regret}^d(RHGD) &\leq \zeta \left(\frac{\zeta - 1}{\zeta}\right)^K \text{Regret}(\varphi), \\
\text{Regret}^d(RHTM) &\leq \zeta^2 \left(\frac{\sqrt{\zeta} - 1}{\sqrt{\zeta}}\right)^{2K} \text{Regret}(\varphi)
\end{aligned}$$

where $K = \lfloor \frac{W-1}{p} \rfloor$, $\text{Regret}^d(\varphi)$ *is the regret of the initial controller:* $u_t(0) = z_{t+1}(0) - A(\mathscr{I}, :)x_t(0)$.

Theorem 3.1 suggests that for any online algorithm φ without predictions, RHGD and RHTM can use predictions to lower the regret by a factor of $\zeta(\frac{\zeta-1}{\zeta})^K$ and $\zeta^2(\frac{\sqrt{\zeta}-1}{\sqrt{\zeta}})^{2K}$ respectively via additional $K = \lfloor \frac{W-1}{p} \rfloor$ gradient updates. Moreover, the factors decay exponentially with $K = \lfloor \frac{W-1}{p} \rfloor$, and K almost linearly increases with W. This indicates that RHGD and RHTM improve the performance exponentially fast with an increase in the prediction window W for any initialization method. In addition, $K = \lfloor \frac{W-1}{p} \rfloor$ decreases with p, implying that the regrets increase with the controllability index p (Definition 3.1). This is intuitive because p roughly indicates how fast the controller can influence the system state effectively: the larger the p is, the longer it takes. To see this, consider Example 3.2. Since u_{t-1} does not directly affect x_t^1, it takes at least $p = 2$ steps to change x_t^1 to a desirable value. Finally, RHTM's regret decays faster than RHGD's, which is intuitive because triple momentum converges faster than gradient descent. Thus, we will focus on RHTM in the following.

An initialization method: follow the optimal steady state (FOSS). To complete the regret analysis for RHTM, we provide a simple initialization method, FOSS, and its dynamic regret bound. As mentioned before, any online control algorithm without predictions, e.g. [116, 117], can be applied as an initialization oracle φ. However, most literature study static regrets rather than dynamic regrets.

Definition 3.2 (Follow the optimal steady state (FOSS))**.** *The optimal steady state for stage cost* $f(x) + g(u)$ *refers to* $(x^e, u^e) \coloneqq \arg\min_{x=Ax+Bu}(f(x) + g(u))$. *Follow the optimal steady state algorithm (FOSS) first solves the optimal steady state* (x_t^e, u_t^e) *for*

cost $f_t(x)+g_t(u)$*, then determines* z_{t+1} *by* x_t^e*, i.e.* $z_{t+1}=(x_t^{e,k_1},\dots,x_t^{e,k_m})^\top$ *at each* $t+1$*.*

FOSS is motivated by the fact that the optimal steady state cost is the optimal infinite-horizon average cost for LTI systems with time-invariant cost functions [124], so FOSS should yield acceptable performance at least for slowly changing cost functions. Nevertheless, we admit that FOSS is proposed mainly for analytical purposes and other online algorithms may outperform FOSS in various perspectives. The following is a regret bound for FOSS, relying on the solution to Bellman equations.

Definition 3.3 (Solution to the Bellman equations [125])**.** *Consider optimal control problem:* $\min \lim_{N\to+\infty}\frac{1}{N}\sum_{t=0}^{N-1}(f(x_t)+g(u_t))$ *where* $x_{t+1}=Ax_t+Bu_t$*. Let* λ^e *be the optimal steady state cost* $f(x^e)+g(u^e)$*, which is also the optimal infinite-horizon average cost [124]. The Bellman equations for the problem is* $h^e(x)+\lambda^e=\min_u(f(x)+g(u)+h^e(Ax+Bu))$*. The solution to the Bellman equations, denoted by* $h^e(x)$*, is sometimes called as a bias function [125]. To ensure the uniqueness of the solution, some extra conditions, e.g.* $h^e(0)=0$*, are usually imposed.*

Theorem 3.2 (Regret bound of FOSS)**.** *Let* (x_t^e,u_t^e) *and* $h_t^e(x)$ *denote the optimal steady state and the bias function with respect to cost* $f_t(x)+g_t(u)$ *respectively for* $0\le t\le N-1$*. Suppose* $h_t^e(x)$ *exists for* $0\le t\le N-1$*,*[4] *then the regret of FOSS can be bounded by*

$$\text{Regret}(FOSS)=O\left(\sum_{t=0}^{N}(\|x_{t-1}^e-x_t^e\|+h_{t-1}^e(x_t^*)-h_t^e(x_t^*))\right),$$

where $\{x_t^*\}_{t=0}^N$ *denotes the optimal state trajectory for* (3.1)*,* $x_{-1}^e=x_0^*=x_0=0$*,* $h_{-1}^e(x)=0, h_N^e(x)=f_N(x)$*,* $x_N^e=\theta_N$*. Consequently, by Theorem 3.1, the regret bound of RHTM with initialization FOSS is* $\text{Regret}(RHTM)=O\left(\zeta^2(\frac{\sqrt{\zeta}-1}{\sqrt{\zeta}})^{2K}\sum_{t=0}^{N}(\|x_{t-1}^e-x_t^e\|+h_{t-1}^e(x_t^*)-h_t^e(x_t^*))\right)$*.*

[4] h_t^e may not be unique, so extra conditions can be imposed on h_t^e for more interesting regret bounds.

Theorem 3.2 bounds the regret by the variation of the optimal steady states x_t^e and the bias functions h_t^e. If f_t and g_t do not change, x_t^e and h_t^e do not change, yielding a small $O(1)$ regret, i.e. $O(\|x_0^e\| + h_0^e(x_0))$, matching our intuition. Though Theorem 3.2 requires h_t^e exists, the existence is guaranteed for many control problems, e.g. LQ tracking and control problems with turnpike properties [37, 126].

3.5 Linear quadratic tracking: a fundamental limit

To provide more intuitive meaning for our regret analysis in Theorem 3.1 and Theorem 3.2, we apply RHTM to the LQ tracking problem in Example 3.1. Results for the time varying Q_t, R_t, θ_t are provided in Appendix E; whereas here we focus on a special case which gives clean expressions for regret bounds: both an upper bound for RHTM with initialization FOSS and a lower bound for any online algorithm. Further, we show that the lower bound and the upper bound almost match each other, implying that our online algorithm RHTM uses the predictions in a nearly optimal way even though it only conducts a few gradient updates at each time step .

The special case of LQ tracking problems is in the following form,

$$\frac{1}{2}\sum_{t=0}^{N-1}\left[(x_t-\theta_t)^\top Q(x_t-\theta_t)+u_t^\top R u_t\right]+\frac{1}{2}x_N^\top P^e x_N, \tag{3.9}$$

where $Q > 0$, $R > 0$, and P^e is the solution to the algebraic Riccati equation with respect to Q, R [127]. Basically, in this special case, $Q_t = Q$, $R_t = R$ for $0 \le t \le N-1$, $Q_N = P^e$, $\theta_N = 0$, and only θ_t changes for $t = 0, 1, \ldots, N-1$. The LQ tracking problem (3.9) aims to follow a time-varying trajectory $\{\theta_t\}$ with constant weights on the tracking cost and the control cost.

Regret upper bound. By Theorem 3.1 and Theorem 3.2, we have the following bound.

Corollary 3.1. *Under the stepsizes in Theorem 3.1, RHTM with FOSS as the initialization rule satisfies*

$$\text{Regret}(RHTM) = O\left(\zeta^2(\frac{\sqrt{\zeta}-1}{\sqrt{\zeta}})^{2K}\sum_{t=0}^{N}\|\theta_t-\theta_{t-1}\|\right)$$

where $K = \lfloor (W-1)/p \rfloor$, ζ *is the condition number of the corresponding* $C(\boldsymbol{z})$, $\theta_{-1} = 0$.

Fundamental limit. For any online algorithm, we have the following lower bound.

Theorem 3.3 (Lower Bound)**.** *Consider* $1 \le W \le N/3$, *any condition number* $\zeta > 1$, *any variation budget* $4\bar{\theta} \le L_N \le (2N+1)\bar{\theta}$, *and any controllability index* $p \ge 1$. *For any online algorithm* $\mathscr{A}$, *there exists an LQ tracking problem in form (3.9) where i) the canonical-form system* (A, B) *has controllability index* p, *ii) the sequence* $\{\theta_t\}$ *satisfies the variation budget* $\sum_{t=0}^{N}\|\theta_t-\theta_{t-1}\| \le L_N$, *and iii) the corresponding* $C(\boldsymbol{z})$ *has condition number* ζ, *such that the following lower bound holds*

$$\text{Regret}(\mathscr{A}) = \Omega\left((\frac{\sqrt{\zeta}-1}{\sqrt{\zeta}+1})^{2K}L_N\right) = \Omega\left((\frac{\sqrt{\zeta}-1}{\sqrt{\zeta}+1})^{2K}\sum_{t=0}^{N}\|\theta_t-\theta_{t-1}\|\right) \tag{3.10}$$

where $K = \lfloor (W-1)/p \rfloor$ *and* $\theta_{-1} = 0$.

Firstly, the lower bound in Theorem 3.3 almost matches the upper bound in Corollary 3.1, especially when ζ is large, demonstrating that RHTM utilizes the predictions in a near-optimal way. The major conditions in Theorem 3.3 require that the prediction window is short compared with the horizon: $W \le N/3$, and the variation of the cost functions should not be too small: $L_N \ge 4\bar{\theta}$, otherwise the online control problem is too easy and the regret can be very small. Moreover, the small gap between the regret bounds is conjectured to be nontrivial, because this gap coincides with the

long lasting gap in the convergence rate of the first-order algorithms for strongly convex and smooth optimization. In particular, the lower bound in Theorem 3.3 matches the fundamental convergence limit in [100], and the upper bound is by triple momentum's convergence rate, which is the best one to our knowledge.

3.6 Numerical experiments

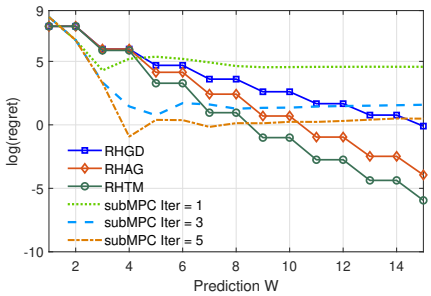


Figure 3.1: LQ tracking

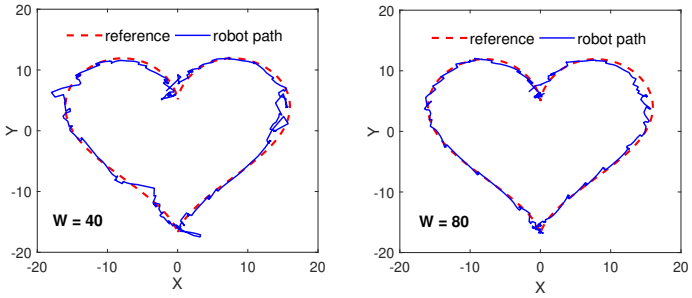


Figure 3.2: Two-wheel robot tracking

LQ tracking problem in Example 3.1. The experiment settings are as follows. Let $A = [0, 1; -1/, 5/6], B = [0; 1]$, $N = 30$. Consider diagonal Q_t, R_t with diagonal entries i.i.d. from Unif$[1, 2]$. Let θ_t i.i.d. from Unif$[-10, 10]$. The stepsizes of RHGD and RHTM are based on the conditions in Theorem 3.1. The stepsizes of RHAG can be viewed as RHTM with $\delta_c = 1/l_c, \delta_y = \delta_\omega = \frac{\sqrt{\zeta}-1}{\sqrt{\zeta}+1}$ and $\delta_z = 0$. Notice that the system considered here has $n = 2$, $m = 1$, and $p = 2$. We compare RHGC with a suboptimal MPC algorithm, fast gradient MPC (subMPC) [122]. Roughly speaking, subMPC solves the W-stage truncated optimal control from t to $t + W - 1$ by Nesterov's accelerated gradient [100], and one iteration of Nesterov's accelerated gradient requires $2W$ gradient evaluations of stage cost function since W stages are considered and each stage has two costs f_t and g_t. This implies that, in terms of the number of gradient evaluations,

subMPC with one iteration corresponds to our RHTM because RHTM also requires roughly $2W$ gradient evaluations per stage. Therefore, Figure 3.1 compares our RHGC algorithms with subMPC with one iteration. Figure 3.1 also plots subMPC with 3 and 5 iterations for more insights. Figure 3.1 shows that all our algorithms achieve exponential decaying regrets with respect to W, and the regrets are piecewise constant, matching Theorem 3.1. Further, it is observed that RHTM and RHAG perform better than RHGD, which is intuitive because triple momentum and Nesterov's accelerated gradient are accelerated versions of gradient descent. In addition, our algorithms are much better than subMPC with 1 iteration, implying that our algorithms utilize the lookahead information more efficiently given similar computational time. Finally, subMPC achieves better performance with larger W but the improvement saturates as W increases, in contrast to the steady improvement of RHGC.

Path tracking for a two-wheel mobile robot. Though we presented our online algorithms on an LTI system, our RHGC methods are applicable to some nonlinear systems as well. Here we consider a two-wheel mobile robot with nonlinear kinematic dynamics $\dot{x} = v\cos\delta, \dot{y} = v\sin\delta, \dot{\delta} = w$ where (x, y) is the robot location, v and w are the tangential and angular velocities respectively, δ denotes the tangent angle between v and the x axis [128]. The control is directly on the v and w, e.g., via the pulse-width modulation (PWM) of the motor [129]. Given a reference path (x_t^r, y_t^r), the objective is to balance the tracking performance and the control cost, i.e., min $\sum_{t=0}^{N} [c_t \cdot ((x_t - x_t^r)^2 + (y_t - y_t^r)^2) + c_t^v \cdot (v_t)^2 + c_t^w \cdot (w_t)^2]$. We discretize the dynamics with time interval $\Delta t = 0.025$s and follow similar ideas in this chapter to reformulate the problem to an unconstrained optimization with respect to (x_t, y_t) and apply RHGC.

Figure 3.2 plots the tracking results with window $W = 40$ and $W = 80$ corresponding to lookahead time 1s and 2s. It is observed that the robot follows the reference trajectory well especially when the path is smooth but deviates a little more when the path has sharp turns, and a longer lookahead window leads to better tracking performance. These results confirm that our RHGC works effectively on nonlinear systems.

3.7 Conclusion and extensions

This chapter studies the role of predictions on dynamic regrets of online control problems with linear dynamics. We design RHGC algorithms and provide regret upper bounds of two specific algorithms: RHGD and RHTM. We also provide a fundamental limit and show the fundamental limit almost matches RHTM's upper bound.

This chapter leads to many interesting directions of extensions, some of which are briefly discussed below. The first direction is to extend the prediction model considered in this chapter to more general prediction models as considered in Chapter 2. The second direction is to consider unknown systems with process noises. The third direction is to consider algorithm design without using the canonical form, since the canonical form can be sensitive to system uncertainties. One possible alternative is the finite-impulse-response formulation, which is left for future investigations. Further, more studies could be conducted on general control problems, e.g. nonlinear control and control with constraints. Besides, it is interesting to consider other performance metrics, such as competitive ratio, since the dynamic regret is non-vanishing. Finally, other future directions include closing the gap of the regret bounds and more discussion on the effect of the canonical-form transformation on the condition number.

Part II

Improving Control Performance with Safety Guarantees

Chapter 4 | Introduction

In recent years, reinforcement learning and online learning have enjoyed great successes in improving *performance under uncertainties* [130–134]. There is a growing interest in applying these learning tools to control systems in the physical world for better control performance, e.g., data centers [87, 106], robotics [135], autonomous vehicles [136], power systems [137, 138], etc. However, *safety issues under uncertainties* emerge as major challenges in the real-world implementation of learning tools, e.g., self-driving car crashes [139]. It is critical to design learning-based controllers to satisfy safety requirements, e.g., to avoid collision with obstacles with feasible actuator inputs. This motivates the major question of this part: *How to design learning-based controllers with both performance and safety guarantees under uncertainties?*

To tackle the question above, this Part II considers a linear time-invariant system, $x_{t+1} = Ax_t + Bu_t + w_t$, with bounded random disturbances $w_t \in \mathbb{W}$, state constraint $x_t \in \mathbb{X}$, and control constraint $u_t \in \mathbb{U}$. We study two types of uncertainties: (i) unknown time-varying future costs in Chapter 6 and (ii) unknown systems in Chapter 7. Other types of uncertainties and the combination of the two uncertainties above are left for future work. We design control algorithms with safety guarantees and performance guarantees for each type of uncertainty. For safety guarantees, we consider hard constraint satisfaction, i.e. $x_t \in \mathbb{X}$ and $u_t \in \mathbb{U}$ for all t despite uncertainties. For performance guarantees, we provide bounds on the policy regret, which is a

non-asymptotic performance measure that compares our algorithms' total costs with an optimal time-invariant policy's total cost in hindsight (more details are provided in Chapter 6 and 7).

In the following, we discuss the related work and provide an outline for this part.

4.1 Related Work

4.1.1 Constrained optimal control

Constrained optimal control has been studied for a long time by the control community. Perhaps the most famous algorithms are model predictive control (MPC) and its variants, e.g., robust MPC that guarantees hard constraint satisfaction under uncertainties [51–55], and stochastic MPC for soft constraint satisfaction under uncertainties [140, 141]. There is also a growing interest recently in the adaptive version of MPC that actively explores the system to reduce the model uncertainties for less conservativeness and better performance with constraint satisfaction, e.g., RAMPC [55–58], and constrained data-enabled MPC [142]. However, most MPC literature focuses on stability, feasibility, and constraint satisfaction guarantees, with fewer results on the optimality performance, especially the non-asymptotic performance analysis.

Next, we note that the structures of optimal controllers in the constrained case remain largely unknown, except for special cases such as linear quadratic regulators (LQR) *without* system disturbances (i.e., no w_t) and with linear constraints, where the optimal controllers are piecewise-affine (PWA) [143]. However, for linear systems with disturbances as considered in this part, the optimal controllers may have more

complicated structures, and the optimal control design calls for computationally demanding algorithms, such as dynamic programming. For computational efficiency and for simplicity, most literature focuses on control design under certain structures, e.g., PWA controllers [52,140], linear dynamical controllers [144], linear static controllers [145], etc. In this part, we also consider controllers with a certain structure: controllers that are linear with history system disturbances. For more discussions, we refer the reader to Chapter 5.

For nonlinear systems with constraints, in addition to MPC, there are other methods, including control barrier functions [146], invariant sets [147], etc.

4.1.2 Control design with learning tools

We review the related work in two aspects: (i) with unknown future time-varying costs, and (ii) with unknown systems.

(i) In the literature, the study on the control design with unknown future time-varying costs is usually called "online optimal control". Most papers on online optimal control focus on the unconstrained case and provide online algorithms with sublinear policy regret guarantees [42–44, 112, 117, 148]. However, online optimal control with constraints remains under-explored. Recently, [149, 150] study constrained online optimal control but assume *no* system disturbances (i.e., no w_t). In Chapter 6, we will focus on constrained online optimal control *with* system disturbances. We leverage the disturbance-action policies developed for the unconstrained online optimal control in [42].

(ii) There is a rich body of literature on learning-based control with unknown systems for both the unconstrained case [59–66, 151–155] and the constrained case [156, 157].

Nevertheless, most non-asymptotic performance guarantees are built upon the unconstrained case [59–66, 152–155]. Recently, there is a growing interest in developing non-asymptotic performance guarantees for the constrained case, but the results remain limited. For example, [158] analyzes the Bayesian regret of adaptive MPC, but [158] allows constraint violation during the process and requires restarting the system at some safe state after updating the model. Besides, [159] considers adaptive learning with input constraint satisfaction, but without state constraints. They provide a sublinear regret bound but their approach requires an oracle that may be computationally intractable. Moreover, [144] guarantees hard constraint satisfaction but does not update policies during the learning process, i.e., they first collect data for a finite number of steps, then use the collected data to compute a policy whose gap with the optimal policy is evaluated. In Chapter 7, we aim to design computationally efficient adaptive controllers with hard constraint satisfaction and regret guarantees. Our design relies on certainty equivalence, which has been shown to be optimal for learning-based control without constraints [61, 160]. Our handling of constraints is somewhat similar to [144], since we both consider linear policies and convert the constraints on the states and actions to the constraints on the policy parameters.

Lastly, we mention that another important notion of safety is system stability, which is also studied in the safe RL/learning-based control literature [63, 161, 162].

4.1.3 Safe reinforcement learning.

Safe reinforcement learning (RL) enjoys a lot of literature with various formulations of safety [162–165]. Perhaps the most relevant formulation is RL with state and action constraints [135, 166–168]. Most safe RL papers focus on numerical performance and

safety guarantees. Theoretical results on the tradeoff between the optimality performance and safety guarantees are under-explored.

4.1.4 Online convex optimization with memory and constraints

Online convex optimization (OCO) with memory considers coupled costs and decoupled constraints [80, 98, 169]. The papers on OCO with coupled constraints usually allow constraint violation [170–172]. Besides, OCO does not consider system dynamics or disturbances.

Outline and contributions

1. Chapter 5 reviews preliminary results that will be useful for the development of Chapter 6 and Chapter 7, including the disturbance-action policy (DAP) and a slow-variation trick studied in [42], and robust constrained optimization [173].

2. Chapter 6 studies online constrained optimal control with time-varying and unknown cost functions. For simplicity, we consider a time-invariant and known linear system here. We design an online algorithm that guarantees constraint satisfaction and feasibility despite system disturbances. We establish a $\tilde{O}(\sqrt{T})$ policy regret upper bound for our online algorithm by comparing it with the optimal linear static policy in hindsight.

3. Chapter 7 studies adaptive control of the constrained linear quadratic regulator with an unknown system. For simplicity, we consider a time-invariant and known cost function here. We design a safe adaptive control algorithm that learns the system and the optimal DAP controller on a single trajectory without violating constraints. We establish a $\tilde{O}(T^{2/3})$ regret bound and a general error bound on the

model estimation.

Notations for Part II.

$\mathbb{I}_E$ denotes an indicator function on set E. For a matrix A, let $\|A\|_2, \|A\|_F, \|A\|_\infty, \|A\|_1$ denote the L_2, Frobenious, L_∞, L_1 norm respectively. For a vector x, let $\|x\|_2, \|x\|_1, \|x\|_\infty$ denote the corresponding vector norms. Let $\mathbf{1}_n$ denote an all-one vector in $\mathbb{R}^n$. For two vectors $a, b \in \mathbb{R}^n$, we write $a \leq b$ if $a_i \leq b_i$ for any entry i. As a special case, for two scalars x and y, we write $(x, y) > 0$ if $x > 0$ and $y > 0$. Let $\text{vec}(A)$ denote the vectorization of matrix A. For better exposition, some bounds use $\Theta(\cdot)$ to omit constants that do not depend on T and/or the problem dimensions explicitly. $\tilde{O}(\cdot)$ hides polynomial factors. For a probability distribution $\mathscr{D}_\eta$, we write $\eta \sim \bar{\eta}\mathscr{D}_\eta$ if $\eta = \bar{\eta}\tilde{\eta}$ and $\tilde{\eta}$ follows distribution $\mathscr{D}_\eta$. Define $\mathbb{B}(\hat{\theta}, r) = \{\theta : \|\theta - \hat{\theta}\|_F \leq r\}$. We define $\sum_{s=1}^{0} a_s = 0$ for any a_s.

Chapter 5 | Preliminaries

This chapter provides preliminary results that will be useful for Chapter 6 and Chapter 7. Firstly, we review the disturbance-action policy (DAP) and its properties for the unconstrained optimal control in [42, 43]. Next, we introduce a slow-variation trick that is commonly adopted for online decision making with couplings [42, 43, 86, 174]. We will leverage this trick to handle constraints in the later chapters. Finally, we review robust constrained optimization with linear constraints.

5.1 Disturbance-action policy and its properties

Consider an optimal control problem with a linear time-invariant system $x_{t+1} = Ax_t + Bu_t + w_t$, and a cost function $l(x_t, u_t)$. Consider i.i.d. disturbances w_t with zero mean. For simplicity, we consider a known system (A, B) and $x_0 = 0$.

When (A, B) is controllable, there exists a controller $u_t = -Kx_t$ such that $(A - BK)$ is exponentially stable, i.e., there exists $\kappa \geq 1$ and $\gamma \in [0, 1)$ such that $\|(A - BK)^t\|_2 \leq \kappa(1 - \gamma)^t$ for all $t \geq 0$. We call such $(A - BK)$ as a (κ, γ)-stable matrix and such controller K as a (κ, γ)-stabilizing controller. A formal definition of the (κ, γ)-stability is provided below. The (κ, γ)-stability serves as a quantative version of exponentially stability to ease the non-asymptotic analysis.

Definition 5.1 ((κ, γ)-stability)**.** *For $\kappa \geq 1$ and $\gamma \in [0, 1)$, a matrix A is called (κ, γ)-stable if $\|A^t\|_2 \leq \kappa(1 - \gamma)^t$ for all $t \geq 0$. A controller $u = -Kx$ is called a*

(κ, γ)-stabilizing controller if $(A - BK)$ is a (κ, γ)-stable matrix and $\|K\|_2 \leq \kappa$.[1]

If the cost function $l(x_t, u_t)$ is a quadratic function, under proper conditions, the optimal controller is known to be in the form of $u_t = -K^* x_t$. A controller in this form is usually called a linear static policy. In [42, 43], instead of trying to design a linear static policy directly, they aim to design another form of controller to approximate the linear static policy, i.e., the disturbance-action policy as defined below.

Definition 5.2 (Disturbance-Action Policy [43])**.** *Fix a (κ, γ)-stabilizing controller K a priori. Given a memory length $H \in \{1, 2, \ldots, T\}$, a disturbance-action policy (DAP) defines the control policy as:*

$$u_t = -\mathsf{K} x_t + \sum_{i=1}^{H} M[i] w_{t-i}, \qquad \forall\, t \geq 0, \tag{5.1}$$

When the system (A, B) is known, the history disturbances can be computed by $w_{t-i} = x_{t-i+1} - A x_{t-i} - B u_{t-i}$. We let $w_t = 0$ for $t \leq 0$ for notational simplicity. The DAP parameters are summarized in a list of matrices denoted by $\mathbf{M} = \{M[i]\}_{i=1}^{H}$.

It is shown in [42, 43] that any (κ, γ)-stabilizing controller K can be approximated by a DAP policy $\mathbf{M}$ in a bounded set $\mathbb{M}_H$ for large enough H.

Lemma 5.1 (Relation between linear static policies and DAPs [42, 43])**.** *For any (κ, γ)-stabilizing controller K, there exists $\mathbf{M} = \{M[i]\}_{i=1}^{H}$ in the set below:*[2]

$$\mathbb{M}_H = \{\mathbf{M} : \|M[k]\|_\infty \leq 2\sqrt{n}\kappa^2(1-\gamma)^{k-1},\ \forall\, 1 \leq k \leq H\}, \tag{5.2}$$

[1] This definition is slightly different from that in [42, 43]. This (κ, γ)-stability corresponds to the $(\sqrt{\kappa}, \gamma)$-strong stability defined in [42, 43].

[2] In [42, 43], they describe the set by L_2 matrix norm, here we change it to L_∞ norm since it is more convenient for constrained optimal control.

such that

$$\max_{t\geq 0}(\|u_t - u'_t\|_2, \|x_t - x'_t\|_2) \leq O(\kappa^3(1-\gamma)^H/\gamma), \tag{5.3}$$

where (x_t, u_t) *is generated by implementing the controller* $u = -Kx$*, and* (x'_t, u'_t) *is generated by implementing the DAP with policy* $\mathbf{M}$ *determined by* K*.*

Lemma 5.1 indicates that the goal of designing an optimal linear static controller can be achieved by designing DAP with $\mathbf{M} \in \mathbb{M}_H$ for large enough H with properly chosen κ, γ. Since a (κ_0, γ_0)-stabilizing controller K^* is also (κ, γ)-stabilizing for $\kappa \geq \kappa_0$ and $\gamma \geq \gamma_0$, one can let κ, γ to be large enough when deciding $\mathbb{M}_H$ and K in DAP (5.1). However, larger κ and γ calls for even larger H to maintain a small approximation error between a linear static policy and a DAP as described in (5.3), which causes an increase in the number of parameters in DAP and thus increase computation complexity. Therefore, in practice, one has to balance between the generality of the DAP policy space and the computation time when deciding the parameters κ, γ, H.

Though DAP increases the number of policy parameters, it enjoys several major advantages. In the following, we provide a major property of DAP, based on which we discuss two major benefits of DAP.

Proposition 5.1 (State and action approximations by DAP [42])**.** *When implementing a disturbance-action policy* (5.1) *with time-varying* $\mathbf{M}_t = \{M_t[i]\}_{i=1}^H$ *at each stage* $t \geq 0$*, the states and actions satisfy:*

$$x_t = A_{\mathsf{K}}^H x_{t-H} + \tilde{x}_t(\mathbf{M}_{t-H:t-1}) \quad \textit{and} \quad u_t = -\mathsf{K}A_{\mathsf{K}}^H x_{t-H} + \tilde{u}_t(\mathbf{M}_{t-H:t}), \tag{5.4}$$

where $A_{\mathsf{K}} = A - B\mathsf{K}$*. The approximate/surrogate state and action* $\tilde{x}_t(\mathbf{M}_{t-H:t-1})$ *and*

$\tilde{u}_t(\mathbf{M}_{t-H:t})$ *are defined as:*

$$\begin{aligned}
\tilde{x}_t(\mathbf{M}_{t-H:t-1}) &= \sum_{k=1}^{2H} \Phi_k^x(\mathbf{M}_{t-H:t-1}) w_{t-k}, \\
\tilde{u}_t(\mathbf{M}_{t-H:t}) &= -\mathsf{K}\tilde{x}_t(\mathbf{M}_{t-H:t-1}) + \sum_{i=1}^{H} M_t[i] w_{t-i} = \sum_{k=1}^{2H} \Phi_k^u(\mathbf{M}_{t-H:t}) w_{t-k}, \\
\Phi_k^x(\mathbf{M}_{t-H:t-1}) &= A_{\mathsf{K}}^{k-1}\mathbb{I}_{(k\le H)} + \sum_{i=1}^{H} A_{\mathsf{K}}^{i-1} B M_{t-i}^{[k-i]}\mathbb{I}_{(1\le k-i\le H)} \\
\Phi_k^u(\mathbf{M}_{t-H:t}) &= M_t^{[k]}\mathbb{I}_{(k\le H)} - \mathsf{K}\Phi_k^x(\mathbf{M}_{t-H:t-1}),
\end{aligned}$$

where $\mathbf{M}_{t-H:t} \coloneqq \{\mathbf{M}_{t-H}, \ldots, \mathbf{M}_t\}$. *Further, when the DAP is time-invariant and equals* $\mathbf{M}$ *for all* t, *we define* $\mathring{\Phi}_k^x(\mathbf{M}) = \Phi_k^x(\mathbf{M}, \ldots, \mathbf{M})$ *and* $\mathring{\Phi}_k^u(\mathbf{M}) = \Phi_k^u(\mathbf{M}, \ldots, \mathbf{M})$.

Proposition 5.1 indicates two major advantages by introducing DAP. The first advantage is the finite-history truncation: the actual state x_t and action u_t can be approximated by the approximate state $\tilde{x}_t(\mathbf{M}_{t-H:t-1})$ and approximate action $\tilde{u}_t(\mathbf{M}_{t-H:t})$ that only depend on the H-step history, i.e., $\mathbf{M}_{t-H}, \ldots, \mathbf{M}_t$, instead of all the history. The approximation error decays exponentially with H. Therefore, one can focus on a finite-length of history when discussing the current stage.

The second advantage is the affine dependence with policy parameters: notice that the approximate state $\tilde{x}_t(\mathbf{M}_{t-H:t-1})$ and approximate action $\tilde{u}_t(\mathbf{M}_{t-H:t})$ are affine functions of the history policy parameters $\mathbf{M}_{t-H}, \ldots, \mathbf{M}_t$, instead of polynomial functions if $u_t = -Kx_t$ is implemented. The affine dependence eases the representation of the cost functions. More specificially, [42] introduces the approximate cost function based on the approximate state and action by

$$f(\mathbf{M}_{t-H:t}) = \mathbb{E}[l(\tilde{x}_t(\mathbf{M}_{t-H:t-1}), \tilde{u}_t(\mathbf{M}_{t-H:t}))]. \tag{5.5}$$

For a convex function $l(x, u)$, by the affine dependence introduced above, the approximate

cost function $f(\mathbf{M}_{t-H:t})$ is also convex with respect to the policy parameters. In this way, the optimal control problem can be approximated by a convex optimization over DAP parameters. Lastly, we note that the affine dependence will also simplify the constraint representation, which will be illustrated in details in Chapter 6.

Remark 5.1. *The disturbance-action policy is related to affine disturbance feedback in stochastic MPC [140, 141], which also considers policies that are linear with disturbances to convexify the control problem in MPC's lookahead horizon.*

Remark 5.2. *If A is open-loop stable, the prefixed controller K in DAP (5.1) can be chosen to be zero, leading to a simpler form $u_t = \sum_{i=1}^{H} M[i]w_{t-i}$.*

Remark 5.3. *The DAP is also related to finite-impulse response (FIR), which is in the form $u_t = \sum_{i=1}^{H} M[i]w_{t-i}$. FIR is a finite truncation of infinite-impulse response (IIR). IIR is adopted in [60, 144] for learning-based robust control design, and its truncated version FIR is utilized for algorithm implementation. Compared with FIR, DAP introduces an additional stabilizing controller K to stabilize the system (A, B), which guarantees an exponential decaying approximation error determined by A_{K}^{H} in Proposition 5.1. FIR does not introduce K but imposes additional constraints on $\mathbf{M}$ to ensure the stability of the closed-loop system and the exponential decaying truncation error. For more details, we refer the reader to [60, 144].*

5.2 Slow-variation trick

The slow variation trick is a popular method to handle couplings across stages in online decision making [42, 43, 86, 174]. Classic online decision making problems such as OCO [14] and multi-armed bandit [134] focus on decoupled decision making at each

stage. For online decision making with couplings, e.g., OCO with memory [42,43], online Markov decision processes [86,174], one possible algorithm design idea is to approximate the coupled problems by decoupled problems and hope that the approximation errors are small. One method to achieve this is the slow variation trick.

Roughly speaking, the slow variation trick approximates a stage-coupled function $f_t(\mathbf{M}_{t-H}, \dots, \mathbf{M}_t)$ as a decoupled function

$$\mathring{f}_t(\mathbf{M}_t) \coloneqq f_t(\mathbf{M}_t, \dots, \mathbf{M}_t) \tag{5.6}$$

by assuming all H-step decisions $\mathbf{M}_{t-H}, \dots, \mathbf{M}_t$ are equal to $\mathbf{M}_t$. If the variation of the decisions $\|\mathbf{M}_{t-1} - \mathbf{M}_t\|$ is small for all t, the approximation error $|f_t(\mathbf{M}_{t-H}, \dots, \mathbf{M}_t) - \mathring{f}_t(\mathbf{M}_t)|$ is also small provided with some smoothness property of f_t.

More specifically, we illustrate the slow variation trick by an example of "OCO with memory" arising from unconstrained online optimal control in [42] and explain how the small approximation error is achieved. Consider a stage cost function $f_t(\mathbf{M}_{t-H}, \dots, \mathbf{M}_t)$ that depends on the decisions in the H-stage history, i.e., $\mathbf{M}_{t-H}, \dots, \mathbf{M}_t$. The OCO with memory problem studies how to select $\mathbf{M}_t$ at each stage t without knowing the current function f_t, which is only revealed after selecting $\mathbf{M}_t$. To design algorithms for this problem, one can apply the slow-variation trick to approximate the coupled cost function as a decoupled one by (5.6) Then, the problem is approximated by a classic OCO problem: select $\mathbf{M}_t$ at each t without knowing $\mathring{f}_t(\mathbf{M}_t)$. Consequently, one can apply classic OCO algorithms on the approximate OCO problem with stage cost function $\mathring{f}_t(\mathbf{M}_t)$, such as online gradient descent (OGD) $\mathbf{M}_t = \mathbf{M}_{t-1} - \eta_t \nabla \mathring{f}_{t-1}(\mathbf{M}_{t-1})$.

OGD minimizes the total approximate cost $\sum_{t=1}^{T} \mathring{f}_t(\mathbf{M}_t)$ up to a sublinear regret $o(T)$. By selecting small enough stepsizes η_t in OGD, the decision variation

$\|\mathbf{M}_t - \mathbf{M}_{t-1}\|$ is small for all t if the gradient of $\mathring{f}_{t-1}$ is bounded. Therefore, the difference between the total approximate cost and the total actual cost, i.e., $\sum_{t=1}^{T} \mathring{f}_t(\mathbf{M}_t) - \sum_{t=1}^{T} f_t(\mathbf{M}_{t-H}, \ldots, \mathbf{M}_t)$ can also be shown to be small. Hence, OGD with small enough stepsizes implemented on the approximate decoupled cost functions can effectively reduce the total coupled costs.

In Chapter 6 and Chapter 7, we will apply the slow variation trick to handle the constraints. For more details, we refer to the corresponding chapters.

5.3 Robust optimization with constraints

In this section, we review a standard method to handle robust constraint satisfaction in robust optimization. Consider a robust optimization problem with linear constraints [173]:

$$\min_x f(x) \quad \text{s.t.} \quad a_i^\top x \leq b_i, \ \forall\, a_i \in \mathscr{C}_i, \ \forall\, 1 \leq i \leq k, \tag{5.7}$$

where the (box) uncertainty sets are defined as $\mathscr{C}_i = \{a_i = \tilde{a}_i + P_i z : \|z\|_\infty \leq \bar{z}\}$ for any i. Notice that the robust constraint $\{a_i^\top x \leq b_i, \ \forall\, a_i \in \mathscr{C}_i\}$ is equivalent to the standard constraint $\{\sup_{a_i \in \mathscr{C}_i} [a_i^\top x] \leq b_i\}$. Further, one can derive

$$\begin{aligned} \sup_{a_i \in \mathscr{C}_i} a_i^\top x &= \sup_{\|z\|_\infty \leq \bar{z}} (\tilde{a}_i + P_i z)^\top x \\ &= \tilde{a}_i^\top x + \sup_{\|z\|_\infty \leq \bar{z}} z^\top (P_i^\top x) = \tilde{a}_i^\top x + \|P_i^\top x\|_1 \bar{z} \end{aligned} \tag{5.8}$$

Therefore, the robust optimization (5.7) can be equivalently reformulated as the linearly constrained optimization below:

$$\min_x f(x) \quad \text{s.t.} \quad \tilde{a}_i^\top x + \|P_i^\top x\|_1 \bar{z} \leq b_i, \ \forall\, 1 \leq i \leq k.$$

Chapter 6 | Online Optimal Control with State and Action Constraints

This chapter studies online optimal control with state constraints and action constraints. We focus on *time-varying stage cost functions* $l_t(x_t, u_t)$ and the stage-t cost function $l_t(x_t, u_t)$ is *unknown* until the action u_t is selected. For simplicity, we consider a *known linear time-invariant system* with i.i.d. and bounded system disturbances. Most online optimal control literature with regret analysis only considers unconstrained control [42–44, 112, 117, 148]. Recent papers on the constrained case only consider linear systems *without* system disturbances [149, 150].

Contributions and chapter outline. We design an online algorithm OGD-BZ with constraint satisfaction *despite system disturbances*. The problem formulation is rigorously stated in Section 6.1 and the algorithm design is presented in Section 6.2. Our design builds upon the disturbance-action policy for the unconstrained problem and leverages robust constrained optimization and the slow variation trick to ensure constraint satisfaction. In Section 6.3, we establish a $\tilde{O}(\sqrt{T})$ policy regret bound for convex cost functions compared with the optimal linear static policy in hindsight. We also provide constraint satisfaction and feasibility guarantees. Section 6.4 provides numerical results to demonstrate the performance of OGD-BZ.

6.1 Problem formulation

In this chapter, we consider an online optimal control problem with linear dynamics and affine constraints. Specifically, at each stage $t \in \{0, 1, \ldots, T\}$, an online algorithm observes the current state x_t and implements an action u_t, which incurs a cost $l_t(x_t, u_t)$. The stage cost function $l_t(\cdot, \cdot)$ is generated adversarially and revealed to the agent after the action u_t is taken. The system evolves to the next state according to $x_{t+1} = Ax_t + Bu_t + w_t$, where x_0 is fixed, w_t is a random disturbance bounded by $w_t \in \mathbb{W} = \{w \in \mathbb{R}^n : \|w\|_\infty \leq w_{\max}\}$. We consider affine constraints on the states and actions:

$$x_t \in \mathbb{X} = \{x \in \mathbb{R}^n : D_x x \leq d_x\}, \qquad u_t \in \mathbb{U} = \{u \in \mathbb{R}^m : D_u u \leq d_u\}, \tag{6.1}$$

where $d_x \in \mathbb{R}^{k_x}$ and $d_u \in \mathbb{R}^{k_u}$. Define $k_c = k_x + k_u$ as the total number of the constraints. Define $\kappa_B = \max(\|B\|_2, 1)$.

For simplicity, this chapter considers that the parameters $A, B, w_{\max}, D_x, d_x, D_u, d_u$ are known a priori and zero initial point $x_0 = 0$. Chapter 7 studies unknown (A, B) with a time-invariant cost $l(x_t, u_t)$ and discusses nonzero x_0. We leave the studies on more general cases as future work, e.g., time-varying costs with unknown systems, unknown constraints, etc.

A desirable online algorithm should satisfy two properties: (i) safety, (ii) small total costs. We formalize each property below.

Safety definitions. We formally define safety considered in this chapter. We also introduce strict safety and loose safety.

Definition 6.1 (Safety)**.** *Consider an online algorithm (a controller) $\mathscr{A}$ that selects*

action $u_t^{\mathscr{A}} \in \mathbb{U}$ *based on history states* $\{x_k^{\mathscr{A}}\}_{k=0}^t$ *and cost functions* $\{l_k(\cdot,\cdot)\}_{k=0}^{t-1}$. *The algorithm* $\mathscr{A}$ *is called safe if* $x_t^{\mathscr{A}} \in \mathbb{X}$ *and* $u_t^{\mathscr{A}} \in \mathbb{U}$ *for all* $t \geq 0$ *and all disturbances* $\{w_k \in \mathbb{W}\}_{k\geq 0}$.

Definition 6.2 (Strict and loose safety)**.** *A safe online algorithm* $\mathscr{A}$ *is called* ϵ*-strictly safe for some* $\epsilon > 0$ *if* $D_x x_t^{\mathscr{A}} \leq d_x - \epsilon \mathbf{1}_{k_x}$ *and* $D_u u_t^{\mathscr{A}} \leq d_u - \epsilon \mathbf{1}_{k_u}$ *for all* $t \geq 0$ *under any disturbance sequence* $\{w_k \in \mathbb{W}\}_{k\geq 0}$. *Similarly, a safe online algorithm* $\mathscr{A}$ *is called* ϵ*-loosely safe for some* $\epsilon > 0$ *if* $D_x x_t^{\mathscr{A}} \leq d_x + \epsilon \mathbf{1}_{k_x}$ *and* $D_u u_t^{\mathscr{A}} \leq d_u + \epsilon \mathbf{1}_{k_u}$ *for all* $t \geq 0$ *under any disturbance sequence* $\{w_k \in \mathbb{W}\}_{k\geq 0}$.

Benchmark policy class and policy regret. We aim to reduce the total expected cost of an online algorithm $\mathscr{A}$, which is defined as

$$J_T(\mathscr{A}) = \mathop{\mathbb{E}}_{\{w_k\}_{k\geq 0}} \left[\sum_{t=0}^{T} l_t(x_t^{\mathscr{A}}, u_t^{\mathscr{A}}) \right]. \tag{6.2}$$

Similar to the unconstrained online optimal control literature [42–44], we measure the online performance by comparing the online total cost $J_T(\mathscr{A})$ with the total cost of the optimal benchmark policy in hindsight, where the benchmark policy is selected from a benchmark policy class. In this chapter, we consider linear policies of the form $u_t = -Kx_t$ as our benchmark policy for simplicity, though the optimal policy for the constrained control of noisy systems may be nonlinear as reviewed in Section 5.1. We leave the discussion on more general benchmark policies as future work.

In particular, our benchmark policy class includes linear controllers $u_t = -Kx_t$ under the conditions below:

$$\mathscr{K} = \{K : u_t = -Kx_t \text{ is safe and } (\kappa, \gamma)\text{-stabilizing}\}, \tag{6.3}$$

where $\kappa \geq 1$, $\gamma \in [0, 1)$ are the parameters for the policy class to be selected by the decision maker. The definition of (κ, γ)-stability is in Definition 5.1, followed by a discussion on the choices of κ, γ. The policy regret of online algorithm $\mathscr{A}$ is defined as:

$$\text{Regret}_T^p(\mathscr{A}) = J_T(\mathscr{A}) - \min_{K \in \mathscr{K}} J_T(K). \tag{6.4}$$

Assumptions. For the well-definedness of our policy regret definition (6.4), there should exist K such that $u_t = -Kx_t$ is safe and (κ, γ)-stabilizing. Here, we impose a slightly stronger assumption on the existence of a strictly safe policy. Strict safety is crucial for our algorithm design and many algorithms in the literature, e.g., tube-based RMPC [51–53], because it provides flexibility for problem approximations and reformulations (see Section 6.2 for more details).

Assumption 6.1. *There exists $K_F \in \mathscr{K}$ such that the policy $u_t = -K_F x_t$ is ϵ_F-strictly safe for some $\epsilon_F > 0$.*

Assumption 6.1 implies that (A, B) is controllable, $\mathbb{X}$ and $\mathbb{U}$ have non-empty interiors, and the disturbance set $\mathbb{W}$ is small enough so that a disturbed linear system $x_{t+1} = (A - BK_F)x_t + w_t$ stays in the interiors of $\mathbb{X}$ and $\mathbb{U}$ for any $\{w_k \in \mathbb{W}\}_{k=0}^T$. In addition, Assumption 6.1 implicitly assumes that 0 belongs to the interiors of $\mathbb{X}$ and $\mathbb{U}$ since we let $x_0 = 0$. A sufficient condition to verify Assumption 6.1 is by LMI reformulation in [175].[1] A verifiable sufficient and necessary condition for Assumption 6.1 remains as a future direction.

[1] [175] provides an LMI program to compute a near-optimal linear controller for a time-invariant constrained control problem, which can be used to verify the existence of a safe solution. To verify Assumption 3, one could run the LMI program with the constraints tightened by ϵ and continue to reduce ϵ if no solution is found until ϵ is smaller than a certain threshold.

In addition, we introduce the following assumptions on the disturbances and the cost functions, which are standard in the literature [42, 43].

Assumption 6.2. *w_t is i.i.d., zero mean. $w_t \in \mathbb{W} = \{w : \|w\|_\infty \leq w_{\max}\}$ with $w_{\max} > 0$.*[2]

Assumption 6.3. *Consider bounded $\mathbb{X}$ and $\mathbb{U}$, i.e., there exist $x_{\max}, u_{\max}$ such that $\|x\|_2 \leq x_{\max}, \forall x \in \mathbb{X}, \|u\|_2 \leq u_{\max}, \forall u \in \mathbb{U}$*

Assumption 6.4. *For any $t \geq 0$, cost function $l_t(x_t, u_t)$ is convex and differentiable with respect to x_t and u_t. Further, there exists $G > 0$, such that for any $\|x\|_2 \leq b$, $\|u\|_2 \leq b$, we have $\|\nabla_x l_t(x, u)\|_2 \leq Gb$ and $\|\nabla_u l_t(x, u)\|_2 \leq Gb$.*

Assumption 6.4 includes the convex quadratic cost $x_t^\top Q x_t + u_t^\top R u_t$ as a special case.

6.2 Online algorithm design

This section introduces our online algorithm design. Roughly speaking, to develop our online algorithm, we first convert the constrained online optimal control into *OCO with memory and coupled constraints*, which is later converted into classical OCO and solved by OCO algorithms. The conversions leverage the approximation and the reformulation techniques in the Chapter 5. During the conversions, we ensure that the outputs of the OCO algorithms are safe for the original control problem. This is achieved by tightening the original constraints (adding buffer zones) to allow for approximation errors. Besides, our method ensures small approximation errors and thus small buffer zones so that the optimality/regret is not sacrificed significantly for safety. The details of the algorithm design are discussed below.

[2]The results of this chapter can be extended to adversarial noises.

Step 1: Constraints on Approximate States and Actions. When applying the disturbance-action policies (5.1), we can use (5.4) to rewrite the state constraint $x_{t+1} \in \mathbb{X}$ as

$$D_x A_{\mathsf{K}}^H x_{t-H+1} + D_x \tilde{x}_{t+1} \leq d_x, \quad \forall \{w_k \in \mathbb{W}\}_{k=0}^T, \tag{6.5}$$

where $\tilde{x}_{t+1}$ is the approximate state. Note that the term $D_x A_{\mathsf{K}}^H x_{t-H+1}$ decays exponentially with H. If there exists H such that $D_x A_{\mathsf{K}}^H x_{t-H+1} \leq \epsilon_1 \mathbf{1}_{k_x}$, $\forall \{w_k \in \mathbb{W}\}_{k=0}^T$, then a tightened constraint on the approximate state, i.e.

$$D_x \tilde{x}_{t+1} \leq d_x - \epsilon_1 \mathbf{1}_{k_x}, \quad \forall \{w_k \in \mathbb{W}\}_{k=0}^T, \tag{6.6}$$

can guarantee the original constraint on the true state (6.5).

The action constraint $u_t \in \mathbb{U}$ can similarly be converted into a tightened constraint on the approximate action $\tilde{u}_t$, i.e.,

$$D_u \tilde{u}_t \leq d_u - \epsilon_1 \mathbf{1}_{k_u}, \quad \forall \{w_k \in \mathbb{W}\}_{k=0}^T, \tag{6.7}$$

if $D_u(-\mathsf{K} A_{\mathsf{K}}^H x_{t-H}) \leq \epsilon_1 \mathbf{1}_{k_u}$ for any $\{w_k \in \mathbb{W}\}_{k=0}^T$.

Step 2: Constraints on the Policy Parameters Next, we reformulate the robust constraints (6.6) and (6.7) on $\tilde{x}_{t+1}$ and $\tilde{u}_t$ as polytopic constraints on policy parameters $\mathbf{M}_{t-H:t}$ based on the robust optimization techniques reviewed in Section 5.3.

Firstly, we consider the ith row of the constraint (6.6), i.e. $D_{x,i}^\top \tilde{x}_{t+1} \leq d_{x,i} - \epsilon_1$ $\forall \{w_k \in \mathbb{W}\}_{k=0}^T$, where $D_{x,i}^\top$ denotes the ith row of the matrix D_x. This constraint is equivalent to $\sup_{\{w_k \in \mathbb{W}\}_{k=0}^T} (D_{x,i}^\top \tilde{x}_{t+1}) \leq d_{x,i} - \epsilon_1$. Further, by (5.8) and the definitions of $\tilde{x}_{t+1}$ and $\mathbb{W}$, we obtain

$$\sup_{\{w_k \in \mathbb{W}\}} D_{x,i}^\top \tilde{x}_{t+1} = \sup_{\{w_k \in \mathbb{W}\}} D_{x,i}^\top \sum_{s=1}^{2H} \Phi_s^x(\mathbf{M}_{t-H+1:t}) w_{t+1-s}$$

$$
\begin{aligned}
&= \sum_{s=1}^{2H} \sup_{w_{t+1-s} \in \mathbb{W}} D_{x,i}^{\top} \Phi_s^x(\mathbf{M}_{t-H+1:t}) w_{t+1-s} \\
&= \sum_{s=1}^{2H} \|D_{x,i}^{\top} \Phi_s^x(\mathbf{M}_{t-H+1:t})\|_1 w_{\max}
\end{aligned}
$$

Define $g_i^x(\mathbf{M}_{t-H+1:t}) = \sum_{s=1}^{2H} \|D_{x,i}^{\top} \Phi_s^x(\mathbf{M}_{t-H+1:t})\|_1 w_{\max}$. Hence, the robust constraint (6.6) on $\tilde{x}_{t+1}$ is equivalent to the following polytopic constraints on $\mathbf{M}_{t-H+1:t}$:

$$
g_i^x(\mathbf{M}_{t-H+1:t}) \leq d_{x,i} - \epsilon_1, \quad \forall\, 1 \leq i \leq k_x. \tag{6.8}
$$

Similarly, the constraint (6.7) on $\tilde{u}_t$ is equivalent to:

$$
g_j^u(\mathbf{M}_{t-H:t}) \leq d_{u,j} - \epsilon_1, \quad \forall\, 1 \leq j \leq k_u, \tag{6.9}
$$

where $g_j^u(\mathbf{M}_{t-H:t}) = \sum_{s=1}^{2H} \|D_{u,j}^{\top} \Phi_s^u(\mathbf{M}_{t-H:t})\|_1 w_{\max}$.

Step 3: OCO with Memory and Temporal-coupled Constraints By Step 2 and our review of robust optimization, we can convert the constrained online optimal control problem into *OCO with memory and coupled constraints*. That is, at each t, the decision maker selects a policy $\mathbf{M}_t$ satisfying constraints (6.8) and (6.9), and then incurs a cost $f_t(\mathbf{M}_{t-H:t})$ as defined in (5.5) in Chapter 5. In our framework, both the constraints (6.8), (6.9) and the cost function $f_t(\mathbf{M}_{t-H:t})$ couple the current policy with the historical policies. This makes the problem far more challenging than OCO with memory which only considers coupled costs [80].

Step 4: Benefits of the Slow Variation of Online Policies We approximate the coupled constraint functions $g_i^x(\mathbf{M}_{t-H+1:t})$ and $g_j^u(\mathbf{M}_{t-H:t})$ as decoupled ones below,

$$
\mathring{g}_i^x(\mathbf{M}_t) = g_i^x(\mathbf{M}_t, \ldots, \mathbf{M}_t), \qquad \mathring{g}_i^u(\mathbf{M}_t) = g_i^u(\mathbf{M}_t, \ldots, \mathbf{M}_t), \tag{6.10}
$$

by letting the historical policies $\mathbf{M}_{t-H:t-1}$ be identical to the current $\mathbf{M}_t$.[3] If the online policy $\mathbf{M}_t$ varies slowly with t, which is satisfied by most OCO algorithms (e.g., OGD with a diminishing stepsize [176]), one may be able to bound the approximation errors by $g_i^x(\mathbf{M}_{t-H+1:t}) - \mathring{g}_i^x(\mathbf{M}_t) \leq \epsilon_2$ and $g_j^u(\mathbf{M}_{t-H:t}) - \mathring{g}_j^u(\mathbf{M}_t) \leq \epsilon_2$ for a small $\epsilon_2 > 0$. Consequently, the constraints (6.8) and (6.9) are ensured by the polytopic constraints that only depend on $\mathbf{M}_t$:

$$\mathring{g}_i^x(\mathbf{M}_t) \leq d_{x,i} - \epsilon_1 - \epsilon_2, \ \mathring{g}_j^u(\mathbf{M}_t) \leq d_{u,j} - \epsilon_1 - \epsilon_2, \tag{6.11}$$

where the buffer zone ϵ_2 allows for the approximation error caused by neglecting the variation of the online policies.

Step 5: Conversion to OCO By Step 4, we define a decoupled search space/constraint set on each policy below,

$$\begin{aligned} \Omega_\epsilon = \{\mathbf{M} \in \mathbb{M}_H : \mathring{g}_i^x(\mathbf{M}) \leq d_{x,i} - \epsilon, \forall 1 \leq i \leq k_x, \\ \mathring{g}_j^u(\mathbf{M}) \leq d_{u,j} - \epsilon, \forall\, 1 \leq j \leq k_u\}, \end{aligned} \tag{6.12}$$

where $\mathbb{M}_H$ is defined in 5.2. Notice that Ω_ϵ provides buffer zones with size ϵ to account for the approximation errors ϵ_1 and ϵ_2. Now, we can further convert the "OCO with memory and coupled constraints" in Step 3 into a classical OCO problem below. We approximate the cost function $f(\mathbf{M}_{t-H:t})$ by the decoupled function $\mathring{f}_t(\mathbf{M}_t) := f_t(\mathbf{M}_t, \ldots, \mathbf{M}_t)$ introduced in (5.6) in Chapter 5. The OCO problem reformulated from the constrained online control is the following. At each t, the agent selects a policy $\mathbf{M}_t \in \Omega_\epsilon$, and then suffers a convex stage cost $\mathring{f}_t(\mathbf{M}_t)$. We apply online gradient descent to solve this OCO problem, as described in Algorithm 6.1. We select the stepsizes of OGD to be small

[3] Though we consider $\mathbf{M}_t = \cdots = \mathbf{M}_{t-H}$ here, the component $M_t[i]$ of $\mathbf{M}_t = \{M_t[i]\}_{i=1}^H$ can be different for different i.

enough to ensure small approximation errors from Step 4 and thus small buffer zones, but also to be large enough to allow online policies to adapt to time-varying environments. Conditions for suitable stepsizes are discussed in Section 6.3.

Algorithm 6.1: OGD-BZ

Input: A (κ, γ)-stabilizing matrix $\mathbb{K}$, parameter $H > 0$, buffer size ϵ, stepsize η_t.

1 Determine the polytopic constraint set Ω_ϵ by (6.12) with buffer size ϵ and initialize $\mathbf{M}_0 \in \Omega_\epsilon$.

2 **for** $t = 0, 1, 2, \ldots, T$ **do**

3 Implement action $u_t = -\mathsf{K}x_t + \sum_{i=1}^{H} M_t[i] w_{t-i}$.

4 Observe the next state x_{t+1} and record $w_t = x_{t+1} - Ax_t - Bu_t$.

5 Run projected OGD $\mathbf{M}_{t+1} = \Pi_{\Omega_\epsilon}\left[\mathbf{M}_t - \eta_t \nabla \mathring{f}_t(\mathbf{M}_t)\right]$ with $\mathring{f}_t(\mathbf{M}_t) \coloneqq f_t(\mathbf{M}_t, \ldots, \mathbf{M}_t)$.

In Algorithm 6.1, the most computationally demanding step at each stage is the projection onto the polytope Ω_ϵ, which requires solving a quadratic program. Nevertheless, one can reduce the online computational burden via offline computation by leveraging the solution structure of quadratic programs (see [91] for more details).

Lastly, we note that other OCO algorithms can be applied to solve this problem, too, e.g., online natural gradient, online mirror descent, etc. One can also apply projection-free methods, e.g., [170], to reduce the computational burden at the expense of $o(T)$ constraint violation.

Remark 6.1. *To ensure safety, safe RL literature usually constructs a safe set for the state [135], while this chapter constructs a safe search space Ω_ϵ for the policies directly. Besides, safe RL literature may employ unsafe policies occasionally. For example, [135]*

allows unsafe exploration policies within the safe set and changes to a safe policy on the boundary of the safe set. However, our search space Ω_ϵ only contains safe policies. Despite a smaller policy search space, our OGD-BZ still achieves desirable (theoretical) performance. Nevertheless, when the system is unknown, larger sets of exploration policies may benefit the performance, which is left as future work.

Remark 6.2. *It is worth comparing our method with a well-known robust MPC method: tube-based robust MPC (see, e.g., [54]). Tube-based robust MPC also tightens the constraints to allow for model inaccuracy and/or disturbances. However, tube-based robust MPC considers constraints on the states, while our method converts the state (and action) constraints into the constraints on the policy parameters by leveraging the properties of disturbance-action policies.*

6.3 Theoretical results

In this section, we show that OGD-BZ guarantees both safety and $\tilde{O}(\sqrt{T})$ policy regret under proper parameters.

Preparation. To establish the conditions on the parameters for our theoretical results, we introduce three quantities $\epsilon_1(H), \epsilon_2(\eta, H), \epsilon_3(H)$ below. We note that $\epsilon_1(H)$ and $\epsilon_2(\eta, H)$ bound the approximation errors in Step 1 and Step 4 of the previous section respectively (see Lemma 6.1 and Lemma 6.3 in the proof of Theorem 6.1 for more details). $\epsilon_3(H)$ bounds the constraint violation of the disturbance-action policy $\mathbf{M}(K)$, where $\mathbf{M}(K)$ approximates the linear controller $u_t = -Kx_t$ for any $K \in \mathscr{K}$ (see Lemma 6.4 in the proof of Theorem 6.1 for more details).

Definition 6.3. *We define*

$$\epsilon_1(H) = c_1 n\sqrt{m}H(1-\gamma)^H, \epsilon_2(\eta, H) = c_2\eta \cdot n^2\sqrt{m}H^2$$
$$\epsilon_3(H) = c_3\sqrt{n}(1-\gamma)^H$$

where c_1, c_2, and c_3 are polynomials of $\|D_x\|_\infty, \|D_u\|_\infty, \kappa, \kappa_B, \gamma^{-1}, w_{\max}, G$.

6.3.1 Safety of OGD-BZ

Theorem 6.1 (Feasibility & Safety)**.** *Consider constant stepsize $\eta_t = \eta$, $\epsilon \geq 0, H \geq \frac{\log(2\kappa)}{\log((1-\gamma)^{-1})}$. If the buffer size ϵ and H satisfy*

$$\epsilon \leq \epsilon_F - \epsilon_1(H) - \epsilon_3(H),$$

the set Ω_ϵ is non-empty. Further, if η, ϵ and H also satisfy

$$\epsilon \geq \epsilon_1(H) + \epsilon_2(\eta, H),$$

our OGD-BZ is safe, i.e., $x_t^{\text{OGD-BZ}} \in \mathbb{X}$ and $u_t^{\text{OGD-BZ}} \in \mathbb{U}$ for all t and for any disturbances $\{w_k \in \mathbb{W}\}_{k=0}^T$.

Discussions: Firstly, Theorem 6.1 shows that ϵ should be small enough to ensure a nonempty Ω_ϵ and thus valid/feasible outputs of OGD-BZ. This is intuitive since the constraints are more conservative as ϵ increases. Since $\epsilon_1(H) + \epsilon_3(H) = \Theta(H(1-\gamma)^H)$ decays with H by Definition 6.3, the first condition also implies a large enough H.

Secondly, Theorem 6.1 shows that to ensure safety, the buffer size ϵ should also be large enough to allow for the total approximation errors $\epsilon_1(H) + \epsilon_2(\eta, H)$, which is consistent with our discussion in the previous section. To ensure the compatibility of the

two conditions on ϵ, the approximation errors $\epsilon_1(H) + \epsilon_2(\eta, H)$ should be small enough, which requires a large enough H and a small enough η by Definition 6.3.

In conclusion, the safety requires a large enough H, a small enough η, and an ϵ which is neither too large nor too small. For example, we can select $\eta \leq \frac{\epsilon_F}{8c_2 n^2 \sqrt{m} H^2}$, $\epsilon_F/4 \leq \epsilon \leq 3\epsilon_F/4$, and $H \geq \max(\frac{\log(\frac{8(c_1+c_3)n\sqrt{m}}{\epsilon_F}T)}{\log((1-\gamma)^{-1}}, \frac{\log(2\kappa)}{\log((1-\gamma)^{-1})})$.

Remark 6.3. *It can be shown that it is safe to implement any* $\mathbf{M} \in \Omega_\epsilon$ *for all* $t \geq 0$ *under the conditions of Theorem 6.1 based on the proof of Theorem 6.1.*

6.3.2 Policy regret bound for OGD-BZ

Theorem 6.2 (Regret Bound)**.** *Under the conditions in Theorem 6.1, OGD-BZ enjoys the regret bound below:*

$$\begin{aligned}\text{Regret}_T^p(\text{OGD-BZ}) \leq O\Bigg(& n^3 m H^3 \eta T + \frac{mn}{\eta} + (1-\gamma)^H H^{2.5} T(n^3 m^{1.5} + \sqrt{k_c} m n^{2.5})/\epsilon_F \\ & + \epsilon T H^{1.5}(n^2 m + \sqrt{k_c m n^3})/\epsilon_F \Bigg),\end{aligned}$$

where the hidden constant depends polynomially on $\kappa, \kappa_B, \gamma^{-1}, \|D_x\|_\infty, \|D_u\|_\infty, \|d_x\|_2, \|d_u\|_2, w_{\max}, G$.

Theorem 6.2 provides a regret bound for OGD-BZ as long as OGD-BZ is safe. Notice that as the buffer size ϵ increases, the regret bound becomes worse. This is intuitive since our OGD-BZ will have to search for policies in a smaller set Ω_ϵ if ϵ increases. Consequently, the buffer size ϵ can serve as a tuning parameter for the trade-off between safety and regrets, i.e., a small ϵ is preferred for low regrets while a large ϵ is preferred for safety (as long as $\Omega_\epsilon \neq \emptyset$). In addition, although a small stepsize η is preferred for safety in Theorem 6.1, Theorem 6.2 suggests that the stepsize should not be too small for low regrets since the regret bound contains a $\Theta(\eta^{-1})$ term. This is intuitive since the

stepsize η should be large enough to allow OGD-BZ to adapt to the varying objectives for better online performance.

Next, we provide a regret bound with specific parameters.

Corollary 6.1. *For sufficiently large T, when $H \geq \frac{\log(8(c_1+c_2)n\sqrt{m}T/\epsilon_F)}{\log((1-\gamma)^{-1})}$, $\eta = \Theta(\frac{1}{n^2\sqrt{m}H\sqrt{T}})$, $\epsilon = \epsilon_1(H) + \epsilon_2(\eta, H) = \Theta(\frac{\log(n\sqrt{m}T)}{\sqrt{T}})$, OGD-BZ is safe and* $\text{Regret}_T^p(\text{OGD-BZ}) \leq \tilde{O}\left((n^3 m^{1.5} {k_c}^{0.5})\sqrt{T}\right)$.

Corollary 6.1 shows that OGD-BZ achieves $\tilde{O}(\sqrt{T})$ regrets when $H \geq \Theta(\log T)$, $\eta^{-1} = \tilde{\Theta}(\sqrt{T})$, and $\epsilon = \tilde{\Theta}(1/\sqrt{T})$. This demonstrates that OGD-BZ can ensure both constraint satisfaction and sublinear regrets under the proper parameters of the algorithm. We remark that a larger H is preferred for better performance due to smaller approximation errors and a potentially larger policy search space Ω_ϵ, but the computational complexity of OGD-BZ increases with H. Besides, though the choices of H, η, and ϵ above require the prior knowledge of T, one can apply doubling tricks [176] to avoid this requirement. Lastly, we note that our $\tilde{O}(\sqrt{T})$ regret bound is consistent with the unconstrained online optimal control literature for convex cost functions [42]. For strongly convex costs, the regret for the unconstrained case is logarithmic in T [43]. We leave the study on the constrained control with strongly convex costs for the future.

6.3.3 Proof of Theorem 6.1

To prove Theorem 6.1, we first provide lemmas to bound errors by $\epsilon_1(H), \epsilon_2(\eta, H)$, and $\epsilon_3(H)$, respectively. Firstly, we show that the approximation error in Step 1 of the previous section can be bounded by $\epsilon_1(H)$.

Lemma 6.1 (Error bound $\epsilon_1(H)$)**.** *When $\mathbf{M}_k \in \mathbb{M}$ for all k and $H \geq \frac{\log(2\kappa)}{\log((1-\gamma)^{-1})}$, we*

have

$$\max_{\|w_k\|_\infty \le w_{\max}} \|D_x A_{\mathsf{K}}^H x_{t-H}\|_\infty \le \epsilon_1(H),$$

$$\max_{\|w_k\|_\infty \le w_{\max}} \|D_u \mathsf{K} A_{\mathsf{K}}^H x_{t-H}\|_\infty \le \epsilon_1(H).$$

The proof of Lemma 6.1 is straightforward with the boundedness of x_t below, whose proof is similar to the proof of the bounded states in [42].

Lemma 6.2 (Bound on x_t)**.** *With $\mathbf{M}_k \in \mathscr{M}$ for all k and $\kappa(1-\gamma)^H < 1$, when $H \ge \frac{\log(2\kappa)}{\log((1-\gamma)^{-1})}$, we have $\max(\|x_t\|_2, \|u_t\|_2) \le b = 8\sqrt{mn^2}Hw_{\max}\kappa^2\kappa_B/\gamma = O(\sqrt{mn^2}H)$.*

Secondly, we show that the error incurred by the Step 3 of the previous section can be bounded by $\epsilon_2(\eta, H)$. The proof is provided in Appendix B.1.1.

Lemma 6.3 (Error bound $\epsilon_2(\eta, H)$)**.** *When $H \ge \frac{\log(2\kappa)}{\log((1-\gamma)^{-1})}$, the policies $\{\mathbf{M}_t\}_{t=0}^T$ generated by OGD-BZ with a constant stepsize η satisfy*

$$\max_{1\le i\le k_x} |\mathring{g}_i^x(\mathbf{M}_t) - g_i^x(\mathbf{M}_{t-H+1:t})| \le \epsilon_2(\eta, H),$$

$$\max_{1\le j\le k_u} |\mathring{g}_j^u(\mathbf{M}_t) - g_j^u(\mathbf{M}_{t-H:t})| \le \epsilon_2(\eta, H).$$

Thirdly, we show that for any $K \in \mathscr{K}$, there exists a disturbance-action policy $\mathbf{M}(K) \in \mathbb{M}$ to approximate the policy $u_t = -Kx_t$. However, $\mathbf{M}(K)$ may not be safe and is only $\epsilon_3(H)$-loosely safe. The proof is based on Lemma 5.1 in Chapter 5.

Lemma 6.4 (Error bound $\epsilon_3(H)$)**.** *For any $K \in \mathscr{K}$, there exists a disturbance-action policy $\mathbf{M}(K) = \{M[i](K)\}_{i=1}^H \in \mathbb{M}_H$ defined as $M[i](K) = (\mathsf{K} - K)(A - BK)^{i-1}$ such that*

$$\max(\|D_x[x_t^K - x_t^{\mathbf{M}(K)}]\|_\infty, \|D_u[u_t^K - u_t^{\mathbf{M}(K)}]\|_\infty) \le \epsilon_3(H)$$

where (x_t^K, u_t^K) *and* $(x_t^{\mathbf{M}(K)}, u_t^{\mathbf{M}(K)})$ *are produced by controller* $u_t = -Kx_t$ *and disturbance-action policy* $\mathbf{M}(K)$ *respectively. Hence,* $\mathbf{M}(K)$ *is* $\epsilon_3(H)$*-loosely safe.*

Based on Lemma 6.4, we can further show that $\mathbf{M}(K)$ belongs to a polytopic constraint set in the following corollary. For the rest of the paper, we will omit the arguments in $\epsilon_1(H), \epsilon_2(\eta, H), \epsilon_3(H)$ for notational simplicity.

Corollary 6.2. *If* K *is* ϵ_0*-strictly safe for* $\epsilon_0 \geq 0$*, then* $\mathbf{M}(K) \in \Omega_{\epsilon_0 - \epsilon_1 - \epsilon_3}$.

Proof of Theorem 6.1. For notational simplicity, we denote the states and actions generated by OGD-BZ as x_t and u_t in this proof. First, we show $\mathbf{M}(K_F) \in \Omega_\epsilon$ below. Since K_F defined in Assumption 6.1 is ϵ_F-strictly safe, by Corollary 6.2, there exists $\mathbf{M}(K_F) \in \Omega_{\epsilon_F - \epsilon_1 - \epsilon_3}$. Since the set Ω_ϵ is smaller as ϵ increases, when $\epsilon_F - \epsilon_1 - \epsilon_3 \geq \epsilon$, we have $\mathbf{M}(K_F) \in \Omega_{\epsilon_F - \epsilon_1 - \epsilon_3} \subseteq \Omega_\epsilon$, so Ω_ϵ is non-empty.

Next, we prove the safety by Lemma 6.1 and Lemma 6.3 based on the discussions in the previous section. Specifically, OGD-BZ guarantees that $\mathbf{M}_t \in \Omega_\epsilon$ for all t. Thus, by Lemma 6.3, we have $g_i^x(\mathbf{M}_{t-H:t-1}) = g_i^x(\mathbf{M}_{t-H:t-1}) - \mathring{g}_i^x(\mathbf{M}_{t-1}) + \mathring{g}_i^x(\mathbf{M}_{t-1}) \leq \epsilon_2 + d_{x,i} - \epsilon$ for any i. Further, by Step 2 of the previous section and Lemma 6.1, we have $D_{x,i}^\top x_t = D_{x,i}^\top A_{\mathsf{K}}^H x_{t-H} + D_{x,i}^\top \tilde{x}_t \leq \|D_x A_{\mathsf{K}}^H x_{t-H}\|_\infty + g_i^x(\mathbf{M}_{t-H:t-1}) \leq \epsilon_1 + \epsilon_2 + d_{x,i} - \epsilon \leq d_{x,i}$ for any $\{w_k \in \mathbb{W}\}_{k=0}^T$ if $\epsilon \geq \epsilon_1 + \epsilon_2$. Therefore, $x_t \in \mathbb{X}$ for all $w_k \in \mathbb{W}$. Similarly, we can show $u_t \in \mathbb{U}$ for any $w_k \in \mathbb{W}$. Thus, OGD-BZ is safe. □

6.3.4 Proof of Theorem 6.2

We divide the regret into three parts and bound each part.

$$\text{Regret}_T^p(\text{OGD-BZ}) = J_T(\mathscr{A}) - \min_{K \in \mathscr{K}} J_T(K)$$

$$= \underbrace{J_T(\mathscr{A}) - \sum_{t=0}^{T} \mathring{f}_t(\mathbf{M}_t)}_{\text{Part i}} + \underbrace{\sum_{t=0}^{T} \mathring{f}_t(\mathbf{M}_t) - \min_{\mathbf{M}\in\Omega_\epsilon} \sum_{t=0}^{T} \mathring{f}_t(\mathbf{M})}_{\text{Part ii}} + \underbrace{\min_{\mathbf{M}\in\Omega_\epsilon} \sum_{t=0}^{T} \mathring{f}_t(\mathbf{M}) - \min_{K\in\mathscr{K}} J_T(K)}_{\text{Part iii}}$$

Bound on Part ii. Firstly, we bound Part ii based on OGD's regret bound in [176], which relies on a gradient bound to be proved in Appendix B.1.2.

Lemma 6.5. *With a constant stepsize η, we have* Part ii $\le \delta^2/2\eta + \eta G_f^2 T/2$, *where* $\delta = \sup_{\mathbf{M},\tilde{\mathbf{M}}\in\Omega_\epsilon} \|\mathbf{M} - \tilde{\mathbf{M}}\|_F \le 4\sqrt{mn}\kappa^2/\gamma$ *and* $G_f = \max_t \sup_{\mathbf{M}\in\Omega_\epsilon} \|\nabla \mathring{f}_t(\mathbf{M})\|_F \le O(bw_{\max}\sqrt{n}\sqrt{H}\frac{1+\gamma}{\gamma})$. *Consequently, when* $H \ge \frac{\log(2\kappa)}{\log((1-\gamma)^{-1})}$, *we have* $G_f \le \Theta(\sqrt{n^3H^3m}w_{\max}^2)$ *and the hidden factor is quadratic on* $w_{\max}$.

Bound on Part iii. For notational simplicity, we denote $\mathbf{M}^* = \arg\min_{\Omega_\epsilon} \sum_{t=0}^{T} \mathring{f}_t(\mathbf{M})$, $K^* = \arg\min_{\mathscr{K}} J_T(K)$. By Lemma 6.4, we can construct a loosely safe $\mathbf{M}_{\text{ap}} = \mathbf{M}(K^*)$ to approximate K^*. By Corollary 6.2, we have

$$\mathbf{M}_{\text{ap}} \in \Omega_{-\epsilon_1-\epsilon_3}. \tag{6.13}$$

We will bound Part iii by leveraging $\mathbf{M}_{\text{ap}}$ as middle-ground and bounding the Part iii-A and Part iii-B defined below.

$$\text{Part iii} = \underbrace{\sum_{t=0}^{T} (\mathring{f}_t(\mathbf{M}^*) - \mathring{f}_t(\mathbf{M}_{\text{ap}}))}_{\text{Part iii-A}} + \underbrace{\sum_{t=0}^{T} \mathring{f}_t(\mathbf{M}_{\text{ap}}) - J_T(K^*)}_{\text{Part iii-B}}$$

Lemma 6.6. *For $K^* \in \mathscr{K}$ and $\mathbf{M}_{ap} = \mathbf{M}(K^*)$, we have* Part iii-B $\le \Theta(Tn^2mH^2(1-\gamma)^H)$.

Lemma 6.7. *Under the conditions in Theorem 6.2, we have*

$$\text{Part iii-A} \le \Theta\left((\epsilon_1+\epsilon_3+\epsilon)TH^{\frac{3}{2}}\frac{n^2m+\sqrt{k_cmn^3}}{\epsilon_F}\right).$$

We highlight that $\mathbf{M}_{\text{ap}}$ may not belong to Ω_ϵ by (6.13). Therefore, even though $\mathbf{M}^*$ is optimal in Ω_ϵ, Part iii-A can still be positive and has to be bounded to yield a regret bound. This is different from the unconstrained online control literature [43], where Part iii-A is non-positive because $\mathbf{M}_{\text{ap}} \in \mathscr{M}$ and $\mathbf{M}^*$ is optimal in the same set $\mathscr{M}$ when there are no constraints (see [43] for more details).

Bound on Part i. Finally, we provide a bound on Part i.

Lemma 6.8. *With a constant stepsize η, we have* Part i$\leq O(Tn^2mH^2(1-\gamma)^H + n^3mH^3\eta T)$.

The proofs of Lemma 6.6 and Lemma 6.8 are similar to those in [43]. The proof of Lemma 6.7 is provided in the next subsection.

Finally, Theorem 6.2 can be proved by summing up the bounds on Part i, Part ii, Part iii-A, and Part iii-B in Lemmas 6.5-6.8 and only explicitly showing the highest order terms.

6.3.5 Proof of Lemma 6.7

We define $\mathbf{M}^\dagger = \arg\min_{\Omega_{-\epsilon_1-\epsilon_3}} \sum_{t=0}^T \mathring{f}_t(\mathbf{M})$. By (6.13), we have $\sum_{t=0}^T \mathring{f}_t(\mathbf{M}_{\text{ap}}) \geq \sum_{t=0}^T \mathring{f}_t(\mathbf{M}^\dagger)$. Therefore, it suffices to bound $\sum_{t=0}^T \mathring{f}_t(\mathbf{M}^*) - \sum_{t=0}^T \mathring{f}_t(\mathbf{M}^\dagger)$, which can be viewed as the difference in the optimal values when perturbing the feasible/safe set from Ω_ϵ to $\Omega_{-\epsilon_1-\epsilon_3}$. To bound Part iii-A, we establish a perturbation result by leveraging the polytopic structure of Ω_ϵ and $\Omega_{-\epsilon_1-\epsilon_3}$.

Proposition 6.1. *Consider two polytopes $\Omega_1 = \{x : Cx \leq h\}$, $\Omega_2 = \{x : Cx \leq h - \Delta\}$, where $\Delta_i \geq 0$ for all i. Consider a convex function $f(x)$ that is L-Lipschitz continuous on Ω_1. If Ω_1 is bounded, i.e., $\sup_{x_1, x_1' \in \Omega_1} \|x_1 - x_1'\|_2 \leq \delta_1$ and if Ω_2 is non-empty, i.e.,*

there exists $\mathring{x} \in \Omega_2$*, then*

$$|\min_{\Omega_1} f(x) - \min_{\Omega_2} f(x)| \leq \frac{L\delta_1 \|\Delta\|_\infty}{\min_{\{i:\Delta_i>0\}}(h - C\mathring{x})_i}. \tag{6.14}$$

Proof. Since $\Omega_2 \subseteq \Omega_1$, we have $\min_{\Omega_2} f(x) - \min_{\Omega_1} f(x) \geq 0$. Let $x_1^* = \arg\min_{\Omega_1} f(x)$. We will show that there exists $x_2^\dagger \in \Omega_2$ such that $\|x_1^* - x_2^\dagger\|_2 \leq \frac{\delta_1\|\Delta\|_\infty}{\min_{i\in S}(h-C\mathring{x})_i}$, where $S = \{i : \Delta_i > 0\}$. Then, by the Lipschitz continuity, we can prove the bound: $\min_{\Omega_2} f(x) - \min_{\Omega_1} f(x) \leq f(x_2^\dagger) - f(x_1^*) \leq \frac{L\delta_1\|\Delta\|_\infty}{\min_{i\in S}(h-C\mathring{x})_i}$.

In the following, we will show, more generally, that there exists $x_2 \in \Omega_2$ that is close to x_1 for any $x_1 \in \Omega_1$. For ease of notation, we define $y = x - \mathring{x}$, $\Omega_1^y = \{y : Cy \leq h - C\mathring{x}\}$, and $\Omega_2^y = \{y : Cy \leq h - C\mathring{x} - \Delta\}$. Notice that $0 \in \Omega_2^y$ and $(h - C\mathring{x} - \Delta)_i \geq 0$. Besides, we have $y_1 = x_1 - \mathring{x} \in \Omega_1^y$. Further, by the convexity of Ω_1^y, we have $\lambda y_1 \in \Omega_1^y$ for $0 \leq \lambda \leq 1$.

If $(Cy_1)_i \leq (h - C\mathring{x} - \Delta)_i$ for all i, then $y_1 \in \Omega_2^y$ and $x_1 \in \Omega_2$. So we can let $x_2 = x_1$ and $\|x_2 - x_1\|_2 = 0$.

If, instead, there exists a set S' such that for any $i \in S'$, $(Cy_1)_i > (h - C\mathring{x} - \Delta)_i$. Then, define

$$\lambda = \min_{i\in S'} \frac{(h - C\mathring{x} - \Delta)_i}{(Cy_1)_i}.$$

Notice that $\lambda \in [0, 1)$. We can show that $\lambda y_1 \in \Omega_2^y$ below. When $i \in S'$, $(\lambda Cy_1)_i \leq (Cy_1)_i \frac{(h-C\mathring{x}-\Delta)_i}{(Cy_1)_i} = (h - C\mathring{x} - \Delta)_i$. When $i \notin S'$, we have $(\lambda Cy_1)_i \leq \lambda(h - C\mathring{x} - \Delta)_i \leq (h - C\mathring{x} - \Delta)_i$. Therefore, $\lambda y_1 \in \Omega_2^y$. Define $x_2 = \lambda y_1 + \mathring{x}$, then $x_2 \in \Omega_2$. Notice that $\|x_1 - x_2\|_2 = \|y_1 - y_2\|_2 = (1 - \lambda)\|y_1\|_2 \leq (1 - \lambda)\delta_1$. Since $y_1 \in \Omega_1^y$, when $i \in S'$, we have $0 \leq (h - C\mathring{x} - \Delta)_i < (Cy_1)_i \leq (h - C\mathring{x})_i$. Therefore, $\frac{(h-C\mathring{x}-\Delta)_i}{(Cy_1)_i} \geq \frac{(h-C\mathring{x}-\Delta)_i}{(h-C\mathring{x})_i} = 1 - \frac{\Delta_i}{(h-C\mathring{x})_i}$. Consequently, by $S' \subseteq S$, we have $1 - \lambda \leq \max_{i\in S'} \frac{\Delta_i}{(h-C\mathring{x})_i} \leq \frac{\|\Delta\|_\infty}{\min_{i\in S'}(h-C\mathring{x})_i} \leq \frac{\|\Delta\|_\infty}{\min_{i\in S}(h-C\mathring{x})_i}$. □

To prove Lemma 6.7, we bound the quantities in (6.14) for our problem in Lemma 6.9 and then plug them in (6.14). The proof is provided in Appendix B.1.3.

Lemma 6.9. *There exists an enlarged polytope* $\Gamma_\epsilon = \{\vec{W} : C\vec{W} \leq h_\epsilon\}$ *that is equivalent to* Ω_ϵ *for any* $\epsilon \in \mathbb{R}$*, where* $\vec{W}$ *contains elements of* $\mathbf{M}$ *and auxiliary variables.*

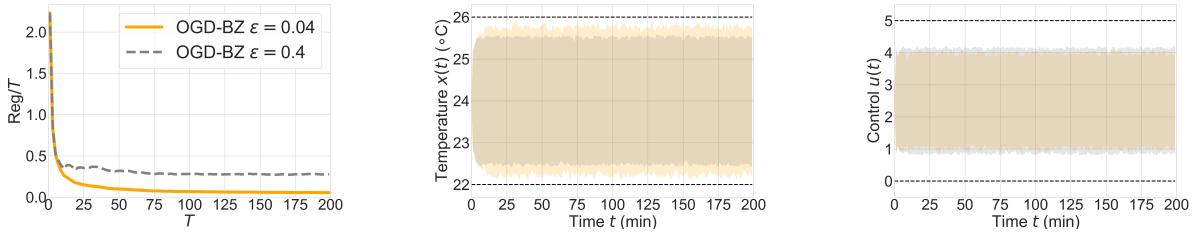
Further, under the conditions of Theorem 6.1, (i) $\Gamma_{-\epsilon_1-\epsilon_3}$ *is bounded by* $\delta_1 = \Theta(\sqrt{mn} + \sqrt{k_c})$*; (ii)* $\sum_{t=0}^{T} \mathring{f}_t(\mathbf{M})$ *is Lipschitz continuous with* $L = \Theta(T(nH)^{1.5}\sqrt{m})$*; (iii) the difference* Δ *between* Γ_ϵ *and* $\Gamma_{-\epsilon_1-\epsilon_3}$ *satisfies* $\|\Delta\|_\infty = \epsilon + \epsilon_1 + \epsilon_3$*; (iv) there exists* $\vec{W}^\circ \in \Gamma_\epsilon$ *s.t.* $\min_{\{i:\Delta_i>0\}}(h_{(-\epsilon_1-\epsilon_3)} - C\vec{W}^\circ)_i \geq \epsilon_F$.

6.4 Numerical experiments

In this section, we numerically test our OGD-BZ on a thermal control problem with a Heating Ventilation and Air Conditioning (HVAC) system. Specifically, we consider the linear thermal dynamics studied in [1] with additional random disturbances, that is, $\dot{x}(t) = \frac{1}{\upsilon\zeta}(\theta^o(t) - x(t)) - \frac{1}{\upsilon}u(t) + \frac{1}{\upsilon}\pi + \frac{1}{\upsilon}w(t)$, where $x(t)$ denotes the room temperature at time t, $u(t)$ denotes the control input that is related with the air flow rate of the HVAC system, $\theta^o(t)$ denotes the outdoor temperature, $w(t)$ represents random disturbances, π represents external heat sources' impact, υ and ζ are physical constants. We discretize the thermal dynamics with $\Delta_t = 60$s. For human comfort and/or safe operation of device, we impose constraints on the room temperature, $x(t) \in [x_{\min}, x_{\max}]$, and the control inputs, $u(t) \in [u_{\min}, u_{\max}]$. Consider a desirable temperature θ^{set} set by the user and a control setpoint u^{set}. Consider the cost function $c(t) = q_t(x(t) - \theta^{set})^2 + r_t(u(t) - u^{set})^2$.

In our experiments, we consider $\upsilon = 100$, $\zeta = 6$, $\theta^o = 30°$C, $\pi = 1.5$, and let w_t be i.i.d. generated from Unif$(-2, 2)$. Besides, we consider $\theta^{set} = 24°$C, $x_{\min} = 22°$C,

$x_{\max} = 26°\text{C}$, $u_{\min} = 0$, $u_{\max} = 5$. We consider $q_t = 2$ for all t and time-varying r_t generated i.i.d. from Unif$(0.1, 4)$. When applying OGD-BZ, we select $H = 7$ and a diminishing stepsize $\eta_t = \Theta(t^{-0.5})$, i.e., we let $\eta_t = 0.5(40)^{-0.5}$ for $t < 40$ and $\eta_t = 0.5(t+1)^{-0.5}$ for $t \geq 40$.



(a) Averaged regret (b) $x(t)$'s range (c) $x(t)$'s range

Figure 6.1: Comparison of OGD-BZ with buffer sizes $\epsilon = 0.04$ and $\epsilon = 0.4$. In Figure (b) and (c), the yellow shade represents the range of $x(t)$ generated by OGD-BZ with $\epsilon = 0.04$, while the grey shade is generated by OGD-BZ with $\epsilon = 0.4$.

Figure 6.1 plots the comparison of OGD-BZ with different buffer sizes. Specifically, $\epsilon = 0.04$ is a properly chosen buffer size, and $\epsilon = 0.4$ offers larger buffer zones. From Figure 6.1-(a), we can observe that the averaged regret with a properly chosen buffer size $\epsilon = 0.04$ quickly diminishes to 0, which is consistent with Theorem 6.2. In addition, Figure 6.1-(b) and Figure 6.1-(c) plot the range of $x(t)$ and $u(t)$ under random disturbances in 1000 trials to demonstrate the safety of OGD-BZ. With a larger buffer zone, i.e., $\epsilon = 0.4$, the range of x_t is smaller and further from the boundaries, thus being safer. Interestingly, the range of $u(t)$ becomes slightly larger, which still satisfies the control constraints because the control constraints are not binding/active in this experiment and which indicates more control power is used here to ensure a smaller range of $x(t)$ under disturbances. Finally, the regret with $\epsilon = 0.4$ is worse than that with

$\epsilon = 0.04$, which demonstrates the trade-off between safety and performance and how the choices of the buffer size affect this trade-off.

6.5 Conclusion and future directions

this chapter studies online optimal control with linear constraints and linear dynamics with random disturbances. We consider unknown future time-varying costs and known system dynamics. We propose OGD-BZ and show that OGD-BZ can satisfy all the constraints despite disturbances and ensure $\tilde{O}(\sqrt{T})$ policy regret. In Chapter 7, we will consider unknown system dynamics with time-invariant costs, but unknown systems with unknown time-varying costs are remained to be explored in the future. Further, other interesting future directions include the investigations on (i) unknown constraints, (ii) more general benchmark policy classes than linear policies, (iii) nonlinear system dynamics, (iv) robust stability, (v) adversarial disturbances, (vi) bandit feedback, (vii) regrets for strongly convex costs, (viii) reducing the dependence on system dimensions, etc.

Chapter 7 | Safe Adaptive Learning for Constrained LQR

This chapter investigates the adaptive control of constrained linear quadratic regulators with unknown systems and hopes to achieve two goals by the control design: (i) to adaptively learn the system and update the policies by improved estimation without violating constraints, (ii) to obtain non-asymptotic performance guarantees, such as model estimation rates and cost difference from the optimal policy (regret). However, most literature only achieves one goal. For example, [55, 56, 58] leverage set-membership identification and RMPC for goal (i) but lacks non-asymptotic performance guarantees. In contrast, methods based on least-square identification or Bayesian estimation achieve goal (ii) but lack constraint satisfaction during adaptive updates, e.g., [158] utilizes Bayesian estimation but restarts the system after model estimation updates and [144] utilizes least-square estimation but does not update the policy until the learning completes.

Contributions and outline. This chapter presents an adaptive control algorithm that achieves both goals above. Specifically, we adopt least-square system identification, design robustly safe controllers under uncertain models, and manage to satisfy constraints during model updates. Our design is built upon Chapter 6. In Section 7.1, we formulate the problem. In Section 7.2, we present and discuss our algorithm. In Section 7.3, we

provide safety guarantees and a $\tilde{O}(T^{2/3})$ regret bound for our algorithm. We also provide model estimation rates by general (nonlinear) policies.

7.1 Problem formulation

This chapter studies the constrained LQR with system dynamics $x_{t+1} = A_* x_t + B_* u_t + w_t$, bounded disturbances $w_t \in \mathbb{W}$ defined in Assumption 6.2, quadratic cost $l(x_t, u_t) = x_t^\top Q x_t + u_t^\top R u_t$, and affine constraints $x_t \in \mathbb{X}$, $u_t \in \mathbb{U}$ defined in (6.1). We consider unknown system parameters $\xi_* := (A_*, B_*)$ and known time-invariant cost parameters Q and R. This is different from the Chapter 6 that considers known system parameters but unknown time-varying costs. For simplicity, we consider known constraints $\mathbb{X}, \mathbb{U}$, positive definite cost matrices Q, R, and zero initial point $x_0 = 0$.[1]

We aim to design a *safe* adaptive control algorithm that gradually learns to *optimize the total cost*. We adopt the safety definition in Definition 6.1 in Chapter 6.

Besides, we use "regret" to characterize the non-asymptotic optimality performance of our adaptive control algorithm. The regret definition in this chapter is provided below.

$$\text{Regret}_T^a = \sum_{t=0}^{T-1} l(x_t, u_t) - TJ^*, \tag{7.1}$$

where

$$J^* = \min_{K \in \mathscr{K}} J(K), \qquad J(K) = \lim_{T\to\infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=0}^{T-1} l(x_t, u_t)\right], \tag{7.2}$$

$J(K)$ represents the infinite-horizon-averaged/steady-state cost of the controller

[1] For non-zero x_0, if it is sufficiently small such that it admits a safe linear controller, then our algorithm can directly be applied. If x_0 is larger but is feasible for RMPC in [52], one can apply RMPC to steer the state to a neighborhood of the origin in a finite steps and then apply our algorithm.

$u_t = -Kx_t$, and $\mathscr{K}$ is our benchmark policy class introduced in (6.3) in Chapter 6. We let J^* denote the optimal steady-state cost with a known model.

Remark 7.1. *Compared with the regret definition (6.4) in Chapter 6, the regret definition (7.1) uses steady-state cost J^* instead of finite-horizon cost as the benchmark, which does not cause too much difference because in the time-invariant case because the latter converges to the former. Besides, the definition (7.1) considers realized online cost instead of expected online cost, so we will provide with-high-probability regret bounds.*

Assumptions and definitions. To ensure the well-definedness of the regret, we impose Assumption 6.1 in Chapter 6. We also consider Assumption 6.2 and 6.3 in Chapter 6. Next, we introduce additional assumptions.

Firstly, though the exact model ξ_* is unavailable, we assume that some prior information on ξ_* is available, i.e., a bounded model uncertainty set Ξ_{ini} that contains the true model. This is a common assumption in papers that aim to guarantee constraint satisfaction from the beginning of the learning process [58, 144]. The set Θ_{ini} may be obtained by history data, physics, domain knowledge, etc.

Assumption 7.1. *There is a known model uncertainty set $\Xi_{\text{ini}} = \{\xi : \|\xi - \hat{\xi}_{\text{ini}}\|_F \leq r_{\text{ini}}\}$[2] such that $\xi_* \in \Xi_{\text{ini}}$ for $r_{\text{ini}} \in (0, +\infty)$.*

With a model uncertainty set, we can define robust safety by constraint satisfaction for any system in the set, which will be useful in our algorithm design.

[2]Here, Ξ_{ini} is symmetric on all directions, which may not be the case in practice. This is assumed for technical simplicity and can be relaxed to assuming that $\Xi^{(0)}$ is compact and contains ξ_*.

Definition 7.1 (Robust Safety)**.** *We call an algorithm to be* robustly safe *on a model uncertainty set* Ξ *if* $x_t \in \mathbb{X}, u_t \in \mathbb{U}$ *for all* t *and all* $w_k \in \mathbb{W}$ *when implementing the algorithm on any system* $\xi \in \Xi$*.*

Further, we assume that the systems in Ξ_{ini} are open-loop stable.

Assumption 7.2. *For any* $\xi = (A, B) \in \Xi$*, matrix* A *is* (κ, γ)*-stable.*

Assumption 7.2 is a strong assumption, which is introduced mainly for the simplicity of the algorithm design and analysis. With Assumption 7.2, we can let $\mathsf{K} = 0$ for DAP defined in (5.1) in Chapter 5. Without Assumption 7.2, we consider nonzero K and need a relaxed assumption on the existence of K, i.e., there exists K such that $A - B\mathsf{K}$ is (κ, γ)-stable for any $(A, B) \in \Xi_{\text{ini}}$. The relaxed assumption is commonly adopted in the robust adaptive MPC literature [55, 58]. When Assumption 6.1 holds, the relaxed assumption above essentially requires Ξ_{ini} to be small enough. If Ξ_{ini} is too large and if a safe controller is available beforehand, one can implement the safe controller to collect more data to reduce the model uncertainty.

Finally, we impose assumptions on the distribution of w_t in addition to Assumption 6.2. We first introduce the anti-concentration property in [177], which essentially assumes a positive lower bound on the probability on each direction of the random vector.

Definition 7.2 (Anti-concentration)**.** *A random vector* $X \in \mathbb{R}^n$ *is said to satisfy* (s, p)*-anti-concentration for some* $s > 0, p \in (0, 1)$ *if* $\mathbb{P}(\lambda^\top X \geq s) \geq p$ *for any* $\|\lambda\|_2 = 1$*,*

Assumption 7.3. w_t *is* σ_{sub}^2*-sub-Gaussian and satisfies* (s_w, p_w)*-anti-concentration.*[3]

[3]Notice that by $\mathbb{W} = \{w : \|w\|_\infty \leq w_{\max}\}$, we have $\sigma_{sub} \leq \sqrt{n} w_{\max}$. .

Many distributions satisfy the assumptions on w_t, including truncated Gaussian, uniform distribution, etc.

7.1.1 Preliminaries: constrained control with known model

This subsection briefly reviews the helpful results in Chapter 5 and Chapter 6 under the settings of this chapter.

We first explain the notation changes from Chapter 5 and Chapter 6. Since we face different estimated models in this chapter, we explicitly include the model parameter ξ to the relevant notations to specify which system the corresponding notation is based on. For example, we use $\tilde{x}_t(\mathbf{M}_{t-H:t-1};\xi)$ to denote the approximate action defined in Proposition 5.1 generated by policies $\mathbf{M}_{t-H},\dots,\mathbf{M}_{t-1}$ on system parameter ξ. The same applies to all the notations that depend on ξ, e.g., $\Phi_k^x(\mathbf{M}_{t-H:t-1};\xi)$, $\mathring{\Phi}_k^x(\mathbf{M}_{t-H:t-1};\xi)$, cost function $f(\mathbf{M}_{t-H:t};\xi)$, decoupled cost function $\mathring{f}(\mathbf{M}_t;\xi)$, and constraint functions $g_i^x(\mathbf{M}_{t-H:t-1};\xi), \mathring{g}_i^x(\mathbf{M}_t;\xi)$ in (6.8) and (6.10).

Due to the simplifying assumption in Assumption 7.2, we let $\mathsf{K}=0$ in the DAP controller defined in (5.1), i.e.,$u_t=\sum_{k=1}^{H} M[k]w_{t-k}$. Notice that such a control input u_t only depends on a finite history, so it does not require the approximation in Proposition 5.1, i.e.,$u_t=\tilde{u}_t(\mathbf{M}_t)$. Further, u_t does not depend on the system parameter ξ, so all the corresponding notations will not include ξ, such as $\tilde{u}_t(\mathbf{M})$, $g_i^u(\mathbf{M}_t)$, $\mathring{g}_i^u(\mathbf{M})$. Since u_t only depends on $\mathbf{M}_t$, the slow-variation trick is irrelevant and we have $g_i^u(\mathbf{M}_t)=\mathring{g}_i^u(\mathbf{M})$.

By considering a time-invariant cost function $l(x_t,u_t)$ in Section 6.2 of Chapter 6, we can compute a near-optimal time-invariant safe DAP controller $\mathbf{M}_H^*$ with memory

length H by the following optimization.

$$\min_{\mathbf{M}\in\mathbb{M}_H} \mathring{f}(\mathbf{M};\xi_*), \quad \text{subject to } \mathring{g}_i^x(\mathbf{M};\xi_*) \le d_{x,i} - \epsilon_H(H), \forall\, i,\ \mathring{g}_j^u(\mathbf{M}) \le d_{u,j},\ \forall\, j, \tag{7.3}$$

where $\epsilon_H(H)$ refers to the approximation error ϵ_1 in Chapter 6 caused by Proposition 5.1, while u_t no longer suffers this error thanks to Assumption 7.2 as discussed above, and ϵ_2 in Chapter 6 is zero here since we consider a time-invariant policy. The policy $\mathbf{M}_H^*$ is near-optimal in the sense of $J(\mathbf{M}_H^*) \approx J^*$, which can be proved by revising our regret analysis on Part iii in Section 6.3.4 in Chapter 6.

It is known that even if every $\mathbf{M}_t$ is safe to implement in a time-invariant fashion and even when the model is known, implementing the time-varying policy sequence $\{\mathbf{M}_t\}_{t\ge0}$ may still violate the state constraints due to policy switchings [83, 144]. Chapter 6 tackles this challenge by the slow-variation trick reviewed in Chapter 5 and establishes the following lemma when the system model is known.

Lemma 7.1 (Constraint satisfaction of slowly varying DAPs)**.** *If the sequence* $\{\mathbf{M}_t\}_{t\ge0}$ *varies slowly, i.e.,*$\|\mathbf{M}_t - \mathbf{M}_{t-1}\|_F \le \Delta_M$*, where* Δ_M *is called the policy variation budget, and if each* $\mathbf{M}_t$ *satisfies*

$$g_i^x(\mathbf{M}_t;\xi_*) \le d_{x,i} - \epsilon_H(H) - \epsilon_v(\Delta_M, H), \quad g_j^u(\mathbf{M}_t) \le d_{u,j},$$

where $\epsilon_v(\Delta_M, H) = O(\Delta_M\sqrt{H})$ *accounts for the error caused by ignoring the policy variation,*[4] *then it is safe to implement the time-varying policies* $\{\mathbf{M}_t\}_{t\ge0}$ *when the model is known.*

Lemma 7.1 indicates that a sequence of time-varying DAPs is safe to implement with a known model if the variation between the neighboring policies is small enough,

[4] $\epsilon_v(\Delta_M)$ corresponds to ϵ_2 in Chapter 6.

and if every individual DAP is $\epsilon_H(H) + \epsilon_v(\Delta_M, H)$-strictly safe to implement in a time-invariant fashion.

7.2 Our safe adaptive control algorithm

This section presents our safe control algorithm in Algorithm 7.1. The major challenges are to ensure constraint satisfaction in two cases: (i) with model uncertainty, (ii) when the estimated model and/or control policies are updated. To tackle the challenge (i), we design a cautious-certainty-equivalence (CCE) controller that is robustly safe on a model uncertainty set Ξ, which will be explained in Section 7.2.1. To tackle the challenge (ii), we design Algorithm 7.2 to safely transit to the new model estimation and the new CCE controller. Our adaptive controllers adopt the disturbance-action policy (DAP) structure reviewed in Chapter 5 with additive exploration noises for excitation. Our safe transition design utilizes the slow-variation trick reviewed in Chapter 5.

In the following, we briefly explain an outline of Algorithm 7.1 and then elaborate on the two major components : CCE control in Section 7.2.1 and the safe-transition algorithm in Section 7.2.2.

Algorithm outline. Algorithm 7.1 is implemented on a single trajectory without restarts. We divide the trajectory into multiple episodes. Each episode e consists of two phases, and the model estimation is updated after Phase 1. More details are discussed below.

- Phase 1: Safe exploration & exploitation (Line 3-6). In this phase, we have access to a model uncertainty set $\Xi^{(e)} := \mathbb{B}(\hat{\xi}^{(e)}, r^{(e)}) \cap \Xi_{\text{ini}}$, where $\hat{\xi}^{(e)}$ is the estimated model and $r^{(e)}$ is a confidence radius. We compute a near-optimal DAP controller $\mathbf{M}_\dagger^{(e)}$ based on

Algorithm 7.1: Safely Adaptive Cautious-Certainty-Equivalence Control

Input: Ξ_{ini}, $T^{(1)} \geq 1$, $T^{(e+1)} = 2T^{(e)}$ for $e \geq 1$. $H^{(e)}$. $\bar{\eta}^{(e)}$, $\Delta_M^{(e)}$, $T_D^{(e)}$ for $e \geq 0$.

Initialize: $\hat{\xi}^{(0)} = \hat{\xi}_{\text{ini}}, r_\xi^{(0)} = r_{\text{ini}}, \Xi^{(0)} = \Xi_{\text{ini}}$, $t_1^{(0)} = 0$. Let $w_t = \hat{w}_t = 0$ for $t < 0$.

for Episode $e = 0, 1, 2, \ldots$ **do**

Phase 1: safe exploration & exploitation Compute a polytopic robustly safe policy set $\Omega_\dagger^{(e)} = \Omega(\Xi^{(e)}, H^{(e)}, \bar{\eta}^{(e)}, \Delta_M^{(e)})$ by (7.5), and compute a cautious -certainty-equivalence (CCE) control $\mathbf{M}_\dagger^{(e)} = \arg\min_{\mathbf{M} \in \Omega_\dagger^{(e)}} \mathring{f}(\mathbf{M}; \hat{\xi}^{(e)})$.

Run Algorithm 7.2 to safely transit from $\mathbf{M}_*^{(e-1)}$ to $\mathbf{M}_\dagger^{(e)}$ when $e \geq 1$, with output $t_1^{(e)}$.

for $t = t_1^{(e)}, \ldots, t_1^{(e)} + T_D^{(e)} - 1$ **do**

Implement DAP (7.4) with $\mathbf{M}_\dagger^{(e)}$ and $\eta_t \overset{\text{i.i.d.}}{\sim} \bar{\eta}\mathscr{D}_\eta$. Estimate $\hat{w}_t$ by $\hat{\xi}^{(e)}$.

Model estimation updates: Estimate $\hat{\xi}^{(e+1)}$ by least square with projection onto Ξ_{ini}: $\tilde{\xi}^{(e+1)} = \arg\min_\xi \sum_{k=t_1^{(e)}}^{t_1^{(e)}+T_D^{(e)}-1} \|x_{k+1} - Ax_k - Bu_k\|_2^2$ and $\hat{\xi}^{(e+1)} = \Pi_{\Xi_{\text{ini}}}(\tilde{\xi}^{(e+1)})$. Update the model uncertainty set: $\Xi^{(e+1)} = B(\hat{\xi}^{(e+1)}, r^{(e+1)}) \cap \Xi_{\text{ini}}$ with confidence radius $r^{(e+1)} = \tilde{O}(\frac{\sqrt{n^2+nm}}{\sqrt{T_D^{(e)}\bar{\eta}^{(e)}}})$ according to Corollary 7.1.

Phase 2: pure exploitation: Compute a new robustly safe policy set with the updated model and no excitation: $\Omega^{(e)} = \Omega(\Xi^{(e+1)}, H^{(e)}, 0, \Delta_M^{(e)})$, then compute a new CCE control: $\mathbf{M}_*^{(e)} = \arg\min_{\mathbf{M} \in \Omega^{(e)}} \mathring{f}(\mathbf{M}; \hat{\xi}^{(e+1)})$.

Run *Algorithm 7.2* to safely transit from $\mathbf{M}_\dagger^{(e)}$ to $\mathbf{M}_*^{(e)}$. Set $t_2^{(e)}$ as the output.

for $t = t_2^{(e)}, \ldots, T^{(e+1)} - 1$ **do**

Implement (7.4) with policy $\mathbf{M}_\dagger^{(e)}$, noise $\eta_t = 0$ and estimate $\hat{w}_t$ by $\hat{\xi}^{(e+1)}$.

cautious-certainty-equivalence (CCE) in Line 3, where the cost function $\mathring{f}(\mathbf{M}; \hat{\xi}^{(e)})$ is introduced in (5.5) and (5.6) in Chapter 5 with estimated system $\hat{\xi}^{(e)}$ explicitly shown in the notation. Then, in Line 6, we implement $u_t = \sum_{k=1}^{H^{(e)}} \mathbf{M}_\dagger^{(e)}[k]\hat{w}_{t-k} + \eta_t$, where an exploration noise η_t is introduced to excite the system for model estimation, and approximated disturbances $\hat{w}_{t-k}$ are computed based on estimated model $\hat{\xi}^{(e)}$ along with the measured states and control inputs. We consider episode-varying memory length $H^{(e)}$ for generality. We note that $\mathbf{M}_\dagger^{(e)}$ is selected from a robustly safe policy set $\Omega_\dagger^{(e)}$ to guarantee robust constraint satisfaction for all possible systems $\xi \in \Xi^{(e)}$ and to allow safe transition from the previous policy. The safe transition is ensured by Algorithm 7.2 implemented in Line 4 and our design of $\Omega_\dagger^{(e)}$ in Section 7.2.1.

- Model estimation updates (Line 7). After Phase 1, we use the collected data to update the model estimation by ordinary least square and refine the confidence radius by Corollary 7.1. The updated model uncertainty set is $\Xi^{(e+1)} := \mathbb{B}(\hat{\xi}^{(e+1)}, r^{(e+1)}) \cap \Xi_{\text{ini}}$.

- Phase 2: Pure exploitation (Line 8-11). This phase is similar to Phase 1 but uses the new model uncertainty set $\Xi^{(e+1)}$ and removes the excitation noises η_t, i.e., $\eta_t = 0$. The CCE controller computed for this phase is denoted as $\mathbf{M}_*^{(e)}$.

7.2.1 Cautious-certainty-equivalence control

For simplicity of exposition, we drop the index of episode (e) in the notations of this subsection without causing any confusion. When the true model is unknown and only an uncertainty set $\Xi = \mathbb{B}(\hat{\xi}, r) \cap \Xi_{\text{ini}}$ is known to contain the true model, we implement DAP with approximated disturbances computed by estimated model $\hat{\xi}$ and inject an

excitation noise to encourage exploration for model estimation updates:

$$u_t = \sum_{k=1}^{H} M[k]\hat{w}_{t-k} + \eta_t, \text{ where } \|\eta_t\|_\infty \le \bar{\eta}, \text{ and } \hat{w}_t = \Pi_{\mathbb{W}}(x_{t+1} - \hat{A}x_t - \hat{B}u_t). \quad (7.4)$$

Here, the projection onto $\mathbb{W}$ is important for robust constraint satisfaction but it also introduces nonlinearity into the policy. We generate the excitation η_t i.i.d. from a distribution $\bar{\eta}\mathscr{D}_\eta$, where the distribution $\bar{\eta}\mathscr{D}_\eta$ should satisfy zero mean $\mathbb{E}[\eta_t] = 0$, bounded support $\|\eta\|_\infty \le \bar{\eta}$ with bound $\bar{\eta}$, and the (s_η, p_η)-anti-concentration property for the rescaled variable $\eta_t/\bar{\eta}$ for some $s_\eta > 0, p_\eta \in (0,1)$. Examples of the distributions include truncated Gaussian, uniform distribution. It is worth mentioning that the implementation of DAP with model estimation (7.4) requires to specify not only the DAP parameters $\mathbf{M}$ but also the estimated model $\hat{\xi}$ and the excitation level $\bar{\eta}$.

When the system is known, the safety of DAP can be ensured by the constraints in (7.3) according to Chapter 6. To ensure the safety of (7.4) *without knowing the true system*, we rely on robust constraint satisfaction, which requires safe implementation on all possible models in the uncertainty set Ξ. This can be achieved by approximating the constraint function $\mathring{g}_i^x$ by the estimated model $\hat{\xi}$ and tightening the constraints with an error term depending on the size of the uncertainty set, denoted by $\epsilon_\xi(r)$. Besides, to ensure safe exploration with excitation noises η_t, we have to further tighten the constraints by an error term $\epsilon_\eta(\bar{\eta})$. To summarize, we define a robustly safe policy set:[5]

$$\Omega(\Xi, H, \bar{\eta}, \Delta_M) = \{\mathbf{M} \in \mathbb{M}_H : \mathring{g}_i^x(\mathbf{M}; \hat{\xi}) \le d_{x,i} - \epsilon_\xi(r) - \epsilon_{\eta,x}(\bar{\eta}) - \epsilon_H(H) - \epsilon_v(\Delta_M, H), \ \forall\, i$$
$$\mathring{g}_j^u(\mathbf{M}) \le d_{u,j} - \epsilon_{\eta,u}(\bar{\eta}), \ \forall\, j\}. \quad (7.5)$$

[5]Thanks to Assumption 7.2 and $\hat{w}_t \in \mathbb{W}$, the input constraint on $\mathring{g}_j^u$ does not suffer conservativeness from the model error ϵ_η. This is because $D_{u,j}^\top u_t \le \sup_{\hat{w}_k \in \mathbb{W}} D_{u,j}^\top M[k]\hat{w}_{t-k} + D_{u,j}^\top \eta_t \le g_j^u(\mathbf{M}) + \epsilon_{\eta,u}(\bar{\eta})$.

In (7.5), the error term $\epsilon_H(H)$ is needed even when the model is known (see (7.3)), and $\epsilon_v(\Delta_M, H)$ allows for safe policy variation with variation budget Δ_M (see Lemma 7.1), which is necessary to ensure safety during policy transitions. The major challenge here is to select $\epsilon_\xi(r)$ and $\epsilon_\eta(\bar{\eta})$ so that they are large enough to guarantee robust safety despite model estimation errors and excitation noises, but not too large to degrade performances and even cause empty policy sets. The selection rules are very technical and are thus deferred to Section 7.3.2 and Appendix B.2.2. Based on (7.3) and our discussions above, we design *cautious-certainty-equivalence* control (CCE) as the solution to the optimization below, where the cost function is approximated by the estimated model $\hat{\xi}$ and the feasible region $\Omega(\Xi, H, \bar{\eta}, \Delta_M)$ guarantees robust constraint satisfaction on Ξ when implementing the policy (7.4).

$$\min_{\mathbf{M}} \mathring{f}(\mathbf{M}; \hat{\xi}), \quad \text{subject to } \mathbf{M} \in \Omega(\Xi, H, \bar{\eta}, \Delta_M). \tag{7.6}$$

7.2.2 Safe-transition algorithm design

Suppose we implement (7.4) with a robustly safe policy $\mathbf{M} \in \Omega(\Xi, H, \bar{\eta}, \Delta_M)$ with an excitation level $\bar{\eta}$, and an estimated model $\hat{\xi} \in \Xi$ before stage t_0. Our goal is to switch to a new robustly safe policy $\mathbf{M}' \in \Omega' = \Omega(\Xi', H', \bar{\eta}', \Delta_M')$ with a new $\bar{\eta}'$, and a new $\hat{\xi}' \in \Xi'$, while ensuring constraint satisfaction all the time. As discussed in Section 7.1.1, switching policies directly may lead to constraint violation even when the model is known. Here, we suffer extra challenges because the estimated models, the excitation levels, H, and the variation budgets Δ_M are changing as well. To address these challenges, we design Algorithm 7.2 for safe transitions by leveraging the slow variation trick reviewed in Chapter 5 and Section 7.1.1. We explain Algorithm 7.2 below.

We first note that W_1 and W_2 in Algorithm 7.2 ensure slow enough policy variations.

Algorithm 7.2: Safe Transition Algorithm

Input: $\mathbf{M}_{t_0-1} = \mathbf{M} \in \Omega = \Omega(\Xi, H, \bar{\eta}, \Delta_M)$, new policy $\mathbf{M}' \in \Omega' = \Omega(\Xi', H', \bar{\eta}', \Delta'_M)$, $H \le H'$.

Set $\bar{\eta}_{\min} = \min(\bar{\eta}, \bar{\eta}')$, $\hat{\xi}_{\min} = \hat{\xi}\mathbf{1}_{(r_\xi \le r'_\xi)} + \hat{\xi}'\mathbf{1}_{(r_\xi > r'_\xi)}$. Find an auxiliary policy $\mathbf{M}_{\text{mid}} \in \Omega \cap \Omega'$.

Step 1: safe transition from $\mathbf{M}$ *to* $\mathbf{M}_{mid}$. Define $W_1 = \max(\lceil \frac{\|\mathbf{M} - \mathbf{M}_{\text{mid}}\|_F}{\min(\Delta_M, \Delta'_M)} \rceil, H')$.

for $t = t_0, \dots, t_0 + W_1 - 1$ **do**

  Slowly update $\mathbf{M}_t$ from $\mathbf{M}$ towards $\mathbf{M}_{mid}$ by $\mathbf{M}_t = \mathbf{M}_{t-1} + \frac{1}{W_1}(\mathbf{M}_{\text{mid}} - \mathbf{M})$.

  Implement (7.4) with policy $\mathbf{M}_t$, noise $\eta_t \overset{\text{i.i.d.}}{\sim} \bar{\eta}_{\min}\mathscr{D}_\eta$, and estimate $\hat{w}_t$ by $\hat{\xi}_{\min}$.

Step 2: safe transition from $\mathbf{M}_{mid}$ *to* $\mathbf{M}'$. Define $W_2 = \max(\lceil \frac{\|\mathbf{M}' - \mathbf{M}_{\text{mid}}\|_F}{\Delta'_M} \rceil)$.

for $t = t_0 + W_{s_1}, \dots, t_0 + W_1 + W_2 - 1$ **do**

  Slowly update $\mathbf{M}_t$ from $\mathbf{M}_{\text{mid}}$ towards $\mathbf{M}'$ by $\mathbf{M}_t = \mathbf{M}_{t-1} + \frac{1}{W_2}(\mathbf{M}' - \mathbf{M}_{\text{mid}})$.

  Implement (7.4) with policy $\mathbf{M}_t$, noise $\eta_t \overset{\text{i.i.d.}}{\sim} \bar{\eta}'\mathscr{D}_\eta$, and estimate $\hat{w}_t$ by $\hat{\xi}'$.

Output: $t_1 = t_0 + W_1 + W_2$

Secondly, Algorithm 7.2 adopts an auxiliary policy $\mathbf{M}_{\text{mid}} \in \Omega \cap \Omega'$ to serve as a middle-ground when transiting from $\mathbf{M}$ to $\mathbf{M}'$. This guarantees $\mathbf{M}_t \in \Omega$ in Step 1 and $\mathbf{M}_t \in \Omega'$ in Step 2. To see this, notice that $\mathbf{M}_t$ in Step 1 is a convex combination of $\mathbf{M}$ and $\mathbf{M}_{\text{ini}}$, which both belong to Ω. The same applies to Step 2. Therefore, every $\mathbf{M}_t$ is in some robustly safe policy sets, which ensures constraint satisfaction along the way. In practice, one can choose $\mathbf{M}_{\text{mid}}$ that is close to $\mathbf{M}$ and $\mathbf{M}'$ to for fewer transition stages.

Thirdly, Step 1 adopts the smaller excitation level $\bar{\eta}_{\min}$ and the better estimated model $\hat{\xi}_{\min}$. This is because the approximated disturbances and excitation noises used in Step 1 will affect the state constraint satisfaction in Step 2 by the dependence of the current state on the history disturbances and excitation noises. Thus, the estimation errors and excitation levels in Step 1 should be small enough for both Ω and Ω'. Further, we can show that the effects of the history are dominated by the recent H' stages, so it suffices to select $W_1 \geq H'$ to provide small history errors for Step 2.

Remark 7.2. *In Line 8 of Algorithm 7.1, we only use a segment of data from the current episode. This is for the simplicity of theoretical analysis. In practice, one should use all the data collected so far to construct a better estimation.*

Remark 7.3. *Notice that $\Omega(\Xi, H, \bar{\eta}, \Delta_M)$ defines a polytopic set on $\boldsymbol{M}$ and $f(\mathbf{M}; \hat{\xi})$ is a convex quadratic function. Then, solving the CCE controller only requires solving a convex quadratic program with linear constraints, which admits polynomial-time solvers.*

Remark 7.4. *We mention that algorithm 7.2 is not the unique way to guarantee safe transitions. One can design other safe transition paths by e.g., borrowing ideas from model predictive control.*

Remark 7.5. *In our CCE formulation (7.6), we construct a robustly safe policy set*

(7.5) *with constraint-tightening epsilons. This can be viewed as uniform tightenings for all policies. To reduce conservativeness, one can also model the constraint-tightening epsilons as decision variables in the CCE optimization, which is left as future work.*

7.3 Theoretical analysis

The results in this section are divided into four parts. Firstly, we provide a bound on the model estimation errors by implementing general (even nonlinear) policies, based on which we establish formulas for the confidence radius in Algorithm 7.1. Secondly, we provide more theoretical discussions on the robustly safe policy set defined in (7.5). Thirdly, we establish theoretical guarantees on feasibility and constraint satisfaction. Fourth, we provide a regret upper bound. The proofs are provided in Appendix B.2.

7.3.1 Model estimation error bounds

Here, we provide a decay rate for our model estimation errors. The major technical difficulty comes from the nonlinearity of the control policies caused by the projection in (7.4). To address this issue, we provide a estimation error bound for general (even nonlinear) policies $u_t = \pi_t(x_0, \{w_k, \eta_k\}_{k=0}^{t-1}) + \eta_t$ with bounded states and actions. This generalizes the existing results on linear policies [63]. Since nonlinear policies are commonly used for constrained linear optimal control, our result may be useful for other research on constrained adaptive control.[6]

Theorem 7.1 (General estimation error bound)**.** *Consider a linear system* $x_{t+1} =$

[6]There are also recent results on estimation errors for general nonlinear systems such as [178], but our problem has special structures and thus enjoys better rates than the general nonlinear system case.

$A_*x_t + B_*u_t + w_t$ with a general controller $u_t = \pi_t(x_0, \{w_k, \eta_k\}_{k=0}^{t-1}) + \eta_t$. η_t is i.i.d., with bounded support $\|\eta_t\|_\infty \le \bar{\eta}$, and is independent from $\{w_t\}_{t\ge 0}$. Further, $\eta_t/\bar{\eta}$ satisfies the (s_η, p_η)-anti-concentration property. Suppose the system is bounded by $\|x_t\|_2 \le b_x$, $\|u_t\|_2 \le b_u$ for all t for some b_x, b_u when implementing the controller above. Let $\tilde{\xi} = \min_\xi \sum_{t=0}^{T-1} \|x_{t+1} - Ax_t - Bu_t\|_2^2$ denote the least square estimator of the true system after a trajectory of T stages. For any $0 < \delta < 1/3$, if

$$T \ge \frac{10}{p_z^2}\left(\log(1/\delta) + 2(m+n)\log(10/p_z) + 2(n+m)\log(\sqrt{b_x^2+b_u^2}/s_z)\right),$$

then

$$\|\tilde{\xi} - \xi_*\|_2 \le \frac{90\sigma_{sub}}{p_z}\frac{\sqrt{n + (n+m)\log(10/p_z) + 2(n+m)\log(\sqrt{b_x^2+b_u^2}/s_z) + \log(1/\delta)}}{\sqrt{T}s_z}$$

holds with probability $1-3\delta$, where $p_z = \min(p_w, p_\eta)$, $s_z = \min(s_w/4, \frac{\sqrt{3}}{2}s_\eta\bar{\eta}, \frac{s_w s_\eta}{4b_u}\bar{\eta})$.

Theorem 7.1 shows a decay rate $\tilde{O}(\frac{\sqrt{m+n}}{\bar{\eta}\sqrt{T}})$ as $T \to +\infty$ and $\bar{\eta} \to 0$ for the model estimation error without assuming linear policies. Interestingly, the rate is the same with the decay rate for linear policies in the literature with respect to $T, \bar{\eta}, n, m$ in [63, 144].

Applying Theorem 7.1 to Algorithm 7.1 yields the following estimation error bound.

Corollary 7.1 (Estimation errors in Algorithm 7.1)**.** *Select $\bar{\eta}^{(e)}$ such that $\bar{\eta}^{(e-1)}/2 \le \bar{\eta}^{(e)} \le \bar{\eta}^{(e-1)} \le \min(\frac{s_w}{2\sqrt{3}s_\eta}, \frac{1}{\sqrt{3}s_\eta}, \frac{2u_{\max}}{s_w s_\eta})$ for $e \ge 1$. Suppose $t_1^{(e)} + T_D^{(e)} \le T^{(e+1)}$ for $e \ge 1$. For any $0 < p < 1$, with a sufficiently large $T^{(1)} \ge \tilde{O}(m+n)$, then*

$$\|\hat{\xi}^{(e)} - \xi_*\|_F \le O\left(\frac{(n+\sqrt{mn})\sqrt{\log(mn/\bar{\eta}^{(e-1)}) + \log(e)}}{\sqrt{T_D^{(e-1)}}\bar{\eta}^{(e-1)}}\right)$$

with probability at least $1 - \frac{p}{2e^2}$ for any $e \ge 1$.

Notice that the estimation error in Corollary 7.1 is represented by the $\|\cdot\|_F$ norm. This is because our model estimation in Line 7 of Algorithm 7.1 projects $\tilde{\xi}^{(e)}$ onto

Ξ_{ini} and the matrix Frobenious norm is more convenient in theoretical analysis when projections are involved (the projection of matrices is non-expansive on $\|\cdot\|_F$). This change of matrix norms introduces an additional $\sqrt{n}$ factor to the estimation error bound in Corollary 7.1.

7.3.2 Constraint tightenings in the robustly safe policy set

Section 7.2.1 introduces the robustly safe policy set (7.5) with constraint-tightening terms $\epsilon_\xi, \epsilon_{\eta,x}, \epsilon_H, \epsilon_v, \epsilon_{\eta,u}$. This subsection will provide more details. Specifically, we provide a general decomposition lemma for the state and action constraints when implementing DAP with model uncertainties and excitation noises (7.4). This lemma is an extension from Chapter 6 (Step 1-2 in Section 6.2). Then, we utilize the decomposition lemma to explain the constraint tightening terms and bound each term.

Firstly, we provide a general decomposition lemma for state and action constraints below. For generality, we consider time-varying policies $\mathbf{M}_t$, memory lengths H_t, excitation levels $\bar{\eta}_t$, estimated models $\hat{\xi}_t$ for disturbance estimation, and allow to use different estimated models $\hat{\xi}_t^g$ to approximate constraint functions.

Lemma 7.2 (General constraint decomposition lemma)**.** *Consider time-varying DAPs* $\{\mathbf{M}_t\}_{t\geq 0}$*, where* $\mathbf{M}_t \in \mathbb{M}_{H_t}$ *and* $\{H_t\}_{t\geq 0}$ *is non-decreasing. Consider implementing* $\{\mathbf{M}_t\}_{t\geq 0}$ *by* (7.4) *with disturbances* $\hat{w}_t$ *estimated by time-varying estimated models* $\hat{\xi}_t$ *and excitation noises with time-varying excitation levels* $\|\eta_t\|_\infty \leq \bar{\eta}_t$*, that is,*

$$u_t = \sum_{t=1}^{H_t} M_t[k]\hat{w}_{t-k} + \eta_t, \quad \hat{w}_t = \Pi_{\mathbb{W}}(x_{t+1} - \hat{\xi}_t z_t), \quad \|\eta_t\|_\infty \leq \bar{\eta}_t, \quad t \geq 0. \tag{7.7}$$

Suppose the true system parameter is ξ_**, then, the left-hand-sides of the state constraint*

$D_{x,i}^\top x_t$ and action constraint $D_{u,j}^\top u_t$ can be decomposed by

$$D_{x,i}^\top x_t \le \underbrace{\mathring{g}_i^x(\mathbf{M}_t;\hat{\xi}_t^g)}_{\text{estimated state constraint function}} \underbrace{+(\mathring{g}_i^x(\mathbf{M}_t;\xi_*)-\mathring{g}_i^x(\mathbf{M}_t;\hat{\xi}_t^g)) + \sum_{k=1}^{H_t} D_{x,i}^\top A_*^{k-1}(w_{t-k}-\hat{w}_{t-k})}_{\text{(i) model estimation error}}$$
$$+ \underbrace{\sum_{i=1}^{H_t} D_{x,i}^\top A_*^{i-1} B_* \eta_{t-i}}_{\text{(ii) error from excitation noises}} + \underbrace{D_{x,i}^\top A_*^{H_t} x_{t-H_t}}_{\text{(iii) history truncation error}} \underbrace{+(g_i^x(\mathbf{M}_{t-H_t:t-1};\xi_*)-\mathring{g}_i^x(\mathbf{M}_t;\xi_*))}_{\text{(iv) policy variation error}}$$

$$D_{u,j}^\top u_t \le \underbrace{\mathring{g}_j^u(\mathbf{M}_t)}_{\text{control constraint function}} + \underbrace{D_{u,j}^\top \eta_t}_{\text{(v) error from excitation noises}},$$

where $\hat{\xi}_t^g$ is an estimated model to approximate the state constraint function, which can be different from the estimated model $\hat{\xi}_t$ used for disturbance approximations.

Let's first consider a state constraint $D_{x,i}^\top x_t \le d_{x,i}$. Notice that Lemma 7.2 decomposes the left-hand-side of the constraint into the constraint function $\mathring{g}_i^x(\mathbf{M}_t;\hat{\xi}_t^g)$ and four error terms. If the sum of the error terms can be bounded by ϵ despite all uncertainties, then the policy constraint tightened by ϵ, i.e., $\mathring{g}_i^x(\mathbf{M}_t;\hat{\xi}_t^g) \le d_{x,i} - \epsilon$, can guarantee robust satisfaction of the state constraint $D_{x,i}^\top x_t \le d_{x,i}$. The action constraints are the same. This motivates our design of the robustly safe policy set in (7.5).

Next, we show that the upper bounds on the error terms above correspond to the constraint-tightening terms used in (7.5).

Lemma 7.3 (Constraint-tightening terms)**.** *Under the conditions in Lemma 7.2. Consider stage t, for any $0 \le k \le H_t$, suppose $x_{t-k} \in \mathbb{X}, u_{t-k} \in \mathbb{U}, \|\eta_t\|_\infty \le \bar{\eta}, \hat{\xi}_{t-k} \in \mathbb{B}(\xi_*, r) \cap \Xi_{ini}, H_t \le H$. Further, suppose $\hat{\xi}_t^g \in \mathbb{B}(\xi_*, r) \cap \Xi_{ini}$ and $\Delta_M \ge \max_{1\le k\le H_t} \frac{\|\mathbf{M}_t - \mathbf{M}_{t-k}\|_F}{k}$. Then, we have*

(i) model estimation error $\le \epsilon_\xi(r) = O(\sqrt{mn}r)$

(ii) error from excitation noises in the state constraint $\leq \epsilon_{\eta,x} = O(\sqrt{m}\bar{\eta})$

(iii) history truncation error $\leq \epsilon_H(H) = O((1-\gamma)^H)$

(iv) policy variation error $\leq \epsilon_v(\Delta_M, H) = O(\sqrt{mnH}\Delta_M)$

(v) error from excitation noises in the control constraint $\leq \epsilon_{\eta,u} = O(\bar{\eta})$

where $O(\cdot)$ hide polynomial factors of $\kappa, \|D_x\|_\infty, \|D_u\|_\infty, \bar{\eta}, w_{\max}, \kappa_B, x_{\max}, u_{\max}, \gamma^{-1}$.

We defer the formulas of the hidden factors and the proof to Appendix B.2.2.

7.3.3 Feasibility and constraint satisfaction

Now, we provide feasibility and constraint satisfaction guarantees under proper algorithm inputs. For simplicity, we assume r_{ini} is small enough, which is commonly assumed in the literature [66] and can be replaced by implementing a safe exploration policy for sufficiently long to reduce the model estimation error.

Assumption 7.4 (Assumption on r_{ini})**.** *r_{ini} is small enough such that $\epsilon_\xi(r_{ini}) \leq \frac{\epsilon_F}{4}$.*

Theorem 7.2 (General conditions for feasibility)**.** *The policies computed by Algorithm 7.1 and Algorithm 7.2 are well-defined for all stages if the following conditions hold.*

$$T_D^{(e)} \geq T_D^{(e-1)}, \bar{\eta}^{(e)} \leq \bar{\eta}^{(e-1)}, H^{(e)} \geq H^{(e-1)}, \Delta_M^{(e)} \leq \sqrt{\frac{H^{(e-1)}}{H^{(e)}}}\Delta_M^{(e-1)}, r^{(e)} \leq r^{(e-1)}, \forall\, e \geq 1, \quad \text{(I)}$$

$$\epsilon_H(H^{(0)}) + \epsilon_P(H^{(0)}) + \epsilon_v(\Delta_M^{(0)}, H^{(0)}) + \epsilon_{\eta,x}(\bar{\eta}^{(0)}) \leq \epsilon_F/2 \quad \text{(II)}$$

$$\epsilon_{\eta,u}(\bar{\eta}^{(0)}) + \epsilon_P(H^{(0)}) \leq \epsilon_F, \quad \text{(III)}$$

$$H^{(0)} \geq \log(2\kappa)/\log((1-\gamma)^{-1}), \quad \text{(IV)}$$

where $\epsilon_P(H) = O(\sqrt{n}(1-\gamma)^H)$ corresponds to $\epsilon_3 + \epsilon_1$ in Corollary 6.2 in Chapter 6.[7]

[7] ϵ_P's dependence on the dimension is reduced here since we consider a simpler time-invariant open-loop stable LQR problem in this chapter.

Conditions (I) in Theorem 7.2 requires monotonicity of the algorithm parameters, which allows us to verify the feasibility based on the initial conditions. Condition (II) and (III) require large enough $H^{(0)}$, small enough $\bar{\eta}^{(0)}$ and $\Delta_M^{(0)}$. Notice that $r^{(1)} \leq r^{(0)} = r_{\text{ini}}$ in (I) requires $\tilde{O}(\sqrt{n^2+nm}(\sqrt{T_D^{(e-1)}}\bar{\eta}^{(e-1)})^{-1}) \leq r_{\text{ini}}$, which suggests that $\bar{\eta}^{(0)}$ should not be too small. This reflects the trade-off between exploration and safety.

Theorem 7.3 (Constraint Satisfaction)**.** *Under the conditions in Theorem 7.2, Corollary 7.1, supposing $t_2^{(e)} \leq T^{(e+1)}$, then we have $u_t \in \mathbb{U}$ for all $t \geq 0$ with probability (w.p.) 1. Further, $x_t \in \mathbb{X}$ holds for all $t \geq 0$ w.p. at least $1-p$, where p is defined in Corollary 7.1.*

Notice that our control inputs satisfy the constraints w.p. 1. This is provided by the projection of the estimated disturbance onto $\mathbb{W}$ in (7.4). Besides, we can show that the state constraints are satisfied if the true model is inside the confidence sets $\Xi^{(e)}$ for all $e \geq 0$, whose probability is at least $1-p$ by Corollary 7.1.

7.3.4 Regret guarantees

Next, we provide a $\tilde{O}(T^{2/3})$ regret bound while guaranteeing feasibility and constraint satisfaction. Further, we explain the reasons behind the pure exploitation phase.

Theorem 7.4 (Regret bound)**.** *Consider any $0 < p < 1/2$. Let $T_D^{(e)} = (T^{(e+1)} - T^{(e)})^{2/3}$, $T^{(1)} \geq \tilde{O}((\sqrt{nm}+n)^3)$. Set $\Delta_M^{(e)} = O(\frac{\epsilon_F^x}{\sqrt{mnH^{(0)}}}(T^{(e+1)})^{-1/3})$, $H^{(e)} \geq O(\log(\max(T^{(e+1)}, \frac{\sqrt{n}}{\min(\epsilon_F)})))$, $\bar{\eta}^{(e)} = \eta_{\max} \leq \min\left(O(\frac{\epsilon_F^x}{\sqrt{m}}), O(\epsilon_F^u), \frac{s_w}{2\sqrt{3}s_\eta}, \frac{1}{\sqrt{3}s_\eta}, \frac{2u_{\max}}{s_w s_\eta}\right)$. Then Algorithm 7.1 is feasible and satisfies $\{u_t \in \mathbb{U}\}_{t\geq 0}$ a.s. and $\{x_t \in \mathbb{X}\}_{t\geq 0}$ w.p. $1-p$. Further, with probability at least $1-2p$,*

$$\text{Regret}_T^a \leq \tilde{O}((n^2m^{1.5}+n^{2.5}m)\sqrt{mn+k_c}T^{2/3})$$

Though our regret bound $\tilde{O}(T^{2/3})$ is worse than the $\tilde{O}(\sqrt{T})$ regret bound for *unconstrained* LQR, it is the same with the robust learning of unconstrained LQR (see [63]). This motivates future work on fundamental lower bounds of learning-based control with safety/robustness guarantees.

Proof ideas. The formal proof is provided in Appendix B.2.4. The main idea behind our proof is summarized below as an optimality gap (single-stage regret) between a CCE controller and the optimal cost J^*.

Lemma 7.4 (Cost error bound for CCE control)**.** *Consider a model uncertainty set* $\Xi = \{\xi : \|\xi - \hat{\xi}\|_F \le r\}$ *that contains the true model* ξ_**. Consider a CCE control*

$$\begin{aligned} \mathbf{M}_{cce} = \ & \arg\min_{\mathbf{M}\in\mathbb{M}_H} \ f(\mathbf{M};\hat{\xi}) \\ s.t. \ \ & g_i^x(\mathbf{M};\xi) \le d_{x,i} - \epsilon_\xi(r) - \epsilon_{\eta,x}(\bar{\eta}) - \epsilon_H(H) - \epsilon_v(\Delta_M, H), \ \forall\, i \\ & g_j^u(\mathbf{M};\xi) \le d_{u,j} - \epsilon_{\eta,u}(\bar{\eta}), \quad \forall\, j \end{aligned}$$

Then, we have

$$f(\mathbf{M}_{cce};\xi_*) - J^* \le \tilde{O}(r + \Delta_M + \bar{\eta}).$$

With this lemma, one can show that $\mathbf{M}_*^{(e)}$ in Phase 2 at episode e of Algorithm 7.1 only generates $\tilde{O}(T^{(e)}(r^{(e+1)} + \Delta_M^{(e)}))$ regret, since $\bar{\eta} = 0$ in Phase 2 and the length of Phase 2 is upper bounded by $T^{(e)}$. Further, we have $r^{(e+1)} = \tilde{O}(\frac{1}{T_D^{(e)}\eta_{\max}}) = \tilde{O}(\frac{1}{(T^{(e)})^{1/3}})$ by Corollary 7.1 and $\Delta_M^{(e)} = \tilde{O}(\frac{1}{(T^{(e)})^{1/3}})$ by our parameter choices. Consequently, the regret in Phase 2 of the episode e can be bounded by $\tilde{O}((T^{(e)})^{2/3})$, which sums up to $\tilde{O}(T^{2/3})$ in total. The regret in the remaining stages can be also bounded by $\tilde{O}(T^{2/3})$, because the total number of the remaining stages is $\tilde{O}(T^{2/3})$ and the regret generated by one stage is $O(1)$ due to the boundedness of the states and actions. This explains our $\tilde{O}(T^{2/3})$ bound.

More discussions on the choices $\eta^{(e)}$ and Phase 2 (pure exploitation). Our Algorithm 7.1 includes a pure exploitation phase with no excitation noises and an active exploration phase with a constant $\bar{\eta}^{(e)}$. However, in most literature that considers certainty-equivalence-based learning, the exploration level $\bar{\eta}^{(e)}$ decreases with e and there is no full-exploitation phase. In fact, our first attempt of algorithm design also considered decreasing $\bar{\eta}^{(e)}$ and no exploitation phase. However, such design can only achieve $\tilde{O}(T^{3/4})$ regret, which is worse than $\tilde{O}(T^{2/3})$. Intuitive explanations are provided below. Suppose we implement CCE with excitation level $\bar{\eta}^{(e)}$ throughout episode e, by Lemma 7.4, the regret at episode e is roughly $\tilde{O}(T^{(e)}(\bar{\eta}^{(e)} + r^{(e)}))$ (we ignore $\Delta_M^{(e)}$ here for simplicity). By Corollary 7.1, we have $r^{(e)} = \tilde{O}(\frac{1}{\sqrt{T^{(e-1)}}\bar{\eta}^{(e-1)}})$. Therefore, the regret in all episodes is roughly $\sum_e (\frac{1}{\sqrt{T^{(e-1)}}\bar{\eta}^{(e-1)}} + \bar{\eta}^{(e)})T^{(e)} \approx \sum_e (\frac{1}{\sqrt{T^{(e)}}\bar{\eta}^{(e)}} + \bar{\eta}^{(e)})T^{(e)}$. To minimize the regret, we let $\frac{1}{\sqrt{T^{(e)}}\bar{\eta}^{(e)}} = \bar{\eta}^{(e)}$, which leads to $\bar{\eta}^{(e)} = (T^{(e)})^{-1/4}$ and a $\tilde{O}(T^{3/4})$ regret bound.

7.4 Conclusion, extension, and future work

This chapter presents an adaptive control algorithm for constrained linear quadratic regulators. Our algorithm learns and updates the system estimation on a single trajectory without violating the constraints. To achieve this, we design a safe transition algorithm to ensure constraint satisfaction during the updates of system estimation and policies. Theoretically, we provide guarantees on feasibility and constraint satisfaction of our algorithm. We also provide a a $\tilde{O}(T^{2/3})$ regret bound compared with linear static policies. The major technical novelty is a general estimation error bound when implementing general (even nonlinear) policies.

In [179], we extend the benchmark policies to include linear dynamical policies

and a special RMPC scheme designed in [52]. We also briefly discuss how to handle nonzero initial points. For future work, we hope to compare with more general RMPC algorithms in our regret bounds, and focus more on expanding the feasible region of initial point x_0. Further, we want to study other performance measures to investigate the control performance when T is small. Other future directions include understanding the fundamental regret lower bound, relaxing the assumptions, reducing the conservativeness of our algorithm by incorporating the constraint tightening terms as optimization decision variables, robust stability guarantees, and considering nonlinear systems.

Part III

Learning to Cooperate under Limited Communication and Partial Observation

Chapter 8 | Distributed Learning of Decentralized Linear Quadratic Control

This chapter considers a distributed reinforcement learning problem for decentralized linear quadratic control with partial state observations and local costs. We propose ZODPO algorithm that learns linear local controllers in a distributed fashion, leveraging the ideas of policy gradient, zero-order optimization and consensus algorithms. ZODPO only requires limited communication and storage even in large-scale systems. Further, we show that the sample complexity to approach a stationary point is polynomial with the error tolerance's inverse and the problem dimensions, demonstrating the scalability of ZODPO. We also show that the controllers generated throughout ZODPO are stabilizing controllers with high probability. Lastly, we numerically test ZODPO on multi-zone HVAC systems.

Chapter outline. In Section 8.1, we provide introduction to this chapter. In Section 8.2, we formally introduce the problem formulation. In Section 8.3, we present our algorithm design and discussions. In Section 8.4, we provide theoretical results including the sample complexity and the stability. Section 8.5 provides numerical results. Section 8.6 concludes this chapter and discusses future work.

8.1 Introduction

Reinforcement learning (RL) has emerged as a promising tool for controller design for dynamical systems, especially when the system model is unknown or complex, and has wide applications in, e.g., robotics [180], games [130], manufacturing [181], autonomous driving [182]. However, theoretical performance guarantees of RL are still under-developed across a wide range of problems, limiting the application of RL to real-world systems. Recently, there have been exciting theoretical results on learning-based control for (centralized) linear quadratic (LQ) control problems [59,60,62]. LQ control is one of the most well-studied optimal control problems, which considers optimal state feedback control for a linear dynamical system such that a quadratic cost on the states and control inputs is minimized over a finite or infinite horizon [183].

Encouraged by the recent success of learning-based centralized LQ control, this chapter aims to extend the results and develop scalable learning algorithms for decentralized LQ control. In decentralized control, the global system is controlled by a group of individual agents with limited communication, each of which observes only a partial state of the global system [184]. Decentralized LQ control has many applications, including transportation [185], power grids [186], robotics [187], smart buildings [1], etc. It is worth mentioning that partial observations and limited communication place major challenges on finding optimal decentralized controllers, even when the global system model is known [188, 189].

Specifically, we consider the following decentralized LQ control setting. Suppose a linear dynamical system, with a global state $x(t) \in \mathbb{R}^n$ and a global control action $u(t)$,

is controlled by a group of agents. The global control action is composed of local control actions: $u(t) = \left[u_1(t)^\top, \ldots, u_N(t)^\top\right]^\top$, where $u_i(t)$ is the control input of agent i. At time t, each agent i directly observes a partial state $x_{\mathscr{I}_i}(t)$ and a quadratic local cost $c_i(t)$ that could depend on the global state and action. The dynamical system model is assumed to be unknown, and the agents can only communicate with their neighbors via a communication network. The goal is to design a cooperative distributed learning scheme to find local control policies for the agents to minimize the global cost that is averaged both among all agents and across an infinite horizon. The local control policies are limited to those that only use local observations.

8.1.1 Our contributions

We propose a Zero-Order Distributed Policy Optimization algorithm (ZODPO) for the decentralized LQ control problem defined above. ZODPO only requires limited communication over the network and limited storage of local policies, thus being applicable for large-scale systems. Roughly, in ZODPO, each agent updates its local control policy using estimate of the partial gradient of the global objective with respect to its local policy. The partial gradient estimation leverages zero-order optimization techniques, which only requires cost estimation of a perturbed version of the current policies. To ensure distributed learning/estimation, we design an approximate sampling method to generate policy perturbations under limited communication among agents; we also develop a consensus-based algorithm to estimate the infinite-horizon global cost by conducting the spatial averaging (of all agents) and the temporal averaging (of infinite horizon) at the same time.

Theoretically, we provide non-asymptotic performance guarantees of ZODPO. For

technical purposes, we consider static linear policies, i.e. $u_i(t) = K_i x_{\mathscr{I}_i}(t)$ for a matrix K_i for agent i, though ZODPO can incorporate more general policies.

Specifically, we show that, to approach some stationary point with error tolerance ϵ, the required number of samples is $O(n_K^3 \max(n, N)\epsilon^{-4})$, where n_K is the dimension of the policy parameter, N is the number of agents and n is the dimension of the state. The polynomial dependence on the problem dimensions indicates the scalability of ZODPO. To the best of our knowledge, this is the first sample complexity result for distributed learning algorithms for the decentralized LQ control considered in this chapter. In addition, we prove that all the policies generated and implemented by ZODPO are stabilizing with high probability, guaranteeing the safety during the learning process.

Numerically, we test ZODPO on multi-zone HVAC systems to demonstrate the optimality and safety of the controllers generated by ZODPO.

8.1.2 Related work

There are numerous studies on related topics including learning-based control, decentralized control, multi-agent reinforcement learning, etc., which are reviewed below.

a) *Learning-based LQ control:* Controller design without (accurate) model information has been studied in the fields of adaptive control [190] and extremum-seeking control [191] for a long time, but most papers focus on stability and asymptotic performance. Recently, much progress has been made on algorithm design and nonasymptotic analysis for learning-based centralized (single-agent) LQ control with full observability, e.g., model-free schemes [59,192,193], identification-based controller design [60,61], Thompson sampling [62], etc.; and with partial observability [61, 194]. As for learning-based

decentralized (multi-agent) LQ control, most studies either adopt a centralized learning scheme [195] or still focus on asymptotic analysis [196–198]. Though, [199] proposes a distributed learning algorithm with a nonasymptotic guarantee, the algorithm requires agents to store and update the model of the whole system, which is prohibitive for large-scale systems.

Our algorithm design and analysis are related to policy gradient for centralized LQ control [59, 192, 195]. Though policy gradient can reach the global optimum in the centralized setting because of the gradient dominance property [59], it does not necessarily hold for decentralized LQ control [200], and thus we only focus on reaching stationary points as most other papers did in nonconvex optimization [201].

b) *Decentralized control:* Even with model information, decentralized control is very challenging. For example, the optimal controller for general decentralized LQ problems may be nonlinear [188], and the computation of such optimal controllers mostly remains unsolved. Even for the special cases with linear optimal controllers, e.g., the quadratic invariance cases, one usually needs to optimize over an infinite dimensional space [189]. For tractability, many papers, including this one, consider finite dimensional linear policy spaces and study suboptimal controller design [202–204].

c) *Multi-agent reinforcement learning:* There are various settings for multi-agent reinforcement learning (MARL), and our problem is similar to the cooperative setting with partial observability, also known as Dec-POMDP [205]. Several MARL algorithms have been developed for Dec-POMDP, including centralized learning decentralized execution approaches, e.g. [206], and decentralized learning decentralized execution approaches, e.g. [207, 208]. Our proposed algorithm can be viewed as a decentralized

learning decentralized execution approach.

In addition, most cooperative MARL papers for Dec-POMDP assume global cost (reward) signals for agents [206–208]. However, this chapter considers that agents only receive local costs and aim to minimize the averaged costs of all agents. In this sense, our setting is similar to [209], but [209] assumes global state and global action signals.

d) *Policy gradient approaches:* Policy gradient and its variants are popular in both RL and MARL. Various gradient estimation schemes have been proposed, e.g., REINFORCE [210], policy gradient theorem [211], deterministic policy gradient theorem [212], zero-order gradient estimation [59], etc. this chapter adopts the zero-order gradient estimation, which has been employed for learning centralized LQ control [59, 192, 213].

e) *Zero-order optimization:* It aims to solve optimization without gradients by, e.g., estimating gradients based on function values [214–218]. This chapter adopts the gradient estimator in [214]. However, due to the distributed setting and communication constraints, we cannot sample policy perturbations exactly as in [214], since the global objective value is not revealed directly but has to be learned/estimated. Besides, we have to ensure the controllers' stability during the learning procedure, which is an additional requirement not considered in the optimization literature [214].

Notations: $\mathrm{vec}((M_i)_{i=1}^N) = [\mathrm{vec}(M_1)^\top, \ldots, \mathrm{vec}(M_N)^\top]^\top$, where $M_1, \ldots, M_N$ are arbitrary matrices. We use $\mathbf{1}$ to denote the vector with all one entries, and $I_p \in \mathbb{R}^{p\times p}$ to denote the identity matrix. The unit sphere $\{x \in \mathbb{R}^p : \|x\| = 1\}$ is denoted by $\mathbb{S}_p$, and $\mathrm{Uni}(\mathbb{S}_p)$ denotes the uniform distribution on $\mathbb{S}_p$. For any $x \in \mathbb{R}^p$ and any subset $S \subset \mathbb{R}^p$, we denote $x + S \coloneqq \{x + y : y \in S\}$. The indicator function of a random event S will be

denoted by $\mathbf{1}_S$ such that $\mathbf{1}_S = 1$ when the event S occurs and $\mathbf{1}_S = 0$ otherwise.

8.2 Problem formulation

Suppose there are N agents jointly controlling a discrete-time linear system of the form

$$x(t+1) = Ax(t) + Bu(t) + w(t), \quad t = 0, 1, 2, \dots, \tag{8.1}$$

where $x(t) \in \mathbb{R}^n$ denotes the state vector, $u(t) \in \mathbb{R}^m$ denotes the joint control input, and $w(t) \in \mathbb{R}^n$ denotes the random disturbance at time t. We assume $w(0), w(1), \dots$ are i.i.d. from the Gaussian distribution $\mathscr{N}(0, \Sigma_w)$ for some positive definite matrix Σ_w. Each agent i is associated with a local control input $u_i(t) \in \mathbb{R}^{m_i}$, which constitutes the global control input $u(t) = \left[u_1(t)^\top, \dots, u_N(t)^\top\right]^\top \in \mathbb{R}^n$.

We consider the case where each agent i only observes a partial state, denoted by $x_{\mathscr{I}_i}(t) \in \mathbb{R}^{n_i}$, at each time t, where $\mathscr{I}_i$ is a fixed subset of $\{1, \dots, n\}$ and $x_{\mathscr{I}_i}(t)$ denotes the subvector of $x(t)$ with indices in $\mathscr{I}_i$.[1] The admissible local control policies are limited to the ones that only use the historical local observations. As a starting point, this chapter only considers static linear policies that use the current observation, i.e., $u_i(t) = K_i x_{\mathscr{I}_i}(t)$.[2] For notational simplicity, we define

$$\mathsf{K} \coloneqq \mathrm{vec}\big((K_i)_{i=1}^N\big) \in \mathbb{R}^{n_K}, \qquad n_K \coloneqq \sum\nolimits_{i=1}^N n_i m_i. \tag{8.2}$$

It is straightforward to see that the global control policy is also a static linear policy on the current state. We use $\mathscr{M}(\mathsf{K})$ to denote the global control gain, i.e., $u(t) = \mathscr{M}(\mathsf{K})x(t)$.

[1] $\mathscr{I}_i$ and $\mathscr{I}_{i'}$ may overlap. Our results can be extended to general case $y_i(t) = C_i x(t)$.

[2] The framework and algorithm can be extended to other policy classes, but analysis is left for future.

Note that $\mathscr{M}(\mathsf{K})$ is often sparse in network control applications. Figure 8.1 gives an illustrative example of the our control setup.

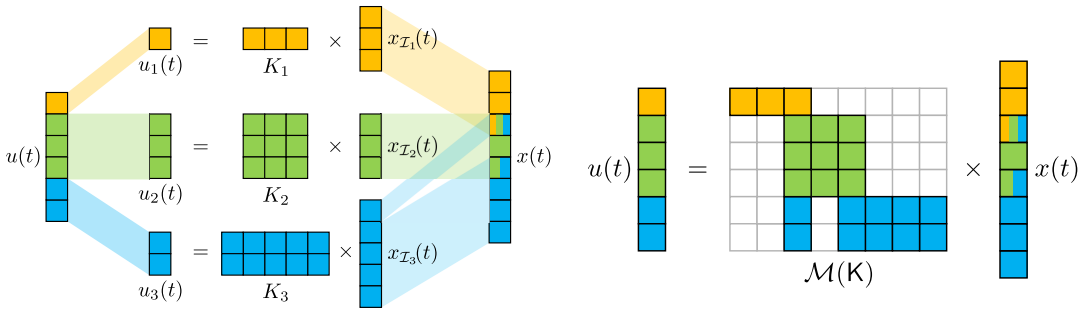


Figure 8.1: An illustrative diagram for $N = 3$ agents, where $x(t) \in \mathbb{R}^8$, $u(t) \in \mathbb{R}^6$, and $\mathscr{I}_1 = \{1, 2, 3\}, \mathscr{I}_2 = \{3, 4, 5\}, \mathscr{I}_3 = \{3, 5, 6, 7, 8\}$. The top figure illustrates the local control inputs, local controllers, and local observations; and the bottom figure provides a global viewpoint of the resulting controller $\mathscr{M}(\mathsf{K})$.

At each time step t, agent i receives a quadratic local stage cost $c_i(t)$ given by $c_i(t) = x(t)^\top Q_i x(t) + u(t)^\top R_i u(t),$, which is allowed to depend on the global state $x(t)$ and control $u(t)$. The goal is to find a control policy that minimizes the infinite-horizon average cost among all agents, that is,

$$\begin{aligned} \min_{\mathsf{K}} \quad & J(\mathsf{K}) := \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} c_i(t)\right] \\ \text{s.t.} \quad & x(t+1) = Ax(t) + Bu(t) + w(t),\ u_i(t) = K_i x_{\mathscr{I}_i}(t),\ i = 1, \ldots, N. \end{aligned} \tag{8.3}$$

When the model parameters are known, the problem (8.3) can be viewed as a decentralized LQ control problem, which is known to be a challenging problem in general. Various heuristic or approximate methods have been proposed (see Section 8.1.2), but most of them require accurate model information that may be hard to obtain in practice. Motivated by the recent progress in learning based control and also the fact that the models are not well-studied or known for many systems, this chapter

studies learning-based decentralized control for (8.3), where each agent i learns the local controller K_i by utilizing the partial states $x_{\mathscr{I}_i}(t)$ and local costs $c_i(t)$ observed along the system's trajectories.

In many real-world applications of decentralized control, limited communication among agents is available via a communication network. Here, we consider a connected and undirected communication network $\mathscr{G}=(\{1,\dots,N\},\mathscr{E})$, where each node represents an agent and $\mathscr{E}$ denotes the set of edges. At each time t, agent i and j can directly communicate a small number of scalars to each other if and only if $(i,j)\in\mathscr{E}$. Further, we introduce a doubly-stochastic and nonnegative communication matrix $W=[W_{ij}]\in\mathbb{R}^{N\times N}$ associated with the communication network $\mathscr{G}$, with $W_{ij}=0$ if $(i,j)\notin\mathscr{E}$ for $i\neq j$ and $W_{ii}>0$ for all i. The construction of the matrix W has been extensively discussed in literature (see, for example, [219]). We denote

$$\rho_W := \left\| W - \frac{1}{N}\mathbf{1}\mathbf{1}^\top \right\|. \tag{8.4}$$

This quantity captures the convergence rate of the consensus via W and is known to be within $[0,1)$ [219, 220].

Finally, we introduce the technical assumptions.

Assumption 8.1. *The dynamical system (A,B) is controllable. The cost matrices Q_i, R_i are positive semidefinite for each i, and the global cost matrices $\frac{1}{N}\sum_{i=1}^N Q_i$ and $\frac{1}{N}\sum_{i=1}^N R_i$ are positive definite.*

Assumption 8.2. *There exists a control policy $\mathsf{K}\in\mathbb{R}^{n_K}$ such that the resulting global dynamics $x(t+1)=(A+B\mathscr{M}(\mathsf{K}))x(t)$ is asymptotically stable.*

Both assumptions are common in LQ control literature. Without Assumption 8.2,

the problem (8.3) does not admit a reasonable solution even if all system parameters are known, let alone learning-based control.[3] For ease of exposition, we denote $\mathscr{K}_{\rm st}$ as the set of stabilizing controller, i.e.,

$$\mathscr{K}_{\rm st} := \{\mathsf{K} \in \mathbb{R}^{n_K} : A + B\mathscr{M}(\mathsf{K}) \text{ is asymptotically stable}\}.$$

8.3 Algorithm design

8.3.1 Review: zero-order policy gradient for centralized LQR

To find a policy K that minimizes $J(\mathsf{K})$, one common approach is the policy gradient method, that is,

$$\mathsf{K}(s+1) = \mathsf{K}(s) - \eta\hat{\mathsf{g}}(s), \quad s = 1, 2, \ldots, \qquad \mathsf{K}(1) = \mathsf{K}_0,$$

where $\hat{\mathsf{g}}(s)$ is an estimator of the gradient $\nabla J(\mathsf{K}(s))$, $\eta > 0$ is a stepsize, and K_0 is some known stabilizing controller. In [59] and [192], the authors have proposed to employ gradient estimators from zero-order optimization. One example is:

$$\mathsf{G}^r(\mathsf{K}, \mathsf{D}) := \frac{n_K}{r} J(\mathsf{K} + r\mathsf{D})\mathsf{D} \tag{8.5}$$

for $\mathsf{K} \in \mathscr{K}_{\rm st}$ and $r > 0$ such that $\mathsf{K} + r\mathbb{S}_{n_K} \subseteq \mathscr{K}_{\rm st}$, where $\mathsf{D} \in \mathbb{R}^{n_K}$ is randomly sampled from $\mathrm{Uni}(\mathbb{S}_{n_K})$. The parameter r is sometimes called the smoothing radius, and it can be shown that the bias $\|\mathbb{E}_{\mathsf{D}}[\mathsf{G}^r(\mathsf{K}, \mathsf{D})] - \nabla J(\mathsf{K})\|$ can be controlled by r under certain smoothness conditions on $J(\mathsf{K})$ [192]. The policy gradient based on the estimator (8.5)

[3] If Assumption 8.2 does not hold but the system is stabilizable, then one has to consider more general controller structures, e.g. linear dynamic or nonlinear controllers, which are left for future.

is given by

$$\begin{aligned}\mathsf{K}(s+1) &= \mathsf{K}(s) - \eta\, \mathsf{G}^r(\mathsf{K}(s), \mathsf{D}(s)) \\ &= \mathsf{K}(s) - \eta \cdot \frac{n_K}{r} J(\mathsf{K}(s) + r\mathsf{D}(s))\mathsf{D}(s),\end{aligned} \tag{8.6}$$

where $\{\mathsf{D}(s)\}_{s=1}^{T_G}$ are i.i.d. random vectors from $\mathrm{Uni}(\mathbb{S}_{n_K})$.

8.3.2 Our algorithm: zero-order distributed policy optimization

Algorithm 8.1: Zero-Order Distributed Policy Optimization (ZODPO)

Input: smoothing radius r, η, $\bar{J} > 0$, T_G, T_J, initial controller $\mathsf{K}_0 \in \mathscr{K}_{\mathrm{st}}$.

Initialize $\mathsf{K}(1) = \mathsf{K}_0$.

for $s = 1, 2, \ldots, T_G$ **do**

`// Step 1: Sampling from the unit sphere`

Each agent i generates $D_i(s) \in \mathbb{R}^{m_i \times n_i}$ by the subroutine `SampleUSphere`.

`// Step 2: Local estimation of the global objective`

Run `GlobalCostEst`$((K_i(s) + rD_i(s))_{i=1}^N, T_J)$, and let agent i's returned value be denoted by $\tilde{J}_i(s)$.

`// Step 3: Local estimation of partial gradients`

Each agent i estimates the partial gradient $\frac{\partial J}{\partial K_i}(\mathsf{K}(s))$ by

$$\hat{J}_i(s) = \min\left\{\tilde{J}_i(s), \bar{J}\right\}, \quad \hat{G}_i^r(s) = \frac{n_K}{r}\hat{J}_i(s) D_i(s).$$

`// Step 4: Distributed policy gradient on local controllers`

Each agent i updates $K_i(s+1)$ by $K_i(s+1) = K_i(s) - \eta \hat{G}_i^r(s)$..

Now, let us consider the decentralized LQ control formulated in Section 8.2. Notice that Iteration (8.6) can be equivalently written in an almost decoupled way for each

agent i:

$$K_i(s+1) = K_i(s) - \eta \cdot \frac{n_K}{r} J(\mathsf{K}(s) + r\mathsf{D}(s)) D_i(s), \tag{8.7}$$

where $\mathsf{D}(s) \sim \mathrm{Uni}(\mathbb{S}_{n_K})$, $K_i(s), D_i(s) \in \mathbb{R}^{n_i \times m_i}$ and

$$\mathsf{K}(s) = \mathrm{vec}((K_i(s))_{i=1}^N), \quad \mathsf{D}(s) = \mathrm{vec}((D_i(s))_{i=1}^N). \tag{8.8}$$

The formulation (8.7) suggests that, if each agent i can sample $D_i(s)$ properly and obtain the value of the global objective $J(\mathsf{K}(s) + r\mathsf{D}(s))$, then the policy gradient (8.6) can be implemented in a decentralized fashion by letting each agent i update its own policy K_i in parallel according to (8.7). This key observation leads us to the ZODPO algorithm (Algorithm 8.1).

Roughly, ZODPO conducts distributed policy gradient with four main steps:

- In *Step 1*, each agent i runs the subroutine `SampleUSphere` to generate a random matrix $D_i(s) \in \mathbb{R}^{m_i \times n_i}$ so that the concatenated $\mathsf{D}(s)$ approximately follows the uniform distribution on $\mathbb{S}_{n_K}$. In the subroutine `SampleUSphere`, each agent i samples a Gaussian random matrix V_i independently, and then employs a simple consensus procedure (8.9) to compute the averaged squared norm $\frac{1}{N}\sum_i \|V_i\|_F^2$. Our analysis shows that the outputs of the subroutine approximately follow the desired distribution for sufficiently large T_S (see Lemma C.1 in Section C.1.1).
- In *Step 2*, each agent i estimates the global objective $J(\mathsf{K}(s) + r\mathsf{D}(s))$ by implementing the local policy $K_i(s) + rD_i(s)$ and executing the subroutine `GlobalCostEst`. The subroutine `GlobalCostEst` allows the agents to form local estimates of the global objective value from observed local stage costs and communication with neighbors. Specifically, given the input controller K of `GlobalCostEst`, the quantity $\mu_i(t)$ records

---

Subroutine `SampleUSphere`**:**

Each agent i samples $V_i \in \mathbb{R}^{n_i \times m_i}$ with i.i.d. entries from $\mathscr{N}(0,1)$, and lets

$q_i(0) = \|V_i\|_F^2$.

for $t = 1, 2, \ldots, T_S$ **do**

Agent i sends $q_i(t-1)$ to its neighbors and updates

$$q_i(t) = \sum_{j=1}^{N} W_{ij} q_j(t-1). \tag{8.9}$$

return $D_i := V_i / \sqrt{N q_i(T_S)}$ *to agent* i *for all* i.

Subroutine `GlobalCostEst`$((K_i)_{i=1}^N, T_J)$**:**

Reset the system's state to $x(0) = 0$.

Each agent i implements K_i, and set $\mu_i(0) \leftarrow 0$.

for $t = 1, 2, \ldots, T_J$ **do**

Each agent i sends $\mu_i(t-1)$ to its neighbors, observes $c_i(t)$ and updates $\mu_i(t)$ by

$$\mu_i(t) = \frac{t-1}{t} \sum_{j=1}^{N} W_{ij} \mu_j(t-1) + \frac{1}{t} c_i(t). \tag{8.10}$$

return $\mu_i(T_J)$ to agent i for each $i = 1, \ldots, N$.

---

agent i's estimation of $J(\mathsf{K})$ at time step t, and is updated based on its neighbors' estimates $\mu_j(t-1)$ and its local stage cost $c_i(t)$. The updating rule (8.10) can be viewed as a combination of a consensus procedure via the communication matrix W and an online computation of the average $\frac{1}{t}\sum_{\tau=1}^{t} c_i(\tau)$. Our theoretical analysis justifies that $\mu_i(T_J) \approx J(\mathsf{K})$ for sufficiently large T_J (see Lemma C.2).

Note that the consensus (8.9) in the subroutine `SampleUSphere` can be carried out simultaneously with the consensus (8.10) in the subroutine `GlobalCostEst` as the linear system evolves, in which case $T_S = T_J$. We present the two subroutines separately for clarity.

- In *Step 3*, each agent i forms its partial gradient estimation $\hat{G}_i^r(s)$ associated with its local controller. The partial gradient estimation $\hat{G}_i^r(s)$ is based on (8.7), but uses local estimation of the global objective instead of its exact value. We also introduce a truncation step $\hat{J}_i(s) = \min\{\mu_i(T_J), \bar{J}\}$ for some sufficiently large $\bar{J}$, which guarantees the boundedness of the gradient estimator in Step 2 to help ensure the stability of our iterating policy $\mathsf{K}(s+1)$ and simplify the analysis.
- In *Step 4*, each agent i updates its local policy K_i by (8.7).

We point out that, per communication round, each agent i only shares a scalar $\mu_i(t)$ for global cost estimation in `GlobalCostEst` and a scalar $q_i(t)$ for jointly sampling in `SampleUSphere`, demonstrating the applicability in the limited-communication scenarios. Besides, each agent i only stores and updates the local policy K_i, indicating that only small storage is used even in large-scale systems.

Remark 8.1. *ZODPO conducts large enough (T_J and T_S) subroutine iterations for each policy gradient update (see Theorem 8.1 in Section 8.4). In practice, one may*

prefer fewer subroutine iterations, e.g. actor-critic algorithms. However, the design and analysis of actor-critic algorithms for our problem are non-trivial since we have to ensure stability/safety during the learning. Currently, ZODPO requires large enough subroutine iterations for good estimated gradients, so that the policy gradient updates do not drive the policy outside the stabilizing region. To overcome this challenge, we consider employing a safe policy and switching to the safe policy whenever the states are too large and resuming the learning when the states are small. In this way, we can use fewer subroutine iterations and ensure safety/stability even with poorer estimated gradients. The theoretical analysis for this method is left as future work.

8.4 Theoretical analysis

In this section, we first discuss some properties of $J(\mathsf{K})$, and then provide the nonasymptotic performance guarantees of ZODPO, followed by some discussions.

As indicated by [195, 200], the objective function $J(\mathsf{K})$ of decentralized LQ control can be nonconvex. Nevertheless, $J(\mathsf{K})$ satisfies some smoothness properties.

Lemma 8.1 (Properties of $J(\mathsf{K})$)**.** *The function $J(\mathsf{K})$ has the following properties:*

1. *$J(\mathsf{K})$ is continuously differentiable over $\mathsf{K} \in \mathscr{K}_{\rm st}$. In addition, any nonempty sublevel set $\mathscr{Q}_\alpha := \{\mathsf{K} \in \mathscr{K}_{\rm st} : J(\mathsf{K}) \le \alpha\}$ is compact.*

2. *Given a nonempty sublevel set $\mathscr{Q}_{\alpha_1}$ and an arbitrary $\alpha_2 > \alpha_1$, there exist constants $\xi > 0$ and $\phi > 0$ such that, for any $\mathsf{K} \in \mathscr{Q}_{\alpha_1}$ and K' with $\|\mathsf{K}' - \mathsf{K}\| \le \xi$, we have $\mathsf{K}' \in \mathscr{Q}_{\alpha_2}$ and $\|\nabla J(\mathsf{K}') - \nabla J(\mathsf{K})\| \le \phi \|\mathsf{K}' - \mathsf{K}\|$.*

This lemma is essentially [195, Lemma 7.3 & Corollary 3.7.1] and [192, Lemmas 1 & 2].

Without loss of generality, we let

$$\mathscr{Q}^0 = \{\mathsf{K} \in \mathscr{K}_{\rm st} : J(\mathsf{K}) \le 10 J(\mathsf{K}_0)\},$$

$$\mathscr{Q}^1 = \{\mathsf{K} \in \mathscr{K}_{\rm st} : J(\mathsf{K}) \le 20 J(\mathsf{K}_0)\}.$$

Lemma 8.1 then guarantees that there exist $\xi_0 > 0$ and $\phi_0 > 0$ such that for any $\mathsf{K} \in \mathscr{Q}^0$ and any K' with $\|\mathsf{K}' - \mathsf{K}\| \le \xi_0$, we have $\mathsf{K}' \in \mathscr{Q}^1$ and $\|\nabla J(\mathsf{K}') - \nabla J(\mathsf{K})\| \le \phi_0$. The constants ξ_0 and ϕ_0 depend on A, B, Σ_w, $J(\mathsf{K}_0)$ and Q_i, R_i for all i.

With the definitions of ξ_0, ϕ_0 above, we are ready for the performance guarantee of our ZODPO.

Theorem 8.1 (Main result)**.** *Let $\mathsf{K}_0 \in \mathscr{K}_{\rm st}$ be an arbitrary initial controller. Let $\epsilon > 0$ be sufficiently small, and suppose*

$$\begin{aligned}
r &\le \frac{\sqrt{\epsilon}}{40\phi_0}, \qquad\qquad \bar{J} \ge 50 J(\mathsf{K}_0), \\
\eta &\le \min\left\{\frac{14\xi_0 r}{15\bar{J} n_K}, \frac{3\epsilon r^2}{320\phi_0 (40 J(\mathsf{K}_0))^2 \cdot n_K^2}\right\}, \\
T_J &\ge 10^3 J(\mathsf{K}_0)\frac{n_K}{r\sqrt{\epsilon}} \max\left\{n\beta_0^2, \frac{N}{1-\rho_W}\right\}, \\
T_S &\ge \frac{\log \frac{8N^2}{\phi_0 \eta}}{-2\log \rho_W}, \qquad T_G = c \cdot \frac{40 J(\mathsf{K}_0)}{\eta\epsilon}, \quad \frac{1}{16} \le c \le 16,
\end{aligned}$$

where β_0 is a constant determined by $A, B, \Sigma_w, \mathsf{K}_0$ and Q_i, R_i for all i. Then, the following two statements hold.

1. *The controllers $\{\mathsf{K}(s)\}_{s=1}^{T_G}$ generated by Algorithm 8.1 are all stabilizing with probability at least $0.9 - 0.05c$.*

2. *The controllers $\{\mathsf{K}(s)\}_{s=1}^{T_G}$ enjoy the bound below with probability at least $0.875 - 0.05(c + c^{-1})$:*

$$\frac{1}{T_G}\sum_{s=1}^{T_G} \|\nabla J(\mathsf{K}(s))\|^2 \le \epsilon. \tag{8.11}$$

Further, if we select $\hat{\mathsf{K}}$ *uniformly randomly from* $\{\mathsf{K}(s)\}_{s=1}^{T_G}$*, then with probability at least* $0.875 - 0.05(c + c^{-1})$*,*

$$\left\|\nabla J(\hat{\mathsf{K}})\right\|^2 \leq \epsilon. \tag{8.12}$$

The proof is deferred to Section C.1. In the following, we provide some discussions regarding Theorem 8.1.

- **Probabilistic guarantees.** Theorem 8.1 establishes the stability and optimality of the controllers generated by ZODPO in a "with high probability" sense. The variable c in the probability bounds represents the value of T_G since $T_G = c \cdot 40J(\mathsf{K}_0)/(\eta\epsilon)$.

  Statement 1 suggests that as T_G increases, the probability that all the generated controllers are stabilizing will decrease. Intuitively, this is because the ZODPO can be viewed as a stochastic gradient descent, and as T_G increases, the biases and variances of the gradient estimation accumulate, resulting in a larger probability of generating destabilizing controllers.

  Statement 2 indicates that as T_G increases, the probability of enjoying the optimality guarantees (8.11) and (8.12) will first increase and then decrease. This is a result of the trade-off between a higher chance of generating destabilizing controllers and improving the policies by more policy gradient iterations as T_G increases. In other words, if T_G is too small, more iterations will improve the performance of the generated controllers; while for large T_G, the probability of generating destabilizing controllers becomes dominant.

  Finally, we mention that the probability bounds are not restrictive and can be improved by, e.g., increasing the numerical factors of T_J, using smaller stepsizes, or by the repeated learning tricks described in [192, 201], etc.

- **Output controller.** Due to the nonconvexity of $J(\mathsf{K})$, we evaluate the algorithm performance by the averaged squared norm of the gradients of $\{\mathsf{K}(s)\}_{s=1}^{T_G}$ in (8.11). Besides, we also consider an output controller that is uniformly randomly selected from $\{\mathsf{K}(s)\}_{s=1}^{T_G}$, and provide its performance guarantee (8.12). Such approaches are common in nonconvex optimization [201, 221]. Our numerical experiments suggest that selecting $\mathsf{K}(T_G)$ also yields satisfactory performance in most cases (see Section 8.5).

- **Sample complexity.** The number of samples to guarantee (8.11) with high probability is given by

$$T_G T_J = \Theta\left(\frac{n_K^3}{\epsilon^4}\max\left\{n\beta_0^2, \frac{N}{1-\rho_W}\right\}\right), \tag{8.13}$$

  where we apply the equality conditions in Theorem 8.1 and neglect the numerical constants since they are conservative and not restrictive. Some discussions are provided below.

  – The sample complexity (8.13) has an explicit polynomial dependence on the error tolerance's inverse ϵ^{-1}, the number of controller parameters n_K and the number of agents N, demonstrating the scalability of ZODPO.

  – The sample complexity depends on the maximum of the two terms: (i) term $N/(1-\rho_W)$ stems from the consensus procedure among N agents, which increases with ρ_W as a larger ρ_W indicates a smaller consensus rate; (ii) term $n\beta_0^2$ stems from approximating the infinite-horizon averaged cost, which exists even for a single agent.

  – Notice that (8.13) is proportional to n_K^3. Detailed analysis reveals that the variance of the single-point gradient estimation contributes a dependence of n_K^2, which also accords with the theoretical lower bound for zero-order optimization in [215]. The

additional n_K comes from the non-zero bias of the global cost estimation.

– While there is an explicit linear asymptotic dependence on the state vector dimension n in (8.13), we point out that the quantities $\beta_0, J(\mathsf{K}_0), \phi_0, \xi_0$ are also implicitly affected by n as they are determined by A, B, Q, R, Σ_w and K_0. Thus, the actual dependence on n is complicated and not straightforward to summarize.

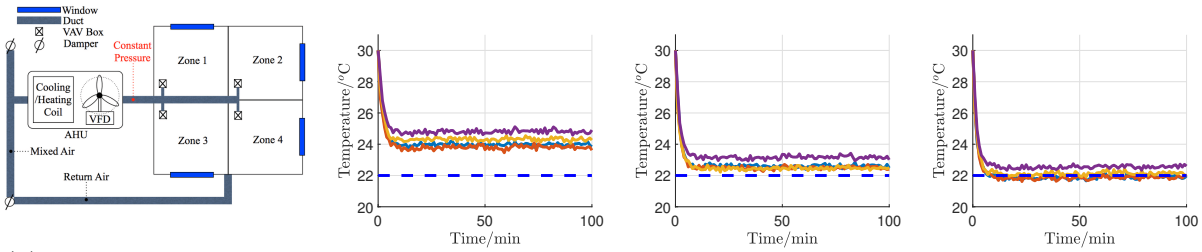
- **Optimization landscape.** Unlike centralized LQ control with full observations, reaching the global optimum is extremely challenging for general decentralized LQ control with partial observations. In some cases, the stabilizing region $\mathscr{K}_{\text{st}}$ may even contain multiple connected components [200]. However, ZODPO only explores the component containing the initial controller K_0, so K_0 affects which stationary points ZODPO converges to. How to initialize K_0 and explore other components effectively based on prior or domain knowledge remain challenging problems and are left as future.

8.5 Numerical studies

In this section, we numerically test our ZODPO on Heating Ventilation and Air Conditioning (HVAC) systems for multi-zone buildings. We consider both time-invariant cases as theoretically analyzed above and the time-varying cases for more realistic implementation.

8.5.1 Thermal dynamics model

this chapter considers multi-zone buildings with HVAC systems. Each zone is equipped with a sensor that can measure the local temperatures, and can adjust the supply air flow rate of its associated HVAC system.



(a) A 4-zone HVAC system

(b) $T_G = 50$

(c) $T_G = 150$

(d) $T_G = 250$

Figure 8.2: (a) is a diagram of the 4-zone HVAC system considered in Section 8.5.2. The figure is from [1]. (b)-(d) shows the dynamics of indoor temperatures of the 4 zones under the controllers generated by ZODPO after $T_G = 50, 150, 250$ iterations.

We adopt the linear thermal dynamics model studied in [1] with additional process noises in the discrete time setting, i.e.

$$x_i(t+1) - x_i(t) = \frac{\Delta}{\upsilon_i \zeta_i}(\theta^{\mathrm{o}}(t) - x_i(t)) + \sum_{j=1}^{N} \frac{\Delta}{\upsilon_i \zeta_{ij}}(x_j(t) - x_i(t)) + \frac{\Delta}{\upsilon_i} u_i(t) + \frac{\Delta}{\upsilon_i}\pi_i + \frac{\sqrt{\Delta}}{\upsilon_i} w_i(t), \quad 1 \le i \le N,$$

where $x_i(t)$ denotes the temperature of zone i at time t, $u_i(t)$ denotes the control input of zone i that is related with the air flow rate of the HVAC system, $\theta^{\mathrm{o}}(t)$ denotes the outdoor temperature, π_i represents a constant heat from external sources to zone i, $w_i(t)$ represents random disturbances, Δ is the time resolution, υ_i is the thermal capacitance of zone i, ζ_i represents the thermal resistance of the windows and walls between the zone i and outside environment, and ζ_{ij} represents the thermal resistance of the walls between zone i and j.

At each zone i, there is a desired temperature θ_i^* set by the users. The local cost function is composed by the deviation from the desired temperature and the control cost, i.e. $c_i(t) = \left(x_i(t) - \theta_i^*\right)^2 + \alpha_i u_i(t)^2$, where $\alpha_i > 0$ is a trade-off parameter.

8.5.2 Time-invariant cases

In this subsection, we consider a system with $N = 4$ zones (see Figure 8.2(a)) and a time-invariant outdoor temperature $\theta^{\text{o}} = 30\,^{\circ}\text{C}$. The system parameters are listed below. We set $\theta_i^* = 22\,^{\circ}\text{C}$, $\alpha_i = 0.01$, $\pi_i = 1\text{kW}$, $\Delta = 60\text{s}$, $\upsilon_i = 200\text{kJ}/^{\circ}\text{C}$, $\zeta_i = 1^{\circ}\text{C}/\text{kW}$ for all i, $\zeta_{ij} = 1^{\circ}\text{C}/\text{kW}$ if zone i and j have common walls and $\zeta_{ij} = 0^{\circ}\text{C}/\text{kW}$ otherwise. Besides, we consider i.i.d. $w_i(t)$ following $N(0, 2.5^2)$.

We consider the following decentralized control policies: $u_i(t) = K_i x_i(t) + b_i, \quad \forall 1 \leq i \leq N$, where a constant term b_i is adopted to deal with nonzero desired temperature θ_i^* and the constant drifting term in the system dynamics π_i. We apply ZODPO to learn both K_i and b_i.[4] The algorithm parameters are listed below. We consider the communication network W where $W_{ii} = 1/2$, and $W_{ij} = W_{ji} = 1/4$ if $i \neq j$ and i are j share common walls. We set $r = 0.5, \eta = 0.0001, \bar{J} = 10^6$. Since the thermal dynamical system is open-loop stable, we select the initial controller as zero, i.e. $\mathsf{K}_i = 0, b_i = 0$ for all i.

Figures 8.2(b)–(d) plot the temperature dynamics of the four zones by implementing the controllers generated by ZODPO at policy gradient iterations $T_G = 50, 150, 250$ respectively with $T_J = 300$. It can be observed that with more iterations, the controllers generated by ZODPO stabilize the system faster and steer the room temperature closer to the desired temperature.

[4]This requires a straightforward modification of Algorithm 8.1: in Step 1, add perturbations onto both K_i and b_i, in Step 2, estimate the partial gradients with respect to K_i and b_i, in Step 3, update K_i and b_i by the estimated partial gradient.

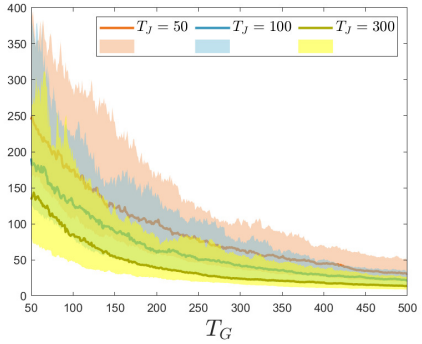


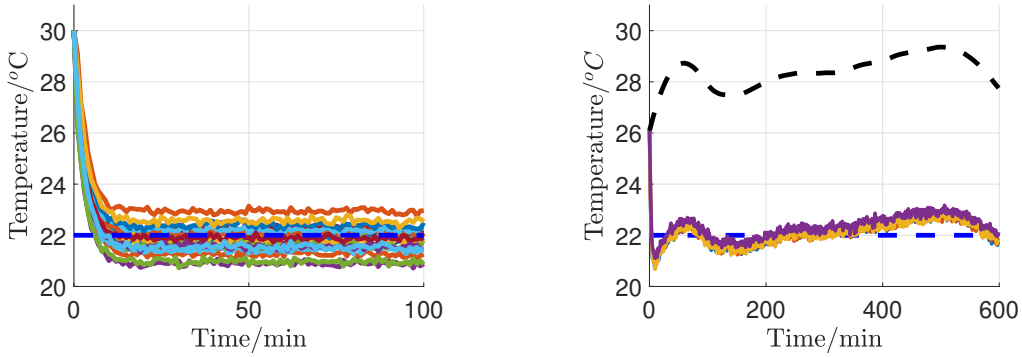
Figure 8.3: A comparison of ZODPO with $T_J = 50, 150, 300$. The solid lines represent the mean values and the shade represents 70% confidence intervals of the actual costs by implementing the controllers generated by ZODPO.

Figure 8.3 plots the infinite-horizon averaged costs of controllers generated by ZODPO for different $50 \leq T_G \leq 500$ when $T_J = 50, 100, 300$ by 500 repeated simulations. As T_G increases, the averaged costs keep decreasing, which is consistent with Figures 8.2(b)–(d). Since $T_G \leq 300$ is not extremely large, we do not observe the increase of the probabilities of generating unstable controllers. Notice that with a larger T_J, the confidence intervals shrink and the averaged costs decrease, indicating less fluctuations and better performance. This is intuitive since a larger T_J indicates a better gradient estimation.

8.5.3 Larger scale systems

Here, we consider an $N = 20$ system to demonstrate that our ZODPO can handle systems with higher dimensions. We consider a 2-floor building with 5×2 rooms on each floor. Other system parameters are provided in Section 8.5.2. Figure 8.4(a) plots the dynamics of the indoor temperatures of 20 rooms when implementing a controller generated by ZODPO after $T_G = 4800$ iterations with $T_J = 500$. Notice that all the room

temperatures stabilize around the desired temperature 22°C, indicating that our ZODPO can output an effective controller for reasonably large T_G and T_J even for a larger-scale system.



(a) An $N = 20$ system

(b) Varying outdoor temperature

Figure 8.4: (a) plots the dynamics of indoor temperatures of an $N = 20$ system with a constant outdoor temperature 30°C. (b) plots the dynamics of indoor temperatures of the 4-zone system with time-varying outdoor temperature, with the black line representing the outdoor temperature.

8.5.4 Varying outdoor temperature

This subsection considers a more realistic scenario where the outdoor temperature is changing with time. The data is collected by Harvard HouseZero Program.[5] To adapt to varying outdoor temperature, we consider the following form of controllers:

$$u_i(t) = K_i x_i(t) + K_i^o \theta^o(t) + b_i,$$

and apply our ZODPO to learn K_i, K_i^o, b_i. During the learning process, we consider different outdoor temperatures in different policy gradient iterations, but fix the outdoor temperature within one iteration for better training. We consider the system in Section

[5] http://harvardcgbc.org/research/housezero/

8.5.2 and set $T_J = 300$, $\eta = 2 \times 10^{-6}, r = 0.5, \bar{J} = 10^8$. Figure 8.4(b) plots the temperature dynamics by implementing the controller generated by ZODPO at policy iteration $T_G = 250$. The figure shows that even with varying outdoor temperatures, ZODPO is still able to find a controller that roughly maintains the room temperature at the desired level.

8.6 Conclusions and future work

this chapter considers distributed learning of decentralized linear quadratic control systems with limited communication, partial observability, and local costs. We propose a ZODPO algorithm that allows agents to learn decentralized controllers in a distributed fashion by leveraging the ideas of policy gradient, consensus algorithms and zero-order optimization. We prove the stability of the output controllers with high probability. We also provide sample complexity guarantees. Finally, we numerically test ZODPO on HVAC systems.

There are various directions for future work. For example, effective initialization and exploration of other stabilizing components to approach the global optima are important topics. It is also worth exploring the optimization landscape of decentralized LQR systems with additional structures. Besides, we are also interested in employing other gradient estimators, e.g. two-point zero-order gradient estimator, to improve sample complexity. Utilizing more general controllers that take advantage of history information is also an interesting direction. Finally, it is worth designing actor-critic-type algorithms to reduce the running time per iteration of policy gradient.

Part IV

Appendix

Appendix A | Appendix to Part I

A.1 Proofs for Chapter 2

A.1.1 Proof of Theorem 2.4

The main proof idea is to construct random cost functions and show that the lower bound (2.9) holds in expectation, and thus there must exist some case with positive probability such that the lower bound holds. Without loss of generality, we consider a 1-dimensional case $\mathbb{X} = [-\frac{D}{2}, \frac{D}{2}]$ and let $x_0 = 0$.[1] For technical reasons, we consider three cases: (i) $2D \le L_T \le DT$, (ii) $D < L_T < 2D$, (iii) $0 \le L_T \le D$, and construct slightly different cost function sequences for each case.

A.1.1.1 When $2D \le L_T \le DT$

Part 1: construct random $\{f_t(\cdot)\}_{t=1}^T$: For any $\alpha > 0$, $\beta > 0$ ($\beta = 0$ is trivial), we construct quadratic function $f_t(x_t) = \frac{\alpha}{2}(x_t - \xi_t)^2$ with parameter $\xi_t \in \mathbb{X}$. Thus, $\xi_t = \arg\min_{\mathbb{X}} f_t(x_t)$. We construct random $\{\xi_t\}_{t=1}^T$ below. Let $\Delta = \lceil T/\lfloor L_T/D \rfloor \rceil$ and divide T stages into $K = \lceil \frac{T}{\Delta} \rceil$ parts with Δ stages per part. Notice that $1 \le \Delta \le T$ and $1 \le K \le T$. For $0 \le k \le K-1$, generate $\xi_{k\Delta+1}$ independently with distribution

[1] Our proof can be extended to any $\mathbb{X} \subseteq \mathbb{R}^n$ by letting $f_t(x_t) = \alpha/2\|x_t - \xi_t - \frac{v_1+v_2}{2}\|^2$, where $v_1, v_2 \in \mathbb{X}$ and $\|v_1 - v_2\| = D$, and by generating ξ_t randomly with probability $\mathbb{P}(\xi = v_1) = \mathbb{P}(\xi = v_2) = 1/2$.

$\mathbb{P}(\xi = \frac{D}{2}) = \mathbb{P}(\xi = -\frac{D}{2}) = \frac{1}{2}$, and let $\xi_t = \xi_{k\Delta+1}$ for $k\Delta + 2 \le t \le (k+1)\Delta$. It is straightforward that $\{\xi_t\}_{t=1}^T$ satisfy the path length budget.

Lemma A.1. *For $f_t(x_t)$ constructed above, the path length of $\{f_t\}_{t=1}^T$ satisfies $\sum_{t=1}^T \|\xi_t - \xi_{t-1}\| \le L_T$, where $\xi_0 = x_0 = 0$.*

Part 2: characterize $\boldsymbol{x}^*$. The problem (2.1) constructed in Part 1 enjoys a closed-form optimal solution: $\boldsymbol{x}^* = \boldsymbol{\xi}\boldsymbol{A}^\top$, where $\boldsymbol{x}^* = (x_1^*, \dots, x_T^*)$ denotes the optimal solution as a row vector in $\mathbb{R}^T$, $\boldsymbol{\xi} = (\xi_1, \dots, \xi_T)$, $\boldsymbol{A} = (a_{i,j})_{i,j=1}^T$. Equivalently, we have $x_t^* = \sum_{\tau=1-t}^{T-t} a_{t,t+\tau}\xi_{t+\tau}$, and $a_{t,t+\tau}$ represents the influence of $\xi_{t,t+\tau}$ on x_t^*. Lemma A.2 will show that the influence $a_{t,t+\tau}$ decays exponentially for $\tau \ge 0$.

Lemma A.2. *Consider the cost in Part 1. The optimal solution to (2.1) is $\boldsymbol{x}^* = \boldsymbol{\xi}\boldsymbol{A}^\top$, where $a_{t,t+\tau} \ge \frac{\alpha}{\alpha+\beta}(1-\rho)\rho^\tau$ for $\tau \ge 0$.*

The proof is mostly based on an explicit formula of $\boldsymbol{A}$ and the technical details are deferred to [108].

Part 3: characterize $\boldsymbol{x}^{\mathscr{A}}$. The key observation here is that the output $x_t^{\mathscr{A}}$ of any online algorithm $\mathscr{A}$ is a random variable determined by $\{\xi_s\}_{s=1}^{t+W-1}$, since $x_t^{\mathscr{A}}$ is decided by $\mathscr{A}$ based on $\{f_s(\cdot)\}_{s=1}^{t+W-1}$, which is determined by $\{\xi_s\}_{s=1}^{t+W-1}$.[2]

Part 4: lower bound $\mathbb{E}[\text{Regret}^d(\mathscr{A})]$. To prove the lower bound, we define a set of stages below:

$$\mathbb{J} := \{1 \le t \le T - W \mid t + W \equiv 1 \pmod{\Delta}\}.$$

[2]Rigorously speaking, to ensure the random variable to be well-defined, some measurability assumption on $\mathscr{A}$ should be imposed, which is satisfied by most algorithms in practice and is thus omitted here for simplicity.

Before the lower bound's proof, we prove two helping lemmas.

Lemma A.3. *Consider the cost in Part 1, for any online algorithm $\mathscr{A}$, we have $\mathbb{E}\,|x_t^{\mathscr{A}} - x_t^*|^2 \geq \frac{a_{t,t+W}^2 D^2}{4}$ for any $t \in \mathbb{J}$, where $a_{t,t+W}$ is an entry of matrix $\boldsymbol{A}$ defined in Lemma A.2.*

Proof. We denote the σ-algebra generated by $\xi_1, \ldots, \xi_t$ as $\mathscr{F}_t$. By our discussion in Part 2, x_t^* is $\mathscr{F}_T$-measurable. In addition, by Part 3, for any online algorithm $\mathscr{A}$, the output $x_t^{\mathscr{A}}$ is $\mathscr{F}_{t+W-1}$-measurable. It is a classical result that for any σ-algebra $\mathscr{F}$ of the probability space, the conditional expectation $\mathbb{E}[X \mid \mathscr{F}]$ minimizes the mean square error $\mathbb{E}(Y - X)^2$ among any random variable Y that is $\mathscr{F}$-measurable. Thus, we have $\mathbb{E}(x_t^{\mathscr{A}} - x_t^*)^2 \geq \mathbb{E}\left(\mathbb{E}[x_t^* \mid \mathscr{F}_{t+W-1}] - x_t^*\right)^2$. Consequently, we do not have to discuss each online algorithm $\mathscr{A}$ but only need to bound $\mathbb{E}(\mathbb{E}[x_t^* \mid \mathscr{F}_{t+W-1}] - x_t^*)^2$ below. For $t \in \mathbb{J}$, $\mathbb{E}(\mathbb{E}[x_t^* \mid \mathscr{F}_{t+W-1}] - x_t^*)^2 = \mathbb{E}[\sum_{\tau=1}^{t+W-1} a_{t,\tau}\xi_\tau - \sum_{\tau=1}^{T} a_{t,\tau}\xi_\tau]^2 = \mathbb{E}(\sum_{\tau=t+W}^{T} a_{t,\tau}\xi_\tau)^2 = \mathbb{E}[(\sum_{i=t+W}^{t+W+\Delta-1} a_{t,i})\xi_{t+W}]^2 + \mathbb{E}[\sum_{\tau=t+W+\Delta}^{T} \xi_\tau]^2 \geq \mathbb{E}[a_{t,t+W}^2 \xi_{t+W}^2] = a_{t,t+W}^2 \frac{D^2}{4}$, where the first equality uses $x_t^* = \sum_{\tau=1}^{T} a_{t,\tau}\xi_\tau$ as discussed in Part 2 and $\mathbb{E}[\xi_\tau \mid \mathscr{F}_{t+W-1}] = 0$ for $\tau \geq t+W$ by our definition of $\{\xi_t\}_{t=1}^T$ in Part 1, the third equality is by $\xi_{t+W} = \cdots = \xi_{t+W+\Delta-1}$ for $t \in \mathbb{J}$ and ξ_{t+W} uncorrelated with ξ_τ for $\tau \geq t + W + \Delta$, the first inequality uses $a_{t,\tau} > 0$ by Lemma A.2, and the last equality is because $\mathbb{E}[\xi_{t+W}^2] = D^2/4$ by our definition of ξ_{t+W} in Part 1. □

Lemma A.4. *When $T \geq 2W$ and $L_T \geq 2D$, we have $|\mathbb{J}| \geq \frac{L_T}{12D}$.*

The proof is very technical and thus deferred to [108].

Next, we bound the expected regret. $\mathbb{E}[\mathrm{Regret}^d(\mathscr{A})] = \mathbb{E}[\mathsf{C}_\mathsf{T}(\boldsymbol{x}^{\mathscr{A}}) - \mathsf{C}_\mathsf{T}(\boldsymbol{x}^*)] \geq \frac{\alpha}{2}\,\mathbb{E}\,\|\boldsymbol{x}^{\mathscr{A}} - \boldsymbol{x}^*\|^2 = \frac{\alpha}{2}\sum_{t=1}^{T}\mathbb{E}\,|x_t^{\mathscr{A}} - x_t^*|^2 \geq \frac{\alpha}{2}\sum_{t\in\mathbb{J}}\mathbb{E}\,|x_t^{\mathscr{A}} - x_t^*|^2 \geq \sum_{t\in\mathbb{J}}\frac{a_{t,t+W}^2 D^2 \alpha}{8} \geq$

$\frac{\alpha D}{96}(1-\rho)^2(\frac{\alpha}{\alpha+\beta})^2 L_T \rho^{2W}$, where the first inequality uses Lemma 2.1, the third one uses Lemma A.3, the last one uses Lemma A.2 and Lemma A.4. Therefore, there exists some realization of $\boldsymbol{\xi}$ generating the regret lower bound, which completes the proof for the case $2D \leq L_T \leq DT$.

A.1.1.2 $D < L_T < 2D$

The proof is very similar to the proof above. We also consider cost function $f_t(x_t) = \frac{\alpha}{2}(x_t - \xi_t)^2$, but we define $\{\xi_t\}_{t=1}^T$ in a different way, i.e. let $\xi_1 = \cdots = \xi_W = 0$, and $\xi_{W+1} = \cdots = \xi_T$ be a random variable following distribution $\mathbb{P}(\xi = \frac{D}{2}) = \mathbb{P}(\xi = -\frac{D}{2}) = \frac{1}{2}$. It is easy to verify that the path length budget L_T is satisfied. Since the matrix $\boldsymbol{A}$ does not depend on $\{\xi_t\}_{t=1}^T$, Lemma A.2 still holds. Besides, similar to Lemma A.3, we have $\mathbb{E}\,|x_1^{\mathscr{A}} - x_1^*|^2 \geq \frac{a_{1,1+W}^2 D^2}{4}$. Consequently, we can lower bound the expected regret below: $\mathbb{E}[\text{Regret}^d(\mathscr{A})] = \mathbb{E}[\mathsf{C}_\mathsf{T}(\boldsymbol{x}^{\mathscr{A}}) - \mathsf{C}_\mathsf{T}(\boldsymbol{x}^*)] \geq \frac{\alpha}{2}\mathbb{E}\,|\boldsymbol{x}^{\mathscr{A}} - \boldsymbol{x}^*|^2 \geq \frac{\alpha}{2}\mathbb{E}\,|x_1^{\mathscr{A}} - x_1^*|^2 \geq \frac{\alpha a_{1,1+W}^2 D^2}{8} \geq \frac{\alpha D^2}{8}\rho^{2W}(1-\rho)^2(\frac{\alpha}{\alpha+\beta})^2 \geq \frac{\alpha D L_T}{96}\rho^{2W}(\frac{\alpha(1-\rho)}{\alpha+\beta})^2$.

A.1.1.3 When $0 \leq L_T \leq D$

The proof is very similar. We consider the function $f_t(x_t) = \frac{\alpha}{2}(x_t - \xi_t)^2$ with different ξ_t: $\xi_1 = \cdots = \xi_W = 0$, and $\xi_{W+1} = \cdots = \xi_T$ is a random variable from $\mathbb{P}(\xi_t = \frac{L_T}{2}) = \mathbb{P}(\xi_t = -\frac{L_T}{2}) = \frac{1}{2}$. Similarly, we have the lower bound: $\mathbb{E}[\text{Regret}^d(\mathscr{A})] = \mathbb{E}[\mathsf{C}_\mathsf{T}(\boldsymbol{x}^{\mathscr{A}}) - \mathsf{C}_\mathsf{T}(\boldsymbol{x}^*)] \geq \frac{\alpha}{2}\mathbb{E}\,|\boldsymbol{x}^{\mathscr{A}} - \boldsymbol{x}^*|^2 \geq \frac{\alpha}{2}\mathbb{E}\,|x_1^{\mathscr{A}} - x_1^*|^2 \geq \frac{\alpha a_{1,1+W}^2 L_T^2}{8} \geq \frac{\alpha L_T^2}{8}\rho^{2W}(1-\rho)^2\left(\frac{\alpha}{\alpha+\beta}\right)^2 \geq \frac{\alpha L_T^2}{96}\rho^{2W}\left(\frac{\alpha(1-\rho)}{\alpha+\beta}\right)^2$.

A.1.2 Proof of Theorem 2.3

The proof is similar to Appendix A.1.1. We also construct random cost functions and prove the lower bound on the expected regret. We discuss two scenarios: $0 < L_T < D$, and $D \le L_T \le DT$ ($L_T = 0$ is trivially true), and construct different functions. We let $x_0 = 0$ without loss of generality.

A.1.2.1 When $0 < L_T < D$

Construct random costs. For each $0 < L_T < D$, we consider $x = (x^{(1)}, x^{(2)}) \in \mathbb{R}^2$ and define $\mathbb{X} = [-\frac{L_T}{2}, \frac{L_T}{2}] \times [-\frac{\sqrt{D^2-L_T^2}}{2}, \frac{\sqrt{D^2-L_T^2}}{2}]$ with diameter D. For any $\alpha > 0$, consider cost function $f_t(x_t) = \frac{\alpha}{2}\|x_t - \xi_t\|^2$ with parameter $\xi_t = (\xi_t^{(1)}, \xi_t^{(2)}) \in [-M, M] \times [-\frac{D}{2}, \frac{D}{2}]$ for $M = D + (1 + \beta/\alpha)\frac{L_T}{2}$. It can be verified that the gradient bound satisfies $G \le \alpha\sqrt{(M + D/2)^2 + D^2} \le (3\alpha + \beta)D$.

Next, we consider the following two possible function sequences and each sequence happens with probability 0.5:

(i) $\xi_1 = (M, 0)$ and $\xi_t = (\frac{L_T}{2}, 0)$ for $t \ge 2$; and

(ii) $\xi_1 = (-M, 0)$ and $\xi_t = (-\frac{L_T}{2}, 0)$ for $t \ge 2$.

By the first-order condition for the constrained convex optimization [100] and by our choice of M above, it can be verified that the optimal offline solution $\boldsymbol{x}^* = \arg\min_{\mathbb{X}\times\cdots\times\mathbb{X}} \mathsf{C_T}(\boldsymbol{x})$ for the two sequences satisfies: sequence (i): $x_t^* = (\frac{L_T}{2}, 0)$ for $t \ge 1$; and sequence (ii): $x_t^* = (-\frac{L_T}{2}, 0)$ for $t \ge 1$.

Prove the regret bound. By the convexity of $\mathsf{C_T}(\boldsymbol{x})$, we have $\mathbb{E}[\mathsf{C_T}(\boldsymbol{x}^{\not\alpha}) - \mathsf{C_T}(\boldsymbol{x}^*)] \ge$

$\mathbb{E}\left[\sum_{t=1}^{T}\langle\frac{\partial \mathsf{C}_\mathsf{T}}{\partial x_t}(\boldsymbol{x}^*), x_t^{\mathscr{A}} - x_t^*\rangle\right] = \mathbb{E}\left[\langle\frac{\partial \mathsf{C}_\mathsf{T}}{\partial x_1}(\boldsymbol{x}^*), x_1^{\mathscr{A}} - x_1^*\rangle\right]$, where the equality is by $\frac{\partial \mathsf{C}_\mathsf{T}}{\partial x_t}(\boldsymbol{x}^*) = 0$ for $t \geq 2$. Further, for the sequence (i) we have $\frac{\partial \mathsf{C}_\mathsf{T}}{\partial x_1}(\boldsymbol{x}^*) = (\alpha(\frac{L_T}{2} - M) + \beta\frac{L_T}{2}, 0)$, and for the sequence (ii) we have $\frac{\partial \mathsf{C}_\mathsf{T}}{\partial x_1}(\boldsymbol{x}^*) = (\alpha(-\frac{L_T}{2} + M) - \beta\frac{L_T}{2}, 0)$. Therefore, $\mathbb{E}\left[\frac{\partial \mathsf{C}_\mathsf{T}}{\partial x_1}(\boldsymbol{x}^*)(x_1^{\mathscr{A}} - x_1^*)\right] = 0.5(\alpha(\frac{L_T}{2} - M) + \beta\frac{L_T}{2})(x_1^{\mathscr{A},(1)} - \frac{L_T}{2}) + 0.5(\alpha(-\frac{L_T}{2} + M) - \beta\frac{L_T}{2})(x_1^{\mathscr{A},(1)} + \frac{L_T}{2}) = (\alpha(-\frac{L_T}{2} + M) - \beta\frac{L_T}{2})\frac{L_T}{2} \geq \alpha D\frac{L_T}{2} \geq \frac{\alpha D L_T}{32}(1-\rho)^2\left(\frac{\alpha}{\alpha+\beta}\right)^2$, which completes the proof for this scenario.

A.1.2.2 $D \leq L_T \leq DT$

The proof is the same as Appendix A.1.1.1. When $W = 0$ we can bound $|J|$ without assuming $L_T \geq 2D$.

Lemma A.5. *When $T \geq 1$, and $D \leq L_T \leq DT$, $|\mathbb{J}| \geq \frac{L_T}{4D}$.*

Proof. By $\Delta = \lceil T/\lfloor L_T/D\rfloor\rceil \leq T/\lfloor L_T/D\rfloor + 1$, when $L_T \geq D$ and $T \geq 1$, we have $|\mathbb{J}| = \lceil\frac{T}{\Delta}\rceil \geq \frac{T}{\Delta} \geq \frac{T}{T/\lfloor L_T/D\rfloor+1} = \lfloor L_T/D\rfloor\frac{T}{T+\lfloor L_T/D\rfloor} \geq \frac{L_T}{2D}\frac{T}{T+T} = \frac{L_T}{4D}$ by $\lfloor x\rfloor \geq x/2$ when $x \geq 1$, and $L_T \leq DT$. □

Then, the lower bound on the expected regret can be proved in the same way as Theorem 2.4: $\mathbb{E}[\mathsf{C}_\mathsf{T}(\boldsymbol{x}^{\mathscr{A}}) - \mathsf{C}_\mathsf{T}(\boldsymbol{x}^*)] \geq \mathbb{E}\frac{\alpha}{2}\|\boldsymbol{x}^{\mathscr{A}} - \boldsymbol{x}^*\|^2 \geq \frac{\alpha}{2}\sum_{t\in\mathbb{J}}\frac{a_{t,t}^2 D^2}{4} \geq \frac{\alpha D L_T}{32}(1-\rho)^2\left(\frac{\alpha}{\alpha+\beta}\right)^2$.

A.1.3 Proof of Theorem 2.2

It suffices to prove OGD's regret bound. For simplicity, we denote OGD's output as x_t instead of $x_t(0)$ below.

Lemma A.6. *When $\gamma = 1/l$, OGD's outputs satisfy $\sum_{t=1}^{T} \|x_t - \xi_t\| \leq \frac{1}{1-\kappa} \sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\|$ and $\sum_{t=1}^{T} \|x_t - x_{t-1}\|^2 \leq \frac{2G}{l(1-\kappa)} \sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\|$, where $x_1 = \xi_0 = x_0$, $\kappa = \sqrt{1 - \frac{\alpha}{l}}$.*

Proof. Firstly, we have $\sum_{t=1}^{T} \|x_t - \xi_t\| \leq \sum_{t=1}^{T} \|x_t - \xi_{t-1}\| + \sum_{t=1}^{T} \|\xi_{t-1} - \xi_t\| = \sum_{t=2}^{T} \|x_t - \xi_{t-1}\| + \sum_{t=1}^{T} \|\xi_{t-1} - \xi_t\| \leq \sum_{t=2}^{T} \kappa \|x_{t-1} - \xi_{t-1}\| + \sum_{t=1}^{T} \|\xi_{t-1} - \xi_t\| \leq \sum_{t=1}^{T} \kappa \|x_t - \xi_t\| + \sum_{t=1}^{T} \|\xi_{t-1} - \xi_t\|$, where we used $x_1 = \xi_0$, OGD's updating rule and Theorem 2.2.8 in [100].[3] Then, we prove the first bound by subtracting $\sum_{t=1}^{T} \kappa \|x_t - \xi_t\|$ from the both sides of the inequality above and dividing the both sides by $1 - \kappa$.

Next, we bound $\sum_{t=1}^{T} \|x_t - x_{t-1}\|^2$ by the following. $\sum_{t=1}^{T} \|x_t - x_{t-1}\|^2 = \sum_{t=2}^{T} \|x_t - x_{t-1}\|^2 = \sum_{t=1}^{T-1} \|x_{t+1} - x_t\|^2 \leq \frac{2}{l} \sum_{t=1}^{T-1} [f_t(x_t) - f_t(x_{t+1})] \leq \frac{2}{l} \sum_{t=1}^{T-1} [f_t(x_t) - f_t(\xi_t)] \leq \frac{2}{l} \sum_{t=1}^{T-1} G \|x_t - \xi_t\| \leq \frac{2G}{l(1-\kappa)} \sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\|$, where the first line uses $x_1 = x_0$, the first inequality uses Corollary 2.2.1 (2.2.16) in [100],[4] the second inequality is by $\xi_t = \arg\min_{\mathbb{X}} f_t(x)$, the third inequality is by $f_t(x_t) - f_t(\xi_t) \leq \langle \nabla f_t(x_t), x_t - \xi_t \rangle \leq G \|x_t - \xi_t\|$, the last one uses the first bound in Lemma A.6. □

Next, we prove OGD's regret bound. $\text{Regret}^d(OGD) \leq \sum_{t=1}^{T} (f_t(x_t) - f_t(x_t^*) + \frac{\beta}{2} \|x_t - x_{t-1}\|^2) \leq \sum_{t=1}^{T} (f_t(x_t) - f_t(\xi_t) + \frac{\beta}{2} \|x_t - x_{t-1}\|^2) \leq \sum_{t=1}^{T} [G \|x_t - \xi_t\| + \frac{\beta}{2} \|x_t - x_{t-1}\|^2] \leq \delta \sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\|$, where we used $\|x_t^* - x_{t-1}^*\|^2 \geq 0$; $\xi_t = \arg\min_{\mathbb{X}} f_t(x)$ and thus $f_t(x_t^*) \geq f_t(\xi_t)$; the third inequality is by $f_t(x_t) - f_t(\xi_t) \leq \nabla f_t(x_t)^\top (x_t - \xi_t) \leq G \|x_t - \xi_t\|$, and the last inequality is by Lemma A.6.

[3] By OGD's rule (2.5), x_t is one gradient iteration on f_{t-1} from initial value x_{t-1}. f_{t-1} is α strongly convex and l smooth with optimal solution ξ_{t-1}.

[4] This is by substituting $\bar{x} = x_t$, $f = f_t$, $x_Q(x_t, l) = x_{t+1}$, $g_Q(x_t, l) = l(x_t - x_{t+1})$ into (2.2.16) in Corollary 2.2.1 in [100].

A.1.4 Proof of Corollary 2.1

The equivalence between the optimal control and (2.1) is straightforward. ξ_t is the optimal steady state at t because $(x = \xi_t, u = 0)$ solves $\min_{x=x-u\in\mathbb{X}} f_t(x) + \beta/2\|u\|^2$. We denote $\xi \coloneqq \sum_{t=1}^{+\infty} \|\xi_t - \xi_{t-1}\|$ and consider RHGD below. For any finite T, $\sum_{t=1}^{T-W} \frac{\alpha}{2}\|x_t(W) - \xi_t\|^2 \leq \sum_{t=1}^{T} [f_t(x_t(W)) - f_t(\xi_t)] \leq \text{Regret}^d(RHGD) + \frac{\beta}{2}\sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\|^2 \leq \delta Q_f \left[1 - \frac{1}{Q_f}\right]^W \xi + \frac{\beta}{2}\xi^2$, where the first term only considers the summation to $T - W$ because the outputs $x_t(W)$ for $t > T - W$ depend on whether RHGD terminates at T or not; the first inequality is by the α-strong convexity of $f_t(\cdot)$ and the optimality of ξ_t; the second one is by (2.1), $\|x_t(W) - x_{t-1}(W)\|^2 \geq 0$, $\mathsf{C_T}(\xi_1, \ldots, \xi_T) \geq \mathsf{C_T}(\boldsymbol{x}^*)$, and (2.2); the last inequality uses Theorem 2.2, $\sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\| \leq \xi$, and $\sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\|^2 \leq (\sum_{t=1}^{T} \|\xi_t - \xi_{t-1}\|)^2 \leq \xi^2$. By letting $T \to +\infty$, we have $\sum_{t=1}^{\infty} \frac{\alpha}{2}\|x_t(W) - \xi_t\|^2 \leq \delta Q_f(1 - 1/Q_f)^W \xi + \frac{\beta}{2}\xi^2 < +\infty$, thus $\|x_t(W) - \xi_t\| \to 0$ as $t \to +\infty$. Since $\xi_t \to \xi_\infty$, we have $\|x_t(W) - \xi_\infty\| \to 0$. The proof for RHAG is the same. □

A.1.5 Proof of Theorem 2.5

Remember that RHIG can be interpreted as inexact projected gradient descent on $C(\boldsymbol{x}; \boldsymbol{\theta})$:

$$\boldsymbol{x}(k+1) = \Pi_{\mathbb{X}^T} \left[\boldsymbol{x}(k) - \eta \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k); \boldsymbol{\theta} - \boldsymbol{\delta}(W-k))\right] \tag{A.1}$$

where the exact gradient should be $\nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k); \boldsymbol{\theta})$ but the parameter prediction error $\boldsymbol{\delta}(W-k)$ results in inexact gradient $\nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k); \boldsymbol{\theta} - \boldsymbol{\delta}(W-k))$. Notice that when $W - k > T$, by our definition, we have $\boldsymbol{\delta}(W-k) = \boldsymbol{\delta}(T)$. Consequently, the regret bound of RHIG can be proved based on the convergence analysis of the projected gradient

descent with inexact gradients. We note that unlike the classic inexact gradient where the gradient errors are uniformly bounded, RHIG's inexact gradients (A.1) have different gradient errors at different iterations, thus calling for slightly different convergence analysis. In the following, we first provide some supportive lemmas, then provide a rigorous proof of Theorem 2.5.

Useful lemmas. Firstly, we provide a bound on the gradient errors with respect to the errors on the parameters.

Lemma A.7 (Gradient prediction error bound)**.** *For any true parameter $\boldsymbol{\theta} \in \Theta^T$ and the predicted parameter $\boldsymbol{\theta}' \in \Theta^T$, the error of the predicted gradient can be bounded below.*

$$\|\nabla_{\boldsymbol{x}} C(\boldsymbol{x}; \boldsymbol{\theta}') - \nabla_{\boldsymbol{x}} C(\boldsymbol{x}; \boldsymbol{\theta})\|^2 \leq h^2 \|\boldsymbol{\theta}' - \boldsymbol{\theta}\|^2, \quad \forall\ \boldsymbol{x} \in \Theta^T$$

Proof. Firstly, we consider the gradient with respect to each stage variable x_t, which is provided by $\nabla_{x_t} C(\boldsymbol{x}; \boldsymbol{\theta}) = \nabla_{x_t} f(x_t; \theta_t) + \nabla_{x_t} d(x_t, x_{t-1}) + \nabla_{x_t} d(x_{t+1}, x_t) \mathbf{1}_{(t \leq T-1)}$. Noticing that $d(x_t, x_{t-1})$ does not depend on the parameter $\boldsymbol{\theta}$, we obtain the prediction error bound of gradient with respect to x_t as follows. $\|\nabla_{x_t} C(\boldsymbol{x}; \boldsymbol{\theta}') - \nabla_{x_t} C(\boldsymbol{x}; \boldsymbol{\theta})\| = \|\nabla_{x_t} f(x_t; \theta_t') - \nabla_{x_t} f(x_t; \theta_t)\| \leq h \|\theta_t' - \theta_t\|$ Therefore, the prediction error of the full gradient can be bounded as follows, $\|\nabla_{\boldsymbol{x}} C(\boldsymbol{x}; \boldsymbol{\theta}') - \nabla_{\boldsymbol{x}} C(\boldsymbol{x}; \boldsymbol{\theta})\|^2 = \sum_{t=1}^T \|\nabla_{x_t} C(\boldsymbol{x}; \boldsymbol{\theta}') - \nabla_{x_t} C(\boldsymbol{x}; \boldsymbol{\theta})\|^2 \leq h^2 \sum_{t=1}^T \|\theta_t' - \theta_t\|^2 = h^2 \|\boldsymbol{\theta}' - \boldsymbol{\theta}\|^2$, which completes the proof. □

Next, we provide an equivalent characterization of the projected gradient update with respect to inexact parameters.

Lemma A.8 (A representation of inexact projected gradient updates)**.** *For any predicted parameter $\boldsymbol{\theta}'$ and any stepsize η, the projected gradient descent with predicted parameter*

$\boldsymbol{x}(k+1) = \Pi_{\mathbb{X}^T}\left[\boldsymbol{x}(k) - \eta \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k); \boldsymbol{\theta}')\right]$ *is equivalent to the following representation.*

$$\boldsymbol{x}(k+1) = \arg\min_{\boldsymbol{x} \in \mathbb{X}^T} \left\{ \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k); \boldsymbol{\theta}'), \boldsymbol{x} - \boldsymbol{x}(k) \rangle + \frac{1}{2\eta} \|\boldsymbol{x} - \boldsymbol{x}(k)\|^2 \right\}$$

Proof. By the definition of projection, the projected gradient descent with predicted parameter is equivalent to the following.

$$\begin{aligned}
\boldsymbol{x}(k+1) &= \arg\min_{\boldsymbol{x} \in \mathbb{X}^T} \left\{ \|\boldsymbol{x} - \boldsymbol{x}(k) + \eta \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k); \boldsymbol{\theta}')\|^2 \right\} \\
&= \arg\min_{\boldsymbol{x} \in \mathbb{X}^T} \left\{ \|\boldsymbol{x} - \boldsymbol{x}(k)\|^2 + \eta^2 \|\nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k); \boldsymbol{\theta}')\|^2 + 2\eta \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k); \boldsymbol{\theta}'), \boldsymbol{x} - \boldsymbol{x}(k) \rangle \right\} \\
&= \arg\min_{\boldsymbol{x} \in \mathbb{X}^T} \left\{ \frac{1}{2\eta} \|\boldsymbol{x} - \boldsymbol{x}(k)\|^2 + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k); \boldsymbol{\theta}'), \boldsymbol{x} - \boldsymbol{x}(k) \rangle \right\}
\end{aligned}$$

where the last equality uses the fact that $\eta^2 \|\nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k); \boldsymbol{\theta}')\|^2$ does not depend on $\boldsymbol{x}$. □

Lastly, we provide a strong-convexity-type inequality and a smoothness-type inequality under inexact gradients. Both inequalities suffer from additional error terms caused by the parameter prediction error.

Lemma A.9 (Strong convexity inequality with errors)**.** *Consider optimization* $\min_{\boldsymbol{x} \in \mathbb{X}^T} C(\boldsymbol{x}; \boldsymbol{\theta})$*. For any* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{X}^T$*, for any inexact parameter* $\boldsymbol{\theta}'$ *and the resulting inexact gradient* $\nabla_{\boldsymbol{x}} C(\boldsymbol{x}; \boldsymbol{\theta}')$*, we have*

$$C(\boldsymbol{y}; \boldsymbol{\theta}) \geq C(\boldsymbol{x}; \boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}; \boldsymbol{\theta}'), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\alpha}{4} \|\boldsymbol{x} - \boldsymbol{y}\|^2 - \frac{h^2}{\alpha} \|\boldsymbol{\theta}' - \boldsymbol{\theta}\|^2$$

Proof. By the strong convexity of $C(\boldsymbol{x}; \boldsymbol{\theta})$, for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{X}^T$ and any $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta^T$, we obtain the following.

$$\begin{aligned}
&C(\boldsymbol{y}; \boldsymbol{\theta}) \geq C(\boldsymbol{x}; \boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}; \boldsymbol{\theta}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\alpha}{2} \|\boldsymbol{y} - \boldsymbol{x}\|^2 \\
&= C(\boldsymbol{x}; \boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}; \boldsymbol{\theta}'), \boldsymbol{y} - \boldsymbol{x} \rangle - \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}; \boldsymbol{\theta}') - \nabla_{\boldsymbol{x}} C(\boldsymbol{x}; \boldsymbol{\theta}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\alpha}{2} \|\boldsymbol{y} - \boldsymbol{x}\|^2
\end{aligned}$$

$$\geq C(\boldsymbol{x};\boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}'), \boldsymbol{y}-\boldsymbol{x}\rangle - \|\nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}') - \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta})\|\|\boldsymbol{y}-\boldsymbol{x}\| + \frac{\alpha}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2$$
$$\geq C(\boldsymbol{x};\boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}'), \boldsymbol{y}-\boldsymbol{x}\rangle - \frac{1}{\alpha}\|\nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}') - \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta})\|^2 + \frac{\alpha}{4}\|\boldsymbol{y}-\boldsymbol{x}\|^2$$
$$\geq C(\boldsymbol{x};\boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}'), \boldsymbol{y}-\boldsymbol{x}\rangle - \frac{h^2}{\alpha}\|\boldsymbol{\theta}'-\boldsymbol{\theta}\|^2 + \frac{\alpha}{4}\|\boldsymbol{y}-\boldsymbol{x}\|^2$$

□

Lemma A.10 (Smoothness inequality with errors)**.** *Consider optimization* $\min_{\boldsymbol{x}\in\mathbb{X}^T} C(\boldsymbol{x};\boldsymbol{\theta})$*. For any* $\boldsymbol{x},\boldsymbol{y} \in \mathbb{X}^T$*, for any inexact parameter* $\boldsymbol{\theta}'$ *and the resulting inexact gradient* $\nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}')$*, we have*

$$C(\boldsymbol{y};\boldsymbol{\theta}) \leq C(\boldsymbol{x};\boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}'), \boldsymbol{y}-\boldsymbol{x}\rangle + L\|\boldsymbol{x}-\boldsymbol{y}\|^2 + \frac{h^2}{2L}\|\boldsymbol{\theta}'-\boldsymbol{\theta}\|^2$$

Proof. By the smoothness of $C(\boldsymbol{x};\boldsymbol{\theta})$, for any $\boldsymbol{x},\boldsymbol{y} \in \mathbb{X}^T$ and any $\boldsymbol{\theta},\boldsymbol{\theta}' \in \Theta^T$, we obtain the following.

$$\begin{aligned}
C(\boldsymbol{y};\boldsymbol{\theta}) &\leq C(\boldsymbol{x};\boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}), \boldsymbol{y}-\boldsymbol{x}\rangle + \frac{L}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2 \\
&= C(\boldsymbol{x};\boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}'), \boldsymbol{y}-\boldsymbol{x}\rangle + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}) - \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}'), \boldsymbol{y}-\boldsymbol{x}\rangle + \frac{L}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2 \\
&\leq C(\boldsymbol{x};\boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}'), \boldsymbol{y}-\boldsymbol{x}\rangle + \|\nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}') - \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta})\|\|\boldsymbol{y}-\boldsymbol{x}\| + \frac{L}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2 \\
&\leq C(\boldsymbol{x};\boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}'), \boldsymbol{y}-\boldsymbol{x}\rangle + \frac{1}{2L}\|\nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}') - \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta})\|^2 + L\|\boldsymbol{y}-\boldsymbol{x}\|^2 \\
&\leq C(\boldsymbol{x};\boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x};\boldsymbol{\theta}'), \boldsymbol{y}-\boldsymbol{x}\rangle + \frac{h^2}{2L}\|\boldsymbol{\theta}'-\boldsymbol{\theta}\|^2 + L\|\boldsymbol{y}-\boldsymbol{x}\|^2
\end{aligned}$$

□

Proof of Theorem 2.5.

According to Algorithm 3 and the definition of the regret, we have $\text{Regret}^d(RHIG) = C(\boldsymbol{x}(W);\boldsymbol{\theta}) - C(\boldsymbol{x}^*;\boldsymbol{\theta})$ and $\text{Regret}^d(\phi) = C(\boldsymbol{x}(0);\boldsymbol{\theta}) - C(\boldsymbol{x}^*;\boldsymbol{\theta})$, where $\boldsymbol{x}^* = \arg\min_{\mathbb{X}^T} C(\boldsymbol{x};\boldsymbol{\theta})$. For notational simplicity, we denote $r_k = \|\boldsymbol{x}(k) - \boldsymbol{x}^*\|^2$.

Step 1: bound $\mathrm{Reg}(RHIG)$ *with* r_{W-1}.

$$\begin{aligned}
C(\boldsymbol{x}(W);\boldsymbol{\theta}) &\leq C(\boldsymbol{x}(W-1);\boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(W-1);\boldsymbol{\theta}-\boldsymbol{\delta}(1)), \boldsymbol{x}(W)-\boldsymbol{x}(W-1)\rangle \\
&\quad + L\|\boldsymbol{x}(W)-\boldsymbol{x}(W-1)\|^2 + \frac{h^2}{2L}\|\boldsymbol{\delta}(1)\|^2 \\
&= \min_{\boldsymbol{x}\in\mathbb{X}^T} \left\{ \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(W-1);\boldsymbol{\theta}-\boldsymbol{\delta}(1)), \boldsymbol{x}-\boldsymbol{x}(W-1)\rangle + L\|\boldsymbol{x}-\boldsymbol{x}(W-1)\|^2 \right\} \\
&\quad + C(\boldsymbol{x}(W-1);\boldsymbol{\theta}) + \frac{h^2}{2L}\|\boldsymbol{\delta}(1)\|^2 \\
&\leq \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(W-1);\boldsymbol{\theta}-\boldsymbol{\delta}(1)), \boldsymbol{x}^*-\boldsymbol{x}(W-1)\rangle + L\|\boldsymbol{x}^*-\boldsymbol{x}(W-1)\|^2 \\
&\quad + C(\boldsymbol{x}(W-1);\boldsymbol{\theta}) + \frac{h^2}{2L}\|\boldsymbol{\delta}(1)\|^2 \\
&\leq C(\boldsymbol{x}^*;\boldsymbol{\theta}) + (L-\frac{\alpha}{4})r_{W-1} + \left(\frac{h^2}{\alpha}+\frac{h^2}{2L}\right)\|\boldsymbol{\delta}(1)\|^2
\end{aligned}$$

where we used Lemma A.10 in the first inequality, Lemma A.8 and $\eta = \frac{1}{2L}$ in the first equality, Lemma A.9 in the last inequality. By rearranging terms, we obtain

$$\mathrm{Regret}^d(RHIG) = C(\boldsymbol{x}(W);\boldsymbol{\theta}) - C(\boldsymbol{x}^*;\boldsymbol{\theta}) \leq L\rho r_{W-1} + \zeta\|\boldsymbol{\delta}(1)\|^2 \tag{A.2}$$

where $\rho = 1-\frac{\alpha}{4L}$, $\zeta = \frac{h^2}{\alpha}+\frac{h^2}{2L}$.

Step 2: a recursive inequality between r_{k+1} *and* r_k.

In the following, we will show that

$$r_{k+1} \leq \rho r_k + \frac{\zeta}{L}\|\boldsymbol{\delta}(W-k)\|^2, \quad \forall\ 0\leq k\leq W-1 \tag{A.3}$$

Firstly, by (A.1), $\eta = \frac{1}{2L}$, Lemma A.8 and its first-order optimality condition, we have

$$\langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k);\boldsymbol{\theta}-\delta(W-k)) + 2L(\boldsymbol{x}(k+1)-\boldsymbol{x}(k)), \boldsymbol{x}-\boldsymbol{x}(k+1)\rangle \geq 0, \quad \forall\ \boldsymbol{x}\in\mathbb{X}^T$$

By substituting $\boldsymbol{x} = \boldsymbol{x}^*$ and rearranging terms, we obtain

$$\frac{1}{2L}\langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k);\boldsymbol{\theta}-\delta(W-k)), \boldsymbol{x}^*-\boldsymbol{x}(k+1)\rangle \geq \langle \boldsymbol{x}(k+1)-\boldsymbol{x}(k), \boldsymbol{x}(k+1)-\boldsymbol{x}^*\rangle \tag{A.4}$$

Next, we will derive the recursive inequality (A.3) by using (A.4).

$$
\begin{aligned}
r_{k+1} &= \|\boldsymbol{x}(k+1)-\boldsymbol{x}^*\|^2 = \|\boldsymbol{x}(k+1)-\boldsymbol{x}(k)+\boldsymbol{x}(k)-\boldsymbol{x}^*\|^2 \\
&= r_k - \|\boldsymbol{x}(k+1)-\boldsymbol{x}(k)\|^2 + 2\langle \boldsymbol{x}(k+1)-\boldsymbol{x}(k), \boldsymbol{x}(k+1)-\boldsymbol{x}^*\rangle \\
&\le r_k - \|\boldsymbol{x}(k+1)-\boldsymbol{x}(k)\|^2 + \frac{1}{L}\langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k);\boldsymbol{\theta}-\boldsymbol{\delta}(W-k)), \boldsymbol{x}^*-\boldsymbol{x}(k+1)\rangle \\
&= r_k - \|\boldsymbol{x}(k+1)-\boldsymbol{x}(k)\|^2 + \frac{1}{L}\langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k);\boldsymbol{\theta}-\boldsymbol{\delta}(W-k)), \boldsymbol{x}^*-\boldsymbol{x}(k)\rangle \\
&\quad + \frac{1}{L}\langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k);\boldsymbol{\theta}-\boldsymbol{\delta}(W-k)), \boldsymbol{x}(k)-\boldsymbol{x}(k+1)\rangle \\
&= r_k + \frac{1}{L}\langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k);\boldsymbol{\theta}-\boldsymbol{\delta}(W-k)), \boldsymbol{x}^*-\boldsymbol{x}(k)\rangle \\
&\quad - \frac{1}{L}\left(\langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k);\boldsymbol{\theta}-\boldsymbol{\delta}(W-k)), \boldsymbol{x}(k+1)-\boldsymbol{x}(k)\rangle + L\|\boldsymbol{x}(k+1)-\boldsymbol{x}(k)\|^2\right) \\
&\le r_k + \frac{1}{L}\langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k);\boldsymbol{\theta}-\boldsymbol{\delta}(W-k)), \boldsymbol{x}^*-\boldsymbol{x}(k)\rangle \\
&\quad - \frac{1}{L}\left(C(\boldsymbol{x}(k+1);\boldsymbol{\theta}) - C(\boldsymbol{x}(k);\boldsymbol{\theta}) - \frac{h^2}{2L}\|\boldsymbol{\delta}(W-k)\|^2\right) \\
&\le r_k + \frac{1}{L}\langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k);\boldsymbol{\theta}-\boldsymbol{\delta}(W-k)), \boldsymbol{x}^*-\boldsymbol{x}(k)\rangle \\
&\quad - \frac{1}{L}\left(C(\boldsymbol{x}^*;\boldsymbol{\theta}) - C(\boldsymbol{x}(k);\boldsymbol{\theta})\right) + \frac{h^2}{2L^2}\|\boldsymbol{\delta}(W-k)\|^2 \\
&= r_k - \frac{1}{L}\left(C(\boldsymbol{x}^*;\boldsymbol{\theta}) - C(\boldsymbol{x}(k);\boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{x}} C(\boldsymbol{x}(k);\boldsymbol{\theta}-), \boldsymbol{x}(k)-\boldsymbol{x}^*\rangle\right) \\
&\quad + \frac{h^2}{2L^2}\|\boldsymbol{\delta}(W-k)\|^2 \\
&\le r_k - \frac{1}{L}\left(\frac{\alpha}{4}\|\boldsymbol{x}(k)-\boldsymbol{x}^*\|^2 - \frac{h^2}{\alpha}\|\boldsymbol{\delta}(W-k)\|^2\right) + \frac{h^2}{2L^2}\|\boldsymbol{\delta}(W-k)\|^2 \\
&= \rho r_k + \frac{\zeta}{L}\|\boldsymbol{\delta}(W-k)\|^2
\end{aligned}
$$

which completes the proof of (A.3).

Step 3: completing the proof by (A.3) *and* (A.2).

By summing (A.3) over $k = 0, \ldots, W-2$, we obtain

$$
r_{W-1} \le \rho^{W-1} r_0 + \frac{\zeta}{L}\left(\|\boldsymbol{\delta}(2)\|^2 + \rho\|\boldsymbol{\delta}(3)\|^2 + \cdots + \rho^{W-2}\|\boldsymbol{\delta}(W)\|^2\right)
$$

$$\leq \rho^{W-1}\frac{2}{\alpha}(C(\boldsymbol{x}(0);\boldsymbol{\theta}) - C(\boldsymbol{x}^*;\boldsymbol{\theta})) + \frac{\zeta}{L}\sum_{k=2}^{W}\rho^{k-2}\|\boldsymbol{\delta}(k)\|^2$$

By (A.2), we obtain the regret bound in Theorem 2.5:

$$\begin{aligned}
\text{Regret}^d(RHIG) &\leq L\rho(\rho^{W-1}\frac{2}{\alpha}(C(\boldsymbol{x}(0);\boldsymbol{\theta}) - C(\boldsymbol{x}^*;\boldsymbol{\theta})) + \frac{\zeta}{L}\sum_{k=2}^{W}\rho^{k-2}\|\boldsymbol{\delta}(k)\|^2) + \zeta\|\boldsymbol{\delta}(1)\|^2 \\
&= \frac{2L}{\alpha}\rho^W\text{Regret}^d(\phi) + \zeta\sum_{k=1}^{W}\rho^{k-1}\|\boldsymbol{\delta}(k)\|^2 \\
&= \frac{2L}{\alpha}\rho^W\text{Regret}^d(\phi) + \zeta\sum_{k=1}^{\min(W,T)}\rho^{k-1}\|\boldsymbol{\delta}(k)\|^2 + \zeta\mathbf{1}_{(W>T)}\sum_{k=T+1}^{W}\rho^{k-1}\|\boldsymbol{\delta}(T)\|^2 \\
&= \frac{2L}{\alpha}\rho^W\text{Regret}^d(\phi) + \zeta\sum_{k=1}^{\min(W,T)}\rho^{k-1}\|\boldsymbol{\delta}(k)\|^2 + \zeta\mathbf{1}_{(W>T)}\frac{\rho^T-\rho^W}{1-\rho}\|\boldsymbol{\delta}(T)\|^2
\end{aligned}$$

where we used the fact that $\|\boldsymbol{\delta}(k)\| = \|\boldsymbol{\delta}(T)\|$ when $k > T$.

A.1.6 Proofs of Theorem 2.6

In this section, we provide a dynamic regret bound for the restarted OGD initialization rule in Section 2.3.3. To achieve this, we will first establish a static regret bound for OGD initialization (2.14). The proof is inspired by [20].

For notational simplicity, we slightly abuse the notation and let x_t denote $x_t(0)$ generated by OGD. Further, by the definition of the prediction errors $\delta_{t-1}(W)$ for $W \geq 1$, we can write the initialization rule (2.14) as the following, which can be interpreted as OGD with inexact gradients:

$$x_t = \Pi_{\mathbb{X}}[x_{t-1} - \xi_t\nabla_x f(x_{t-1};\theta_{t-1} - \delta_{t-1}(\min(W,T)))], \quad t \geq 2; \tag{A.5}$$

and $x_1 = x_0$. Here, we used the facts that $\theta_{t-1|t-W-1} = \theta_{t-1} - \delta_{t-1}(W)$ and $\delta_{t-1}(W) = \delta_{t-1}(T)$ for $W > T$.

Static regret bound for OGD with inexact gradients In this part, we consider the

OGD with inexact gradients (A.5) with diminishing stepsize $\xi_t = \frac{4}{\alpha t}$ for $t \geq 1$. We will prove its static regret bound below.

Theorem A.1 (Static regret of OGD with inexact gradients)**.** *Consider the OGD with inexact gradients* (A.5) *with diminishing stepsize* $\xi_t = \frac{4}{\alpha t}$ *for* $t \geq 1$ *and any* x_0*. Then, for* $z^* = \arg\min_{z \in \mathbb{X}} \sum_{t=1}^{T} f(z; \theta_t)$*, we have the following static regret bound:*

$$\sum_{t=1}^{T} [f(x_t; \theta_t) - f(z^*; \theta_t)] \leq \frac{2G^2}{\alpha} \log(T+1) + \sum_{t=1}^{T} \frac{h^2}{\alpha} \|\delta_t(\min(W, T))\|^2$$

Further, the total switching cost can be bounded by:

$$\sum_{t=1}^{T} d(x_t, x_{t-1}) \leq \frac{16G^2\beta}{\alpha^2}$$

Proof. Firstly, we prove the static regret bound. Define $q_t = \|x_t - z^*\|^2$. Then, for $t \geq 1$, we have the following.

$$\begin{aligned}
q_{t+1} =& \|x_{t+1} - z^*\|^2 \leq \|x_t - \xi_{t+1} \nabla_x f(x_t; \theta_t - \delta_t(\min(W, T))) - z^*\|^2 \\
=& q_t + \xi_{t+1}^2 \|\nabla_x f(x_t; \theta_t - \delta_t(\min(W, T)))\|^2 - 2\xi_{t+1} \langle x_t - z^*, \nabla_x f(x_t; \theta_t - \delta_t(\min(W, T))) \rangle \\
\leq& q_t + \xi_{t+1}^2 G^2 - 2\xi_{t+1} \langle x_t - z^*, \nabla_x f(x_t; \theta_t) \rangle \\
& - 2\xi_{t+1} \langle x_t - z^*, \nabla_x f(x_t; \theta_t - \delta_t(\min(W, T))) - \nabla_x f(x_t; \theta_t) \rangle
\end{aligned}$$

By rearranging terms, we obtain

$$\begin{aligned}
\langle x_t - z^*, \nabla_x f(x_t; \theta_t) \rangle \leq & \frac{q_t - q_{t+1}}{2\xi_{t+1}} + \frac{\xi_{t+1}}{2} G^2 \\
& - \langle x_t - z^*, \nabla_x f(x_t; \theta_t - \delta_t(\min(W, T))) - \nabla_x f(x_t; \theta_t) \rangle
\end{aligned} \tag{A.6}$$

By the strong convexity of $f(x; \theta_t)$, we have $f(z^*; \theta_t) \geq f(x_t; \theta_t) + \langle z^* - x_t, \nabla_x f(x_t; \theta_t) \rangle + \frac{\alpha}{2} \|z^* - x_t\|^2$. By rearranging terms and by (A.6), we obtain

$$f(x_t; \theta_t) - f(z^*; \theta_t) \leq \langle x_t - z^*, \nabla_x f(x_t; \theta_t) \rangle - \frac{\alpha}{2} \|z^* - x_t\|^2$$

$$
\begin{aligned}
&\leq \frac{q_t - q_{t+1}}{2\xi_{t+1}} + \frac{\xi_{t+1}}{2}G^2 - \langle x_t - z^*, \nabla_x f(x_t;\theta_t - \delta_t(\min(W,T))) - \nabla_x f(x_t;\theta_t)\rangle - \frac{\alpha}{2}\|z^* - x_t\|^2 \\
&\leq \frac{q_t - q_{t+1}}{2\xi_{t+1}} + \frac{\xi_{t+1}}{2}G^2 + \|x_t - z^*\|\|\nabla_x f(x_t;\theta_t - \delta_t(\min(W,T))) - \nabla_x f(x_t;\theta_t)\| - \frac{\alpha}{2}\|z^* - x_t\|^2 \\
&\leq \frac{q_t - q_{t+1}}{2\xi_{t+1}} + \frac{\xi_{t+1}}{2}G^2 + \frac{1}{\alpha}\|\nabla_x f(x_t;\theta_t - \delta_t(\min(W,T))) - \nabla_x f(x_t;\theta_t)\|^2 - \frac{\alpha}{4}\|z^* - x_t\|^2 \\
&\leq \frac{q_t - q_{t+1}}{2\xi_{t+1}} + \frac{\xi_{t+1}}{2}G^2 + \frac{h^2}{\alpha}\|\delta_t(\min(W,T))\|^2 - \frac{\alpha}{4}q_t
\end{aligned}
$$

where we used $ab \leq \frac{\epsilon}{2}a^2 + \frac{1}{2\epsilon}b^2$ for any $a, b \in \mathbb{R}$ and any $\epsilon > 0$ in the second last inequality and Assumption 2.4 in the last inequality. By summing over $t = 1, \ldots, T$, we obtain

$$
\begin{aligned}
\sum_{t=1}^{T}[f(x_t;\theta_t) - f(z^*;\theta_t)] \leq & \sum_{t=2}^{T}\left(\frac{1}{2\xi_{t+1}} - \frac{1}{2\xi_t} - \frac{\alpha}{4}\right)q_t + \left(\frac{1}{2\xi_2} - \frac{\alpha}{4}\right)q_1 - \frac{1}{\xi_{T+1}}q_{T+1} \\
& + \sum_{t=1}^{T}\frac{\xi_{t+1}}{2}G^2 + \sum_{t=1}^{T}\frac{h^2}{\alpha}\|\delta_t(\min(W,T))\|^2 \\
\leq & \ \log(T+1)\frac{2G^2}{\alpha} + \sum_{t=1}^{T}\frac{h^2}{\alpha}\|\delta_t(\min(W,T))\|^2
\end{aligned}
$$

which completes the proof of the static regret bound.

Next, we bound the switching costs. We have

$$
\sum_{t=1}^{T}\frac{\beta}{2}\|x_t - x_{t-1}\|^2 \leq \sum_{t=1}^{T}\frac{\beta}{2}\|\xi_t\nabla_x f(x_{t-1};\theta_{t-1} - \delta_{t-1}(\min(W,T)))\|^2 \leq \frac{\beta G^2}{2}\sum_{t=1}^{T}\xi_t^2 \leq \frac{16\beta G^2}{\alpha^2}
$$

□

Proof of Theorem 2.6: dynamic regret bound for restarted OGD with inexact gradients

We denote the set of stages in epoch k as $\mathscr{T}_k = \{k\Delta + 1, \ldots, \min(k\Delta + \Delta, T)\}$ for $k = 0, \ldots, \lceil T/\Delta \rceil - 1$. We introduce $z_k^* = \arg\min_{z\in\mathbb{X}}\sum_{t\in\mathscr{T}_k}[f(z;\theta_t)]$ for all k; $y_t^* = \arg\min_{x_t\in\mathbb{X}} f(x_t;\theta_t)$ for all t; and $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathbb{X}^T}\sum_{t=1}^{T}[f(x_t;\theta_t) + d(x_t, x_{t-1})]$.

The dynamic regret of the restarted OGD with inexact gradients can be bounded as follows.

$$\begin{aligned}
\text{Regret}^d(OGD) &= \sum_{t=1}^{T}[f(x_t;\theta_t)+d(x_t,x_{t-1})] - \sum_{t=1}^{T}[f(x_t^*;\theta_t)+d(x_t^*,x_{t-1}^*)] \\
&\leq \sum_{t=1}^{T}[f(x_t;\theta_t)+d(x_t,x_{t-1})] - \sum_{t=1}^{T}[f(x_t^*;\theta_t)] \\
&\leq \sum_{t=1}^{T}[f(x_t;\theta_t)+d(x_t,x_{t-1})] - \sum_{t=1}^{T}[f(y_t^*;\theta_t)] \\
&= \sum_{k=0}^{\lceil T/\Delta\rceil-1}\sum_{t\in\mathscr{T}_k}[f(x_t;\theta_t)+d(x_t,x_{t-1})-f(y_t^*;\theta_t)] \\
&= \sum_{k=0}^{\lceil T/\Delta\rceil-1}\sum_{t\in\mathscr{T}_k}[f(x_t;\theta_t)-f(z_k^*;\theta_t)] + \sum_{k=0}^{\lceil T/\Delta\rceil-1}\sum_{t\in\mathscr{T}_k} d(x_t,x_{t-1}) \\
&\quad + \sum_{k=0}^{\lceil T/\Delta\rceil-1}\sum_{t\in\mathscr{T}_k}[f(z_k^*;\theta_t)-f(y_t^*;\theta_t)] \\
&\leq \lceil T/\Delta\rceil \log(\Delta+1)\frac{2G^2}{\alpha} + \frac{h^2}{\alpha}\|\boldsymbol{\delta}(\min(W,T))\|^2 + \lceil T/\Delta\rceil\frac{16\beta G^2}{\alpha^2} \\
&\quad + \sum_{k=0}^{\lceil T/\Delta\rceil-1}\sum_{t\in\mathscr{T}_k}[f(z_k^*;\theta_t)-f(y_t^*;\theta_t)]
\end{aligned}$$

where the second inequality uses the optimality of y_t^*, the last inequality uses Theorem A.1 and the fact that the OGD considered here restarts at the beginning of each epoch k and repeats the stepsizes defined in Theorem A.1, thus satisfying the static regret bound and the switching cost bound in Theorem A.1 within each epoch.

Now, it suffices to bound $\sum_{k=0}^{\lceil T/\Delta\rceil-1}\sum_{t\in\mathscr{T}_k}[f(z_k^*;\theta_t)-f(y_t^*;\theta_t)]$. By the optimality of z_k^*, we have:

$$\sum_{k=0}^{\lceil T/\Delta\rceil-1}\sum_{t\in\mathscr{T}_k}[f(z_k^*;\theta_t)-f(y_t^*;\theta_t)] \leq \sum_{k=0}^{\lceil T/\Delta\rceil-1}\sum_{t\in\mathscr{T}_k}[f(y_{k\Delta+1}^*;\theta_t)-f(y_t^*;\theta_t)]. \tag{A.7}$$

We define $V^k = \sum_{t\in\mathscr{T}_k}\sup_{x\in\mathbb{X}}|f(x;\theta_t)-f(x;\theta_{t-1})|$. Then, for any $t\in\mathscr{T}_k$, we obtain

$$f(y_{k\Delta+1}^*;\theta_t)-f(y_t^*;\theta_t) = f(y_{k\Delta+1}^*;\theta_t)-f(y_{k\Delta+1}^*;\theta_{k\Delta+1})+f(y_{k\Delta+1}^*;\theta_{k\Delta+1})-f(y_t^*;\theta_{k\Delta+1})$$

$$
\begin{aligned}
&+ f(y_t^*;\theta_{k\Delta+1}) - f(y_t^*;\theta_t) \\
&\le V^k + 0 + V^k = 2V^k
\end{aligned}
$$

By summing over $t \in \mathscr{T}_k$ and $k = 0, \ldots, \lceil T/\Delta \rceil - 1$ and by the inequality (A.7), we obtain

$$
\sum_{k=0}^{\lceil T/\Delta \rceil - 1} \sum_{t \in \mathscr{T}_k} [f(z_k^*;\theta_t) - f(y_t^*;\theta_t)] \le \sum_{k=0}^{\lceil T/\Delta \rceil - 1} \sum_{t \in \mathscr{T}_k} 2V^k = \sum_{k=0}^{\lceil T/\Delta \rceil - 1} 2\Delta V^k = 2\Delta V_T
$$

Combining the bounds above yields the desired bound on the dynamic regret of OGD below by letting $\Delta = \lceil \sqrt{2T/V_T} \rceil$:

$$
\begin{aligned}
&\text{Regret}^d(OGD) \le \lceil T/\Delta \rceil \log(\Delta+1)\frac{2G^2}{\alpha} + \frac{h^2}{\alpha}\|\boldsymbol{\delta}(\min(W,T))\|^2 + \lceil T/\Delta \rceil \frac{16\beta G^2}{\alpha^2} + 2\Delta V_T \\
&\le \left(\sqrt{\frac{V_T T}{2}} + 1\right) \log(2+\sqrt{2T/V_T})\frac{2G^2}{\alpha} + \frac{h^2}{\alpha}\|\boldsymbol{\delta}(\min(W,T))\|^2 \\
&\quad + \left(\sqrt{\frac{V_T T}{2}} + 1\right) \frac{16\beta G^2}{\alpha^2} + 2(\sqrt{2V_T T} + V_T) \\
&\le (\sqrt{V_T T/2}+1) \log(2+\sqrt{2T/V_T}) \left(\frac{2G^2}{\alpha} + \frac{16\beta G^2}{\alpha^2} + 2(2+\sqrt{2})\right) + \frac{h^2}{\alpha}\|\boldsymbol{\delta}(\min(W,T))\|^2 \\
&\le \sqrt{2V_T T} \log(2+\sqrt{2T/V_T}) \left(\frac{2G^2}{\alpha} + \frac{16\beta G^2}{\alpha^2} + 2(2+\sqrt{2})\right) + \frac{h^2}{\alpha}\|\boldsymbol{\delta}(\min(W,T))\|^2 \\
&\le \sqrt{V_T T} \log(1+\sqrt{T/V_T}) \left(\frac{4\sqrt{2}G^2}{\alpha} + \frac{32\sqrt{2}\beta G^2}{\alpha^2} + 8(1+\sqrt{2})\right) + \frac{h^2}{\alpha}\|\boldsymbol{\delta}(\min(W,T))\|^2 \\
&\le \sqrt{V_T T} \log(1+\sqrt{T/V_T}) \left(\frac{4\sqrt{2}G^2}{\alpha} + \frac{32\sqrt{2}\beta G^2}{\alpha^2} + 20\right) + \frac{h^2}{\alpha}\|\boldsymbol{\delta}(\min(W,T))\|^2
\end{aligned}
$$

where we used the facts that $\lceil x \rceil \le x + 1$, $1 \le V_T \le T$, $T > 2$, $\log(2 + \sqrt{2T/V_T}) \le 2\log(1 + \sqrt{T/V_T})$, and $8(1 + \sqrt{2}) < 20$.

A.1.7 Proof of Theorem 2.7

By taking expectation on both sides of the regret bound in Theorem 2.5, we have

$$\begin{aligned}\mathbb{E}[\text{Regret}^d(RHIG)] \leq \frac{2L}{\alpha}\rho^W \mathbb{E}[\text{Regret}^d(\phi)] + \zeta \sum_{k=1}^{\min(W,T)} \rho^{k-1} \mathbb{E}[\|\boldsymbol{\delta}(k)\|^2] \\ + \mathbf{1}_{(W>T)} \frac{\rho^T - \rho^W}{1-\rho} \zeta \mathbb{E}[\|\boldsymbol{\delta}(T)\|^2].\end{aligned} \tag{A.8}$$

Therefore, it suffices to bound $\mathbb{E}[\|\boldsymbol{\delta}(k)\|^2]$ for $1 \leq k \leq T$. By $\boldsymbol{\delta}(k) = (\delta_1(k)^\top, \ldots, \delta_T(k)^\top)^\top$, $\delta_t(k) = \theta_t - \theta_{t|t-k} = P(0)e_t + \cdots + P(k-1)e_{t-k+1}$ for $k \leq t$ and $\delta_t(k) = \delta_t(t)$ for $k > t$, we have

$$\boldsymbol{\delta}(k) = \boldsymbol{M}_k \boldsymbol{e}, \quad 1 \leq k \leq T \tag{A.9}$$

where we define $\boldsymbol{e} = (e_1^\top, \ldots, e_T^\top)^\top \in \mathbb{R}^{qT}$ and

$$\boldsymbol{M}_k = \begin{bmatrix} P(0) & 0 & \ldots & \ldots & \ldots & 0 \\ P(1) & P(0) & \ldots & \ldots & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ P(k-1) & \ldots & P(1) & P(0) & \ldots & 0 \\ \vdots & \ddots & & \ddots & \ddots & 0 \\ 0 & \ldots & P(k-1) & \ldots & P(1) & P(0) \end{bmatrix}.$$

Let $\boldsymbol{R_e}$ denote the covariance matrix of $\boldsymbol{e}$, i.e.

$$\boldsymbol{R_e} = \begin{bmatrix} R_e & 0 & \ldots & 0 \\ 0 & R_e & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & \ldots & R_e \end{bmatrix}$$

Then, for $k \leq T$, we have

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{\delta}(k)\|^2] &= \mathbb{E}[\boldsymbol{e}^\top \boldsymbol{M}_k^\top \boldsymbol{M}_k \boldsymbol{e}] = \mathbb{E}[\mathrm{tr}(\boldsymbol{e}\boldsymbol{e}^\top \boldsymbol{M}_k^\top \boldsymbol{M}_k)] \\ &= \mathrm{tr}\left(\boldsymbol{R}_{\boldsymbol{e}} \boldsymbol{M}_k^\top \boldsymbol{M}_k\right) \\ &\leq \|R_e\| \|\boldsymbol{M}_k\|_F^2 = \|R_e\| \sum_{t=0}^{k-1} (T-t)\|P(t)\|_F^2\end{aligned}$$

where the first inequality is by $\mathrm{tr}(AB) \leq \|A\|\mathrm{tr}(B)$ for any symmetrix matrices A, B, and $\|\mathrm{diag}(R_e, \ldots, R_e)\| = \|R_e\|$ and $\mathrm{tr}(A^\top A) = \|A\|_F^2$ for any matrix A. In addition, for $k \geq T$, we have $\mathbb{E}[\|\boldsymbol{\delta}(k)\|^2] \leq \|R_e\| \sum_{t=0}^{T-1} (T-t)\|P(t)\|_F^2$. In conclusion, for any $k \geq 1$, we have

$$\mathbb{E}[\|\boldsymbol{\delta}(k)\|^2] \leq \|R_e\| \sum_{t=0}^{\min(k,T)-1} (T-t)\|P(t)\|_F^2 \tag{A.10}$$

When $W \leq T$, substituting the bounds on $\mathbb{E}[\|\boldsymbol{\delta}(k)\|^2]$ into (A.8) yields the bound on the expected regret below.

$$\begin{aligned}\mathbb{E}[\mathrm{Regret}^d(RHIG)] &\leq \frac{2L}{\alpha}\rho^W \mathbb{E}[\mathrm{Regret}^d(\phi)] + \zeta \sum_{k=1}^{W} \rho^{k-1} \|R_e\| \sum_{t=0}^{k-1} (T-t)\|P(t)\|_F^2 \\ &= \frac{2L}{\alpha}\rho^W \mathbb{E}[\mathrm{Regret}^d(\phi)] + \zeta \sum_{t=0}^{W-1} \|R_e\|(T-t)\|P(t)\|_F^2 \sum_{k=t+1}^{W} \rho^{k-1} \\ &= \frac{2L}{\alpha}\rho^W \mathbb{E}[\mathrm{Regret}^d(\phi)] + \zeta \sum_{t=0}^{W-1} \|R_e\|(T-t)\|P(t)\|_F^2 \frac{\rho^t - \rho^W}{1-\rho}\end{aligned}$$

When $W \geq T$, substituting the bounds on $\mathbb{E}[\|\boldsymbol{\delta}(k)\|^2]$ into (A.8) yields the bound on the expected regret below.

$$\begin{aligned}\mathbb{E}[\mathrm{Regret}^d(RHIG)] \leq{}& \frac{2L}{\alpha}\rho^W \mathbb{E}[\mathrm{Regret}^d(\phi)] + \zeta \sum_{k=1}^{T} \rho^{k-1} \|R_e\| \sum_{t=0}^{k-1} (T-t)\|P(t)\|_F^2 \\ &+ \zeta \frac{\rho^T - \rho^W}{1-\rho} \|R_e\| \sum_{t=0}^{T-1} (T-t)\|P(t)\|_F^2 \\ ={}& \frac{2L}{\alpha}\rho^W \mathbb{E}[\mathrm{Regret}^d(\phi)] + \zeta \sum_{t=0}^{T-1} \|R_e\|(T-t)\|P(t)\|_F^2 \Big(\sum_{k=t+1}^{T} \rho^{k-1} + \frac{\rho^T - \rho^W}{1-\rho} \Big)\end{aligned}$$

$$= \frac{2L}{\alpha}\rho^W \mathbb{E}[\text{Regret}^d(\phi)] + \zeta \sum_{t=0}^{T-1} \|R_e\|(T-t)\|P(t)\|_F^2 \frac{\rho^t - \rho^W}{1-\rho}$$

In conclusion, we have the regret bound for general $W \geq 0$ below.

$$\mathbb{E}[\text{Regret}^d(RHIG)] \leq \frac{2L}{\alpha}\rho^W \text{Regret}^d(\phi) + \zeta \sum_{t=0}^{\min(W,T)-1} \|R_e\|(T-t)\|P(t)\|_F^2 \frac{\rho^t - \rho^W}{1-\rho}$$

A.1.8 Proof of Theorem 2.8

The proof relies on the Hanson-Wright inequality in [222].[5]

Proposition A.1 (Hanson-Wright Inequality [222]). *Consider random Gaussian vector* $\boldsymbol{u} = (u_1, \ldots, u_n)^\top$ *with* u_i *i.i.d. following* $N(0,1)$. *There exists an absolute constant* $c > 0$,[6] *such that*

$$\mathbb{P}(\boldsymbol{u}^\top A \boldsymbol{u} \geq \mathbb{E}[\boldsymbol{u}^\top A \boldsymbol{u}] + b) \leq \exp\left(-c \min(\frac{b^2}{\|A\|_F^2}, \frac{b}{\|A\|})\right), \quad \forall\ b > 0$$

Now, we are ready for the proof. For any realization of the random vectors $\{e_t\}_{t=1}^T$, our regret bound in Corollary 2.2 still holds, i.e.

$$\begin{aligned}
\text{Regret}^d(RHIG) \leq & \rho^W \frac{2L}{\alpha} C_1 \sqrt{V_T T} \log(1 + \sqrt{T/V_T}) + \frac{2L}{\alpha}\frac{h^2}{\alpha}\rho^W \|\boldsymbol{\delta}(\min(W,T))\|^2 \\
& + \sum_{k=1}^{\min(W,T)} \zeta \rho^{k-1} \|\boldsymbol{\delta}(k)\|^2 + \mathbf{1}_{(W>T)} \frac{\rho^T - \rho^W}{1-\rho} \zeta \|\boldsymbol{\delta}(T)\|^2 \\
\leq & \rho^W \frac{2L}{\alpha} C_1 T \log(2) + \frac{2L}{\alpha}\frac{h^2}{\alpha}\rho^W \|\boldsymbol{\delta}(\min(W,T))\|^2 + \sum_{k=1}^{\min(W,T)} \zeta \rho^{k-1} \|\boldsymbol{\delta}(k)\|^2 \\
& + \mathbf{1}_{(W>T)} \frac{\rho^T - \rho^W}{1-\rho} \zeta \|\boldsymbol{\delta}(T)\|^2 =: R(W)
\end{aligned}$$

[5] Here we use the fact that $\|X_i\|_\varphi = 1$ where $\|\cdot\|_\varphi$ is the subGaussian norm defined in [222].

[6] An absolute constant refers to a quantity that does not change with anything.

where we used the technical assumption that $V_T \leq T$. Notice that it can be verified that $\mathbb{E}[R(W)] \leq \mathbb{E}[\text{Regbdd}]$.

From (A.9) in the proof of Theorem 2.7, we have that $\boldsymbol{\delta}(k) = \boldsymbol{M}_k \boldsymbol{e} = \boldsymbol{M}_k \boldsymbol{R}_{\boldsymbol{e}}^{1/2} \boldsymbol{u}$, where $\boldsymbol{u}$ is a standard Gaussian vector for $k \leq T$; and $\boldsymbol{\delta}(k) = \boldsymbol{M}_T \boldsymbol{R}_{\boldsymbol{e}}^{1/2} \boldsymbol{u}$ for $k \geq T$.

When $W \leq T$, we have the following formula for $R(W)$.

$$
\begin{aligned}
R(W) &= \rho^W \frac{2L}{\alpha} C_1 T \log(2) + \rho^W \frac{2L}{\alpha} \frac{h^2}{\alpha} \|\boldsymbol{\delta}(W)\|^2 + \zeta \sum_{k=1}^{W} \rho^{k-1} \|\boldsymbol{\delta}(k)\|^2 \\
&= \rho^W \frac{2L}{\alpha} C_1 T \log(2) \\
&\quad + \boldsymbol{u}^\top \underbrace{\left(\rho^W \frac{2L}{\alpha} \frac{h^2}{\alpha} \boldsymbol{R}_{\boldsymbol{e}}^{1/2} \boldsymbol{M}_W^\top \boldsymbol{M}_W \boldsymbol{R}_{\boldsymbol{e}}^{1/2} + \zeta \sum_{k=1}^{W} \rho^{k-1} \boldsymbol{R}_{\boldsymbol{e}}^{1/2} \boldsymbol{M}_k^\top \boldsymbol{M}_k \boldsymbol{R}_{\boldsymbol{e}}^{1/2} \right)}_{\boldsymbol{A}_W} \boldsymbol{u}
\end{aligned}
$$

We bound $\|\boldsymbol{A}_W\|_F$ below.

$$
\begin{aligned}
\|\boldsymbol{A}_W\|_F &\leq \rho^W \frac{2L}{\alpha} \frac{h^2}{\alpha} \|\boldsymbol{R}_{\boldsymbol{e}}^{1/2} \boldsymbol{M}_W^\top \boldsymbol{M}_W \boldsymbol{R}_{\boldsymbol{e}}^{1/2}\|_F + \zeta \sum_{k=1}^{W} \rho^{k-1} \|\boldsymbol{R}_{\boldsymbol{e}}^{1/2} \boldsymbol{M}_k^\top \boldsymbol{M}_k \boldsymbol{R}_{\boldsymbol{e}}^{1/2}\|_F \\
&\leq \rho^W \frac{2L}{\alpha} \frac{h^2}{\alpha} \|\boldsymbol{M}_W \boldsymbol{R}_{\boldsymbol{e}}^{1/2}\|_F^2 + \zeta \sum_{k=1}^{W} \rho^{k-1} \|\boldsymbol{M}_k \boldsymbol{R}_{\boldsymbol{e}}^{1/2}\|_F^2 \\
&= \rho^W \frac{2L}{\alpha} \frac{h^2}{\alpha} \mathrm{tr}(\boldsymbol{R}_{\boldsymbol{e}}^{1/2} \boldsymbol{M}_W^\top \boldsymbol{M}_W \boldsymbol{R}_{\boldsymbol{e}}^{1/2}) + \zeta \sum_{k=1}^{W} \rho^{k-1} \mathrm{tr}(\boldsymbol{R}_{\boldsymbol{e}}^{1/2} \boldsymbol{M}_k^\top \boldsymbol{M}_k \boldsymbol{R}_{\boldsymbol{e}}^{1/2}) \\
&= \rho^W \frac{2L}{\alpha} \frac{h^2}{\alpha} \mathrm{tr}(\boldsymbol{R}_{\boldsymbol{e}} \boldsymbol{M}_W^\top \boldsymbol{M}_W) + \zeta \sum_{k=1}^{W} \rho^{k-1} \mathrm{tr}(\boldsymbol{R}_{\boldsymbol{e}} \boldsymbol{M}_k^\top \boldsymbol{M}_k) \\
&\leq \rho^W \frac{2L}{\alpha} \frac{h^2}{\alpha} \|R_e\| \mathrm{tr}(\boldsymbol{M}_W^\top \boldsymbol{M}_W) + \zeta \|R_e\| \sum_{k=1}^{W} \rho^{k-1} \mathrm{tr}(\boldsymbol{M}_k^\top \boldsymbol{M}_k) \\
&= \rho^W \frac{2L}{\alpha} \frac{h^2}{\alpha} \|R_e\| \|\boldsymbol{M}_W\|_F^2 + \zeta \|R_e\| \sum_{k=1}^{W} \rho^{k-1} \|\boldsymbol{M}_k\|_F^2 \\
&= \rho^W \frac{2L}{\alpha} \frac{h^2}{\alpha} \|R_e\| \sum_{t=0}^{W-1} (T-t) \|P(t)\|_F^2 + \zeta \|R_e\| \sum_{k=1}^{W} \rho^{k-1} \sum_{t=0}^{k-1} (T-t) \|P(t)\|_F^2 \\
&= \rho^W \frac{2L}{\alpha} \frac{h^2}{\alpha} \|R_e\| \sum_{t=0}^{W-1} (T-t) \|P(t)\|_F^2 + \zeta \|R_e\| \sum_{t=0}^{W-1} \sum_{k=t+1}^{W} \rho^{k-1} (T-t) \|P(t)\|_F^2
\end{aligned}
$$

$$
\begin{aligned}
&= \rho^W \frac{2L}{\alpha}\frac{h^2}{\alpha}\|R_e\|\sum_{t=0}^{W-1}(T-t)\|P(t)\|_F^2 + \zeta\|R_e\|\sum_{t=0}^{W-1}\frac{\rho^t-\rho^W}{1-\rho}(T-t)\|P(t)\|_F^2 \\
&\le \zeta\sum_{t=0}^{W-1}\|R_e\|(T-t)\|P(t)\|_F^2\frac{\rho^t}{1-\rho}
\end{aligned}
$$

where we used (A.10) and $\zeta = \frac{h^2}{\alpha} + \frac{h^2}{2L}$ and $\rho = 1 - \frac{\alpha}{4L}$.

When $W > T$, we have the following formula for the regret bound $R(W)$.

$$
\begin{aligned}
R(W) =& \rho^W \frac{2L}{\alpha} C_1 T\log(2) + \frac{2h^2L}{\alpha^2}\rho^W\|\boldsymbol{\delta}(T)\|^2 + \zeta\sum_{k=1}^{T}\rho^{k-1}\|\boldsymbol{\delta}(k)\|^2 + \zeta\|\boldsymbol{\delta}(T)\|^2\frac{\rho^T-\rho^W}{1-\rho} \\
=& \rho^W\frac{2L}{\alpha}C_1T\log(2) \\
&+ \boldsymbol{u}^\top\underbrace{\left(\left(\frac{2h^2L}{\alpha^2}\rho^W + \zeta\frac{\rho^T-\rho^W}{1-\rho}\right)\boldsymbol{R}_e^{1/2}\boldsymbol{M}_T^\top\boldsymbol{M}_T\boldsymbol{R}_e^{1/2} + \zeta\sum_{k=1}^{T}\rho^{k-1}\boldsymbol{R}_e^{1/2}\boldsymbol{M}_k^\top\boldsymbol{M}_k\boldsymbol{R}_e^{1/2}\right)}_{\boldsymbol{A}_W}\boldsymbol{u}
\end{aligned}
$$

Similarly, we bound $\|\boldsymbol{A}_W\|_F$ below.

$$
\begin{aligned}
\|\boldsymbol{A}_W\|_F &\le \left(\frac{2h^2L}{\alpha^2}\rho^W + \zeta\frac{\rho^T-\rho^W}{1-\rho}\right)\|\boldsymbol{R}_e^{1/2}\boldsymbol{M}_T^\top\boldsymbol{M}_T\boldsymbol{R}_e^{1/2}\|_F + \zeta\sum_{k=1}^{T}\rho^{k-1}\|\boldsymbol{R}_e^{1/2}\boldsymbol{M}_k^\top\boldsymbol{M}_k\boldsymbol{R}_e^{1/2}\|_F \\
&\le \left(\frac{2h^2L}{\alpha^2}\rho^W + \zeta\frac{\rho^T-\rho^W}{1-\rho}\right)\|R_e\|\|\boldsymbol{M}_T\|_F^2 + \zeta\sum_{k=1}^{T}\rho^{k-1}\|R_e\|\|\boldsymbol{M}_k\|_F^2 \\
&\le \left(\frac{2h^2L}{\alpha^2}\rho^W + \zeta\frac{\rho^T-\rho^W}{1-\rho}\right)\|R_e\|\sum_{t=0}^{T-1}(T-t)\|P(t)\|_F^2 + \zeta\sum_{k=1}^{T}\rho^{k-1}\|R_e\|\sum_{t=0}^{k-1}(T-t)\|P(t)\|_F^2 \\
&\le \left(\frac{2h^2L}{\alpha^2}\rho^W + \zeta\frac{\rho^T-\rho^W}{1-\rho}\right)\|R_e\|\sum_{t=0}^{T-1}(T-t)\|P(t)\|_F^2 + \zeta\|R_e\|\sum_{t=0}^{T-1}\frac{\rho^t-\rho^T}{1-\rho}(T-t)\|P(t)\|_F^2 \\
&\le \zeta\sum_{t=0}^{T-1}\|R_e\|(T-t)\|P(t)\|_F^2\frac{\rho^t}{1-\rho}
\end{aligned}
$$

In conclusion, for any $W \ge 1$, we have that $R(W) = \rho^W\frac{2L}{\alpha}C_1T\log(2) + \boldsymbol{u}^\top\boldsymbol{A}_W\boldsymbol{u}$, and $\|\boldsymbol{A}_W\|_F \le \zeta\sum_{t=0}^{\min(W,T)-1}\|R_e\|(T-t)\|P(t)\|_F^2\frac{\rho^t}{1-\rho}$. Further, we have $\|\boldsymbol{A}_W\| \le \|\boldsymbol{A}_W\|_F$. Therefore, by Proposition A.1, we prove the concentration bound below. For any $b > 0$,

$$
\mathbb{P}(\text{Regret}^d(RHIG) \ge \mathbb{E}[\text{Regbdd}] + b) \le \mathbb{P}(R(W) \ge \mathbb{E}[\text{Regbdd}] + b)
$$

$$\begin{aligned}
&\leq \mathbb{P}(R(W) \geq \mathbb{E}[R(W)] + b) \\
&= \mathbb{P}(\boldsymbol{u}^\top \boldsymbol{A}_W \boldsymbol{u} \geq \mathbb{E}[\boldsymbol{u}^\top \boldsymbol{A}_W \boldsymbol{u}] + b) \\
&\leq \exp\left(-c \min(\frac{b^2}{K^2}, \frac{b}{K})\right)
\end{aligned}$$

where $K = \zeta \sum_{t=0}^{\min(T,W)-1} \|R_e\|(T-t)\|P(t)\|_F^2 \frac{\rho^t}{1-\rho}$.

A.2 Proofs for Chapter 3

A.2.1 Proof of Lemma 3.1

Property ii) and iii) can be directly verified by definition. Thus, it suffices to prove i): the strong convexity and smoothness of $C(\boldsymbol{z})$.

Notice that $x_t,\ u_t$ are linear with respect to $\boldsymbol{z}$ by (3.5) (3.6). For ease of reference, we define matrix M^{x_t}, M^{u_t} to represent the relation between x_t, u_t and $\boldsymbol{z}$, i.e, $x_t = M^{x_t}\boldsymbol{z}$ and $u_t = M^{u_t}\boldsymbol{z}$. Similarly, we write $\tilde{f}_t(z_{t-p+1}, \ldots, z_t)$ and $\tilde{g}_t(z_{t-p+1}, \ldots, z_{t+1})$ in terms of $\boldsymbol{z}$ for simplicity of notation:

$$\begin{aligned}
\tilde{f}_t(z_{t-p+1}, \ldots, z_t) = \tilde{f}_t(\boldsymbol{z}) = f_t(M^{x_t}\boldsymbol{z}) \\
\tilde{g}_t(z_{t-p+1}, \ldots, z_{t+1}) = \tilde{g}_t(\boldsymbol{z}) = g_t(M^{u_t}\boldsymbol{z})
\end{aligned}$$

A direct consequence of the linear relations is that $\tilde{f}_t(\boldsymbol{z})$ and $\tilde{g}_t(\boldsymbol{z})$ are convex with respect to $\boldsymbol{z}$ because $f_t(x_t), g_t(u_t)$ are convex and the linear transformation preserves convexity.

In the following, we will focus on the proof of strong convexity and smoothness. For simplicity, in the following, we only consider cost function f_t, g_t with minimum values zero: $f_t(\theta_t) = 0$, and $g_t(\xi_t) = 0$ for all t. This is without loss of generality because by

strong convexity and smoothness, f_t, g_t have minimum values, and by subtracting the minimum value, we can let f_t, g_t have minimum value 0.

A.2.1.0.1 Strong convexity. Since $\tilde{g}_t$ is convex, we only need to prove that $\sum_t \tilde{f}_t(\boldsymbol{z})$ is strongly convex then the sum $C(\boldsymbol{z})$ is strongly convex because the sum of convex functions and a strongly convex function is strongly convex.

In particular, by the strong convexity of $f_t(x_t)$, we have the following result: for any $\boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{Nm}$ and $x_t = M^{x_t}\boldsymbol{z}$, $x_t' = M^{x_t}\boldsymbol{z}'$:

$$\begin{aligned}
&\tilde{f}_t(\boldsymbol{z}') - \tilde{f}_t(\boldsymbol{z}) - \langle \nabla \tilde{f}_t(\boldsymbol{z}), \boldsymbol{z}' - \boldsymbol{z}\rangle - \frac{\mu_f}{2}\|z_t' - z_t\|^2 \\
&\quad = \tilde{f}_t(\boldsymbol{z}') - \tilde{f}_t(\boldsymbol{z}) - \langle (M^{x_t})^\top \nabla f_t(x_t), \boldsymbol{z}' - \boldsymbol{z}\rangle - \frac{\mu_f}{2}\|z_t' - z_t\|^2 \\
&\quad = \tilde{f}_t(\boldsymbol{z}') - \tilde{f}_t(\boldsymbol{z}) - \langle \nabla f_t(x_t), M^{x_t}(\boldsymbol{z}' - \boldsymbol{z})\rangle - \frac{\mu_f}{2}\|z_t' - z_t\|^2 \\
&\quad = \tilde{f}_t(\boldsymbol{z}') - \tilde{f}_t(\boldsymbol{z}) - \langle \nabla f_t(x_t), x_t' - x_t\rangle - \frac{\mu_f}{2}\|z_t' - z_t\|^2 \\
&\quad \geq f_t(x_t') - f_t(x_t) - \langle \nabla f_t(x_t), x_t' - x_t\rangle - \frac{\mu_f}{2}\|x_t' - x_t\|^2 \geq 0
\end{aligned}$$

where the first equality is by the chain rule, the second equality is by the definition of inner product, the third equality is by the definition of x_t, x_t', the first inequality is by $\tilde{f}_t(z) = f_t(x)$ and $z_t = (x_t^{k_1}, \ldots, x_t^{k_m})^\top$, and the last inequality is because $f_t(x_t)$ is μ_f strongly convex.

Summing over t on both sides of the inequality results in the strong convexity of $\sum_t \tilde{f}_t(\boldsymbol{z})$:

$$\begin{aligned}
&\sum_{t=1}^{N}\left[\tilde{f}_t(\boldsymbol{z}') - \tilde{f}_t(\boldsymbol{z}) - \langle \nabla \tilde{f}_t(\boldsymbol{z}), \boldsymbol{z}' - \boldsymbol{z}\rangle - \frac{\mu_f}{2}\|z_t' - z_t\|^2\right] \\
=&\sum_{t=1}^{N}\tilde{f}_t(\boldsymbol{z}') - \sum_{t=1}^{N}\tilde{f}_t(\boldsymbol{z}) - \langle \nabla \sum_{t=1}^{N}\tilde{f}_t(\boldsymbol{z}), \boldsymbol{z}' - \boldsymbol{z}\rangle - \frac{\mu_f}{2}\|\boldsymbol{z}' - \boldsymbol{z}\|^2 \geq 0
\end{aligned}$$

Consequently, $C(\boldsymbol{z})$ is strongly convex with parameter at least μ_f by the convexity of $\tilde{g}_t$.

A.2.1.0.2 Smoothness. We will prove the smoothness by considering $\tilde{f}_t(\boldsymbol{z})$ and $\tilde{g}_t(\boldsymbol{z})$ respectively.

Firstly, let's consider $\tilde{f}_t(\boldsymbol{z})$. Similar to the proof for strong convexity, we use the smoothness of $f_t(x_t)$. For any $\boldsymbol{z}, \boldsymbol{z}'$, and $x_t = M^{x_t}\boldsymbol{z}$, $x'_t = M^{x_t}\boldsymbol{z}'$, we can show that

$$\begin{aligned}\tilde{f}_t(\boldsymbol{z}') = f_t(x'_t) &\leq f_t(x_t) + \langle \nabla f_t(x_t), x'_t - x_t\rangle + \frac{l_f}{2}\|x'_t - x_t\|^2 \\ &\leq \tilde{f}_t(\boldsymbol{z}) + \langle \nabla \tilde{f}_t(\boldsymbol{z}), \boldsymbol{z}' - \boldsymbol{z}\rangle + \frac{l_f}{2}(\|z'_{t-p+1} - z_{t-p+1}\|^2 + \cdots + \|z'_t - z_t\|^2)\end{aligned}$$

where the second inequality is by $x_t = M^{x_t}\boldsymbol{z}$ and the chain rule and (3.5).

Secondly, we consider $\tilde{g}_t(z)$ in a similar way. For any $\boldsymbol{z}, \boldsymbol{z}'$, and $u_t = M^{u_t}\boldsymbol{z}$, $u'_t = M^{u_t}\boldsymbol{z}'$, we have

$$\begin{aligned}\tilde{g}_t(\boldsymbol{z}') = g_t(u'_t) &\leq g_t(u_t) + \langle \nabla g_t(u_t), u'_t - u_t\rangle + \frac{l_g}{2}\|u'_t - u_t\|^2 \\ &= \tilde{g}_t(\boldsymbol{z}) + \langle (M^{u_t})^\top \nabla g_t(u_t), \boldsymbol{z}' - \boldsymbol{z}\rangle + \frac{l_g}{2}\|u'_t - u_t\|^2 \\ &= \tilde{g}_t(\boldsymbol{z}) + \langle \nabla \tilde{g}_t(\boldsymbol{z}), \boldsymbol{z} - \boldsymbol{z}\rangle + \frac{l_g}{2}\|u'_t - u_t\|^2\end{aligned}$$

Since $u_t = z_{t+1} - A(\mathscr{I}, :)x_t = [I_m, -A(\mathscr{I}, :)](z_{t+1}^\top, x_t^\top)^\top$, we have that

$$\begin{aligned}\frac{l_g}{2}\|u'_t - u_t\|^2 &\leq \frac{l_g}{2}\|[I_m, -A(\mathscr{I}, :)]\left[((z'_{t+1})^\top, (x'_t)^\top)^\top - (z_{t+1}^\top, x_t^\top)^\top\right]\|^2 \\ &\leq \frac{l_g}{2}\|[I_m, -A(\mathscr{I}, :)]\|^2(\|z_{t+1} - z'_{t+1}\|^2 + \|x_t - x'_t\|^2) \\ &\leq \frac{l_g}{2}\|[I_m, -A(\mathscr{I}, :)]\|^2(\|z_{t+1} - z'_{t+1}\|^2 + \cdots + \|z_{t-p+1} - z'_{t-p+1}\|^2)\end{aligned}$$

Finally, by summing $\tilde{f}_t(\boldsymbol{z}'), \tilde{g}_t(\boldsymbol{z}')$'s inequalities above over all t, we have

$$C(\boldsymbol{z}') \leq C(\boldsymbol{z}) + \langle \nabla C(\boldsymbol{z}), \boldsymbol{z}' - \boldsymbol{z}\rangle + (pl_f + (p+1)l_g\|[I_m, -A(\mathscr{I}, :)]\|^2)/2\|\boldsymbol{z}' - \boldsymbol{z}\|^2$$

Thus, we have proved the smoothness of $C(\boldsymbol{z})$.

A.2.2 Proof of Theorem 3.2

Remember that Regret$(FOSS) = J(FOSS) - J^*$. To bound the regret, we let the sum of the optimal steady state costs, $\sum_{t=0}^{N-1} \lambda_t^e$, be a middle ground and bound $J(FOSS) - \sum_{t=0}^{N-1} \lambda_t^e$ and $\sum_{t=0}^{N-1} \lambda_t^e - J^*$ in Lemma A.11 and Lemma A.12 respectively. Then, the regret bound can be obtained by combining the two bounds.

Lemma A.11 (Bound on $J(FOSS) - \sum_{t=0}^{N-1} \lambda_t^e$)**.** *Let $x_t(0)$ denote the state determined by FOSS.*

$$J(FOSS) - \sum_{t=0}^{N-1} \lambda_t^e \leq c_1 \sum_{t=0}^{N-1} \|x_t^e - x_{t-1}^e\| + f_N(x_N(0)) = O\left(\sum_{t=0}^{N} \|x_t^e - x_{t-1}^e\|\right)$$

where we define $x_N^e := \delta_N$, $x_{-1}^e := x_0 = 0$ for simplicity of notation, c_1 is a constant that does not depend on N, W and big O hides a constant that does not depend on N, W.

Lemma A.12 (Bound on $\sum_{t=0}^{N-1} \lambda_t^e - J^*$)**.** *Let $h_t^e(x)$ denote a solution to the Bellman equations under cost $f_t(x) + g_t(u)$. Let $\{x_t^*\}$ denote the optimal state trajectory to the offline optimal control (3.1).*

$$\sum_{t=0}^{N-1} \lambda_t^e - J^* \leq \sum_{t=1}^{N} (h_{t-1}^e(x_t^*) - h_t^e(x_t^*)) - h_0^e(x_0) = \sum_{t=0}^{N} (h_{t-1}^e(x_t^*) - h_t^e(x_t^*))$$

where we define $h_N^e(x) := f_N(x)$, $h_{-1}^e(x) := 0$ and $x_0^* := x_0$ for simplicity of notation.

Then, we can complete the proof by applying Lemma A.11 and A.12:

$$\begin{aligned} J(FOSS) - J^* &= J(FOSS) - \sum_{t=0}^{N-1} \lambda_t^e + \sum_{t=0}^{N-1} \lambda_t^e - J^* \\ &= O\left(\sum_{t=0}^{N} (\|x_{t-1}^e - x_t^e\| + h_{t-1}^e(x_t^*) - h_t^e(x_t^*))\right) \end{aligned}$$

In the following, we will prove Lemma A.11 and A.12 respectively. For simplicity, we only consider cost function f_t, g_t with minimum values zero: $f_t(\theta_t) = 0$, and $g_t(\xi_t) = 0$

for all t. There is no loss of generality because by strong convexity and smoothness, f_t, g_t have minimum values, and by subtracting the minimum value, we can let f_t, g_t have minimum value 0.

A.2.2.1 Proof of Lemma A.11.

Notice that $J(FOSS) = \sum_{t=0}^{N-1}(f_t(x_t(0)) + g_t(u_t(0))) + f_N(x_N(0))$ and $\sum_{t=0}^{N-1}\lambda_t^e = \sum_{t=0}^{N-1}(f_t(x_t^e) + g_t(u_t^e))$. Thus, it suffices to bound $f_t(x_t(0)) - f_t(x_t^e)$ and $g_t(u_t(0)) - g_t(u_t^e)$ for $0 \le t \le N-1$. We will first focus on $f_t(x_t(0)) - f_t(x_t^e)$, then bound $g_t(u_t(0)) - g_t(u_t^e)$ in the same way.

For $0 \le t \le N-1$, by the convexity of f_t, and the property of L_2 norm,

$$f_t(x_t(0)) - f_t(x_t^e) \le \langle \nabla f_t(x_t(0)), x_t(0) - x_t^e \rangle \le \|\nabla f_t(x_t(0))\| \|x_t(0) - x_t^e\| \tag{A.11}$$

In the following, we will bound $\|\nabla f_t(x_t(0))\|$ and $\|x_t(0) - x_t^e\|$ respectively.

Firstly, we provide a bound on $\|\nabla f_t(x_t(0))\|$:

$$\|\nabla f_t(x_t(0))\| = \|\nabla f_t(x_t(0)) - \nabla f_t(\theta_t)\| \le l_f \|x_t(0) - \theta_t\| \le l_f(\sqrt{n}\bar{x}^e + \bar{\theta}) \tag{A.12}$$

where the first equality is because θ_t is the global minimizer of f_t, and first inequality is by Lipschitz smoothness, the second inequality is by $\|\theta_t\| \le \bar{\theta}$ according to Assumption 3.3 and by $\|x_t(0)\| \le \sqrt{n}\bar{x}^e\|$ proved in the following lemma.

Lemma A.13 (Uniform upper bounds on $x_t^e, u_t^e, x_t(0), u_t(0)$)**.** *There exist $\bar{x}^e$ and $\bar{u}^e$ that are independent of N, W, such that $\|x_t^e\| \le \bar{x}^e$ and $\|u_t^e\| \le \bar{u}^e$ for all $0 \le t \le N-1$. Moreover, $\|x_t(0)\| \le \sqrt{n}\bar{x}^e$ for $0 \le t \le N$ and $\|u_t(0)\| \le \sqrt{n}\bar{u}^e$ for $0 \le t \le N-1$, where $x_t(0), u_t(0)$ denote the state and control at t determined by FOSS.*

The proof is technical and is deferred to Appendix A.2.2.3.

Secondly, we provide a bound on $\|x_t(0) - x_t^e\|$. The proof relies on the expressions of the steady state x_t^e and the initialized state $x_t(0)$ of a canonical-form system.

Lemma A.14 (The steady state and the initialized state of canonical-form systems)**.** *Consider a canonical-form system:* $x_{t+1} = Ax_t + Bu_t$.

(a) Any steady state (x, u) *is in the form of*

$$x = (\underbrace{z^1, \dots, z^1}_{p_1}, \underbrace{z^2, \dots, z^2}_{p_2}, \dots, \underbrace{z^m, \dots, z^m}_{p_m})^\top$$
$$u = (z^1, \dots, z^m)^\top - A(\mathscr{I}, :)x$$

for some $z^1, \dots, z^m \in \mathbb{R}$. *Let* $z = (z^1, \dots, z^m)^\top$. *For the optimal steady state with respect to cost* $f_t + g_t$, *we denote the corresponding* z *as* z_t^e, *and the optimal steady state can be represented as* $x_t^e = (z_t^{e,1}, \dots, z_t^{e,1}, z_t^{e,2}, \dots, z_t^{e,2}, \dots, z_t^{e,m}, \dots, z_t^{e,m})^\top$ *and* $u_t^e = z_t^e - A(\mathscr{I}, :)x_t^e$ *for* $0 \le t \le N-1$.

(b) By FOSS initialization, $z_{t+1}(0) = z_t^e$, *and* $x_t(0)$, $u_t(0)$ *satisfy*

$$x_t(0) = (\underbrace{z_{t-p_1}^{e,1}, \dots, z_{t-1}^{e,1}}_{p_1}, \underbrace{z_{t-p_2}^{e,2}, \dots, z_{t-1}^{e,2}}_{p_2}, \dots, \underbrace{z_{t-p_m}^{e,m}, \dots, z_{t-1}^{e,m}}_{p_m}), \qquad 0 \le t \le N$$
$$u_t(0) = z_t^e - A(\mathscr{I}, :)x_t(0) \qquad 0 \le t \le N-1$$

where $z_t^e = 0$ *for* $t \le -1$.

Proof. (a) This is by the definition of the canonical form and the definition of the steady state.

(b) By the initialization, $z_t(0) = x_{t-1}^{e,\mathscr{I}} = z_{t-1}^e$. By the relation between $z_t(0)$ and $x_t(0)$, $u_t(0)$, we have $x_t^{\mathscr{I}}(0) = z_t(0) = z_{t-1}^e$, and $x_t^{\mathscr{I}^{-1}}(0) = z_{t-1}(0) = z_{t-2}^e$, so on and so forth. This proves the structure of $x_t(0)$. The structure of $u_t(0)$ is because $u_t(0) = z_{t+1}(0) - A(\mathscr{I},:)x_t(0) = z_t^e - A(\mathscr{I},:)x_t(0)$

□

By Lemma A.14, we can bound $\|x_t(0) - x_t^e\|$ for $0 \le t \le N-1$ by

$$\begin{aligned}\|x_t(0) - x_t^e\| &\le \sqrt{\|z_{t-1}^e - z_t^e\|^2 + \cdots + \|z_{t-p}^e - z_t^e\|^2} \\ &\le \sqrt{\|x_{t-1}^e - x_t^e\|^2 + \cdots + \|x_{t-p}^e - x_t^e\|^2} \\ &\le \|x_{t-1}^e - x_t^e\| + \cdots + \|x_{t-p}^e - x_t^e\| \\ &\le p(\|x_{t-1}^e - x_t^e\| + \cdots + \|x_{t-p}^e - x_{t-p+1}^e\|) \end{aligned} \tag{A.13}$$

Combining (A.11) (A.12) and (A.13) yields

$$\begin{aligned}\sum_{t=0}^{N-1} f_t(x_t(0)) - f_t(x_t^e) &\le \sum_{t=0}^{N-1} \|\nabla f_t(x_t(0))\| \|x_t(0) - x_t^e\| \\ &\le \sum_{t=0}^{N-1} l_f(\sqrt{n}\bar{x}^e + \bar{\theta}) p(\|x_{t-1}^e - x_t^e\| + \cdots + \|x_{t-p}^e - x_{t-p+1}^e\|) \\ &\le p^2 l_f(\sqrt{n}\bar{x}^e + \bar{\theta}) \sum_{t=0}^{N-1} \|x_{t-1}^e - x_t^e\| \end{aligned} \tag{A.14}$$

Notice that the constant term $p^2 l_f(\sqrt{n}\bar{x}^e + \bar{\theta})$ does not depend on N, W.

Similarly, we can provide a bound on $g_t(u_t(0)) - g_t(u_t^e)$.

$$\begin{aligned}\sum_{t=0}^{N-1} g_t(u_t(0)) - g_t(u_t^e) &\le \sum_{t=0}^{N-1} \|\nabla g_t(u_t(0))\| \|u_t(0) - u_t^e\| \\ &\le \sum_{t=0}^{N-1} l_g \|u_t(0) - \xi_t\| \|u_t(0) - u_t^e\| \end{aligned}$$

$$\leq \sum_{t=0}^{N-1} l_g(\sqrt{n}\bar{u}^e + \bar{\xi})\|A(\mathscr{I},:)x_t(0) - A(\mathscr{I},:)x_t^e\|$$
$$\leq \sum_{t=0}^{N-1} l_g(\sqrt{n}\bar{u}^e + \bar{\xi})\|A(\mathscr{I},:)\|\|x_t(0) - x_t^e\|$$
$$\leq p^2 l_g(\sqrt{n}\bar{u}^e + \bar{\xi})\|A(\mathscr{I},:)\| \sum_{t=0}^{N-1} \|x_{t-1}^e - x_t^e\| \tag{A.15}$$

where the first inequality is by the convexity, the second inequality is because ξ_t is the global minimizer of g_t and g_t is l_g-smooth, the third inequality is by Assumption 3.3, Lemma A.13 and Lemma A.14, the fifth inequality is by (A.13). Notice that the constant term $p^2 l_g(\sqrt{n}\bar{u}^e + \bar{\xi})\|A(\mathscr{I},:)\|$ does not depend on N, W.

By (A.14) and (A.15), we complete the proof of the first inequality in the statement of Lemma A.11:

$$J(FOSS) - \sum_{t=0}^{N-1} \lambda_t^e \leq c_1 \sum_{t=0}^{N-1} \|x_{t-1}^e - x_t^e\| + f_N(x_N(0))$$

where c_1 does not depend on N, W.

By defining $x_N^e = \theta_N$, we can bound $f_N(x_N(0))$ by $\|x_N(0) - x_N^e\|$ up to some constants because $f_N(x_N(0)) = f_N(x_N(0)) - f_N(\theta_N) \leq \frac{l_f}{2}(\sqrt{n}\bar{x}^e + \bar{\theta})\|x_N(0) - x_N^e\|$. By the same argument as in (A.13), we have $\|x_N(0) - x_N^e\| = O(\sum_{t=0}^{N} \|x_{t-1}^e - x_t^e\|)$, where the big O hides some constant that does not depend on N, W. Consequently,

$$J(FOSS) - \sum_{t=0}^{N-1} \lambda_t^e = O\left(\sum_{t=0}^{N} \|x_{t-1}^e - x_t^e\|\right)$$

□

A.2.2.2 Proof of Lemma A.12.

The proof heavily relies on dynamic programming and the Bellman equations. For simplicity, we introduce a Bellman operator $\mathbb{B}(f+g,h)$ defined by $\mathbb{B}(f+g,h)(x) = \min_u(f(x)+g(u)+h(Ax+Bu))$. Now the Bellman equations can be written as $\mathbb{B}(f+g,h^e)(x) = h^e(x)+\lambda^e$ for any x.

We define a sequence of auxiliary functions S_k: $S_k(x) = h_k^e(x) + \sum_{t=k}^{N-1}\lambda_t^e$ for $k = 0,\ldots,N$, where $h_N^e(x) = f_N(x)$.

We first provide a recursive equation for S_k. By Bellman equations, we have $h_k^e(x)+\lambda_k^e = \mathbb{B}(f_k+g_k,h_k^e)(x)$ for $0 \le k \le N-1$. Let π_k^e be the corresponding optimal control policy that solves the Bellman equations. We have the following recursive relation for S_k when $0 \le k \le N-1$:

$$S_k(x) = \mathbb{B}(f_k+g_k, S_{k+1}-h_{k+1}^e+h_k^e)(x)$$

where $S_N(x) = f_N(x)$.

Further, let $V_k(x)$ denote the optimal cost-to-go function from k to N, then we obtain a recursive equation for V_k by dynamic programming:

$$V_k(x) = \mathbb{B}(f_k+g_k,V_{k+1})(x) = f_k(x)+g_k(\pi_k^*(x))+V_{k+1}(Ax+B\pi_k^*(x))$$

where $0 \le k \le N-1$, and π_k^* denotes the optimal control policy and $V_N(x) = f_N(x)$.

Now, we are ready for a recursive inequality for $S_k(x_k^*) - V_k(x_k^*)$. Let $\{x_k^*\}$ denote the optimal trajectory, then $x_{k+1}^* = Ax_k^* + B\pi_k^*(x_k^*)$. For any $k = 0,\ldots,N-1$,

$$S_k(x_k^*) - V_k(x_k^*) = \mathbb{B}(f_k+g_k, S_{k+1}-h_{k+1}^e+h_k^e)(x_k^*) - \mathbb{B}(f_k+g_k,V_{k+1})(x_k^*)$$

$$\leq f_k(x_k^*) + g_k(\pi_k^*(x_k^*)) + S_{k+1}(x_{k+1}^*) - h_{k+1}^e(x_{k+1}^*) + h_k^e(x_{k+1}^*)$$
$$- (f_k(x_k^*) + g_k(\pi_k^*(x_k^*)) + V_{k+1}(x_{k+1}^*))$$
$$= S_{k+1}(x_{k+1}^*) - h_{k+1}^e(x_{k+1}^*) + h_k^e(x_{k+1}^*) - V_{k+1}(x_{k+1}^*)$$

where the first inequality is because π_k^* is not optimal for the Bellman operator $\mathbb{B}(f_k + g_k, S_{k+1} - h_{k+1}^e + h_k^e)(x_k^*)$.

Summing over $k = 0, \ldots, N-1$ the recursive inequality for $S_k(x_k^*) - V_k(x_k^*)$ yields

$$S_0(x_0) - V_0(x_0) \leq \sum_{k=0}^{N-1} (h_k^e(x_{k+1}^*) - h_{k+1}^e(x_{k+1}^*))$$

By subtracting $h_0^e(x_0)$ on both sides,

$$\sum_{t=0}^{N-1} \lambda_t^e - J^* \leq \sum_{k=0}^{N-1} (h_k^e(x_{k+1}^*) - h_{k+1}^e(x_{k+1}^*)) - h_0^e(x_0)$$

For the simplicity of notation, we define $h_{-1}^e(x_0) = 0$ and $x_0^* = x_0$, then the bound can be written as

$$\sum_{t=0}^{N-1} \lambda_t^e - J^* \leq \sum_{k=0}^{N} (h_{k-1}^e(x_k^*) - h_k^e(x_k^*))$$

□

A.2.2.3 Proof of Lemma A.13

The proof relies on the (strong) convexity and smoothness of the cost functions and the uniform upper bounds on θ_t, ξ_t.

First of all, suppose there exists $\bar{x}^e$ such that $\|x_t^e\|_2 \leq \bar{x}^e$ for all $0 \leq t \leq N-1$. We will bound $u_t^e, x_t(0), u_t(0)$ by using $\bar{x}^e$. Notice that the optimal steady state and the

corresponding steady control satisfy: $u_t^e = x_t^{e,\mathscr{I}} - A(\mathscr{I},:)x_t^e$. If we can bound x_t^e by $\|x_t^e\| \le \bar{x}^e$ for all t, u_t^e can be bounded accordingly:

$$\|u_t^e\| \le \|x_t^{e,\mathscr{I}}\| + \|A(\mathscr{I},:)x_t^e\| \le \|x_t^e\| + \|A(\mathscr{I},:)\|\|x_t^e\| \le (1 + \|A(\mathscr{I},:)\|)\bar{x}^e =: \bar{u}^e$$

Moreover, $x_t(0)$ can also be bounded by $\bar{x}^e$ multiplied by some factors, because by Lemma A.14, $x_t(0)$'s each entry is determined by some entry of x_s^e for $s < t$. As a result, for $0 \le t \le N$

$$\|x_t(0)\|_2 \le \sqrt{n}\|x_t(0)\|_\infty \le \sqrt{n}\max_{s<t}\|x_s^e\|_\infty \le \sqrt{n}\max_{s<t}\|x_s^e\|_2 \le \sqrt{n}\bar{x}^e$$

We can bound $u_t(0)$ by noticing that $u_t(0) = x_{t+1}^{\mathscr{I}}(0) - A(\mathscr{I},:)x_t(0)$ and

$$\begin{aligned}\|u_t(0)\| &\le \|x_{t+1}^{\mathscr{I}}(0)\| + \|A(\mathscr{I},:)x_t(0)\| \le \|x_{t+1}(0)\| + \|A(\mathscr{I},:)\|\|x_t(0)\| \\ &\le (1 + \|A(\mathscr{I},:)\|)\sqrt{n}\bar{x}^e = \sqrt{n}\bar{u}^e\end{aligned}$$

Next, it suffices to prove $\|x_t^e\| \le \bar{x}^e$ for all t for some $\bar{x}^e$. To prove this bound, we construct another (suboptimal) steady state: $\hat{x}_t = (\theta_t^1, \ldots, \theta_t^1)$. Let $\hat{u}_t = \hat{x}_t^{\mathscr{I}} - A(\mathscr{I},:)\hat{x}_t$. It can be easily verified that $(\hat{x}_t, \hat{u}_t)$ is indeed a steady state of the canonical-form system. Moreover, $\hat{x}_t$ and $\hat{u}_t$ can be bounded similarly as follows.

$$\begin{aligned}\|\hat{x}_t\| &\le \sqrt{n}|\theta_t^1| \le \sqrt{n}\|\theta_t\|_\infty \le \sqrt{n}\|\theta_t\| \le \sqrt{n}\bar{\theta} \\ \|\hat{u}_t\|_2 &\le (1 + \|A(\mathscr{I},:)\|)\|\hat{x}_t\| \le (1 + \|A(\mathscr{I},:)\|)\sqrt{n}\bar{\theta}\end{aligned}$$

Now, we can bound $\|x_t^e - \theta_t\|$.

$$\begin{aligned}\frac{\mu}{2}\|x_t^e - \theta_t\|^2 &\le f_t(x_t^e) - f_t(\theta_t) + g_t(u_t^e) - g_t(\xi_t) \\ &\le f_t(\hat{x}_t) - f_t(\theta_t) + g_t(\hat{u}_t) - g_t(\xi_t) \\ &\le \frac{l_f}{2}\|\hat{x}_t - \theta_t\|^2 + \frac{l_g}{2}\|\hat{u}_t - \xi_t\|^2\end{aligned}$$

$$\leq l_f(\|\hat{x}_t\|^2 + \|\theta_t\|^2) + l_g(\|\hat{u}_t\|^2 + \|\xi_t\|^2)$$
$$\leq l_f(n\bar{\theta}^2 + \bar{\theta}^2) + l_g(((1 + \|A(\mathscr{I}, :)\|)\sqrt{n}\bar{\theta})^2 + \bar{\xi}) =: c_5$$

where the first inequality is by f_t's strong convexity and g_t's convexity, the second inequality is because (x_t^e, u_t^e) is an optimal steady state, the third inequality is by the smoothness and $\nabla f_t(\theta_t) = \nabla g_t(\xi_t) = 0$, the last inequality is by the bounds of $\|\hat{x}_t\|, \|\hat{u}_t\|$, θ_t, and ξ_t.

As a result, we have $\|x_t^e - \theta_t\| \leq \sqrt{2c_5/\mu}$. Then, we can bound x_t^e by $\|x_t^e\| \leq \|\theta_t\| + \sqrt{2c_5/\mu} \leq \bar{\theta} + \sqrt{2c_5/\mu} =: \bar{x}^e$ for all t. It can be verified that $\bar{x}^e$ does not depend on N, W.

□

A.2.3 Linear quadratic tracking

In this section, we will provide a regret bound in Corollary A.1 for the general LQT defined in Example 3.1. Based on this, we prove Corollary 3.1, which is a special case when Q_t, R_t are not changing.

A.2.3.1 Regret bound on the general online LQT problems

Before the regret bound, we provide an important lemma to characterize the solution to the Bellman equations of the LQT problem.

Lemma A.15. *One solution to the Bellman equations with stage cost* $\frac{1}{2}(x-\theta)^\top Q(x-\theta) + \frac{1}{2}u^\top Ru$ *can be represented by*

$$h^e(x) = \frac{1}{2}(x - \beta^e)^\top P^e(x - \beta^e) \tag{A.16}$$

where P^e denotes the solution to the discrete-time algebraic Riccati equation (DARE) with respect to Q, R, A, B

$$P^e = Q + A^\top (P^e - P^e B(B^\top P^e B + R)^{-1} B^\top P^e) A \tag{A.17}$$

and $\beta^e = F\theta$ where F is a matrix determined by A, B, Q, R.

The proof is in [6].

For simplicity of notation, let $P^e(Q, R)$ denote the solution to the DARE under the parameters Q, R, A, B and $F(Q, R)$ denote the matrix in $\beta^e = F\theta$ given parameters Q, R, A, B. Here we omit A, B in the arguments of the functions because they will not change in the following.

In addition, we introduce the following useful notations: $\underline{Q} = \mu_f I_n, \bar{Q} = l_f I_n, \underline{R} = \mu_g I_m, \bar{R} = l_g I_m$ for $\mu_f, \mu_g > 0$, $0 < l_f, l_g < +\infty$; and $\bar{P} = P^e(\bar{Q}, \bar{R})$ and $\underline{P} = P^e(\underline{Q}, \underline{R})$. Based on the notations above, we define some sets of matrices to be used later:

$$\mathscr{Q} = \{Q \mid \underline{Q} \leq Q \leq \bar{Q}\},$$
$$\mathscr{R} = \{R \mid \underline{R} \leq R \leq \bar{R}\},$$
$$\mathscr{P} = \{P \mid \underline{P} \leq P \leq \bar{P}\}.$$

Now, we are ready for the regret bound for the general LQT problem.

Corollary A.1 (Bound on general LQT)**.** *Consider the LQT problem in Example 3.1. Suppose for $t = 0, 1, \ldots, N-1$, the cost matrices satisfy $Q_t \in \mathscr{Q}$, $R_t \in \mathscr{R}$. Suppose the terminal cost function satisfies $Q_N \in \mathscr{P}$.*[7] *Then, the regret of RHTM with initialization*

[7] This additional condition is for technical simplicity and can be removed.

FOSS can be bounded by

$$\text{Regret}(RHTM) = O\left(\zeta^2(\frac{\sqrt{\zeta}-1}{\sqrt{\zeta}})^{2K}\left(\sum_{t=1}^{N}(\|P_t^e - P_{t-1}^e\| + \|\beta_t^e - \beta_{t-1}^e\|) + \sum_{t=0}^{N}\|x_{t-1}^e - x_t^e\|\right)\right)$$

where $K = \lfloor (W-1)/p \rfloor$, $x_{-1}^e = x_0$, $x_N^e = \theta_N$, ζ *is the condition number of the corresponding* $C(\boldsymbol{z})$, (x_t^e, u_t^e) *is the optimal steady state under cost* Q_t, R_t, θ_t, $P_t^e = P^e(Q_t, R_t)$ *and* $\beta_t^e = F(Q_t, R_t)\theta_t$ *for* $t = 0, \ldots, N-1$ *and* $\beta_N^e = \theta_N$, $P_N^e = Q_N$.

Proof. Before the proof, we introduce some supportive lemmas on the uniform bounds of P_t^e, β_t^e, x_t^* respectively. The intuition behind these uniform bounds is that the cost function coefficients Q_t, R_t, θ_t are all uniformly bounded by Assumption 3.2 and 3.3. The proofs are technical and deferred to [6].

Lemma A.16 (Upper bound on x_t^*)**.** *For any* $Q_t \in \mathscr{Q}, R_t \in \mathscr{R}, Q_N \in \mathscr{P}$, *there exists* $\bar{x}$ *that does not depend on* t, N, W, *such that*

$$\|x_t^*\|_2 \le \bar{x}, \qquad \forall\ 0 \le t \le N.$$

Lemma A.17 (Upper bound on β^e)**.** *For any* $Q \in \mathscr{Q}, R \in \mathscr{R}$, *any* $\|\theta\| \le \bar{\theta}$, *there exists* $\bar{\beta} \ge 0$ *that does not depend on* N *and only depends on* $A, B, l_f, \mu_f, l_g, \mu_g, \bar{\theta}$, *such that* $\max(\bar{\theta}, \|\beta^e\|) \le \bar{\beta}$, *where* β^e *is defined in Lemma A.15.*

Lemma A.18 (Upper bound on P^e)**.** *For any* $Q \in \mathscr{Q}, R \in \mathscr{R}$, *we have* $P^e = P^e(Q, R) \in \mathscr{P}$. *Consequently,* $\|P^e\| \le \upsilon_{\max}(\bar{P})$, *where* $\upsilon_{\max}(\bar{P})$ *denotes the largest eigenvalue of* $\bar{P}$.

Now, we are ready for the proof of Corollary A.1.

By Theorem 3.2, we only need to bound $\sum_{t=0}^{N}(h_{t-1}^e(x_t^*) - h_t^e(x_t^*))$. By definition, $P_N^e = Q_N, \beta_N^e = \theta_N$, $h_N^e(x) = f_N(x)$, so we can write $h_t^e(x) = \frac{1}{2}(x - \beta_t^e)^\top P_t^e(x - \beta_t^e)$ for $0 \le t \le N$.

For $0 \leq t \leq N-1$, we split $h_t^e(x_{t+1}^*) - h_{t+1}^e(x_{t+1}^*)$ into two parts.

$$\begin{aligned} h_t^e(x_{t+1}^*) - h_{t+1}^e(x_{t+1}^*) &= \frac{1}{2}(x_{t+1}^* - \beta_t^e)^\top P_t^e (x_{t+1}^* - \beta_t^e) - \frac{1}{2}(x_{t+1}^* - \beta_{t+1}^e)^\top P_{t+1}^e (x_{t+1}^* - \beta_{t+1}^e) \\ &= \underbrace{\frac{1}{2}(x_{t+1}^* - \beta_t^e)^\top P_t^e (x_{t+1}^* - \beta_t^e) - \frac{1}{2}(x_{t+1}^* - \beta_{t+1}^e)^\top P_t^e (x_{t+1}^* - \beta_{t+1}^e)}_{\text{Part 1}} \\ &\quad + \underbrace{\frac{1}{2}(x_{t+1}^* - \beta_{t+1}^e)^\top P_t^e (x_{t+1}^* - \beta_{t+1}^e) - \frac{1}{2}(x_{t+1}^* - \beta_{t+1}^e)^\top P_{t+1}^e (x_{t+1}^* - \beta_{t+1}^e)}_{\text{Part 2}} \end{aligned}$$

Part 1 can be bounded by the following

$$\begin{aligned} \text{Part 1} &= \frac{1}{2}(x_{t+1}^* - \beta_t^e + x_{t+1}^* - \beta_{t+1}^e)^\top P_t^e (x_{t+1}^* - \beta_t^e - (x_{t+1}^* - \beta_{t+1}^e)) \\ &\leq \frac{1}{2}\|x_{t+1}^* - \beta_t^e + x_{t+1}^* - \beta_{t+1}^e\|_2 \|P_t^e\|_2 \|\beta_{t+1}^e - \beta_t^e\|_2 \\ &\leq (\bar{x} + \bar{\beta})\upsilon_{max}(\bar{P})\|\beta_{t+1}^e - \beta_t^e\|_2 \end{aligned}$$

where the last inequality is by Lemma A.16, A.17 A.18.

Part 2 can be bounded by the following when $0 \leq t \leq N-1$,

$$\begin{aligned} \text{Part 2} &= \frac{1}{2}(x_{t+1}^* - \beta_{t+1}^e)^\top (P_t^e - P_{t+1}^e)(x_{t+1}^* - \beta_{t+1}^e) \\ &\leq \frac{1}{2}\|x_{t+1}^* - \beta_{t+1}^e\|_2^2 \|P_t^e - P_{t+1}^e\|_2 \leq \frac{1}{2}(\bar{x} + \bar{\beta})^2 \|P_t^e - P_{t+1}^e\|_2 \end{aligned}$$

Therefore, we have

$$\begin{aligned} \sum_{t=0}^{N}(h_{t-1}^e(x_t^*) - h_t^e(x_t^*)) &\leq \sum_{t=0}^{N-1}(h_t^e(x_{t+1}^*) - h_{t+1}^e(x_{t+1}^*)) \\ &= O(\sum_{t=0}^{N-1}(\|\beta_{t+1}^e - \beta_t^e\|_2 + \|P_t^e - P_{t+1}^e\|_2)) \end{aligned} \tag{A.18}$$

where the first inequality is by $h_0^e(x) \geq 0$ and $h_{-1}^e(x) = 0$. Thus, by Theorem 3.2, we have

$$\text{Regret}(RHTM) = O\left(\zeta^2(\frac{\sqrt{\zeta}-1}{\sqrt{\zeta}})^{2K}\left(\sum_{t=1}^{N}\left(\|P_t^e - P_{t-1}^e\| + \|\beta_t^e - \beta_{t-1}^e\|\right) + \sum_{t=0}^{N}\|x_{t-1}^e - x_t^e\|\right)\right)$$

□

A.2.3.2 Proof of Corollary 3.1

Roughly speaking, the proof is mostly by applying Corollary A.1 and by showing $\|\beta_t^e - \beta_{t-1}^e\|$ and $\|x_t^e - x_{t-1}^e\|$ can be bounded by $\|\theta_t - \theta_{t-1}\|$ up to some constants and $\|P_t^e - P_{t-1}^e\| = 0$ in the LQT problem (3.9) where Q and R are not changing. However, directly applying the results in Theorem 3.2 and Corollary A.1 will result in some extra constant terms because some inequalities used to derive the bounds in Theorem 3.2 and Corollary A.1 are not necessary when Q, R are not changing. Therefore, we will need some intermediate results in the proofs of Theorem 3.2 and Corollary A.1 to prove Corollary 3.1.

Firstly, by Lemma A.11 and Lemma A.12, we have

$$\begin{aligned} J(FOSS) - J^* &= J(FOSS) - \sum_{t=0}^{N-1} \lambda_t^e + \sum_{t=0}^{N-1} \lambda_t^e - J^* \\ &\leq \underbrace{c_1 \sum_{t=0}^{N-1} \|x_{t-1}^e - x_t^e\|}_{\text{Part I}} + \underbrace{\sum_{t=0}^{N-1} (h_t^e(x_{t+1}^*) - h_{t+1}^e(x_{t+1}^*))}_{\text{Part II}} + \underbrace{f_N(x_N(0)) - h_0^e(x_0)}_{\text{Part III}} \end{aligned}$$

We are going to bound each part by $\sum_t \|\theta_t - \theta_{t-1}\|$ in the following.

Part I: We will bound Part I by $\sum_t \|\theta_t - \theta_{t-1}\|$ through showing that $x_t^e = F_1 F_2 \theta_t$ for some matrices F_1, F_2. The representation of x_t^e relies on Lemma A.14.

By Lemma A.14, any steady state (x, u) can be represented as a matrix multiplied by z:

$$x = (\underbrace{z^1, \dots, z^1}_{p_1}, \underbrace{z^2, \dots, z^2}_{p_2}, \dots, \underbrace{z^m, \dots, z^m}_{p_m})^\top =: F_1 z$$

$$u = (z^1, \dots, z^m)^\top - A(\mathscr{I}, :)x = (I_m - A(\mathscr{I}, :)F_1)z$$

where $F_1 \in \mathbb{R}^{n,m}$ is a binary matrix with full column rank.

Consider cost function $\frac{1}{2}(x-\theta)^\top Q(x-\theta)+\frac{1}{2}u^\top Ru$. By the steady-state representation above, the optimal steady state can be solved by the following unconstrained optimization:

$$\min_z (F_1 z-\theta)^\top Q(F_1 z-\theta)+z^\top(I-A(\mathscr{I},:)F_1)^\top R(I-A(\mathscr{I},:)F_1)z$$

Since F_1 is full column rank, the function is strongly convex and has the unique solution

$$z^e = F_2\theta \tag{A.19}$$

where $F_2 = (F_1^\top Q F_1 + (I-A(\mathscr{I},:)F_1)^\top R(I-A(\mathscr{I},:)F_1))^{-1}F_1^\top Q$. Accordingly, the optimal steady state can be represented as

$$x^e = F_1F_2\theta, \qquad u^e = (I_m - A(\mathscr{I},:)F_1)F_2\theta. \tag{A.20}$$

Consequently, when $1\le t\le N-1$,$\|x_t^e - x_{t-1}^e\| \le \|F_1F_2\|\|\theta_t-\theta_{t-1}\|$. When $t=0$, $\|x_0^e - x_{-1}^e\| \le \|F_1F_2\|\|\theta_0-\theta_{-1}\|$ holds since $x_{-1}^e = x_0 = \theta_{-1} = 0$. Combining the upper bounds above, we have

$$\text{Part I} = O\left(\sum_{t=0}^{N-1}\|\theta_t-\theta_{t-1}\|\right)$$

Part II: By (A.18) in the proof of Corollary A.1, and by noticing that $P_t^e = P^e(Q,R)$ does not change, we have

$$\sum_{t=0}^{N-1}(h_t^e(x_{t+1}^*) - h_{t+1}^e(x_{t+1}^*)) = O\left(\sum_{t=0}^{N-1}\|\beta_{t+1}^e-\beta_t^e\|\right)$$

By Lemma A.15, $\beta_t^e = F(Q,R)\theta_t$ for $0\le t\le N-1$. In addition, since $\beta_N^e = \theta_N = 0$ as defined in (3.9) and Corollary A.1, we can also write $\beta_N^e = F(Q,R)\theta_N$. Thus,

$$\text{Part II} = O\left(\sum_{t=0}^{N-1}\|\beta_{t+1}^e-\beta_t^e\|\right) = O\left(\sum_{t=1}^{N}\|\theta_t-\theta_{t-1}\|\right)$$

Part III: By our condition for the terminal cost function, we have $f_N(x_N(0)) = \frac{1}{2}(x_N(0)-\beta_N^e)^\top P^e(x_N(0)-\beta_N^e)$. By Lemma A.15, we have $h_0^e(x_0) = \frac{1}{2}(x_0-\beta_0^e)^\top P^e(x_0-\beta_0^e)$. So Part III can be bounded by

$$\begin{aligned}
\text{Part III} &= \frac{1}{2}(x_N(0)-\beta_N^e)^\top P^e(x_N(0)-\beta_N^e) - \frac{1}{2}(x_0-\beta_0^e)^\top P^e(x_0-\beta_0^e) \\
&= \frac{1}{2}(x_N(0)-\beta_N^e+x_0-\beta_0^e)^\top P^e(x_N(0)-\beta_N^e-(x_0-\beta_0^e)) \\
&\le \frac{1}{2}\|x_N(0)-\beta_N^e+x_0-\beta_0^e\|\|P^e\|\|x_N(0)-\beta_N^e-(x_0-\beta_0^e)\| \\
&\le \frac{1}{2}(\sqrt{n}\bar{x}^e+\bar{\beta}+\bar{\beta})\|P^e\|(\|x_N(0)-x_0\|+\|\beta_N^e-\beta_0^e\|)
\end{aligned}$$

where the last inequality is by $x_0 = 0$, Lemma A.13, Lemma A.17.

Next we will bound $\|x_N(0)-x_0\|$ and $\|\beta_N^e-\beta_0^e\|$ respectively. Firstly, by $\beta_t^e = F(Q,R)\theta_t$ in Lemma A.15, we have

$$\|\beta_N^e-\beta_0^e\| \le \sum_{t=0}^{N-1}\|\beta_{t+1}^e-\beta_t^e\| \le \|F(Q,R)\|\sum_{t=0}^{N-1}\|\theta_{t+1}-\theta_t\|$$

Secondly, we will bound $\|x_N(0)-x_0\|$.

$$\begin{aligned}
\|x_N(0)-x_0\| &\le \|x_N(0)-x_{N-1}^e\| + \|x_{N-1}^e-x_0\| \\
&\le \|x_N(0)-x_{N-1}^e\| + \sum_{t=0}^{N-1}\|x_t^e-x_{t-1}^e\| \\
&\le \|x_N(0)-x_{N-1}^e\| + \|F_1F_2\|\sum_{t=0}^{N-1}\|\theta_t-\theta_{t-1}\|
\end{aligned}$$

where the second inequality is by $x_0^e = x_0$, the third inequality is by (A.20).

Next, we will focus on $\|x_N(0)-x_{N-1}^e\|$. By Lemma A.14,

$$\begin{aligned}
x_N(0) &= (z_{N-p_1}^{e,1},\dots,z_{N-1}^{e,1},z_{N-p_2}^{e,2},\dots,z_{N-1}^{e,2},\dots,z_{N-p_m}^{e,m},\dots,z_{N-1}^{e,m})^\top \\
x_{N-1}^e &= (z_{N-1}^{e,1},\dots,z_{N-1}^{e,1},z_{N-1}^{e,2},\dots,z_{N-1}^{e,2},\dots,z_{N-1}^{e,m},\dots,z_{N-1}^{e,m})^\top
\end{aligned}$$

As a result,

$$\begin{aligned}\|x_N(0) - x^e_{N-1}\|^2 &\le \|z^e_{N-2} - z^e_{N-1}\|^2 + \cdots + \|z^e_{N-p} - z^e_{N-1}\|^2 \\ &= \|F_2\|^2(\|\theta_{N-2} - \theta_{N-1}\|^2 + \cdots + \|\theta_{N-p} - \theta_{N-1}\|^2)\end{aligned}$$

where the equality is by (A.19). Taking square root on both sides yields

$$\begin{aligned}\|x_N(0) - x^e_{N-1}\| &\le \|F_1\|\sqrt{\|\theta_{N-2} - \theta_{N-1}\|^2 + \cdots + \|\theta_{N-p} - \theta_{N-1}\|^2} \\ &\le \|F_2\|(\|\theta_{N-2} - \theta_{N-1}\| + \cdots + \|\theta_{N-p} - \theta_{N-1}\|) \\ &\le \|F_2\|(p-1)\sum_{t=N-p}^{N-2}\|\theta_{t+1} - \theta_t\|\end{aligned}$$

Combining the bounds above leads to

$$\text{Part III} = O\left(\sum_{t=0}^{N-1}\|\theta_{t+1} - \theta_t\|\right)$$

The proof is completed by summing up the bounds for Part I, II, III.

A.2.4 Proof of Theorem 3.3

Proof intuition: By the problem transformation in Section 3.3.1, the fundamental limit of the online control problem is equivalent to the fundamental limit of the online convex optimization problem with objective $C(\boldsymbol{z})$. Therefore, we will focus on $C(\boldsymbol{z})$. Since the lower bound is for the worst case scenario, we only need to construct some tracking trajectories $\{\theta_t\}$ for Theorem 3.3 to hold. However, it is generally difficult to construct the tracking trajectories, so we consider randomly generated θ_t and show that the regret in expectation can be lower bounded. Then, there must exist some realization of the randomly generated $\{\theta_t\}$ such that the regret lower bound holds.

Formal proof:

Step 1: construct LQ tracking. For simplicity, we construct a single-input system with $n = p$ and $A \in \mathbb{R}^{n,n}$ and $B \in \mathbb{R}^{n\times 1}$ as follows: [8]

$$A = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \\ & & 0 & 1 \\ 1 & 0 & \cdots & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

(A, B) is controllable because $(B, AB, \ldots, A^{p-1}B)$ is full rank. A's controllability index is $p = n$.

Next, we construct Q and R. For any $\zeta > 1$ and p, define $\delta = \frac{4}{(\zeta-1)p}$. Let $Q = \delta I_n$ and $R = 1$ for $0 \le t \le N-1$. Let $P^e = P^e(Q, R)$ be the solution to the DARE. The next lemma shows that P^e is a diagonal matrix and its diagonal entries can be characterized.

Lemma A.19 (Form of P^e)**.** *Let P^e denote the solution to the DARE determined by A, B, Q, R defined above. Then P^e satisfies the form*

$$P^e = \begin{pmatrix} q_1 & 0 & \cdots & 0 \\ 0 & q_2 & \cdots & 0 \\ & & \ddots & \\ 0 & & \cdots & q_n \end{pmatrix},$$

where $q_i = q_1 + (i-1)\delta$ for $1 \le i \le n$ and $\delta < q_1 < \delta + 1$.

Proof of Lemma A.19. The DARE exists a unique positive definite solution [127]. Suppose the solution is diagonal and substitute it in the DARE as follows.

$$P^e = Q + A^\top (P^e - P^e B(B^\top P^e B + R)^{-1} B^\top P^e) A$$

[8] It is easy to generalize the construction to multi-input case by constructing m decoupled subsystems.

$$\begin{pmatrix} q_1 & 0 & \cdots & 0 \\ 0 & q_2 & \cdots & 0 \\ & & \ddots & \\ 0 & & \cdots & q_n \end{pmatrix} = \begin{pmatrix} q_n/(1+q_n)+\delta & 0 & \cdots & 0 \\ 0 & q_1+\delta & \cdots & 0 \\ & & \ddots & \\ 0 & & \cdots & q_{n-1}+\delta \end{pmatrix}$$

So we have $q_i = q_{i-1} + \delta$ for $1 \le i \le n-1$, and $q_n/(1+q_n) + \delta = q_1 = q_n - (n-1)\delta$. Thus, $q_n = \frac{n\delta + \sqrt{n^2\delta^2 + 4n\delta}}{2} > n\delta$. It is straightforward that $q_1 = q_n - (n-1)\delta > \delta > 0$, and $q_1 < \delta + 1$ by $q_n/(1+q_n) < 1$. So we have found the unique positive definite solution to the DARE. □

Next, we will construct θ_t. Let $\theta_0 = \theta_N = \beta_N^e = 0$ for simplicity. For θ_t when $1 \le t \le N-1$, we divide the $N-1$ stages into E epochs, each with length $\Delta = \lceil \frac{N-1}{\lfloor \frac{L_N}{2\bar{\theta}} \rfloor} \rceil$, possibly except the last epoch. This is possible because $1 \le \Delta \le N-1$ by the conditions in Theorem 3.3. Thus, $E = \lceil \frac{N-1}{\Delta} \rceil$. Let $\mathscr{J}$ be the first stage of the each epoch: $\mathscr{J} = \{1, \Delta+1, \ldots, (E-1)\Delta+1\}$. Let θ_t for $t \in \mathscr{J}$ independently and identically follow the distribution below.

$$\Pr(\theta_t^i = a) = \begin{cases} 1/2 & \text{if } a = \sigma \\ 1/2 & \text{if } a = -\sigma \end{cases}, \quad \text{i.i.d. for all } i \in [n],\ t \in \mathscr{J},$$

where $\sigma = \frac{\bar{\theta}}{\sqrt{n}}$. It can be easily verified that $\|\theta\| = \bar{\theta}$ for any realization of this distribution, so Assumption 3.3 is satisfied. Let the other θ_t in each epoch be equal to the θ at the start of their corresponding epochs, i.e. $\theta_{k\Delta+1} = \theta_{k\Delta+2} = \cdots = \theta_{(k+1)\Delta}$, when $k \le E-1$, and $\theta_{k\Delta+1} = \cdots = \theta_{N-1}$ when $k = E$. The following inequalities show that the constructed $\{\theta_t\}$ satisfies the variation budget:

$$\sum_{t=0}^{N} \|\theta_t - \theta_{t-1}\| = \|\theta_1 - \theta_0\| + \sum_{k=1}^{E-1} \|\theta_{k\Delta+1} - \theta_{k\Delta}\| + \|\theta_{N-1} - \theta_N\|$$

$$\leq \bar{\theta} + 2(E-1)\bar{\theta} + \bar{\theta} = 2\bar{\theta}E$$
$$\leq 2\bar{\theta}\lfloor \frac{L_N}{2\bar{\theta}} \rfloor \leq 2\bar{\theta}\frac{L_N}{2\bar{\theta}} = L_N$$

where the first equality is by $\theta_0 = \theta_{-1} = \theta_N = 0$, the first inequality is by $\|\theta_t\| = \bar{\theta}$ when $1 \leq t \leq N-1$, the second inequality is by $\Delta = \lceil \frac{N-1}{\lfloor \frac{L_N}{2\bar{\theta}} \rfloor} \rceil \geq \frac{N-1}{\lfloor \frac{L_N}{2\bar{\theta}} \rfloor}$, and thus $\lfloor \frac{L_N}{2\bar{\theta}} \rfloor \geq \lceil \frac{N-1}{\Delta} \rceil = E$.

The total cost of our constructed LQ tracking problem is

$$J(\boldsymbol{x}, \boldsymbol{u}) = \sum_{t=0}^{N-1} (\frac{\delta}{2}\|x_t - \theta_t\|^2 + \frac{1}{2}u_t^2) + \frac{1}{2}x_N^\top P^e x_N$$

We will verify that $C(\boldsymbol{z})$'s condition number is ζ in Step 2.

Step 2: problem transformation and the optimal solution $\boldsymbol{z}^*$. By the problem transformation in Section 3.3.1, we let $z_t = x_t^n$, and the equivalent cost function $C(\boldsymbol{z})$ is given below.

$$C(\boldsymbol{z}) = \sum_{t=0}^{N-1} (\frac{\delta}{2}\sum_{i=1}^{n}(z_{t-n+i} - \theta_t^i)^2 + \frac{1}{2}(z_{t+1} - z_{t-n+1})^2) + \frac{1}{2}\sum_{i=1}^{n} q_i z_{N-n+i}^2$$

and $z_t = 0$ and $\theta_t = 0$ for $t \leq 0$.

Since $C(\boldsymbol{z})$ is strongly convex, $\min C(\boldsymbol{z})$ admits a unique optimal solution, denoted as $\boldsymbol{z}^*$, which is determined by the first-order optimality condition: $\nabla C(\boldsymbol{z}^*) = 0$. In addition, our constructed $C(\boldsymbol{z})$ is a quadratic function, so there exists a matrix $H \in \mathbb{R}^{N \times N}$ and a vector $\eta \in \mathbb{R}^N$ such that $\nabla C(\boldsymbol{z}^*) = H\boldsymbol{z}^* - \eta = 0$. By the partial gradients of $C(\boldsymbol{z})$ below,

$$\frac{\partial C}{\partial z_t} = \delta(z_t - \theta_t^n + z_t - \theta_{t+1}^{n-1} + \cdots + z_t - \theta_{t+n-1}^1) + z_t - z_{t+n} + z_t - z_{t-n},\ 1 \leq t \leq N-n$$
$$\frac{\partial C}{\partial z_t} = \delta(z_t - \theta_t^n + \cdots + z_t - \theta_{N-1}^{n+t-N+1}) + q_{n+t-N} z_t + z_t - z_{t-n}, \qquad N-n+1 \leq t \leq N$$

For simplicity and without loss of generality, we assume that N/n is an integer. Then, by Lemma A.19, H can be represented as the block matrix below

$$H = \begin{pmatrix} (\delta n+2)I_n & -I_n & \cdots & & \\ -I_n & (\delta n+2)I_n & \ddots & & \\ & \ddots & \ddots & -I_n & \\ & & -I_n & (q_n+1)I_n \end{pmatrix} \in \mathbb{R}^{N\times N}.$$

η is a linear combination of θ: for $1 \le t \le N$, we have $\eta_t = \delta(\theta_t^n + \cdots + \theta_{t+n-1}^1) = \delta(e_n^\top \theta_t + \cdots + e_1^\top \theta_{t+n-1})$ where $e_1, \ldots, e_n \in \mathbb{R}^n$ are standard basis vectors and $\theta_t = 0$ for $t \ge N$.

By Gergoskin's Disc Theorem and Lemma A.19, H's condition number is $(\delta n+4)/\delta n = \zeta$ by our choice of δ in Step 1 and $p = n$. Thus we have shown that $C(\boldsymbol{z})$'s condition number is ζ.

Since H is strictly diagonally dominant with positive diagonal entries and nonpositive off-diagonal entries, H is invertible and its inverse, denoted by Y, is nonnegative. Consequently, the optimal solution can be represented as $\boldsymbol{z}^* = Y\eta$. Since η is linear in $\{\theta_t\}$, z_t^* is also linear in $\{\theta_t\}$ and can be characterized by the following.

$$\begin{aligned} z_{t+1}^* &= \sum_{i=1}^{N} Y_{t+1,i}\eta_i = \delta \sum_{i=1}^{N} Y_{t+1,i} \sum_{j=0}^{n-1} e_{n-j}^\top \theta_{i+j} \\ &= \delta \sum_{k=1}^{N-1} \left(\sum_{i=1}^{n} Y_{t+1,i+k-n} e_i^\top \right) \theta_k \\ &=: \delta \sum_{k=1}^{N-1} v_{t+1,k}\theta_k \end{aligned} \tag{A.21}$$

where $\theta_t = 0$ for $t \ge N$, $Y_{t+1,i} = 0$ for $i \le 0$, and $v_{t+1,k} \coloneqq \sum_{i=1}^{n} Y_{t+1,i+k-n} e_i^\top$.

In addition, we are able to show in the next lemma that Y has decaying row entries

starting at the diagonal entries. The proof is technical and deferred to the Appendix F.1.

Lemma A.20. *When N/n is an integer, the inverse of H, denoted by Y, can be represented as a block matrix*

$$Y = \begin{pmatrix} y_{1,1}I_n & y_{1,2}I_n & \cdots & y_{1,N/n}I_n \\ y_{2,1}I_n & y_{2,2}I_n & \cdots & y_{2,N/n}I_n \\ \vdots & \ddots & \ddots & \vdots \\ y_{N/n,1}I_n & y_{N/n,2}I_n & \cdots & y_{N/n,N/n}I_n \end{pmatrix}$$

where $y_{t,t+\tau} \geq \frac{1-\rho}{\delta n+2}\rho^\tau > 0$ for $\tau \geq 0$ and $\rho = \frac{\sqrt{\zeta}-1}{\sqrt{\zeta}+1}$.

Step 3: characterize $z_{t+1}(\mathscr{A}^z)$. For any online control algorithm $\mathscr{A}$, we can define an equivalent online algorithm for z, denoted as $\mathscr{A}^z$. $\mathscr{A}^z$, at each time t, outputs $z_{t+1}(\mathscr{A}^z)$ based on the predictions and the history, i.e.,

$$z_{t+1}(\mathscr{A}^z) = \mathscr{A}^z(\{\theta_s\}_{s=0}^{t+W-1}), \quad t \geq 0$$

For simplicity, we consider online deterministic algorithm.[9] Notice that z_{t+1} is a random variable because $\theta_1, \ldots, \theta_{t+W-1}$ are random. Based on this observation and Lemma A.20, we are able to provide a regret lower bound in Step 4.

Step 4: prove the regret lower bound on $\mathscr{A}$. Roughly speaking, the regret occurs when something unexpected happens beyond the prediction window, that is, at each t, the prediction window goes as far as $t + W - 1$, but if θ_{t+W} changes from θ_{t+W-1}, the online algorithm cannot prepare for it, resulting in poor control and positive regret.

[9]The proof can be easily generalized to random algorithms

By our construction, when $t+W \in \mathscr{J}$, θ_{t+W} changes from θ_{t+W-1}. To study such t, we define a set $\mathscr{J}_1 = \{0 \le t \le N-W-1 \mid t+W \in \mathscr{J}\}$. It can be shown that the cardinality of $\mathscr{J}_1$ can be lower bounded by L_N up to some constants:

$$|\mathscr{J}_1| \ge \frac{1}{18\bar{\theta}} L_N \tag{A.22}$$

The proof of (A.22) is provided below.

$$\begin{aligned}
|\mathscr{J}_1| &= |\{W \le t \le N-1 \mid t \in \mathscr{J}\}| \\
&= |\mathscr{J}| - |\{1 \le t \le W-1 \mid t \in \mathscr{J}\}| \\
&= \lceil \frac{N-1}{\Delta} \rceil - \lceil \frac{W-1}{\Delta} \rceil \\
&\ge \lfloor \frac{N-W}{\Delta} \rfloor \\
&\ge \frac{1}{2} \frac{N-W}{\Delta} \\
&\ge \frac{1}{2} \frac{N-W}{N-1+\lfloor \frac{L_N}{2\bar{\theta}} \rfloor} \lfloor \frac{L_N}{2\bar{\theta}} \rfloor \\
&\ge \frac{1}{2} \frac{N-\frac{1}{3}N}{N-1+N+1/2} \lfloor \frac{L_N}{2\bar{\theta}} \rfloor \ge \frac{1}{6} \lfloor \frac{L_N}{2\bar{\theta}} \rfloor \\
&\ge \frac{1}{6}\frac{2}{3}\frac{L_N}{2\bar{\theta}} = \frac{1}{18}\frac{L_N}{\bar{\theta}}
\end{aligned}$$

where the first inequality is by the definition of the ceiling and floor operators, the second inequality is by $\frac{N-W}{\Delta} \ge 1$ under the conditions on N, W, L_N in Theorem 3.3, the third inequality is by $\Delta = \lceil \frac{N-1}{\lfloor \frac{L_N}{2\bar{\theta}} \rfloor} \rceil \le \frac{N-1}{\lfloor \frac{L_N}{2\bar{\theta}} \rfloor} + 1$, the fourth inequality is by $L_N \le (2N+1)\bar{\theta}$ in Theorem 3's statement, the last inequality is by $L_N \ge 4\bar{\theta}$ in Theorem 3's statement.

Moreover, we can show in Lemma A.21 that, for all $t \in \mathscr{J}_1$, the online decision $z_{t+1}(\mathscr{A}^z)$ is different from the optimal solution z^*_{t+1} and the difference is lower bounded,

Lemma A.21. *For any online algorithm $\mathscr{A}^z$, when $t \in \mathscr{J}_1$,*

$$\mathbb{E}\,|z_{t+1}(\mathscr{A}^z) - z^*_{t+1}|^2 \ge c_{10}\sigma^2\rho^{2K}$$

where c_{10} is a constant determined by A, B, n, Q, R constructed above and $\rho = \frac{\sqrt{\zeta}-1}{\sqrt{\zeta}+1}$.

The proof is provided in Appendix F.2.

The lower bound on the difference between the online decision and the optimal decision results in a lower bound on the regret. By the $n\delta$-strong convexity of $C(\boldsymbol{z})$,

$$\begin{aligned}\mathbb{E}(C(\boldsymbol{z}(\mathscr{A}^z)) - C(\boldsymbol{z}^*)) &\geq \frac{\delta n}{2} \sum_{t \in \mathscr{J}_1} \mathbb{E}\,|z_{t+1}(\mathscr{A}^z) - z_{t+1}^*|^2 \\ &\geq |\mathscr{J}_1| c_{10} \sigma^2 \rho^{2K} \\ &\geq \frac{L_N}{18\bar{\theta}} c_{10} \sigma^2 \rho^{2K} = \Omega(L_N \rho^{2K})\end{aligned}$$

By the equivalence between $\mathscr{A}$ and $\mathscr{A}^z$, we have $\mathbb{E}\, J(\mathscr{A}) - \mathbb{E}\, J^* = \Omega(\rho^{2K} L_N)$. By the property of expectation, there must exist some realization of the random $\{\theta_t\}$ such that $J(\mathscr{A}) - J^* = \Omega(\rho^{2K} L_N)$, where $\rho = \frac{\sqrt{\zeta}-1}{\sqrt{\zeta}+1}$. This completes the proof. □

A.2.4.1 Proof of Lemma A.20

Proof. Since H is a block matrix

$$H = \begin{pmatrix} (\delta n + 2) I_n & -I_n & \cdots & \\ -I_n & (\delta n + 2) I_n & \ddots & \\ & \ddots & \ddots & -I_n \\ & & -I_n & (q_n + 1) I_n \end{pmatrix}$$

its inverse matrix Y can also be represented as a block matrix. Moreover, let

$$H_1 = \begin{pmatrix} \delta n + 2 & -1 & \cdots & 0 \\ -1 & \delta n + 2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & -1 & q_n + 1 \end{pmatrix}$$

and define $\bar{Y} = (H_1)^{-1} = (y_{ij})_{i,j=1}^{N/n}$. Then the inverse matrix Y can be represented as the block matrix: $Y = (y_{ij} I_n)_{i,j=1}^{N/n}$.

Now, it suffices to provide a lower bound on y_{ij}.

Since H_1 is a symmetric positive definite tridiagonal matrix, by [223], the inverse has an explicit formula given by $(H_1)_{ij}^{-1} = a_i b_j$ and

$$
\begin{aligned}
a_t &= \frac{\rho}{1-\rho^2}\left(\frac{1}{\rho^t} - \rho^t\right) \\
b_t &= c_3 \frac{1}{\rho^{N-t}} + c_4 \rho^{N-t} \\
c_3 &= b_N \left(\frac{(q_n+1)\rho - \rho^2}{1-\rho^2}\right) \\
c_4 &= b_N \frac{1-(q_n+1)\rho}{1-\rho^2} \\
b_N &= \frac{1}{-a_{N-1} + (q_n+1)a_N}
\end{aligned}
$$

In the following, we will show $y_{t,t+\tau} = a_t b_{t+\tau} \geq \frac{1-\rho}{\delta n+2}\rho^\tau$ when $\tau \geq 0$. Firstly, it is easy to verify that

$$
\rho^t a_t = \frac{\rho}{1-\rho^2}(1-\rho^{2t}) \geq \rho
$$

since $t \geq 1$ and $\rho < 1$.

Secondly, we bound b_N in the following way:

$$
\rho^{-N} b_N = \frac{1}{(q_n+1)(1-\rho^{2N}) - (\rho - \rho^{2N-1})} \frac{1-\rho^2}{\rho} \geq \frac{1}{(\delta n+2)} \frac{1-\rho^2}{\rho}
$$

because $0 < (q_n+1)(1-\rho^{2N}) - (\rho - \rho^{2N-1}) \leq (\delta n+2)$ by $n\delta < q_n < n\delta + 1$ in Lemma A.19.

Thirdly, we bound $b_{t+\tau}$. When $1-(q_n+1)\rho \geq 0$

$$
\rho^{N-t-\tau} b_{t+\tau} = b_N \left(\frac{(q_n+1)\rho - \rho^2}{1-\rho^2}\right) + b_N \frac{1-(q_n+1)\rho}{1-\rho^2} \rho^{2(N-t-\tau)}
$$

$$\geq b_N \left(\frac{(q_n+1)\rho - \rho^2}{1-\rho^2} \right)$$
$$\geq b_N \left(\frac{(\delta n+1)\rho - \rho^2}{1-\rho^2} \right)$$
$$= \frac{1-\rho}{1-\rho^2} b_N$$

where the first inequality is by $1-(q_n+1)\rho \geq 0$, the second inequality is by $qn > n\delta$ in Lemma A.19, and the last equality is by $\rho^2 - (\delta n+2)\rho + 1 = 0$.

When $1-(q_n+1)\rho < 0$

$$\rho^{N-t-\tau} b_{t+\tau} = b_N \left(\frac{(q_n+1)\rho - \rho^2}{1-\rho^2} \right) + b_N \frac{1-(q_n+1)\rho}{1-\rho^2} \rho^{2(N-t-\tau)}$$
$$\geq b_N \left(\frac{(q_n+1)\rho - \rho^2}{1-\rho^2} \right) + b_N \frac{1-(q_n+1)\rho}{1-\rho^2}$$
$$\geq b_N \geq \frac{1-\rho}{1-\rho^2} b_N$$

where the first inequality is by $1-(q_n+1)\rho < 0, \rho \leq 1$, the second inequality is by $\rho^{2(N-t-\tau)} \leq 1$. Thus, we obtained a lower bound for $b_{t+\tau}$.

Combining bounds of $a_t, b_{t+\tau}, b_N$ together yields

$$y_{t,t+\tau} = a_t b_{t+\tau} \geq \rho b_N \frac{1-\rho}{1-\rho^2} \rho^{\tau-N} \geq \frac{1-\rho}{(\delta n+2)} \rho^{\tau}$$

□

A.2.4.2 Proof of Lemma A.21

Proof. By our construction, θ_t is random, $z_{t+1}^{\mathscr{A}}$ is also random and its randomness is provided by $\theta_1, \ldots, \theta_{t+W-1}$, while z_{t+1}^* is determined by all θ_t. When $t \in \mathscr{J}_1$,

$$\mathbb{E}\,|z_{t+1}^{\mathscr{A}} - z_{t+1}^*|^2 = \mathbb{E}\,|z_{t+1}^{\mathscr{A}} - \delta \sum_{i=1}^{N-1} v_{t+1,i}\theta_i|^2$$

$$= \mathbb{E}\,|z_{t+1}^{\mathscr{A}} - \delta \sum_{i=1}^{t+W-1} v_{t+1,i}\theta_i\|^2 + \delta^2\,\mathbb{E}\,|\sum_{i=t+W}^{N-1} v_{t+1,i}\theta_i|^2$$
$$\geq \delta^2\,\mathbb{E}\,|\sum_{i=t+W}^{N-1} v_{t+1,i}\theta_i|^2,$$

where the first equality is by (A.21), the second equality is by $\mathbb{E}\,\theta_\tau = 0$ for all τ, and $\theta_{t+W}, \ldots, \theta_N$ are independent of $\theta_1, \ldots, \theta_{t+W-1}$ when $t \in \mathscr{J}_1$.

Further,

$$\begin{aligned}
\mathbb{E}\,|\sum_{i=t+W}^{N-1} v_{t+1,i}\theta_i|^2 &= \mathbb{E}\,|\sum_{i=t+W}^{t+W+\Delta-1} v_{t+1,i}\theta_{t+W}|^2 + \cdots + \mathbb{E}\,|\sum_{i=(E-1)\Delta+1}^{N-1} v_{t+1,i}\theta_{(E-1)\Delta+1}|^2 \\
&= \|\sum_{i=t+W}^{t+W+\Delta-1} v_{t+1,i}\|^2\sigma^2 + \cdots + \|\sum_{i=(E-1)\Delta+1}^{N-1} v_{t+1,i}\|^2\sigma^2 \\
&\geq \sigma^2 \sum_{i=t+W}^{N-1} \|v_{t+1,i}\|^2 \\
&= \sigma^2 \sum_{i=t+W}^{N-1} (\sum_{k=0}^{n-1} Y_{t+1,i-k}^2) \geq \sigma^2 \sum_{i=t+1+W-n}^{N-1} Y_{t+1,i}^2 \\
&= \sigma^2 \sum_{i=t+1+W-n}^{N} Y_{t+1,i}^2
\end{aligned}$$

where the first equality is because the theta in one epoch are equal by our construction, the second equality is because $\mathrm{cov}(\theta_\tau) = \sigma^2 I_n$, the first inequality is because the entries of $v_{t+1,i}$ are nonnegative, the third equality is by the definition of $v_{t+1,i}$ in (A.21), and the last equality is because when $t \in \mathscr{J}_1$, $Y_{t+1,N} = 0$.

When $1 \leq W \leq n$, $\sum_{i=t+1+W-n}^{N} Y_{t+1,i}^2 \geq Y_{t+1,t+1}^2 = Y_{t+1,t+1+n\lfloor\frac{W-1}{n}\rfloor}^2$. When $W > n$, $\sum_{i=t+1+W-n}^{N} Y_{t+1,i}^2 \geq Y_{t+1,t+1+n\lceil\frac{W-n}{n}\rceil}^2$. Moreover, when $W \geq 1$, $\lceil\frac{W-n}{n}\rceil = \lfloor\frac{W-1}{n}\rfloor$. In summary, for $W \geq 1$,

$$\sum_{i=t+1+W-n}^{N} Y_{t+1,i}^2 \geq Y_{t+1,t+1+n\lfloor\frac{W-1}{n}\rfloor}^2 \geq \rho^{2K}(\frac{1-\rho}{\delta n+2})^2$$

where the last inequality is by Lemma A.20. This completes the proof.

□

Appendix B | Appendix to Part II

This appendix includes the proofs for the technical results in Chapter 6 and Chapter 7.

B.1 Proofs for Chapter 6

B.1.1 Proof of Lemma 6.3

Lemma 6.3 is proved by first establishing a smoothness property of $g_i^x(\cdot)$ and $g_j^u(\cdot)$ in Lemma B.1 and then leveraging the slow updates of OGD. The details of the proof are provided below.

Lemma B.1. *Consider any* $\mathbf{M}_t \in \mathbb{M}_H$ *and any* $\tilde{\mathbf{M}}_\mathbf{t} \in \mathbb{M}_H$ *for all* t*, then*

$$\max_{1\le i\le k_x} \left| g_i^x(\mathbf{M}_{t-H:t-1}) - g_i^x(\tilde{\mathbf{M}}_{t-H:t-1}) \right| \le L_g(H) \sum_{k=1}^{H} (1-\gamma)^{k-1} \|\mathbf{M}_{t-k} - \tilde{\mathbf{M}}_{t-k}\|_F$$

$$\max_{1\le j\le k_u} \left| g_j^u(\mathbf{M}_{t-H:t}) - g_j^u(\tilde{\mathbf{M}}_{t-H:t}) \right| \le L_g(H) \sum_{k=0}^{H} (1-\gamma)^{\max(k-1,0)} \|\mathbf{M}_{t-k} - \tilde{\mathbf{M}}_{t-k}\|_F$$

where $L_g(H) = w_{\max}\sqrt{n}\max(\|D_x\|_\infty, \|D_u\|_\infty)\kappa^2\kappa_B\sqrt{H}$.

Proof. We first provide a bound on $\|D_{x,i}^\top \Phi_k^x(\mathbf{M}_{t-H:t-1})\|_1 - \|D_{x,i}^\top \Phi_k^x(\tilde{\mathbf{M}}_{t-H:t-1})\|_1$.

$$\begin{aligned}
&\left| \|D_{x,i}^\top \Phi_k^x(\mathbf{M}_{t-H:t-1})\|_1 - \|D_{x,i}^\top \Phi_k^x(\tilde{\mathbf{M}}_{t-H:t-1})\|_1 \right| \le \|D_{x,i}^\top \Phi_k^x(\mathbf{M}_{t-H:t-1}) - D_{x,i}^\top \Phi_k^x(\tilde{\mathbf{M}}_{t-H:t-1})\|_1 \\
&= \left\| D_{x,i}^\top \left(\sum_{s=1}^{H} A_\mathsf{K}^{s-1} B (M_{t-s}[k-s] - \tilde{M}_{t-s}[k-s]) \mathbf{1}_{(1\le k-s\le H)} \right) \right\|_1 \\
&= \left\| \sum_{s=1}^{H} D_{x,i}^\top A_\mathsf{K}^{s-1} B (M_{t-s}[k-s] - \tilde{M}_{t-s}[k-s]) \mathbf{1}_{(1\le k-s\le H)} \right\|_1
\end{aligned}$$

$$\leq \sqrt{n}\left\|\sum_{s=1}^{H} D_{x,i}^{\top} A_{\mathsf{K}}^{s-1} B(M_{t-s}[k-s]-\tilde{M}_{t-s}[k-s])\mathbf{1}_{(1\leq k-s\leq H)}\right\|_2$$
$$\leq \sqrt{n}\sum_{s=1}^{H}\|D_{x,i}^{\top}\|_2\|A_{\mathsf{K}}^{s-1}\|_2\|B\|_2\|M_{t-s}[k-s]-\tilde{M}_{t-s}[k-s]\|_2\mathbf{1}_{(1\leq k-s\leq H)}$$
$$\leq \sqrt{n}\sum_{s=1}^{H}\|D_{x,i}^{\top}\|_1\kappa(1-\gamma)^{s-1}\kappa_B\|M_{t-s}[k-s]-\tilde{M}_{t-s}[k-s]\|_2\mathbf{1}_{(1\leq k-s\leq H)}$$
$$\leq \sqrt{n}\|D_x\|_\infty\kappa\kappa_B\sum_{s=1}^{H}(1-\gamma)^{s-1}\|M_{t-s}[k-s]-\tilde{M}_{t-s}[k-s]\|_2\mathbf{1}_{(1\leq k-s\leq H)}$$

Therefore, for any $1\leq i\leq k_x$, we have

$$\begin{aligned}
&|g_i^x(\mathbf{M}_{t-H:t-1})-g_i^x(\tilde{\mathbf{M}}_{t-H:t-1})|\\
&\leq w_{\max}\sum_{k=1}^{2H}|\|D_{x,i}^{\top}\Phi_k^x(\mathbf{M}_{t-H:t-1})\|_1-\|D_{x,i}^{\top}\Phi_k^x(\tilde{\mathbf{M}}_{t-H:t-1})\|_1|\\
&\leq w_{\max}\sqrt{n}\|D_x\|_\infty\kappa\kappa_B\sum_{k=1}^{2H}\sum_{s=1}^{H}(1-\gamma)^{s-1}\|M_{t-s}[k-s]-\tilde{M}_{t-s}[k-s]\|_2\mathbf{1}_{(1\leq k-s\leq H)}\\
&= w_{\max}\sqrt{n}\|D_x\|_\infty\kappa\kappa_B\sum_{s=1}^{H}\sum_{k=1}^{2H}(1-\gamma)^{s-1}\|M_{t-s}[k-s]-\tilde{M}_{t-s}[k-s]\|_2\mathbf{1}_{(1\leq k-s\leq H)}\\
&\leq w_{\max}\sqrt{n}\|D_x\|_\infty\kappa\kappa_B\sum_{s=1}^{H}\sum_{k=1}^{2H}(1-\gamma)^{s-1}\|M_{t-s}[k-s]-\tilde{M}_{t-s}[k-s]\|_F\mathbf{1}_{(1\leq k-s\leq H)}\\
&\leq w_{\max}\sqrt{n}\|D_x\|_\infty\kappa\kappa_B\sqrt{H}\sum_{s=1}^{H}(1-\gamma)^{s-1}\|\mathbf{M}_{t-s}-\tilde{\mathbf{M}}_{t-s}\|_F
\end{aligned}$$

Next, we provide a bound on $\|D_{u,j}^{\top}\Phi_k^u(\mathbf{M}_{t-H:t})\|_1-\|D_{u,j}^{\top}\Phi_k^u(\tilde{\mathbf{M}}_{t-H:t})\|_1$.

$$\begin{aligned}
&|\|D_{u,j}^{\top}\Phi_k^u(\mathbf{M}_{t-H:t})\|_1-\|D_{u,j}^{\top}\Phi_k^u(\tilde{\mathbf{M}}_{t-H:t})\|_1|\leq\|D_{u,j}^{\top}(\Phi_k^u(\mathbf{M}_{t-H:t})-\Phi_k^u(\tilde{\mathbf{M}}_{t-H:t})\|_1\\
&\leq\|D_{u,j}^{\top}(M_t[k]-\tilde{M}_t[k])\|_1\mathbf{1}_{(k\leq H)}\\
&\quad+\left\|\sum_{s=1}^{H}D_{u,j}^{\top}\mathsf{K}A_{\mathsf{K}}^{s-1}B(M_{t-s}[k-s]-\tilde{M}_{t-s}[k-s])\right\|_1\mathbf{1}_{(1\leq k-s\leq H)}\\
&\leq\sqrt{n}\|D_{u,j}^{\top}(M_t[k]-\tilde{M}_t[k])\|_2\mathbf{1}_{(k\leq H)}\\
&\quad+\sqrt{n}\left\|\sum_{s=1}^{H}D_{u,j}^{\top}\mathsf{K}A_{\mathsf{K}}^{s-1}B(M_{t-s}[k-s]-\tilde{M}_{t-s}[k-s])\right\|_2\mathbf{1}_{(1\leq k-s\leq H)}
\end{aligned}$$

$$\leq \sqrt{n}\|D_{u,j}^\top\|_2\|M_t[k]-\tilde{M}_t[k]\|_2\mathbf{1}_{(k\leq H)}$$
$$+\sqrt{n}\sum_{s=1}^{H}\|D_{u,j}\|_2\kappa^2(1-\gamma)^{s-1}\kappa_B\|M_{t-s}[k-s]-\tilde{M}_{t-s}[k-s]\|_2\mathbf{1}_{(1\leq k-s\leq H)}$$
$$\leq \sqrt{n}\|D_{u,j}^\top\|_1\|M_t[k]-\tilde{M}_t[k]\|_2\mathbf{1}_{(k\leq H)}$$
$$+\sqrt{n}\sum_{s=1}^{H}\|D_{u,j}\|_1\kappa^2(1-\gamma)^{s-1}\kappa_B\|M_{t-s}[k-s]-\tilde{M}_{t-s}[k-s]\|_2\mathbf{1}_{(1\leq k-s\leq H)}$$

Therefore, for any $1\leq j\leq k_u$, we have

$$|g_j^u(\mathbf{M}_{t-H:t})-g_j^u(\tilde{\mathbf{M}}_{t-H:t})|\leq\sum_{k=1}^{2H}w_{\max}|\|D_{u,j}^\top\Phi_k^u(\mathbf{M}_{t-H:t})\|_1-\|D_{u,j}^\top\Phi_k^u(\tilde{\mathbf{M}}_{t-H:t})\|_1|$$
$$\leq w_{\max}\sqrt{n}\|D_u\|_\infty\sqrt{H}\|\mathbf{M}_t-\tilde{\mathbf{M}}_t\|_F$$
$$+w_{\max}\sqrt{n}\|D_u\|_\infty\kappa^2\kappa_B\sqrt{H}\sum_{s=1}^{H}(1-\gamma)^{s-1}\|\mathbf{M}_{t-s}-\tilde{\mathbf{M}}_{t-s}\|_F$$
$$\leq w_{\max}\sqrt{n}\|D_u\|_\infty\kappa^2\kappa_B\sqrt{H}\sum_{s=0}^{H}(1-\gamma)^{\max(s-1,0)}\|\mathbf{M}_{t-s}-\tilde{\mathbf{M}}_{t-s}\|_F$$

where the last inequality uses $\kappa\geq 1,\kappa_B\geq 1$. □

Proof of Lemma 6.3. Firstly, by OGD's definition and Lemma B.2, we have $\|\mathbf{M}_t-\mathbf{M}_{t-1}\|_F\leq\eta G_f$ and $\|\mathbf{M}_t-\mathbf{M}_{t-k}\|_F\leq k\eta G_f$. By Lemma B.1, we have

$$\max_{1\leq i\leq k_x}|g_i^x(\mathbf{M}_{t-H+1:t})-g_i^x(\mathbf{M}_t,\ldots,\mathbf{M}_t)|\leq L_g(H)\sum_{k=1}^{H-1}(1-\gamma)^{k-1}\|\mathbf{M}_{t-k}-\mathbf{M}_t\|_F$$
$$\leq L_g(H)\sum_{k=1}^{H-1}(1-\gamma)^{k-1}k\eta G_f\leq L_g(H)\eta G_f\frac{1}{\gamma^2}$$

and

$$\max_{1\leq j\leq k_u}\left|g_j^u(\mathbf{M}_{t-H:t})-g_j^u(\tilde{\mathbf{M}}_t,\ldots,\mathbf{M}_t)\right|\leq L_g(H)\sum_{k=1}^{H}(1-\gamma)^{k-1}\|\mathbf{M}_{t-k}-\mathbf{M}_t\|_F\leq\epsilon_2(\eta,H)$$
$$\leq L_g(H)\sum_{k=1}^{H}(1-\gamma)^{k-1}k\eta G_f\leq L_g(H)\eta G_f\frac{1}{\gamma^2}\leq\epsilon_2(\eta,H)$$

where $\epsilon_2(\eta,H)=c_2n^2\sqrt{m}H^2\eta$. □

B.1.2 Proof of Lemma 6.5

The proof is built upon the following lemma.

Lemma B.2. *Consider any* $\mathbf{M}_t \in \mathbb{M}_H$ *and any* $\tilde{\mathbf{M}}_\mathbf{t} \in \mathbb{M}_H$ *for all* t*, then*

$$|f_t(\mathbf{M}_{t-H:t}) - f_t(\tilde{\mathbf{M}}_{t-H:t})| \le Gb(1+\kappa)\sqrt{n}w_{\max}\kappa\kappa_B \sum_{i=0}^{H}(1-\gamma)^{\max(i-1,0)} \sum_{j=1}^{H} \|M[j]_{t-i} - \tilde{M}[j]_{t-i}\|_2.$$

Further, $\|\nabla \mathring{f}_t(\mathbf{M}_t)\|_F \le G_f$ *for* $\mathbf{M}_t \in \mathbb{M}_H$*, where* $G_f = Gb(1+\kappa)\sqrt{n}w_{\max}\kappa\kappa_B\sqrt{H}\frac{1+\gamma}{\gamma}$.

Consequently, when $H \ge \frac{\log(2\kappa)}{\log((1-\gamma)^{-1})}$*, then* $G_f \le \Theta(\sqrt{n^3H^3m})$.

Proof. Let $\tilde{x}_t$ and $\tilde{\tilde{x}}_t$ denote the approximate states generated by $\mathbf{M}_{t-H:t-1}$ and $\tilde{\mathbf{M}}_{t-H:t-1}$ respectively. Define $\tilde{u}_t$ and $\tilde{\tilde{u}}_t$ similarly. We have

$$\begin{aligned}
\|\tilde{x}_t - \tilde{\tilde{x}}_t\|_2 &= \|\sum_{k=1}^{2H}(\Phi_k^x(\mathbf{M}_{t-H:t-1}) - \Phi_k^x(\tilde{\mathbf{M}}_{t-H:t-1}))w_{t-k}\|_2 \\
&\le \sum_{k=1}^{2H}\|\Phi_k^x(\mathbf{M}_{t-H:t-1}) - \Phi_k^x(\tilde{\mathbf{M}}_{t-H:t-1})\|_2\sqrt{n}w_{\max} \\
&\le \sqrt{n}w_{\max}\sum_{k=1}^{2H}\|\sum_{i=1}^{H}A_{\mathsf{K}}^{i-1}B(M_{t-i}[k-i] - \tilde{M}_{t-i}[k-i])\mathbf{1}_{(1\le k-i\le H)}\|_2 \\
&\le \sqrt{n}w_{\max}\sum_{k=1}^{2H}\sum_{i=1}^{H}\kappa(1-\gamma)^{i-1}\kappa_B\|M_{t-i}[k-i] - \tilde{M}_{t-i}[k-i]\|_2\mathbf{1}_{(1\le k-i\le H)} \\
&= \sqrt{n}w_{\max}\kappa\kappa_B\sum_{i=1}^{H}(1-\gamma)^{i-1}\sum_{j=1}^{H}\|M_{t-i}[j] - \tilde{M}_{t-i}[j]\|_2
\end{aligned}$$

and

$$\begin{aligned}
\|\tilde{u}_t - \tilde{\tilde{u}}_t\|_2 &= \|-\mathsf{K}\tilde{x}_t + \mathsf{K}\tilde{\tilde{x}}_t + \sum_{i=1}^{H}M_t[i]w_{t-i} - \sum_{i=1}^{H}\tilde{M}_t[i]w_{t-i}\|_2 \\
&\le \kappa\|\tilde{x}_t - \tilde{\tilde{x}}_t\|_2 + \sqrt{n}w_{\max}\sum_{i=1}^{H}\|M_t[i] - \tilde{M}_t[i]\|_2
\end{aligned}$$

$$\leq \sqrt{n} w_{\max} \kappa^2 \kappa_B \sum_{i=0}^{H} (1-\gamma)^{\max(i-1,0)} \sum_{j=1}^{H} \|M[j]_{t-i} - \tilde{M}[j]_{t-i}\|_2$$

Consequently, by Assumption 6.4 and Lemma 6.2, we can prove the first bound in the lemma's statement below.

$$\begin{aligned}
|f_t(\mathbf{M}_{t-H:t-1}) - f_t(\tilde{\mathbf{M}}_{t-H:t-1})| &= |c_t(\tilde{x}_t, \tilde{u}_t) - c_t(\tilde{\tilde{x}}_t, \tilde{\tilde{u}}_t)| \\
&\leq Gb(\|\tilde{x}_t - \tilde{\tilde{x}}_t\|_2 + \|\tilde{u}_t - \tilde{\tilde{u}}_t\|_2) \\
&\leq Gb(1+\kappa)\sqrt{n} w_{\max} \kappa \kappa_B \sum_{i=0}^{H} (1-\gamma)^{\max(i-1,0)} \sum_{j=1}^{H} \|M[j]_{t-i} - \tilde{M}[j]_{t-i}\|_2.
\end{aligned}$$

□

Proof of Lemma 6.5. The regret bound on Part ii is a direct consequence of the OGD's regret bound in [14]. Hence, we only need to prove the gradient bound on $\mathring{f}_t(\mathbf{M}_t)$. Define a set $\mathbb{M}_{out,H} = \{\mathbf{M} : \|M[k]\|_\infty \leq 4\kappa\sqrt{n}(1-\gamma)^{k-1}\}$, whose interior contains $\mathbb{M}_H$. Similar to Lemma B.2, we can show $\mathring{f}_t(\mathbf{M} + \Delta\mathbf{M}) - \mathring{f}_t(\mathbf{M}) \leq \Theta(b\sqrt{n} w_{\max} \sum_{i=0}^{H} (1-\gamma)^{\max(i-1,0)} \sum_{j=1}^{H} \|\Delta M[j]\|_2)$ for any $\mathbf{M} \in \mathbb{M}_H$ and $\mathbf{M} + \Delta\mathbf{M} \in \mathbb{M}_{out,H}$.

By the definition of the operator's norm, we have

$$\begin{aligned}
\|\nabla \mathring{f}_t(\mathbf{M})\|_F &= \sup_{\Delta\mathbf{M} \neq 0, \mathbf{M}+\Delta\mathbf{M} \in \mathbb{M}_{out,H}} \frac{\langle \nabla \mathring{f}_t(\mathbf{M}), \Delta\mathbf{M} \rangle}{\|\Delta\mathbf{M}\|_F} \\
&\leq \sup_{\Delta\mathbf{M} \neq 0, \mathbf{M}+\Delta\mathbf{M} \in \mathbb{M}_{out,H}} \frac{\mathring{f}_t(\mathbf{M} + \Delta\mathbf{M}) - \mathring{f}_t(\mathbf{M})}{\|\Delta\mathbf{M}\|_F} \\
&\leq \sup_{\Delta\mathbf{M} \neq 0, \mathbf{M}+\Delta\mathbf{M} \in \mathbb{M}_{out,H}} \frac{\Theta(b\sqrt{n} w_{\max} \sum_{i=0}^{H} (1-\gamma)^{\max(i-1,0)} \sum_{j=1}^{H} \|\Delta M[j]\|_2)}{\|\Delta\mathbf{M}\|_F} \\
&\leq \sup_{\Delta\mathbf{M} \neq 0, \mathbf{M}+\Delta\mathbf{M} \in \mathbb{M}_{out,H}} \frac{\Theta(b\sqrt{n} w_{\max} \sum_{i=0}^{H} (1-\gamma)^{\max(i-1,0)} \sqrt{H} \|\Delta\mathbf{M}\|_F)}{\|\Delta\mathbf{M}\|_F} \\
&\leq \Theta(b\sqrt{n} w_{\max} \sqrt{H} \frac{1+\gamma}{\gamma})
\end{aligned}$$

□

B.1.3 Proof of Lemma 6.9

We first provide an explicit expression for Γ_ϵ and then prove the bounds on Γ_ϵ based on the explicit expression.

Lemma B.3. *For any $\epsilon \in \mathbb{R}$, $\mathbf{M} \in \Omega_\epsilon$ if and only if there exist $\{Y^x_{i,k,l}\}_{(1\le i\le k_x, 1\le k\le 2H, 1\le l\le n)}$, $\{Y^u_{j,k,l}\}_{(1\le j\le k_u, 1\le k\le 2H, 1\le l\le n)}$, $\{Z[i]_{k,j}\}_{(1\le i\le H, 1\le k\le m, 1\le j\le n)}$ such that*

$$\begin{cases} \sum_{k=1}^{2H}\sum_{l=1}^{n} Y^x_{i,k,l} w_{\max} \le d_{x,i} - \epsilon, \ \forall 1 \le i \le k_x \\ \sum_{k=1}^{2H}\sum_{l=1}^{n} Y^u_{j,k,l} w_{\max} \le d_{u,j} - \epsilon, \ \forall 1 \le j \le k_u \\ \sum_{j=1}^{n} Z[i]_{k,j} \le 2\sqrt{n}\kappa^2(1-\gamma)^{i-1}, \forall 1 \le i \le H, 1 \le k \le m \\ -Y^x_{i,k,l} \le (D^\top_{x,i}\mathring{\Phi}^x_k(\mathbf{M}))_l \le Y^x_{i,k,l}, \quad \forall i,k,l \\ -Y^u_{j,k,l} \le (D^\top_{u,j}\mathring{\Phi}^u_k(\mathbf{M}))_l \le Y^u_{j,k,l} \quad \forall i,k,l \\ -Z[i]_{k,j} \le M[i]_{k,j} \le Z[i]_{k,j}, \quad \forall i,k,j. \end{cases}$$

Let $\vec{W}$ denote the vector containing the elements of $\mathbf{M}, \boldsymbol{Y}^x = \{Y^x_{i,k,l}\}, \boldsymbol{Y}^u = \{Y^u_{j,k,l}\}, \boldsymbol{Z} = \{Z[i]_{k,j}\}$. Thus, the constraints above can be written as $\Gamma_\epsilon = \{\vec{W} : C\vec{W} \le h_\epsilon\}$.

Since Lemma B.3 holds for any $\epsilon \in \mathbb{R}$, we can similarly define $\Gamma_{-\epsilon_1-\epsilon_3} = \{\vec{W} : C\vec{W} \le h_{-\epsilon_1-\epsilon_3}\}$ which is equivalent to $\Omega_{-\epsilon_1-\epsilon_3}$. Lemma B.3 is based on a standard reformulation method in constrained optimization to handle inequalities involving absolute values so the proof is omitted.

Proof of (i). Firstly, notice that $\sum_{j=1}^{n}(M_{k,j}[i])^2 \le \sum_{j=1}^{n}(Z_{k,j}[i])^2 \le (\sum_{j=1}^{n} Z_{k,j}[i])^2 \le 4n\kappa^4(1-\gamma)^{2i-2}$. Then,

$$\sum_{k=1}^{m}\sum_{i=1}^{H}\sum_{j=1}^{n}(M_{k,j}[i])^2 \le \sum_{k=1}^{m}\sum_{i=1}^{H}\sum_{j=1}^{n}(Z_{k,j}[i])^2 \le 4nm\kappa^4\frac{1}{2\gamma-\gamma^2}$$

Similarly, by the first two constraints in Lemma B.3 and by the definition of $\Gamma_{-\epsilon_1-\epsilon_3}$, we have $\sum_{k=1}^{2H}\sum_{l=1}^{n}(Y^x_{i,k,l})^2 \le (d_{x,i}+\epsilon_1+\epsilon_3)^2/w_{\max}^2 \le (d_{x,i}+\epsilon_F)^2/w_{\max}^2$ and $\sum_{k=1}^{2H}\sum_{l=1}^{n}(Y^u_{j,k,l})^2 \le (d_{u,j}+\epsilon_1+\epsilon_3)^2/w_{\max}^2 \le (d_{u,j}+\epsilon_F)^2/w_{\max}^2$. Therefore, $\sum_{i=1}^{k_x}\sum_{k=1}^{2H}\sum_{l=1}^{n}(Y^x_{i,k,l})^2 \le \sum_{i=1}^{k_x}(d_{x,i}+\epsilon_F)^2/w_{\max}^2$, and $\sum_{j=1}^{k_u}\sum_{k=1}^{2H}\sum_{l=1}^{n}(Y^u_{j,k,l})^2 \le \sum_{j=1}^{k_u}(d_{u,j}+\epsilon_F)^2/w_{\max}^2$. Consequently,

$$\begin{aligned}&\|\mathbf{M}\|_F^2+\|\boldsymbol{Y}^x\|_F^2+\|\boldsymbol{Y}^u\|_F^2+\|\boldsymbol{Z}\|_F^2\\ \le\ &\frac{8nm\kappa^4}{2\gamma-\gamma^2}+\frac{\sum_{i=1}^{k_x}(d_{x,i}+\epsilon_F)^2+\sum_{j=1}^{k_u}(d_{u,j}+\epsilon_F)^2}{w_{\max}^2}=\delta_1^2\end{aligned}$$

where $\delta_1=\Theta(\sqrt{mn}+\sqrt{k_c})$ by the boundedness of ϵ_F, d_x, d_u. (Although δ_1 depends linearly on $1/w_{\max}$, we will show $L=TG_f$ and G_f is quadratic on $w_{\max}$ by Lemma 6.5, hence, $L\delta_1$ is still linear with $w_{\max}$.)

Proof of (ii). Since the gradient of $\mathring{f}_t(\mathbf{M})$ is bounded by $G_f=\Theta(\sqrt{n^3mH^3})$, the gradient of $\sum_{t=0}^{T}\mathring{f}_t(\mathbf{M})$ is bounded by $LG_f=\Theta(T\sqrt{n^3mH^3})$.

Proof of (iii). Notice that the differences between Γ_ϵ and $\Gamma_{-\epsilon_1-\epsilon_3}$ come from the first two lines of the right-hand-side of inequalities in Lemma B.3, which is $\epsilon+\epsilon_1+\epsilon_3$ in total.

Proof of (iv). From the proof of Theorem 6.1, we know that $\mathbf{M}(K_F)\in\Omega_{\epsilon_F-\epsilon_1-\epsilon_3}\subseteq\Omega_\epsilon$. Therefore, there exist corresponding $\boldsymbol{Y}^x(K_F),\boldsymbol{Y}^u(K_F),\boldsymbol{Z}(K_F)$ such that $\vec{W}^\circ=\mathrm{vec}(\mathbf{M}(K_F),\boldsymbol{Y}^x(K_F),\boldsymbol{Y}^u(K_F),\boldsymbol{Z}(K_F))\in\Gamma_{\epsilon_F-\epsilon_1-\epsilon_3}\subseteq\Gamma_\epsilon$. Therefore, $\min_{\{i:\Delta_i>0\}}(h_{-\epsilon_1-\epsilon_3}-C\vec{W}^\circ)_i\ge\epsilon_1+\epsilon_3-(-\epsilon_F+\epsilon_1+\epsilon_3)=\epsilon_F$.

B.2 Proof for Chapter 7

B.2.1 Proofs of Theorem 7.1 and Corollary 7.1

Our proof of Theorem 7.1 relies on a recently developed least square estimation error bound (Theorem 2.4 in [65]) for general time series satisfying a block matingale small-ball (BMSB) condition (Definition 2.1 [65]). In [63, 144], only linear policies are considered and shown to satisfy the BMSB condition. Our contribution is to show that BMSB still holds for general (even nonlinear) policies with bounded trajectories.

Proof of Theorem 7.1. Define $\mathscr{F}_t = \{w_0, \dots, w_{t-1}, \eta_0, \dots, \eta_t\}$ To use Theorem 2.4 in [65], we need to verify the three conditions. Condition 1): $x_{t+1} = \xi_* z_t + w_t$, and $w_t \mid \mathscr{F}_t$ is w_t which is mean 0 and σ_{sub}^2-sub-Gaussian by Assumption 7.3. Condition 3): notice that $\upsilon_{\max}(z_t z_t^\top) \le \operatorname{trace}(z_t z_t^\top) = \|z_t\|_2^2 \le b_x^2 + b_u^2$. Therefore, we can define $\bar{\Gamma} = (b_x^2 + b_u^2) I_{n+m}$, and then $\mathbb{P}(\sum_{t=1}^T z_t z_t^\top \not\preceq T\bar{\Gamma}) = 0 \le \delta$. The BMSB condition is verified below.

Lemma B.4 (Verification of BMSB condition)**.** *Define $\mathscr{F}_t^m = \mathscr{F}(w_0, \dots, w_{t-1}, \eta_0, \dots, \eta_{t-1})$ and consider controller $u_t = \pi_t(\mathscr{F}_t^m) + \eta_t$. Under the conditions in Theorem 7.1, $\{z_t\}_{t\ge 0}$ satisfies the $(1, s_z^2 I_{n+m}, p_z)$-BMSB condition, where p_z, s_z are defined in Theorem 7.1.*

Proof of Lemma B.4. Note that $z_t \in \mathscr{F}_t$ is by definition. Next,

$$z_{t+1} \mid \mathscr{F}_t = \begin{bmatrix} x_{t+1} \\ u_{t+1} \end{bmatrix} \mid \mathscr{F}_t = \begin{bmatrix} \xi_* z_t + w_t \mid \mathscr{F}_t \\ \pi_{t+1}(\mathscr{F}_{t+1}^m) + \eta_{t+1} \mid \mathscr{F}_t \end{bmatrix},$$

where $\mathscr{F}_{t+1}^m = \mathscr{F}(w_0, \dots, w_t, \eta_0, \dots, \eta_t)$. Conditioning on $\mathscr{F}_t$, the variable $\xi_* z_t$ is determined, but the variable $\pi_{t+1}(\mathscr{F}_{t+1}^m)$ is still random due to random w_t. For the rest of the proof, we always condition on $\mathscr{F}_t$ so we omit $\cdot \mid \mathscr{F}_t$ for simplicity.

Consider any $\lambda = (\lambda_1^\top, \lambda_2^\top)^\top \in \mathbb{R}^{m+n}$, where $\lambda_1 \in \mathbb{R}^n$, $\lambda_2 \in \mathbb{R}^m$, $\|\lambda\|_2^2 = \|\lambda_1\|_2^2 + \|\lambda_2\|_2^2 = 1$. Define $k_0 = \max(2/\sqrt{3}, 4b_u/s_w)$. We consider three cases: (i) when $\|\lambda_2\|_2 \le 1/k_0$ and $\lambda_1^\top \xi_* z_t \ge 0$, (ii) when $\|\lambda_2\|_2 \le 1/k_0$ and $\lambda_1^\top \xi_* z_t < 0$, (iii) when $\|\lambda_2\|_2 > 1/k_0$. We will show $\mathbb{P}(|\lambda^\top z_{t+1}| \ge s_z) \ge p_z$ in all three cases. Consequently, by Definition 2.1 in [65], we have $\{z_t\}$ is $(1, s_z^2 I, p_z)$-BMSB.

Case 1: when $\|\lambda_2\|_2 \le 1/k_0$ and $\lambda_1^\top \xi_* z_t \ge 0$. Notice that $\lambda_1^\top w_t \le \lambda_1^\top (w_t + \xi_* z_t) \le |\lambda_1^\top (w_t + \xi_* z_t)| = |\lambda^\top z_{t+1} - \lambda_2^\top u_{t+1}| \le |\lambda^\top z_{t+1}| + |\lambda_2^\top u_{t+1}| \le |\lambda^\top z_{t+1}| + \|\lambda_2\|_2 b_u \le |\lambda^\top z_{t+1}| + b_u/k_0 \le |\lambda^\top z_{t+1}| + s_w/4$, where the last inequality uses $k_0 \ge 4b_u/s_w$.

Further, notice that $k_0 \ge 2/\sqrt{3}$, so $\|\lambda_2\|_2^2 \le 1/k_0^2 \le 3/4$ and $\|\lambda_1\|_2 \ge 1/2$. Therefore, $\mathbb{P}(\lambda_1^\top w_t \ge s_w/2) = \mathbb{P}(\frac{\lambda_1^\top w_t}{\|\lambda_1\|_2} \ge \frac{s_w}{2\|\lambda_1\|_2}) \ge \mathbb{P}(\frac{\lambda_1^\top w_t}{\|\lambda_1\|_2} \ge s_w) = p_w$. Then, we have $\mathbb{P}(|\lambda^\top z_{t+1}| \ge s_z) \ge \mathbb{P}(|\lambda^\top z_{t+1}| \ge s_w/4) = \mathbb{P}(|\lambda^\top z_{t+1}| + s_w/4 \ge s_w/2) \ge \mathbb{P}(\lambda_1^\top w_t \ge s_w/2) \ge p_w$.

Case 2: when $\|\lambda_2\|_2 \le 1/k_0$ and $\lambda_1^\top \xi_* z_t < 0$. This is the same as Case 1.

Case 3: when $\|\lambda_2\|_2 > 1/k_0$. Define $v = \bar{\eta} s_\eta / k_0 = \min(\sqrt{3}\bar{\eta} s_\eta/2, s_w \bar{\eta} s_\eta/(4b_u))$. Define events $\Omega_1^\lambda = \{w_t \in \mathbb{R}^n \mid \lambda_1^\top (w_t + \xi_* z_t) + \lambda_2^\top (\pi_{t+1}(\mathscr{F}_{t+1}^m)) \ge 0\}$ and $\Omega_2^\lambda = \{w_t \in \mathbb{R}^n \mid \lambda_1^\top (w_t + \xi_* z_t) + \lambda_2^\top (\pi_{t+1}(\mathscr{F}_{t+1}^m)) < 0\}$. Notice that $\mathbb{P}(w_t \in \Omega_1^\lambda) + \mathbb{P}(w_t \in \Omega_2^\lambda) = 1$.

$$
\begin{aligned}
\mathbb{P}(|\lambda^\top z_{t+1}| \ge s_z) &\ge \mathbb{P}(|\lambda^\top z_{t+1}| \ge v) = \mathbb{P}(\lambda^\top z_{t+1} \ge v) + \mathbb{P}(\lambda^\top z_{t+1} \le -v) \\
&\ge \mathbb{P}(\lambda^\top z_{t+1} \ge v, w_t \in \Omega_1^\lambda) + \mathbb{P}(\lambda^\top z_{t+1} \le -v, w_t \in \Omega_2^\lambda) \\
&\ge \mathbb{P}(\lambda_2^\top \eta_{t+1} \ge v, w_t \in \Omega_1^\lambda) + \mathbb{P}(\lambda_2^\top \eta_{t+1} \le -v, w_t \in \Omega_2^\lambda) \\
&= \mathbb{P}(\lambda_2^\top \eta_{t+1} \ge v)\mathbb{P}(w_t \in \Omega_1^\lambda) + \mathbb{P}(\lambda_2^\top \eta_{t+1} \le -v)\mathbb{P}(w_t \in \Omega_2^\lambda) \ge p_\eta
\end{aligned}
$$

where the last inequality is by the following result: $\mathbb{P}(\lambda_2^\top \eta_{t+1} \le -v) = \mathbb{P}(\lambda_2^\top \eta_{t+1} \ge v) = \mathbb{P}(\lambda_2^\top \eta_{t+1}/\|\lambda_2\|_2 \ge v/\|\lambda_2\|_2) = \mathbb{P}(\lambda_2^\top \tilde{\eta}_{t+1}/\|\lambda_2\|_2 \ge v/(\|\lambda_2\|_2 \bar{\eta})) \ge \mathbb{P}(\lambda_2^\top \tilde{\eta}_{t+1}/\|\lambda_2\|_2 \ge k_0 v/(\bar{\eta})) = \mathbb{P}(\lambda_2^\top \tilde{\eta}_{t+1}/\|\lambda_2\|_2 \ge s_\eta) \ge p_\eta$. This completes the case 3. □

□

Proof of Corollary 7.1. The proof is straightforward by verifying $u_t \in \mathbb{U}$ by our algorithm design and $\|x_t\|_2 \le b_x = O(\sqrt{mn})$ as proved in the literature [42, 83]. □

B.2.2 Proofs of Lemma 7.2 and 7.3

The proof of Lemma 7.2 is straightforward by definitions. We focus on the proof of Lemma 7.3 below. We also provide the epsilon functions with explicit factors below.

Definition B.1 (Formulas of the constraint-tightening terms)**.**

(i) $\epsilon_\xi(r) = \|D_x\|_\infty z_{\max}\kappa/\gamma \cdot r + 5\kappa^4\kappa_B\|D_x\|_\infty w_{\max}/\gamma^3 \cdot \sqrt{mn}r = O(\sqrt{mn}r)$

(ii) $\epsilon_{\eta,x} = \|D_x\|_\infty \kappa\kappa_B/\gamma\sqrt{m}\bar{\eta} = O(\sqrt{m}\bar{\eta})$

(iii) $\epsilon_H(H) = \|D_x\|_\infty \kappa x_{\max}(1-\gamma)^H = O((1-\gamma)^H)$

(iv) $\epsilon_v(\Delta_M, H) = \|D_x\|_\infty w_{\max}\kappa\kappa_B/\gamma^2\sqrt{mnH}\Delta_M = O(\sqrt{mnH}\Delta_M)$

(v) $\epsilon_{\eta,u} = \|D_u\|_\infty \bar{\eta} = O(\bar{\eta})$

where $z_{\max} = \sqrt{x_{\max}^2 + u_{\max}^2} \le x_{\max} + u_{\max}$.

Proof of Lemma 7.3. The proofs of the error terms (iii) and (iv) are similar to the unconstrained case so we omit them here. The bounds on (ii) and (v) are straightforward. We focus on the error term (i) below, which consists of two parts. The second part is bounded by $\|D_x \sum_{k=1}^{H_t} A_*^{k-1}(w_{t-k} - \hat{w}_{t-k})\|_\infty \le \|D_x\|_\infty \sum_{k=1}^{H_t} \|A_*^{k-1}(w_{t-k} - \hat{w}_{t-k})\|_\infty \le \|D_x\|_\infty \sum_{k=1}^{H_t} \|A_*^{k-1}(w_{t-k} - \hat{w}_{t-k})\|_2 \le \|D_x\|_\infty \sum_{k=1}^{H_t} \kappa(1-\gamma)^{k-1} r z_{\max} \le \|D_x\|_\infty \kappa/\gamma z_{\max} r$, where we use the following bound on disturbance estimation errors.

Lemma B.5 (Disturbance estimation errors)**.** *Consider* $\hat{w}_t = \Pi_{\mathbb{W}}(x_{t+1} - \hat{\xi} z_t)$ *and* $x_{t+1} = \xi_* z_t + w_t$. *Suppose* $\|z_t\|_2 \le b_z$ *and* $\|\xi_* - \hat{\xi}\|_F \le r$, *then* $\|w_t - \hat{w}_t\|_2 \le b_z r$.

The first part relies on the following perturbation result on function g_i^x.

Lemma B.6. *Under the conditions in Lemma 7.3, we have*

$$|g_i^x(\mathbf{M};\hat{\xi}) - g_i^x(\mathbf{M};\xi)| \le 5\kappa^4\kappa_B\|D_x\|_\infty w_{\max}/\gamma^3 \cdot \sqrt{mn}r =: \epsilon_{\hat{\xi}}(r) = O(\sqrt{mn}r).$$

Proof. Firstly, notice that

$$\begin{aligned}
|g_i^x(\mathbf{M};\hat{\xi}) - g_i^x(\mathbf{M};\xi)| &= |\sum_{k=1}^{2H} \|D_{x,i}^\top \Phi_k^x(\mathbf{M};\hat{\xi})\|_1 - \|D_{x,i}^\top \Phi_k^x(\mathbf{M};\xi)\|_1| w_{\max} \\
&\le \sum_{k=1}^{2H} |\|D_{x,i}^\top \Phi_k^x(\mathbf{M};\hat{\xi})\|_1 - \|D_{x,i}^\top \Phi_k^x(\mathbf{M};\xi)\|_1| w_{\max} \\
&\le \sum_{k=1}^{2H} \|D_{x,i}^\top \Phi_k^x(\mathbf{M};\hat{\xi}) - D_{x,i}^\top \Phi_k^x(\mathbf{M};\xi)\|_1 w_{\max} \\
&\le \sum_{k=1}^{2H} \|D_x\|_\infty \|\Phi_k^x(\mathbf{M};\hat{\xi}) - \Phi_k^x(\mathbf{M};\xi)\|_\infty w_{\max}.
\end{aligned}$$

Then, it suffices to bound $\sum_{k=1}^{2H} \|\Phi_k^x(\mathbf{M};\hat{\xi}) - \Phi_k^x(\mathbf{M};\xi)\|_\infty$.

$$\begin{aligned}
&\|\Phi_k^x(\mathbf{M};\hat{\xi}) - \Phi_k^x(\mathbf{M};\xi)\|_\infty \\
&\le \|\hat{A}^{k-1} - A^{k-1}\|_\infty \mathbf{1}_{(k\le H)} + \sum_{i=1}^{H} \|(\hat{A}^{i-1}\hat{B} - A^{i-1}B) M_{t-i}[k-i]\|_\infty \mathbf{1}_{(1\le k-i\le H)} \\
&\le \sqrt{n}\|\hat{A}^{k-1} - A^{k-1}\|_2 \mathbf{1}_{(k\le H)} + \sqrt{m}\sum_{i=1}^{H} \|\hat{A}^{i-1}\hat{B} - A^{i-1}B\|_2 2\sqrt{n}\kappa^2(1-\gamma)^{k-i-1}\mathbf{1}_{(1\le k-i\le H)}
\end{aligned}$$

Then, by perturbation bounds $\|A^k - \hat{A}^k\|_2 \le k\kappa^2(1-\gamma)^{k-1} r\mathbf{1}_{(k\ge 1)}$, $\|A^k B - \hat{A}^k\hat{B}\|_2 \le k\kappa^2\kappa_B(1-\gamma)^{k-1} r\mathbf{1}_{(k\ge 1)} + \kappa(1-\gamma)^k r$, we complete the proof. □

□

B.2.3 Proofs of Theorem 7.2 and Theorem 7.3

For ease of notation, we define a new representation of policy sets by

$$\Omega_H(\xi,\epsilon) = \{\mathbf{M} \in \mathbb{M}_H : g_i^x(\mathbf{M};\xi) \le d_{x,i} - \epsilon_x, g_j^u(\mathbf{M};\xi) \le d_{u,j} - \epsilon_u, \forall\, i,j\}, \tag{B.1}$$

where $\epsilon = (\epsilon_x, \epsilon_u)$. Notice that the set $\Omega(\Xi, H, \bar{\eta}, \Delta_M)$ in (7.5) satisfies $\Omega(\Xi, H, \bar{\eta}, \Delta_M) = \Omega_H(\hat{\xi}, \epsilon)$, for $\epsilon_x = \epsilon_\xi(r) + \epsilon_{\eta,x}(\bar{\eta}) + \epsilon_H(H) + \epsilon_v(\Delta_M, H), \epsilon_u = \epsilon_{\eta,u}(\bar{\eta})$.

Proof of Theorem 7.2. Firstly, we note that feasibility is guaranteed if $\Omega_\dagger^{(e)} \neq \varnothing$ (Line 3), $\Omega_\dagger^{(e)} \cap \Omega^{(e-1)} \neq \varnothing$ (Line 4), $\Omega^{(e)} \neq \varnothing$ (Line 8), and $\Omega_\dagger^{(e)} \cap \Omega^{(e)} \neq \varnothing$ (Line 9) for all $e \geq 0$. Therefore, it suffices to construct a policy $\mathbf{M}_F$ that belongs to $\Omega_\dagger^{(e)} \cap \Omega^{(e)}$ for all $e \geq 0$. Specifically, we construct $\mathbf{M}_F$ by $\mathbf{M}(K_F)$ according to Lemma 6.4. By Corollary 6.2, we have $\mathbf{M}_F \in \Omega_{H^{(0)}}(\xi_*, \epsilon_F - \epsilon_P(H^{(0)}))$. Then, under the conditions of Theorem 7.2, by using our definitions in (7.5) and Lemma 7.3, we can show $\mathbf{M}_F \in \Omega_\dagger^{(e)} \cap \Omega^{(e)}$ for all $e \geq 0$, which completes the proof. □

Proof of Theorem 7.3. The control constraint satisfaction is straightforward. Hence, we focus on state constraint satisfaction here. Define an event

$$\mathscr{E}_{\text{safe}} = \{\xi_* \in \bigcap_{e=0}^{N-1} \Xi^{(e)}\}.$$

Notice that $\mathbb{P}(\mathscr{E}_{\text{safe}}) = 1 - \mathbb{P}(\mathscr{E}_{\text{safe}}^c) \geq 1 - \sum_{e=0}^{N} \mathbb{P}(\xi_* \notin \Xi^{(e)}) \geq 1 - \sum_{e=1}^{N} p/(2e^2) \geq 1 - p$ by Corollary 7.1 and $\xi_* \in \Xi^{(0)} = \Xi_{\text{ini}}$. In the following, we will condition on event $\mathscr{E}_{\text{safe}}$ and show $x_t \in \mathbb{X}$ for all $t \geq 0$ under this event. We prove this by induction. When $t \leq 0$, we have $x_0 = 0 \in \mathbb{X}$. Suppose at stage $t \geq 1$, we have $x_s \in \mathbb{X}$ for all $s < t$. We will show $x_t \in \mathbb{X}$ below. We discuss three possible cases based on the value of t. We introduce some notations for our case-by-case discussion: let $W_1^{(e)}, W_2^{(e)}$ denote the W_1, W_2 defined in Algorithm 7.2 during the transition in Phase 1, and let $\tilde{W}_1^{(e)}, \tilde{W}_2^{(e)}$ denote the W_1, W_2 defined in Algorithm 7.2 during the transition in Phase 2.

Case 1: when $T^{(e)} \leq t \leq T^{(e)} + W_1^{(e)} - 1$. By Algorithm 7.1 and Algorithm 7.2, we can verify that the conditions of Lemma 7.3 are satisfied with $r = r^{(e)}, \bar{\eta} = 0, H =$

$H^{(e-1)}, \Delta_M = \Delta_M^{(e-1)}$. Let $\hat{\xi}_t^g = \hat{\xi}^{(e)}$. Notice that $\mathbf{M}_t \in \Omega^{(e-1)}$, so the constraints imposed by our algorithms, i.e. $g_i^x(\mathbf{M}_t; \hat{\xi}^{(e)}) \leq d_{x,i} - \epsilon_H(H^{(e-1)}) - \epsilon_v(\Delta_M^{(e-1)}, H^{(e-1)}) - \epsilon_\xi(r^{(e)})$, guarantee $x_t \in \mathbb{X}$ according to Lemma 7.2.

Case 2: when $T^{(e)}+W_1^{(e)} \leq t \leq t_1^{(e)}+T_D^{(e)}+\tilde{W}_1^{(e)}-1$. By Algorithm 7.1 and Algorithm 7.2, we can verify that the conditions of Lemma 7.3 are satisfied with $r = r^{(e)}, \bar{\eta} = \bar{\eta}^{(e)}, H = H^{(e)}, \Delta_M = \Delta_M^{(e)}$. Let $\hat{\xi}_t^g = \hat{\xi}^{(e)}$. Notice that $\mathbf{M}_t \in \Omega^{(e)}$, so the constraints imposed by our algorithms, i.e. $g_i^x(\mathbf{M}_t; \hat{\xi}^{(e)}) \leq d_{x,i} - \epsilon_H(H^{(e)}) - \epsilon_v(\Delta_M^{(e)}, H^{(e)}) - \epsilon_\xi(r^{(e)}) - \epsilon_{\eta,x}(\bar{\eta}^{(e)})$, guarantee $x_t \in \mathbb{X}$ according to Lemma 7.2.

Case 3: when $t_1^{(e)} + T_D^{(e)} + \tilde{W}_1^{(e)} \leq t \leq T^{(e+1)} - 1$. Case 3 is similar to the cases above.

In summary, we have shown that $x_t \in \mathbb{X}$ for all $t \geq 0$ by induction. □

B.2.4 Proof of Theorem 7.4

B.2.4.1 Preparations

This subsubsection provides some preparation steps before our main regret analysis. To start with, we note that the parameters in Theorem 7.4 can be verified to satisfy the conditions in Theorem 7.2 and 7.3. Therefore, the feasibility and constraint satisfaction guarantees hold.

Useful facts. Here, we introduce useful facts for our regret analysis.

Lemma B.7. *Consider $T^{(e)} = 2^{e-1}T^{(1)}$ and let N denote the number of episodes in T stages. Then, we have $N \leq O(\log T)$. Further, for any $\alpha > 0$, we have $\sum_{e=1}^{N}(T^{(e)})^\alpha = O(T^\alpha)$.*

Lemma B.8. *When the event $\mathscr{E}_{safe}$ is true, $l(x_t, u_t) \le \|Q\|_2 x_{\max}^2 + \|R\|_2 u_{\max}^2 = O(1)$. Consequently, the single-stage regret $l(x_t, u_t) - J^* \le O(1)$.*

Similar to the perturbation bound on $\mathring{g}_i^x(\mathbf{M}; \xi) - \mathring{g}_i^x(\mathbf{M}; \xi)$ in Lemma B.6, we have the perturbation bound on $\mathring{f}$ below.

Lemma B.9 (Perturbation bound on $\mathring{f}$ with respect to ξ)**.** *For any $H \ge 1$, $\mathbf{M} \in \mathbb{M}_H$, any $\xi, \hat{\xi} \in \Xi_{ini}$ with $\|\xi - \hat{\xi}\|_F \le r$, when $H \ge \log(2\kappa)/\log((1-\gamma)^{-1})$,*

$$|\mathring{f}(\mathbf{M}; \xi) - \mathring{f}(\mathbf{M}; \hat{\xi})| \le O(mnr)$$

Notation definitions. In the following, we introduce some useful notations to simplify the representation of our analysis

Firstly, for every episode e, we divide the stages into two segments.

$$\mathscr{T}_1^{(e)} = \{T^{(e)} \le t \le t_2(e) + H^{(e)} - 1\}, \quad \mathscr{T}_2^{(e)} = \{t_2(e) + H^{(e)} \le t \le T^{(e+1)} - 1\}. \quad \text{(B.2)}$$

Further, we introduce two policy sets below. We will use the notation (B.1) in Appendix B.2.3 to define the sets.

$$\begin{aligned}
&\Omega^{(e)} = \Omega(\Xi^{(e+1)}, H^{(e)}, 0, \Delta_M^{(e)}) = \Omega_{H^{(e)}}(\xi^{(e+1)}, \tilde{\epsilon}^{(e+1)}),\\
&\qquad \text{where } \tilde{\epsilon}_x^{(e+1)} = \epsilon_\xi(r^{(e+1)}) + \epsilon_H(H^{(e)}) + \epsilon_v(\Delta_M^{(e)}, H^{(e)}), \text{ and } \tilde{\epsilon}_u^{(e+1)} = 0.\\
&\Omega_*^{(e)} = \Omega_H^{(e)}(\xi_*, \epsilon_*^{(e)}), \quad \text{where } \epsilon_{*,x}^{(e)} = \epsilon_H(H^{(e)}), \text{ and } \epsilon_{*,u}^{(e)} = 0.
\end{aligned}$$

Notice that $\Omega^{(e)}$ is used for CCE computation in Phase 2 of Algorithm 7.1 and $\Omega_*^{(e)}$ represents a policy set with known true system ξ_*.

Next, we define a list of policies that will be used to divide the regret. The policies are CCE solutions with different policy sets and different cost functions.

$$\mathbf{M}_*^{(e)} = \arg\min \mathring{f}(\mathbf{M}; \xi^{(e+1)}), \quad \text{subject to } \mathbf{M} \in \Omega^{(e)}, \quad \text{(B.3)}$$

$$\mathbf{M}_{\alpha}^{(e)} = \arg\min \mathring{f}(\mathbf{M}; \xi_*), \quad \text{subject to } \mathbf{M} \in \Omega^{(e)}, \tag{B.4}$$

$$\mathbf{M}_{H^{(e)}}^{*} = \arg\min \mathring{f}(\mathbf{M}; \xi_*), \quad \text{subject to } \mathbf{M} \in \Omega_*^{(e)}. \tag{B.5}$$

Notice that $\mathbf{M}_*^{(e)}$ is the policy implemented in Phase 2 of Algorithm 7.1.

Finally, we let $\tilde{x}_t, \tilde{u}_t$ denote the approximate state and action generated by the policy sequence determined by Algorithm 7.1 when the disturbances are accurately known, i.e.,

$$\tilde{x}_t = \sum_{k=1}^{2H^{(e)}} \Phi_k^x(\mathbf{M}_*^{(e)}; \xi_*) w_{t-k}, \quad \tilde{u}_t = \sum_{k=1}^{H^{(e)}} M_*^{(e)}[k] w_{t-k}.$$

B.2.4.2 Regret analysis

Firstly, notice that the total number of stages in $\mathscr{T}_1^{(e)}$ for all e are bounded by $O(T^{2/3})$ according to Lemma B.7. When $\mathscr{E}_{\text{safe}}$ holds, the single-stage regret is bounded by $O(1)$ by Lemma B.8. Thus, the stages in $\mathscr{T}_1^{(e)}$ for all e give rise to a regret bound $O(T^{2/3})$.

Next, we focus on the regret in $\mathscr{T}_2^{(e)}$ for all e, which is further divided into five parts by the policies and $\tilde{x}_t, \tilde{u}_t$ introduced above.

$$\begin{aligned}
\sum_{e=0}^{N-1} \sum_{t \in \mathscr{T}_2^{(e)}} (l(x_t, u_t) - J^*) = & \underbrace{\sum_{e=0}^{N-1} \sum_{t \in \mathscr{T}_2^{(e)}} (l(x_t, u_t) - l(\tilde{x}_t, \tilde{u}_t))}_{\text{Part i}} + \underbrace{\sum_{e=0}^{N-1} \sum_{t \in \mathscr{T}_2^{(e)}} (l(\tilde{x}_t, \tilde{u}_t) - \mathring{f}(\mathbf{M}_*^{(e)}; \xi_*))}_{\text{Part ii}} \\
& + \underbrace{\sum_{e=0}^{N-1} \sum_{t \in \mathscr{T}_2^{(e)}} (\mathring{f}(\mathbf{M}_*^{(e)}; \xi_*) - \mathring{f}(\mathbf{M}_{\alpha}^{(e)}; \xi_*))}_{\text{Part iii}} \\
& + \underbrace{\sum_{e=0}^{N-1} \sum_{t \in \mathscr{T}_2^{(e)}} (\mathring{f}(\mathbf{M}_{\alpha}^{(e)}; \xi_*) - \mathring{f}(\mathbf{M}_{H^{(e)}}^{*}; \xi_*))}_{\text{Part iv}}
\end{aligned}$$

$$\underbrace{+\sum_{e=0}^{N-1}\sum_{t\in\mathscr{T}_2^{(e)}}(\mathring{f}(\mathbf{M}^*_{H^{(e)}};\xi_*)-J^*)}_{\text{Part v}}$$

We will provide bounds on the five parts below.

Bound Part iii.

Lemma B.10 (Bound on Part iii)**.** *When $\mathscr{E}_{safe}$ holds, we have*

$$\sum_{e=0}^{N-1}\sum_{t\in\mathscr{T}_2^{(e)}}(\mathring{f}(\mathbf{M}^{(e)}_*;\xi_*)-\mathring{f}(\mathbf{M}^{(e)}_\alpha;\xi_*))\le\tilde{O}(mn\sqrt{mn+n^2}T^{2/3}).$$

Proof. To bound Part iii, notice that $\mathring{f}(\mathbf{M}^{(e)}_*;\xi_*)-\mathring{f}(\mathbf{M}^{(e)}_\alpha;\xi_*)=\mathring{f}(\mathbf{M}^{(e)}_*;\xi_*)-\mathring{f}(\mathbf{M}^{(e)}_*;\hat{\xi}^{(e+1)})+\mathring{f}(\mathbf{M}^{(e)}_*;\hat{\xi}^{(e+1)})-\mathring{f}(\mathbf{M}^{(e)}_\alpha;\hat{\xi}^{(e+1)})+\mathring{f}(\mathbf{M}^{(e)}_\alpha;\hat{\xi}^{(e+1)})-\mathring{f}(\mathbf{M}^{(e)}_\alpha;\xi_*)$.

Since $\mathbf{M}^{(e)}_\alpha=\arg\min_{\mathbf{M}\in\Omega^{(e)}}\mathring{f}(\mathbf{M};\xi_*)$ and $\mathbf{M}^{(e)}_*=\arg\min_{\mathbf{M}\in\Omega^{(e)}}\mathring{f}(\mathbf{M}^{(e)}_*;\xi^{(e+1)})$, we have $\mathring{f}(\mathbf{M}^{(e)}_*;\hat{\xi}^{(e+1)})-\mathring{f}(\mathbf{M}^{(e)}_\alpha;\hat{\xi}^{(e+1)})\le 0$. Therefore, we have $\mathring{f}(\mathbf{M}^{(e)}_*;\xi_*)-\mathring{f}(\mathbf{M}^{(e)}_\alpha;\xi_*)=\mathring{f}(\mathbf{M}^{(e)}_*;\xi_*)-\mathring{f}(\mathbf{M}^{(e)}_*;\hat{\xi}^{(e+1)})+\mathring{f}(\mathbf{M}^{(e)}_*;\hat{\xi}^{(e+1)})-\mathring{f}(\mathbf{M}^{(e)}_\alpha;\hat{\xi}^{(e+1)})+\mathring{f}(\mathbf{M}^{(e)}_\alpha;\hat{\xi}^{(e+1)})-\mathring{f}(\mathbf{M}^{(e)}_\alpha;\xi_*)\le O(mn\|\hat{\xi}^{(e+1)}-\xi_*\|_F)=O(mnr^{(e+1)})\le\tilde{O}(mn\sqrt{mn+n^2}(T^{(e+1)})^{-1/3})$ by Lemma B.9 and Corollary 7.1. The proof is completed by summing over e and applying Lemma B.7. □

Bound on Part iv. This is the dominating term of our regret bound.

Lemma B.11 (Bound on Part iv)**.** *When $\mathscr{E}_{safe}$ holds, we have*

$$\sum_{e=0}^{N-1}\sum_{t\in\mathscr{T}_2^{(e)}}(\mathring{f}(\mathbf{M}^{(e)}_\alpha;\xi_*)-\mathring{f}(\mathbf{M}^*_{H^{(e)}};\xi_*))\le\tilde{O}((n^2m^{1.5}+n^{2.5}m)\sqrt{mn+k_c}(T^{2/3}).$$

Proof. Notice that in the definitions of $\mathbf{M}^{(e)}_\alpha$ in (B.4) and $\mathbf{M}^*_{H^{(e)}}$ in (B.5), the objective functions are the same but the constraint sets are different. This suggests using the

perturbation results for constrained optimization in Proposition 6.1. However, the constraints considered in Proposition 6.1 share the same left-hand-sides, but $\Omega^{(e)}$ and $\Omega_*^{(e)}$ have different left-hand-sides due to different estimated systems. To handle this, we introduce two auxiliary sets below:

$$\bar{\Omega}_*^{(e)} = \Omega_{H^{(e)}}(\xi_*; \tilde{\epsilon}^{(e+1)} - \epsilon_{\hat{\xi}}(r^{(e+1)})), \quad \bar{\Omega}_e = \Omega_{H^{(e)}}(\hat{\xi}^{(e+1)}; \epsilon_*^{(e)} - \epsilon_{\hat{\xi}}(r^{(e+1)})),$$

where $\epsilon_{\hat{\xi}}(r)$ is introduced in Lemma B.6. By Lemma B.6, we have $\mathbf{M}_\alpha^{(e)} \in \bar{\Omega}_*^{(e)}$, and $\mathbf{M}_{H^{(e)}}^* \in \bar{\Omega}^{(e)}$. Then, define

$$\Omega_1 = \Omega^{(e)} \cap \bar{\Omega}_*^{(e)}, \quad \Omega_2 = \Omega_*^{(e)} \cap \bar{\Omega}^{(e)}$$

Notice that $\Omega_1 \cap \Omega_2 = \Omega^{(e)} \cap \Omega_*^{(e)}$. Now, we have

$$\mathbf{M}_\alpha^{(e)} = \arg\min_{\mathbf{M} \in \Omega_1} \mathring{f}(\mathbf{M}_*^{(e)}; \xi_*), \quad \mathbf{M}_{H^{(e)}}^* = \arg\min_{\mathbf{M} \in \Omega_2} \mathring{f}(\mathbf{M}; \xi_*)$$

Now, the constraints in the two optimization above share the same left-hand-side functions, so we can apply Proposition 6.1.[1] The gradient bound $G_f = O(n\sqrt{m}H^{(e)})$ and the diameter bound $d_{\Gamma_0} = O(\sqrt{mn} + \sqrt{k_c})$ can be proved in the same way as in Lemma 6.9. The non-emptiness of $\Omega_1 \cap \Omega_2$ can be verified by the conditions of this theorem. The difference on the right-hand-sides can be upper bounded by $\|\tilde{\epsilon}^{(e+1)} - \epsilon_*^{(e)}\|_\infty \leq \epsilon_{\hat{w}}^{(e+1)} + \epsilon_{\hat{\xi}}^{(e+1)} + \epsilon_v^{(e)} \leq O(\sqrt{mn}r^{(e+1)} + \sqrt{mnH^{(e)}}\Delta_M^{(e)}) \leq \tilde{O}((T^{(e+1)})^{-1/3}(\sqrt{m^2n^2 + n^3m} + \sqrt{mnH^{(e)}})$. By applying the bounds above and by Proposition 6.1, we have

$$\mathring{f}(\mathbf{M}_\alpha^{(e)}; \xi_*)s - \mathring{f}(\mathbf{M}_{H^{(e)}}^*; \xi_*)$$

[1] In Proposition 6.1, one constraint set is assumed to be contained by another, which is not necessarily the case here. But, the result can be easily extended to two intersected constraint sets by introducing an auxiliary set $\Omega_1 \cap \Omega_2$ and apply Proposition 6.1 twice: on the perturbation from $\Omega_1 \cap \Omega_2$ to Ω_1 and from $\Omega_1 \cap \Omega_2$ to Ω_2. The order of the bound remain the same.

$$\leq \tilde{O}\left((\sqrt{mn}+\sqrt{k_c})(n\sqrt{m}H^{(e)})((T^{(e+1)})^{-1/3}(\sqrt{m^2n^2+n^3m}+\sqrt{mnH^{(e)}})\right)$$
$$\leq \tilde{O}\left((\sqrt{mn}+\sqrt{k_c})(n\sqrt{m})((T^{(e+1)})^{-1/3}\sqrt{m^2n^2+n^3m}\right)$$
$$\leq \tilde{O}\left((n^2m^{1.5}+n^{2.5}m)\sqrt{mn+k_c}(T^{(e+1)})^{-1/3}\right)$$

The proof is completed by summing over e and by Lemma B.7. □

Bound Part v The bound on this part does not depend on T due to our sufficiently large choices of $H^{(e)}$, which is consistent with our discussions in Section 7.1.1.

Lemma B.12 (Bound on Part v)**.** *By our choice of $H^{(e)}$ in Theorem 7.3, we have*

$$\sum_{e=0}^{N-1}\sum_{t\in\mathscr{T}_2^{(e)}}(\mathring{f}(\mathbf{M}^*_{H^{(e)}};\xi_*)-J^*)=\tilde{O}(n\sqrt{m}\sqrt{mn+k_c}\sqrt{n})$$

The proof is the same as the proofs of Lemma 6.7 and Lemma 6.6 and is omitted here for brevity.

Bound Part ii Notice that this part is not a dominating term in the regret bound. The proof relies on a martingale concentration analysis and is very technical.

Lemma B.13 (Bound on Part ii)**.** *With probability $1-p$, Part ii $\leq \tilde{O}(mn\sqrt{T})$.*

The proof relies on the Azuma-Hoeffding Inequality below.

Proposition B.1 (Azuma-Hoeffding Inequality)**.** *$\{X_t\}_{t\geq 0}$ is a martingale with respect to $\{\mathscr{F}_t\}_{t\geq 0}$. If (i) $X_0=0$, (ii) $|X_t-X_{t-1}|\leq\sigma$ for any $t\geq 1$, then, for any $\delta\in(0,1)$, $|X_t|\leq\sqrt{2t}\sigma\sqrt{\log(2/\delta)}$ w.p. at least $1-\delta$.*

Proof of Lemma B.13. Firstly, we construct martingales by the following definitions.

$$\mathscr{T}_{2,h}^{(e)}=\{t\in\mathscr{T}_2^{(e)}:t\equiv h \mod (2H^{(e)})\}$$

$$=: \{t_h^{(e)} + 2H^{(e)}, \dots, t_h^{(e)} + 2H^{(e)} k_h^{(e)}\}, \quad \forall\, 0 \le h \le 2H^{(e)} - 1$$

$$q_t = l(\tilde{x}_t, \tilde{u}_t) - \mathring{f}(\mathbf{M}_*^{(e)}; \xi_*)$$

$$\tilde{q}_{h,j}^{(e)} = q_{t_h^{(e)} + j(2H^{(e)})}, \quad \forall\, 1 \le j \le k_h^{(e)}$$

$$S_{h,j}^{(e)} = \sum_{s=1}^{j} \tilde{q}_{h,s}^{(e)}, \quad \forall\, 0 \le j \le k_h^{(e)},$$

$$\mathscr{F}_{h,j}^{(e)} = \mathscr{F}_{t_h^{(e)} + j(2H^{(e)})}, \quad \forall\, 0 \le j \le k_h^{(e)}.$$

where $t_h^{(e)}$ and $k_h^{(e)}$ are introduced to enumerate all the elements in $\mathscr{T}_{2,h}^{(e)}$. Notice that $t_h^{(e)} \ge t_2^{(e)} - H^{(e)}$ and $k_h^{(e)} \le T^{(e+1)}/(2H^{(e)})$. Thus, $\mathscr{F}_{h,0}^{(e)} = \mathscr{F}_{t_h^{(e)}} \supseteq \mathscr{F}_{t_2^{(e)} - H^{(e)}}$.

Lemma B.14. *$S_{h,j}^{(e)}$ is a martingale with respect to $\mathscr{F}_{h,j}^{(e)}$ for $j \ge 0$. Further, $S_{k,0}^{(e)} = 0$, $|S_{h,j+1}^{(e)} - S_{h,j}^{(e)}| \le O(mn)$.*

Proof. As proved in the literature [42,83], we have $\|x_t\|_2 \le b_x = O(\sqrt{mn})$ even when $\mathscr{E}_{\text{safe}}^{\circ}$ does not hold. Thus, we have $|q_t| \le O(mn)$ and $|S_{h,j}^{(e)}| \le O(Tmn) < +\infty$. Further, notice that, for $t \in \mathscr{T}_2^{(e)}$, $w_{t-1}, \dots, w_{t-2H^{(e)}} \in \mathscr{F}_t$. and $\mathbf{M}_*^{(e)} \in \mathscr{F}_t$, so $q_t \in \mathscr{F}_t$, so $S_{h,j}^{(e)} \in \mathscr{F}_{h,j}^{(e)}$. Finally, $\mathbb{E}[S_{h,j+1}^{(e)} \mid \mathscr{F}_{h,j}^{(e)}] = S_{h,j}^{(e)} + \mathbb{E}[q_{h,j+1}^{(e)} \mid \mathscr{F}_{h,j}^{(e)}] = S_{h,j}^{(e)}$. This has proved that $S_{h,j}^{(e)}$ is a martingale. The rest is by definition, and q_t's bound. □

Next, by Lemma B.14, we can apply Proposition B.1 with $\delta = \frac{p}{2\sum_{e=0}^{N-1} H^{(e)}}$ and obtain the bound below:

$$|S_{h,k_h^{(e)}}^{(e)}| \le \tilde{O}\left(\sqrt{k_h^{(e)}} mn\right), \quad \text{w.p. } 1 - \delta,$$

where we used $\log(2/\delta) = \tilde{O}(1)$. Then, by summing over h, we have

$$|\sum_{h=0}^{2H^{(e)}-1} S_{h,k_h^{(e)}}^{(e)}| \le \tilde{O}\left(\sqrt{T^{(e+1)}} mn\right), \quad \text{w.p. } 1 - 2H^{(e)}\delta,$$

where we used the Cauchy-Schwartz inequality. Finally, we can prove Lemma B.13 by summing over e and Lemma B.7. □

Bound Part i

Lemma B.15 (Bound on Part i)**.** *When $\mathscr{E}_{safe}$ is true, under conditions in Theorem 7.3,*

$$\sum_{e}\sum_{t\in\mathscr{T}_2^{(e)}} l(x_t,u_t)-l(\tilde{x}_t,\tilde{u}_t)\le \tilde{O}(n\sqrt{m}\sqrt{m+n}T^{2/3})$$

Proof. When $\mathscr{E}_{\text{safe}}$ is true, we have $x_t,\tilde{x}_t\in\mathbb{X}$ and $u_t,\tilde{u}_t\in\mathbb{U}$. Therefore, it suffices to bound $\|x_t-\tilde{x}_t\|_2$ and $\|u_t-\tilde{u}_t\|_2$. By Proposition 5.1 and Lemma B.5, we obtain

$$\|x_t-\tilde{x}_t\|_2\le O((1-\gamma)^{H^{(e)}}x_{\max}+\sqrt{mn}z_{\max}r^{(e+1)})=\tilde{O}(n\sqrt{m}\sqrt{m+n}(T^{(e+1)})^{-1/3})$$
$$\|u_t-\tilde{u}_t\|_2\le O(z_{\max}\sqrt{mn}r^{(e+1)})=\tilde{O}(n\sqrt{m}\sqrt{m+n}(T^{(e+1)})^{-1/3}).$$

By the quadratic form of $l(x_t,u_t)$, we have $l(x_t,u_t)-l(\tilde{x}_t,\tilde{u}_t)\le\tilde{O}(n\sqrt{m}\sqrt{m+n}(T^{(e+1)})^{-1/3})$. By summing over t and e and by Lemma B.7, we obtain the bound on Part i. □

Completing the proof. The proof is completed by summing over the bounds on Parts i-v. Notice that Lemma B.10, B.11, B.12, B.15 all condition on $\mathscr{E}_{\text{safe}}$, which holds w.p. $1-p$. But Lemmas B.13 conditions on a different event, which also holds w.p. $1-p$. Putting them together, we conclude that our regret bound holds w.p. $1-2p$.

Appendix C | Appendix to Part III

C.1 Proof of Theorem 8.1

This section provides the proof of Theorem 8.1. We introduce necessary notations, outline the main ideas of the proof, remark on the differences between our proof and the proofs in related literature [59, 192], and then provide proof details in subsections.

Notations. In the following, we introduce some useful notations for any stabilizing controller $\mathsf{K} \in \mathscr{K}_{\mathrm{st}}$. First, we let $\tilde{J}_i(\mathsf{K})$ denote agent i's estimation of the global objective $J(\mathsf{K})$ through the subroutine `GlobalCostEst`, i.e.,

$$\big(\tilde{J}_1(\mathsf{K}), \dots, \tilde{J}_N(\mathsf{K})\big) \coloneqq \texttt{GlobalCostEst}\big((K_i)_{i=1}^N, T_J\big),$$

and we let $\hat{J}_i(\mathsf{K}) \coloneqq \min\big\{\tilde{J}_i(\mathsf{K}), \bar{J}\big\}$ denote the truncation of $\tilde{J}_i(\mathsf{K})$. Notice that $\tilde{J}_i(\mathsf{K}(s))$ and $\hat{J}_i(\mathsf{K}(s))$ correspond to $\tilde{J}_i(s)$ and $\hat{J}_i(s)$ in Algorithm 8.1 respectively. Then, for any $r > 0$ and $\mathsf{D} \in \mathbb{R}^{n_K}$ such that $\mathsf{K} + r\mathsf{D} \in \mathscr{K}_{\mathrm{st}}$, we define

$$\begin{aligned}
\hat{G}_i^r(\mathsf{K}, \mathsf{D}) &\coloneqq \frac{n_K}{r} \hat{J}_i(\mathsf{K}+r\mathsf{D}) D_i, \quad \forall\, 1 \le i \le N, \\
\hat{\mathsf{G}}^r(\mathsf{K}, \mathsf{D}) &\coloneqq \mathrm{vec}\big((\hat{G}_i^r(\mathsf{K}, \mathsf{D}))_{i=1}^N\big),
\end{aligned}$$

where D_i are $m_i \times n_i$ matrices such that $\mathsf{D} = \mathrm{vec}\big((D_i)_{i=1}^N\big)$. Notice that $\hat{G}_i^r(\mathsf{K}, \mathsf{D})$ denotes agent i's estimate of the partial gradient $\frac{\partial J}{\partial K_i}(\mathsf{K})$ given the controller K and perturbation D. In particular, $\hat{G}_i^r(\mathsf{K}(s), \mathsf{D}(s))$ corresponds to $\hat{G}_i^r(s)$ in Step 3 of Algorithm 8.1. The

vector $\hat{\mathsf{G}}^r(\mathsf{K},\mathsf{D})$ that concatenates all the (vectorized) partial gradient estimates of the agents then gives an estimate of the complete gradient vector $\nabla J(\mathsf{K})$.

Proof Outline. Our proof mainly consists of six parts.

(a) Bound the sampling error in Step 1 of Algorithm 8.1.

(b) Bound the estimation error of the global objective generated by Step 2 of Algorithm 8.1.

(c) Bound the estimation error of partial gradients generated by Step 3 of Algorithm 8.1.

(d) Characterize the improvement by one-step distributed policy update in Step 4 of Algorithm 8.1.

(e) Prove Statement 1 in Theorem 8.1, i.e. all the generated controllers are stabilizing with high probability.

(f) Prove Statement 2 in Theorem 8.1, i.e. the bound (8.11).

Each part is discussed in detail in the subsequent subsections. Some proofs of the technical lemmas are deferred to the next section.

C.1.1 Bounding the sampling inaccuracy

In this part, we focus on the subroutine `SampleUSphere` and bound the deviation of it outputs from the desired distribution $\mathrm{Uni}(\mathbb{S}_{n_K})$. The proofs are deferred to the next section.

Lemma C.1. *Consider the subroutine* `SampleUSphere`*, let* $D_i^0 = \frac{V_i}{\sqrt{\sum_{i=1}^N \|V_i\|_F^2}}$ *and* $\mathsf{D}^0 = \mathrm{vec}\big((D_i^0)_{i=1}^N\big)$*. Then, we have* $\mathsf{D}^0 \sim \mathrm{Uni}(\mathbb{S}_{n_K})$*. Further, for* $T_S \geq \log 2N/(-\log \rho_W)$*,*

$$\left\|D_i - D_i^0\right\|_F \le N\rho_W^{T_S} \cdot \|D_i^0\|_F, \quad i = 1, \dots, N, \quad \sum_{i=1}^{N} \|D_i\|_F^2 \le \left(1 + N\rho_W^{T_S}\right)^2. \tag{C.1}$$

C.1.2 Bounding the global cost estimation error

In this part, we bound the difference between the global cost estimation $\tilde{J}_i(\mathsf{K})$ and the true cost $J(\mathsf{K})$ for any $\mathsf{K} \in \mathscr{Q}^1$. Besides, we bound the expected difference between $\tilde{J}_i(\mathsf{K})$ and the truncated estimation $\hat{J}_i(\mathsf{K})$. Later in Section C.1.5, we will show that the outputs generated by Algorithm 8.1 are inside $\mathscr{Q}^0 \subseteq \mathscr{Q}^1$ with high probability, thus the bounds here characterize the properties of the output controllers.

Lemma C.2 (Estimation error of `GlobalCostEst`)**.** *There exists $\beta_0 > 0$ determined by $A, B, Q_i, R_i, \mathsf{K}_0, \Sigma_w$, such that for any $\mathsf{K} \in \mathscr{Q}^1$ and any $1 \le i \le N$,*

$$\left|\mathbb{E}\left[\tilde{J}_i(\mathsf{K})\right] - J(\mathsf{K})\right| \le \frac{J(\mathsf{K})}{T_J}\left[\frac{N}{1-\rho_W} + \beta_0\right], \tag{C.2}$$

$$\mathbb{E}\left(\tilde{J}_i(\mathsf{K}) - J(\mathsf{K})\right)^2 \le \frac{6nJ(\mathsf{K})^2}{T_J}\beta_0^2 + \frac{8J(\mathsf{K})^2}{T_J^2}\left[\frac{N}{1-\rho_W}\right]^2 \tag{C.3}$$

where the expectation is taken with respect to the process noises when implementing `GlobalCostEst`.

Lemma C.3 (Effect of truncation)**.** *Given $\mathsf{K} \in \mathscr{Q}^1$, when the constant $\bar{J}$ satisfies $\bar{J} \ge J(\mathsf{K}) \max\left\{\frac{5}{2}, \frac{5N}{T_J(1-\rho_W)}\right\}$, the truncation $\hat{J}_i(\mathsf{K}) = \min(\tilde{J}_i(\mathsf{K}), \bar{J})$ will satisfy*

$$0 \le \mathbb{E}\left[\tilde{J}_i(\mathsf{K}) - \hat{J}_i(\mathsf{K})\right] \le \frac{90J(\mathsf{K})}{T_J^2}\left[n^2\beta_0^4 + \frac{N^2}{(1-\rho_W)^2}\right].$$

where β_0 is defined in Lemma C.2, and the expectation is taken with respect to the process noises when implementing `GlobalCostEst`.

C.1.3 Bounding the gradient estimation error

In this part, we bound the bias and the second moment of the gradient estimator $\hat{\mathsf{G}}^r(\mathsf{K},\mathsf{D})$ for any $\mathsf{K}\in\mathscr{Q}^0$. Our $\hat{\mathsf{G}}^r(\mathsf{K},\mathsf{D})$ is based on the zero-order gradient estimator $\mathsf{G}^r(\mathsf{K},\mathsf{D})$ defined in (8.5), whose bias can be bounded by the following lemma.

Lemma C.4 ([192, Lemma 6]). *Consider any $\mathsf{K}\in\mathscr{Q}^0$ and $\mathsf{D}\sim\mathrm{Uni}(\mathbb{S}_{n_K})$, then $\|\mathbb{E}_{\mathsf{D}}[\mathsf{G}^r(\mathsf{K},\mathsf{D})]-\nabla J(\mathsf{K})\|\le\phi_0 r$ for $r\le\xi_0$.*

Notice that our gradient estimator $\hat{\mathsf{G}}^r(\mathsf{K},\mathsf{D})$ relies on the estimated objective value $\hat{J}_i(\mathsf{K}+r\mathsf{D})$ instead of the accurate value $J(\mathsf{K}+r\mathsf{D})$ as in $\mathsf{G}^r(\mathsf{K},\mathsf{D})$; in addition, the distribution of D is only an approximation of $\mathrm{Uni}(\mathbb{S}_{n_K})$. Consequently, there will be additional error in the gradient estimation step. By leveraging Lemma C.4 and the cost estimation error bounds in Lemmas C.2 and C.3 in Section C.1.2, we obtain bounds on the bias and second moment of our gradient estimator $\hat{\mathsf{G}}^r(\mathsf{K},\mathsf{D})$.

We introduce an auxiliary quantity $\kappa_0\coloneqq\sup_{\mathsf{K}\in\mathscr{Q}^1}\|\nabla J(\mathsf{K})\|$. Lemma 8.1 guarantees that $\kappa_0<+\infty$.

Lemma C.5 (Properties of gradient estimation). *Let $\delta\in(0,1/14]$ be arbitrary. Suppose*

$$r\le\min\left\{\frac{14}{15}\xi_0,\frac{20J(\mathsf{K}_0)}{\kappa_0}\right\},\quad \bar{J}\ge 50J(\mathsf{K}_0),$$
$$T_J\ge 120\max\left\{n\beta_0^2,\frac{N}{1-\rho_W}\right\},\quad T_S\ge\frac{\log(N/\delta)}{-\log\rho_W},$$

and let D be generated by `SampleUSphere`. *Then for any $\mathsf{K}\in\mathscr{Q}^0$, we have $\mathsf{K}+r\mathsf{D}\in\mathscr{Q}^1$. Furthermore,*

$$\left\|\mathbb{E}\left[\hat{\mathsf{G}}^r(\mathsf{K},\mathsf{D})\right]-\nabla J(\mathsf{K})\right\|^2\le 5\phi_0^2r^2+2\left(\frac{50\delta n_K J(\mathsf{K}_0)}{r}\right)^2$$

$$+ 5\left(\frac{50 n_K J(\mathsf{K}_0)}{r T_J} \max\left\{n\beta_0^2, \frac{N}{1-\rho_W}\right\}\right)^2 \tag{C.4}$$

$$\mathbb{E}\Big[\big\|\hat{\mathsf{G}}^r(\mathsf{K},\mathsf{D})\big\|^2\Big] \le \Big(30 J(\mathsf{K}_0)\frac{n_K}{r}\Big)^2, \tag{C.5}$$

where the expectation $\mathbb{E}$ *is with respect to* D *and the system process noises in the subroutine* `GlobalCostEst`*.*

Proof. Firstly, the condition on T_S implies $N\rho_W^{T_S} \le \delta$ since $\rho_W < 1$, and by Lemma C.1, we have

$$\|\mathsf{D}\|^2 = \sum_{i=1}^N \|D_i\|_F^2 \le (1+\delta)^2 \le \left(\frac{15}{14}\right)^2, \tag{C.6}$$

and consequently $\|r\mathsf{D}\| \le \xi_0$. By the definition of ξ_0 below Lemma 8.1, we have that $\mathsf{K} + r\mathsf{D} \in \mathscr{Q}^1$ for any $\mathsf{K} \in \mathscr{Q}^0$.

We then proceed to prove the two inequalities. We let D_i^0, $i = 1, \dots, N$ and D^0 denote the random matrices and the random vector as defined in Lemma C.1, so that $\mathsf{D}^0 \sim \mathrm{Uni}(\mathbb{S}_{n_K})$ and the bounds (C.1) hold.

- *Proof of* (C.4)*:* Let $\mathsf{K} \in \mathscr{Q}^0$ be arbitrary. Notice that

$$\begin{aligned}
&\Big\|\mathbb{E}\Big[\hat{\mathsf{G}}^r(\mathsf{K},\mathsf{D})\Big] - \nabla J(\mathsf{K})\Big\|^2 \\
\le\ & \frac{5}{4}\Big\|\mathbb{E}\Big[\hat{\mathsf{G}}^r(\mathsf{K},\mathsf{D}) - \mathsf{G}^r(\mathsf{K},\mathsf{D}^0)\Big]\Big\|^2 + 5\Big\|\mathbb{E}\big[\mathsf{G}^r(\mathsf{K},\mathsf{D}^0)\big] - \nabla J(\mathsf{K})\Big\|^2 \\
\le\ & \frac{5}{4}\frac{n_K^2}{r^2}\sum_{i=1}^N \Big\|\mathbb{E}\Big[\hat{J}_i(\mathsf{K}+r\mathsf{D})D_i - J(\mathsf{K}+r\mathsf{D}^0)D_i^0\Big]\Big\|_F^2 + 5\phi_0^2 r^2,
\end{aligned}$$

where we use Lemma C.4 and $2xy \le (x/r)^2 + (ry)^2$ for any x, y, r. By leveraging the same trick, we bound the first term:

$$\sum_{i=1}^N \Big\|\mathbb{E}\Big[\hat{J}_i(\mathsf{K}+r\mathsf{D})D_i - J(\mathsf{K}+r\mathsf{D}^0)D_i^0\Big]\Big\|_F^2$$

$$\leq \frac{5}{3}\sum_{i=1}^{N}\left\|\mathbb{E}\Big[\Big(\hat{J}_i(\mathsf{K}+r\mathsf{D})D_i - J(\mathsf{K}+r\mathsf{D})\Big)\,D_i\Big]\right\|_F^2 \tag{P1}$$

$$+5\sum_{i=1}^{N}\left\|\mathbb{E}\big[J(\mathsf{K}+r\mathsf{D})\left(D_i - D_i^0\right)\big]\right\|_F^2 \tag{P2}$$

$$+5\sum_{i=1}^{N}\left\|\mathbb{E}\big[\big(J(\mathsf{K}+r\mathsf{D}) - J(\mathsf{K}+r\mathsf{D}^0)\big)\,D_i^0\big]\right\|_F^2. \tag{P3}$$

Next, we bounds (P1), (P2), (P3). Remember that $K + rD \in \mathscr{Q}^1$. For (P1), by (C.6), we have

$$\begin{aligned}
&\sum_{i=1}^{N}\left\|\mathbb{E}\Big[\Big(\hat{J}_i(\mathsf{K}+r\mathsf{D})D_i - J(\mathsf{K}+r\mathsf{D})\Big)\,D_i\Big]\right\|_F^2\\
&=\sum_{i=1}^{N}\left\|\mathbb{E}_{\mathsf{D}}\Big[\mathbb{E}\Big[\hat{J}_i(\mathsf{K}+r\mathsf{D}) - J(\mathsf{K}+r\mathsf{D})\Big|\,\mathsf{D}\Big]\,D_i\Big]\right\|_F^2\\
&\leq\sum_{i=1}^{N}\sup_{\mathsf{K}'\in\mathscr{Q}^1}\left\{\left|\mathbb{E}_w\Big[\hat{J}_i(\mathsf{K}')\Big] - J(\mathsf{K}')\right|^2\right\}\cdot\mathbb{E}\big[\|D_i\|_F^2\big]\\
&\leq(1+\delta)^2\max_{1\leq i\leq N}\sup_{\mathsf{K}'\in\mathscr{Q}^1}\left\{\left|\mathbb{E}_w\Big[\hat{J}_i(\mathsf{K}')\Big] - J(\mathsf{K}')\right|^2\right\},
\end{aligned}$$

where $\mathbb{E}_w$ denotes expectation with respect to the noise process of the dynamical system in the subroutine `GlobalCostEst`. For any i and $\mathsf{K}' \in \mathscr{Q}^1$, we have $\left|\mathbb{E}_w\Big[\hat{J}_i(\mathsf{K}')\Big] - J(\mathsf{K}')\right| \leq \left|\mathbb{E}_w\Big[\hat{J}_i(\mathsf{K}') - \mathbb{E}\,\tilde{J}_i(\mathsf{K}')\Big]\right| + \left|\mathbb{E}_w\Big[\tilde{J}_i(\mathsf{K}')\Big] - J(\mathsf{K}')\right| \leq \frac{90J(\mathsf{K}')}{T_J^2}\left(n^2\beta_0^4 + \frac{N^2}{(1-\rho_W)^2}\right) + \frac{J(\mathsf{K}')}{T_J}\left(\beta_0 + \frac{N}{1-\rho_W}\right)$, where we used Lemma C.3 and the bound (C.2) in Lemma C.2 in the second inequality. Notice that the condition on T_J implies

$$\frac{1}{T_J^2}\left(n^2\beta_0^4 + \frac{N^2}{(1-\rho_W)^2}\right) \leq \frac{1}{120T_J}\left(n\beta_0^2 + \frac{N}{1-\rho_W}\right),$$

and since $J(\mathsf{K}') \le 20J(\mathsf{K}_0)$ for $\mathsf{K}' \in \mathscr{Q}^1$, we obtain

$$\begin{aligned}
&\left|\mathop{\mathbb{E}}_{w}\left[\hat{J}_i(\mathsf{K}')\right] - J(\mathsf{K}')\right| \\
&\le \frac{3J(\mathsf{K}')}{4T_J}\left(n\beta_0^2 + \frac{N}{1-\rho_W}\right) + \frac{J(\mathsf{K}')}{T_J}\left(\beta_0 + \frac{N}{1-\rho_W}\right) \\
&\le \frac{20J(\mathsf{K}_0)}{T_J}\left(\left(\frac{3}{4}n+1\right)\beta_0^2 + \left(\frac{3}{4}+1\right)\frac{N}{1-\rho_W}\right) \\
&\le \frac{35J(\mathsf{K}_0)}{T_J}\left(n\beta_0^2 + \frac{N}{1-\rho_W}\right) \le \frac{70J(\mathsf{K}_0)}{T_J}\max\left\{n\beta_0^2, \frac{N}{1-\rho_W}\right\}.
\end{aligned}$$

Therefore, by $\delta \le 1/14$, we have

$$\begin{aligned}
&\sum_{i=1}^N \left\|\mathbb{E}\left[\left(\hat{J}_i(\mathsf{K}+r\mathsf{D})D_i - J(\mathsf{K}+r\mathsf{D})\right)D_i\right]\right\|^2 \\
&\le (1+\delta)^2\left(\frac{70J(\mathsf{K}_0)}{T_J}\max\left\{n\beta_0^2, \frac{N}{1-\rho_W}\right\}\right)^2 \\
&\le \left(\frac{75J(\mathsf{K}_0)}{T_J}\max\left\{n\beta_0^2, \frac{N}{1-\rho_W}\right\}\right)^2.
\end{aligned}$$

Next, by Lemma C.1 and $\mathsf{D}^0 \sim \mathrm{Uni}(\mathbb{S}_{n_K})$, we bound (P2) by

$$\begin{aligned}
&\sum_{i=1}^N \left\|\mathbb{E}\left[J(\mathsf{K}+r\mathsf{D})\left(D_i - D_i^0\right)\right]\right\|^2 \\
&\le \sup_{\mathsf{K}\in\mathscr{Q}^1} J(\mathsf{K}')^2 \cdot \sum_{i=1}^N \mathbb{E}\left[\|D_i - D_i^0\|_F^2\right] \\
&\le (20J(\mathsf{K}_0))^2 \cdot \delta^2 \mathbb{E}\left[\|D_i^0\|_F^2\right] = \delta^2(20J(\mathsf{K}_0))^2.
\end{aligned}$$

Further, by $\kappa_0 = \sup_{\mathsf{K}\in\mathscr{Q}^1}\|\nabla J(\mathsf{K})\|$, we can bound (P3) by

$$\begin{aligned}
&\sum_{i=1}^N \left\|\mathbb{E}\left[\left(J(\mathsf{K}+r\mathsf{D}) - J(\mathsf{K}+r\mathsf{D}^0)\right)D_i^0\right]\right\|_F^2 \\
&\le \sum_{i=1}^N \mathbb{E}\left[r^2\kappa_0^2\|\mathsf{D}-\mathsf{D}^0\|^2\|D_i^0\|_F^2\right] = r^2\kappa_0^2\,\mathbb{E}\left[\|\mathsf{D}-\mathsf{D}^0\|^2\right] \\
&\le \delta^2 r^2\kappa_0^2 \le \delta^2(20J(\mathsf{K}_0))^2,
\end{aligned}$$

where we used the assumption that $r \le 20J(\mathsf{K}_0)/\kappa_0$.

Now we summarize all the previous results and obtain

$$\begin{aligned}
&\left\|\mathbb{E}\Big[\hat{\mathsf{G}}^r(\mathsf{K},\mathsf{D})\Big]-\nabla J(\mathsf{K})\right\|^2\\
&\le 5\phi_0^2 r^2+\frac{5}{4}\frac{n_K^2}{r^2}\left[5\delta^2(20J(\mathsf{K}_0))^2+5\delta^2(20J(\mathsf{K}_0))^2\right.\\
&\quad\left.+\frac{5}{3}\left(\frac{75J(\mathsf{K}_0)}{T_J}\max\left\{n\beta_0^2,\frac{N}{1-\rho_W}\right\}\right)^2\right]\\
&\le 5\phi_0^2 r^2+2\left(\frac{50\delta n_K J(\mathsf{K}_0)}{r}\right)^2\\
&\quad+5\left(\frac{50 n_K J(\mathsf{K}_0)}{rT_J}\max\left\{n\beta_0^2,\frac{N}{1-\rho_W}\right\}\right)^2.
\end{aligned}$$

- *Proof of* (C.5)*:* Let $\mathsf{K}\in\mathscr{Q}^0$ be arbitrary. It can be seen that

$$\begin{aligned}
&\mathbb{E}\Big[\big\|\hat{\mathsf{G}}^r(\mathsf{K},\mathsf{D})\big\|^2\Big]=\sum_{i=1}^N\mathbb{E}\Big[\big\|\hat{G}_i^r(\mathsf{K},\mathsf{D})\big\|_F^2\Big]\\
&=\sum_{i=1}^N\mathbb{E}\left[\frac{n_K^2}{r^2}\,\mathbb{E}\Big[\hat{J}_i(\mathsf{K}+r\mathsf{D})^2\Big|\,\mathsf{D}\Big]\cdot\|D_i\|_F^2\right]\\
&\le\frac{n_K^2}{r^2}\max_{1\le i\le N}\sup_{\mathsf{K}'\in\mathscr{Q}^1}\Big\{\mathbb{E}_w\big[\hat{J}_i(\mathsf{K}')^2\big]\Big\}\cdot\mathbb{E}\left[\sum_{i=1}^N\|D_i\|_F^2\right]\\
&\le\frac{n_K^2}{r^2}(1+\delta)^2\max_{1\le i\le N}\sup_{\mathsf{K}'\in\mathscr{Q}^1}\Big\{\mathbb{E}_w\big[\tilde{J}_i(\mathsf{K}')^2\big]\Big\},
\end{aligned}$$

where the last inequality follows from $0\le\hat{J}_i(\mathsf{K}')\le\tilde{J}_i(\mathsf{K}')$. On the other hand, for any $\mathsf{K}'\in\mathscr{Q}^1$, we have

$$\begin{aligned}
&\mathbb{E}_w\big[\tilde{J}_i(\mathsf{K}')^2\big]\le 4\,\mathbb{E}_w\Big[\big(\tilde{J}_i(\mathsf{K}')-J(\mathsf{K}')\big)^2\Big]+\frac{4}{3}J(\mathsf{K}')^2\\
&\le 4\left[\frac{6nJ(\mathsf{K}')^2}{T_J}\beta_0^2+\frac{8J(\mathsf{K}')^2}{T_J^2}\left(\frac{N}{1-\rho_W}\right)^2\right]+\frac{4}{3}J(\mathsf{K}')^2\\
&\le 4(20J(\mathsf{K}_0))^2\left[\frac{6n\beta_0^2}{T_J}+\frac{8}{T_J^2}\left(\frac{N}{1-\rho_W}\right)^2\right]+\frac{4}{3}(20J(\mathsf{K}_0))^2\\
&\le(20J(\mathsf{K}_0))^2\left[4\left(\frac{6}{120}+\frac{8}{120^2}\right)+\frac{4}{3}\right]<(28J(\mathsf{K}_0))^2,
\end{aligned}$$

where the second inequality uses (C.3) in Lemma C.2, the third inequality follows from $J(\mathsf{K}')\le 20J(\mathsf{K}_0)$ for $\mathsf{K}'\in\mathscr{Q}^1$, and the last two inequalities follow from the condition on

T_J. By combining this bound with previous results and noting that $1+\delta \le 15/14$, we obtain the bound on $\mathbb{E}\left[\left\|\hat{\mathsf{G}}^r(\mathsf{K},\mathsf{D})\right\|^2\right]$.

□

C.1.4 Analysis of one-step stochastic gradient update

Step 4 of Algorithm 8.1 can be viewed as a stochastic gradient descent update with biased gradient estimation. In this part, we characterize the change in the objective value of this step.

We shall use $\mathscr{F}_s$ to denote the filtration $\sigma(K_i(s') : s' \le s)$ for each $s = 1, \ldots, T_G$.

Lemma C.6. *Suppose* $\bar{J} \ge 50J(\mathsf{K}_0)$ *and*

$$r \le \min\left\{\frac{14}{15}\xi_0, \frac{20J(\mathsf{K}_0)}{\kappa_0}\right\},\ \eta \le \min\left\{\frac{14\xi_0 r}{15n_K\bar{J}}, \frac{1}{25\phi_0}\right\},$$
$$T_J \ge 120\max\left\{n\beta_0^2, \frac{N}{1-\rho_W}\right\},\quad T_S \ge \frac{\log(8N^2/(\phi_0\eta))}{-2\log\rho_W}.$$

Then, as long as $\mathsf{K}(s) \in \mathscr{Q}^0$*, we will have* $\mathsf{K}(s+1) \in \mathscr{Q}^1$ *and*

$$\mathbb{E}[J(\mathsf{K}(s+1)) \mid \mathscr{F}_s] \le J(\mathsf{K}(s)) - \frac{\eta}{2}\|\nabla J(\mathsf{K}(s))\|^2 + \frac{\eta}{2}Z \tag{C.7}$$

where

$$Z \coloneqq 5\left[\phi_0^2 r^2 + \left(\frac{50J(\mathsf{K}_0)n_K}{rT_J}\max\left\{n\beta_0^2, \frac{N}{1-\rho_W}\right\}\right)^2\right]$$
$$+ \phi_0\eta\left(40J(\mathsf{K}_0)\frac{n_K}{r}\right)^2. \tag{C.8}$$

Proof. By denoting $\delta = \sqrt{\phi_0\eta/8}$, we see that $N\rho_W^{T_S} \le \delta$, and by the upper bound on η we

have $\delta \le 1/14$, so δ satisfies the condition in Lemma C.5. Therefore, by (C.6), we have

$$\begin{aligned}\|\mathsf{K}(s+1)-\mathsf{K}(s)\|^2 &= \eta^2 \sum_{i=1}^N \|\hat{G}_i^r(s)\|_F^2 \\ \le \eta^2 \sum_{i=1}^N \frac{n_K^2 \bar{J}^2}{r^2}\|D_i(s)\|_F^2 &\le \left(\frac{(1+\delta)\eta\, n_K \bar{J}}{r}\right)^2 \le \xi_0^2,\end{aligned}$$

which implies $\mathsf{K}(s+1) \in \mathscr{Q}^1$ as long as $\mathsf{K}(s) \in \mathscr{Q}^0$. Secondly, since $\mathsf{K}(s) \in \mathscr{Q}^0$, by Lemma 8.1, we have

$$J(\mathsf{K}(s+1)) \le J(\mathsf{K}(s)) - \eta\langle \nabla J(\mathsf{K}(s)), \hat{\mathsf{G}}^r(s)\rangle + \frac{\phi_0\eta^2}{2}\left\|\hat{\mathsf{G}}^r(s)\right\|^2.$$

Taking expectation conditioned on the filtration $\mathscr{F}_s$ yields

$$\begin{aligned}&\mathbb{E}[J(\mathsf{K}(s+1)) \,|\, \mathscr{F}_s] \\ \le\ & J(\mathsf{K}(s)) - \eta\left\langle \nabla J(\mathsf{K}(s)), \mathbb{E}\left[\hat{\mathsf{G}}^r(s)\middle|\mathscr{F}_s\right]\right\rangle \\ &+ \frac{\phi_0}{2}\eta^2\,\mathbb{E}\left[\left\|\hat{\mathsf{G}}^r(s)\right\|^2\middle|\mathscr{F}_s\right] \\ =\ & J(\mathsf{K}(s)) - \eta\|\nabla J(\mathsf{K}(s))\|^2 + \frac{\phi_0}{2}\eta^2\,\mathbb{E}\left[\left\|\hat{\mathsf{G}}^r(s)\right\|^2\middle|\mathscr{F}_s\right] \\ &+ \eta\left\langle \nabla J(\mathsf{K}(s)), \nabla J(\mathsf{K}(s)) - \mathbb{E}\left[\hat{\mathsf{G}}^r(s)\middle|\mathscr{F}_s\right]\right\rangle \\ \le\ & J(\mathsf{K}(s)) - \frac{\eta}{2}\|\nabla J(\mathsf{K}(s))\|^2 + \frac{\phi_0}{2}\eta^2\,\mathbb{E}\left[\left\|\hat{\mathsf{G}}^r(s)\right\|^2\middle|\mathscr{F}_s\right] \\ &+ \frac{\eta}{2}\left\|\nabla J(\mathsf{K}(s)) - \mathbb{E}\left[\hat{\mathsf{G}}^r(s)\middle|\mathscr{F}_s\right]\right\|^2,\end{aligned}$$

where we used Cauchy's inequality in the last step.

By applying the results of Lemma C.5 to above, we obtain

$$\begin{aligned}
&\mathbb{E}[J(\mathsf{K}(s+1)) \,|\, \mathscr{F}_s] \\
&\leq J(\mathsf{K}(s)) - \frac{\eta}{2}\|\nabla J(\mathsf{K}(s))\|^2 \\
&\quad + \frac{\phi_0 \eta^2}{2}\left(30 J(\mathsf{K}_0)\frac{n_K}{r}\right)^2 + \frac{\eta}{2}\cdot 2\left(\frac{50\delta n_K J(\mathsf{K}_0)}{r}\right)^2 \\
&\quad + \frac{\eta}{2}\cdot 5\left[\phi_0^2 r^2 + \left(\frac{50 n_K J(\mathsf{K}_0)}{r T_J}\max\left\{n\beta_0^2, \frac{N}{1-\rho_W}\right\}\right)^2\right] \\
&\leq J(\mathsf{K}(s)) - \frac{\eta}{2}\|\nabla J(\mathsf{K}(s))\|^2 + \frac{\phi_0 \eta^2}{2}\left(40 J(\mathsf{K}_0)\frac{n_K}{r}\right)^2 \\
&\quad + \frac{\eta}{2}\cdot 5\left[\phi_0^2 r^2 + \left(\frac{50 n_K J(\mathsf{K}_0)}{r T_J}\max\left\{n\beta_0^2, \frac{N}{1-\rho_W}\right\}\right)^2\right],
\end{aligned}$$

which concludes the proof. □

C.1.5 Proving stability of the output controllers

Next, we show that all the output controllers $\{\mathsf{K}(s)\}_{s=1}^{T_G}$ are in $\mathscr{Q}^0$ with high probability, which then implies that all the output controllers are stabilizing with high probability.

We assume that the algorithmic parameters satisfy the conditions in Theorem 8.1. It's not hard to see that for sufficiently small $\epsilon > 0$, the conditions of Lemma C.6 are satisfied.

We define a stopping time τ to be the first time step when $K(s)$ escapes $\mathscr{Q}^0$:

$$\tau \coloneqq \min\left\{s \in \{1, \ldots, T_G{+}1\} : J(\mathsf{K}(s)) > 10 J(\mathsf{K}_0)\right\}. \tag{C.9}$$

Our goal is then to bound the probability $\mathbb{P}(\tau \leq T_G)$. We first note that, under the conditions of Theorem 8.1,

$$Z \leq 5\left[\phi_0^2 r^2 + \left(\frac{J(\mathsf{K}_0) n_K}{r}\frac{r\sqrt{\epsilon}}{20 J(\mathsf{K}_0) n_K}\right)^2\right]$$

$$+\phi_0\Big(40J(\mathsf{K}_0)\frac{n_K}{r}\Big)^2\frac{3\epsilon r^2}{320\phi_0(40J(\mathsf{K}_0))^2n_K^2}$$
$$\leq 5\left(\frac{\epsilon}{1600}+\frac{\epsilon}{400}\right)+\frac{3\epsilon}{320}=\frac{\epsilon}{40}. \tag{C.10}$$

Now, we define a nonnegative supermartingale $Y(s)$ by

$$Y(s) \coloneqq J(\mathsf{K}(\min\{s,\tau\}))+(T_G-s)\cdot\frac{\eta}{2}Z,\quad 1\leq s\leq T_G.$$

It is straightforward that $Y(s)\geq 0$ for $1\leq s\leq T_G$. To verify that it is a supermartingale, we notice that when $\tau > s$,

$$\begin{aligned}\mathbb{E}[Y(s+1)|\mathscr{F}_s] =\ & \mathbb{E}[J(\mathsf{K}(s+1))|\mathscr{F}_s]+(T_G-s-1)\cdot\frac{\eta}{2}Z\\ \leq\ & J(\mathsf{K}(s))-\frac{\eta}{2}\|\nabla J(\mathsf{K}(s))\|^2+\frac{\eta}{2}Z\\ &+(T_G-1-s)\cdot\frac{\eta}{2}Z\leq Y(s),\end{aligned}$$

and when $\tau\leq s$, $\mathbb{E}[Y(s+1)|\mathscr{F}_s]=J(\mathsf{K}(\tau)+(T_G-1-s)\frac{\eta}{2}Z\leq Y(s)$. Now, by the monotonicity and Doob's maximal inequality for supermartingales, we obtain the following bound:

$$\begin{aligned}\mathbb{P}(\tau\leq T_G)&\leq\mathbb{P}\left(\max_{s=1,\ldots,T_G}Y(s)>10J(\mathsf{K}_0)\right)\leq\frac{\mathbb{E}[Y(1)]}{10J(\mathsf{K}_0)}\\ &=\frac{J(\mathsf{K}_0)+(T_G-1)\eta Z/2}{10J(\mathsf{K}_0)}\leq\frac{1}{10}+\frac{c}{20},\end{aligned} \tag{C.11}$$

where the last inequality used $T_G=c\cdot 40J(\mathsf{K}_0)/(\eta\epsilon)$ and $Z\leq\epsilon/40$. This implies that all the output controllers are stabilizing with probability at least $1-(1/10+c/20)=9/10-c/20$.

C.1.6 Proving the performance bound in Theorem 8.1.

To prove the performance bound (8.11), we first extend the results in Lemma C.6 and show that

$$\begin{aligned}&\mathbb{E}\big[J(\mathsf{K}(s+1))\mathbf{1}_{\{\tau>s+1\}}|\mathscr{F}_s\big]\\ &\leq J(\mathsf{K}(s))\mathbf{1}_{\{\tau>s\}}-\frac{\eta}{2}\|\nabla J(\mathsf{K}(s))\|^2\mathbf{1}_{\{\tau>s\}}+Z,\end{aligned}\tag{C.12}$$

If $\{\tau > s\}$ occurs, then $\mathsf{K}(s) \in Q^0$, and it can be verified that the other conditions of Lemma C.6 also hold. Then, we have

$$\begin{aligned}&\mathbb{E}[J(\mathsf{K}(s+1))\mathbf{1}_{\{\tau>s+1\}}|\mathscr{F}_s]\leq\mathbb{E}[J(\mathsf{K}(s+1))|\mathscr{F}_s]\\ &\leq J(\mathsf{K}(s))-\frac{\eta}{2}\|\nabla J(\mathsf{K}(s))\|^2+\frac{\eta}{2}Z\\ &=J(\mathsf{K}(s))\mathbf{1}_{\{\tau>s\}}-\frac{\eta}{2}\|\nabla J(\mathsf{K}(s))\|^2\mathbf{1}_{\{\tau>s\}}+\frac{\eta}{2}Z.\end{aligned}$$

Otherwise, if $\{\tau \leq s\}$ occurs, by $Z \geq 0$, we trivially have $\mathbb{E}[J(\mathsf{K}(s+1))\mathbf{1}_{\{\tau>s+1\}}|\mathscr{F}_s] = 0 \leq J(\mathsf{K}(s))\mathbf{1}_{\{\tau>s\}} - \frac{\eta}{2}\|\nabla J(\mathsf{K}(s))\|^2\mathbf{1}_{\{\tau>s\}} + \frac{\eta}{2}Z$. Summarizing the two cases, we have proved (C.12).

We then establish the following bound:

$$\mathbb{E}\left[\left(\frac{1}{T_G}\sum_{s=1}^{T_G}\|\nabla J(\mathsf{K}(s))\|^2\right)\mathbf{1}_{\{\tau>T_G\}}\right]\leq\frac{\epsilon}{40}+\frac{\epsilon}{20c}.\tag{C.13}$$

Proof of (C.13). By taking the total expectation of (C.12), we have

$$\begin{aligned}&\mathbb{E}\big[J(\mathsf{K}(s+1))\mathbf{1}_{\{\tau>s+1\}}\big]\\ &\leq\mathbb{E}\big[J(\mathsf{K}(s))\mathbf{1}_{\{\tau>s\}}\big]-\frac{\eta}{2}\mathbb{E}\big[\|\nabla J(\mathsf{K}(s))\|^2\mathbf{1}_{\{\tau>s\}}\big]+\frac{\eta}{2}Z.\end{aligned}$$

By reorganizing terms and taking the telescoping sum, we have

$$\frac{1}{T_G}\sum_{s=1}^{T_G}\mathbb{E}\big[\|\nabla J(\mathsf{K}(s))\|^2\mathbf{1}_{\{\tau>T_G\}}\big]$$

$$
\begin{aligned}
&\leq \frac{1}{T_G}\sum_{s=1}^{T_G}\mathbb{E}\big[\|\nabla J(\mathsf{K}(s))\|^2 1_{\{\tau>s\}}\big] \\
&\leq \frac{2}{\eta T_G}\mathbb{E}\big[J(\mathsf{K}(1)) - J(\mathsf{K}(T_G+1))\mathbf{1}_{(\tau>T_G+1)}\big] + Z \\
&\leq \frac{2}{\eta}\cdot\frac{\eta\epsilon}{40cJ(\mathsf{K}_0)}J(\mathsf{K}_0) + \frac{\epsilon}{40} \leq \frac{\epsilon}{20c} + \frac{\epsilon}{40},
\end{aligned}
$$

where we used the fact that $J(\mathsf{K}) \geq 0$ over $\mathsf{K} \in \mathscr{K}_{\rm st}$ and (C.10). □

By (C.13) above and the bound (C.11), the performance bound (8.11) of Theorem 8.1 can now be proved as follows:

$$
\begin{aligned}
&\mathbb{P}\left(\frac{1}{T_G}\sum_{s=1}^{T_G}\|\nabla J(\mathsf{K}(s))\|^2 \geq \epsilon\right) \\
=&\,\mathbb{P}\left(\frac{1}{T_G}\sum_{s=1}^{T_G}\|\nabla J(\mathsf{K}(s))\|^2 \geq \epsilon, \tau > T_G\right) \\
&+\mathbb{P}\left(\frac{1}{T_G}\sum_{s=1}^{T_G}\|\nabla J(\mathsf{K}(s))\|^2 \geq \epsilon, \tau \leq T_G\right) \\
\leq&\,\mathbb{P}\left(\frac{1}{T_G}\sum_{s=1}^{T_G}\|\nabla J(\mathsf{K}(s))\|^2 1_{\{\tau>T_G\}} \geq \epsilon\right) + \mathbb{P}\left(\tau \leq T_G\right) \\
\leq&\,\frac{1}{\epsilon}\,\mathbb{E}\left[\frac{1}{T_G}\sum_{s=1}^{T_G}\|\nabla J(\mathsf{K}(s))\|^2 1_{\{\tau>T_G\}}\right] + \mathbb{P}\left(\tau \leq T_G\right) \\
\leq&\,\frac{1}{8} + \frac{1}{20c} + \frac{c}{20},
\end{aligned}
$$

where we used Markov's inequality. We also have

$$
\begin{aligned}
&\mathbb{P}\left(\|\nabla J(\hat{\mathsf{K}})\|^2 \geq \epsilon\right) \\
\leq&\,\mathbb{P}\left(\|\nabla J(\hat{\mathsf{K}})\|^2 1_{\{\tau>T_G\}} \geq \epsilon\right) + \mathbb{P}\left(\tau \leq T_G\right) \\
\leq&\,\frac{1}{\epsilon}\,\mathbb{E}\Big[\|\nabla J(\hat{\mathsf{K}})\|^2 1_{\{\tau>T_G\}}\Big] + \mathbb{P}\left(\tau \leq T_G\right) \\
=&\,\frac{1}{\epsilon}\,\mathbb{E}\left[\mathbb{E}\Big[\|\nabla J(\hat{\mathsf{K}})\|^2 1_{\{\tau>T_G\}} \mid \{\mathsf{K}(s)\}_{s=1}^{T_G}\Big]\right] + \mathbb{P}\left(\tau \leq T_G\right) \\
=&\,\frac{1}{\epsilon}\,\mathbb{E}\left[\frac{1}{T_G}\sum_{s=1}^{T_G}\|\nabla J(\mathsf{K}(s))\|^2 1_{\{\tau>T_G\}}\right] + \mathbb{P}\left(\tau \leq T_G\right)
\end{aligned}
$$

$$\leq \frac{1}{8} + \frac{1}{20c} + \frac{c}{20}.$$

where the second equality follows by noticing that, conditioning on $\{\mathsf{K}(s)\}_{s=1}^{T_G}$, $1_{\{\tau > T_G\}}$ is a constant and $\hat{\mathsf{K}}$ is uniformly randomly selected from $\{\mathsf{K}(s)\}_{s=1}^{T_G}$.

C.2 Proofs to technical lemmas in Appendix C.1

C.2.1 Additional notations and auxiliary results

Notations: Recall that $\mathscr{M}(\mathsf{K}) \in \mathbb{R}^{n\times m}$ denotes the global control gain given $\mathsf{K} \in \mathscr{K}_{\mathrm{st}}$, and notice that $\mathscr{M}$ is an injective linear map from $\mathbb{R}^{n_K}$ to $\mathbb{R}^{n\times m}$. For simplicity, we denote

$$A_{\mathsf{K}} \coloneqq A + B\mathscr{M}(\mathsf{K}), \quad Q_{i,\mathsf{K}} \coloneqq Q_i + \mathscr{M}(\mathsf{K})^\top R_i \mathscr{M}(\mathsf{K}),$$
$$Q \coloneqq \frac{1}{N}\sum_{i=1}^N Q_i, \quad R \coloneqq \frac{1}{N}\sum_{i=1}^N R_i, \quad Q_{\mathsf{K}} \coloneqq \frac{1}{N}\sum_{i=1}^N Q_{i,\mathsf{K}}.$$

Besides, we define

$$\Sigma_{\mathsf{K},t} = \mathbb{E}\big[x(t)x(t)^\top\big], \quad \Sigma_{\mathsf{K},\infty} = \lim_{t\to\infty} \mathbb{E}\big[x(t)x(t)^\top\big], \tag{C.14}$$

where $x(t)$ is the state generated by controller K. Notice that

$$\Sigma_{\mathsf{K},t} = \sum_{\tau=0}^{t-1} A_{\mathsf{K}}^\tau \Sigma_w \big(A_{\mathsf{K}}^\top\big)^\tau \preceq \Sigma_{\mathsf{K},\infty} = \sum_{\tau=0}^{\infty} A_{\mathsf{K}}^\tau \Sigma_w \big(A_{\mathsf{K}}^\top\big)^\tau. \tag{C.15}$$

The objective function $J(\mathsf{K})$ can be represented as

$$J(\mathsf{K}) = \lim_{t\to\infty} \mathbb{E}\big[x(t)^\top Q_{\mathsf{K}} x(t)\big] = \mathrm{tr}(Q_{\mathsf{K}}\Sigma_{\mathsf{K},\infty}), \tag{C.16}$$

Auxiliary results: We provide an auxiliary lemma showing that $\|\big(\Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}} A_{\mathsf{K}} \Sigma_{\mathsf{K},\infty}^{\frac{1}{2}}\big)^t\|$ decays exponentially as t increases.

Lemma C.7. *There exists a continuous function $\varphi : \mathscr{K}_{\mathrm{st}} \to [1, +\infty)$ such that*

$$\left\| \left(\Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}} A_{\mathsf{K}} \Sigma_{\mathsf{K},\infty}^{\frac{1}{2}} \right)^t \right\| \leq \varphi(\mathsf{K}) \left(\frac{1 + \rho(A_{\mathsf{K}})}{2} \right)^t$$

for any $t \in \mathbb{N}$ and any $\mathsf{K} \in \mathscr{K}_{\mathrm{st}}$.

Proof. Denote $\tilde{\rho}(A_{\mathsf{K}}) \coloneqq (1 + \rho(A_{\mathsf{K}}))/2$. Since $\rho\left(\Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}} A_{\mathsf{K}} \Sigma_{\mathsf{K},\infty}^{\frac{1}{2}} \right) = \rho(A_{\mathsf{K}}) < \tilde{\rho}(A_{\mathsf{K}})$, the matrix series $\tilde{P}_{\mathsf{K}} = \sum_{t=0}^{\infty} \tilde{\rho}(A_{\mathsf{K}})^{-2t} \left(\Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}} A_{\mathsf{K}} \Sigma_{\mathsf{K},\infty}^{\frac{1}{2}} \right)^t \left(\Sigma_{\mathsf{K},\infty}^{\frac{1}{2}} A_{\mathsf{K}}^\top \Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}} \right)^t$ converges, and satisfies the Lyapunov equation

$$\tilde{\rho}(A_{\mathsf{K}})^{-2} \left(\Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}} A_{\mathsf{K}} \Sigma_{\mathsf{K},\infty}^{\frac{1}{2}} \right) \tilde{P}_{\mathsf{K}} \left(\Sigma_{\mathsf{K},\infty}^{\frac{1}{2}} A_{\mathsf{K}}^\top \Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}} \right) + I_n = \tilde{P}_{\mathsf{K}}.$$

By denoting $\tilde{A}_{\mathsf{K}} = \tilde{P}_{\mathsf{K}}^{-\frac{1}{2}} \Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}} A_{\mathsf{K}} \Sigma_{\mathsf{K},\infty}^{\frac{1}{2}} \tilde{P}_{\mathsf{K}}^{\frac{1}{2}}$, we obtain $\tilde{A}_{\mathsf{K}} \tilde{A}_{\mathsf{K}}^\top + \tilde{\rho}(A_{\mathsf{K}})^2 \tilde{P}_{\mathsf{K}}^{-1} = \tilde{\rho}(A_{\mathsf{K}})^2 I_n$, and thus $\|\tilde{A}_{\mathsf{K}}\| \leq \tilde{\rho}(A_{\mathsf{K}})$. Consequently, $\left\| \left(\Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}} A_{\mathsf{K}} \Sigma_{\mathsf{K},\infty}^{\frac{1}{2}} \right)^t \right\| = \left\| \left(\tilde{P}_{\mathsf{K}}^{\frac{1}{2}} \tilde{A}_K \tilde{P}_{\mathsf{K}}^{-\frac{1}{2}} \right)^t \right\| \leq \left\| \tilde{P}_{\mathsf{K}}^{-\frac{1}{2}} \right\| \left\| \tilde{P}_{\mathsf{K}}^{\frac{1}{2}} \right\| \left\| \tilde{A}_{\mathsf{K}} \right\|^t \leq \varphi(\mathsf{K}) \tilde{\rho}(A_{\mathsf{K}})^t$, where we define $\varphi(\mathsf{K}) \coloneqq \left\| \tilde{P}_{\mathsf{K}}^{-\frac{1}{2}} \right\| \left\| \tilde{P}_{\mathsf{K}}^{\frac{1}{2}} \right\|$. It's easy to see that $\varphi(\mathsf{K}) \geq \left\| P_{\mathsf{K}}^{-\frac{1}{2}} P_{\mathsf{K}}^{\frac{1}{2}} \right\| = 1$, and by the results of perturbation analysis of Lyapunov equations [224], we can see that $\varphi(\mathsf{K})$ is a continuous function over $\mathsf{K} \in \mathscr{K}_{\mathrm{st}}$. □

C.2.2 Proof of Lemma C.1

Firstly, notice that $(D_i^0)_{i=1}^N \sim \mathrm{Uni}(\mathbb{S}_{n_K})$ is a direct consequence of the isotropy of the standard Gaussian distribution. The rest of the section will focus on proving (C.1).

Let $q(t) = [q_1(t), \cdots, q_N(t)]^\top$, where $q_i(t)$ is defined in `SampleUSphere`. Notice that $q(t) = Wq(t-1)$ and $\mathbf{1}^\top q(t) = \mathbf{1}^\top q(t-1)$ for $t \geq 1$. Consequently, $\frac{Nq(t)}{\mathbf{1}^\top q(0)} - \mathbf{1} = \left(W - N^{-1} \mathbf{1}\mathbf{1}^\top \right) \left(\frac{Nq(t-1)}{\mathbf{1}^\top q(0)} - \mathbf{1} \right)$, where we use the fact that W is doubly stochastic. Thus,

$$\left| \frac{Nq_i(t)}{\mathbf{1}^\top q(0)} - 1 \right| \leq \left\| \frac{Nq(t)}{\mathbf{1}^\top q(0)} - \mathbf{1} \right\| \leq \rho_W^t \left\| \frac{Nq(0)}{\mathbf{1}^\top q(0)} - \mathbf{1} \right\| \leq N \rho_W^t,$$

where the last inequality uses $\|Nv - \mathbf{1}\| \le N$ for any $v \in \mathbb{R}^N$ such that $v_i \ge 0$ and $\mathbf{1}^\top v = 1$. We then have

$$\begin{aligned}
&\left| \frac{1}{\sqrt{Nq_i(T_S)}} - \frac{1}{\sqrt{\mathbf{1}^\top q(0)}} \right| \\
&= \frac{1}{\sqrt{\mathbf{1}^\top q(0)}} \cdot \frac{\left| Nq_i(T_S)/\mathbf{1}^\top q(0) - 1 \right|}{\sqrt{Nq_i(T_S)/\mathbf{1}^\top q(0)}(1 + \sqrt{Nq_i(T_S)/\mathbf{1}^\top q(0)})} \\
&\le \frac{1}{\sqrt{\mathbf{1}^\top q(0)}} \cdot \frac{N\rho_W^{T_S}}{\sqrt{1-N\rho_W^{T_S}}\left(1 + \sqrt{1-N\rho_W^{T_S}}\right)} \le \frac{N\rho_W^{T_S}}{\sqrt{\mathbf{1}^\top q(0)}},
\end{aligned}$$

where the last inequality uses that $N\rho_W^{T_S} \le 1/2$ and $\inf_{x\in[0,1/2]} \sqrt{1-x}\big(1 + \sqrt{1-x}\big) \ge 1$. Finally, we obtain

$$\begin{aligned}
\|D_i - D_i^0\|_F &= \|V_i\|_F \left| \frac{1}{\sqrt{Nq_i(T_J)}} - \frac{1}{\sqrt{1^\top q_0}} \right| \\
&\le \frac{\|V_i\|_F}{\sqrt{1^\top q_0}} \cdot N\rho_W^{T_S} = \|D_i^0\|_F \cdot N\rho_W^{T_S}, \\
\sum\nolimits_{i=1}^N \|D_i\|_F^2 &\le \sum\nolimits_{i=1}^N \big(\|D_i^0\|_F + \|D_i - D_i^0\|_F\big)^2 \\
&\le \sum\nolimits_{i=1}^N \big(1+N\rho_W^{T_S}\big)^2 \|D_i^0\|_F^2 = \big(1+N\rho_W^{T_S}\big)^2.
\end{aligned}$$

C.2.3 Proof of Lemma C.2

This section analyzes the error of the estimated cost $\tilde{J}_i(\mathsf{K})$, also denoted as $\mu_i(T_J)$, generated by `GlobalCostEst`.

The main insight behind the proof is that $\mu_i(T_J)$ can be represented by quadratic forms of a Gaussian vector (see Lemma C.8). The proof follows by utilizing the properties of the quadratic forms of Gaussian vectors (Proposition C.1).

(a) Representing $\mu_i(T_J)$ by Quadratic Gaussian. We define

$$\varpi := \left[(\Sigma_w^{-\frac{1}{2}} w(0))^\top, \dots, (\Sigma_w^{-\frac{1}{2}} w(T_J - 1))^\top \right]^\top,$$

$$\Psi := \begin{bmatrix} \Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}}\Sigma_w^{\frac{1}{2}} & & & \\ \Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}}A_{\mathsf{K}}\Sigma_w^{\frac{1}{2}} & \Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}}\Sigma_w^{\frac{1}{2}} & & \\ \vdots & \vdots & \ddots & \\ \Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}}A_{\mathsf{K}}^{T_J-1}\Sigma_w^{\frac{1}{2}} & \Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}}A_{\mathsf{K}}^{T_J-2}\Sigma_w^{\frac{1}{2}} & \cdots & \Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}}\Sigma_w^{\frac{1}{2}} \end{bmatrix},$$

$$\Phi_\gamma := \Psi^\top \cdot \mathrm{blkdiag}\left[\left(\gamma^{T_J-t}\Sigma_{\mathsf{K},\infty}^{\frac{1}{2}}Q_{\mathsf{K}}\Sigma_{\mathsf{K},\infty}^{\frac{1}{2}}\right)_{t=1}^{T_J}\right]\cdot\Psi, \quad \gamma\in[0,1],$$

where $\mathrm{blkdiag}[(M_l)_{l=1}^p]$ denotes the block diagonal matrix formed by $M_1,\dots,M_p$. Notice that $\varpi \sim \mathscr{N}(0, I_{nT_J})$.

The following lemma shows that $\mu_i(T_J)$ can be written as a quadratic form in terms of the above auxiliary quantities.

Lemma C.8 (Quadratic Gaussian representation)**.** *For $\gamma \in [0,1]$,*

$$\sum_{t=1}^{T_J}\gamma^{T_J-t}x(t)^\top Q_{\mathsf{K}}x(t) = \varpi^\top\Phi_\gamma\varpi. \tag{C.17}$$

Moreover, for any $1 \le i \le N$, the global objective estimation $\mu_i(T_J)$ (a.k.a. $\tilde{J}_i(\mathsf{K})$) satisfies

$$\left|\mu_i(T_J) - \frac{1}{T_J}\varpi^\top\Phi_1\varpi\right| \le \frac{N}{T_J}\varpi^\top\Phi_{\rho_W}\varpi. \tag{C.18}$$

Proof. We first prove (C.17). For a closed-loop system $x(t+1) = A_{\mathsf{K}}x(t) + w(t)$ started with $x(0) = 0$, we have $x(t) = \sum_{\tau=1}^t A_{\mathsf{K}}^{t-\tau}w(\tau-1)$. Then, by the definitions of ϖ and Φ_γ, we have $\sum_{t=1}^{T_J}\gamma^{T_J-t}x(t)^\top Q_{\mathsf{K}}x(t) = \sum_{t=1}^{T_J}\gamma^{T_J-t}\Big(\sum_{\tau=1}^t w(\tau-1)^\top\big(A_{\mathsf{K}}^\top\big)^{t-\tau}\Big)Q_{\mathsf{K}}\Big(\sum_{\tau=1}^t A_{\mathsf{K}}^{t-\tau}w(\tau-1)\Big) = \sum_{t=1}^{T_J}(\Psi\varpi)_t^\top(\gamma^{T_J-t}\Sigma_{\mathsf{K},\infty}^{\frac{1}{2}}Q_{\mathsf{K}}\Sigma_{\mathsf{K},\infty}^{\frac{1}{2}})(\Psi\varpi)_t = \varpi^\top\Phi_\gamma\varpi$, where $(\Psi\varpi)_t = \sum_{\tau=1}^t\Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}}A_{\mathsf{K}}^{t-\tau}w(\tau-1)$ is the t'th block of $\Psi\varpi$.

Next, we prove the inequality (C.18). Notice that

$$\left|T_J\mu_i(T_J) - \varpi^\top\Phi_1\varpi\right|$$

$$
\begin{aligned}
&= \left| \sum_{t=1}^{T_J} \sum_{j=1}^{N} \left[W^{T_J - t} \right]_{ij} x(t)^\top Q_{j,\mathsf{K}} x(t) - \sum_{t=1}^{T_J} x(t)^\top Q_{\mathsf{K}} x(t) \right| \\
&= \left| \sum\nolimits_{t=1}^{T_J} \sum\nolimits_{j=1}^{N} \left(\left[W^{T_J - t} \right]_{ij} - \frac{1}{N} \right) x(t)^\top Q_{j,\mathsf{K}} x(t) \right| \\
&= \left| \sum\nolimits_{t=1}^{T_J} \sum\nolimits_{j=1}^{N} \left[\left(W - \frac{1}{N} \mathbf{1}\mathbf{1}^\top \right)^{T_J - t} \right]_{ij} x(t)^\top Q_{j,\mathsf{K}} x(t) \right| \\
&\leq \sum\nolimits_{t=1}^{T_J} \left| \sum\nolimits_{j=1}^{N} \left[\left(W - \frac{1}{N} \mathbf{1}\mathbf{1}^\top \right)^{T_J - t} \right]_{ij} x(t)^\top Q_{j,\mathsf{K}} x(t) \right| \\
&\leq \sum\nolimits_{t=1}^{T_J} \rho_W^{T_J - t} \sqrt{\sum\nolimits_{j=1}^{N} \left(x(t)^\top Q_{j,\mathsf{K}} x(t) \right)^2} \\
&\leq \sum\nolimits_{t=1}^{T_J} \rho_W^{T_J - t} \sum\nolimits_{j=1}^{N} x(t)^\top Q_{j,\mathsf{K}} x(t) \\
&= N \sum\nolimits_{t=1}^{T_J} \rho_W^{T_J - t} x(t)^\top Q_{\mathsf{K}} x(t) = N \varpi^\top \Phi_{\rho_W} \varpi,
\end{aligned}
$$

where the first step uses the definition of $\mu_i(T_J)$ and (C.17); the second step uses the definition of Q_{K}; the third step uses a property of a doubly stochastic matrix W that $\left(W - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\right)^t = W^t - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$; the fifth step uses the fact that for any vector $v \in \mathbb{R}^N$, we have $\left| \sum_{j=1}^{N} \left[\left(W - N^{-1}\mathbf{1}\mathbf{1}^\top\right)^{T_J - t} \right]_{ij} v_j \right| \leq \left\| \left(W - N^{-1}\mathbf{1}\mathbf{1}^\top\right)^{T_J - t} v \right\|_\infty \leq \left\| \left(W - N^{-1}\mathbf{1}\mathbf{1}^\top\right)^{T_J - t} v \right\| \leq \rho_W^{T_J - t} \|v\|$; the sixth step follows from $\|v\| \leq \mathbf{1}^\top v$ for any vector v with nonnegative entries; the last step uses (C.17). □

(b) Properties of the Parameter Matrices Φ_{ρ_W} and Φ_1.

Lemma C.9 (Properties of Φ_{ρ_W} and Φ_1)**.** *The parameter matrices Φ_{ρ_W} and Φ_1 in the quadratic Gaussian representation enjoy the following properties.*

- Φ_{ρ_W} *satisfies* $\operatorname{tr}\left(\Phi_{\rho_W}\right) \leq \frac{J(\mathsf{K})}{1-\rho_W}$.
- Φ_1 *satisfies* $\left\|\Phi_1\right\| \leq J(\mathsf{K}) \left(\frac{2\varphi(\mathsf{K})}{1-\rho(A_{\mathsf{K}})}\right)^2$ *and* $\left\|\Phi_1\right\|_F^2 \leq J(\mathsf{K})^2 n T_J \left(\frac{2\varphi(\mathsf{K})}{1-\rho(A_{\mathsf{K}})}\right)^4$, *where $\varphi(\mathsf{K})$ is defined in Lemma C.7.*

Proof. • For Φ_{ρ_W}, by $\varpi \sim \mathcal{N}(0, I_{nT_J})$, Lemma C.8, and $\mathbb{E}\big[x(t)x(t)^\top\big] \preceq \Sigma_{\mathsf{K},\infty}$, we have $\operatorname{tr}(\Phi_{\rho_W}) = \mathbb{E}[\varpi^\top \Phi_{\rho_W} \varpi] = \mathbb{E}[\sum_{t=1}^{T_J} \rho_W^{T_J-t} x(t)^\top Q_{\mathsf{K}} x(t)] \le \sum_{t=1}^{T_J} \rho_W^{T_J-t} \operatorname{tr}(Q_{\mathsf{K}}\Sigma_{\mathsf{K},\infty}) = J(\mathsf{K}) \sum_{t=1}^{T_J} \rho_W^{T_J-t} \le \frac{J(\mathsf{K})}{1-\rho_W}$.

• For Φ_1, since the matrix $\Sigma_{\mathsf{K},\infty}^{\frac{1}{2}} Q_{\mathsf{K}} \Sigma_{\mathsf{K},\infty}^{\frac{1}{2}}$ is positive definite, we have that $\big\|\Phi_1\big\| \le \Big\|\Sigma_{\mathsf{K},\infty}^{\frac{1}{2}} Q_{\mathsf{K}} \Sigma_{\mathsf{K},\infty}^{\frac{1}{2}}\Big\| \big\|\Psi\big\|^2 \le \operatorname{tr}(\Sigma_{\mathsf{K},\infty}^{\frac{1}{2}} Q_{\mathsf{K}} \Sigma_{\mathsf{K},\infty}^{\frac{1}{2}}) \big\|\Psi\big\|^2 = \operatorname{tr}(Q_{\mathsf{K}}\Sigma_{\mathsf{K},\infty}) \big\|\Psi\big\|^2 = J(\mathsf{K}) \big\|\Psi\big\|^2$.

Now it suffices to bound $\big\|\Psi\big\|^2$. Consider any $v = \begin{bmatrix} v_0^\top & v_1^\top & \cdots & v_{T_J-1}^\top \end{bmatrix}^\top \in \mathbb{S}_{nT_J}$, and then we have

$$
\begin{aligned}
&\big\|\Psi v\big\|^2 \\
&= \sum_{t=1}^{T_J} \sum_{\tau,\tau'=1}^{t} v_{\tau-1}^\top \Sigma_w^{\frac{1}{2}} \big(A_{\mathsf{K}}^\top\big)^{t-\tau} \Sigma_{\mathsf{K},\infty}^{-1} A_{\mathsf{K}}^{t-\tau'} \Sigma_w^{\frac{1}{2}} v_{\tau'-1} \\
&= \sum_{t=1}^{T_J} \sum_{\tau,\tau'=1}^{t} v_{\tau-1}^\top \Sigma_w^{\frac{1}{2}} \Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}} \big(\hat{A}_{\mathsf{K}}^\top\big)^{t-\tau} \hat{A}_{\mathsf{K}}^{t-\tau'} \Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}} \Sigma_w^{\frac{1}{2}} v_{\tau'-1} \\
&\le \sum_{t=1}^{T_J} \sum_{\tau,\tau'=1}^{t} \|v_{\tau-1}\| \, \Big\|\big(\hat{A}_{\mathsf{K}}^\top\big)^{t-\tau}\Big\| \, \Big\|\big(\hat{A}_{\mathsf{K}}\big)^{t-\tau'}\Big\| \, \|v_{\tau'-1}\| \\
&\le \varphi(\mathsf{K})^2 \sum_{t=1}^{T_J} \sum_{\tau,\tau'=1}^{t} \|v_{\tau-1}\| \cdot \tilde{\rho}(A_{\mathsf{K}})^{t-\tau} \cdot \tilde{\rho}(A_{\mathsf{K}})^{t-\tau'} \cdot \|v_{\tau'-1}\| \\
&= \varphi(\mathsf{K})^2 \left\| \underbrace{\begin{bmatrix} 1 & & & \\ \tilde{\rho}(A_{\mathsf{K}}) & 1 & & \\ \vdots & \vdots & \ddots & \\ \tilde{\rho}(A_{\mathsf{K}})^{T_J-1} & \tilde{\rho}(A_{\mathsf{K}})^{T_J-2} & \cdots & 1 \end{bmatrix}}_{\mathbf{H}^{(T_J)}} \underbrace{\begin{bmatrix} \|v_0\| \\ \|v_1\| \\ \vdots \\ \|v_{T-1}\| \end{bmatrix}}_{\mathbf{v}} \right\|^2 \\
&\le \varphi(\mathsf{K})^2 \big\|\mathbf{H}^{(T_J)}\big\|^2 \|\mathbf{v}\|^2 = \varphi(\mathsf{K})^2 \|\mathbf{H}^{(T_J)}\|^2
\end{aligned}
$$

where the second step denotes $\hat{A}_{\mathsf{K}} = \Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}} A_{\mathsf{K}} \Sigma_{\mathsf{K},\infty}^{\frac{1}{2}}$; the third step uses $\Sigma_w \preceq \Sigma_{\mathsf{K},\infty}$ and thus $\big\|\Sigma_{\mathsf{K},\infty}^{-1/2} \Sigma_w^{1/2}\big\| \le 1$; the fourth step uses Lemma C.7 and $\tilde{\rho}(A_{\mathsf{K}}) = (1+\rho(A_{\mathsf{K}}))/2$; the last step is because v is on the unit sphere.

Notice that $\mathbf{H}^{(T_J)}$ can be viewed as a finite-horizon truncation of the block-Toeplitz representation of the linear system with transfer function $\mathbf{H}(z) = \sum_{t=0}^{\infty} \tilde{\rho}(A_{\mathsf{K}})^t z^{-t} = 1/(1-\tilde{\rho}(A_{\mathsf{K}})z^{-1})$. Therefore, $\left\|\mathbf{H}^{(T_J)}\right\| \le \|\mathbf{H}\|_{\mathscr{H}_\infty} = \sup_{\|z\|=1}|\mathbf{H}(z)| = \frac{1}{1-\tilde{\rho}(A_{\mathsf{K}})} = \frac{2}{1-\rho(A_{\mathsf{K}})}$. This completes the proof of the bound on $\|\Phi_1\|$. The bound on $\left\|\Phi_1\right\|_F^2$ follows from $\left\|\Phi_1\right\|_F^2 \le nT_J\left\|\Phi_1\right\|^2$. □

(c) Bounding the Bias and Second Moment. The proof relies on the following properties of quadratic Gaussian variables.

Proposition C.1 ([225, Theorems 1.5 & 1.6])**.** *Let $z \sim \mathcal{N}(0, I_p)$, and let $M \in \mathbb{R}^{p\times p}$ be any symmetric matrix. Then we have $\mathbb{E}\left[z^\top M z\right] = \mathrm{tr}(M)$ and $\mathrm{Var}\left(z^\top M z\right) = 2\left\|M\right\|_F^2$.*

Now we are ready for the proof of Lemma C.2.

Firstly, we consider the bias.

$$
\begin{aligned}
&|\mathbb{E}[\mu_i(T_J)] - J(\mathsf{K})| \\
\le& \left|\mathbb{E}\left[\mu_i(T_J) - \frac{1}{T_J}\varpi^\top \Phi_1 \varpi\right]\right| + \left|\mathbb{E}\left[\frac{1}{T_J}\varpi^\top \Phi_1 \varpi\right] - J(\mathsf{K})\right| \\
\le& \ \mathbb{E}\left[\frac{N}{T_J}\varpi^\top \Phi_{\rho_W} \varpi\right] + \left|\frac{1}{T_J}\mathbb{E}\left[\sum_{t=1}^{T_J} x(t)^\top Q_{\mathsf{K}} x(t)\right] - J(\mathsf{K})\right| \\
=& \ \frac{N}{T_J}\mathrm{tr}(\Phi_{\rho_W}) + \frac{1}{T_J}\left|\sum\nolimits_{t=1}^{T_J} \mathrm{tr}(Q_{\mathsf{K}}(\Sigma_{\mathsf{K},t} - \Sigma_{\mathsf{K},\infty}))\right| \\
\le& \ \frac{N}{T_J}\frac{J(\mathsf{K})}{1-\rho_W} + \frac{1}{T_J}\sum\nolimits_{t=1}^{T_J}\left|\mathrm{tr}\left(Q_{\mathsf{K}}\sum\nolimits_{\tau=t}^{\infty} A_{\mathsf{K}}^{\tau}\Sigma_w\left(A_{\mathsf{K}}^\top\right)^\tau\right)\right| \\
\le& \ \frac{N}{T_J}\frac{J(\mathsf{K})}{1-\rho_W} + \frac{J(K)}{T_J}\left(\frac{2\varphi(\mathsf{K})}{1-\rho(A_{\mathsf{K}})}\right)^2
\end{aligned}
$$

where the second step uses Lemma C.8; the third step follows from $\varpi \sim \mathcal{N}(0, I_{nT_J})$, (C.14) and (C.16); the fourth step uses Lemma C.9 and (C.15); the last step uses the

following fact:

$$\begin{aligned}
&\sum_{t=1}^{T_J}\left|\mathrm{tr}\left(Q_{\mathsf{K}}\sum_{\tau=t}^{\infty}A_{\mathsf{K}}^{\tau}\Sigma_w\big(A_{\mathsf{K}}^{\top}\big)^{\tau}\right)\right|\\
&=\sum_{t=1}^{T_J}\left|\mathrm{tr}\left(\Sigma_{\mathsf{K},\infty}^{\frac{1}{2}}Q_{\mathsf{K}}\Sigma_{\mathsf{K},\infty}^{\frac{1}{2}}\sum_{\tau=t}^{\infty}\hat{A}_{\mathsf{K}}^{\tau}\hat{\Sigma}_w\big(\hat{A}_{\mathsf{K}}^{\top}\big)^{\tau}\right)\right|\\
&\le\sum_{t=1}^{T_J}\mathrm{tr}\big(\Sigma_{\mathsf{K},\infty}^{\frac{1}{2}}Q_{\mathsf{K}}\Sigma_{\mathsf{K},\infty}^{\frac{1}{2}}\big)\left\|\sum_{\tau=t}^{\infty}\hat{A}_{\mathsf{K}}^{\tau}\hat{\Sigma}_w\big(\hat{A}_{\mathsf{K}}^{\top}\big)^{\tau}\right\|\\
&\le\sum_{t=1}^{T_J}J(\mathsf{K})\sum_{\tau=t}^{\infty}\big\|\hat{A}_{\mathsf{K}}^{\tau}\big\|^2\big\|\hat{\Sigma}_w\big\|\\
&\le\sum_{t=1}^{T_J}J(\mathsf{K})\sum_{\tau=t}^{\infty}\varphi(\mathsf{K})^2\left(\frac{1+\rho(A_{\mathsf{K}})}{2}\right)^{2\tau}\le J(\mathsf{K})\varphi(\mathsf{K})^2\left(\frac{2}{1-\rho(A_{\mathsf{K}})}\right)^2,
\end{aligned}$$

where we denote $\hat{A}_{\mathsf{K}}=\Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}}A_{\mathsf{K}}\Sigma_{\mathsf{K},\infty}^{\frac{1}{2}}$ and $\hat{\Sigma}_w=\Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}}\Sigma_w\Sigma_{\mathsf{K},\infty}^{-\frac{1}{2}}$, the fourth step uses Lemma C.7 and $\big\|\hat{\Sigma}_w\big\|\le 1$, the last step uses $\sum_{t=1}^{\infty}\sum_{\tau=t}^{\infty}\left(\frac{1+\rho(A_{\mathsf{K}})}{2}\right)^{2\tau}\le\left(\frac{2}{1-\rho(A_{\mathsf{K}})}\right)^2$.

Define the constant β_0 as the following:

$$\beta_0\coloneqq\sup_{\mathsf{K}\in\mathscr{Q}^1}\left(\frac{2\varphi(\mathsf{K})}{1-\rho(A_{\mathsf{K}})}\right)^2. \tag{C.19}$$

Lemmas 8.1, C.7 and the continuity of the map $\mathsf{K}\mapsto\rho(A_{\mathsf{K}})$ ensure that β_0 is finite and only depends on the system parameters A, B, Σ_w,Q_i, R_i as well as the initial cost $J(\mathsf{K}_0)$. By substituting β_0 into the inequality above, we prove (C.2) for any $\mathsf{K}\in\mathscr{Q}^1$.

Next, we bound $\mathbb{E}[(\mu_i(T_J)-J(\mathsf{K}))^2]$. By (C.2), we have: $\mathbb{E}[(\mu_i(T_J)-J(\mathsf{K}))^2]\le(\mathbb{E}[\mu_i(T_J)-J(\mathsf{K})])^2+\mathrm{Var}(\mu_i(T_J))\le\frac{J(\mathsf{K})^2}{T_J^2}\left[2\left(\frac{N}{1-\rho_W}\right)^2+2\beta_0^2\right]+\mathrm{Var}(\mu_i(T_J))$. Then, we can bound then $\mathrm{Var}(\mu_i(T_J))$ below:

$$\begin{aligned}
&\mathrm{Var}(\mu_i(T_J))\\
&\le 2\,\mathrm{Var}\left(\mu_i(T_J)-\frac{1}{T_J}\varpi^{\top}\Phi_1\varpi\right)+2\,\mathrm{Var}\left(\frac{1}{T_J}\varpi^{\top}\Phi_1\varpi\right)\\
&\le 2\,\mathbb{E}\left[\left|\mu_i(T_J)-\frac{1}{T_J}\varpi^{\top}\Phi_1\varpi\right|^2\right]+\frac{4}{T_J^2}\|\Phi_1\|_F^2\\
&\le 2\,\frac{N^2}{T_J^2}\,\mathbb{E}\Big[\big(\varpi^{\top}\Phi_{\rho_W}\varpi\big)^2\Big]+\frac{4nJ(\mathsf{K})^2}{T_J}\beta_0^2
\end{aligned}$$

$$= \frac{2N^2}{T_J^2}\Big(\big(\mathbb{E}[\varpi^\top \Phi_{\rho_W}\varpi]\big)^2 + \mathrm{Var}\big(\varpi^\top \Phi_{\rho_W}\varpi\big)\Big) + \frac{4nJ(\mathsf{K})^2}{T_J}\beta_0^2$$
$$\le \frac{6J(\mathsf{K})^2}{T_J^2}\left(\frac{N}{1-\rho_W}\right)^2 + \frac{4nJ(\mathsf{K})^2}{T_J}\beta_0^2.$$

where we use Proposition C.1, Lemmas C.8 and C.9, $\|M\|_F \le \mathrm{tr}(M)$ for any positive semidefinite M, $\mathsf{K} \in \mathscr{Q}^1$ and (C.19). Finally, we obtain (C.3) by $1/T_J^2 \le n/T_J$.

C.2.4 Proof of Lemma C.3

The proof is based on the following concentration inequality.

Proposition C.2 ([226]). *Let $z \sim \mathcal{N}(0, I_p)$, and let $M \in \mathbb{R}^{p\times p}$ be any symmetric positive definite matrix. Then for any $\delta \ge 0$, $\mathbb{P}\Big(z^\top M z > \mathrm{tr}\, M + 2\|M\|_F\sqrt{\delta} + 2\|M\|\delta\Big) \le e^{-\delta}$.*

By Lemma C.8, we have $\mu_i(T_J) \le T_J^{-1}\varpi^\top(\Phi_1 + N\Phi_{\rho_W})\,\varpi$. Therefore for any $\varepsilon_1 \ge 1$ and $\varepsilon_2 \ge 0$, we have

$$\begin{aligned}
&\mathbb{P}\left(\mu_i(T_J) > (\varepsilon_1+\varepsilon_2)J(\mathsf{K})\right)\\
&\le \mathbb{P}\left(\frac{1}{T_J}\varpi^\top(\Phi_1 + N\Phi_{\rho_W})\varpi > (\varepsilon_1+\varepsilon_2)J(\mathsf{K})\right)\\
&\le \mathbb{P}\left(\frac{1}{T_J}\varpi^\top\Phi_1\varpi > \varepsilon_1 J(\mathsf{K}) + \frac{\mathrm{tr}\big(\Phi_1\big)}{T_J} - J(\mathsf{K})\right)\\
&\quad + \mathbb{P}\left(\frac{N}{T_J}\varpi^\top\Phi_{\rho_W}\varpi > \varepsilon_2 J(\mathsf{K}) + J(\mathsf{K}) - \frac{\mathrm{tr}\big(\Phi_1\big)}{T_J}\right)\\
&\le \mathbb{P}\left(\frac{1}{T_J}\varpi^\top\Phi_1\varpi > \varepsilon_1 J(\mathsf{K}) + \frac{\mathrm{tr}\big(\Phi_1\big)}{T_J} - J(\mathsf{K})\right)\\
&\quad + \mathbb{P}\left(\frac{N}{T_J}\varpi^\top\Phi_{\rho_W}\varpi > \varepsilon_2 J(\mathsf{K})\right),
\end{aligned}$$

where we used $J(\mathsf{K}) \ge T_J^{-1}\sum_{t=1}^{T_J}\mathbb{E}\big[x(t)^\top Q_{\mathsf{K}}x(t)\big] = T_J^{-1}\,\mathrm{tr}\big(\Phi_1\big)$ by (C.15), (C.17) and Proposition C.1. For the first term, by Proposition C.2 and the bound $\big\|\Phi_1\big\|_F \le \sqrt{nT_J}\big\|\Phi_1\big\|$, we get

$$\mathbb{P}\Big(\varpi^\top\Phi_1\varpi > \mathrm{tr}\big(\Phi_1\big) + 2\big\|\Phi_1\big\|\sqrt{nT_J\delta} + 2\big\|\Phi_1\big\|\delta\Big) \le e^{-\delta},$$

for any $\delta \geq 0$, and by letting δ satisfy $2\|\Phi_1\|\sqrt{nT_J\delta} + 2\|\Phi_1\|\delta = (\varepsilon_1 - 1)T_J J(\mathsf{K})$ with $\varepsilon_1 \geq 1$, we can get

$$\begin{aligned}
&\mathbb{P}\left(\frac{1}{T_J}\varpi^\top \Phi_1 \varpi > \varepsilon_1 J(\mathsf{K}) + \frac{\operatorname{tr}(\Phi_1)}{T_J} - J(\mathsf{K})\right) \\
&\leq \exp\left[-\frac{1}{4}\left(\sqrt{2\frac{(\varepsilon_1 - 1)T_J J(\mathsf{K})}{\|\Phi_1\|} + nT_J} - \sqrt{nT_J}\right)^2\right] \\
&\leq \exp\left[-\frac{1}{4}\min\left\{\frac{(\varepsilon_1 - 1)T_J J(\mathsf{K})}{\|\Phi_1\|}, \frac{(\varepsilon_1 - 1)^2 T_J J(\mathsf{K})^2}{4n\|\Phi_1\|^2}\right\}\right] \\
&\leq \exp\left(-\frac{(\varepsilon_1 - 1)T_J J(\mathsf{K})}{4\|\Phi_1\|}\right) + \exp\left(-\frac{(\varepsilon_1 - 1)^2 T_J J(\mathsf{K})^2}{16n\|\Phi_1\|^2}\right),
\end{aligned}$$

where we used $\left(\sqrt{2x + nT_J} - \sqrt{nT_J}\right)^2 \geq \min\{x, x^2/(4nT_J)\}$ for all $x \geq 0$ in the second inequality. For the second term, by Proposition C.2 and the bound $\|\Phi_{\rho_W}\| \leq \|\Phi_{\rho_W}\|_F \leq \operatorname{tr}(\Phi_{\rho_W})$, we obtain

$$\mathbb{P}\left(\varpi^\top \Phi_{\rho_W} \varpi > \operatorname{tr}(\Phi_{\rho_W})\left(1 + 2\sqrt{\delta} + 2\delta\right)\right) \leq e^{-\delta}$$

for any $\delta \geq 0$, and by letting $\delta = \frac{1}{4}\left(\sqrt{\frac{2T_J\varepsilon_2 J(\mathsf{K})}{N\operatorname{tr}(\Phi_{\rho_W})} - 1} - 1\right)^2$ for $\varepsilon_2 \geq N\operatorname{tr}(\Phi_{\rho_W})/(T_J J(\mathsf{K}))$, we obtain

$$\begin{aligned}
\mathbb{P}\left(\frac{N\varpi^\top \Phi_{\rho_W}\varpi}{T_J} > \varepsilon_2 J(\mathsf{K})\right) &\leq \exp\left[-\frac{1}{4}\left(\sqrt{2\frac{\varepsilon_2 T_J J(\mathsf{K})}{N\operatorname{tr}(\Phi_{\rho_W})} - 1} - 1\right)^2\right] \\
&\leq \exp\left[-\frac{1}{3}\left(\frac{\varepsilon_2 T_J J(\mathsf{K})}{N\operatorname{tr}(\Phi_{\rho_W})} - 2\right)\right],
\end{aligned}$$

where we used $\left(\sqrt{2x-1} - 1\right)^2 \geq \frac{4}{3}(x-2)$ for any $x > 1$.

Thus, by letting $\varepsilon_1 = 4\varepsilon/5$ and $\varepsilon_2 = \varepsilon/5$, we obtain

$$\begin{aligned}
&\mathbb{P}\left(\mu_i(T_J) > \varepsilon J(\mathsf{K})\right) \\
&\leq \exp\left(-\frac{(4\varepsilon/5 - 1)T_J J(\mathsf{K})}{4\|\Phi_1\|}\right) + \exp\left(-\frac{(4\varepsilon/5 - 1)^2 T_J J(\mathsf{K})^2}{16n\|\Phi_1\|^2}\right) \\
&\quad + \exp\left[-\frac{1}{3}\left(\frac{\varepsilon T_J J(\mathsf{K})}{5N\operatorname{tr}(\Phi_{\rho_W})} - 2\right)\right]
\end{aligned}$$

for $\varepsilon \geq 5N \operatorname{tr}\big(\Phi_{\rho_W}\big)/(T_J J(\mathsf{K}))$. Now we have

$$\begin{aligned}
&\mathbb{E}\left[\mu_i(T_J) - \min\big\{\mu_i(T_J), \bar{J}\big\}\right] \\
&= \int_0^{+\infty} \mathbb{P}\big(\mu_i(T_J) - \min\big\{\mu_i(T_J), \bar{J}\big\} \geq x\big)\, dx \\
&= \int_0^{+\infty} \mathbb{P}\big(\mu_i(T_J) \geq \bar{J} + x\big)\, dx \\
&= J(\mathsf{K}) \int_{\bar{J}/J(\mathsf{K})}^{+\infty} \mathbb{P}(\mu_i(T_J) \geq \varepsilon J(\mathsf{K}))\, d\varepsilon.
\end{aligned}$$

By using $e^{-x} < 1/(2x)$ and $\int_x^{+\infty} e^{-u^2} du < e^{-x^2}/(2x)$ for any $x > 0$, we can see that

$$\begin{aligned}
&\int_{\bar{J}/J(\mathsf{K})}^{+\infty} \exp\left(-\frac{(4\varepsilon/5 - 1)T_J J(\mathsf{K})}{4\big\|\Phi_1\big\|}\right) d\varepsilon \\
&= \frac{5\big\|\Phi_1\big\|}{T_J J(\mathsf{K})} \exp\left[-\frac{T_J J(\mathsf{K})}{4\big\|\Phi_1\big\|}\left(\frac{4\bar{J}}{5J(\mathsf{K})} - 1\right)\right] \\
&< \frac{10\big\|\Phi_1\big\|^2}{T_J^2 J(\mathsf{K})^2} \cdot \frac{1}{4\bar{J}/(5J(\mathsf{K})) - 1}, \\
&\int_{\bar{J}/J(\mathsf{K})}^{+\infty} \exp\left(-\frac{(4\varepsilon/5 - 1)^2 T_J J(\mathsf{K})^2}{16n\big\|\Phi_1\big\|^2}\right) d\varepsilon \\
&< \frac{10n\big\|\Phi_1\big\|^2}{T_J J(\mathsf{K})^2} \frac{\exp\left[-\frac{T_J J(\mathsf{K})^2}{16n\big\|\Phi_1\big\|^2}\left(\frac{4\bar{J}}{5J(\mathsf{K})} - 1\right)^2\right]}{4\bar{J}/(5J(\mathsf{K})) - 1} \\
&< \frac{80n^2\big\|\Phi_1\big\|^4}{T_J^2 J(\mathsf{K})^4} \frac{1}{\big(4\bar{J}/(5J(\mathsf{K})) - 1\big)^3}, \\
&\int_{\bar{J}/J(\mathsf{K})}^{+\infty} \exp\left[-\frac{1}{3}\left(\frac{\varepsilon T_J J(\mathsf{K})}{5N \operatorname{tr}\big(\Phi_{\rho_W}\big)} - 2\right)\right] d\varepsilon \\
&= \frac{15 e^{\frac{2}{3}} N \operatorname{tr}\big(\Phi_{\rho_W}\big)}{T_J J(\mathsf{K})} \exp\left(-\frac{\bar{J} T_J}{15N \operatorname{tr}\big(\Phi_{\rho_W}\big)}\right) \\
&< \frac{225N^2 \operatorname{tr}\big(\Phi_{\rho_W}\big)^2}{T_J^2 J(\mathsf{K})^2} \cdot \frac{J(\mathsf{K})}{\bar{J}}.
\end{aligned}$$

Finally, by Lemma C.9 and the condition on $\bar{J}$, we see that

$$\mathbb{E}\left[\mu_i(T_J) - \min\big\{\mu_i(T_J), \bar{J}\big\}\right]$$

$$
\begin{aligned}
&\leq \frac{J(\mathsf{K})}{T_J^2}\left[10\left(\frac{2\varphi(\mathsf{K})}{1-\rho(A_{\mathsf{K}})}\right)^4+80n^2\left(\frac{2\varphi(\mathsf{K})}{1-\rho(A_{\mathsf{K}})}\right)^8+\frac{90N^2}{(1-\rho_W)^2}\right]\\
&\leq \frac{90J(\mathsf{K})}{T_J^2}\left[n^2\left(\frac{2\varphi(\mathsf{K})}{1-\rho(A_{\mathsf{K}})}\right)^8+\frac{N^2}{(1-\rho_W)^2}\right]\\
&\leq \frac{90J(\mathsf{K})}{T_J^2}\left[n^2\beta_0^4+\frac{N^2}{(1-\rho_W)^2}\right].
\end{aligned}
$$

The inequality $\mathbb{E}\left[\mu_i(T_J)-\min\left\{\mu_i(T_J),\bar{J}\right\}\right]\geq 0$ is obvious.

References

[1] X. Zhang, W. Shi, X. Li, B. Yan, A. Malkawi, and N. Li, "Decentralized temperature control via HVAC systems in energy efficient buildings: An approximate solution procedure," in *Proceedings of 2016 IEEE Global Conference on Signal and Information Processing*, pp. 936–940, 2016.

[2] L. Gan, A. Wierman, U. Topcu, N. Chen, and S. H. Low, "Real-time deferrable load control: handling the uncertainties of renewable generation," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 3, pp. 77–79, 2014.

[3] S. Kouro, P. Cortés, R. Vargas, U. Ammann, and J. Rodríguez, "Model predictive control—a simple and powerful method to control power converters," *IEEE Transactions on industrial electronics*, vol. 56, no. 6, pp. 1826–1838, 2008.

[4] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2016.

[5] F. Zanini, D. Atienza, G. De Micheli, and S. P. Boyd, "Online convex optimization-based algorithm for thermal management of mpsocs," in *Proceedings of the 20th symposium on Great lakes symposium on VLSI*, pp. 203–208, ACM, 2010.

[6] Y. Li, Y. Tang, R. Zhang, and N. Li, "Distributed reinforcement learning for decentralized linear quadratic control: A derivative-free policy optimization approach," *arXiv preprint arXiv:1912.09135*, 2019.

[7] E. Perea-Lopez, B. E. Ydstie, and I. E. Grossmann, "A model predictive control strategy for supply chain optimization," *Computers & Chemical Engineering*, vol. 27, no. 8-9, pp. 1201–1218, 2003.

[8] W. Wang, D. E. Rivera, and K. G. Kempf, "Model predictive control strategies for supply chain management in semiconductor manufacturing," *International Journal of Production Economics*, vol. 107, no. 1, pp. 56–77, 2007.

[9] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of

unmanned aerial vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6753–6760, IEEE, 2018.

[10] A. Gepperth and B. Hammer, "Incremental learning algorithms and applications," in *European symposium on artificial neural networks (ESANN)*, 2016.

[11] P. Kormushev, S. Calinon, and D. G. Caldwell, "Reinforcement learning in robotics: Applications and real-world challenges," *Robotics*, vol. 2, no. 3, pp. 122–148, 2013.

[12] A. Guillemin and N. Morel, "An innovative lighting controller integrated in a self-adaptive building control system," *Energy and buildings*, vol. 33, no. 5, pp. 477–487, 2001.

[13] K. Hazelwood, S. Bird, D. Brooks, S. Chintala, U. Diril, D. Dzhulgakov, M. Fawzy, B. Jia, Y. Jia, A. Kalro, *et al.*, "Applied machine learning at facebook: A datacenter infrastructure perspective," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 620–629, IEEE, 2018.

[14] E. Hazan, *Introduction to Online Convex Optimization*. Foundations and Trends(r) in Optimization Series, Now Publishers, 2016.

[15] S. Shalev-Shwartz, *Online Learning and Online Convex Optimization*. Foundations and Trends(r) in Machine Learning, Now Publishers, 2012.

[16] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 928–936, 2003.

[17] M. Lin, Z. Liu, A. Wierman, and L. L. Andrew, "Online algorithms for geographical load balancing," in *Green Computing Conference (IGCC), 2012 International*, pp. 1–10, IEEE, 2012.

[18] N. Chen, G. Goel, and A. Wierman, "Smoothed online convex optimization in high dimensions via online balanced descent," in *Conference On Learning Theory*, pp. 1574–1594, 2018.

[19] Y. Lin, G. Goel, and A. Wierman, "Online optimization with predictions and non-convex losses," *arXiv preprint arXiv:1911.03827*, 2019.

[20] O. Besbes, Y. Gur, and A. Zeevi, "Non-stationary stochastic optimization," *Operations research*, vol. 63, no. 5, pp. 1227–1244, 2015.

[21] D. Fay and J. V. Ringwood, "On the influence of weather forecast errors in short-term load forecasting models," *IEEE transactions on power systems*, vol. 25, no. 3, pp. 1751–1758, 2010.

[22] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 5, pp. 1378–1391, 2013.

[23] K.-D. Kim and P. R. Kumar, "An mpc-based approach to provable system-wide safety and liveness of autonomous ground traffic," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3341–3356, 2014.

[24] M. Diehl, R. Amrit, and J. B. Rawlings, "A lyapunov function for economic optimizing model predictive control," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 703–707, 2010.

[25] M. A. Müller and F. Allgöwer, "Economic and distributed model predictive control: Recent developments in optimization-based control," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 2, pp. 39–52, 2017.

[26] M. Ellis, H. Durand, and P. D. Christofides, "A tutorial review of economic model predictive control methods," *Journal of Process Control*, vol. 24, no. 8, pp. 1156–1178, 2014.

[27] A. Ferramosca, J. B. Rawlings, D. Limón, and E. F. Camacho, "Economic mpc for a changing economic criterion," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 6131–6136, IEEE, 2010.

[28] M. Ellis and P. D. Christofides, "Economic model predictive control with time-varying objective function for nonlinear process systems," *AIChE Journal*, vol. 60, no. 2, pp. 507–519, 2014.

[29] D. Angeli, A. Casavola, and F. Tedesco, "Theoretical advances on economic model predictive control with time-varying costs," *Annual Reviews in Control*, vol. 41, pp. 218–224, 2016.

[30] R. Amrit, J. B. Rawlings, and D. Angeli, "Economic optimization using model predictive control with a terminal cost," *Annual Reviews in Control*, vol. 35, no. 2, pp. 178–186, 2011.

[31] L. Grüne, "Economic receding horizon control without terminal constraints," *Automatica*, vol. 49, no. 3, pp. 725–734, 2013.

[32] D. Angeli, R. Amrit, and J. B. Rawlings, "On average performance and stability of economic model predictive control," *IEEE transactions on automatic control*, vol. 57, no. 7, pp. 1615–1626, 2012.

[33] L. Grüne and M. Stieler, "Asymptotic stability and transient optimality of economic mpc without terminal conditions," *Journal of Process Control*, vol. 24, no. 8, pp. 1187–1196, 2014.

[34] L. Grüne and A. Panin, "On non-averaged performance of economic mpc with terminal conditions," in *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 4332–4337, IEEE, 2015.

[35] A. Ferramosca, D. Limon, and E. F. Camacho, "Economic mpc for a changing economic criterion for linear systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2657–2667, 2014.

[36] L. Grüne and S. Pirkelmann, "Closed-loop performance analysis for economic model predictive control of time-varying systems," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 5563–5569, IEEE, 2017.

[37] L. Grüne and S. Pirkelmann, "Economic model predictive control for time-varying system: Performance and stability results," *Optimal Control Applications and Methods*, 2018.

[38] M. N. Zeilinger, C. N. Jones, and M. Morari, "Real-time suboptimal model predictive control using a combination of explicit mpc and online optimization," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1524–1534, 2011.

[39] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.

[40] K. Graichen and A. Kugi, "Stability and incremental improvement of suboptimal mpc without terminal constraints," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2576–2580, 2010.

[41] D. A. Allan, C. N. Bates, M. J. Risbeck, and J. B. Rawlings, "On the inherent robustness of optimal and suboptimal nonlinear mpc," *Systems & Control Letters*, vol. 106, pp. 68–78, 2017.

[42] N. Agarwal, B. Bullins, E. Hazan, S. Kakade, and K. Singh, "Online control with adversarial disturbances," in *International Conference on Machine Learning*, pp. 111–119, 2019.

[43] N. Agarwal, E. Hazan, and K. Singh, "Logarithmic regret for online control," in *Advances in Neural Information Processing Systems*, pp. 10175–10184, 2019.

[44] D. J. Foster and M. Simchowitz, "Logarithmic regret for adversarial online control," *arXiv preprint arXiv:2003.00189*, 2020.

[45] O. Plevrakis and E. Hazan, "Geometric exploration for online control," *arXiv preprint arXiv:2010.13178*, 2020.

[46] N. Chen, A. Agarwal, A. Wierman, S. Barman, and L. L. Andrew, "Online convex optimization using predictions," in *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pp. 191–204, 2015.

[47] A. Rakhlin and K. Sridharan, "Online learning with predictable sequences," in *Conference on Learning Theory*, pp. 993–1019, 2013.

[48] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4649–4672, 2010.

[49] M. Vasic and A. Billard, "Safety issues in human-robot interactions," in *2013 ieee international conference on robotics and automation*, pp. 197–204, IEEE, 2013.

[50] H. E. Roland and B. Moriarty, *System safety engineering and management*. John Wiley & Sons, 1990.

[51] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*, pp. 207–226, Springer, 1999.

[52] D. Q. Mayne, M. M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[53] W. Langson, I. Chryssochoos, S. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.

[54] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design*. Nob Hill Pub., 2009.

[55] X. Lu, M. Cannon, and D. Koksal-Rivet, "Robust adaptive model predictive control: Performance and parameter estimation," *International Journal of Robust and Nonlinear Control*, 2019.

[56] K. Zhang and Y. Shi, "Adaptive model predictive control for a class of constrained linear systems with parametric uncertainties," *Automatica*, vol. 117, p. 108974, 2020.

[57] M. Bujarbaruah, X. Zhang, M. Tanaskovic, and F. Borrelli, "Adaptive mpc under time varying uncertainty: Robust and stochastic," *arXiv preprint arXiv:1909.13473*, 2019.

[58] J. Köhler, E. Andina, R. Soloperto, M. A. Müller, and F. Allgöwer, "Linear robust adaptive model predictive control: Computational complexity and conservatism," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 1383–1388, IEEE, 2019.

[59] M. Fazel, R. Ge, S. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80 of *Proceedings of Machine Learning Research*, pp. 1467–1476, 2018.

[60] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the sample complexity of the linear quadratic regulator," *Foundations of Computational Mathematics*, pp. 1–47, 2019.

[61] H. Mania, S. Tu, and B. Recht, "Certainty equivalence is efficient for linear quadratic control," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 10154–10164, Curran Associates, Inc., 2019.

[62] Y. Ouyang, M. Gagrani, and R. Jain, "Learning-based control of unknown linear systems with Thompson sampling," *arXiv preprint arXiv:1709.04047*, 2017.

[63] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "Regret bounds for robust adaptive control of the linear quadratic regulator," in *Advances in Neural Information Processing Systems*, pp. 4188–4197, 2018.

[64] M. Simchowitz, K. Singh, and E. Hazan, "Improper learning for non-stochastic control," *arXiv preprint arXiv:2001.09254*, 2020.

[65] M. Simchowitz, H. Mania, S. Tu, M. I. Jordan, and B. Recht, "Learning without mixing: Towards a sharp analysis of linear system identification," in *Conference On Learning Theory*, pp. 439–473, PMLR, 2018.

[66] A. Cohen, T. Koren, and Y. Mansour, "Learning linear-quadratic regulators efficiently with only $\sqrt{\text{t}}$ regret," in *International Conference on Machine Learning*, pp. 1300–1309, PMLR, 2019.

[67] Y. Li and N. Li, "Mechanism design for reliability in demand response with uncertainty," in *2017 American Control Conference (ACC)*, pp. 3400–3405, IEEE, 2017.

[68] Y. Li, Q. Hu, and N. Li, "Learning and selecting the right customers for reliability: A multi-armed bandit approach," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 4869–4874, IEEE, 2018.

[69] Y. Li, Q. Hu, and N. Li, "A reliability-aware multi-armed bandit approach to learn and select users in demand response," *Automatica*, vol. 119, p. 109015, 2020.

[70] X. Chen, Y. Li, J. Shimada, and N. Li, "Online learning and distributed control for residential demand response," *IEEE Transactions on Smart Grid*, 2021.

[71] R. Rosales and S. Sclaroff, "Improved tracking of multiple humans with trajectory prediction and occlusion modeling," tech. rep., Boston University Computer Science Department, 1998.

[72] G. Goel and A. Wierman, "An online algorithm for smoothed regression and lqr control," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2504–2513, 2019.

[73] A. Pentina, V. Sharmanska, and C. H. Lampert, "Curriculum learning of multiple tasks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5492–5500, 2015.

[74] S.-M. Chen and J.-R. Hwang, "Temperature prediction using fuzzy time series," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 30, no. 2, pp. 263–275, 2000.

[75] P. Cortez, M. Rio, M. Rocha, and P. Sousa, "Multi-scale internet traffic forecasting using neural networks and time series methods," *Expert Systems*, vol. 29, no. 2, pp. 143–155, 2012.

[76] N. Chen, J. Comden, Z. Liu, A. Gandhi, and A. Wierman, "Using predictions in online optimization: Looking forward with an eye on the past," in *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*, pp. 193–206, ACM, 2016.

[77] G. Goel, Y. Lin, H. Sun, and A. Wierman, "Beyond online balanced descent: An optimal algorithm for smoothed online optimization," in *Advances in Neural Information Processing Systems*, pp. 1875–1885, 2019.

[78] M. Lin, A. Wierman, A. Roytman, A. Meyerson, and L. L. Andrew, "Online optimization with switching cost," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 3, pp. 98–100, 2012.

[79] L. Andrew, S. Barman, K. Ligett, M. Lin, A. Meyerson, A. Roytman, and A. Wierman, "A tale of two metrics: Simultaneous bounds on competitiveness and regret," in *Conference on Learning Theory*, pp. 741–763, 2013.

[80] O. Anava, E. Hazan, and S. Mannor, "Online learning for adversaries with memory: price of past mistakes," in *Advances in Neural Information Processing Systems*, pp. 784–792, 2015.

[81] G. Shi, Y. Lin, S.-J. Chung, Y. Yue, and A. Wierman, "Beyond no-regret: Competitive control via online optimization with memory," *arXiv preprint arXiv:2002.05318*, 2020.

[82] Y. Li, X. Chen, and N. Li, "Online optimal control with linear dynamics and predictions: Algorithms and regret analysis," in *Advances in Neural Information Processing Systems*, pp. 14887–14899, 2019.

[83] Y. Li, S. Das, and N. Li, "Online optimal control with affine constraints," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 8527–8537, 2021.

[84] G. Goel, N. Chen, and A. Wierman, "Thinking fast and slow: Optimization decomposition across timescales," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 1291–1298, IEEE, 2017.

[85] M. Badiei, N. Li, and A. Wierman, "Online convex optimization with ramp constraints," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pp. 6730–6736, IEEE, 2015.

[86] E. Even-Dar, S. M. Kakade, and Y. Mansour, "Online markov decision processes," *Mathematics of Operations Research*, vol. 34, no. 3, pp. 726–736, 2009.

[87] Y. Li and N. Li, "Online learning for markov decision processes in nonstationary environments: A dynamic regret analysis," in *2019 American Control Conference (ACC)*, pp. 1232–1237, IEEE, 2019.

[88] Y. Li, A. Zhong, G. Qu, and N. Li, "Online markov decision processes with time-varying transition probabilities and rewards," in *ICML workshop on Real-world Sequential Decision Making*, 2019.

[89] A. Alessandretti, A. P. Aguiar, and C. N. Jones, "On convergence and performance certification of a continuous-time economic model predictive control scheme with time-varying performance index," *Automatica*, vol. 68, pp. 305–313, 2016.

[90] V. M. Zavala and M. Anitescu, "Real-time nonlinear optimization as a generalized equation," *SIAM Journal on Control and Optimization*, vol. 48, no. 8, pp. 5444–5467, 2010.

[91] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," in *Nonlinear model predictive control*, pp. 345–369, Springer, 2009.

[92] S. Paternain, M. Morari, and A. Ribeiro, "A prediction-correction method for model predictive control," in *2018 Annual American Control Conference (ACC)*, pp. 4189–4194, IEEE, 2018.

[93] M. Diehl, R. Findeisen, F. Allgöwer, H. G. Bock, and J. P. Schlöder, "Nominal stability of real-time iteration scheme for nonlinear model predictive control," *IEEE Proceedings-Control Theory and Applications*, vol. 152, no. 3, pp. 296–308, 2005.

[94] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, "A class of prediction-correction methods for time-varying convex optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 17, pp. 4576–4591, 2016.

[95] Y. Tang, E. Dall'Anese, A. Bernstein, and S. Low, "Running primal-dual gradient method for time-varying nonconvex problems," *arXiv preprint arXiv:1812.00613*, 2018.

[96] Y. Tang, *Time-varying optimization and its application to power system operation*. PhD thesis, California Institute of Technology, 2019.

[97] Y. Tang and S. Low, "A second-order saddle point method for time-varying optimization," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 3928–3935, IEEE, 2019.

[98] Y. Li and N. Li, "Leveraging predictions in smoothed online convex optimization via gradient-based algorithms," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[99] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 7195–7201, IEEE, 2016.

[100] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87. Springer Science & Business Media, 2013.

[101] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan, "Online optimization: Competing with dynamic comparators," in *Artificial Intelligence and Statistics*, pp. 398–406, 2015.

[102] "Logistic regression," 2012.

[103] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 647–662, 2015.

[104] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation*. No. BOOK, Prentice Hall, 2000.

[105] J. D. Hamilton, *Time series analysis*, vol. 2. Princeton New Jersey, 1994.

[106] N. Lazic, C. Boutilier, T. Lu, E. Wong, B. Roy, M. Ryu, and G. Imwalle, "Data center cooling using model-predictive control," in *Advances in Neural Information Processing Systems*, pp. 3814–3823, 2018.

[107] W. Xu, X. Zhu, S. Singhal, and Z. Wang, "Predictive control for dynamic resource allocation in enterprise data centers," in *2006 IEEE/IFIP Network Operations and Management Symposium NOMS 2006*, pp. 115–126, IEEE, 2006.

[108] Y. Li, G. Qu, and N. Li, "Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit," *arXiv preprint arXiv:1801.07780*, 2018.

[109] B. Van Scoy, R. A. Freeman, and K. M. Lynch, "The fastest known globally convergent first-order method for minimizing strongly convex functions," *IEEE Control Systems Letters*, vol. 2, no. 1, pp. 49–54, 2017.

[110] D. Luenberger, "Canonical forms for linear multivariable systems," *IEEE Transactions on Automatic Control*, vol. 12, no. 3, pp. 290–293, 1967.

[111] R. Zhang, Y. Li, and N. Li, "On the regret analysis of online lqr control with predictions," *arXiv preprint arXiv:2102.01309*, 2021.

[112] G. Goel and B. Hassibi, "The power of linear controllers in lqr control," *arXiv preprint arXiv:2002.02574*, 2020.

[113] C. Yu, G. Shi, S.-J. Chung, Y. Yue, and A. Wierman, "The power of predictions in online control," 2020.

[114] Y. Lin, Y. Hu, H. Sun, G. Shi, G. Qu, and A. Wierman, "Perturbation-based regret analysis of predictive control in linear time varying systems," *arXiv preprint arXiv:2106.10497*, 2021.

[115] T. Li, R. Yang, G. Qu, G. Shi, C. Yu, A. Wierman, and S. Low, "Robustness and consistency in linear quadratic control with predictions," *arXiv preprint arXiv:2106.09659*, 2021.

[116] Y. Abbasi-Yadkori, P. Bartlett, and V. Kanade, "Tracking adversarial targets," in *International Conference on Machine Learning*, pp. 369–377, 2014.

[117] A. Cohen, A. Hasidim, T. Koren, N. Lazic, Y. Mansour, and K. Talwar, "Online linear quadratic control," in *International Conference on Machine Learning*, pp. 1028–1037, 2018.

[118] S. Tu and B. Recht, "Least-squares temporal difference learning for the linear quadratic regulator," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80 of *Proceedings of Machine Learning Research*, pp. 5005–5014, 2018.

[119] L. Lu, J. Tu, C.-K. Chau, M. Chen, and X. Lin, *Online energy generation scheduling for microgrids with intermittent energy sources and co-generation*, vol. 41. ACM, 2013.

[120] A. Borodin, N. Linial, and M. E. Saks, "An optimal on-line algorithm for metrical task system," *Journal of the ACM (JACM)*, vol. 39, no. 4, pp. 745–763, 1992.

[121] J. P. Hespanha, *Linear systems theory*. Princeton university press, 2018.

[122] S. Richter, C. N. Jones, and M. Morari, "Computational complexity certification for real-time mpc with input constraints based on the fast gradient method," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1391–1403, 2011.

[123] J. Rawlings and D. Mayne, "Postface to model predictive control: Theory and design," *Nob Hill Pub*, pp. 155–158, 2012.

[124] D. Angeli, R. Amrit, and J. B. Rawlings, "Receding horizon cost optimization for overly constrained nonlinear plants," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 7972–7977, IEEE, 2009.

[125] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[126] T. Damm, L. Grüne, M. Stieler, and K. Worthmann, "An exponential turnpike theorem for dissipative discrete time optimal control problems," *SIAM Journal on Control and Optimization*, vol. 52, no. 3, pp. 1935–1957, 2014.

[127] D. P. Bertsekas, *Dynamic programming and optimal control*, vol. 1. 2011.

[128] G. Klancar, D. Matko, and S. Blazic, "Mobile robot control on a reference path," in *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, pp. 1343–1348, IEEE, 2005.

[129] P. Corporation, *Pololu m3pi User's Guide*. Available at https://www.pololu.com/docs/pdf/0J48/m3pi.pdf.

[130] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, pp. 354–359, 2017.

[131] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[132] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[133] O. Chapelle and L. Li, "An empirical evaluation of thompson sampling," *Advances in neural information processing systems*, vol. 24, pp. 2249–2257, 2011.

[134] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[135] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2018.

[136] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, 2017.

[137] X. Chen, G. Qu, Y. Tang, S. Low, and N. Li, "Reinforcement learning for decision-making and control in power systems: Tutorial, review, and vision," *arXiv preprint arXiv:2102.01168*, 2021.

[138] X. Chen, Y. Nie, and N. Li, "Online residential demand response via contextual multi-armed bandits," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 433–438, 2020.

[139] "Uber self-driving car operator charged in pedestrian death." https://www.cnn.com/2020/09/18/cars/uber-vasquez-charged/index.html. Accessed: 2020-09-18.

[140] F. Oldewurtel, C. N. Jones, and M. Morari, "A tractable approximation of chance constrained stochastic mpc based on affine disturbance feedback," in *2008 47th IEEE conference on decision and control*, pp. 4731–4736, IEEE, 2008.

[141] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, 2016.

[142] J. Coulson, J. Lygeros, and F. Dörfler, "Distributionally robust chance constrained data-enabled predictive control," *arXiv preprint arXiv:2006.01702*, 2020.

[143] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[144] S. Dean, S. Tu, N. Matni, and B. Recht, "Safely learning to control the constrained linear quadratic regulator," in *2019 American Control Conference (ACC)*, pp. 5582–5588, IEEE, 2019.

[145] G. Schildbach, P. Goulart, and M. Morari, "Linear controller design for chance constrained systems," *Automatica*, vol. 51, pp. 278–284, 2015.

[146] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[147] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[148] G. Goel and B. Hassibi, "Regret-optimal control in dynamic environments," *arXiv preprint arXiv:2010.10473*, 2020.

[149] M. Nonhoff and M. A. Müller, "An online convex optimization algorithm for controlling linear systems with state and input constraints," *arXiv preprint arXiv:2005.11308*, 2020.

[150] M. Nonhoff and M. A. Müller, "Data-driven online convex optimization for control of dynamical systems," *arXiv preprint arXiv:2103.09127*, 2021.

[151] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the deepc," in *2019 18th European Control Conference (ECC)*, pp. 307–312, IEEE, 2019.

[152] Y. Zheng and N. Li, "Non-asymptotic identification of linear dynamical systems using multiple trajectories," *IEEE Control Systems Letters*, vol. 5, no. 5, pp. 1693–1698, 2020.

[153] 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*, pp. 559–570, PMLR, 2021.

[154] Y. Zheng, Y. Tang, and N. Li, "Analysis of the optimization landscape of linear quadratic gaussian (lqg) control," *arXiv preprint arXiv:2102.04393*, 2021.

[155] G. Qu, C. Yu, S. Low, and A. Wierman, "Combining model-based and model-free methods for nonlinear control: A provably convergent policy gradient approach," *arXiv preprint arXiv:2006.07476*, 2020.

[156] S. Paternain, M. Calvo-Fullana, L. F. Chamon, and A. Ribeiro, "Safe policies for reinforcement learning via primal-dual methods," *arXiv preprint arXiv:1911.09101*, 2019.

[157] Q. Yang, T. D. Simão, S. H. Tindemans, and M. T. Spaan, "Wcsac: Worst-case soft actor critic for safety-constrained reinforcement learning," in *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence. AAAI Press, online*, 2021.

[158] K. P. Wabersich and M. N. Zeilinger, "Performance and safety of bayesian model predictive control: Scalable model-based rl with guarantees," *arXiv preprint arXiv:2006.03483*, 2020.

[159] D. Muthirayan, J. Yuan, and P. P. Khargonekar, "Regret guarantees for online receding horizon control," *arXiv preprint arXiv:2010.07269*, 2020.

[160] M. Simchowitz and D. Foster, "Naive exploration is optimal for online lqr," in *International Conference on Machine Learning*, pp. 8937–8948, PMLR, 2020.

[161] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the sample complexity of the linear quadratic regulator," *Foundations of Computational Mathematics*, pp. 1–47, 2019.

[162] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A lyapunov-based approach to safe reinforcement learning," in *Advances in neural information processing systems*, pp. 8092–8101, 2018.

[163] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[164] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[165] O. Mihatsch and R. Neuneier, "Risk-sensitive reinforcement learning," *Machine learning*, vol. 49, no. 2, pp. 267–290, 2002.

[166] Z. Marvi and B. Kiumarsi, "Safe reinforcement learning: A control barrier function optimization approach," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 6, pp. 1923–1940, 2021.

[167] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3387–3395, 2019.

[168] N. Fulton and A. Platzer, "Safe reinforcement learning via formal methods," in *AAAI Conference on Artificial Intelligence*, 2018.

[169] Y. Li, G. Qu, and N. Li, "Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit," *IEEE Transactions on Automatic Control*, 2020.

[170] J. Yuan and A. Lamperski, "Online convex optimization for cumulative constraints," in *Advances in Neural Information Processing Systems*, pp. 6137–6146, 2018.

[171] X. Cao, J. Zhang, and H. V. Poor, "A virtual-queue-based algorithm for constrained online convex optimization with applications to data center resource allocation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 4, pp. 703–716, 2018.

[172] B. Kveton, J. Y. Yu, G. Theocharous, and S. Mannor, "Online learning with expert advice and finite-horizon constraints.," in *AAAI*, pp. 331–336, 2008.

[173] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization*, vol. 28. Princeton University Press, 2009.

[174] X. Wei, H. Yu, and M. J. Neely, "Online learning in weakly coupled markov decision processes: A convergence time study," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 1, pp. 1–38, 2018.

[175] D. Limon, I. Alvarado, T. Alamo, and E. Camacho, "Robust tube-based mpc for tracking of constrained linear systems with additive disturbances," *Journal of Process Control*, vol. 20, no. 3, pp. 248–260, 2010.

[176] E. Hazan, "Introduction to online convex optimization," *arXiv preprint arXiv:1909.05207*, 2019.

[177] M. Abeille and A. Lazaric, "Linear thompson sampling revisited," in *Artificial Intelligence and Statistics*, pp. 176–184, PMLR, 2017.

[178] D. Foster, T. Sarkar, and A. Rakhlin, "Learning nonlinear dynamical systems from a single trajectory," in *Learning for Dynamics and Control*, pp. 851–861, PMLR, 2020.

[179] Y. Li, J. Shamma, S. Das, and N. Li, "Regret and safety guarantees for adaptive linear quadratic control with constraints."

[180] M. Riedmiller, T. Gabel, R. Hafner, and S. Lange, "Reinforcement learning for robot soccer," *Autonomous Robots*, vol. 27, no. 1, pp. 55–73, 2009.

[181] Y.-C. Wang and J. M. Usher, "Application of reinforcement learning for agent-based production scheduling," *Engineering Applications of Artificial Intelligence*, vol. 18, no. 1, pp. 73–82, 2005.

[182] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics* (M. Hutter and R. Siegwart, eds.), vol. 5 of *Springer Proceedings in Advanced Robotics*, pp. 621–635, Springer International Publishing, 2018.

[183] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.

[184] L. Bakule, "Decentralized control: An overview," *Annual Reviews in Control*, vol. 32, no. 1, pp. 87–98, 2008.

[185] A. L. C. Bazzan, "Opportunities for multiagent systems and multiagent reinforcement learning in traffic control," *Autonomous Agents and Multi-Agent Systems*, vol. 18, no. 3, pp. 342–375, 2009.

[186] M. Pipattanasomporn, H. Feroze, and S. Rahman, "Multi-agent systems in a distributed smart grid: Design and implementation," in *2009 IEEE/PES Power Systems Conference and Exposition*, pp. 1–8, 2009.

[187] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Autonomous Robots*, vol. 4, no. 1, pp. 7–27, 1997.

[188] H. S. Witsenhausen, "A counterexample in stochastic optimum control," *SIAM Journal on Control*, vol. 6, no. 1, pp. 131–147, 1968.

[189] M. Rotkowitz and S. Lall, "A characterization of convex problems in decentralized control," *IEEE Transactions on Automatic Control*, vol. 50, no. 12, pp. 1984–1996, 2005.

[190] K. J. Åström and B. Wittenmark, *Adaptive Control*. Dover Publications, second ed., 2008.

[191] K. B. Ariyur and M. Krstic, *Real-Time Optimization by Extremum-Seeking Control*. John Wiley & Sons, 2003.

[192] D. Malik, A. Pananjady, K. Bhatia, K. Khamaru, P. L. Bartlett, and M. J. Wainwright, "Derivative-free methods for policy optimization: Guarantees for linear quadratic systems," *Journal of Machine Learning Research*, vol. 21, no. 21, pp. 1–51, 2020.

[193] Z. Yang, Y. Chen, M. Hong, and Z. Wang, "Provably global convergence of actor-critic: A case for linear quadratic regulator with ergodic cost," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 8351–8363, Curran Associates, Inc., 2019.

[194] S. Oymak and N. Ozay, "Non-asymptotic identification of LTI systems from a single trajectory," in *2019 American Control Conference (ACC)*, pp. 5655–5661, 2019.

[195] J. Bu, A. Mesbahi, M. Fazel, and M. Mesbahi, "LQR through the lens of first order methods: Discrete-time case," *arXiv preprint arXiv:1907.08921*, 2019.

[196] M. I. Abouheaf, F. L. Lewis, K. G. Vamvoudakis, S. Haesaert, and R. Babuska, "Multi-agent discrete-time graphical games and reinforcement learning solutions," *Automatica*, vol. 50, no. 12, pp. 3038–3053, 2014.

[197] H. Zhang, H. Jiang, Y. Luo, and G. Xiao, "Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement

learning method," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4091–4100, 2016.

[198] K. Zhang, E. Miehling, and T. Başar, "Online planning for decentralized stochastic control with partial history sharing," in *2019 American Control Conference (ACC)*, pp. 3544–3550, IEEE, 2019.

[199] M. Gagrani and A. Nayyar, "Thompson sampling for some decentralized control problems," in *Proceedings of the 57th IEEE Conference on Decision and Control (CDC)*, pp. 1053–1058, 2018.

[200] H. Feng and J. Lavaei, "On the exponential number of connected components for the feasible set of optimal decentralized control problems," in *2019 American Control Conference*, pp. 1430–1437, 2019.

[201] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.

[202] K. Mårtensson and A. Rantzer, "Gradient methods for iterative distributed control synthesis," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 549–554, IEEE, 2009.

[203] A. Al Alam, A. Gattami, and K. H. Johansson, "Suboptimal decentralized controller design for chain structures: Applications to vehicle formations," in *Proceedings of the 50th IEEE Conference on Decision and Control (CDC) and European Control Conference*, pp. 6894–6900, 2011.

[204] L. Furieri, Y. Zheng, and M. Kamgarpour, "Learning the globally optimal distributed lq regulator," in *Learning for Dynamics and Control*, pp. 287–297, PMLR, 2020.

[205] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," *Mathematics of Operations Research*, vol. 27, no. 4, pp. 819–840, 2002.

[206] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian, "Deep decentralized multi-task multi-agent reinforcement learning under partial observability," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, pp. 2681–2690, 2017.

[207] L. Peshkin, K.-E. Kim, N. Meuleau, and L. P. Kaelbling, "Learning to cooperate via policy search," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 489–496, 2000.

[208] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *The Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 2974–2982, 2018.

[209] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80 of *Proceedings of Machine Learning Research*, pp. 5872–5881, 2018.

[210] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[211] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems*, vol. 12, pp. 1057–1063, MIT Press, 2000.

[212] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32 of *Proceedings of Machine Learning Research*, pp. 387–395, 2014.

[213] Y. Tang, Y. Zheng, and N. Li, "Analysis of the optimization landscape of linear quadratic gaussian (lqg) control," in *Learning for Dynamics and Control*, pp. 599–610, PMLR, 2021.

[214] A. D. Flaxman, A. T. Kalai, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 385–394, 2005.

[215] O. Shamir, "On the complexity of bandit and derivative-free stochastic convex optimization," in *Proceedings of the 26th Annual Conference on Learning Theory*, vol. 30 of *Proceedings of Machine Learning Research*, pp. 3–24, 2013.

[216] X. Chen, J. I. Poveda, and N. Li, "Model-free optimal voltage control via continuous-time zeroth-order methods," *arXiv preprint arXiv:2103.14703*, 2021.

[217] Y. Tang, J. Zhang, and N. Li, "Distributed zero-order algorithms for nonconvex multiagent optimization," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 269–281, 2020.

[218] Y. Tang, Z. Ren, and N. Li, "Zeroth-order feedback optimization for cooperative multi-agent systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3649–3656, IEEE, 2020.

[219] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[220] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1245–1260, 2017.

[221] S. J. Reddi, A. Hefny, S. Sra, B. Póczós, and A. Smola, "Stochastic variance reduction for nonconvex optimization," in *Proceedings of the 33rd International Conference on Machine Learning*, vol. 48 of *Proceedings of Machine Learning Research*, pp. 314–323, 2016.

[222] M. Rudelson, R. Vershynin, *et al.*, "Hanson-wright inequality and sub-gaussian concentration," *Electronic Communications in Probability*, vol. 18, 2013.

[223] P. Concus, G. H. Golub, and G. Meurant, "Block preconditioning for the conjugate gradient method," *SIAM Journal on Scientific and Statistical Computing*, vol. 6, no. 1, pp. 220–252, 1985.

[224] P. M. Gahinet, A. J. Laub, C. S. Kenney, and G. A. Hewer, "Sensitivity of the stable discrete-time Lyapunov equation," *IEEE Transactions on Automatic Control*, vol. 35, no. 11, pp. 1209–1217, 1990.

[225] G. A. F. Seber and A. J. Lee, *Linear Regression Analysis*. John Wiley & Sons, second ed., 2003.

[226] D. Hsu, S. Kakade, and T. Zhang, "A tail inequality for quadratic forms of subgaussian random vectors," *Electronic Communications in Probability*, vol. 17, no. 52, pp. 1–6, 2012.