# The sample complexity of simple reinforcement learning

*Horia Mania*

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2020-150
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2020/EECS-2020-150.html

August 13, 2020

# The sample complexity of simple reinforcement learning

by

Horia S Mania

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Michael I. Jordan, Co-chair
Professor Benjamin Recht, Co-chair
Professor Francesco Borrelli

Summer 2020

The sample complexity of simple reinforcement learning

Abstract

The sample complexity of simple reinforcement learning

by

Horia S Mania

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Michael I. Jordan, Co-chair

Professor Benjamin Recht, Co-chair

With potential applications as diverse as self-driving cars, medical robots, and network protocols, recent years witnessed a staggering interest in building autonomous agents that learn to interact with the world. Despite impressive successes in games and robotic demonstrations, known reinforcement learning (RL) algorithms remain data hungry, unreliable, and complex. To address these issues we need to better understand the limits of training agents that interact with the world.

We theoretically analyze the data requirements of RL in several simple settings. To understand the sample complexity of system identification, a fundamental building block of model-based RL and feedback control, we focus on the estimation of linear dynamical systems and, more generally, on the estimation of dynamical systems whose state transitions depend linearly on a known feature embedding of state-action pairs. For linear dynamical systems we present a specialized analysis that captures the correct signal-to-noise behavior of the problem, showing that more unstable linear systems are easier to estimate. While linear systems can be identified from data generated by i.i.d. random inputs, to estimate nonlinear dynamical systems we must use a judicious choice of inputs. We propose an active learning method that addresses this challenge.

Then, we study the Linear Quadratic Regulator (LQR), a classical problem in control theory, from a RL perspective by assuming the underlying dynamics are unknown. We consider two solutions that use estimates of the dynamics to synthesize controllers: certainty equivalence and robust LQR. Certainty equivalence is the most straightforward approach to controller synthesis for LQR with unknown dynamics. It generates the optimal controller for the estimated dynamics, disregarding the effects of the estimation error. We show that when the estimation error is sufficiently small the difference between the cost achieved by the certainty equivalent controller and the optimal cost scales like the square of the estimation error.

We also consider a robust LQR approach that can operate with larger estimation errors. Our robust LQR method relies on System Level Synthesis to formulate the robust control problem as a quasi-convex optimization problem. We show that the performance gap of robust LQR scales linearly with the estimation error. Therefore, certainty equivalence can outperform robust LQR when the estimation error is small, but the latter approach can operate with larger estimation error.

Finally, in many settings RL agents have to operate in the presence of other decision makers. To study RL in such a scenario we take inspiration from the study of two-sided markets and stable matchings. Agents acting in markets often have to learn about their preferences through exploration. With the advent of massive online markets powered by data-driven matching platforms, it has become necessary to better understand the interplay between learning and market objectives. We propose a statistical learning model in which one side of the market does not have a priori knowledge about its preferences for the other side and is required to learn these from stochastic rewards. Our model extends the standard multi-armed bandits framework to multiple players, with the added feature that arms have preferences over players. We show surprising exploration-exploitation trade-offs compared to the single player multi-armed bandits setting.

To my family

# Contents

## Acknowledgments

This work would not have been possible without the help of many exquisite people and the welcoming UC Berkeley community. First of all, I would like to thank my advisors Michael I. Jordan and Benjamin Recht, who gave me the opportunity to be part of a vibrant research community and generously shared their knowledge and perspective on machine learning. Their mentorship and encouragement have been invaluable.

I would also like to thank my committee members Francesco Borrelli and Martin Wainwright who gave me valuable feedback and from whom I learned a lot during countless group meetings and conversations. I have also been fortunate to have many research conversation and meetings with other Berkeley faculty: Anca Dragan, Mortiz Hardt, Andrew Packard, Shankar Sastry, Koushil Sreenath, Ion Stoica, Clarie Tomlin.

The work presented in this thesis has been completed by working together with many excellent collaborators, from whom I learned a lot: Sarah Dean, Aurelia Guy, Lydia Liu, Nikolai Matni, Max Simchowitz, Stephen Tu. I also had the pleasure to work with and learn from Alex Feng, Moritz Hardt, John Miller, Xinghao Pan, Dimitris Papailiopoulos, Kannan Ramchandran, Aaditya Ramdas, Rebecca Roelofs, Ludwig Schmidt, Vaishaal Shankar, and Martin Wainwright on other research projects. I would like to thank Dimitris Papailiopoulos and Aaditya Ramdas who introduced me to the first two research projects I completed at Berkeley and who encouraged me every step of the way.

My time at Berkeley has been enriched by an amazing community. I was lucky to share an apartment, stimulating conversations, and fun hobbies with Robert Nishihara, Max Rabinovich, Ludwig Schmidt, and Mitchell Stern. Also, more broadly, I benefited tremendously, both professionally and personally, from the many interactions with the members of ModestYachts, SAIL, AMP lab, RISE lab, and BAIR. I am grateful for the many friends and colleagues I met at Berkeley: Ahmed El Alaoui, Yasaman Bahri, Ross Boczar, Nicholas Boyd, Xiang Cheng, Mihaela Curmei, Orianna DeMasi, Jelena Diakonikolas, Nicolas Flammarion, Sara Fridovich-Keil, Wenshuo Guo, Kevin Jamieson, Chi Jin, Eric Jonas, Koulik Khamaru, Marc Khoury, Karl Krauth, Lihua Lei, Laurent Lessard, Romain Lopez, Jeff Mahler, Eric Mazumdar, Philipp Moritz, Alysa Morrow, Michael Muelebach, Samet Oymak, Aldo Pacchiano, Juan Carlos Perdomo, Esther Rolf, Feng Ruan, Judy Savitskaya, Geoffery Schiebinger, Adam Sealfon, Jake Soloff, Mahdi Soltanolkotabi, Evan Sparks, Nilesh Tripuraneni, Shivaram Venkataraman, Andre Wibisono, Ahia Wilson, Vickie Ye, Tijana Zrnic, and many others.

Outside of Berkeley, I would like to thank Sébastien Bubeck, who introduced me to machine learning research and offered invaluable guidance during my undergraduate years. I would also like to thank Francis Bach, Indejit Dhillon, Daniel Hill, Aria Mazumdar, and Sujay Sanghavi for their mentorship and for exposing me to new research. Also, I am grateful for my long time friend Andrei Parvu.

While at Berkeley, I was lucky to meet my amazing partner, Clarke Knight, who brightened the past several years. Finally, I would like to thank my wonderful parents, Oltita and Marius, and my sister, Sensy, for their love and invaluable support.

# Chapter 1

# Introduction

Reinforcement learning (RL) aims design autonomous agents that learn to interact with the world in order to achieve desired goals. With potential areas of application as diverse as autonomous vehicles, recommender systems, medical devices, and robotics, RL can be a boon to society. It has already produced agents that surpass human players in games [100, 139, 159] and has also led to exciting robotics demos [5, 84, 86, 133]. Although these results are impressive, there are several factors prohibiting the wide adoption of RL methods for controlling physical systems. RL methods require too much data to achieve reasonable performance and many algorithms are difficult to implement, evaluate, and deploy [66].

In the quest to find methods that are *sample efficient* (i.e. methods that need little data) the general trend in RL has been to develop increasingly complicated methods and compare these methods through empirical evaluation on games and other simulated tasks [57, 58, 65, 86, 101, 103, 113, 115, 125, 132, 134, 135, 138, 162, 166]. Unfortunately, this research trend has led to a reproducibility crisis. Recent studies demonstrate that many RL methods are not robust to changes in hyperparameters, random seeds, or even different implementations of the same algorithm [66, 69]. Such unreliability precludes the integration of RL algorithms into mission critical control systems.

We illustrate the drawbacks of comparing RL algorithms solely through empirical evaluations on simulated tasks and argue that the theoretical analysis of simple RL methods and problems offers a viable path to illuminating fundamental concepts. In Chapter 2 we propose a simple baseline that is competitive with popular RL methods on standard continuous control benchmarks, the MuJoCo locomotion tasks [26, 153]. We use the evaluation of our simple method to illustrate several limitations of common evaluation practices in RL.

Theoretical analysis is a natural complement to empirical evaluation and can be used to alleviate some of the issues of common evaluation practices. In general, a RL problem is defined by a dynamical system with *state* $\mathbf{x}_t \in \mathbb{R}^n$ that can be acted on by a *control* $\mathbf{u}_t \in \mathbb{R}^p$ and obeys the stochastic dynamics

$$\mathbf{x}_{t+1} = f_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t),$$

where $\mathbf{w}_t$ is a disturbance that can be stochastic or adversarial. Often, the full state $\mathbf{x}_t$

cannot be observed directly, but information about it can be inferred from observations $\mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{u}_t, \nu_t)$, where $h_t$ is an observation model and $\nu_t$ is a noise process. Then, reinforcement learning and optimal control seek to find inputs $\mathbf{u}_t$ that

$$\begin{array}{ll} \text{minimize} & \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(\mathbf{y}_t, \mathbf{u}_t)\right] \\ \text{subject to} & \mathbf{x}_{t+1} = f_t(\mathbf{y}_t, \mathbf{u}_t, \mathbf{w}_t), \\ & \mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{u}_t, \nu_t). \end{array} \tag{1.0.1}$$

Here, $c_t$ denotes the state-control cost at every time step (more generally, the costs can be stochastic or adversarial). The input $\mathbf{u}_t$ is allowed to depend on the current observation $\mathbf{y}_t$ and all previous observations and actions. In this generality, problem (1.0.1) encapsulates many of the problems considered in the RL literature.

A solution to problem (1.0.1) is expressed in terms of a map $\pi : \{\mathbf{u}_j, \mathbf{y}_{j+1}\}_{j=0}^{t-1} \to \mathbf{u}_t$, called a *policy* or *controller*. Then, the goal of RL is to find a policy that achieves small cost, a difficult problem even when the dynamics model $f_t$, the observation model $h_t$, and the cost function $c_t$ are known. RL, nevertheless, aims to solve (1.0.1) when one or more of these elements are unknown. To deal with the unknown components, in RL we assume that it is possible to interact with the dynamical system and collect data, i.e. take actions $\mathbf{u}_t$ and observe the response of the system $\mathbf{y}_{t+1}$. Through repeated interactions with the system, RL methods aim to either directly learn a good policy $\pi$ or to learn estimates of the unknown components that can be used for controller synthesis.

In this thesis we aim to theoretically quantify the amount of data needed to find a policy that achieves close to optimal performance. To achieve this goal we must consider a simpler problem than (1.0.1) since (1.0.1) is too general. The simplest optimal control problem with continuous state is the Linear Quadratic Regulator (LQR), in which we fully observe the states (i.e., $\mathbf{y}_t = \mathbf{x}_t$), the costs are a fixed quadratic function of state and control, and the dynamics are linear and time-invariant:

$$\begin{array}{ll} \text{minimize} & \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t^\top Q \mathbf{x}_t + \mathbf{u}_{t-1}^\top R \mathbf{u}_{t-1}\right] \\ \text{subject to} & \mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{w}_t \end{array} . \tag{1.0.2}$$

Here $Q$ (resp. $R$) is a $n \times n$ (resp. $p \times p$) positive definite matrix, $A$ and $B$ are called the *state transition matrices*, and $\mathbf{w}_t \in \mathbb{R}^n$ is Gaussian noise with zero-mean and covariance $\Sigma_w$. While it may seem that the modeling assumptions underlying (1.0.2) are too restrictive, LQR is the basis for many successful nonlinear control methods [27, 85, 151].

We are concerned with the *infinite time horizon* variant of the LQR problem (1.0.2) where we let the time horizon $T$ go to infinity and minimize the average cost. When the dynamics are known, this problem has a celebrated closed form controller based on the solution of matrix Riccati equations [169]. Indeed, the optimal policy sets $u_t = Kx_t$ for a fixed $p \times n$ matrix $K$, and the corresponding optimal cost is the gold-standard to which we compare the cost of our algorithms.

LQR has been studied for decades and consequently is well understood: it has a simple, closed form solution on the infinite time horizon and an efficient, dynamic programming

solution on finite time horizons. However, prior to the work presented in this thesis, little was known about LQR when the transition parameters $A$ and $B$ are unknown. In Chapter 3 we start by determining the amount of data required to estimate $A$ and $B$.

The problem of estimating dynamical systems from data is known as system identification, a fundamental building block of model-based RL and feedback control. We focus on the identification of linear dynamical systems and, more generally, on the identification of dynamical systems whose state transitions depend linearly on a known feature embedding of state-action pairs. For linear dynamical systems we present a specialized analysis that captures the correct signal-to-noise behavior of the problem, showing that more unstable linear systems are easier to estimate. While linear systems can be identified from data generated by i.i.d. random inputs, to estimate nonlinear dynamical systems we must use a judicious choice of inputs.

Then, in Chapter 4, we analyze the performance of two methods that use estimates of $A$ and $B$ to synthesize controllers. One of the most straightforward methods for controlling a dynamical system with unknown transitions is based on the *certainty equivalence principle*: a model of the system is fit by observing its time evolution, and then a controller is designed by treating the fitted model as the truth [15]. Despite the simplicity of this method, it is challenging to guarantee its efficiency because small modeling errors may propagate to large, undesirable behaviors on long time horizons. We show that this cannot happen when the estimation error is sufficiently small. Concretely, we show that when the estimation error is small the gap between the performance of the certainty equivalent controller and the optimal controller scales quadratically with the estimation error.

While this is a strong guarantee, certainty equivalence can fail when the estimation error is moderately large [41]. For this reason, many methods for controlling systems with unknown dynamics explicitly incorporate robustness against model uncertainty [41, 42, 70, 107, 167, 169]. We also propose a robust control approach that couples the uncertainty in estimation with the control design. Namely, our method uses an uncertainty set of the transition parameters to find a controller that performs well on the worst case dynamics in the uncertainty set. While this worst case optimization problem cannot be solved exactly, we offer a quasi-convex relaxation that achieves a performance gap that scales linearly with the estimation error. This guarantee distinguishes our method from prior robust methods for LQR.

Finally, in Chapter 5 we consider a problem that cannot be fully captured by (1.0.1) since (1.0.1) assumes the existence of a single decision maker. In real-world settings, individual decisions must be made in the context of actions taken by other decision makers. Moreover, such decisions often involve scarcity, with competition among multiple decision-makers. To study such settings we need to blend economics with learning.

We take inspiration from the study of two-sided markets, an important area of study in economics. Agents acting in markets often have to learn about their preferences through exploration. With the advent of massive online markets powered by data-driven matching platforms, it has become necessary to better understand the interplay between learning and market objectives. We propose a statistical learning model in which one side of the

market does not have a priori knowledge about its preferences for the other side and is required to learn these from stochastic rewards. Our model extends the standard multi-armed bandits framework to multiple players, with the added feature that arms have preferences over players. We show surprising exploration-exploitation trade-offs compared to the single player multi-armed bandits setting.

# Chapter 2

# Simple random search

In this chapter, we aim to determine *the simplest* model-free RL method that can solve standard benchmarks. Two different directions have been proposed for simplifying RL. Salimans et al. [125] introduced a derivative-free policy optimization method, called Evolution Strategies. The authors showed that, for several RL tasks, their method can easily be parallelized to train policies faster than other methods. While the method of Salimans et al. [125] is simpler than previously proposed methods, it employs several complicated algorithmic elements, which we discuss at the end of Section 2.1. As a second simplification to model-free RL, Rajeswaran et al. [115] have shown that linear policies can be trained via natural policy gradients to obtain competitive performance on the MuJoCo locomotion tasks, showing that complicated neural network policies are not needed to solve these continuous control problems. In this work, we combine ideas from the work of Salimans et al. [125] and Rajeswaran et al. [115] to obtain the simplest model-free RL method yet, a derivative-free optimization algorithm for training static, linear policies. We demonstrate that a simple random search method can match or exceed state-of-the-art sample efficiency on the MuJoCo locomotion tasks, included in the OpenAI Gym.

Henderson et al. [66] and Islam et al. [69] pointed out that standard evaluation methodology does not accurately capture the performance of RL methods by showing that existing RL algorithms exhibit high sensitivity to both the choice of random seed and the choice of hyperparameters. We show similar limitations of common evaluation methodology through a different lens. We exhibit a simple derivative free optimization algorithm which matches or surpasses the performance of more complex methods when using the same evaluation methodology. However, a more thorough evaluation of ARS reveals worse performance. Moreover, our method uses static linear policies and a simple local exploration scheme, which might be limiting for more difficult RL tasks. Therefore, better evaluation schemes are needed for determining the benefits of more complex RL methods. Our contributions are as follows:

- In Section 2.1, for applications to continuous control, we augment a basic random search method with three simple features. First, we scale each update step by the standard deviation of the rewards collected for computing that update step. Second, we

normalize the system's states by online estimates of their mean and standard deviation. Third, we discard from the computation of the update steps the directions that yield the least improvement of the reward. We refer to this method as *Augmented Random Search* (ARS).

- In Section 2.2, we evaluate the performance of ARS on the benchmark MuJoCo locomotion tasks, included in the OpenAI Gym. Our method learns static, linear policies that achieve high rewards on all MuJoCo tasks. No neural networks are used, and yet state-of-the-art average rewards are achieved. For example, for Humanoid-v1 ARS finds linear policies which achieve average rewards of over 11500, the highest value reported in the literature. To put ARS on equal footing with competing methods, we evaluate its sample complexity over three random seeds and compare it to results reported in the literature [58, 115, 125, 135]. ARS matches or exceeds state-of-the-art sample efficiency on the locomotion tasks when using standard evaluation methodology.

- For a more thorough evaluation, we measured the performance of ARS over a hundred random seeds and also evaluated its sensitivity to hyperparameter choices. Though ARS successfully trains policies for the MuJoCo tasks a large fraction of the time when hyperparameters and random seeds are varied, ARS exhibits large variance. We measure the frequency with which ARS finds policies that yield suboptimal locomotion gaits.

The material presented in this chapter is based on the work by Mania et al. [94].

**Problem setup.** Problems in reinforcement learning require finding policies for controlling dynamical systems that maximize an average reward. Such problems can be abstractly formulated as

$$\max_{\theta \in \mathbb{R}^d} \mathbb{E}_\xi \left[ r(\pi_\theta, \xi) \right] , \tag{2.0.1}$$

where $\theta$ parametrizes a policy $\pi_\theta \colon \mathbb{R}^n \to \mathbb{R}^p$. The random variable $\xi$ encodes the randomness of the environment, i.e., random initial states and stochastic transitions. The value $r(\pi_\theta, \xi)$ is the reward achieved by the policy $\pi_\theta$ on one trajectory generated from the system. In general one could use stochastic policies $\pi_\theta$, but our proposed method uses deterministic policies.

**Basic random search.** Note that the problem formulation (2.0.1) aims to optimize reward by directly optimizing over the policy parameters $\theta$. We consider methods which explore in the parameter space rather than the action space. This choice renders RL training equivalent to derivative-free optimization with noisy function evaluations. One of the simplest and oldest optimization methods for derivative-free optimization is *random search* [99].

A primitive form of random search, which we call *basic random search* (BRS), simply computes a finite difference approximation along the random direction and then takes a step

along this direction without using a line search. Our method ARS, described in Section 2.1, is based on this simple strategy. For updating the parameters $\theta$ of a policy $\pi_\theta$, BRS and ARS exploit update directions of the form:

$$\frac{r(\pi_{\theta+\nu\delta}, \xi_1) - r(\pi_{\theta-\nu\delta}, \xi_2)}{\nu}, \tag{2.0.2}$$

for two i.i.d. random variables $\xi_1$ and $\xi_2$, $\nu$ a positive real number, and $\delta$ a zero mean Gaussian vector. It is known that such an update increment is an unbiased estimator of the gradient with respect to $\theta$ of $\mathbb{E}_\delta \mathbb{E}_\xi [r(\pi_{\theta+\nu\delta}, \xi)]$, a smoothed version of the objective (2.0.1) which is close to the original objective when $\nu$ is small [105]. When the function evaluations are noisy, minibatches can be used to reduce the variance in this gradient estimate. Evolution Strategies is a version of this algorithm with several complicated algorithmic enhancements [125]. Another version of this algorithm is called Bandit Gradient Descent by Flaxman et al. [52]. The convergence of random search methods for derivative free optimization has been understood for several types of convex optimization [9, 18, 71, 105]. Jamieson et al. [71] offer an information theoretic lower bound for derivative free convex optimization and show that a coordinate based random search method achieves the lower bound with nearly optimal dependence on the dimension.

The rewards $r(\pi_{\theta+\nu\delta}, \xi_1)$ and $r(\pi_{\theta-\nu\delta}, \xi_2)$ in Eq. (2.0.2) are obtained by collecting two trajectories from the dynamical system of interest, according to the policies $\pi_{\theta+\nu\delta}$ and $\pi_{\theta-\nu\delta}$, respectively. The random variables $\xi_1$, $\xi_2$, and $\delta$ are mutually independent, and independent from previous trajectories. One trajectory is called an *episode* or a *rollout*. The goal of RL algorithms is to approximately solve problem (2.0.1) by using as few rollouts from the dynamical system as possible.

## 2.1 Proposed algorithm

We now introduce the Augmented Random Search (ARS) method, which relies on three augmentations of BRS that build on successful heuristics employed in deep reinforcement learning. Throughout the rest of the paper we use $M$ to denote the parameters of policies because our method uses linear policies, and hence $M$ is a $p \times n$ matrix. The different versions of ARS are detailed in Algorithm 1.

The first version, ARS **V1**, is obtained from BRS by scaling the update steps by the standard deviation $\sigma_R$ of the rewards collected at each iteration; see Line 7 of Algorithm 1. As shown in Section 2.2, ARS **V1** can train linear policies, which achieve the reward thresholds previously proposed in the literature, for five MuJoCo benchmarks. However, ARS **V1** requires a larger number of episodes, and it cannot train policies for the Humanoid-v1 task. To address these issues in Algorithm 1 we also propose ARS **V2**. This version of ARS trains policies which are linear maps of states normalized by a mean and standard deviation computed online. Finally, to further enhance the performance of ARS, we introduce a third algorithmic enhancement, shown in Algorithm 1 as ARS **V1-t** and ARS **V2-t**. These

versions of ARS can drop perturbation directions that yield the least improvement of the reward. Now, we motivate and offer intuition for each of these algorithmic elements.

**Algorithm 1** Augmented Random Search (ARS): four versions **V1**, **V1-t**, **V2** and **V2-t**

1: **Hyperparameters:** step-size $\alpha$, number of directions sampled per iteration $N$, standard deviation of the exploration noise $\nu$, number of top-performing directions to use $b$ ($b < N$ is allowed only for **V1-t** and **V2-t**)
2: **Initialize:** $M_0 = \mathbf{0} \in \mathbb{R}^{p\times n}$, $\mu_0 = \mathbf{0} \in \mathbb{R}^n$, and $\Sigma_0 = \mathbf{I}_n \in \mathbb{R}^{n\times n}$, $j = 0$.
3: **while** ending condition not satisfied **do**
4: Sample $\delta_1, \delta_2, \ldots, \delta_N$ in $\mathbb{R}^{p\times n}$ with i.i.d. standard normal entries.
5: Collect $2N$ rollouts of horizon $H$ and their corresponding rewards using the $2N$ policies

$$\textbf{V1:} \begin{cases} \pi_{j,k,+}(x) = (M_j + \nu\delta_k)x \\ \pi_{j,k,-}(x) = (M_j - \nu\delta_k)x \end{cases}$$

$$\textbf{V2:} \begin{cases} \pi_{j,k,+}(x) = (M_j + \nu\delta_k)\,\mathrm{diag}\,(\Sigma_j)^{-\frac{1}{2}}\,(x - \mu_j) \\ \pi_{j,k,-}(x) = (M_j - \nu\delta_k)\,\mathrm{diag}(\Sigma_j)^{-\frac{1}{2}}(x - \mu_j) \end{cases}$$

for $k \in \{1, 2, \ldots, N\}$.
6: **V1-t, V2-t:** Sort the directions $\delta_k$ by $\max\{r(\pi_{j,k,+}), r(\pi_{j,k,-})\}$, denote by $\delta_{(k)}$ the $k$-th largest direction, and by $\pi_{j,(k),+}$ and $\pi_{j,(k),-}$ the corresponding policies.
7: Make the update step:

$$M_{j+1} = M_j + \tfrac{\alpha}{b\sigma_R} \sum_{k=1}^{b} \left[r(\pi_{j,(k),+}) - r(\pi_{j,(k),-})\right] \delta_{(k)},$$

where $\sigma_R$ is the standard deviation of the $2b$ rewards used in the update step.
8: **V2:** Set $\mu_{j+1}$, $\Sigma_{j+1}$ to be the mean and covariance of the $2NH(j+1)$ states encountered from the start of training.[1]
9: $j \leftarrow j + 1$
10: **end while**

.

**Scaling by the standard deviation $\sigma_R$.** As the training of policies progresses, random search in the parameter space of policies can lead to large variations in the rewards observed across iterations. As a result, it is difficult to choose a fixed step-size $\alpha$ which does not allow harmful variations in the size of the update steps. Salimans et al. [125] address this issue by transforming the rewards into rankings and then using the adaptive optimization algorithm Adam for computing the update step. Both of these techniques change the direction of the

updates, obfuscating the behavior of the algorithm and making it difficult to ascertain the objective Evolution Strategies is actually optimizing. Instead, to address the large variations of the differences $r(\pi_{M+\nu\delta}) - r(\pi_{M-\nu\delta})$, we scale the update steps by the standard deviation $\sigma_R$ of the $2N$ rewards collected at each iteration (see Line 7 of Algorithm 1).

While training a policy for Humanoid-v1, we observed that the standard deviations $\sigma_R$ have an increasing trend; see Figure 2.1. This behavior occurs because perturbations of the policy weights at high rewards can cause Humanoid-v1 to fall early, yielding large variations in the rewards collected. Without scaling the update steps by $\sigma_R$, eventually random search would take update steps which are a thousand times larger than in the beginning of training. Therefore, $\sigma_R$ adapts the step sizes according to the local sensitivity of the rewards to perturbations of the policy parameters. The same training performance could probably be obtained by tuning a step size schedule. However, one of our goals was to minimize the amount of tuning required.

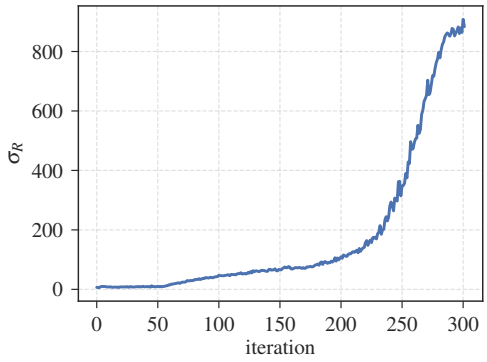


**Figure 2.1:** Showing the standard deviation $\sigma_R$ of the rewards collected at each iteration, while training Humanoid-v1.

**Normalization of the states.** The normalization of states used by ARS **V2** is akin to data whitening for regression tasks. Intuitively, it ensures that policies put equal weight on the different components of the states. To see why this might help, suppose that a state coordinate only takes values in the range $[90, 100]$ while another state component takes values in the range $[-1, 1]$. Then, small changes in the control gain with respect to the first state coordinate would lead to larger changes in the actions than the same sized changes with respect to the second state component. Hence, state normalization allows different state components to have equal influence during training.

Previous work has also implemented such state normalization for fitting a neural network model for several MuJoCo environments [103]. A similar normalization is used by ES as part

[1]Of course, we implement this in an efficient way that does not require the storage of all the states. Also, we only keep track of the diagonal of $\Sigma_{j+1}$. Finally, to ensure that the ratio $0/0$ is treated as 0, if a diagonal entry of $\Sigma_j$ is smaller than $10^{-8}$ we make it equal to $+\infty$.

of the virtual batch normalization of the neural network policies [125]. In the case of ARS, the state normalization can be seen as a form of non-isotropic exploration in the parameter space of linear policies.

The main empirical motivation for ARS **V2** comes from the Humanoid-v1 task. We were not able to train a linear policy for this task without the normalization of the states described in Algorithm 1. Moreover, ARS **V2** performs better than ARS **V1** on other MuJoCo tasks as well, as shown in Section 2.2. However, the usefulness of state normalization is likely to be problem specific.

**Using top performing directions.** To further improve the performance of ARS on the MuJoCo locomotion tasks, we propose ARS **V1-t** and **V2-t**. In the update steps used by ARS **V1** and **V2** each perturbation direction $\delta_k$ is weighted by the difference of the rewards $r(\pi_{j,k,+})$ and $r(\pi_{j,k,-})$. If $r(\pi_{j,k,+}) > r(\pi_{j,k,-})$, ARS pushes the policy weights $M_j$ in the direction of $\delta_k$. If $r(\pi_{j,k,+}) < r(\pi_{j,k,-})$, ARS pushes the policy weights $M_j$ in the direction of $-\delta_k$. However, since $r(\pi_{j,k,+})$ and $r(\pi_{j,k,-})$ are noisy evaluations of the performance of the policies parametrized by $M_j + \nu\delta_k$ and $M_j - \nu\delta_k$, ARS **V1** and **V2** might push the weights $M_j$ in the direction $\delta_k$ even when $-\delta_k$ is better, or vice versa. Moreover, there can be perturbation directions $\delta_k$ such that updating the policy weights $M_j$ in either the direction $\delta_k$ or $-\delta_k$ would lead to sub-optimal performance. To address these issues, ARS **V1-t** and **V2-t** order decreasingly the perturbation directions $\delta_k$, according to $\max\{r(\pi_{j,k,+}), r(\pi_{j,k,-})\}$, and then use only the top $b$ directions for updating the policy weights; see Line 7 of Algorithm 1.

This algorithmic enhancement intuitively improves the performance of ARS because it ensures that the update steps are an average over directions that obtained high rewards. However, without theoretical investigation we cannot be certain of the effect of using this algorithmic enhancement, i.e., choosing $b < N$. When $b = N$ versions **V1-t** and **V2-t** are equivalent to **V1** and **V2**. Therefore, it is certain that after tuning ARS **V1-t** and **V2-t**, they will not perform any worse than ARS **V1** and **V2**.

**Comparison to Salimans et al. [125].** ARS simplifies Evolution Strategies in several ways. First, ES feeds the gradient estimate into the Adam algorithm. Second, instead of using the actual reward values $r(\theta \pm \sigma\epsilon_i)$, ES transforms the rewards into rankings and uses the ranks to compute update steps. The rankings are used to make training more robust. Instead, our method scales the update steps by the standard deviation of the rewards. Third, ES bins the action space of the Swimmer-v1 and Hopper-v1 to encourage exploration. Our method surpasses ES without such binning. Fourth, ES relies on policies parametrized by neural networks with virtual batch normalization, while we show that ARS achieves state-of-the-art performance with linear policies.

## 2.2 Empirical evaluation

**Implementation details.** We implemented a parallel version of Algorithm 1 using the Python library Ray [102]. To avoid the computational bottleneck of communicating perturbations $\delta$, we created a shared noise table which stores independent standard normal entries. Then, instead of communicating perturbations $\delta$, the workers communicate indices in the shared noise table. This approach has been used in the implementation of Evolution Strategies by Moritz et al. [102] and is similar to the approach proposed by Salimans et al. [125]. Our code sets the random seeds for the random generators of all the workers and for all copies of the OpenAI Gym environments held by the workers. All these random seeds are distinct and are a function of a single integer to which we refer as *the random seed*. Furthermore, we made sure that the states and rewards produced during the evaluation rollouts were not used in any form during training.

We evaluate the performance of ARS on the MuJoCo locomotion tasks included in the OpenAI Gym-v0.9.3 [26, 153]. The OpenAI Gym provides benchmark reward functions for the different MuJoCo locomotion tasks. We used these default reward functions for evaluating the performance of the linear policies trained with ARS. The reported rewards obtained by a policy were averaged over 100 independent rollouts. For the Hopper-v1, Walker2d-v1, Ant-v1, and Humanoid-v1 tasks the default reward functions include a survival bonus, which rewards RL agents with a constant reward at each timestep, as long as a termination condition (i.e., falling over) has not been reached. During training, we removed these survival bonuses, a choice we motivate in the full version of this work [94]. The interested reader will also find in the full version of this work a sensitivity analysis of ARS to the choice of hyperparameters.

**Three random seeds evaluation:** We compare the different versions of ARS to the following methods: Trust Region Policy Optimization (TRPO), Deep Deterministic Policy Gradient (DDPG), Natural Gradients (NG), Evolution Strategies (ES), Proximal Policy Optimization (PPO), Soft Actor Critic (SAC), Soft Q-Learning (SQL), A2C, and the Cross Entropy Method (CEM). For the performance of these methods we used values reported by Rajeswaran et al. [115], Salimans et al. [125], Schulman et al. [135], and Haarnoja et al. [58]. In light of well-documented reproducibility issues of reinforcement learning methods [66, 69], reporting the values listed in papers rather than rerunning these algorithms casts prior work in the most favorable light possible.

Rajeswaran et al. [115] and Schulman et al. [135] evaluated the performance of RL algorithms on three random seeds, while Salimans et al. [125] and Haarnoja et al. [58] used six and five random seeds respectively. To put all methods on equal footing, for the evaluation of ARS, we sampled three random seeds uniformly from the interval $[0, 1000)$ and fixed them. For each of the six OpenAI Gym MuJoCo locomotion tasks we chose a grid of hyperparameters[2] and for each set of hyperparameters we ran ARS **V1**, **V2**, **V1-t**, and

[2]Recall that ARS **V1** and **V2** take in only three hyperparameters: the step-size $\alpha$, the number of perturbation directions $N$, and scale of the perturbations $\nu$. ARS **V1-t** and **V2-t** take in an additional

**V2-t** three times, once for each of the three fixed random seeds.

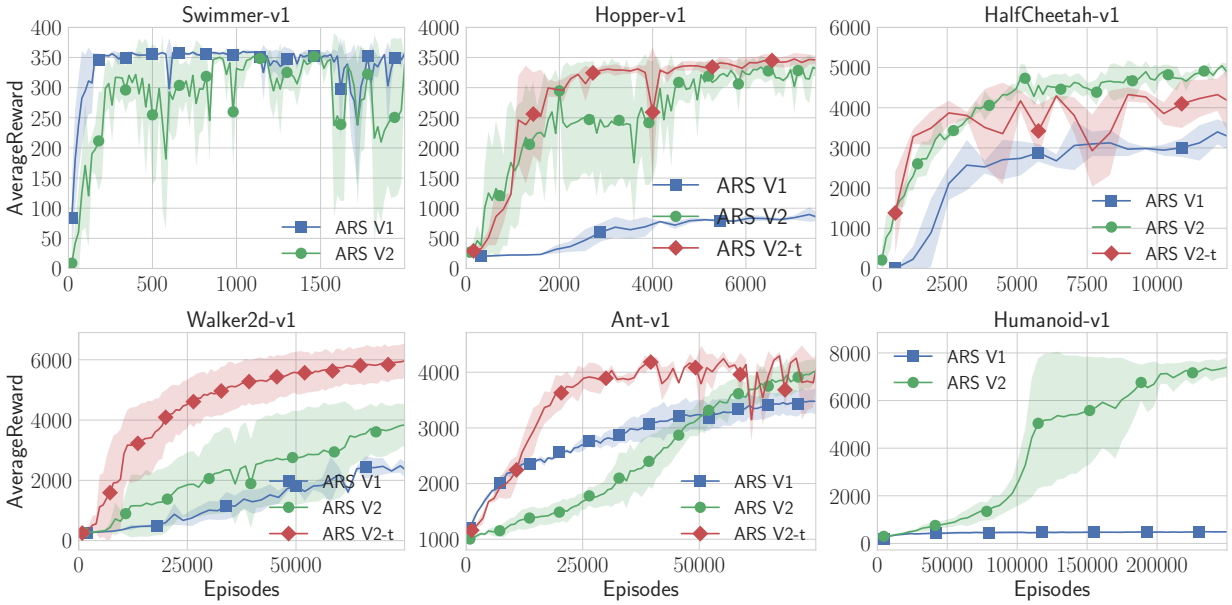


**Figure 2.2:** An evaluation of four versions of ARS on the MuJoCo locomotion tasks. The training curves are averaged over three random seeds, and the shaded region shows the standard deviation. ARS **V2-t** is only shown for the tasks to which it offered an improvement over ARS **V2**.

Table 2.1 shows the average number of episodes required by ARS, NG, and TRPO to reach a prescribed reward threshold, using the values reported by Rajeswaran et al. [115] for NG and TRPO. For each version of ARS and each MuJoCo task we chose the hyperparameters which minimize the average number of episodes required to reach the reward threshold. The corresponding training curves of ARS are shown in Figure 2.2. For all MuJoCo tasks, except Humanoid-v1, we used the same reward thresholds as Rajeswaran et al. [115]. Our choice to increase the reward threshold for Humanoid-v1 is motivated by the presence of the survival bonuses, as discussed in the full version of this work [94].

Table 2.1 shows that ARS **V1** can train policies for all tasks except Humanoid-v1, which is successfully solved by ARS **V2**. Secondly, we note that ARS **V2** reaches the prescribed thresholds for Swimmer-v1, Hopper-v1, and HalfCheetah-v1 faster than NG or TRPO, and matches the performance of NG on the Humanoid-v1. On Walker2d-v1 and Ant-v1, ARS **V2** is outperformed by NG. Nonetheless, ARS **V2-t** surpasses the performance of NG on these two tasks. Although TRPO hits the reward threshold for Walker2d-v1 faster than ARS, our method either matches or surpasses TRPO in the metrics reported by Haarnoja et al. [58] and Schulman et al. [135].

hyperparameter, the number of top directions used $b$ ($b \leq N$).

[3]N/A means that the method did not reach the reward threshold.

[4]UNK stands for unknown.

| | | Average # episodes to reach reward threshold | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Task** | **Threshold** | **ARS** | | | | **NG-lin** | **NG-rbf** | **TRPO-nn** |
| | | **V1** | **V1-t** | **V2** | **V2-t** | | | |
| Swimmer-v1 | 325 | 100 | 100 | 427 | 427 | 1450 | 1550 | N/A [3] |
| Hopper-v1 | 3120 | 89493 | 51840 | 3013 | 1973 | 13920 | 8640 | 10000 |
| HalfCheetah-v1 | 3430 | 10240 | 8106 | 2720 | 1707 | 11250 | 6000 | 4250 |
| Walker2d-v1 | 4390 | 392000 | 166133 | 89600 | 24000 | 36840 | 25680 | 14250 |
| Ant-v1 | 3580 | 101066 | 58133 | 60533 | 20800 | 39240 | 30000 | 73500 |
| Humanoid-v1 | 6000 | N/A | N/A | 142600 | 142600 | ≈130000 | ≈130000 | UNK[4] |

**Table 2.1:** A comparison of ARS, NG, and TRPO on the MuJoCo locomotion tasks. For each task we show the average number of episodes required to achieve a prescribed reward threshold, averaged over three random seeds. We estimated the number of episodes required by NG to reach a reward of 6000 for Humanoid-v1 based on the learning curves presented by Rajeswaran et al. [115].

Precise comparisons to more RL methods are provided in the full version of this work [94]. Here we offer a summary. Salimans et al. [125] reported the average number of episodes required by ES to reach a prescribed reward threshold, on four of the locomotion tasks. ARS surpassed ES on all of those tasks. Haarnoja et al. [58] reported the maximum reward achieved by SAC, DDPG, SQL, and TRPO after a prescribed number of timesteps, on four of the locomotion tasks. With the exception of SAC on HalfCheetah-v1 and Ant-v1, ARS outperformed competing methods. Schulman et al. [135] reported the maximum reward achieved by PPO, A2C, CEM, and TRPO after a prescribed number of timesteps, on four of the locomotion tasks. With the exception of PPO on Walker2d-v1, ARS matched or surpassed the performance of competing methods.

**A hundred seeds evaluation:** For a more thorough evaluation of ARS, we sampled 100 distinct random seeds uniformly at random from the interval $[0, 10000)$. Then, using the hyperparameters selected for Table 2.1, we ran ARS for each of the six MuJoCo locomotion tasks and the 100 random seeds. The results are shown in Figure 2.3. Such a thorough evaluation was feasible because ARS has a small computational footprint. ARS is at least 15 times more computationally efficient on the MuJoCo benchmarks than competing methods.

Figure 2.3 shows that 70% of the time ARS trains policies for all the MuJoCo locomotion tasks, with the exception of Walker2d-v1 for which it succeeds only 20% of the time. Moreover, ARS succeeds at training policies a large fraction of the time while using a competitive number of episodes.

There are two types of random seeds represented in Figure 2.3 that cause ARS to not reach high rewards. There are random seeds on which ARS eventually finds high reward policies when sufficiently many iterations of ARS are performed, and there are random seeds which lead ARS to discover locally optimal behaviors. For the Humanoid model, ARS found numerous distinct gaits, including ones during which the Humanoid hops only on one

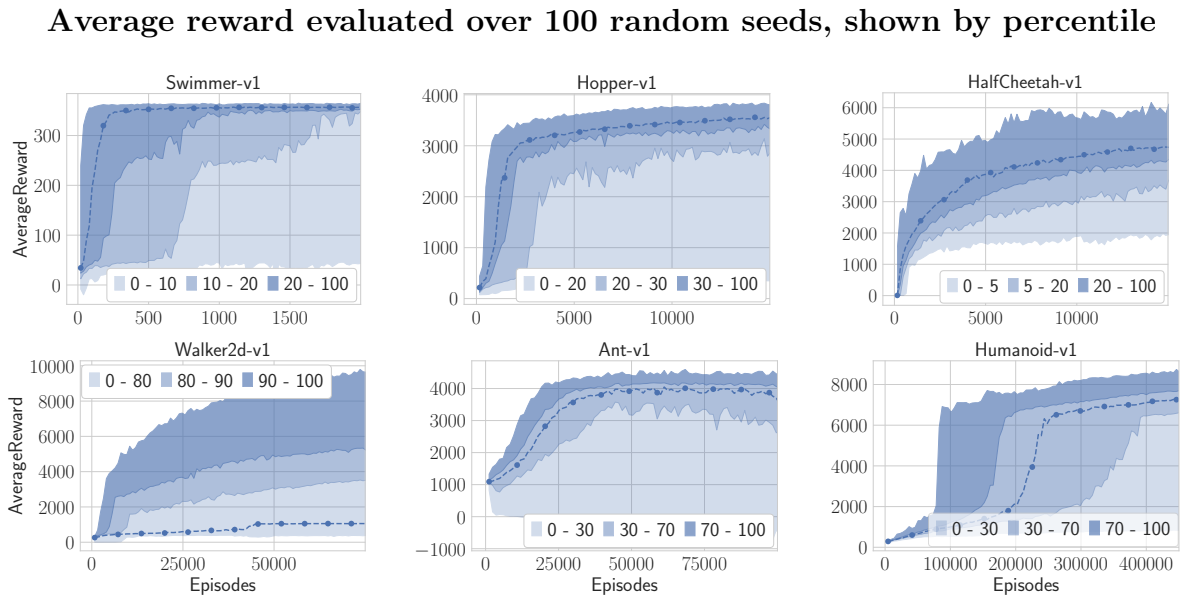


**Figure 2.3:** An evaluation of ARS over 100 random seeds on the MuJoCo locomotion tasks. The dotted lines represent median rewards and the shaded regions represent percentiles. For Swimmer-v1 we used ARS **V1**. For Hopper-v1, Walker2d-v1, and Ant-v1 we used ARS **V2-t**. For HalfCheetah-v1 and Humanoid-v1 we used ARS **V2**.

leg, walks backwards, or moves in a swirling motion. Such gaits were found by ARS on the random seeds which cause slower training. While multiple gaits for Humanoid models have been previously observed [65], our evaluation better emphasizes their prevalence. The presence of local optima is inherent to non-convex optimization, and our results show that RL algorithms should be evaluated on many random seeds for determining the frequency with which local optima are found. Finally, we remark that ARS is the least sensitive to the choice of random seed used when applied to HalfCheetah-v1, a task which is often used for the evaluation of sensitivity of algorithms to the choice of random seeds.

**Linear policies are sufficiently expressive for MuJoCo:** We discussed how linear policies can produce diverse gaits for the MuJoCo models, showing that they are sufficiently expressive to capture diverse behaviors. Table 2.2 shows that linear policies can also achieve high rewards on all the MuJoCo locomotion tasks. In particular, for Humanoid-v1 and Walker2d-v1, ARS found policies that achieve significantly higher rewards than any other results we encountered in the literature. These results show that linear policies are perfectly adequate for the MuJoCo locomotion tasks, reducing the need for more expressive and more computationally expensive policies.

**Maximum reward achieved**

| Task | ARS | Task | ARS | Task | ARS |
|---|---|---|---|---|---|
| Swimmer-v1 | 365 | HalfCheetah-v1 | 6722 | Ant | 5146 |
| Hopper-v1 | 3909 | Walker | 11389 | Humanoid | 11600 |

**Table 2.2:** Maximum average reward achieved by ARS, where we took the maximum over all sets of hyperparameters considered and the three fixed random seeds.

## 2.3 Discussion

With a few algorithmic augmentations, basic random search of static, linear policies achieves state-of-the-art sample efficiency on the MuJoCo locomotion tasks. Surprisingly, no special nonlinear controllers are needed to match the performance recorded in the RL literature. Moreover, since our algorithm and policies are simple, we were able to perform extensive sensitivity analysis. This analysis brings us to an uncomfortable conclusion that the current evaluation methods adopted in the deep RL community are insufficient to evaluate whether proposed methods are actually solving the studied problems.

The choice of benchmark tasks and the small number of random seeds do not represent the only issues of current evaluation methodology. Though many RL researchers are concerned about minimizing sample complexity, *it does not make sense to optimize the running time of an algorithm on a single problem instance.* The running time of an algorithm is only a meaningful notion if either (a) evaluated on a family of problem instances, or (b) when clearly restricting the class of algorithms.

Common RL practice, however, does not follow either (a) or (b). Instead, researchers run an algorithm $\mathcal{A}$ on a task $\mathcal{T}$ with a given hyperparameter configuration, and plot a "learning curve" showing the algorithm reaches a target reward after collecting $X$ samples. Then the "sample complexity" of the method is reported as the number of samples required to reach a target reward threshold, with the given hyperparameter configuration. However, any number of hyperparameter configurations can be tried. Any number of algorithmic enhancements can be added or discarded and then tested in simulation. For a fair measurement of sample complexity, should we not count the number of rollouts used for all tested hyperparameters?

Through optimal hyperparameter tuning one can artificially improve the perceived sample efficiency of a method. Indeed, this is what we see in our work. By adding a third algorithmic enhancement to basic random search (i.e., enhancing ARS **V2** to **V2-t**), we are able to improve the sample efficiency of an already highly performing method. Considering that most of the prior work in RL uses algorithms with far more tunable parameters and neural nets whose architectures themselves are hyperparameters, the significance of the reported sample complexities for those methods is not clear. This issue is important because a meaningful sample complexity of an algorithm should inform us on the number of samples required to solve a new, previously unseen task.

In light of these issues and of our empirical results, we make several suggestions for future work:

- Simple baselines should be established before moving forward to more complex benchmarks and methods. We propose the Linear Quadratic Regulator as a reasonable testbed for RL algorithms. LQR is well-understood when the model is known, problem instances can be easily generated with a variety of different levels of difficulty, and little overhead is required for replication.

- When games and physics simulators are used for evaluation, separate problem instances should be used for tuning and evaluating RL methods. Moreover, large numbers of random seeds should be used for statistically significant evaluations.

- Rather than trying to develop general purpose algorithms, it might be better to focus on specific problems of interest and find targeted solutions.

- More emphasis should be put on the development of model-based methods. For many problems, such methods have been observed to require fewer samples than model-free methods. Moreover, the physics of the systems should inform the parametric classes of models used for different problems. Model-based methods incur many computational challenges themselves, and it is quite possible that tools from deep RL, such as improved tree search, can provide new paths forward for tasks that require the navigation of complex and uncertain environments.

# Chapter 3

# System identification

In the previous chapter we exemplified and discussed the challenges of determining the sample complexity of reinforcement learning algorithms solely through empirical evaluation. In an attempt to eschew these issues we lay the foundation for a theoretical understanding of how machine learning interfaces with control. We start by quantifying the sample complexity of estimating transition models for the dynamical problems at hand. In control theory this area of study is called system identification. The material presented in this chapter is based on the work by Dean et al. [41], Mania et al. [96], and Simchowitz et al. [141].

In this chapter we focus on two classes of dynamical systems whose unknown parameters appear linearly. Namely, we consider linear dynamical:

$$\mathbf{x}_{t+1} = A_\star \mathbf{x}_t + B_\star \mathbf{x}_t + \mathbf{w}_t, \tag{3.0.1}$$

where $\mathbf{x}_t$ is the state of the system, $\mathbf{u}_t$ is the input to the system, and $\mathbf{w}_t$ is stochastic noise. The state $\mathbf{x}_t$ and noise $\mathbf{w}_t$ have dimension $n$ and the input $\mathbf{u}_t$ has dimension $p$. The matrices $A_\star$ and $B_\star$ are unknown matrices of appropriate dimensions.

Then, the goal is to identify $A_\star$ and $B_\star$ from data collected from the true dynamical system (3.0.1). Because of the linear structure of the dynamics we can obtain an estimate by using the ordinary least squares (OLS) estimator:

$$(\widehat{A}, \widehat{B}) \in \operatorname*{argmin}_{A,B} \sum_{t=0}^{T-1} \|A\mathbf{x}_t + B\mathbf{u}_t - \mathbf{x}_{t+1}\|^2. \tag{3.0.2}$$

Unless otherwise noted, the norm $\|\cdot\|$ denotes the Euclidean norm when applied to vectors and denotes the spectral norm when applied to matrices.

The OLS estimator is well understood when the data is generated i.i.d.. However, in the case of dynamical systems data is dependent across time, i.e., both states and inputs depend on past observations. We address this challenge and quantify the estimation errors $\|A_\star - \widehat{A}\|$ and $\|\widehat{B} - B_\star\|$ as a function of the amount of data used. In Section 3.1 we present an empirical approach based on the bootstrap, which works well in practice. Then, in Section 3.2

we discuss a theoretical approach for analyzing the estimation error of (3.0.1) and then in Section 3.3 we extend this theoretical analysis to a class of nonlinear systems. Finally, there is a long line of work studying system identification, which we discuss in Section 3.4.

## 3.1 Empirical confidence sets via the bootstrap

In the previous sections we offered theoretical guarantees on the performance of the least squares estimation of $A_\star$ and $B_\star$ from independent samples. However, there are two important limitations to using such guarantees in practice to offer upper bounds on $\epsilon_A = \|A_\star - \widehat{A}\|$ and $\epsilon_B = \|B_\star - \widehat{B}\|$. First, using only one sample per system rollout is empirically less efficient than using all available data for estimation. Second, even optimal statistical analyses often do not recover constant factors that match practice. For purposes of robust control, it is important to obtain upper bounds on $\epsilon_A$ and $\epsilon_B$ that are not too conservative. Thus, we aim to find $\widehat{\epsilon}_A$ and $\widehat{\epsilon}_B$ such that $\epsilon_A \le \widehat{\epsilon}_A$ and $\epsilon_B \le \widehat{\epsilon}_B$ with high probability.

We propose a vanilla bootstrap method for estimating $\widehat{\epsilon}_A$ and $\widehat{\epsilon}_B$. Bootstrap methods have had a profound impact in both theoretical and applied statistics since their introduction [45]. These methods are used to estimate statistical quantities (e.g. confidence intervals) by sampling synthetic data from an empirical distribution determined by the available data. For the problem at hand we propose the procedure described in Algorithm 2.[1]

**Algorithm 2** Bootstrap estimation of $\epsilon_A$ and $\epsilon_B$

**Input:** confidence parameter $\delta$, number of trials $M$, data $\{(\mathbf{x}_t^{(i)}, \mathbf{u}_t^{(i)})\}_{\substack{1\le i\le N\\ 1\le t\le T}}$, and $(\widehat{A}, \widehat{B})$ a minimizer of $\sum_{\ell=1}^{N}\sum_{t=0}^{T-1}\frac{1}{2}\|A\mathbf{x}_t^{(\ell)} + B\mathbf{u}_t^{(\ell)} - \mathbf{x}_{t+1}^{(\ell)}\|^2$.
**for** $M$ trials **do**
  **for** $\ell$ from 1 to $N$ **do**
    $\widehat{\mathbf{x}}_0^{(\ell)} = \mathbf{x}_0^{(\ell)}$
    **for** $t$ from 0 to $T-1$ **do**
      $\widehat{\mathbf{x}}_{t+1}^{(\ell)} = \widehat{A}\widehat{\mathbf{x}}_t^{(\ell)} + \widehat{B}\widehat{\mathbf{u}}_t^{(\ell)} + \widehat{\mathbf{w}}_t^{(\ell)}$ with $\widehat{\mathbf{w}}_t^{(\ell)} \sim \mathcal{N}(0, \sigma_w^2 I_n)$ and $\widehat{\mathbf{u}}_t^{(\ell)} \sim \mathcal{N}(0, \sigma_u^2 I_p)$.
    **end for**
  **end for**
  $(\widetilde{A}, \widetilde{B}) \in \arg\min_{(A,B)} \sum_{\ell=1}^{N}\sum_{t=0}^{T-1}\frac{1}{2}\|A\widehat{\mathbf{x}}_t^{(\ell)} + B\widehat{\mathbf{u}}_t^{(\ell)} - \widehat{\mathbf{x}}_{t+1}^{(\ell)}\|_2^2$.
  record $\widetilde{\epsilon}_A = \|\widehat{A} - \widetilde{A}\|_2$ and $\widetilde{\epsilon}_B = \|\widehat{B} - \widetilde{B}\|_2$.
**end for**
**Output:** $\widehat{\epsilon}_A$ and $\widehat{\epsilon}_B$, the $100(1-\delta)$th percentiles of the $\widetilde{\epsilon}_A$'s and the $\widetilde{\epsilon}_B$'s.

For $\widehat{\epsilon}_A$ and $\widehat{\epsilon}_B$ estimated by Algorithm 2 we intuitively have

$$\mathbb{P}(\|A - \widehat{A}\| \le \widehat{\epsilon}_A) \approx 1 - \delta \quad \text{and} \quad \mathbb{P}(\|B - \widehat{B}\| \le \widehat{\epsilon}_B) \approx 1 - \delta.$$

[1] We assume that $\sigma_u$ and $\sigma_w$ are known. Otherwise they can be estimated from data.

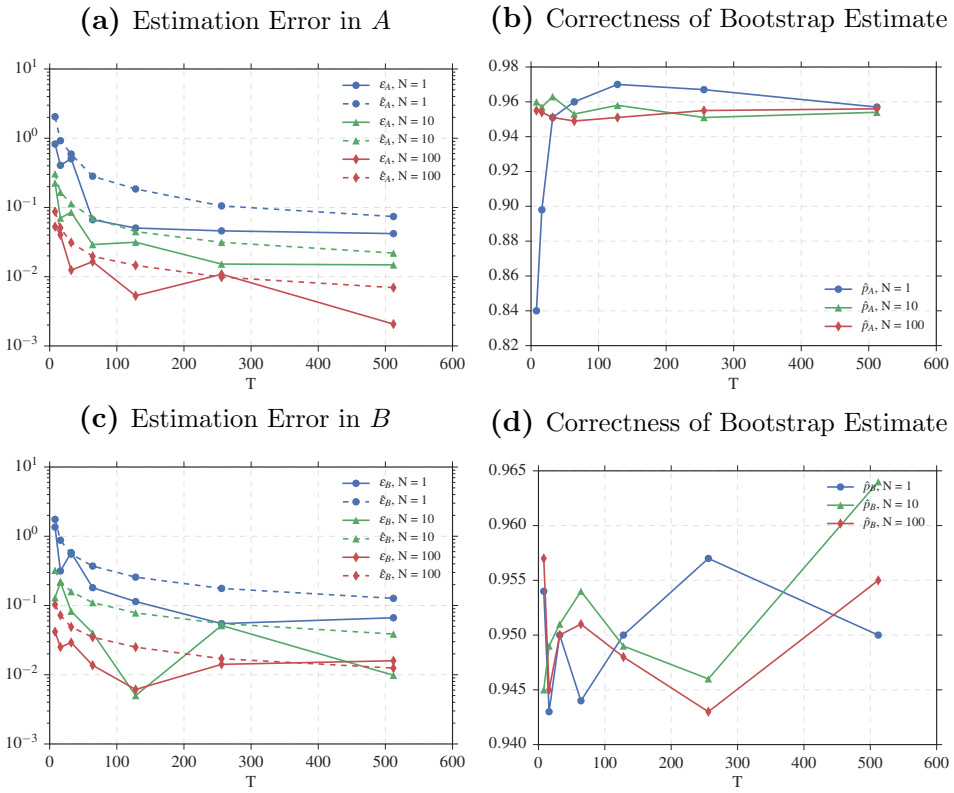


**Figure 3.1:** In these simulations: $n = 3$, $p = 1$, $\rho = 0.9$, and $M = 2000$. In (a), the spectral distances to $A_\star$ (shown in the solid lines) are compared with the bootstrap estimates (shown in the dashed lines). In (b), the probability $A_\star$ lies in $B_{\widehat{A}}(\widehat{\epsilon}_A)$ estimated from 2000 trials. In (c), the spectral distances to $B_*$ are compared with the bootstrap estimates. In (d), the probability $B_\star$ lies in $B_{\widehat{B}}(\widehat{\epsilon}_B)$ estimated from 2000 trials.

There are many known guarantees for the bootstrap, particularly for the parametric version we use. We do not discuss these results here; for more details see texts by Van Der Vaart and Wellner [158], Shao and Tu [137], and Hall [59]. Instead, we show empirically the performance of the bootstrap for our estimation problem.

We evaluate the efficacy of the bootstrap procedure introduced in Algorithm 2. Although in Section 3.2 we provide theoretical upper bounds on the estimation error of system identification, for practical purposes we want bounds that are the least conservative possible.

For given state dimension $n$, input dimension $p$, and scalar $\rho$, we generate upper triangular matrices $A_\star \in \mathbb{R}^{n\times n}$ with all diagonal entries equal to $\rho$ and the upper triangular entries i.i.d. samples from $\mathcal{N}(0,1)$, clipped at magnitude 1. By construction, matrices will have spectral radius $\rho$. The entries of $B_\star \in \mathbb{R}^{n\times p}$ were sampled i.i.d. from $\mathcal{N}(0,1)$, clipped at magnitude 1. The variance terms $\sigma_u^2$ and $\sigma_w^2$ were fixed to be 1.

Recall that $M$ represents the number of trials used for the bootstrap estimation, and $\widehat{\epsilon}_A$, $\widehat{\epsilon}_B$ are the bootstrap estimates for $\epsilon_A$, $\epsilon_B$. To check the validity of the bootstrap procedure

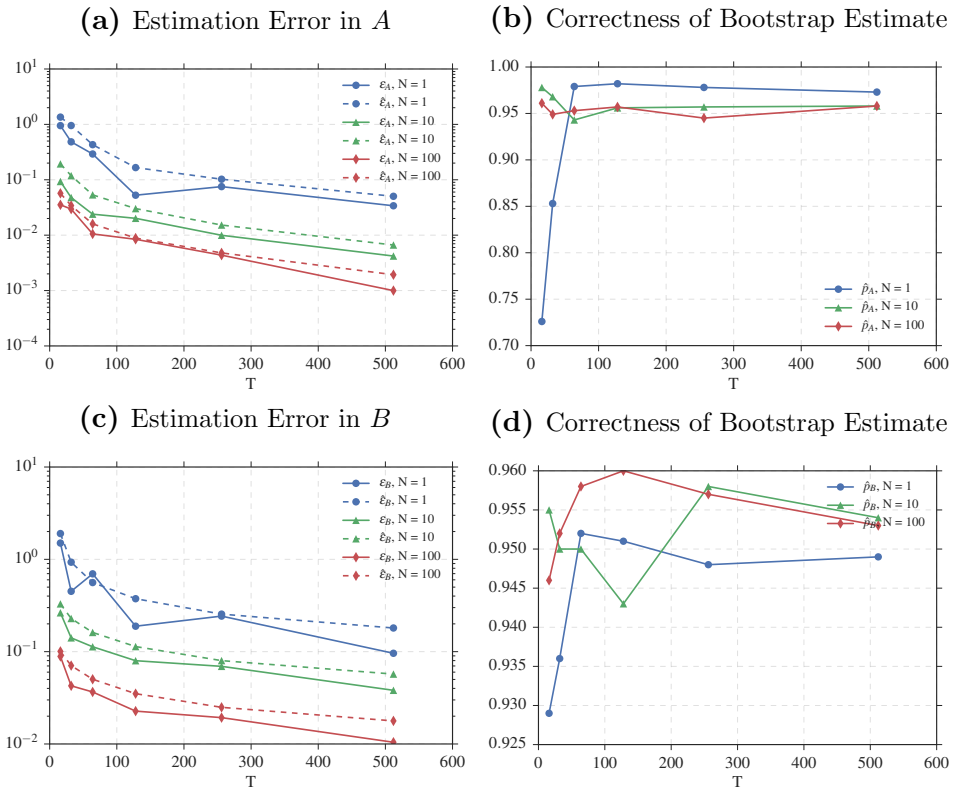


**Figure 3.2:** In these simulations: $n = 6$, $p = 2$, $\rho = 1.01$, and $M = 2000$. In (a), the spectral distances to $A_\star$ are compared with the bootstrap estimates. In (b), the probability $A_\star$ lies in $B_{\widehat{A}}(\widehat{\epsilon}_A)$ estimated from 2000 trials. In (c), the spectral distances to $B_\star$ are compared with the bootstrap estimates. In (d), the probability $B_\star$ lies in $B_{\widehat{B}}(\widehat{\epsilon}_B)$ estimated from 2000 trials.

we empirically estimate the fraction of time $A_\star$ and $B_\star$ lie in the balls $B_{\widehat{A}}(\widehat{\epsilon}_A)$ and $B_{\widehat{B}}(\widehat{\epsilon}_B)$, where $B_X(r) = \{X' \colon \|X' - X\|_2 \leq r\}$.

Our findings are summarized in Figures 3.1 and 3.2. Although not plotted, the theoretical bounds found in Section 3.2 would be orders of magnitude larger than the true $\epsilon_A$ and $\epsilon_B$, while the bootstrap bounds offer a good approximation.

## 3.2 General framework for theoretical analysis

In this work, we consider both the specific problem of estimating linear dynamical systems, where we measure the estimation error in the operator norm $\|\cdot\|$. In Section 3.2.1 we present upper bounds on the estimation error of the parameters $A_\star$ of a linear dynamical system, which hold for any $A_\star$ with $\rho(A_\star) \leq 1$. In the full version of this work we also show that these upper bounds are nearly optimal in many regimes of interest [141] and we also offer a more general results that applies to the problem of linear estimation in time series. This general

results is useful in extending the analysis to nonlinear dynamical systems, as explained in Section 3.3.3.

**Notation:** We let $\mathcal{S}^{d-1}$ denote the unit sphere in $\mathbb{R}^d$. Given a matrix $M$ we denote by $M^\dagger$ its pseudoinverse. For a symmetric matrix $M \in \mathbb{R}^{d\times d}$, we let $\lambda_{\max}(M)$ and $\lambda_{\min}(M)$ denote its largest and smallest eigenvalues. If $M \in \mathbb{R}^{d\times d}$ and $M \succ 0$, we denote by $\mathcal{S}_M$ the set of all points $x \in \mathbb{R}^d$ such that $\|M^{-1/2}x\|_2 = 1$.

### 3.2.1 Linear dynamical systems

We analyze the statistical performance of the OLS estimator of the parameter $A_\star$ from a single observed trajectory $\mathbf{x}_1, \ldots, \mathbf{x}_{T+1}$ satisfying $\mathbf{x}_{t+1} = A_\star \mathbf{x}_t + \mathbf{w}_t$, where $\mathbf{x}_0 = 0$ and $\mathbf{w}_t \sim \mathcal{N}(0, \sigma^2 I_d)$:

$$\widehat{A}(T) := \arg\min_{A\in\mathbb{R}^{d\times d}} \sum_{t=1}^{T} \frac{1}{2}\|\mathbf{x}_{t+1} - A\mathbf{x}_t\|_2^2 \,. \tag{3.2.1}$$

Our bounds are stated in terms of the finite-time controllability Gramian of the system, denoted by $\Gamma_t := \sum_{s=0}^{t-1}(A_\star^s)(A_\star^s)^\top$, which captures the magnitude of the excitations induced by the process noise. Indeed, we can write $\mathbf{x}_t$ explicitly as

$$\mathbf{x}_t = \sum_{s=1}^{t} A_\star^{t-s}\mathbf{w}_{s-1} \quad \text{which implies that} \quad \mathbb{E}[\mathbf{x}_t\mathbf{x}_t^\top] = \sigma^2\Gamma_t \,. \tag{3.2.2}$$

Hence, the expected covariance can be expressed in terms of the Gramians via $\mathbb{E}[\sum_{t=1}^{T}\mathbf{x}_t\mathbf{x}_t^\top] = \sigma^2 \cdot \sum_{t=1}^{T}\Gamma_t$. As is standard in analyses of least-squares, "larger" covariates/covariance matrices correspond to faster rates of learning. We are ready to state our first result, whose proof can be found in full version of our work [141].

**Theorem 3.2.1.** *Fix $\delta \in (0, 1/2)$ and consider the linear dynamical system $\mathbf{x}_{t+1} = A_\star\mathbf{x}_t + \mathbf{w}_t$, where $A_\star$ is a marginally stable matrix in $\mathbb{R}^{d\times d}$ (i.e. $\rho(A_\star) \le 1$), $\mathbf{x}_0 = 0$, and $\mathbf{w}_t \sim \mathcal{N}(0, \sigma^2 I)$. Then there exist universal constants $c, C > 0$ such that*

$$\mathbb{P}\left[\left\|\widehat{A}(T) - A_\star\right\|_{\mathrm{op}} > \frac{C}{\sqrt{T\lambda_{\min}(\Gamma_k)}}\sqrt{d\log\frac{d}{\delta} + \log\det(\Gamma_T\Gamma_k^{-1})}\right] \le \delta, \tag{3.2.3}$$

*for any $k \ge 1$ such that $\frac{T}{k} \ge c(d\log(d/\delta) + \log\det(\Gamma_T\Gamma_k^{-1}))$ holds.*

Note that $\sigma^2$ does not appear in the bound from Theorem 3.2.1 because scaling the noise also rescales the covariates. In the full version of this work [141], we show that for any marginally stable $A_\star$, we can always choose a $k \ge 1$ provided $T$ is sufficiently large. Therefore, even when $\rho(A_\star) = 1$ and the system does not mix, we obtain finite-sample estimation guarantees which also guarantees consistency of estimation. In many cases, these rates are qualitatively no-worse than random-design linear regression with independent covariates.

In general, $\lambda_{\min}(\Gamma_k)$ is a nondecreasing function of the block length $k$. The intuition for this is that larger $k$ takes into account more long-term excitations to lower bound the size of our covariance matrix. However, as we use longer blocks, our high probability bounds degrade. Thus, the optimal block length is the maximal value $k$ which satisfies the condition in Theorem 3.2.1.

The dependence on the minimum eigenvalue of the Gramian $\lambda_{\min}(\Gamma_k)$ has two interpretations. From a *statistical* perspective, we have $\frac{1}{2k\cdot\sigma^2}\mathbb{E}[\sum_{t=1}^{2k}\mathbf{x}_t\mathbf{x}_t^\top] = \frac{1}{2k}\sum_{t=1}^{2k}\Gamma_t \succeq \frac{1}{2}\lambda_{\min}(\Gamma_k)\cdot I$. Thus, $\lambda_{\min}(\Gamma_k)$ gives a lower bound on the smallest eigenvalue value of the covariance matrix associated with the first $2k$ covariates. In fact, one can also show that for any $t_0 \geq 0$, we still have $\frac{1}{2k\cdot\sigma^2}\mathbb{E}[\sum_{t=t_0+1}^{t_0+2k}\mathbf{x}_t\mathbf{x}_t^\top|\mathbf{x}_{t_0}] \succeq \frac{1}{2}\lambda_{\min}(\Gamma_k)\cdot I$. Theorem 3.2.1 thus states that the larger the expected covariance matrix, the faster $A_\star$ is estimated. Note that $\Gamma_k \succeq I$ for all $k \geq 1$.

The second interpretation is *dynamical*. The term $\lambda_{\min}(\Gamma_k)$ corresponds to the "excitability" of the system, which is the extent to which the process noise influences future covariates. This can be seen from (3.2.2), where the slower $(A_\star^{t_0})(A_\star^{t_0})^\top$ decays as $t_0$ grows, the larger the contribution of $\mathbf{w}_{t-t_0-1}$ is. This is precisely the reason why linear systems with larger spectral radii mix slowly, and do not mix when $\rho(A_\star) \geq 1$. In this light, Theorem 3.2.1 shows that with high-probability, the more a linear system is excited by the noise $\mathbf{w}_t$, the easier it is to estimate the parameter matrix $A_\star$. We now explicitly describe the consequences of Theorem 3.2.1 for three illustrative classes of linear systems:

1. **Scalar linear system.** In this case the states $\mathbf{x}_t$ and the parameter $A_\star$ are scalars, and denoted $a_* = A_\star$. For $|a_*| \leq 1$, we can apply Theorem 3.2.1 with block length $k = \mathcal{O}(T/\log(1/\delta))$. This then guarantees that $|\widehat{a} - a_*| \leq \mathcal{O}\left(\sqrt{\log(1/\delta)/\left(T\sum_{t=1}^{k_*}a_*^{2t}\right)}\right)$ with probability $1-\delta$. In the full version of this work [141], we show this statistical rate is minimax optimal. In Section 3.2.2, we offer a specialized analysis for the scalar case (Theorem 3.2.4) which yields sharper constants and also applies to the unstable case $|a_*| > 1$. Stated succinctly, our results in Section 3.2.2 imply that the OLS estimator satisfies with probability $1-\delta$ error guarantees which can be categorized into three regimes:

$$|\widehat{a} - a_*| = \begin{cases} \Theta\left(\sqrt{\frac{\log(1/\delta)(1-|a_*|)}{T}}\right) & \text{if } |a_*| \leq 1 - \frac{c\log(1/\delta)}{T}, \\ \Theta\left(\frac{\log(1/\delta)}{T}\right) & \text{if } 1 - \frac{c\log(1/\delta)}{T} < |a_*| \leq 1 + \frac{1}{T} \\ \Theta\left(\frac{\log(1/\delta)}{|a_*|^T}\right) & \text{if } 1 + \frac{1}{T} \leq |a_*|. \end{cases}$$

   White [163] showed the same dependence on $|a_*|$ of the estimation error by characterizing the asymptotic distribution of $\widehat{a} - a_*$ when appropriately scaled. However, our results offer finite sample guarantees.

2. **Scaled orthogonal systems.** Let us assume $A_\star = \rho \cdot O$ for an orthogonal $d \times d$ matrix $O$ and $|\rho| \leq 1$. Then, one can verify that $\Gamma_t = I \cdot \sum_{s=0}^{t-1}\rho^{2s}$ and that we can

choose the block length $k = \mathcal{O}\left(\frac{T}{d\log(d/\delta)}\right)$. Therefore, Theorem 3.2.1 guarantees that with probability $1-\delta$:

$$\|\widehat{A} - A_\star\|_{\text{op}} \leq \begin{cases} \mathcal{O}\left(\sqrt{(1-|\rho|)\cdot\frac{d\log(d/\delta)}{T}}\right) & \text{if } |\rho| \leq 1 - \frac{cd\log(d/\delta)}{T}, \\ \mathcal{O}\left(\frac{d\log(d/\delta)}{T}\right) & \text{if } 1 - \frac{cd\log(d/\delta)}{T} < |\rho|. \end{cases} \tag{3.2.4}$$

3. **Diagonalizable linear systems.** Let $A_\star = SDS^{-1}$ define a diagonalizable linear system. We denote by $\underline{\rho}$ the smallest magnitude of an eigenvalue of $A_\star$. It can be shown that the block length $k$ can be chosen such that $k \geq \frac{T}{cd\log\left(\frac{d\,\text{cond}(S)}{\delta}\right)}$. With this choice of $k$ the OLS estimator satisfies

$$\mathbb{P}\left[\|\widehat{A} - A_\star\|_{\text{op}} \leq \mathcal{O}\left(\sqrt{\frac{d\log(d\,\text{cond}(S)/\delta)}{T\left(1+\text{cond}(S)^{-2}\sum_{s=0}^{k-1}\underline{\rho}^{2s}\right)}}\right)\right] \geq 1-\delta$$

which could once again be split into a slow and fast rate, as in the examples presented above, depending on the size $\underline{\rho}$ of the least excitable mode of the system defined by $A_\star$. Note that up to a factor of $\log(d\,\text{cond}(S)/\delta)$, the above bound is no worse than the worst-case rate for standard random-design least-squares in the operator norm.

**Remark 3.2.2** (Noise dependence)**.** *As mentioned before, the estimation guarantee provided by Theorem 3.2.1 does not depend on the variance $\sigma^2$ of the noise $\mathbf{w}_t$. For Gaussian noise with a general identity covariance $\mathbf{w}_t \sim \mathcal{N}(0,\Sigma)$, one can rederive rates from a more general theorem, shown in the full version of this work, to get a more precise dependence on $\Gamma_t$ and $\Sigma$. Note that if the covariance $\Sigma$ is known, an alternative estimator would be to choose $\widehat{A}$ to minimize a loss which takes $\Sigma$ into account in the same way that one would for non-dynamic linear regression with heteroskedastic noise, e.g. $\widehat{A}^{\Sigma}(T) := \arg\min_{A\in\mathbb{R}^{d\times d}}\sum_{t=1}^{T}\frac{1}{2}\left\|\Sigma^{-1/2}\left(\mathbf{x}_{t+1} - A\mathbf{x}_t\right)\right\|_2^2$.*

**Remark 3.2.3** (Learning with input sequences)**.** *We can also consider the case where the linear system $\mathbf{x}_{t+1} = A_\star\mathbf{x}_t + B_\star u_t + \eta_t$ is driven by a known sequence of inputs $u_0, u_1, \ldots,$ with known $B_\star$. Defining the control Gramian $\Gamma_t^{B_*} := \sum_{s=1}^{t} A_\star^{t-s}B_\star B_\star^\top A_\star^{t-s}$ , the proof of Theorem 3.2.1 can be modified to show that, if the inputs are white noise $u_t \overset{i.i.d}{\sim} \mathcal{N}(0,\sigma_u^2 I)$, then there exist universal constants $c, C > 0$ such that, with probability $1-\delta$,*

$$\|\widehat{A}(T) - A_\star\|_{\text{op}} \leq \frac{C\sigma^2}{\sqrt{T\lambda_{\min}\left(\sigma^2\Gamma_k + \sigma_u^2\Gamma_k^{B_*}\right)}}\sqrt{d\log\left(\frac{1}{\delta}\frac{\text{tr}\left(\sigma^2\Gamma_T + \sigma_u^2\Gamma_T^{B_*}\right)}{\lambda_{\min}\left(\sigma^2\Gamma_k + \sigma_u^2\Gamma_k^{B_*}\right)}\right)}$$

*for any $k$ such that $\frac{T}{k} \geq cd\log\left(\frac{\text{tr}(\sigma^2\Gamma_T + \sigma_u^2\Gamma_T^{B_*})}{\delta\lambda_{\min}(\sigma^2\Gamma_k + \sigma_u^2\Gamma_k^{B_*})}\right)$. Process noise with covariance not equal to a multiple of the identity can be absorbed into $B_\star$.*

### 3.2.2 Scalar case

In this appendix, we present specialized upper and lower bounds in the case of scalar systems. Specifically, we consider $x_{t+1} = a_* x_t + \mathbf{w}_t$, where $\mathbf{w}_t \sim \mathcal{N}(0, \sigma^2)$, and $x_0 = 0$. Our upper bound has sharp, explicit constants, and captures the correct qualitative behavior for unstable scalar systems:

**Theorem 3.2.4.** *Let $\epsilon \in (0,1)$ and $\delta \in (0, 1/2)$. Then $\mathbb{P}[|\widehat{a}(T) - a_*| \le \epsilon] \ge 1 - \delta$ as long as*

$$T \ge \begin{cases} \frac{8}{\epsilon}\log\left(\frac{2}{\delta}\right) + \frac{4}{\epsilon^2}(1 - (|a_*| - \epsilon)^2)\log\left(\frac{2}{\delta}\right) & a_* \le 1 + \epsilon \\ \max\left\{\frac{8}{(|a_*|-\epsilon)^2 - 1}\log\left(\frac{2}{\delta}\right), \frac{4\log(\frac{1}{\epsilon})}{\log(|a_*|-\epsilon)} + 8\log\left(\frac{2}{\delta}\right)\right\} & a_* > 1 + \epsilon \ . \end{cases}$$

To prove Theorem 3.2.4, we write the error $E = \hat{a} - a = \frac{\sum_{t=0}^{T-1} x_t \mathbf{w}_t}{\sum_{t=0}^{T-1} x_t^2}$. Since are interested in upper bounding the probability that $|E| > \epsilon$ it suffices to to show that the following two probabilities are small:

$$\mathbb{P}\left(\epsilon \sum_{t=0}^{T-1} x_t^2 - \sum_{t=0}^{T-1} x_t \mathbf{w}_t < 0\right) \quad \text{and} \quad \mathbb{P}\left(\epsilon \sum_{t=0}^{T-1} x_t^2 + \sum_{t0}^{T-1} x_t \mathbf{w}_t < 0\right).$$

These probabilities are upper bounded by a standard Chernoff bound

$$\mathbb{P}\left(\epsilon \sum_{t=0}^{T-1} x_t^2 \pm \sum_{t=0}^{T-1} x_t \mathbf{w}_t < 0\right) \le \inf_{\lambda \le 0} \mathbb{E} \exp\left(\lambda\epsilon \sum_{t=0}^{T-1} x_t^2 \pm \lambda \sum_{t=0}^{T-1} x_t \mathbf{w}_t\right). \tag{3.2.5}$$

We will apply this equation with $\lambda = -\epsilon$, controlling its magnitude with following lemma.

**Lemma 3.2.5.** *Let $a$, $\nu$, $\mu$, and $x$ be real numbers with $\nu < 1$ and let $\mathbf{w} \sim \mathcal{N}(0,1)$. Then*

$$\mathbb{E}_{\mathbf{w}} \exp\left(\tfrac{\nu}{2}(ax + \mathbf{w})^2 + \mu x \mathbf{w}\right) = \frac{\exp\left(x^2 \frac{\nu a^2 + 2\nu a\mu + \mu^2}{2(1-\nu)}\right)}{\sqrt{1-\nu}}.$$

*Proof.*

$$\mathbb{E}_{\mathbf{w}} \exp\left(\frac{\nu}{2}(ax + \mathbf{w})^2 + \mu x \mathbf{w}\right) = e^{\frac{\nu}{2}a^2x^2} \mathbb{E}_{\mathbf{w}} e^{\frac{\nu}{2}\mathbf{w}^2 + \mathbf{w}x(\nu a + \mu)}$$

$$= \frac{e^{\frac{\nu}{2}a^2x^2}}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{\frac{\nu-1}{2}\mathbf{w}^2 + \mathbf{w}x(\nu a+\mu)} d\mathbf{w} = e^{\frac{\nu}{2}a^2x^2} \frac{e^{x^2\frac{(\nu a+\mu)^2}{2(1-\nu)}}}{\sqrt{1-\nu}} = \frac{\exp\left(x^2\frac{\nu a^2 + 2\nu a\mu + \mu^2}{2(1-\nu)}\right)}{\sqrt{1-\nu}}.$$

□

With this lemma in hand, we can construct a recursive sequence which upper bounds $|a - \widehat{a}|$ with high probability:

**Proposition 3.2.6.** *Let $a$ be a real number and for $\alpha \in \mathbb{R}_+$ and $\epsilon \in (0,1)$ define recursively the sequence $\rho_t$ by $\rho_{T-1} = 1$ and*

$$\rho_t = \begin{cases} 1 + r\rho_{t+1} & \rho_{t+1} \leq \alpha/\epsilon^2, \\ \alpha/\epsilon^2 & \rho_{t+1} > \alpha/\epsilon^2. \end{cases} \qquad \text{where } r = \frac{(|a| - \epsilon)^2}{1+\alpha}.$$

*With this notation,* $\mathbb{P}\left(|\widehat{a} - a| \leq \epsilon\right) \leq 2\exp\left(-\frac{\epsilon^2}{2(1+\alpha)} \sum_{t=1}^{T-1} \rho_t\right)$.

*Proof.* The proof of this result is similar to the proof of the Azuma-Hoeffding inequality. It requires upper-bounding the MGF introduced in (3.2.5) by inductively applying the tower property of conditional expectation. We restrict ourselves to the case $a \geq 0$ (the case $a < 0$ can be analyzed analogously), and hence $r = (a-\epsilon)^2/(1+\alpha)$. We upper bound the MGF (3.2.5) when $\lambda = -\epsilon$. Note that

$$\begin{aligned} \mathbb{E}\exp\left(-\epsilon^2 \sum_{t=0}^{T-1} x_t^2 \pm \epsilon \sum_{t=0}^{T-1} x_t \mathbf{w}_t\right) &= \mathbb{E}\left[e^{-\epsilon^2 \sum_{t=0}^{T-1} x_t^2 \pm \epsilon \sum_{t=0}^{T-2} x_t \mathbf{w}_t} \mathbb{E}_{\mathbf{w}_{T-1}}\left[e^{\pm \epsilon x_{T-1} \mathbf{w}_{T-1}} \,|\mathcal{F}_{T-1}\right]\right] \\ &= \mathbb{E}\left[e^{-\epsilon^2 \sum_{t=0}^{T-2} x_t^2 \pm \epsilon \sum_{t=0}^{T-3} x_t \mathbf{w}_t} \mathbb{E}\left[e^{-\frac{\epsilon^2}{2} x_{T-1}^2 \pm \epsilon x_{T-2} \mathbf{w}_{T-2}} \,|\mathcal{F}_{T-2}\right]\right]. \end{aligned}$$

Then, from Lemma 3.2.5 we can upper bound the MGF by induction on $k$ by

$$\mathbb{E}\left[e^{-\epsilon^2 \sum_{t=0}^{T-k-1} x_t^2 - \epsilon \sum_{t=0}^{T-k-2} x_t \mathbf{w}_t} \mathbb{E}\left[e^{-\frac{\epsilon^2 \beta_{T-k}}{2} x_{T-k}^2 - \epsilon x_{T-k-1} \mathbf{w}_{T-k-1}} \,|\mathcal{F}_{T-k-1}\right]\right] \prod_{j=T-k+1}^{T-1} (1+\epsilon^2\beta_j)^{-1/2},$$

where $\beta_t$ is any positive sequence such that $\beta_{T-1} = 1$ and for $1 \leq t < T-1$ it satisfies $\beta_t \leq 1 + \frac{\beta_{t+1}(a-\epsilon)^2}{1+\epsilon^2\beta_{t+1}}$. It is straightforward to check that the sequence $\rho_t$ defined in the proposition statement above satisfies this recursive inequality for any $\alpha \in (0,1)$. Therefore, we obtain the upper bound

$$\begin{aligned} \mathbb{E}\exp\left(-\epsilon^2 \sum_{t=0}^{T-1} x_t^2 - \epsilon \sum_{t=0}^{T-1} x_t \mathbf{w}_t\right) &\leq \prod_{t=1}^{T-1} (1+\epsilon^2\rho_t)^{-1/2} = \exp\left(\sum_{t=1}^{T-1} -\frac{1}{2}\log(1+\epsilon^2\rho_t)\right) \\ &\leq \exp\left(\sum_{t=1}^{T-1} -\frac{\epsilon^2\rho_t}{2(1+\epsilon^2\rho_t)}\right) \leq \exp\left(-\frac{\epsilon^2}{2(1+\alpha)} \sum_{t=1}^{T-1} \rho_t\right). \end{aligned}$$

□

Now, we return to the proof of Theorem 3.2.4. Once again we let $a \geq 0$ for simplicity and recall from Proposition 3.2.6 that we denote $r = (a-\epsilon)^2/(1+\alpha)$. We study the case $a \leq 1$ first. Let us consider the sequence $\rho_t$ introduced in Proposition 3.2.6 with $\alpha = 2\epsilon$ and note that

$$1 + r + \ldots + r^t \leq 1 + (1+2\epsilon)^{-1} + \ldots + (1+2\epsilon)^{-t} \leq \frac{1}{1-(1+2\epsilon)^{-1}} \leq \frac{2}{\epsilon},$$

which shows that for all $t$ we have $\rho_{T-1-t} = 1+r+\ldots+r^t$ and hence $\sum_{t=1}^{T-1}\rho_t = \sum_{t=1}^{T-1}\frac{1-r^t}{1-r} = \frac{T}{1-r} - \frac{\sum_{t=0}^{T-1} r^t}{1-r}$. Since $T/2 \geq 1+r+r^2+\ldots+r^{T-1}$ when $T \geq 6/\epsilon$, we obtain that $\sum_{T=1}^{T-1}\rho_t \geq \frac{T}{2(1-r)}$,, which, together with Proposition 3.2.6, it implies that

$$\mathbb{P}(|\widehat{a}-a| \leq \epsilon) \leq 2\exp\left(-\frac{\epsilon^2 T}{4(1+2\epsilon)(1-r)}\right) = 2\exp\left(-\frac{\epsilon^2 T}{4(1+2\epsilon-(a-\epsilon)^2)}\right).$$

The first part of the corollary follows immediately.

We turn to the case $|a| > 1+\epsilon$. Once again we assume $a > 0$ for simplicity. Recall that we have the freedom to choose any $\alpha \in \mathbb{R}_+$ for defining the sequence $\rho_t$. Since $a > 1+\epsilon$, if we choose $\alpha < (a-\epsilon)^2 - 1$ we guarantee that $r > 1$. To satisfy this inequality we choose $\alpha = ((a-\epsilon)^2-1)/2$. Then, with this choice of $\alpha$, the sequence $\rho_t$ grows exponentially to $\alpha/\epsilon^2$. More precisely, by construction, since

$$\left[\frac{(a-\epsilon)^2}{1+\alpha}\right]^{T-2} = \left[\frac{2(a-\epsilon)^2}{1+(a-\epsilon)^2}\right]^{T-2} \geq (a-\epsilon)^{T-2},$$

$\rho_1$ is guaranteed to be equal to $\alpha/\epsilon^2$ as long as $(a-\epsilon)^{T-2} \geq \alpha/\epsilon^2$. This last inequality holds when $T \geq \frac{\log\left(\frac{(a-\epsilon)^2-1}{2\epsilon^2}\right)}{\log(a-\epsilon)} + 2$. In particular, if we choose $T$ to be at least double the right-hand side of the previous expression, then at least half of the terms $\rho_t$ are equal to $\alpha/\epsilon^2$, implying

$$\mathbb{P}(|\widehat{a}-a| \leq \epsilon) \leq 2\exp\left(-\frac{\alpha T}{4(1+\alpha)}\right).$$

The conclusion now follows easily.

## 3.3 Extension to nonlinear dynamics

The estimation of nonlinear dynamical systems with continuous states and inputs is generally based on data-collection procedures inspired by the study of optimal input design for linear dynamical systems [131]. Unfortunately, these data-collection methods are not sufficient in general to enable the estimation of nonlinear systems. To attempt to circumvent this issue, studies of system identification have either assumed that the available data is informative enough for estimation [67, 89, 131] or considered systems for which i.i.d. random inputs produce informative data [19, 53, 109, 130]. However, as we will see, there are many nonlinear dynamical systems that require a more judicious choice of inputs for estimation to be possible.

Inspired by experimental design and active learning, we present a data-collection scheme that is guaranteed to enable system identification in finite time. Our method applies to dynamical systems whose transitions depend linearly on a known feature embedding of state-input pairs. This class of models can capture many types of dynamics and is used widely in system identification [67, 89]. For example, Ng et al. [106] used such a model to estimate

the dynamics of a helicopter and Brunton et al. [28] showed that sparse linear regression of polynomial and trigonometric feature embeddings can be used to fit models of the chaotic Lorentz system and of a fluid shedding behind an obstacle. These models can be parametrized as follows:

$$\mathbf{x}_{t+1} = A_\star \phi(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{w}_t, \tag{3.3.1}$$

where $\mathbf{x}_t$ and $\mathbf{u}_t$ are the state and input of the system at time $t$, and $\mathbf{w}_t$ is stochastic noise. The feature map $\phi$ is assumed known and the goal is to estimate $A_\star$ from one trajectory by choosing a good sequence of inputs. The input $\mathbf{u}_t$ is allowed to depend on the history of states $\{\mathbf{x}_j\}_{j=0}^t$ and is independent of $\mathbf{w}_t$.

The class of systems (3.3.1) contains any linear system, with fully observed states, when the features include the states and inputs of the system. Moreover, any piecewise affine (PWA) system can be expressed using (3.3.1) if the support of its pieces is known. First introduced by Sontag [146] as an approximation of nonlinear systems, PWA systems are a popular model of hybrid systems [24, 32, 64] and have been successfully used in a wide range of applications [23, 55, 60, 97, 124, 168].

While linear dynamical systems can be estimated from trajectories induced by i.i.d. random inputs [141], the following example shows that this is not possible for PWA systems.

**Example 3.3.1.** *Let us consider the feature map* $\phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{3d}$ *defined by:*

$$\phi(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{x} \cdot \mathbb{1}\{\|\mathbf{x}\| \leq \frac{3}{2}\} \\ \mathbf{x} \cdot \mathbb{1}\{\|\mathbf{x}\| > \frac{3}{2}\} \\ \mathbf{u} \cdot \mathbb{1}\{\|\mathbf{u}\| \leq 1\} \end{bmatrix},$$

*where* $\mathbb{1}\{\cdot\}$ *is the indicator function and the multiplication with* $\mathbf{x}$ *is coordinatewise. We assume there is no process noise and let* $A_\star = \begin{bmatrix} \frac{1}{2} I_d & A_2 & I_d \end{bmatrix}$ *for some* $d \times d$ *matrix* $A_2$ *and the* $d \times d$ *identity matrix* $I_d$*. Also, we assume* $\mathbf{x}_0 = 0$*.*

*Then, since the inputs to the system can have magnitude at most* 1*, the state of the system can have magnitude larger than* $3/2$ *only if consecutive inputs point in the same direction. However, the probability that two or more random vectors, uniformly distributed on the unit sphere, point in the same direction is exponentially small in the dimension* $d$*. Therefore, if we used random inputs, we would have to wait for a long time in order to reach a state with magnitude larger than* $3/2$*.*

*On the other hand, if we chose a sequence of inputs* $\mathbf{u}_t = \mathbf{u}$ *for a fixed unit vector* $\mathbf{u}$*, we would be guaranteed to reach a state with norm larger than* $3/2$ *in a couple of steps. Hence, despite the input constraint, we would be able to reach the region* $\|\mathbf{x}\| > 3/2$ *with a good choice of inputs.* ■

Therefore, in general the estimation of (3.3.1) requires a judicious choice of inputs. To address this challenge we propose a method based on trajectory planning, which, at a high level, repeats three steps:

- Given past observations and an estimate $\widehat{A}$, our method plans a reference trajectory from the current state of the system to a high uncertainty region of the feature space.
- Then, our method attempts to track the reference trajectory using $\widehat{A}$.
- Finally, using all data collected so far, our method re-estimates $\widehat{A}$.

The ability to find reference trajectories from a given state to a desired goal set is related to the notion of controllability, a standard notion in control theory. A system is called *controllable* if it is possible to take the system from any state to any other state in a finite number of steps by using an appropriate sequence of inputs. In our case, a system is considered more controllable the bigger we can make the inner product between the system's features and goal directions in feature space. The number of time steps required to obtain a large inner product is called the *planning horizon*.

The controllability of the system and the planning horizon are system-dependent properties that influence our ability to estimate the system. Intuitively, the more controllable a system is, the easier it is to collect the data we need to estimate it. The following informal version of our main result clarifies this relationship.

**Theorem 3.3.2** (Informal)**.** *Our method chooses actions* $\mathbf{u}_t$ *such that with high probability the ordinary least squares (OLS) estimate* $\widehat{A} \in \arg\min_A \sum_{t=0}^{T-1} \|A\phi(\mathbf{x}_t, \mathbf{u}_t) - \mathbf{x}_{t+1}\|^2$ *satisfies*

$$\|\widehat{A} - A_\star\| \leq \frac{\textit{size of the noise}}{\textit{controllability of the system}} \sqrt{\frac{\textit{dimension} \times \textit{planning horizon}}{\textit{number of data points}}}.$$

This statistical rate is akin to that of standard supervised linear regression, but it has an additional dependence on the controllability of the system and the planning horizon. To better understand why these two terms appear, recall that our method uses $\widehat{A}$, an estimate of $A_\star$, to plan and track reference trajectories. Therefore, the tracking step is not guaranteed to reach the desired region of the feature space. The main insight of our analysis is that when trajectory tracking fails, we are still guaranteed to collect at least one informative data point per reference trajectory. Therefore, in the worst case, the effective size of the data collected by our method is equal to the total number of data points collected over the planning horizon.

In the next section we present our mathematical assumptions and in Section 3.3.2 we discuss our method and main result. Section 3.3.3 includes a general result derived using the same techniques as those that led us to the results shown in Section 3.2. Then, in Section 3.3.4 we present in detail the proof of our main result.

**Notation:** We use $c_1$, $c_2$, $c_3$, ... to denote different universal constants. Also, $\mathbb{S}^{p-1}$ is the unit sphere in $\mathbb{R}^p$ and $\mathbb{B}_r^p$ is the ball in $\mathbb{R}^p$ centered at the origin and of radius $r$. The symbol ■ is used to indicate the end of an example or of a proof.

### 3.3.1 Assumptions

To guarantee the estimation of (3.3.1) we must make several assumptions about the true system we are trying to identify. We denote the dimensions of the states and inputs by $d$ and $p$ respectively. The feature map $\phi$ maps state-action pairs to feature vectors in $\mathbb{R}^k$.

The main challenge in the estimation of (3.3.1) is choosing inputs $\mathbf{u}_t$ so that the minimal singular value of the design matrix is $\Omega(\sqrt{T})$, where $T$ is the length of the trajectory collected from the system. To reliably achieve this we must assume the feature map $\phi$ has some degree of smoothness. Without a smoothness assumption the noise term $\mathbf{w}_t$ at time $t$ might affect the feature vector $\phi(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})$ at time $t+1$ in arbitrary ways, regardless of the choice of input at time $t$.

**Assumption 1.** *The map $\phi\colon \mathbb{R}^d \times \mathbb{B}^p_{r_u} \to \mathbb{R}^k$ is L-Lipschitz.*[2]

In order to use known techniques for the analysis of online linear least squares [2, 39, 123, 141] we also assume that the feature map $\phi$ is bounded. For some classes of systems (e.g., certain linear systems) this condition can be removed [141].

**Assumption 2.** *There exists $b_\phi > 0$ such that $\|\phi(\mathbf{x}, \mathbf{u})\| \le b_\phi$ for all $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{u} \in \mathbb{B}_{r_u}$.*

This assumption implies that the states of the system (3.3.1) are bounded, a consequence which can be limiting in some applications. To address this issue we could work instead with the system

$$\mathbf{x}_{t+1} = A_\star \phi(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{x}_t + \mathbf{w}_t. \tag{3.3.2}$$

In this case, $\phi$ being bounded implies that the increments $\mathbf{x}_{t+1} - \mathbf{x}_t$ are bounded, allowing the states to grow in magnitude. However, formulation (3.3.2) complicates the exposition so we choose to focus on (3.3.1).

As mentioned in the introduction, our method relies on trajectory planning and tracking to determine the inputs to the system. Suppose we would like to track a reference trajectory $\{(\mathbf{x}^R_t, \mathbf{u}^R_t)\}_{t\ge 0}$ that satisfies $\mathbf{x}^R_{t+1} = A_\star \phi(\mathbf{x}^R_t, \mathbf{u}^R_t)$. In other words, we wish to choose inputs $\mathbf{u}_t$ to ensure that the tracking error $\|\mathbf{x}_t - \mathbf{x}^R_t\|$ is small. Simply choosing $\mathbf{u}_t = \mathbf{u}^R_t$ does not work even when the initial states $\mathbf{x}_0$ and $\mathbf{x}^R_0$ are equal because the true system (3.3.1) experiences process noise.

To ensure that tracking is possible we assume that there always exists an input to the true system that can keep the tracking error small. There are multiple ways to formalize such an assumption. We make the following choice.

[2] Since $\phi$ is continuous and since $\mathbf{u}$ lies in a compact set, we know that any continuous function of $\phi(\mathbf{x}, \mathbf{u})$ achieves its maximum and minimum with respect to $\mathbf{u}$. This is the only reason we assume the inputs to the system are bounded. Alternatively, we could let the inputs be unbounded and work with approximate maximizers and minimizers.

**Assumption 3.** *There exist positive constants $\gamma$ and $b_u$ such that for any $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ and any $\mathbf{u}' \in \mathbb{B}^p_{b_u}$ we have*

$$\min_{\mathbf{u} \in \mathbb{B}^p_{r_u}} \|A_\star \left(\phi(\mathbf{x}, \mathbf{u}) - \phi(\mathbf{x}', \mathbf{u}')\right)\| \leq \gamma \|\mathbf{x} - \mathbf{x}'\|. \tag{3.3.3}$$

*Moreover, if $\|\mathbf{u}'\| \leq b_u/2$, there exists $\mathbf{u}$, with $\|\mathbf{u}\| \leq b_u$, that satisfies (3.3.3).*

Suppose we wish to track a trajectory $\{(\mathbf{x}^R_t, \mathbf{u}^R_t)\}_{t \geq 0}$ that satisfies $\mathbf{x}_{t+1} = A_\star \phi(\mathbf{x}_t, \mathbf{u}_t)$. Then, Assumption 3 guarantees the existence of an input $\mathbf{u}_t \in \mathbb{B}^p_{b_u}$ such that

$$\begin{aligned} \|\mathbf{x}_{t+1} - \mathbf{x}^R_{t+1}\| &= \|A_\star \phi(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{w}_t - A_\star \phi(\mathbf{x}^R_t, \mathbf{u}^R_t)\| \\ &\leq \gamma \|\mathbf{x}_t - \mathbf{x}^R_t\| + \|\mathbf{w}_t\|. \end{aligned}$$

In other words, Assumption 3 allows us to find an input $\mathbf{u}_t$ such that the tracking error $\|\mathbf{x}_{t+1} - \mathbf{x}^R_{t+1}\|$ is upper bounded in terms of noise $\mathbf{w}_t$ and the tracking error at time $t$. By induction, Assumption 3 guarantees the existence of inputs to the system such that

$$\|\mathbf{x}_H - \mathbf{x}^R_H\| \leq \max_{t=0,\ldots,H-1} \|\mathbf{w}_t\| (1 + \gamma + \ldots + \gamma^{H-1}) + \gamma^H \|\mathbf{x}_0 - \mathbf{x}^R_0\|.$$

Hence, when $\gamma < 1$ we can choose a sequence of inputs such that the state $\mathbf{x}_H$ at time $H$ is close to $\mathbf{x}^R_H$, as long as the process noise is well behaved.

Note that in Assumption 3 we allow $\gamma \geq 1$. However, we pay a price when $\gamma$ is large. The larger $\gamma$ is the more stringent the following assumptions become. Finally, we note that the parameter $b_u$ appearing in Assumption 3 makes it easier for systems to satisfy the assumption than requiring that (3.3.3) holds for all $\mathbf{u}'$.

To estimate (3.3.1) we must collect measurements of state transitions from feature vectors that point in different directions. To ensure that such data can be collected from the system we must assume that there exist sequences of actions which take the dynamical system from a given state to some desired direction in feature space. This type of assumption can be formulated in terms of controllability. Recall that a linear system $\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t$ is said to be controllable when the matrix $\begin{bmatrix} B & AB & \ldots & A^{d-1}B \end{bmatrix}$ has full row rank. It can be easily checked that for a controllable linear system it is possible to get from any state to any other state in $d$ steps by appropriately choosing a sequence of inputs. Moreover, this notion of controllability can be extended to a class of nonlinear systems, called control affine systems, through the use of Lie brackets [129, 145]. We require, however, a different notion of controllability. In particular, we assume that in the absence of process noise we can take the system (3.3.1) from any state to a feature vector that aligns sufficiently with a desired direction in feature space.

**Assumption 4.** *There exist $\alpha$ and $H$, a positive real number and a positive integer, such that for any initial state $\mathbf{x}_0$ and goal vector $v \in \mathbb{S}^{k-1}$ there exists a sequence of actions $\mathbf{u}_t$, with $\|\mathbf{u}_t\| \leq b_u/2$, such that $|\langle \phi(\mathbf{x}_t, \mathbf{u}_t), v \rangle| \geq \alpha > 0$ for some $0 \leq t \leq H$, with $\mathbf{x}_{j+1} = A_\star \phi(\mathbf{x}_j, \mathbf{u}_j)$ for all $j$.*

If the assumption is satisfied for some horizon $H$, it is clear that it is also satisfied for larger horizons. Moreover, one expects that a larger horizon $H$ allows a larger controllability parameter $\alpha$. As discussed in the introduction, the larger $H$ is, the weaker our guarantee on estimation will be. However, the larger $\alpha$ is, the better our guarantee on estimation will be. Therefore, there is a tension between $\alpha$ and $H$ in our final result.

Assumptions 1 to 4 impose many constraints. Therefore, it is important to give examples of nonlinear dynamical systems that satisfy these assumptions. We give two simple examples. First we present a synthetic example for which it is easy to check that it satisfies all the assumptions, and then we discuss the simple pendulum.

**Example 3.3.3** (Smoothed Piecewise Linear System)**.** *When the support sets of the different pieces are known, piecewise affine systems can be easily expressed as* (3.3.1)*. However, the feature map $\phi$ would not be continuous. In this example, we present a smoothed version of a PWA system, which admits a 1-Lipschitz feature map. Let $f : \mathbb{R} \to \mathbb{R}$ be defined by*

$$f(x) = \begin{cases} 0 & \text{if } x < -1/2, \\ x + 1/2 & \text{if } x \in [-1/2, 1/2], \\ 1 & \text{if } x > 1/2. \end{cases}$$

*We also consider the maps $g(\mathbf{x}) = \frac{\mathbf{x}_t}{\|\mathbf{x}_t\|} \min\{\|\mathbf{x}_t\|, b_x\}$ and $h(\mathbf{u}) = \frac{\mathbf{u}}{\|\mathbf{u}\|} \min\{\|\mathbf{u}\|, r_u\}$, for some values $b_x$ and $r_u$. In this example both the inputs and the states are $d$-dimensional. Then, we define the feature map $\phi : \mathbb{R}^{2d} \to \mathbb{R}^{3d}$ as follows:*

$$\phi(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} g(\mathbf{x}) f(x_1) \\ g(\mathbf{x})(1 - f(x_1)) \\ h(\mathbf{u}) \end{bmatrix},$$

*where $x_1$ denotes the first coordinate of $\mathbf{x}$. Now, let us consider the following dynamical system:*

$$\mathbf{x}_{t+1} = \begin{bmatrix} A_1 & A_2 & I_d \end{bmatrix} \phi(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{w}_t, \tag{3.3.4}$$

*where $A_1$ and $A_2$ are two unknown $d \times d$ matrices. For the purpose of this example we can assume the noise $\mathbf{w}_t$ is zero almost surely.*

*To better understand the system* (3.3.4) *note that when $\|\mathbf{x}_t\| \le b_x$ and $\|\mathbf{u}_t\| \le r_u$ we have*

$$\begin{aligned} \mathbf{x}_{t+1} &= A_1 \mathbf{x}_t + \mathbf{u}_t \quad \text{if } x_{t1} \ge 1/2, \\ \mathbf{x}_{t+1} &= A_2 \mathbf{x}_t + \mathbf{u}_t \quad \text{if } x_{t1} \le -1/2. \end{aligned}$$

*By construction, the feature map of the system is 1-Lipschitz and bounded. Therefore,* (3.3.4) *satisfies Assumptions 1 and 2. We are left to show that we can choose $A_1$, $A_2$, $b_x$, and $r_u$ so that* (3.3.4) *satisfies Assumptions 3 and 4 as well.*

*It is easy to convince oneself that if $b_x > 2\sqrt{2}$, Assumptions 3 and 4 hold for any $A_1$ and $A_2$ as long as $r_u$ is sufficiently large relative to $A_1$, $A_2$, and $b_x$. In fact, if $r_u$ is sufficiently large, Assumption 3 is satisfied with $\gamma = 0$.* ■

**Example 3.3.4** (Simple Pendulum)**.** *The dynamics of a simple pendulum are described in continuous time by the equation*

$$m\ell^2\ddot{\theta}(t) + mg\ell\sin\theta(t) = -b\dot{\theta}(t) + u(t), \tag{3.3.5}$$

*where $\theta(t)$ is the angle of the pendulum at time $t$, $m$ is the mass of the pendulum, $\ell$ is its length, $b$ is a friction coefficient, and $g$ is the gravitational acceleration.*

*Discretizing (3.3.5) according to Euler's method*[3] *with step size $h$ and assuming stochastic process noise, we obtain the following two-dimensional system:*

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \begin{bmatrix} a_1 & a_2 \\ h & 0 \end{bmatrix} \begin{bmatrix} x_{t1} \\ \sin(x_{t2}) \end{bmatrix} + \begin{bmatrix} a_3 \\ 0 \end{bmatrix} \mathbf{u}_t + \mathbf{w}_t,$$

*where $x_{t1}$ and $x_{t2}$ are the coordinates of $\mathbf{x}_t$ and $a_1$, $a_2$, and $a_3$ are unknown real values. The first coordinate of $\mathbf{x}_t$ represents the angular velocity of the pendulum at time $t$, while the second coordinate represents the angle of the pendulum. Therefore, to put the inverted pendulum in the form of (3.3.2) we consider the feature map*

$$\phi(\mathbf{x}_t, \mathbf{u}_t) = \begin{bmatrix} x_{t1} \\ \sin(x_{t2}) \\ \mathbf{u}_t \end{bmatrix}.$$

*It can be easily checked that this feature map is 1-Lipschitz. While it is not bounded, if the pendulum experiences friction, we can ensure the feature values stay bounded by clipping the inputs $\mathbf{u}_t$; i.e., we replace $\mathbf{u}_t$ with $\operatorname{sgn}(\mathbf{u}_t)\min\{|\mathbf{u}_t|, r_u\}$ for some value $r_u$.*

*The simple pendulum satisfies Assumption 4 because we can drive the system in a finite number of steps from any state $\mathbf{x}_t$ to states $\mathbf{x}_{t+H}$ for which the signs of $x_{(t+h)1}$ and $\sin(x_{t1})$ can take any value in $\{-1, 1\}^2$, with their absolute values lower bounded away from zero.*

*Finally, Assumption 3 holds with $\gamma \geq 1 + h$. This assumption is pessimistic because the simple pendulum is stabilizable and can track reference trajectories. However, Assumption 3 does not hold with $\gamma < 1$ since the input at time $t$ does not affect the position at time $t+1$.* ■

We now turn to our final two assumptions. We need to make an assumption about the process noise and we also must assume access to an initial $\widehat{A}$ to warm start our method.

**Assumption 5.** *The random vectors $\mathbf{w}_t$ are independent, zero mean, and $\|\mathbf{w}_t\| \leq b_w$ almost surely*[4]*. Also, $\mathbf{w}_t$ is independent of $(\mathbf{x}_t, \mathbf{u}_t)$. Furthermore, we assume*

$$b_w \leq \frac{\alpha}{c_1 L(1 + \gamma + \ldots + \gamma^{H-1})}, \tag{3.3.6}$$

*for some universal constant $c_1 > 2$.*

[3] Using a more refined discretization method, such as a Runge-Kutta method, would be more appropriate. Unfortunately, such discretization methods yield a discrete-time system which cannot be easily put in the form (3.3.2).

[4] We can relax this assumption to only require $\mathbf{w}_t$ to be sub-Gaussian. In this case, we would make a truncation argument to obtain an upper bound on all $\mathbf{w}_t$ with high probability.

Equation 3.3.6 imposes on upper bound on the size of the process noise in terms of system-dependent quantities: the controllability parameter $\alpha$ introduced in Assumption 4, the Lipschitz constant $L$ of the feature map, and the control parameter $\gamma$ introduced in Assumption 3. An upper bound on $b_w$ is required because when the process noise is too large, it can be difficult to counteract its effects through feedback.

Finally, we assume access to an initial guess $\widehat{A}$ that is sufficiently close to $A_\star$. Namely, we require an initial estimate $\widehat{A}$ such that $\|\widehat{A} - A_\star\| = \mathcal{O}\left(L^{-1}(1+\gamma+\ldots+\gamma^{H-1})^{-1}\right)$. To understand the key issue this assumption resolves, suppose we are trying to track a reference trajectory $\{(\mathbf{x}_t^R, \mathbf{u}_t^R)\}_{t\geq 0}$ and $\|(\widehat{A} - A_\star)\phi(\mathbf{x}_t^R, \mathbf{u}_t^R)\|$ is large. Without an assumption on the size of $\|\widehat{A} - A_\star\|$, the magnitude of $(\widehat{A} - A_\star)\phi(\mathbf{x}_t^R, \mathbf{u}_t^R)$ might be large while $\|\phi(\mathbf{x}_t^R, \mathbf{u}_t^R)\|$ is small. Then, making a measurement at a point $(\mathbf{x}_t, \mathbf{u}_t)$ close to $(\mathbf{x}_t^R, \mathbf{u}_t^R)$ might not be helpful for estimation because $\phi(\mathbf{x}_t, \mathbf{u}_t)$ could be zero. Therefore, if $\|\widehat{A} - A_\star\|$ is too large, we might both fail to track a reference trajectory and to collect a useful measurement. For ease of exposition, instead of assuming access to an initial guess $\widehat{A}$, we assume access to a dataset.

**Assumption 6.** *We have access to an initial trajectory $\mathcal{D} = \{(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1})\}_{0\leq t<t_0}$ of transitions from the true system such that*

$$\lambda_{\min}\left(\sum_{t=0}^{t_0-1}\phi(\mathbf{x}_t,\mathbf{u}_t)\phi(\mathbf{x}_t,\mathbf{u}_t)^\top\right) \geq 1 + c_2 b_w^2 L^2 \left(\sum_{i=0}^{H-1}\gamma^i\right)^2 \left(d + k\log(b_\phi^2 T) + \log\left(\frac{\pi^2 T^2}{6\delta}\right)\right),$$

*where $c_2$ is a sufficiently large universal constant and $T$ is the number of samples to be collected by our method. We make explicit the requirement on $c_2$ in Section 3.3.4. In the full version of this work we show how to replace $T$ by a fixed quantity $T_\star$ [96].*

As shown in Section 3.3.3, Assumption 6 guarantees that the OLS estimate $\widehat{A}$ obtained from $\mathcal{D}$ satisfies $\|\widehat{A} - A_\star\| \leq \frac{c_3}{\sqrt{c_2}} L^{-1}(1+\gamma+\ldots+\gamma^{H-1})^{-1}$ for some universal constant $c_3$. Since the features $\phi(\mathbf{x}, \mathbf{u})$ can have magnitude as large as $b_\phi$, Assumption 6 only implies $\|(\widehat{A} - A_\star)\phi(\mathbf{x}, \mathbf{u})\| = \mathcal{O}(b_\phi L^{-1}(1+\gamma+\ldots+\gamma^{H-1})^{-1})$. Therefore, Assumption 6 does not imply a stringent upper bound on $\|(\widehat{A} - A_\star)\phi(\mathbf{x}, \mathbf{u})\|$ because $b_\phi$ can be arbitrarily large relative to $L$ and $\gamma$.

### 3.3.2 Main result

Our method for estimating the parameters of a dynamical system (3.3.1) is shown in Algorithm 3. The trajectory planning and tracking routines are discussed in detail below. We now state our main result.

**Theorem 3.3.5.** *Suppose $\mathbf{x}_{t+1} = A_\star\phi(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{w}_t$ is a nonlinear dynamical system which satisfies Assumptions 1-5 and suppose $\mathcal{D}$ is an initial trajectory that satisfies Assumption 6.*

**Algorithm 3** Active learning for nonlinear system identification

**Require:** Parameters: the feature map $\phi$, initial trajectory $\mathcal{D}$, and parameters $T$, $\alpha$, and $\beta$.

1: Initialize $\Phi$ to have rows $\phi(\mathbf{x}_j, \mathbf{u}_j)^\top$ and $Y$ to have rows $(\mathbf{x}_{j+1})^\top$, for $(\mathbf{x}_j, \mathbf{u}_j, \mathbf{x}_{j+1}) \in \mathcal{D}$.
2: Set $\widehat{A} \leftarrow Y^\top \Phi(\Phi^\top \Phi)^{-1}$; i.e., the OLS estimate according to $\mathcal{D}$.
3: Set $t \leftarrow t_0$.
4: **while** $t \le T + t_0$ **do**
5: Set $\mathbf{x}_0^R \leftarrow \mathbf{x}_t$,
6: Set $v$ to be a minimal eigenvector of $\Phi^\top \Phi$, with $\|v\| = 1$.
7: **Trajectory planning:** find inputs $\mathbf{u}_0^R$, $\mathbf{u}_1^R$, $\ldots$, $\mathbf{u}_r^R$, with $\|\mathbf{u}_j^R\| \le b_u$ and $r \le H$, such that

$$\left|\langle \phi(\mathbf{x}_r^R, \mathbf{u}_r^R), v\rangle\right| \ge \frac{\alpha}{2} \text{ or } \phi(\mathbf{x}_r^R, \mathbf{u}_r^R)^\top (\Phi^\top \Phi)^{-1} \phi(\mathbf{x}_r^R, \mathbf{u}_r^R) \ge \beta,$$

where $\mathbf{x}_{j+1}^R = \widehat{A}\phi(\mathbf{x}_j^R, \mathbf{u}_j^R)$ for all $j \in \{0, 1, \ldots, r-1\}$.
8: **Trajectory tracking:** track the reference trajectory $\{(\mathbf{x}_j^R, \mathbf{u}_j^R)\}_{j=0}^r$ and increment $t$ as described in the main text.
9: Set $\Phi^\top \leftarrow [\phi_0, \phi_1, \ldots, \phi_{t-1}]$ and $Y^\top \leftarrow [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t]$, where $(\phi_j, \mathbf{x}_{j+1})$ are all feature-state transitions observed so far.
10: **Re-estimate:** $\widehat{A} \leftarrow Y^\top \Phi(\Phi^\top \Phi)^{-1}$.
11: **end while**
12: Output the last estimate $\widehat{A}$.

*Also, let* $\beta = c_4 \left(d + k\log(\beta_\phi^2 T) + \log(\pi^2 T^2/(6\delta))\right)^{-1}$ *with* $c_4 \le \frac{(c_1-2)^2}{36c_3^2}$ *and let:*[5]

$$N_e := \left\lceil \frac{2k \log\left(\frac{2kb_\phi^2}{\log(1+\beta/2)}\right)}{\log(1+\beta/2)} \right\rceil.$$

*Then, with probability* $1-\delta$*, and given parameters* $T$ *and* $\beta$*, Algorithm 3 outputs* $\widehat{A}$ *such that*

$$\|\widehat{A} - A_\star\| \le c_5 \frac{b_w}{\alpha} \sqrt{\frac{d + k\log(b_\phi^2 T) + \log\left(\frac{\pi^2 T}{6\delta}\right)}{T/H - N_e}},$$

*whenever* $T \ge \frac{32kb_\phi^2 H}{\alpha^2} + HN_e$.

There are several aspects of this result worth emphasizing. First, the statistical rate we obtain in Theorem 3.3.5 has the same form as the standard statistical rate for linear

[5]Recall that $c_1$ is the universal constant appearing in Assumption 4 and $c_3$ is the universal constant appearing in the upper bound on the error of the OLS estimate, shown in Section 3.3.3.

regression, which is $\mathcal{O}\left(b_w\sqrt{\frac{k}{T}}\right)$. The two important distinctions are the dependence on the planning horizon $H$ and the controllability term $\alpha$, both of which are to be expected in our case. Algorithm 3 uses trajectory planning for data collection and the length of the reference trajectories is at most $H$. Since we can only guarantee one useful measurement per reference trajectory, it is to be expected that we can only guarantee an effective sample size of $T/H$. The controllability term $\alpha$ is also natural in our result because it quantifies how large the feature vectors can become in different directions. Larger feature vectors imply a larger signal-to-noise ratio, which in turn implies faster estimation.

**Trajectory planning.** The trajectory planning routine shown in Algorithm 3 uses the current estimate $\widehat{A}$ to plan, assuming no process noise, a trajectory from the current state of the system $\mathbf{x}_0^R = \mathbf{x}_t$ to a high-uncertainty region of the feature space, assuming no process noise. More precisely, it finds a sequence of actions $\{\mathbf{u}_j^R\}_{j=0}^r$ which produces a sequence of reference states $\{\mathbf{x}_j^R\}_{j=0}^r$ with the following properties:

- $\mathbf{x}_{j+1}^R = \widehat{A}\phi(\mathbf{x}_j^R, \mathbf{u}_j^R)$.
- The last reference state-action pair $(\mathbf{x}_r^R, \mathbf{u}_r^R)$ is either well aligned with $v$, the minimum eigenvector of $\Phi^\top\Phi$, or its feature vector is in a high-uncertainty region of the state space. More precisely, $(\mathbf{x}_r^R, \mathbf{u}_r^R)$ must satisfy one of the following two inequalities:

$$\left|\langle\phi(\mathbf{x}_r^R, \mathbf{u}_r^R), v\rangle\right| \geq \frac{\alpha}{2} \text{ or } \phi(\mathbf{x}_r^R, \mathbf{u}_r^R)^\top(\Phi^\top\Phi)^{-1}\phi(\mathbf{x}_r^R, \mathbf{u}_r^R) \geq \beta.$$

It is not immediately obvious that we can always find such a sequence of inputs. In Section 3.3.4 we prove that when Assumptions 4 and 6 hold the trajectory planning problem is feasible.

From the study of OLS, discussed in Section 3.3.3, we know that the matrix $\Phi^\top\Phi$ determines the uncertainty set of OLS. The larger $\lambda_{\min}\left(\Phi^\top\Phi\right)$ is, the smaller the uncertainty set will be. Therefore, to reduce the size of the uncertainty set we want to collect measurements at feature vectors $\phi$ such that the smallest eigenvalues of $\Phi^\top\Phi + \phi\phi^\top$ are larger than the smallest eigenvalues of $\Phi^\top\Phi$. Ideally, $\phi$ is a minimal eigenvector of $\Phi^\top\Phi$. However, we cannot always drive the system to such a feature vector, especially in the presence of process noise.

Instead, we settle for feature vectors of the following two types. Firstly, the trajectory planner tries to drive the system to feature vectors $\phi$ that are well aligned with the minimal eigenvector $v$ of $\Phi^\top\Phi$; i.e., such that $|\langle\phi, v\rangle| \geq \alpha$. Such a data collection scheme is an instance of E-optimal design [114], which has been shown by Wagenmaker and Jamieson [160] to produce inputs that allow the estimation of linear dynamics at an optimal rate.

However, if reaching a feature vector that aligns with the minimal eigenvector is not possible, the trajectory planner finds a reference trajectory to a feature vector $\phi$ such that $\phi^\top(\Phi^\top\Phi)^{-1}\phi \geq \beta$. When this inequality holds our uncertainty about the estimate $\widehat{A}$ in the direction $\phi$ is large. As shown in Section 3.3.4, such feature vectors can be encountered for only a small number of iterations.

Finally, trajectory planning is computationally intractable in general. However, in this work we quantify the data requirements of identifying $A_\star$, leaving computational considerations for future work. We assume access to a computational oracle. This assumption is reasonable since trajectory planning is often solved successfully in practice [75, 83, 170].

**Trajectory tracking.** Now we detail the trajectory tracking component of our method. We saw that the trajectory planner produces a reference trajectory $\{(\mathbf{x}_j^R, \mathbf{u}_j^R)\}_{j=0}^r$, with $r \leq H$. However, the planner assumes no process noise to generate this reference trajectory. Therefore, if we were to simply plug the sequence of actions $\{\mathbf{u}_j^R\}_{j=0}^r$ into (3.3.1), the states of the system would diverge from $\mathbf{x}_j^R$. Instead, after observing each state $\mathbf{x}_t$ of the system (3.3.1), our method chooses an input $\mathbf{u}_t$ as follows:

- Given the current state $\mathbf{x}_t$, our method chooses an input $\mathbf{u}_t$ such that

$$\phi(\mathbf{x}_t, \mathbf{u}_t)^\top (\Phi^\top \Phi)^{-1} \phi(\mathbf{x}_t, \mathbf{u}_t) \geq \beta,$$

  if there exists such an input. In other words, if there is an opportunity to greedily collect an informative measurement, our method takes it. If this situation is encountered, the trajectory tracker increments $t$ by 1 and then stops tracking and returns.

- If there is no opportunity for greedy exploration, our method chooses an input $\mathbf{u}_t$ that minimizes $\|\widehat{A}(\phi(\mathbf{x}_t, \mathbf{u}_t) - \phi(\mathbf{x}_j^R, \mathbf{u}_j^R))\|$, and then increments $t$ and $j$ by one ($t$ indexes the time steps of the system (3.3.1) and $j$ indexes the reference trajectory). Therefore, our method uses closed loop control for data generation since minimizing $\|\widehat{A}(\phi(\mathbf{x}_t, \mathbf{u}_t) - \phi(\mathbf{x}_j^R, \mathbf{u}_j^R))\|$ requires access to the current state $\mathbf{x}_t$. At time $t$ we choose $\mathbf{u}_t$ in this fashion in order to minimize the tracking error $\mathbb{E}\|\mathbf{x}_{t+1} - \mathbf{x}_{t+1}^R\|^2$ at the next time step, where the expectation is taken with respect to $\mathbf{w}_t$.

- Our method repeats these steps until $j = r$; i.e., until it reaches the end of the reference trajectory. When $j = r$ the trajectory tracker sets $\mathbf{u}_t = \mathbf{u}_j^R$, increments $t$ by one, and returns.

### 3.3.3 General guarantee on estimation

In this section we provide a general upper bound on the error between an OLS estimate $\widehat{A}$ and the true parameters $A_\star$. The guarantee is based on the work of Simchowitz et al. [141]. We note also that results of this kind have been previously used in the study of online least squares and linear bandits [2, 39, 123]. We assume that we are given a sequence of observations $\{(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1})\}_{t \geq 0}$, generated by the system (3.3.1), with $u_t$ allowed to depend on $\mathbf{x}_0$, $\mathbf{x}_1$, ..., $\mathbf{x}_{t-1}$ and independent of $\mathbf{w}_j$ for all $j \geq t$. In what follows we denote $\phi_t := \phi(\mathbf{x}_t, \mathbf{u}_t)$.

Our method re-estimates the parameters $A_\star$ as more data is being collected. For the purpose of this section let us denote by $\widehat{A}_j$ the OLS estimate obtained using the first $j$

measurements $(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1})$:

$$\widehat{A}_j = \arg\min_A \sum_{t=0}^{j-1} \|A\phi_t - \mathbf{x}_{t+1}\|^2. \tag{3.3.7}$$

**Proposition 3.3.6.** *If the system (3.3.1) satisfies Assumptions 5 and 2 and if* $\lambda_{\min}\left(\sum_{t=0}^{t_0-1} \phi_t\phi_t^\top\right) \geq \underline{\lambda}$, *for some* $\underline{\lambda} > 0$ *and* $t_0 > 0$, *the OLS estimates (3.3.7) satisfy*

$$\mathbb{P}\left[\exists u \in \mathbb{S}^{k-1} \text{ and } j \geq t_0 \text{ s.t. } \|(\widehat{A}_j - A_\star)u\| \geq \mu_j \sqrt{u^\top \left(\sum_{t=0}^{j-1} \phi_t\phi_t^\top\right)^{-1} u}\right] \leq \delta,$$

*where* $\mu_j = c_3 b_w \sqrt{d + k\log\left(\frac{b_\phi^2 j}{\underline{\lambda}}\right) + \log\left(\frac{\pi^2 j^2}{6\delta}\right)}$ *for some universal constant* $c_3$.

*Proof.* By assumption $\lambda_{\min}\left(\sum_{t=0}^{t_0-1} \phi_t\phi_t^\top\right) \geq \underline{\lambda} > 0$. Therefore, $\sum_{t=0}^{j-1} \phi_t\phi_t^\top$ is invertible and

$$\widehat{A}_j - A_\star = W_j^\top \Phi_j (\Phi_j^\top \Phi_j)^{-1},$$

where $W_j^\top = [\mathbf{w}_0, \ldots, \mathbf{w}_{j-1}]$ and $\Phi_j^\top = [\phi_0, \ldots, \phi_{j-1}]$. Now, we fix the index $j$ and we consider the SVD decomposition $\Phi_j = U\Sigma V^\top$. Therefore, $\widehat{A}_j - A_\star = W_j^\top U \Sigma^\dagger V^\top$.

Recall that $\sup_{\mathbf{x},\mathbf{u}} \|\phi(\mathbf{x},\mathbf{u})\|_2 \leq b_\phi$ by assumption. Then, according to the analysis of Simchowitz et al. [141] we know that $\|W_j^\top U\| \leq \mu_j$ with probability at least $1 - 6\delta/(\pi^2 j^2)$. Note that for all $u \in \mathbb{S}^{k-1}$ we have

$$\begin{aligned}\|(\widehat{A}_j - A_\star)u\| &\leq \|W_j^\top U\| \|\Sigma^\dagger V^\top u\| = \|W_j^\top U\| \sqrt{u^\top V (\Sigma^\dagger)^\top \Sigma^\dagger V^\top u} \\ &= \|W_j^\top U\| \sqrt{u^\top (\Phi_j^\top \Phi_j)^{-1} u}.\end{aligned}$$

Therefore, for a fixed index $j$, we have

$$\mathbb{P}\left[\exists u \in \mathbb{S}^{k-1} \text{ s.t. } \|(\widehat{A}_j - A_\star)u\|_2 \geq \mu_j \sqrt{u^\top \left(\sum_{t=0}^{j-1} \phi_t\phi_t^\top\right)^{-1} u}\right] \leq \frac{6\delta}{\pi^2 j^2}.$$

A direct application of the union bound yields the desired conclusion. □

### 3.3.4 Proof of the main result

First let us observe that when $b_w = 0$ the result is trivial. Because we assume access to an initial trajectory $\mathcal{D}$ which satisfies Assumption 6 we are guaranteed $\widehat{A} = A_\star$ when $b_w = 0$. Therefore, we can assume that $b_w > 0$, which implies that $\alpha$ must be strictly positive according to Assumption 4. Throughout the proof we denote $\phi_t := \phi(\mathbf{x}_t, \mathbf{u}_t)$ and $\phi_j^R := \phi(\mathbf{x}_j^R, \mathbf{u}_j^R)$.

The proof of our result has three parts, which we now outline:

- We show that the trajectory planning step in Algorithm 3 is always feasible.
- We show that during the execution of Algorithm 3 there are at most $N_e$ iterations for which there exist $t$ and $j$ with:

$$\max_{\mathbf{u}\in\mathbb{B}_{r_u}} \phi(\mathbf{x}_t,\mathbf{u})^\top(\Phi^\top\Phi)^{-1}\phi(\mathbf{x}_t,\mathbf{u}) \geq \beta \quad \text{or} \quad (\phi_j^R)^\top(\Phi^\top\Phi)^{-1}\phi_j^R \geq \beta.$$

- We show that Algorithm 3 collects at least $T/H - N_e$ measurements $(\phi_t, \mathbf{x}_{t+1})$ such that $|\langle \phi_t, v\rangle| \geq \alpha/4$, where $v$ is a minimal eigenvector used to plan the reference trajectories. As a consequence, we show that Algorithm 3 collects measurements $(\phi_t, \mathbf{x}_{t+1})$ such that

$$\lambda_{\min}\left(\sum_{t=1}^{T+t_0}\phi_t\phi_t^\top\right) \geq \mathcal{O}(1)\alpha^2\left(\frac{T}{H}-N_e\right) - \frac{k-1}{2}b_\phi^2. \tag{3.3.8}$$

Once we have shown (3.3.8) is true, Theorem 3.3.5 follows from Proposition 3.3.6 and some algebra.

**Part 1 of the Proof of Theorem 3.3.5.** We show that the trajectory planning step of Algorithm 3 is always feasible. Let

$$\mu = c_3 b_w\sqrt{d + k\log\left(b_\phi^2 T\right) + \log\left(\frac{\pi^2T^2}{6\delta}\right)},$$

where $c_3$ is the universal constant appearing in Proposition 3.3.6. Since Assumption 6 guarantees that the minimum eigenvalue of the design matrix is at least 1, we know that

$$\|(\widehat{A}-A_\star)\phi\| \leq \mu\sqrt{\phi^\top\left(\Phi^\top\Phi\right)^{-1}\phi}, \tag{3.3.9}$$

for all $\phi \in \mathbb{S}^{k-1}$ and all iterations of Algorithm 3 with probability $1-\delta$.

Now, let $\beta = c_4\left(d + k\log(\beta_\phi^2 T) + \log(\pi^2T^2/(6\delta))\right)^{-1}$ with $c_4 \leq c_1^2/(4c_3^2)$. Then, since $\alpha \geq c_1 L b_w(1+\gamma+\ldots+\gamma^{H-1})$, we have

$$\beta \leq \left(\frac{\alpha}{2L(1+\gamma+\ldots+\gamma^{H-1})\mu}\right)^2. \tag{3.3.10}$$

Let us $\tilde{\mathbf{x}}_0$ be equal to the initial state $\mathbf{x}_0^R$ of the trajectory planning and let $v \in \mathbb{R}^k$ be the desired goal direction. By Assumption 4 we know that there must exist a sequence of inputs $\tilde{\mathbf{u}}_0$, $\tilde{\mathbf{u}}_1$, …, $\tilde{\mathbf{u}}_r$, with $r \leq H$ and $\|\tilde{\mathbf{u}}_j\| \leq b_u/2$, such that $|\langle\phi(\tilde{\mathbf{x}}_r,\tilde{\mathbf{u}}_r), v\rangle| \geq \alpha$, where $\tilde{\mathbf{x}}_{j+1} = A_\star\phi(\tilde{\mathbf{x}}_j,\tilde{\mathbf{u}}_j)$. Now, let $\mathbf{x}_{j+1}^R = \widehat{A}\phi(\mathbf{x}_j^R,\mathbf{u}_j^R)$, where $\mathbf{u}_j^R$ is any input vector with $\|\mathbf{u}_j^R\| \leq b_u$ such that

$$\|A_\star[\phi(\mathbf{x}_j^R,\mathbf{u}_j^R) - \phi(\tilde{\mathbf{x}}_j,\tilde{\mathbf{u}}_j)]\| \leq \gamma\|\mathbf{x}_j^R - \tilde{\mathbf{x}}_j\|, \tag{3.3.11}$$

for $j < r$. Assumption 4 guarantees the existence of $\mathbf{u}_j^R$. We set $\mathbf{u}_r^R = \tilde{\mathbf{u}}_r$ and denote $\tilde{\phi}_j = \phi(\tilde{\mathbf{x}}_j, \tilde{\mathbf{u}}_j)$ and $\phi_j^R = \phi(\mathbf{x}_j^R, \mathbf{u}_j^R)$.

*Case 1.* There exists $j \in \{0, 1, 2, \ldots, r\}$ such that $(\phi_j^R)^\top \left(\Phi^\top \Phi\right)^{-1} \phi_j^R \geq \beta$. If this is the case, we are done because we found a feasible sequence of inputs $\mathbf{u}_0^R$, $\mathbf{u}_1^R$, $\ldots$, $\mathbf{u}_j^R$.

*Case 2.* We have $(\phi_j^R)^\top \left(\Phi^\top \Phi\right)^{-1} \phi_j^R \leq \beta$ for all $j \in \{0, 1, 2, \ldots, r\}$. In this case, we have

$$\tilde{\mathbf{x}}_{j+1} - \mathbf{x}_{j+1}^R = A_\star \tilde{\phi}_j - \widehat{A}\phi_j^R = A_\star(\tilde{\phi}_j - \phi_j^R) + (A_\star - \widehat{A})\phi_j^R.$$

Therefore, using (3.3.9), (3.3.10), and (3.3.11) we find

$$\begin{aligned}\|\tilde{\mathbf{x}}_{j+1} - \mathbf{x}_{j+1}^R\| &\leq \|A_\star(\tilde{\phi}_j - \phi_j^R)\| + \|(A_\star - \widehat{A})\phi_j^R\| \\ &\leq \gamma\|\tilde{\mathbf{x}}_j - \mathbf{x}_j^R\| + \frac{\alpha}{2L(1 + \gamma + \ldots + \gamma^{H-1})}.\end{aligned}$$

Applying this inequality recursively, we find $\|\tilde{\mathbf{x}}_r - \mathbf{x}_r^R\| \leq \frac{\alpha}{2L}$, which implies $|\langle \phi_r^R, v\rangle| \geq \alpha/2$ because $\|\phi_r^R - \tilde{\phi}_r\| \leq L\|\tilde{\mathbf{x}}_r - \mathbf{x}_r^R\|$ by Assumption 1 and $|\langle \tilde{\phi}_r, v\rangle| \geq \alpha$ by construction. Hence, we constructed a feasible sequence of inputs $\{\mathbf{u}_j\}_{j=0}^r$ and Part 1 of the proof is complete.

**Part 2 of the Proof of Theorem 3.3.5.** Now, we show that the number of iterations for which Algorithm 3 satisfies

$$\max_{\mathbf{u} \in \mathbb{B}_{r_u}} \phi(\mathbf{x}_t, \mathbf{u})^\top (\Phi^\top \Phi)^{-1} \phi(\mathbf{x}_t, \mathbf{u}) \geq \beta \quad \text{or} \quad (\phi_j^R)^\top (\Phi^\top \Phi)^{-1} \phi_j^R \geq \beta \tag{3.3.12}$$

is upper bounded by

$$N_e := \left\lceil \frac{2k \log\left(\frac{2kb_\phi^2}{\log(1+\beta/2)}\right)}{\log(1 + \beta/2)} \right\rceil. \tag{3.3.13}$$

We rely on the following proposition.

**Proposition 3.3.7.** *Let $M_0$ be a positive definite matrix and let us consider a sequence of vectors $\{v_t\}_{t \geq 1}$ in $\mathbb{R}^k$ with $\max_{t \geq 1}\|v_t\| \leq b$. Then, the number of vectors $v_{t+1}$ such that*

$$v_{t+1}^\top \left(M_0 + \sum_{i=1}^{t} v_i v_i^\top\right)^{-1} v_{t+1} \geq \beta,$$

*is upper bounded by*

$$\left\lceil \frac{2k \log\left(\frac{2kb^2}{\lambda_k(M_0)\log(1+\beta)}\right)}{\log(1 + \beta)} \right\rceil. \tag{3.3.14}$$

*Proof.* First we relate the scaling of ellipsoids in one direction with the scaling of their volumes. Namely, if $M$ and $N$ are two positive definite matrices with $M \succ N \succ 0$, then

$$\sup_{v \neq 0} \frac{v^\top M v}{v^\top N v} \leq \frac{\det(M)}{\det(N)}. \tag{3.3.15}$$

A proof of this fact can be found in the work by Abbasi-Yadkori et al. [2].

Now, we are ready to prove Proposition 3.3.7. We denote by $N_t = N_0 + \sum_{i=1}^t v_i v_i^\top$. First, we prove that $\det(N_{t+1}^{-1}) \leq \det(N_t^{-1})/(1+\beta)$ whenever $v_{t+1}^\top N_t^{-1} v_{t+1} \geq \beta$.

By definition we have $N_{t+1} \succeq N_t \succ 0$. Therefore, $N_{t+1}^{-1} \preceq N_t^{-1}$. Now, we apply the Sherman-Morrison rank-one update formula to find

$$\begin{aligned} v_{t+1}^\top N_{t+1}^{-1} v_{t+1} &= v_{t+1}^\top N_t^{-1} v_{t+1} - \frac{\left(v_{t+1}^\top N_t^{-1} v_{t+1}\right)^2}{1 + v_{t+1}^\top N_t^{-1} v_{t+1}} \\ &= \left(1 - \frac{v_{t+1}^\top N_t^{-1} v_{t+1}}{1 + v_{t+1}^\top N_t^{-1} v_{t+1}}\right) v_{t+1}^\top N_t^{-1} v_{t+1}. \end{aligned}$$

Since the function $x \mapsto \frac{x}{1+x}$ is increasing for $x > -1$, we find

$$v_{t+1}^\top N_{t+1}^{-1} v_{t+1} \leq \frac{v_{t+1}^\top N_t^{-1} v_{t+1}}{1+\beta},$$

whenever $v_{t+1}^\top N_t^{-1} v_{t+1} \geq \beta$. Then, (3.3.15) implies that $\det(N_{t+1}^{-1}) \leq \det(N_t^{-1})/(1+\beta)$ whenever $v_{t+1}^\top N_t^{-1} v_{t+1} \geq \beta$, which in turn implies $\det(N_{t+1}) \geq (1+\beta)\det(N_t)$ whenever $v_{t+1}^\top N_t^{-1} v_{t+1} \geq \beta$.

Let us denote by $\lambda_1(t), \lambda_2(t), \ldots, \lambda_k(t)$ the eigenvalues of $N_t$ sorted in decreasing order. Recall that $\lambda_i(t)$ is a non-decreasing function of $t$. Now, let $\varepsilon_{i,t} = \log_{1+\beta}(\lambda_i(t)/\lambda_i(t-1))$. Therefore, we have $\lambda_i(t) = (1+\beta)^{\varepsilon_{i,t}} \lambda_i(t-1)$. We know $\varepsilon_{i,t} \geq 0$ for all $i$ and $t$ and we know that $\sum_{i=1}^k \varepsilon_{i,t} \geq 1$ when $v_t^\top N_{t-1}^{-1} v_t \geq \beta$ because $\det(N_{t+1}) \geq (1+\beta)\det(N_t)$.

By definition, we have $\lambda_i(t) = (1+\beta)^{\sum_{j=1}^t \varepsilon_{i,j}} \lambda_i(N_0) \geq (1+\beta)^{\sum_{j=1}^t \varepsilon_{i,j}} \lambda_k(N_0)$. Since $\max_j \|v_j\| \leq b$, we know that $\lambda_i(t+1) \leq \lambda_i(t) + b^2$. Therefore,

$$(1+\beta)^{\varepsilon_{i,t+1}} = \frac{\lambda_i(t+1)}{\lambda_i(t)} \leq 1 + \frac{b^2}{\lambda_i(t)} \leq 1 + \frac{b^2}{(1+\beta)^{\sum_{j=1}^t \varepsilon_{i,j}} \lambda_k(N_0)}.$$

In other words, we have

$$\varepsilon_{i,t+1} \leq \frac{\log\left(1 + \frac{b^2}{(1+\beta)^{\sum_{j=1}^t \varepsilon_{i,j}} \lambda_k(N_0)}\right)}{\log(1+\beta)} \leq \frac{b^2}{(1+\beta)^{\sum_{j=1}^t \varepsilon_{i,j}} \lambda_k(N_0) \log(1+\beta)}.$$

Therefore, when $\sum_{j=1}^t \varepsilon_{i,j} > \log\left(\frac{2kb^2}{\lambda_k(N_0)\log(1+\beta)}\right)/\log(1+\beta)$, we have $\varepsilon_{i,t+1} \leq 1/(2k)$. We denote $\rho = \log\left(\frac{2kb^2}{\lambda_k(N_0)\log(1+\beta)}\right)/\log(1+\beta)$.

Suppose there are $n$ vectors $v_j$ such that $v_j^\top N_{j-1}^{-1} v_j \geq \beta$ with $j \leq t$. Since $\sum_{i=1}^k \varepsilon_{i,j} \geq 1$ whenever $v_j^\top N_{j-1}^{-1} v_j \geq \beta$, we have $\sum_{j=1}^t \sum_{i=1}^k \varepsilon_{i,j} \geq n$. Moreover, at each time $j$ with $v_j^\top N_{j-1}^{-1} v_j \geq \beta$ we know that

$$\begin{aligned}\varepsilon_{i,j} &\geq 1 - \sum_{i' \neq i} \varepsilon_{i',j} \geq 1 - \sum_{i' : \sum_{s=1}^{j-1} \varepsilon_{i',s} < \rho} \varepsilon_{i',j} - \sum_{i' : \sum_{s=1}^{j-1} \varepsilon_{i',s} \geq \rho} \varepsilon_{i',j} \\ &\geq 1 - \sum_{i' : \sum_{s=1}^{j-1} \varepsilon_{i',s} < \rho} \varepsilon_{i',j} - \sum_{i' : \sum_{s=1}^{j-1} \varepsilon_{i',s} \geq \rho} \frac{1}{2k} \geq \frac{1}{2} - \sum_{i' : \sum_{s=1}^{j-1} \varepsilon_{i',s} < \rho} \varepsilon_{i',j}.\end{aligned}$$

Summing these inequalities over $j$, for any $i$ we have

$$\begin{aligned}\sum_{j=1}^t \varepsilon_{i,j} &\geq \frac{n}{2} - \sum_{i' \neq i} \min \left\{ \log\left( \frac{2kb^2}{\lambda_k(N_0) \log(1+\beta)} \right) / \log(1+\beta), \sum_{j=1}^t \varepsilon_{i',j} \right\} \\ &\geq \frac{n}{2} - (k-1) \log\left( \frac{2kb^2}{\lambda_k(N_0) \log(1+\beta)} \right) / \log(1+\beta).\end{aligned}$$

Then, once $n \geq 2k \log\left( \frac{2kb^2}{\lambda_k(N_0) \log(1+\beta)} \right) / \log(1+\beta)$, we obtain

$$\sum_{j=1}^t \varepsilon_{i,j} \geq \log\left( \frac{2kb^2}{\lambda_k(N_0) \log(1+\beta)} \right) / \log(1+\beta),$$

which implies $\varepsilon_{i,j} < \frac{1}{2k}$ for all $j > t$. Since $i$ was chosen arbitrary, we see that whenever $n \geq 2k \log\left( \frac{2kb^2}{\lambda_k(N_0) \log(1+\beta)} \right)$ and $j > t$ we get $\sum_{i=1}^k \varepsilon_{ij} < k \frac{1}{2k} < 1$. Hence, $n$ must be smaller or equal than $\left\lceil 2k \log\left( \frac{2kb^2}{\lambda_k(N_0) \log(1+\beta)} \right) \right\rceil$. □

Given Proposition 3.3.7, to prove (3.3.13) it suffices to show that during each iteration of Algorithm 3 when (3.3.12) occurs our method collects a measurement $(\phi_t, \mathbf{x}_{t+1})$ such that $\phi_t^\top (\Phi^\top \Phi)^{-1} \phi_t \geq \beta/2$.

By the definition of our trajectory tracker, whenever $\sup_{\mathbf{u}} \phi(\mathbf{x}_t, \mathbf{u})^\top (\Phi^\top \Phi)^{-1} \phi(\mathbf{x}_t, \mathbf{u}) \geq \beta$ we collect a measurement $(\phi_t, \mathbf{x}_{t+1})$ such that $\phi_t^\top (\Phi^\top \Phi)^{-1} \phi_t \geq \beta$.

Next, we show that when $(\phi_j^R)^\top (\Phi^\top \Phi)^{-1} \phi_j^R \geq \beta$, for some $j \leq r$, Algorithm 3 is guaranteed to collect a measurement $(\phi_t, \mathbf{x}_{t+1})$ such that $\phi_t^\top (\Phi^\top \Phi)^{-1} \phi_t \geq \beta/2$. Let $s$ be the smallest index in the reference trajectory such that $(\phi_s^R)^\top (\Phi^\top \Phi)^{-1} \phi_s^R \geq \beta$.

For the remainder of this section we re-index the trajectory $\{(\mathbf{x}_t, \mathbf{u}_t)\}_{t \geq 0}$ collected by Algorithm 3 so that $\mathbf{x}_j^R = \mathbf{x}_j$ for all $j \in \{0, 1, \ldots, s\}$. Then, we show that $(\phi_s^R)^\top (\Phi^\top \Phi)^{-1} \phi_s^R \geq \beta$ implies the existence of $j \in \{0, 1, \ldots, s\}$ such that $\phi_j^\top (\Phi^\top \Phi)^{-1} \phi_j \geq \beta/2$.

Let $\Delta = \phi_s^R - \phi_s$. The Cauchy-Schwarz inequality implies

$$\begin{aligned}\phi_s (\Phi^\top \Phi)^{-1} \phi_s &= \phi_s^R (\Phi^\top \Phi)^{-1} \phi_s^R + \Delta^\top (\Phi^\top \Phi)^{-1} \Delta + 2\Delta^T (\Phi^\top \Phi)^{-1} \phi_s^R \\ &\geq \left( \sqrt{\phi_s^R (\Phi^\top \Phi)^{-1} \phi_s^R} - \sqrt{\Delta^\top (\Phi^\top \Phi)^{-1} \Delta} \right)^2.\end{aligned}$$

Then, as long as $\Delta^\top(\Phi^\top\Phi)^{-1}\Delta \leq \frac{\beta}{2}(3-2\sqrt{2})$, we are guaranteed to have $\phi_s^\top(\Phi^\top\Phi)^{-1}\phi_s \geq \beta/2$.

Now, since $s$ is the smallest index such that $(\phi_s^R)^\top(\Phi^\top\Phi)^{-1}\phi_s^R \geq \beta$, we know that for all $j \in \{0,1,\ldots,s-1\}$ we have $(\phi_j^R)^\top(\Phi^\top\Phi)^{-1}\phi_j^R \leq \beta$. Also, we can assume that during reference tracking we do not encounter a state $\mathbf{x}_j$, with $j \in \{0,1,\ldots,s-1\}$, such that

$$\max_{\mathbf{u}\in\mathbb{B}_{r_u}} \phi(\mathbf{x}_j,\mathbf{u})^\top(\Phi^\top\Phi)^{-1}\phi(\mathbf{x}_t,\mathbf{u}) \geq \beta,$$

because we already treated this case. Now, let us consider the difference

$$\mathbf{x}_{j+1} - \mathbf{x}_{j+1}^R = A_\star\phi_j + \mathbf{w}_j - \widehat{A}\phi_j^R = (A_\star - \widehat{A})\phi_j + \mathbf{w}_j - \widehat{A}[\phi_j^R - \phi_j].$$

We obtain

$$\|\mathbf{x}_{j+1} - \mathbf{x}_{j+1}^R\| \leq \|(A_\star - \widehat{A})\phi_j\| + b_w + \|\widehat{A}[\phi_j^R - \phi_j]\|.$$

Let us denote $\delta_j(\mathbf{u}) = \phi(\mathbf{x}_j,\mathbf{u}) - \phi_j^R$. Hence, $\delta_j(\mathbf{u}_j) = \phi_j - \phi_j^R$. Now, let $\mathbf{u}_\star \in \mathbb{B}_{r_u}$ an input such that $\|A_\star\delta_t(\mathbf{u}_\star)\| \leq \gamma\|\mathbf{x}_j - \mathbf{x}_j^R\|$, which we know exists by Assumption 3 (note that $\mathbf{u}_\star$ depends on the index $j$, but we dropped this dependency from the notation for simplicity). Since our method attempts trajectory tracking by choosing $\mathbf{u}_j \in \arg\min_{\mathbf{u}\in\mathbb{B}_{r_u}}\|\widehat{A}(\phi(\mathbf{x}_t,\mathbf{u}) - \phi(\mathbf{x}_j^R,\mathbf{u}_j^R))\|$ we have

$$\begin{aligned}
\|\widehat{A}\delta_j(\mathbf{u}_j)\| &\leq \|\widehat{A}\delta_j(\mathbf{u}_\star)\| \leq \|A_\star\delta_j(\mathbf{u}_\star)\| + \|(A_\star - \widehat{A})\delta_j(\mathbf{u}_\star)\| \\
&\leq \gamma\|\mathbf{x}_j - \mathbf{x}_j^R\| + \|(A_\star - \widehat{A})\delta_j(\mathbf{u}_\star)\| \\
&\leq \gamma\|\mathbf{x}_j - \mathbf{x}_j^R\| + \|(A_\star - \widehat{A})\phi(\mathbf{x}_j,\mathbf{u}_\star)\| + \|(A_\star - \widehat{A})\phi_j^R\|.
\end{aligned}$$

As mentioned above, we can assume $\phi(\mathbf{x}_j,\mathbf{u}_\star)^\top(\Phi^\top\Phi)^{-1}\phi(\mathbf{x}_j,\mathbf{u}_\star) < \beta$. Also, recall that

$$(\phi_j^R)^\top(\Phi^\top\Phi)^{-1}\phi_j^R \leq \beta,$$

since $j < s$ and $s$ is the smallest index so that this inequality does not hold. Hence, Proposition 3.3.6 implies that $\|(A_\star - \widehat{A})\phi(\mathbf{x}_t,\mathbf{u}_\star)\| \leq \mu\sqrt{\beta}$ and $\|(A_\star - \widehat{A})\phi_r^R\| \leq \mu\sqrt{\beta}$. Putting everything together we find

$$\|\mathbf{x}_{j+1} - \mathbf{x}_{j+1}^R\| \leq \gamma\|\mathbf{x}_j - \mathbf{x}_j^R\| + 3\mu\sqrt{\beta} + b_w.$$

Then, since the reference trajectory is initialized with the state $\mathbf{x}_0^R = \mathbf{x}_0$, we find

$$\begin{aligned}
\|\Delta\| = \|\phi_s - \phi_s^R\| &\leq L(b_w + 3\mu\sqrt{\beta})(1+\gamma+\ldots+\gamma^{s-1}) \\
&= (3c_3\sqrt{c_4}+1)Lb_w(1+\gamma+\ldots+\gamma^{s-1}),
\end{aligned}$$

where the last identity follows because $\mu\sqrt{\beta} = c_3\sqrt{c_4}b_w$.

Then, as long as $c_2 \geq \frac{2(3c_3\sqrt{c_4}+1)^2}{(3-2\sqrt{2})c_4}$, Assumption 6 offers a lower bound on $\lambda_{\min}(\Phi^\top\Phi)$ which ensures that $\Delta^\top(\Phi^\top\Phi)^{-1}\Delta \leq \frac{\beta}{2}(3-2\sqrt{2})$, implying $\phi_s^\top(\Phi^\top\Phi)^{-1}\phi_s \geq \beta/2$.

To summarize, we have shown whenever Algorithm 3 encounters a situation in which either

$$\sup_{\mathbf{u}} \phi(\mathbf{x}_t, \mathbf{u})^\top (\Phi^\top \Phi)^{-1} \phi(\mathbf{x}_t, \mathbf{u}) \geq \beta \quad \text{or} \quad (\phi_j^R)^\top (\Phi^\top \Phi)^{-1} \phi_j^R \geq \beta, \tag{3.3.16}$$

it collects a measurement $(\phi_t, \mathbf{x}_{t+1})$ such that $\phi_t (\Phi^\top \Phi)^{-1} \phi_t \geq \beta/2$. Hence, according to Proposition 3.3.7, the event (3.3.16) can occur at most $N_e$ times (the value $N_e$ was defined in (3.3.13)).

**Part 3 of the Proof of Theorem 3.3.5.** In this final part of the proof we analyze what happens when the trajectory planning problem returns a reference trajectory $(\mathbf{x}_j^R, \mathbf{u}_j^R)$ for which $|\langle \phi_r^R, v \rangle| \geq \alpha/2$, where $v$ is a minimal eigenvector with unit norm of $\Phi^\top \Phi$.

During its execution the algorithm produces $T/H$ reference trajectories. Part 2 of the proof implies that at least $T/H - N_e$ of the reference trajectories satisfy $|\langle \phi_r^R, v \rangle| \geq \alpha/2$, with all states $\mathbf{x}_t$ encountered during tracking satisfying $\sup_{\mathbf{u}} \phi(\mathbf{x}_t, \mathbf{u})^\top \left(\Phi^\top \Phi\right)^{-1} \phi(\mathbf{x}_t, \mathbf{u}) \leq \beta$ and all reference features $\phi_j^R$ satisfying $(\phi_j^R)^\top (\Phi^\top \Phi)^{-1} \phi_j^R \leq \beta$.

Following the same argument as in Part 2 of the proof we know that tracking the reference trajectory in this case takes the system to a state $\mathbf{x}_t$ such that

$$\|\mathbf{x}_t - \mathbf{x}_r^R\| \leq (3c_3\sqrt{c_4} + 1) b_w (1 + \gamma + \ldots + \gamma^{r-1}),$$

which implies by Assumption 1 that

$$\|\phi_t - \phi_r^R\| \leq (3c_3\sqrt{c_4} + 1) L b_w (1 + \gamma + \ldots + \gamma^{r-1}).$$

This last inequality implies that $|\langle \phi_t, v \rangle| \geq \alpha/4$ if $3c_3\sqrt{c_4} + 1 \leq c_1/2$. Recall that the only condition we imposed so far on $c_4$ is $c_4 \leq c_1^2/(4c_3^2)$ in Part 1 of the proof. Hence, since $c_1 > 2$, we can choose $c_4 \leq \frac{(c_1-2)^2}{36c_3^2}$ to ensure that $c_4 \leq c_1^2/(4c_3^2)$ and $3c_3\sqrt{c_4} + 1 < c_1/2$. Now, to finish the proof of Theorem 3.3.5 we rely on the following result, whose proof we defer to the end of this section.

**Proposition 3.3.8.** *Let $\mathcal{V} \subset \mathbb{R}^k$ be a bounded set, with $\sup_{v \in \mathcal{V}} \|v\| \leq b$, such that for any $u \in \mathbb{S}^{k-1}$ there exists $v \in \mathcal{V}$ with $|\langle u, v \rangle| \geq \alpha$. Then, for all $T \geq 0$, given any sequence of vectors $\{v_t\}_{t \geq 0}$ in $\mathbb{R}^k$ we have*

$$\lambda_{\min} \left( \sum_{i=1}^{T} v_i v_i^\top \right) \geq \frac{\alpha^2 K(T)}{2k} - \frac{k-1}{2} \left( b^2 - \frac{\alpha^2}{2} \right),$$

*where $K(T)$ is the number of times*

$$v_{t+1} \in \{ v | v \in \mathcal{V} \text{ and } |\langle \tilde{v}_{t+1}, v \rangle| \geq \alpha \},$$

*with $\tilde{v}_{t+1} \in \arg\min_{\|v\|=1} v^\top \left( \sum_{i=1}^{t} v_i v_i^\top \right) v$ and $t < T$.*

We have shown that at least $T/H - N_e$ times the algorithm collects a state transition $(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1})$ for which $\phi_t$ is at least $\alpha/4$ aligned with the minimal eigenvector of $\Phi^\top\Phi$, where $\Phi$ is the matrix of all $\phi_j$ observed prior to the last trajectory planning episode. Therefore, Proposition 3.3.8 implies that Algorithm 3 collects a sequence of measurements $(\phi_t, \mathbf{x}_{t+1})$ such that

$$\lambda_{\min}\left(\sum_{t=1}^{T+t_0}\phi_t\phi_t^\top\right) \geq \frac{\alpha^2}{32}\left(\frac{T}{H} - N_e\right) - \frac{k-1}{2}b_\phi^2.$$

Putting this result together with Proposition 3.3.6 yields the desired conclusion, as long as we prove Proposition 3.3.6.

**Proof of Proposition 3.3.6** To prove Proposition 3.3.8 we need the following lemma, which intuitively shows that the sum of the smallest eigenvalues cannot lag behind the larger eigenvalues by too much.

**Lemma 3.3.9.** *Let $M \in \mathbb{R}^{k\times k}$ be a positive semi-definite matrix. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$ be the eigenvalues of $M$ and let $u_1, u_2, \ldots, u_k$ be the corresponding eigenvectors of unit norm. Suppose $v$ is a vector in $\mathbb{R}^k$ such that $\|v\| \leq b$ and $|\langle v, u_k\rangle| \geq \alpha$. Let $\nu_1 \geq \nu_2 \geq \ldots \geq \nu_k$ be the eigenvalues of $M + vv^\top$. Then, for any $s \in \{2, \ldots k\}$ such that $\lambda_{s-1} \geq \lambda_s + b^2 - \alpha^2/2$ we have*

$$\sum_{i=s}^{k}\nu_i \geq \sum_{i=s}^{k}\lambda_i + \frac{\alpha^2}{2}. \tag{3.3.17}$$

*Proof.* First, we express $v$ in $M$'s eigenbasis: $v = \sum_{i=1}^k z_i u_i$. Then, by assumption we know that $\|v\|^2 = \sum_{i=1}^k z_i^2 \leq b^2$ and $z_k^2 \geq \alpha^2$. Using a result by Bunch, Nielsen, and Sorensen [31] we know that $\nu_1 \geq \lambda_1$ and $\nu_i \in [\lambda_i, \lambda_{i-1}]$ for every $i \in \{2, \ldots k\}$ and that the $k$ eigenvalues $\nu_i$ are the $k$ solutions of the secular equation:

$$f(\nu) := 1 + \sum_{i=1}^{k}\frac{z_i^2}{\lambda_i - \nu} = 0 \tag{3.3.18}$$

if $z_i \neq 0$ for all $i$. If $z_i = 0$, there is an eigenvalue $\nu_j$ such that $\nu_j = \lambda_i$. We assume $z_i \neq 0$ for all $i$.

If $\nu_s \geq \lambda_s + \frac{\alpha^2}{2}$, there is nothing to prove. Let us assume $\nu_s < \lambda_s + \frac{\alpha^2}{2}$. Hence, the eigenvalues $\nu_s, \nu_{s+1}, \ldots, \nu_k$ lie in the interval $[\lambda_k, \lambda_s + \alpha^2/2)$. For any $\nu \in [\lambda_k, \lambda_s + \alpha^2/2)$ we have

$$0 \leq \zeta(\nu) := \sum_{i=1}^{s-1}\frac{z_i^2}{\lambda_i - \nu} \leq \frac{\sum_{i=1}^{s-1} z_i^2}{\lambda_{s-1} - \nu} \tag{3.3.19}$$

$$\leq \frac{b^2 - \alpha^2}{\lambda_{s-1} - \nu} \leq \frac{b^2 - \alpha^2}{b^2 - \alpha^2} = 1. \tag{3.3.20}$$

By rewriting the equation $f(\nu) = 0$, for any solution $\nu_\star$ which lies in $[\lambda_k, \lambda_s + \alpha^2/2)$ we obtain

$$0 = 1 + \sum_{i=s}^{k} \frac{z_i^2}{(\lambda_i - \nu_\star)(1 + \zeta(\nu_\star))} \leq 1 + \sum_{i=s}^{k} \frac{z_i^2}{2(\lambda_i - \nu_\star)}$$

because $1 < 1 + \zeta(\nu_\star) \leq 2$ and $\sum_{i=s}^{k} \frac{z_i^2}{(\lambda_i - \nu_\star)} < 0$. Now, let $\nu_j$ be the unique solution $f(\nu_j) = 0$ in the interval $[\lambda_j, \lambda_{j-1}]$ for $j \in \{s+1, \ldots, k\}$ or in the interval $[\lambda_s, \lambda_s + \alpha^2/2)$ for $j = s$.

Since the function $g(\nu) = 1 + \sum_{i=s}^{k} \frac{z_i^2}{2(\lambda_i - \nu)}$ is increasing on the interval $[\lambda_j, \lambda_{j-1}]$ (if $j = s$, the interval is $[\lambda_s, \infty)$) we know that the unique solution $\nu_j' \in [\lambda_j, \lambda_{j-1}]$ of the equation $g(\nu) = 0$ satisfies $\nu_j' \leq \nu_j$ for all $j \in \{s, \ldots, k\}$.

Therefore, we have shown that $\sum_{j=s}^{k} \nu_j \geq \sum_{j=s}^{k} \nu_j'$, where $\nu_j'$ are the solutions to the equation

$$1 + \sum_{i=s}^{k} \frac{z_i^2}{2(\lambda_i - \nu)} = 0.$$

However, the solutions of this equation are the eigenvalues of $Q = \operatorname{diag}(\lambda_s, \lambda_{s+2}, \ldots, \lambda_k) + \frac{1}{2} z z^\top$, where $z = [z_s, z_{s+1}, \ldots, z_k]^\top$. Hence,

$$\sum_{j=s}^{k} \nu_j \geq \sum_{j=s}^{k} \nu_j' = \operatorname{tr}(Q) = \sum_{j=s}^{k} \lambda_j + \frac{1}{2} \sum_{j=s}^{k} z_j^2 \geq \sum_{j=s}^{k} \lambda_j + \frac{\alpha^2}{2}.$$

□

Now we can turn back to the proof of Proposition 3.3.8. Let $\lambda_i(t)$ be the $i$-th largest eigenvalue of $\sum_{j=1}^{t} v_j v_j^\top$ and let $K(t)$ be the number of times

$$v_{j+1} \in \{v | v \in \mathcal{V} \text{ and } |\langle \tilde{v}_{j+1}, v \rangle| \geq \alpha\}$$

with $\tilde{v}_{j+1} \in \arg\min_{\|v\|=1} v^\top \left( \sum_{i=1}^{j} v_i v_i^\top \right) v$ and $j < t$.

Suppose we know that $\sum_{i=j-1}^{k} \lambda_i(t) \geq c_{j-1} \alpha^2 K(t) - d_{j-1}$ for all $t \geq 1$, where $c_{j-1} > 0$ and $d_{j-1} \geq 0$ are some real values. Since $\|v_j\| \geq \alpha$ for all $j$, we can choose $c_1 = 1$ and $d_1 = 0$. Now, we lower bound $\sum_{i=j}^{k} \lambda_i(t)$ as a function of $t$. To this end, we define $t_j$ to be the maximum time in $\{1, 2, \ldots, t\}$ such that $\lambda_{s-1}(t_2) - \lambda_s(t_2) < b^2 - \frac{\alpha^2}{2}$ for all $s \in \{j, j+1, \ldots, k\}$.

Then, Lemma 3.3.9 and our induction hypothesis guarantee that

$$\begin{aligned} \sum_{i=j}^{k} \lambda_i(t) &\geq \sum_{i=j}^{k} \lambda_i(t_j) + \frac{\alpha^2 (K(t) - K(t_j))}{2} \\ &\geq \frac{\alpha^2 (K(t) - K(t_j))}{2} + c_{j-1} \alpha^2 K(t_j) - d_{j-1} - \lambda_{j-1}(t_j). \end{aligned}$$

By the definition of $t_j$ we know that $\lambda_i(t_j) \geq \lambda_{j-1}(t_j) - (i-j+1)(b^2-\alpha^2)$ for all $i \geq j$. Therefore, we have the lower bound:

$$\sum_{i=j}^{k} \lambda_i(t) \geq \sum_{i=j}^{k} \lambda_i(t_j) \geq (k-j+1)\lambda_{j-1}(t_j) - \frac{(k-j+1)(k-j+2)}{2}\left(b^2 - \frac{\alpha^2}{2}\right).$$

We minimize the maximum of the previous two lower bounds with respect to $\lambda_{j-1}(t_j)$, which can be done by finding the value of $\lambda_{j-1}(t_j)$ which makes the two lower bounds equal. Then, we find

$$\sum_{i=j}^{k} \lambda_i(t) \geq \frac{\alpha^2}{2}\frac{k-j+1}{k-j+2}\left((2c_{j-1}-1)K(t_j) + K(t)\right) - \frac{k-j+1}{k-j+2}d_{j-1} - \frac{k-j+1}{2}\left(b^2 - \frac{\alpha^2}{2}\right).$$

*Case 1:* $2c_{j-1} \geq 1$. Then, since $K(t_j) \geq 0$, we obtain

$$\sum_{i=j}^{k} \lambda_i(t) \geq \frac{\alpha^2}{2}\frac{k-j+1}{k-j+2}K(t) - \frac{k-j+1}{k-j+2}d_{j-1} - \frac{k-j+1}{2}\left(b^2 - \frac{\alpha^2}{2}\right).$$

*Case 2:* $2c_{j-1} < 1$. Then, since $K(t_j) \leq K(t)$, we obtain

$$\sum_{i=j}^{k} \lambda_i(t) \geq \alpha^2\frac{k-j+1}{k-j+2}c_{j-1}K(t) - \frac{k-j+1}{k-j+2}d_{j-1} - \frac{k-j+1}{2}\left(b^2 - \frac{\alpha^2}{2}\right).$$

We see that $c_2 = \frac{1}{2}\frac{k-1}{k} < \frac{1}{2}$ and $\frac{k-j+1}{k-j+2}c_{j-1} < c_{j-1}$. Therefore, the following recursions hold

$$\begin{aligned} c_j &= \frac{k-j+1}{k-j+2}c_{j-1} \\ d_j &= \frac{k-j+1}{k-j+2}d_{j-1} + \frac{k-j+1}{2}\left(b^2 - \frac{\alpha^2}{2}\right), \end{aligned}$$

with $c_2 = \frac{k-1}{2k}$ and $d_2 = \frac{k-1}{2}\left(b^2 - \frac{\alpha^2}{2}\right)$. By unrolling the recursions, we obtain the conclusion.

## 3.4 Related work

System identification, being one of the cornerstones of control theory, has a rich history, which we cannot hope to summarize here. For an in-depth presentation of the field we direct the interested reader to the book by Ljung [89] and the review articles by Åström and Eykhoff [16], Bombois et al. [22], Chiuso and Pillonetto [35], Hong et al. [67], Juditsky et al. [74], Ljung et al. [90], Schoukens and Ljung [131], and Sjöberg et al. [144]. Instead, we discuss recent studies of system identification that develop finite-time statistical guarantees.

Most recent theoretical guarantees of system identification apply to linear systems under various sets of assumptions [33, 38, 47, 48, 61–63, 111, 126–128, 142, 149, 154, 155, 160]. Several works built on the results and techniques we developed in Section 3.2. Notably, Sarkar and Rakhlin [126] developed a more general analysis that also applies to a certain class of unstable linear systems. Both of these studies assumed that the estimation method can directly observe the state of the system. We make the same assumption in our work. However, in many applications full state observation is not possible. Recently, Simchowitz et al. [142] proved that marginally stable linear systems can be estimated from partial observations by using a prefiltered least squares method. From the study of linear dynamics, the work of Wagenmaker and Jamieson [160] is the closest to our own. Inspired by E-optimal design [114], the authors propose and analyze an adaptive data collection method for linear system identification which maximizes the minimal eigenvalue $\lambda_{\min}(\sum_{t=0}^{T-1} \mathbf{x}_t \mathbf{x}_t^\top)$ under power constraints on the inputs. Wagenmaker and Jamieson [160] prove matching upper and lower bounds for their method.

There is comparatively little known about the sample complexity of nonlinear system identification. Oymak [109] and Bahmani and Romberg [19] studied the estimation of the parameters $A$ and $B$ of a dynamical system of the form $\mathbf{x}_{t+1} = \phi(A\mathbf{x}_t + B\mathbf{u}_t)$, where $\phi$ is a known activation function and the inputs $u_t$ are i.i.d. standard Gaussian vectors. Importantly, in this model both $\mathbf{x}_t$ and $\mathbf{u}_t$ are observed and there is no unobserved noise, which makes estimation easy when the map $\phi$ is invertible. In follow-up work, Sattar and Oymak [130] and Foster et al. [53] generalized these results. In particular, Foster et al. [53] took inspiration from the study of generalized linear models and showed that a method developed for the standard i.i.d. setting can estimate dynamical systems of the form $\mathbf{x}_{t+1} = \phi(A\mathbf{x}_t) + \mathbf{w}_t$ at an an optimal rate, where $\mathbf{w}_t$ is unobserved i.i.d. noise. All these works share a common characteristic, they study systems for which identification is possible through the use of non-adaptive inputs. We take the first step towards understanding systems that require adaptive methods for successful identification.

In a different line of work, Singh et al. [143] proposed a learning framework for trajectory planning from learned dynamics. They propose a regularizer of dynamics that promotes stabilizability of the learned model, which allows the tracking of reference trajectories based on estimated dynamics. Also, Khosravi and Smith [76] and Khosravi and Smith [77] developed learning methods that exploit other control-theoretic priors. Nonetheless, none of these works characterize the sample complexity of the problem.

While most work that studies sample-complexity questions in the setting of tabular MDPs focuses on finding optimal policies, Jin et al. [72] and Wolfer and Kontorovich [164] recently analyzed data collection for system identification. More precisely, Jin et al. [72] developed an efficient algorithm for the exploration of tabular MDPs that enables near-optimal policy synthesis for an arbitrary number of reward functions, which are unknown during data collection, while Wolfer and Kontorovich [164] derived minimax sample complexity guarantees for the estimation of ergodic Markov chains. Finally, we note that Abbeel and Ng [4] quantified the sample complexity of learning policies from demonstrations for tabular MDPs and for a simpler version of the model class (3.3.1).

# Chapter 4

# The Linear Quadratic Regulator

In this chapter we study the amount of data needed to find a controller that achieves near optimal performance on the Linear Quadratic Regulator (LQR), one of the most well-studied problems in classical optimal control. The results in this chapter are based on the work by Dean et al. [41] and Mania et al. [95].

An instance of the LQR is defined by four matrices: two matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times p}$ that define the linear dynamics and two positive semidefinite matrices $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{p\times p}$ that define the cost function. Given these matrices, the goal of LQR is to solve the optimization problem

$$\min_{\mathbf{u}_0,\mathbf{u}_1,\dots} \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T} \mathbf{x}_t^\top Q\mathbf{x}_t + \mathbf{u}_t^\top R\mathbf{u}_t\right] \quad \text{s.t.} \quad \mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{w}_t, \tag{4.0.1}$$

where $\mathbf{x}_t$, $\mathbf{u}_t$ and $\mathbf{w}_t$ denote the state, input (or action), and noise at time $t$, respectively. The expectation is over the i.i.d. noise $\mathbf{w}_t \sim \mathcal{N}(0, \sigma_w^2 I_n)$. For simplicity of bookkeeping, in our analysis we further assume that we can prepare the system in initial state $x_0 = 0$. As implied by the dimensions of the matrices $A$ and $B$, the state and noise vectors are $n$ and $p$ dimensional, respectively. The input at time $t$ is allowed to depend on the state at time $t$ and all the previous states and actions. Nonetheless, when the problem parameters $(A, B, Q, R)$ are known the optimal policy is given by linear feedback, $\mathbf{u}_t = K_\star \mathbf{x}_t$, and can be computed efficiently [e.g., see 11]. More precisely, $K_\star = -(R + B^\top PB)^{-1}B^\top PA$ where $P$ is the unique positive-definite solution to the discrete Riccati equation

$$P = A^\top PA - A^\top PB(R + B^\top PB)^{-1}B^\top PA + Q \tag{4.0.2}$$

and can be computed efficiently. A few standard assumptions are needed to ensure that (4.0.2) has a unique positive-definite solution: $R$ is positive definite, $(A, B)$ is controllable, $(Q, A)$ is observable [see 169, for details]. Problem (4.0.1) considers an average cost over an infinite horizon. The optimal controller for the finite horizon variant is also static and linear,

but time-varying. The LQR solution in this case can be computed efficiently via dynamic programming.

In this chapter we are interested in the control of a linear dynamical system with unknown transition parameters $(A_\star, B_\star)$based on estimates $(\widehat{A}, \widehat{B})$ and upper bounds $\varepsilon_A$ and $\varepsilon_B$ on the estimation error, i.e., $\|A_\star - \widehat{A}\| \leq \varepsilon_A$ and $\|B_\star - \widehat{B}\| \leq \varepsilon_B$. Unless otherwise noted, $\|\cdot\|$ denotes the Euclidean norm when applied to vectors and the operator norm when applied to matrices. In Chapter 3 we discussed how to find estimates $(\widehat{A}, \widehat{B}, \varepsilon_A, \varepsilon_B)$. From now on we assume access to estimates $\widehat{A}$ and $\widehat{B}$ and to upper bounds on their estimation error. The cost matrices $Q$ and $R$ are assumed known.

Given an estimate $(\widehat{A}, \widehat{B}, \varepsilon_A, \varepsilon_B)$ of the linear dynamics, there are two main strategies for producing a controller: compute the optimal LQR controller for $(\widehat{A}, \widehat{B})$ or compute a robust controller for the worst case transition matrices $(A, B)$ such that $\|A - \widehat{A}\| \leq \varepsilon_A$ and $\|B - \widehat{B}\| \leq \varepsilon_B$. The former approach is known as certainty equivalence or as the plug-in method. We refer to the second approach as robust LQR. If we denote by $\widehat{J}$ the LQR cost achieved by one of these methods and by $J_\star$ the optimal LQR cost, in Section 4.2 we show that $\widehat{J} - J_\star = \mathcal{O}(\max\{\varepsilon_A^2, \varepsilon_B^2\})$ in the case of certainty equivalence and in Section 4.2 we show that $\widehat{J} - J_\star = \mathcal{O}(\max\{\varepsilon_A, \varepsilon_B\})$ in the case of robust LQR.

The importance of these results becomes clearer when viewed from a statistical lens. In Chapter 3 we saw that if we use $T$ data points to estimate the dynamics $(A, B)$ we get $\max\{\varepsilon_A, \varepsilon_B\} = \widetilde{\mathcal{O}}(\sqrt{n + p/T})$. Therefore, putting everything together, we see that as we collect more data $\widehat{J} - J_\star$ decays quadratically faster in the case certainty equivalence than in the case of robust LQR. Nonetheless, we also show that robust LQR can surpass the performance of certainty equivalence in certain regimes.

## 4.1 Robust controller synthesis

In this section we analyze the suboptimality gap $\widehat{J} - J_\star$ obtained by optimizing the objective

$$\begin{array}{lll} \text{minimize} & \sup\limits_{\substack{\|\Delta_A\|_2 \le \epsilon_A \\ \|\Delta_B\|_2 \le \epsilon_B}} & \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[x_t^\top Q x_t + u_t^\top R u_t\right] \\ & \text{subject to} & x_{t+1} = (\widehat{A} + \Delta_A) x_t + (\widehat{B} + \Delta_B) u_t + w_t \end{array}, \tag{4.1.1}$$

where we dropped the boldface notation from the vectors $x_t$, $u_t$, and $w_t$ because throughout this section we reserve boldface notation to denote transfer functions, as described shortly.

Although classic methods exist for computing such controllers [50, 112, 150, 165], they typically require solving nonconvex optimization problems, and it is not readily obvious how to extract interpretable measures of controller performance as a function of the perturbation sizes $\epsilon_A$ and $\epsilon_B$. To that end, we leverage the recently developed System Level Synthesis (SLS) framework [161] to create an alternative robust synthesis procedure. Described in detail in Section 4.1.2, SLS lifts the system description into a higher dimensional space that enables efficient search for controllers. At the cost of some conservatism, we are able to guarantee robust stability of the resulting closed-loop system for all admissible perturbations and bound the performance gap between the resulting controller and the optimal LQR controller.

With estimates of the system $(\widehat{A}, \widehat{B})$ and operator norm error bounds $(\epsilon_A, \epsilon_B)$ in hand, we now turn to control design. In this section we introduce some useful tools from *System Level Synthesis* (SLS), a recently developed approach to control design that relies on a particular parameterization of signals in a control system [98, 161]. We review the main SLS framework, highlighting the key constructions that we will use to solve the robust LQR problem. As we show in this and the following section, using the SLS framework, as opposed to traditional techniques from robust control, allows us to (a) compute robust controllers using semidefinite programming, and (b) provide sub-optimality guarantees in terms of the size of the uncertainties on our system estimates.

### 4.1.1 Useful results from system level synthesis

The SLS framework focuses on the *system responses* of a closed-loop system. As a motivating example, consider linear dynamics under a fixed a static state-feedback control policy $K$, i.e., let $u_k = K x_k$. Then, the closed loop map from the disturbance process $\{w_0, w_1, \dots\}$ to the state $x_k$ and control input $u_k$ at time $k$ is given by

$$\begin{array}{rcl} x_k &=& \sum_{t=1}^{k} (A_\star + B_\star K)^{k-t} w_{t-1} \,, \\ u_k &=& \sum_{t=1}^{k} K (A_\star + B_\star K)^{k-t} w_{t-1} \,. \end{array} \tag{4.1.2}$$

Letting $\Phi_x(k) := (A_\star + B_\star K)^{k-1}$ and $\Phi_u(k) := K(A_\star + B_\star K)^{k-1}$, we can rewrite Eq. (4.1.2) as

$$\begin{bmatrix} x_k \\ u_k \end{bmatrix} = \sum_{t=1}^{k} \begin{bmatrix} \Phi_x(k-t+1) \\ \Phi_u(k-t+1) \end{bmatrix} w_{t-1} \,, \tag{4.1.3}$$

where $\{\Phi_x(k), \Phi_u(k)\}$ are called the *closed-loop system response elements* induced by the static controller $K$.

Note that even when the control is a linear function of the state and its past history (i.e. a linear dynamic controller), the expression (4.1.3) is valid. Though we conventionally think of the control policy as a function mapping states to input, whenever such a mapping is linear, both the control input and the state can be written as linear functions of the disturbance signal $w_t$. With such an identification, the dynamics require that the $\{\Phi_x(k), \Phi_u(k)\}$ must obey the constraints

$$\Phi_x(k+1) = A_\star \Phi_x(k) + B_\star \Phi_u(k) \,, \;\; \Phi_x(1) = I \,, \;\; \forall k \geq 1 \,, \tag{4.1.4}$$

As we describe in more detail below in Theorem 4.1.1, these constraints are in fact both necessary and sufficient. Working with closed-loop system responses allows us to cast optimal control problems as optimization problems over elements $\{\Phi_x(k), \Phi_u(k)\}$, constrained to satisfy the affine equations (4.1.4). Comparing equations (4.1.2) and (4.1.3), we see that the former is non-convex in the controller $K$, whereas the latter is affine in the elements $\{\Phi_x(k), \Phi_u(k)\}$.

As we work with infinite horizon problems, it is notationally more convenient to work with *transfer function* representations of the above objects, which can be obtained by taking a $z$-transform of their time-domain representations. The frequency domain variable $z$ can be informally thought of as the time-shift operator, i.e., $z\{x_k, x_{k+1}, \dots\} = \{x_{k+1}, x_{k+2}, \dots\}$, allowing for a compact representation of LTI dynamics. We use boldface letters to denote such transfer functions signals in the frequency domain, e.g., $\mathbf{\Phi}_x(z) = \sum_{k=1}^{\infty} \Phi_x(k) z^{-k}$. Then, the constraints (4.1.4) can be rewritten as

$$\begin{bmatrix} zI - A_\star & -B_\star \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} = I \,,$$

and the corresponding (not necessarily static) control law $\mathbf{u} = \mathbf{K}\mathbf{x}$ is given by $\mathbf{K} = \mathbf{\Phi}_u \mathbf{\Phi}_x^{-1}$.

We formalize our discussion by introducing notation that is common in the controls literature. For a thorough introduction to the functional analysis commonly used in control theory, see Chapters 2 and 3 of Zhou et al. [169]. Let $\mathbb{T}$ (resp. $\mathbb{D}$) denote the unit circle (resp. open unit disk) in the complex plane. The restriction of the Hardy spaces $\mathcal{H}_\infty(\mathbb{T})$ and $\mathcal{H}_2(\mathbb{T})$ to matrix-valued real-rational functions that are analytic on the complement of $\mathbb{D}$ will be referred to as $\mathcal{RH}_\infty$ and $\mathcal{RH}_2$, respectively. In controls parlance, this corresponds to (discrete-time) stable matrix-valued transfer functions. For these two function spaces, the

$\mathcal{H}_\infty$ and $\mathcal{H}_2$ norms simplify to

$$\|\mathbf{G}\|_{\mathcal{H}_\infty} = \sup_{z\in\mathbb{T}} \|G(z)\|_2\,, \quad \|\mathbf{G}\|_{\mathcal{H}_2} = \sqrt{\frac{1}{2\pi}\int_{\mathbb{T}} \|G(z)\|_F^2\, dz}\,. \tag{4.1.5}$$

Finally, the notation $\frac{1}{z}\mathcal{RH}_\infty$ refers to the set of transfer functions $\mathbf{G}$ such that $z\mathbf{G} \in \mathcal{RH}_\infty$. Equivalently, $\mathbf{G} \in \frac{1}{z}\mathcal{RH}_\infty$ if $\mathbf{G} \in \mathcal{RH}_\infty$ and $\mathbf{G}$ is strictly proper.

The most important transfer function for the LQR problem is the map from the state sequence to the control actions: the control policy. Consider an arbitrary transfer function $\mathbf{K}$ denoting the map from state to control action, $\mathbf{u} = \mathbf{K}\mathbf{x}$. Then the closed-loop transfer matrices from the process noise $\mathbf{w}$ to the state $\mathbf{x}$ and control action $\mathbf{u}$ satisfy

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{u}\end{bmatrix} = \begin{bmatrix}(zI - A - B\mathbf{K})^{-1}\\ \mathbf{K}(zI - A - B\mathbf{K})^{-1}\end{bmatrix}\mathbf{w}. \tag{4.1.6}$$

We then have the following theorem parameterizing the set of stable closed-loop transfer matrices, as described in equation (4.1.6), that are achievable by a given stabilizing controller $\mathbf{K}$.

**Theorem 4.1.1** (State-Feedback Parameterization [161])**.** *The following are true:*

- *The affine subspace defined by*

$$\begin{bmatrix}zI - A & -B\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}_x\\ \mathbf{\Phi}_u\end{bmatrix} = I,\ \ \mathbf{\Phi}_x, \mathbf{\Phi}_u \in \frac{1}{z}\mathcal{RH}_\infty \tag{4.1.7}$$

  *parameterizes all system responses* (4.1.6) *from* $\mathbf{w}$ *to* $(\mathbf{x}, \mathbf{u})$*, achievable by an internally stabilizing state-feedback controller* $\mathbf{K}$*.*

- *For any transfer matrices* $\{\mathbf{\Phi}_x, \mathbf{\Phi}_u\}$ *satisfying* (4.1.7)*, the controller* $\mathbf{K} = \mathbf{\Phi}_u\mathbf{\Phi}_x^{-1}$ *is internally stabilizing and achieves the desired system response* (4.1.6)*.*

Note that in particular, $\{\mathbf{\Phi}_x, \mathbf{\Phi}_u\} = \{(zI-A-B\mathbf{K})^{-1}, \mathbf{K}(zI-A-B\mathbf{K})^{-1}\}$ as in (4.1.6) are elements of the affine space defined by (4.1.7) whenever $\mathbf{K}$ is a causal stabilizing controller.

We will also make extensive use of a robust variant of Theorem 4.1.1.

**Theorem 4.1.2** (Robust Stability [98])**.** *Suppose that the transfer matrices* $\{\mathbf{\Phi}_x, \mathbf{\Phi}_u\} \in \frac{1}{z}\mathcal{RH}_\infty$ *satisfy*

$$\begin{bmatrix}zI - A & -B\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}_x\\ \mathbf{\Phi}_u\end{bmatrix} = I + \mathbf{\Delta}. \tag{4.1.8}$$

*Then the controller* $\mathbf{K} = \mathbf{\Phi}_u\mathbf{\Phi}_x^{-1}$ *stabilizes the system described by* $(A, B)$ *if and only if* $(I + \mathbf{\Delta})^{-1} \in \mathcal{RH}_\infty$*. Furthermore, the resulting system response is given by*

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{u}\end{bmatrix} = \begin{bmatrix}\mathbf{\Phi}_x\\ \mathbf{\Phi}_u\end{bmatrix}(I + \mathbf{\Delta})^{-1}\mathbf{w}. \tag{4.1.9}$$

**Corollary 4.1.3.** *Under the assumptions of Theorem 4.1.2, if* $\|\mathbf{\Delta}\| < 1$ *for any induced norm* $\|\cdot\|$*, then the controller* $\mathbf{K} = \mathbf{\Phi}_u \mathbf{\Phi}_x^{-1}$ *stabilizes the system described by* $(A, B)$*.*

*Proof.* Follows immediately from the small gain theorem, see for example Section 9.2 in [169]. □

### 4.1.2 Robust LQR synthesis

We return to the problem setting where estimates $(\widehat{A}, \widehat{B})$ of a true system $(A, B)$ satisfy

$$\|\Delta_A\|_2 \leq \epsilon_A, \quad \|\Delta_B\|_2 \leq \epsilon_B$$

where $\Delta_A := \widehat{A} - A$ and $\Delta_B := \widehat{B} - B$ and where we wish to minimize the LQR cost for the worst instantiation of the parametric uncertainty.

Before proceeding, we must formulate the LQR problem in terms of the system responses $\{\Phi_x(k), \Phi_u(k)\}$. It follows from Theorem 4.1.1 and the standard equivalence between infinite horizon LQR and $\mathcal{H}_2$ optimal control that, for a disturbance process distributed as $w_t \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_w^2 I)$, the standard LQR problem (1.0.2) can be equivalently written as

$$\min_{\mathbf{\Phi}_x, \mathbf{\Phi}_u} \sigma_w^2 \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} \right\|_{\mathcal{H}_2}^2 \text{ s.t. equation (4.1.7).} \tag{4.1.10}$$

A full derivation of this equivalence can be found in appendix of the paper by Dean et al. [41]. Going forward, we drop the $\sigma_w^2$ multiplier in the objective function as it affects neither the optimal controller nor the sub-optimality guarantees.

We begin with a simple sufficient condition under which any controller $\mathbf{K}$ that stabilizes $(\widehat{A}, \widehat{B})$ also stabilizes the true system $(A, B)$. To state the lemma, we introduce one additional piece of notation. For a matrix $M$, we let $\mathfrak{R}_M$ denote the resolvent

$$\mathfrak{R}_M := (zI - M)^{-1}. \tag{4.1.11}$$

We now can state our robustness lemma.

**Lemma 4.1.4.** *Let the controller* $\mathbf{K}$ *stabilize* $(\widehat{A}, \widehat{B})$ *and* $(\mathbf{\Phi}_x, \mathbf{\Phi}_u)$ *be its corresponding system response (4.1.6) on system* $(\widehat{A}, \widehat{B})$*. Then if* $\mathbf{K}$ *stabilizes* $(A, B)$*, it achieves the following LQR cost*

$$J(A, B, \mathbf{K}) := \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} \left( I + \begin{bmatrix} \Delta_A & \Delta_B \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} \right)^{-1} \right\|_{\mathcal{H}_2}. \tag{4.1.12}$$

*Furthermore, letting*

$$\hat{\mathbf{\Delta}} := \begin{bmatrix} \Delta_A & \Delta_B \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} = (\Delta_A + \Delta_B \mathbf{K}) \mathfrak{R}_{\widehat{A} + \widehat{B}\mathbf{K}}. \tag{4.1.13}$$

*a sufficient condition for* $\mathbf{K}$ *to stabilize* $(A, B)$ *is that* $\|\hat{\mathbf{\Delta}}\|_{\mathcal{H}_\infty} < 1$*.*

*Proof.* Follows immediately from Theorems 4.1.1, 4.1.2 and corollary 4.1.3 by noting that for system responses $(\mathbf{\Phi}_x, \mathbf{\Phi}_u)$ satisfying

$$\begin{bmatrix} zI - \widehat{A} & -\widehat{B} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} = I,$$

it holds that

$$\begin{bmatrix} zI - A & -B \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} = I + \widehat{D}$$

for $\widehat{D}$ as defined in equation (4.1.13). □

We can therefore recast the robust LQR problem (4.1.1) in the following equivalent form

$$\begin{aligned} &\min_{\mathbf{\Phi}_x, \mathbf{\Phi}_u} \sup_{\substack{\|\Delta_A\|_2 \le \epsilon_A \\ \|\Delta_B\|_2 \le \epsilon_B}} J(A, B, \mathbf{K}) \\ &\text{s.t.} \begin{bmatrix} zI - \widehat{A} & -\widehat{B} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} = I, \quad \mathbf{\Phi}_x, \mathbf{\Phi}_u \in \frac{1}{z}\mathcal{RH}_\infty \,. \end{aligned} \tag{4.1.14}$$

The resulting robust control problem is one subject to real-parametric uncertainty, a class of problems known to be computationally intractable [25]. Although effective computational heuristics (e.g., DK iteration [169]) exist, the performance of the resulting controller on the true system is difficult to characterize analytically in terms of the size of the perturbations.

To circumvent this issue, we take a slightly conservative approach and find an upper-bound to the cost $J(A, B, \mathbf{K})$ that is independent of the uncertainties $\Delta_A$ and $\Delta_B$. First, note that if $\|\widehat{D}\|_{\mathcal{H}_\infty} < 1$, we can write

$$J(A, B, \mathbf{K}) \le \|(I + \widehat{D})^{-1}\|_{\mathcal{H}_\infty} J(\widehat{A}, \widehat{B}, \mathbf{K}) \le \frac{1}{1 - \|\widehat{D}\|_{\mathcal{H}_\infty}} J(\widehat{A}, \widehat{B}, \mathbf{K}). \tag{4.1.15}$$

Because $J(\widehat{A}, \widehat{B}, \mathbf{K})$ captures the performance of the controller $\mathbf{K}$ on the nominal system $(\widehat{A}, \widehat{B})$, it is not subject to any uncertainty. It therefore remains to compute a tractable bound for $\|\widehat{D}\|_{\mathcal{H}_\infty}$, which we do using the following fact.

**Proposition 4.1.5.** *For any $\alpha \in (0, 1)$ and $\hat{\mathbf{\Delta}}$ as defined in (4.1.13)*

$$\|\hat{\mathbf{\Delta}}\|_{\mathcal{H}_\infty} \le \left\| \begin{bmatrix} \frac{\epsilon_A}{\sqrt{\alpha}} \mathbf{\Phi}_x \\ \frac{\epsilon_B}{\sqrt{1-\alpha}} \mathbf{\Phi}_u \end{bmatrix} \right\|_{\mathcal{H}_\infty} =: H_\alpha(\mathbf{\Phi}_x, \mathbf{\Phi}_u) \,. \tag{4.1.16}$$

*Proof.* Note that for any block matrix of the form $\begin{bmatrix} M_1 & M_2 \end{bmatrix}$, we have

$$\left\| \begin{bmatrix} M_1 & M_2 \end{bmatrix} \right\|_2 \le \left( \|M_1\|_2^2 + \|M_2\|_2^2 \right)^{1/2} . \tag{4.1.17}$$

To verify this assertion, note that

$$\left\|\begin{bmatrix} M_1 & M_2 \end{bmatrix}\right\|_2^2 = \lambda_{\max}(M_1 M_1^* + M_2 M_2^*) \leq \lambda_{\max}(M_1 M_1^*) + \lambda_{\max}(M_2 M_2^*) = \|M_1\|_2^2 + \|M_2\|_2^2 .$$

With (4.1.17) in hand, we have

$$\begin{aligned}\left\|\begin{bmatrix} \Delta_A & \Delta_B \end{bmatrix}\begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix}\right\|_{\mathcal{H}_\infty} &= \left\|\begin{bmatrix} \frac{\sqrt{\alpha}}{\epsilon_A}\Delta_A & \frac{\sqrt{1-\alpha}}{\epsilon_B}\Delta_B \end{bmatrix}\begin{bmatrix} \frac{\epsilon_A}{\sqrt{\alpha}}\mathbf{\Phi}_x \\ \frac{\epsilon_B}{\sqrt{1-\alpha}}\mathbf{\Phi}_u \end{bmatrix}\right\|_{\mathcal{H}_\infty} \\ &\leq \left\|\begin{bmatrix} \frac{\sqrt{\alpha}}{\epsilon_A}\Delta_A & \frac{\sqrt{1-\alpha}}{\epsilon_B}\Delta_B \end{bmatrix}\right\|_2 \left\|\begin{bmatrix} \frac{\epsilon_A}{\sqrt{\alpha}}\mathbf{\Phi}_x \\ \frac{\epsilon_B}{\sqrt{1-\alpha}}\mathbf{\Phi}_u \end{bmatrix}\right\|_{\mathcal{H}_\infty} \leq \left\|\begin{bmatrix} \frac{\epsilon_A}{\sqrt{\alpha}}\mathbf{\Phi}_x \\ \frac{\epsilon_B}{\sqrt{1-\alpha}}\mathbf{\Phi}_u \end{bmatrix}\right\|_{\mathcal{H}_\infty},\end{aligned}$$

completing the proof. □

The following corollary is then immediate.

**Corollary 4.1.6.** *Let the controller* $\mathbf{K}$ *and resulting system response* $(\mathbf{\Phi}_x, \mathbf{\Phi}_u)$ *be as defined in Lemma 4.1.4. Then if* $H_\alpha(\mathbf{\Phi}_x, \mathbf{\Phi}_u) < 1$*, the controller* $\mathbf{K} = \mathbf{\Phi}_u \mathbf{\Phi}_x^{-1}$ *stabilizes the true system* $(A, B)$*.*

Applying Proposition 4.1.5 in conjunction with the bound (4.1.15), we arrive at the following upper bound to the cost function of the robust LQR problem (4.1.1), which is independent of the perturbations $(\Delta_A, \Delta_B)$:

$$\sup_{\substack{\|\Delta_A\|_2 \leq \epsilon_A \\ \|\Delta_B\|_2 \leq \epsilon_B}} J(A, B, \mathbf{K}) \leq \left\|\begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}} \end{bmatrix}\begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix}\right\|_{\mathcal{H}_2} \frac{1}{1 - H_\alpha(\mathbf{\Phi}_x, \mathbf{\Phi}_u)} = \frac{J(\widehat{A}, \widehat{B}, \mathbf{K})}{1 - H_\alpha(\mathbf{\Phi}_x, \mathbf{\Phi}_u)} . \quad (4.1.18)$$

The upper bound is only valid when $H_\alpha(\mathbf{\Phi}_x, \mathbf{\Phi}_u) < 1$, which guarantees the stability of the closed-loop system as in corollary 4.1.6. We remark that corollary 4.1.6 and the bound in (4.1.18) are of interest independent of the synthesis procedure for $\mathbf{K}$. In particular, they can be applied to the optimal LQR controller $\widehat{K}$ computed using the nominal system $(\widehat{A}, \widehat{B})$.

As the next lemma shows, the right hand side of Equation (4.1.18) can be efficiently optimized by an appropriate decomposition. The proof of the lemma is immediate.

**Lemma 4.1.7.** *For functions* $f : \mathcal{X} \to \mathbb{R}$ *and* $g : \mathcal{X} \to \mathbb{R}$ *and constraint set* $C \subseteq \mathcal{X}$*, consider*

$$\min_{x \in C} \frac{f(x)}{1 - g(x)} .$$

*Assuming that* $f(x) \geq 0$ *and* $0 \leq g(x) < 1$ *for all* $x \in C$*, this optimization problem can be reformulated as an outer single-variable problem and an inner constrained optimization problem (the objective value of an optimization over the emptyset is defined to be infinity):*

$$\min_{x \in C} \frac{f(x)}{1 - g(x)} = \min_{\gamma \in [0,1)} \frac{1}{1-\gamma} \min_{x \in C} \{f(x) \mid g(x) \leq \gamma\}$$

Then combining Lemma 4.1.7 with the upper bound in (4.1.18) results in the following optimization problem:

$$\begin{aligned} \text{minimize}_{\gamma\in[0,1)} \frac{1}{1-\gamma} \min_{\mathbf{\Phi}_x,\mathbf{\Phi}_u} & \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} \right\|_{\mathcal{H}_2} \\ \text{s.t.} & \begin{bmatrix} zI - \widehat{A} & -\widehat{B} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} = I, \quad \left\| \begin{bmatrix} \frac{\epsilon_A}{\sqrt{\alpha}} \mathbf{\Phi}_x \\ \frac{\epsilon_B}{\sqrt{1-\alpha}} \mathbf{\Phi}_u \end{bmatrix} \right\|_{\mathcal{H}_\infty} \leq \gamma \\ & \mathbf{\Phi}_x, \mathbf{\Phi}_u \in \frac{1}{z}\mathcal{RH}_\infty. \end{aligned} \tag{4.1.19}$$

We note that this optimization objective is jointly quasi-convex in $(\gamma, \mathbf{\Phi}_x, \mathbf{\Phi}_u)$. Hence, as a function of $\gamma$ alone the objective is quasi-convex, and furthermore is smooth in the feasible domain. Therefore, the outer optimization with respect to $\gamma$ can effectively be solved with methods like golden section search. We remark that the inner optimization is a convex problem, though an infinite dimensional one. We show in Section 4.1.3 that a simple finite impulse response truncation yields a finite dimensional problem with similar guarantees of robustness and performance.

We further remark that because $\gamma \in [0, 1)$, any feasible solution $(\mathbf{\Phi}_x, \mathbf{\Phi}_u)$ to optimization problem (4.1.19) generates a controller $\mathbf{K} = \mathbf{\Phi}_u \mathbf{\Phi}_x^{-1}$ satisfying the conditions of corollary 4.1.6, and hence stabilizes the true system $(A, B)$. Therefore, even if the solution is approximated, as long as it is feasible, it will be stabilizing. As we show in the next section, for sufficiently small estimation error bounds $\epsilon_A$ and $\epsilon_B$, we can further bound the sub-optimality of the performance achieved by our robustly stabilizing controller relative to that achieved by the optimal LQR controller $K_\star$.

Now, we upper bound the performance of the controller synthesized using the optimization (4.1.19) in terms of the size of the perturbations $(\Delta_A, \Delta_B)$ and a measure of complexity of the LQR problem defined by $A$, $B$, $Q$, and $R$. The following result is one of our main contributions.

**Theorem 4.1.8.** *Let $J_\star$ denote the minimal LQR cost achievable by any controller for the dynamical system with transition matrices $(A, B)$, and let $K_\star$ denote the optimal contoller. Let $(\widehat{A}, \widehat{B})$ be estimates of the transition matrices such that $\|\Delta_A\|_2 \leq \epsilon_A$, $\|\Delta_B\|_2 \leq \epsilon_B$. Then, if $\mathbf{K}$ is synthesized via (4.1.19) with $\alpha = 1/2$, the relative error in the LQR cost is*

$$\frac{J(A_\star, B_\star, \mathbf{K}) - J_\star}{J_\star} \leq 5(\epsilon_A + \epsilon_B \|K_\star\|_2) \|\mathfrak{R}_{A+BK_\star}\|_{\mathcal{H}_\infty}, \tag{4.1.20}$$

*as long as $(\epsilon_A + \epsilon_B \|K_\star\|_2) \|\mathfrak{R}_{A+BK_\star}\|_{\mathcal{H}_\infty} \leq 1/5$.*

This result offers a guarantee on the performance of the SLS synthesized controller regardless of the estimation procedure used to estimate the transition matrices. Together with the results shown in Chapter 3 on system identification, Theorem 4.1.8 yields a sample complexity upper bound on the performance of the robust SLS controller $\mathbf{K}$ when $(A, B)$ are not known. The rest of the section is dedicated to proving Theorem 4.1.8.

Recall that $K_\star$ is the optimal LQR static state feedback matrix for the true dynamics $(A, B)$, and let $\mathbf{\Delta} := -\left[\Delta_A + \Delta_B K_\star\right] \mathfrak{R}_{A+BK_\star}$. We begin with a technical result.

**Lemma 4.1.9.** *Define* $\zeta := (\epsilon_A + \epsilon_B \|K_\star\|_2)\|\mathfrak{R}_{A+BK_\star}\|_{\mathcal{H}_\infty}$, *and suppose that* $\zeta < (1+\sqrt{2})^{-1}$. *Then* $(\gamma_0, \tilde{\mathbf{\Phi}}_x, \tilde{\mathbf{\Phi}}_u)$ *is a feasible solution of* (4.1.19) *with* $\alpha = 1/2$, *where*

$$\gamma_0 = \frac{\sqrt{2}\zeta}{1-\zeta}, \quad \tilde{\mathbf{\Phi}}_x = \mathfrak{R}_{A+BK_\star}(I+\mathbf{\Delta})^{-1}, \quad \tilde{\mathbf{\Phi}}_u = K_\star \mathfrak{R}_{A+BK_\star}(I+\mathbf{\Delta})^{-1}. \tag{4.1.21}$$

*Proof.* By construction $\tilde{\mathbf{\Phi}}_x, \tilde{\mathbf{\Phi}}_u \in \frac{1}{z}\mathcal{R}\mathcal{H}_\infty$. Therefore, we are left to check three conditions:

$$\gamma_0 < 1, \quad \begin{bmatrix} zI - \widehat{A} & -\widehat{B} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{\Phi}}_x \\ \tilde{\mathbf{\Phi}}_u \end{bmatrix} = I \text{ , and } \quad \left\| \begin{bmatrix} \frac{\epsilon_A}{\sqrt{\alpha}} \tilde{\mathbf{\Phi}}_x \\ \frac{\epsilon_B}{\sqrt{1-\alpha}} \tilde{\mathbf{\Phi}}_u \end{bmatrix} \right\|_{\mathcal{H}_\infty} \leq \frac{\sqrt{2}\zeta}{1-\zeta}. \tag{4.1.22}$$

The first two conditions follow by simple algebraic computations. Before we check the last condition, note that $\|\mathbf{\Delta}\|_{\mathcal{H}_\infty} \leq (\epsilon_A + \epsilon_B\|K_\star\|_2)\|\mathfrak{R}_{A+BK_\star}\|_{\mathcal{H}_\infty} = \zeta < 1$. Now observe that,

$$\begin{aligned}
\left\| \begin{bmatrix} \frac{\epsilon_A}{\sqrt{\alpha}} \tilde{\mathbf{\Phi}}_x \\ \frac{\epsilon_B}{\sqrt{1-\alpha}} \tilde{\mathbf{\Phi}}_u \end{bmatrix} \right\|_{\mathcal{H}_\infty} &= \sqrt{2} \left\| \begin{bmatrix} \epsilon_A \mathfrak{R}_{A+BK_\star} \\ \epsilon_B K_\star \mathfrak{R}_{A+BK_\star} \end{bmatrix} (I+\mathbf{\Delta})^{-1} \right\|_{\mathcal{H}_\infty} \\
&\leq \sqrt{2}\|(I+\mathbf{\Delta})^{-1}\|_{\mathcal{H}_\infty} \left\| \begin{bmatrix} \epsilon_A \mathfrak{R}_{A+BK_\star} \\ \epsilon_B K_\star \mathfrak{R}_{A+BK_\star} \end{bmatrix} \right\|_{\mathcal{H}_\infty} \\
&\leq \frac{\sqrt{2}}{1-\|\mathbf{\Delta}\|_{\mathcal{H}_\infty}} \left\| \begin{bmatrix} \epsilon_A I \\ \epsilon_B K_\star \end{bmatrix} \mathfrak{R}_{A+BK_\star} \right\|_{\mathcal{H}_\infty} \\
&\leq \frac{\sqrt{2}(\epsilon_A + \epsilon_B\|K_\star\|_2)\|\mathfrak{R}_{A+BK_\star}\|_{\mathcal{H}_\infty}}{1-\|\mathbf{\Delta}\|_{\mathcal{H}_\infty}} \leq \frac{\sqrt{2}\zeta}{1-\zeta}\,.
\end{aligned}$$

□

*Proof of Theorem 4.1.8.* Let $(\gamma_\star, \mathbf{\Phi}_x^\star, \mathbf{\Phi}_u^\star)$ be an optimal solution to problem (4.1.19) and let $\mathbf{K} = \mathbf{\Phi}_u^\star (\mathbf{\Phi}_x^\star)^{-1}$. We can then write

$$J(A, B, \mathbf{K}) \leq \frac{1}{1-\|\widehat{D}\|_{\mathcal{H}_\infty}} J(\widehat{A}, \widehat{B}, \mathbf{K}) \leq \frac{1}{1-\gamma_\star} J(\widehat{A}, \widehat{B}, \mathbf{K}),$$

where the first inequality follows from the bound (4.1.15), and the second follows from the fact that $\|\widehat{D}\|_{\mathcal{H}_\infty} \leq \gamma_\star$ due to Proposition 4.1.5 and the constraint in optimization problem (4.1.19).

From Lemma 4.1.9 we know that $(\gamma_0, \tilde{\mathbf{\Phi}}_x, \tilde{\mathbf{\Phi}}_u)$ defined in equation (4.1.21) is also a feasible solution. Therefore, because $K_\star = \tilde{\mathbf{\Phi}}_u \tilde{\mathbf{\Phi}}_x^{-1}$, we have by optimality,

$$\frac{1}{1-\gamma_\star} J(\widehat{A}, \widehat{B}, \mathbf{K}) \leq \frac{1}{1-\gamma_0} J(\widehat{A}, \widehat{B}, K_\star) \leq \frac{J(A, B, K_\star)}{(1-\gamma_0)(1-\|\mathbf{\Delta}\|_{\mathcal{H}_\infty})} = \frac{J_\star}{(1-\gamma_0)(1-\|\mathbf{\Delta}\|_{\mathcal{H}_\infty})}\,,$$

where the second inequality follows by the argument used to derive (4.1.15) with the true and estimated transition matrices switched. Recall that $\|\mathbf{\Delta}\|_{\mathcal{H}_\infty} \leq \zeta$ and that $\gamma_0 = \sqrt{2}\zeta/(1+\zeta)$. Therefore

$$\frac{J(A,B,\mathbf{K}) - J_\star}{J_\star} \leq \frac{1}{1-(1+\sqrt{2})\zeta} - 1 = \frac{(1+\sqrt{2})\zeta}{1-(1+\sqrt{2})\zeta} \leq 5\zeta\,,$$

where the last inequality follows because $\zeta < 1/5 < 1/(2+2\sqrt{2})$. The conclusion follows. □

Therefore, we have shown that robust LQR that leverages System Level Synthesis achieves a suboptimality gap $\widehat{J} - J_\star = \mathcal{O}(\max\{\varepsilon_A, \varepsilon_B\})$, as promised.

### 4.1.3 Finite impulse response approximation

As posed, the main optimization problem (4.1.19) is a semi-infinite program, and we are not aware of a way to solve this problem efficiently. An elementary approach to reducing the aforementioned semi-infinite program to a finite dimensional one is to only optimize over the first $L$ elements of the transfer functions $\mathbf{\Phi}_x$ and $\mathbf{\Phi}_u$, effectively taking a finite impulse response (FIR) approximation. Since these are both stable maps, we expect the effects of such an approximation to be negligible as long as the optimization horizon $L$ is chosen to be sufficiently large – in what follows, we show that this is indeed the case.

By restricting our optimization to FIR approximations of $\mathbf{\Phi}_x$ and $\mathbf{\Phi}_u$, we can cast the $\mathcal{H}_2$ cost as a second order cone constraint. The only difficulty arises in posing the $\mathcal{H}_\infty$ constraint as a semidefinite program. Though there are several ways to cast $\mathcal{H}_\infty$ constraints as linear matrix inequalities, we use the formulation in Theorem 5.8 of Dumitrescu's text to take advantage of the FIR structure in our problem [44]. We note that using Dumitrescu's formulation, the resulting problem is affine in $\alpha$ when $\gamma$ is fixed, and hence we can solve for the optimal value of $\alpha$. Then the resulting system response elements can be cast as a dynamic feedback controller using Theorem 2 of Anderson and Matni [12].

#### 4.1.3.1 Sub-optimality guarantees

In this subsection we show that optimizing over FIR approximations incurs only a small degradation in performance relative to the solution to the infinite-horizon problem. In particular, this degradation in performance decays exponentially in the FIR horizon $L$, where the rate of decay is specified by the decay rate of the spectral elements of the optimal closed loop system response $\mathfrak{R}_{A_\star+B_\star K_\star}$.

Before proceeding, we introduce additional concepts and notation needed to formalize guarantees in the FIR setting. A linear-time-invariant transfer function is stable if and only if it is exponentially stable, i.e., $\mathbf{\Phi} = \sum_{t=0}^{\infty} z^{-t}\Phi(t) \in \mathcal{RH}_\infty$ if and only if there exists positive values $C$ and $\rho \in [0,1)$ such that for every spectral element $\Phi(t)$, $t \geq 0$, it holds that

$$\|\Phi(t)\|_2 \leq C\rho^t. \tag{4.1.23}$$

In what follows, we pick $C_\star$ and $\rho_\star$ to be any such constants satisfying $\|\mathfrak{R}_{A_\star+B_\star K_\star}(t)\|_2 \leq C_\star \rho_\star^t$ for all $t \geq 0$.

We introduce a version of the optimization problem (4.1.14) with a finite number of decision variables:

$$\begin{aligned} \text{minimize}_{\gamma\in[0,1)} \frac{1}{1-\gamma} \min_{\mathbf{\Phi}_x,\mathbf{\Phi}_u,V} & \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} \right\|_{\mathcal{H}_2} \\ & \text{s.t.} \begin{bmatrix} zI - \widehat{A} & -\widehat{B} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} = I + \frac{1}{z^L} V, \\ & \left\| \begin{bmatrix} \frac{\epsilon_A}{\sqrt{\alpha}} \mathbf{\Phi}_x \\ \frac{\epsilon_B}{\sqrt{1-\alpha}} \mathbf{\Phi}_u \end{bmatrix} \right\|_{\mathcal{H}_\infty} + \|V\|_2 \leq \gamma \\ & \mathbf{\Phi}_x = \sum_{t=1}^{L} \frac{1}{z^t} \Phi_x(t), \; \mathbf{\Phi}_u = \sum_{t=1}^{L} \frac{1}{z^t} \Phi_u(t). \end{aligned} \tag{4.1.24}$$

In this optimization problem we search over finite response transfer functions $\mathbf{\Phi}_x$ and $\mathbf{\Phi}_u$. Given a feasible solution $\mathbf{\Phi}_x$, $\mathbf{\Phi}_u$ of problem (4.1.24), we can implement the controller $\mathbf{K}_L = \mathbf{\Phi}_u \mathbf{\Phi}_x^{-1}$ with an equivalent state-space representation $(A_K, B_K, C_K, D_K)$ using the response elements $\{\Phi_x(k)\}_{k=1}^L$ and $\{\Phi_u(k)\}_{k=1}^L$ via Theorem 2 of [12].

The slack term $V$ accounts for the error introduced by truncating the infinite response transfer functions of problem (4.1.14). Intuitively, if the truncated tail is sufficiently small, then the effects of this approximation should be negligible on performance. The next result formalizes this intuition.

**Theorem 4.1.10.** *Set $\alpha = 1/2$ in (4.1.24) and let $C_\star > 0$ and $\rho_\star \in [0, 1)$ be such that $\|\mathfrak{R}_{(A_\star+B_\star K_\star)}(t)\|_2 \leq C_\star \rho_\star^t$ for all $t \geq 0$. Then, if $\mathbf{K}_L$ is synthesized via (4.1.24), the relative error in the LQR cost is*

$$\frac{J(A_\star, B_\star, \mathbf{K}_L) - J_\star}{J_\star} \leq 10(\epsilon_A + \epsilon_B \|K_\star\|_2) \|\mathfrak{R}_{A_\star+B_\star K_\star}\|_{\mathcal{H}_\infty},$$

*as long as*

$$\epsilon_A + \epsilon_B \|K_\star\|_2 \leq \frac{1-\rho_\star}{10 C_\star} \quad \textit{and} \quad L \geq \frac{4 \log\left(\frac{C_\star}{(\epsilon_A+\epsilon_B\|K_\star\|_2)\|\mathfrak{R}_{A_\star+B_\star K_\star}\|_{\mathcal{H}_\infty}}\right)}{1-\rho_\star}.$$

The proof of this result is conceptually the same as that of the infinite horizon setting. The main difference is that care must be taken to ensure that the approximation horizon $L$ is sufficiently large so as to ensure stability and performance of the resulting controller. From the theorem statement, we see that for such an appropriately chosen FIR approximation horizon $L$, our performance bound is the same, up to universal constants, to that achieved

by the solution to the infinite horizon problem. Furthermore, the approximation horizon $L$ only needs to grow logarithmically with respect to one over the estimation rate in order to preserve the same statistical rate as the controller produced by the infinite horizon problem.

Now, we turn to the proof of Theorem 4.1.10. To understand the effect of restricting the optimization to FIR transfer functions we need to understand the decay of the transfer functions $\mathfrak{R}_{\widehat{A}+\widehat{B}K_\star}$ and $K_\star\mathfrak{R}_{\widehat{A}+\widehat{B}K_\star}$. To this end we consider $C_\star > 0$ and $\rho_\star \in (0,1)$ such that $\|(A_\star + B_\star K_\star)^t\|_2 \le C_\star \rho_\star^t$ for all $t \ge 0$. Such $C_\star$ and $\rho_\star$ exist because $K_\star$ stabilizes the system $(A_\star, B_\star)$. The next lemma quantifies how well $K_\star$ stabilizes the system $(\widehat{A}, \widehat{B})$ when the estimation error is small.

**Lemma 4.1.11.** *Suppose* $\epsilon_A + \epsilon_B\|K_\star\|_2 \le \frac{1-\rho_\star}{2C_\star}$. *Then,*

$$\|(\widehat{A} + \widehat{B}K_\star)^t\|_2 \le C_\star\left(\frac{1+\rho_\star}{2}\right)^t \text{, for all } t \ge 0.$$

*Proof.* The claim is obvious when $t = 0$. Fix an integer $t \ge 1$ and denote $M = A_\star + B_\star K_\star$. Then, if $\Delta = \Delta_A + \Delta_B K_\star$, we have $\widehat{A} + \widehat{B}K_\star = M + \Delta$.

Consider the expansion of $(M+\Delta)^t$ into $2^t$ terms. Label all these terms as $T_{i,j}$ for $i = 0, ..., t$ and $j = 1, ..., \binom{t}{i}$ where $i$ denotes the degree of $\Delta$ in the term. Since $\Delta$ has degree $i$ in $T_{i,j}$, the term $T_{i,j}$ has the form $M^{\alpha_1}\Delta M^{\alpha_2}\Delta \dots \Delta M^{\alpha_{i+1}}$, where each $\alpha_k$ is a non-negative interger and $\sum_k \alpha_k = t - i$. Then, using the fact that $\|M^k\|_2 \le C_\star\rho_\star^k$ for all $k \ge 0$, we have $\|T_{i,j}\|_2 \le C^{i+1}\rho^{t-i}\|\Delta\|_2^i$. Hence by triangle inequality:

$$\begin{aligned}
\|(M+\Delta)^t\|_2 &\le \sum_{i=0}^{t}\sum_j \|T_{i,j}\|_2 \\
&\le \sum_{i=0}^{t}\binom{t}{i} C_\star^{i+1}\rho_\star^{t-i}\|\Delta\|_2^i \\
&= C_\star\sum_{i=0}^{t}\binom{t}{i}(C_\star\|\Delta\|_2)^i\rho_\star^{t-i} \\
&= C_\star(C_\star\|\Delta\|_2 + \rho_\star)^t \\
&\le C_\star\left(\frac{1+\rho_\star}{2}\right)^t,
\end{aligned}$$

where the last inequality uses the fact $\|\Delta\|_2 \le \epsilon_A + \epsilon_B\|K_\star\|_2 \le \frac{1-\rho_\star}{2C_\star}$. □

For the remainder of this discussion, we use the following notation to denote the restriction of a system response to its first $L$ time-steps:

$$\mathbf{\Phi}_x(1:L) = \sum_{t=1}^{L}\frac{1}{z^t}\Phi_x(t),\ \mathbf{\Phi}_u(1:L) = \sum_{t=1}^{L}\frac{1}{z^t}\Phi_u(t). \tag{4.1.25}$$

To prove Theorem 4.1.10 we must relate the optimal controller $K_\star$ with the optimal solution of the optimization problem (4.1.24). In the next lemma we use $K_\star$ to construct a feasible solution for problem (4.1.24). As before, we denote $\zeta = (\epsilon_A + \epsilon_B\|K_\star\|_2)\|\mathfrak{R}_{A_\star+B_\star K_\star}\|_{\mathcal{H}_\infty}$.

**Lemma 4.1.12.** *Set $\alpha = 1/2$ in problem (4.1.24), and assume that $\epsilon_A + \epsilon_B\|K_\star\|_2 \le \frac{1-\rho_\star}{2C_\star}$, $\zeta < 1/5$, and*

$$L \ge \frac{4\log\left(\frac{C_\star}{\zeta}\right)}{1-\rho_\star}. \tag{4.1.26}$$

*Then, optimization problem (4.1.24) is feasible, and the following is one such feasible solution:*

$$\widetilde{\mathbf{\Phi}}_x = \mathfrak{R}_{\widehat{A}+\widehat{B}K_\star}(1:L), \quad \widetilde{\mathbf{\Phi}}_u = K_\star\mathfrak{R}_{\widehat{A}+\widehat{B}K_\star}(1:L), \quad \widetilde{V} = -\mathfrak{R}_{\widehat{A}+\widehat{B}K_\star}(L+1), \quad \tilde{\gamma} = \frac{4\zeta}{1-\zeta}. \tag{4.1.27}$$

*Proof.* From Lemma 4.1.11 and the assumption on $\zeta$ we have that $\|(\widehat{A}+\widehat{B}K_\star)^t\|_2 \le C_\star\left(\frac{1+\rho_\star}{2}\right)^t$ for all $t \ge 0$. In particular, since $\mathfrak{R}_{\widehat{A}+\widehat{B}K_\star}(L+1) = (\widehat{A}+\widehat{B}K_\star)^L$, we have $\|\widetilde{V}\| = \|(\widehat{A}+\widehat{B}K_\star)^L\| \le C_\star\left(\frac{1+\rho_\star}{2}\right)^L \le \zeta$. The last inequality is true because we assumed $L$ is sufficiently large.

Once again, since $\mathfrak{R}_{\widehat{A}+\widehat{B}K_\star}(L+1) = (\widehat{A}+\widehat{B}K_\star)^L$, it can be easily seen that our choice of $\widetilde{\mathbf{\Phi}}_x$, $\widetilde{\mathbf{\Phi}}_u$, and $\widetilde{V}$ satisfy the linear constraint of problem (4.1.24). It remains to prove that

$$\sqrt{2}\left\|\begin{bmatrix}\epsilon_A\mathbf{\Phi}_x\\ \epsilon_B\mathbf{\Phi}_u\end{bmatrix}\right\|_{\mathcal{H}_\infty} + \|\widetilde{V}\|_2 \le \tilde{\gamma} < 1.$$

The second inequality holds because of our assumption on $\zeta$. We already know that $\|\widetilde{V}\|_2 \le \zeta$. Now, we bound:

$$\begin{aligned}\left\|\begin{bmatrix}\epsilon_A\widetilde{\mathbf{\Phi}}_x\\ \epsilon_B\widetilde{\mathbf{\Phi}}_u\end{bmatrix}\right\|_{\mathcal{H}_\infty} &\le (\epsilon_A + \epsilon_B\|K_\star\|_2)\|\mathfrak{R}_{\widehat{A}+\widehat{B}K_\star}(1:L)\|_{\mathcal{H}_\infty}\\ &\le (\epsilon_A + \epsilon_B\|K_\star\|_2)(\|\mathfrak{R}_{\widehat{A}+\widehat{B}K_\star}\|_{\mathcal{H}_\infty} + \|\mathfrak{R}_{\widehat{A}+\widehat{B}K_\star}(L+1:\infty)\|_{\mathcal{H}_\infty}).\end{aligned}$$

These inequalities follow from the definition of $(\widetilde{\mathbf{\Phi}}_x, \widetilde{\mathbf{\Phi}}_u)$ and the triangle inequality.

Now, we recall that $\mathfrak{R}_{\widehat{A}+\widehat{B}K_\star} = \mathfrak{R}_{A_\star+B_\star K_\star}(I+\mathbf{\Delta})^{-1}$, where $\mathbf{\Delta} = -(\Delta_A+\Delta_B K_\star)\mathfrak{R}_{A_\star+B_\star K_\star}$. Since $\|\mathbf{\Delta}\|_{\mathcal{H}_\infty} \le \zeta$ (due to Proposition 4.1.5), we have $\|\mathfrak{R}_{\widehat{A}+\widehat{B}K_\star}\|_{\mathcal{H}_\infty} \le \frac{1}{1-\zeta}\|\mathfrak{R}_{A_\star+B_\star K_\star}\|_{\mathcal{H}_\infty}$.

We can upper bound

$$\begin{aligned}\|\mathfrak{R}_{\widehat{A}+\widehat{B}K_\star}(L+1:\infty)\|_{\mathcal{H}_\infty} &\le \sum_{t=L+1}^{\infty}\|\mathfrak{R}_{\widehat{A}+\widehat{B}K_\star}(t)\|_2 \le C_\star\left(\frac{1+\rho_\star}{2}\right)^L\sum_{t=0}^{\infty}\left(\frac{1+\rho_\star}{2}\right)^t\\ &= \frac{2C_\star}{1-\rho_\star}\left(\frac{1+\rho_\star}{2}\right)^L.\end{aligned}$$

Then, since we assumed that $\epsilon_A$ and $\epsilon_B$ are sufficiently small and that $L$ is sufficiently large, we obatin

$$(\epsilon_A + \epsilon_B \|K_\star\|_2) \|\mathfrak{R}_{\widehat{A}+\widehat{B}K_\star}(L+1:\infty)\|_{\mathcal{H}_\infty} \leq \zeta.$$

Therefore,

$$\left\| \begin{bmatrix} \epsilon_A \widetilde{\mathbf{\Phi}}_x \\ \epsilon_B \widetilde{\mathbf{\Phi}}_u \end{bmatrix} \right\|_{\mathcal{H}_\infty} \leq \frac{\zeta}{1-\zeta} + \zeta \leq \frac{2\zeta}{1-\zeta}.$$

The conclusion follows. □

*Proof of Theorem 4.1.10.* As all of the assumptions of Lemma 4.1.12 are satisfied, optimization problem (4.1.24) is feasible. We denote $(\mathbf{\Phi}_x^\star, \mathbf{\Phi}_u^\star, V_\star, \gamma_\star)$ the optimal solution of problem (4.1.24). We denote

$$\widehat{D} := \Delta_A \mathbf{\Phi}_x^\star + \Delta_B \mathbf{\Phi}_u^\star + \frac{1}{z^L} V_\star.$$

Then, we have

$$\begin{bmatrix} zI - A_\star & -B_\star \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x^\star \\ \mathbf{\Phi}_u^\star \end{bmatrix} = I + \widehat{D}.$$

Applying the triangle inequality, and leveraging Proposition 4.1.5, we can verify that

$$\|\widehat{D}\|_{\mathcal{H}_\infty} \leq \sqrt{2} \left\| \begin{bmatrix} \epsilon_A \mathbf{\Phi}_x^\star \\ \epsilon_B \mathbf{\Phi}_u^\star \end{bmatrix} \right\|_{\mathcal{H}_\infty} + \|V_\star\|_2 \leq \gamma_\star < 1,$$

where the last two inequalities are true because the optimal solution is a feasible point of the optimization problem (4.1.24).

We now apply Lemma 4.1.4 to characterize the response achieved by the FIR approximate controller $\mathbf{K}_L$ on the true system $(A_\star, B_\star)$:

$$\begin{aligned} J(A_\star, B_\star, \mathbf{K}_L) &= \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x^\star \\ \mathbf{\Phi}_u^\star \end{bmatrix} (I + \widehat{D})^{-1} \right\|_{\mathcal{H}_2} \\ &\leq \frac{1}{1-\gamma_\star} \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x^\star \\ \mathbf{\Phi}_u^\star \end{bmatrix} \right\|_{\mathcal{H}_2}. \end{aligned}$$

Denote by $(\widetilde{\mathbf{\Phi}}_x, \widetilde{\mathbf{\Phi}}_u, \widetilde{V}, \tilde{\gamma})$ the feasible solution constructed in Lemma 4.1.12, and let $J_L(\widehat{A}, \widehat{B}, K_\star)$ denote the truncation of the LQR cost achieved by controller $K_\star$ on system $(\widehat{A}, \widehat{B})$ to its first $L$ time-steps.

Then,

$$\begin{aligned}\frac{1}{1-\gamma_\star}\left\|\begin{bmatrix}Q^{\frac{1}{2}} & 0\\ 0 & R^{\frac{1}{2}}\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}_x^\star\\ \mathbf{\Phi}_u^\star\end{bmatrix}\right\|_{\mathcal{H}_2} &\le \frac{1}{1-\tilde{\gamma}}\left\|\begin{bmatrix}Q^{\frac{1}{2}} & 0\\ 0 & R^{\frac{1}{2}}\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{\Phi}}_x\\ \widetilde{\mathbf{\Phi}}_u\end{bmatrix}\right\|_{\mathcal{H}_2}\\ &= \frac{1}{1-\tilde{\gamma}}J_L(\widehat{A},\widehat{B},K_\star)\\ &\le \frac{1}{1-\tilde{\gamma}}J(\widehat{A},\widehat{B},K_\star)\\ &\le \frac{1}{1-\tilde{\gamma}}\frac{1}{1-\|\mathbf{\Delta}\|_{\mathcal{H}_\infty}}J_\star,\end{aligned}$$

where $\mathbf{\Delta} = -(\Delta_A + \Delta_B K_\star)\mathfrak{R}_{A_\star+B_\star K_\star}$. The first inequality follows from the optimality of $(\mathbf{\Phi}_x^\star, \mathbf{\Phi}_u^\star, V_\star, \gamma_\star)$, the equality and second inequality from the fact that $(\widetilde{\mathbf{\Phi}}_x, \widetilde{\mathbf{\Phi}}_u)$ are truncations of the response of $K_\star$ on $(\widehat{A}, \widehat{B})$ to the first $L$ time steps, and the final inequality by following similar arguments to the proof of Theorem 4.1.8, and in applying Theorem 4.1.2.

Noting that

$$\|\mathbf{\Delta}\|_{\mathcal{H}_\infty} = \|(\Delta_A + \Delta_B K_\star)\mathfrak{R}_{A_\star+B_\star K_\star}\|_{\mathcal{H}_\infty} \le \zeta < 1,$$

we then have that

$$J(A_\star, B_\star, \mathbf{K}_L) \le \frac{1}{1-\tilde{\gamma}}\frac{1}{1-\zeta}J_\star,$$

Recalling that $\tilde{\gamma} = \frac{4\zeta}{1-\zeta}$, we obtain

$$\frac{J(A_\star, B_\star, \mathbf{K}_L) - J_\star}{J_\star} \le \frac{1-\zeta}{1-5\zeta}\frac{1}{1-\zeta} - 1 = \frac{5\zeta}{(1-5\zeta)} \le 10\zeta,$$

where the last equality is true when $\zeta \le 1/10$. The conclusion follows. □

## 4.2 Certainty equivalence

We analyze the *certainty equivalence approach*: use the estimates $(\widehat{A}, \widehat{B})$ to solve the optimization problem (4.0.1) while disregarding the modeling error, and use the resulting controller on the true system $(A_\star, B_\star)$. We interchangeably refer to the resulting policy as the *certainty equivalent controller* or, following Dean et al. [41], the *nominal controller*. We denote by $\widehat{P}$ the solution to the Riccati equation (4.0.2) associated with the parameters $(\widehat{A}, \widehat{B})$ and let $\widehat{K}$ be the corresponding controller. We denote by $J(A, B, K)$ the cost (4.0.1) obtained by using the actions $\mathbf{u}_t = K\mathbf{x}_t$ on the system $(A, B)$, and we use $\widehat{J}$ and $J_\star$ to denote $J(A_\star, B_\star, \widehat{K})$ and $J(A_\star, B_\star, K_\star)$, respectively.

Let $\varepsilon \geq 0$ such that $\|A_\star - \widehat{A}\| \leq \varepsilon$ and $\|B_\star - \widehat{B}\| \leq \varepsilon$. (Here and throughout this work we use $\|\cdot\|$ to denote the Euclidean norm for vectors as well as the spectral (operator) norm for matrices.) Dean et al. [41] introduced a robust controller that achieves $\widehat{J} - J_\star \leq C_1(A_\star, B_\star, Q, R)\varepsilon$ for some complexity term $C_1(A_\star, B_\star, Q, R)$ that depends on the problem parameters. We show that the nominal controller $\mathbf{u}_t = \widehat{K}\mathbf{x}_t$ achieves $\widehat{J} - J_\star \leq C_2(A_\star, B_\star, Q, R)\varepsilon^2$. Both results require $\varepsilon$ to be sufficiently small (as a function of the problem parameters) and it is important to note that $\varepsilon$ must be much smaller for the nominal controller to be guaranteed to stabilize the system than for the robust controller proposed by Dean et al. [41]. However, our result shows that once the estimation error $\varepsilon$ is small enough, the nominal controller performs better: the sub-optimality gap scales as $\mathcal{O}(\varepsilon^2)$ versus $\mathcal{O}(\varepsilon)$. Both the more stringent requirement on $\varepsilon$ and better performance of nominal control compared to robust control, when the estimation error is sufficiently small, were observed empirically by Dean et al. [41].

Before we can formally state our result we need to introduce a few more concepts and assumptions. It is common to assume that the cost matrices $Q$ and $R$ are positive definite. Under an additional observability assumption, this condition can be relaxed to $Q$ being positive semidefinite.

**Assumption 7.** *The cost matrices $Q$ and $R$ are positive definite. Since scaling both $Q$ and $R$ does not change the optimal controller $K_\star$, we can assume without loss of generality that $\sigma_{\min}(R) \geq 1$.*

A square matrix $M$ is *stable* if its spectral radius $\rho(M)$ is (strictly) smaller than one. Recall that the spectral radius is defined as $\rho(M) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } M\}$. A linear dynamical system $(A, B)$ in feedback with $K$ is fully described by the *closed loop matrix* $A + BK$. More precisely, in this case $\mathbf{x}_{t+1} = (A + BK)\mathbf{x}_t + \mathbf{w}_t$. For a static linear controller $\mathbf{u}_t = K\mathbf{x}_t$ to achieve finite LQR cost it is necessary and sufficient that the closed loop matrix is stable.

In order to quantify the growth or decay of powers of a square matrix $M$, we define

$$\tau(M, \rho) := \sup\left\{\|M^k\|\rho^{-k} \colon k \geq 0\right\}. \tag{4.2.1}$$

In other words, $\tau(M,\rho)$ is the smallest value such that $\|M^k\| \leq \tau(M,\rho)\rho^k$ for all $k \geq 0$. We note that $\tau(M,\rho)$ might be infinite, depending on the value of $\rho$, and it is always greater or equal than one. If $\rho$ is larger than $\rho(M)$, we are guaranteed to have a finite $\tau(M,\rho)$ (this is a consequence of Gelfand's formula). In particular, if $M$ is a stable matrix, we can choose $\rho < 1$ such that $\tau(M,\rho)$ is finite. Also, we note that $\tau(M,\rho)$ is a decreasing function of $\rho$; if $\rho \geq \|M\|$, we have $\tau(M,\rho) = 1$. At a high level, the quantity $\tau(M,\rho)$ measures the degree of transient response of the linear system $\mathbf{x}_{t+1} = M\mathbf{x}_t + \mathbf{w}_t$. In particular, when $M$ is stable, $\tau(M,\rho)$ can be upper bounded by the $\mathcal{H}_\infty$-norm of the system defined by $M$, which is the $\ell_2$ to $\ell_2$ operator norm of the system and a fundamental quantity in robust control [see 157, for more details].

Throughout this work we use the quantities $\Gamma_\star := 1 + \max\{\|A_\star\|, \|B_\star\|, \|P_\star\|, \|K_\star\|\}$ and $L_\star := A_\star + B_\star K_\star$. We use $\Gamma_\star$ as a uniform upper bound on the spectral norms of the relevant matrices for the sake of algebraic simplicity. We are ready to state our meta theorem.

**Theorem 4.2.1.** *Suppose $p \leq n$. Let $\gamma > 0$ such that $\rho(L_\star) \leq \gamma < 1$. Also, let $\varepsilon > 0$ such that $\|\widehat{A} - A_\star\| \leq \varepsilon$ and $\|\widehat{B} - B_\star\| \leq \varepsilon$ and assume $\|\widehat{P} - P_\star\| \leq f(\varepsilon)$ for some function $f$ such that $f(\varepsilon) \geq \varepsilon$. Then, under Assumption 7 the certainty equivalent controller $\mathbf{u}_t = \widehat{K}\mathbf{x}_t$ achieves*

$$\widehat{J} - J_\star \leq 200\,\sigma_w^2\, p\,\Gamma_\star^9\,\frac{\tau(L_\star,\gamma)^2}{1-\gamma^2} f(\varepsilon)^2, \tag{4.2.2}$$

*as long as $f(\varepsilon)$ is small enough so that the right hand side is smaller than $\sigma_w^2$.*

In Section 4.2.4 we present two upper bounds $f(\varepsilon)$ on $\|\widehat{P} - P_\star\|$: one based on a proof technique proposed by Konstantinov et al. [79] and one based on our direct approach. Both of these upper bounds satisfy $f(\varepsilon) = \mathcal{O}(\varepsilon)$ for $\varepsilon$ sufficiently small. For simplicity, in this section we only specialize our meta-theorem (Theorem 4.2.1) using the perturbation result from our direct approach.

To state a specialization of Theorem 4.2.1 we need a few more concepts. A linear system $(A,B)$ is called *controllable* when the *controllability matrix* $\begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}$ has full row rank. Controllability is a fundamental concept in control theory; it states that there exists a sequence of inputs to the system $(A,B)$ that moves it from any starting state to any final state in at most $n$ steps. In this work we quantify how controllable a linear system is. We denote, for any integer $\ell \geq 1$, the matrix $\mathcal{C}_\ell := \begin{bmatrix} B & AB & \dots & A^{\ell-1}B \end{bmatrix}$ and call the system *$(\ell,\nu)$-controllable* if the $n$-th singular value of $\mathcal{C}_\ell$ is greater or equal than $\nu$, i.e. $\sigma_{\min}(\mathcal{C}_\ell) = \sqrt{\lambda_{\min}\left(\mathcal{C}_\ell \mathcal{C}_\ell^\top\right)} \geq \nu$. Intuitively, the larger $\nu$ is, the less control effort is needed to move the system between two different states.

**Assumption 8.** *We assume the unknown system $(A_\star, B_\star)$ is $(\ell,\nu)$-controllable, with $\nu > 0$.*

Assumption 8 was used in a different context by Cohen et al. [36]. For any controllable system and any $\ell \geq n$ there exists $\nu > 0$ such that the system is $(\ell,\nu)$-controllable. Therefore,

$(\ell, \nu)$-controllability is really not much stronger of an assumption than controllability. As $\ell$ grows minimum singular value $\sigma_{\min}(\mathcal{C}_\ell)$ also grows and therefore a larger $\nu$ can be chosen so that the system is still $(\ell, \nu)$ controllable.

Note that controllability is not necessary for LQR to have a well-defined solution: the weaker requirement is that of *stabilizability*, in which there exists a feedback matrix $K$ so that $A_\star + B_\star K$ is stable. The result of Dean et al. [41] only requires stabilizability. While our upper bound on $\|\widehat{P} - P_\star\|$ requires controllability, the result of Konstantinov et al. [79] only requires stabilizability. However, our upper bound on $\|\widehat{P} - P_\star\|$ is sharper for some classes of systems (see Section 4.2.4). Together with Theorem 4.2.1, our perturbation result, presented in Section 4.2.4, yields the following guarantee.

**Theorem 4.2.2.** *Suppose that $p \le n$. Let $\rho$ and $\gamma$ be two real values such that $\rho(A_\star) \le \rho$ and $\rho(L_\star) \le \gamma < 1$. Also, let $\varepsilon > 0$ such that $\|\widehat{A} - A_\star\| \le \varepsilon$ and $\|\widehat{B} - B_\star\| \le \varepsilon$ and define $\beta = \max\{1, \varepsilon\tau(A_\star, \rho) + \rho\}$. Under Assumptions 7 and 8, the certainty equivalent controller $\mathbf{u}_t = \widehat{K}\mathbf{x}_t$ satisfies the suboptimality gap*

$$\widehat{J} - J_\star \le \mathcal{O}(1)\, \sigma_w^2\, p\, \ell^5\, \Gamma_\star^{15}\, \tau(A_\star, \rho)^6 \beta^{4(\ell-1)}\, \frac{\tau(L_\star, \gamma)^2}{1-\gamma^2} \frac{\max\{\|Q\|^2, \|R\|^2\}}{\min\{\sigma_{\min}(Q)^2, \sigma_{\min}(R)^2\}} \left(1 + \frac{1}{\nu}\right)^2 \varepsilon^2\,, \tag{4.2.3}$$

*as long as the right hand side is smaller than $\sigma_w^2$. Here, $\mathcal{O}(1)$ denotes a universal constant.*

The exact form of Equation 4.2.3, such as the polynomial dependence on $\ell$, $\Gamma_\star$, etc, can be improved at the expense of conciseness of the expression. In our proof we optimized for the latter. The factor $\max\{\|Q\|^2, \|R\|^2\} / \min\{\sigma_{\min}(Q)^2, \sigma_{\min}(R)^2\}$ is the squared condition number of the cost function, a natural quantity in the context of the optimization problem (4.0.1), which can be seen as an infinite dimensional quadratic program with a linear constraint. The term $\frac{\tau(L_\star,\gamma)^2}{1-\gamma^2}$ quantifies the rate at which the optimal controller drives the state towards zero. Generally speaking, the less stable the optimal closed loop system is, the larger this term becomes.

An interesting trade-off arises between the factor $\ell^5\beta^{4(\ell-1)}$ (which arises from upper bounding perturbations of powers of $A_\star$ on a time interval of length $\ell$) and the factor $\nu$ (the lower bound on $\sigma_{\min}(\mathcal{C}_\ell)$), which is increasing in $\ell$. Hence, the parameter $\ell$ should be seen as a free-parameter that can be tuned to minimize the right hand side of (4.2.3). Now, we specialize Theorem 4.2.2 to a few cases.

**Case: $A_\star$ is contractive, i.e. $\|A_\star\| < 1$.** In this case, we can choose $\rho = \|A_\star\|$ and $\varepsilon$ small enough so that $\varepsilon \le 1 - \|A_\star\|$. Then, (4.2.3) simplifies to:

$$\widehat{J} - J_\star \le \mathcal{O}(1)\, p\, \sigma_w^2\, \ell^5\, \Gamma_\star^{15} \frac{\tau(L_\star, \gamma)^2}{1-\gamma^2} \frac{\max\{\|Q\|^2, \|R\|^2\}}{\min\{\sigma_{\min}(Q)^2, \sigma_{\min}(R)^2\}} \left(1 + \frac{1}{\nu}\right)^2 \varepsilon^2\,.$$

**Case: $B_\star$ has rank $n$ .** In this case, we can choose $\ell = 1$. Then, (4.2.3) simplifies to:

$$\widehat{J} - J_\star \le \mathcal{O}(1)\, p\, \sigma_w^2\, \Gamma_\star^{15} \tau(A_\star, \rho)^6 \frac{\tau(L_\star, \gamma)^2}{1-\gamma^2} \frac{\max\{\|Q\|^2, \|R\|^2\}}{\min\{\sigma_{\min}(Q)^2, \sigma_{\min}(R)^2\}} \left(1 + \frac{1}{\nu}\right)^2 \varepsilon^2 .$$

### 4.2.1 Comparison to robust LQR

In Section 4.1 we saw that when our robust synthesis procedure is run with estimates $(\widehat{A}, \widehat{B})$ satisfying $\max\{\|\widehat{A} - A_\star\|, \|\widehat{B} - B_\star\|\} \le \varepsilon \le [5(1 + \|K_\star\|)\Psi_\star]^{-1}$, the resulting controller satisfies:

$$\widehat{J} - J_\star \le 10(1 + \|K_\star\|)\Psi_\star J_\star \varepsilon + \mathcal{O}(\varepsilon^2) . \tag{4.2.4}$$

Here, the quantity $\Psi_\star := \sup_{z \in \mathbb{T}} \|(zI_n - L_\star)^{-1}\|$ is the $\mathcal{H}_\infty$-norm of the optimal closed loop system $L_\star$. In order to compare Equation 4.2.4 to Equation 4.2.3, we upper bound the quantity $\Psi_\star$ in terms of $\tau(L_\star, \gamma)$ and $\gamma$. In particular, by a infinite series expansion of the inverse $(zI_n - L_\star)^{-1}$ we can show $\Psi_\star \le \frac{\tau(L_\star,\gamma)}{1-\gamma}$. Also, we have $J_\star = \sigma_w^2 \mathbf{Tr}(P_\star) \le \sigma_w^2 n \Gamma_\star$. Therefore, Equation 4.2.4 gives us that:

$$\widehat{J} - J_\star \le \mathcal{O}(1) n \sigma_w^2 \Gamma_\star^2 \frac{\tau(L_\star, \gamma)}{1-\gamma} \varepsilon + \mathcal{O}(\varepsilon^2) .$$

We see that the dependence on the parameters $\Gamma_\star$ and $\tau(L_\star, \gamma)$ is significantly milder compared to Equation 4.2.3. Furthermore, this upper bound is valid for larger $\varepsilon$ than the upper bound given in Theorem 4.2.2. Comparing these upper bound suggests that there is a price to pay for obtaining a fast rate, and that in regimes of moderate uncertainty (moderate size of $\varepsilon$), being robust to model uncertainty is important. This observation is supported by the empirical results of Dean et al. [41].

A similar trade-off between slow and fast rates arises in the setting of first-order convex stochastic optimization. The convergence rate $\mathcal{O}(1/\sqrt{T})$ of the stochastic gradient descent method can be improved to $\mathcal{O}(1/T)$ under a strong convexity assumption. However, the performance of stochastic gradient descent, which can achieve a $\mathcal{O}(1/T)$ rate, is sensitive to poorly estimated problem parameters [104]. Similarly, in the case of LQR, the nominal controller achieves a fast rate, but it is much more sensitive to estimation error than the robust controller of Dean et al. [41].

**End-to-end guarantees.** Theorem 4.2.2 can be combined with finite sample learning guarantees as the ones described in Chapter 3 to obtain an end-to-end guarantee similar to Proposition 1.2 of Dean et al. [41]. In general, estimating the transition parameters from $N$ samples yields an estimation error that scales as $\mathcal{O}(1/\sqrt{N})$. Therefore, Theorem 4.2.2 implies that $\widehat{J} - J_\star \le \mathcal{O}(1/N)$ instead of the $\widehat{J} - J_\star \le \mathcal{O}(1/\sqrt{N})$ rate from Proposition 1.2 of Dean et al. [41]. This is similar to the case of linear regression, where $\mathcal{O}(1/\sqrt{N})$ estimation error for the parameters translates to a $\mathcal{O}(1/N)$ *fast rate* for prediction error. Furthermore,

we discussed in Chapter 3 that faster estimation rates are possible for some linear dynamical systems. Theorem 4.2.2 translates such rates into control suboptimality guarantees in a transparent way.

Our result explains the behavior observed in Figure 4 of Dean et al. [41]. The authors propose two procedures for synthesizing robust controllers for LQR with unknown transitions: one which guarantees robustness of the performance gap $\widehat{J} - J_\star$, and one which only guarantees the stability of the closed loop system. Dean et al. [41] observed that the latter performs better in the small estimation error regime, which happens because the robustness constraint of the synthesis procedure becomes inactive when the estimation error is small enough. Then, the second robust synthesis procedure effectively outputs the certainty equivalent controller, which we now know to achieve a fast rate.

### 4.2.2 Implications for the online setting

The regret formulation of adaptive LQR was first proposed by Abbasi-Yadkori and Szepesvári [1]. The task is to design an adaptive algorithm $\{\mathbf{u}_t\}_{t\geq 0}$ to minimize regret, as defined by $\mathsf{Regret}(T) := \sum_{t=1}^{T} \mathbf{x}_t^\top Q \mathbf{x}_t + \mathbf{u}_t^\top R \mathbf{u}_t - T J_\star$. Abbasi-Yadkori and Szepesvári [1] study the performance of optimism in the face of uncertainty (OFU) and show that it has $\widetilde{\mathcal{O}}(\sqrt{T})$ regret, which is nearly optimal for this problem formulation. However, the OFU algorithm requires repeated solutions to a non-convex optimization problem for which no known efficient algorithm exists.

To deal with the computational issues of OFU, Dean et al. [42] propose to analyze the behavior of $\varepsilon$-greedy exploration using the suboptimality gap results shown in Section 4.1. In the context of continuous control, $\varepsilon$-greedy exploration refers to the application of the control law $\mathbf{u}_t = \pi(\mathbf{x}_t, \mathbf{x}_{t-1}, ..., \mathbf{x}_0) + \eta_t$ with $\eta_t \sim \mathcal{N}(0, \sigma_{\eta,t}^2 I_p)$, where $\pi$ is the policy, updated in epochs, and $\sigma_{\eta,t}^2$ is the variance of the exploration noise. Dean et al. [42] set the variance of the exploration noise as $\sigma_{\eta,t}^2 \sim t^{-1/3}$, and show that their method achieves $\widetilde{\mathcal{O}}(T^{2/3})$ regret. They use epochs of size $2^i$ and decompose the regret roughly as $\mathsf{Regret}(T) = \mathcal{O}\left(T(\widehat{J} - J_\star) + T\sigma_{\eta,T}^2\right)$. Since the estimation error of the model parameters scales as $\mathcal{O}((\sigma_{\eta,T}\sqrt{T})^{-1})$, and since the suboptimality gap $\widehat{J} - J_\star$ of the robust controller is linear in the estimation error, we have $\mathsf{Regret}(T) = \mathcal{O}\left(\frac{\sqrt{T}}{\sigma_{\eta,T}} + T\sigma_{\eta,T}^2\right)$. Then, setting $\sigma_{\eta,t}^2 \sim t^{-1/3}$ balances these two terms and yields $\widetilde{\mathcal{O}}(T^{2/3})$ regret. However, Theorem 4.2.2, which states that the gap $\widehat{J} - J_\star$ for the nominal controller depends quadratically on the estimation rate, implies that online certainty equivalent control achieves $\mathsf{Regret}(T) = \mathcal{O}\left(\frac{1}{\sigma_{\eta,T}^2} + T\sigma_{\eta,T}^2\right)$. Here, the optimal variance of the exploration noise scales as $\sigma_{\eta,t}^2 \sim t^{-1/2}$, yielding $\widetilde{\mathcal{O}}(\sqrt{T})$ regret. We note that the observation that certainty equivalence coupled with $\varepsilon$-greedy exploration achieves $\widetilde{\mathcal{O}}(\sqrt{T})$ regret was first made by Faradonbeh et al. [46].

**Corollary 4.2.3.** *(Informal) $\varepsilon$-greedy exploration with exploration schedule $\sigma_{\eta,t}^2 \sim t^{-1/2}$ combined with certainty equivalent control yields an adaptive LQR algorithm with regret bounded as $\widetilde{\mathcal{O}}(\sqrt{T})$.*

### 4.2.3 Proof of a meta-theorem

In this section we prove our meta theorem; we show how an upper bound $\|\widehat{P} - P_\star\| \le f(\varepsilon)$ can be used to quantify the mismatch between the performance of the the nominal controller and the optimal controller. First, we upper bound $\|\widehat{K} - K_\star\|$ and offer a condition on this mismatch size so that $A_\star + B_\star \widehat{K}$ is a stable matrix. The next two optimization results are helpful in proving $\|\widehat{K} - K_\star\|$ is small.

**Lemma 4.2.4.** *Let $f_1, f_2$ be two $\mu$-strongly convex twice differentiable functions. Let $\mathbf{x}_1 = \arg\min_{\mathbf{x}} f_1(\mathbf{x})$ and $\mathbf{x}_2 = \arg\min_{\mathbf{x}} f_2(\mathbf{x})$. Suppose $\|\nabla f_1(\mathbf{x}_2)\| \le \varepsilon$, then $\|\mathbf{x}_1 - \mathbf{x}_2\| \le \frac{\varepsilon}{\mu}$.*

*Proof.* Taylor expanding $\nabla f_1$, we have:

$$\nabla f_1(\mathbf{x}_2) = \nabla f_1(\mathbf{x}_1) + \nabla^2 f_1(\tilde{\mathbf{x}})(\mathbf{x}_2 - \mathbf{x}_1) = \nabla^2 f_1(\tilde{\mathbf{x}})(\mathbf{x}_2 - \mathbf{x}_1)\,.$$

for $\tilde{\mathbf{x}} = t\mathbf{x}_1 + (1-t)\mathbf{x}_2$ with some $t \in [0,1]$. Therefore:

$$\mu\|\mathbf{x}_1 - \mathbf{x}_2\| \le \|\nabla^2 f_1(\tilde{\mathbf{x}})(\mathbf{x}_2 - \mathbf{x}_1)\| = \|\nabla f_1(\mathbf{x}_2)\| \le \varepsilon\,.$$

□

**Lemma 4.2.5.** *Define $f_i(\mathbf{u};\mathbf{x}) = \frac{1}{2}\mathbf{u}^\mathsf{T} R\mathbf{u} + \frac{1}{2}(A_i\mathbf{x} + B_i\mathbf{u})^\mathsf{T} P_i (A_i\mathbf{x} + B_i\mathbf{u})$ for $i = 1, 2$, with $R$, $P_1$, and $P_2$ positive definite matrices. Let $K_i$ be the unique matrix such that $\mathbf{u}_i := \arg\min_{\mathbf{u}} f_i(\mathbf{u};\mathbf{x}) = K_i\mathbf{x}$ for any vector $\mathbf{x}$. Denote $\Gamma := 1 + \max\{\|A_1\|, \|B_1\|, \|P_1\|, \|K_1\|\}$. Suppose there exists $\varepsilon$ such that $0 \le \varepsilon < 1$ and $\|A_1 - A_2\| \le \varepsilon$, $\|B_1 - B_2\| \le \varepsilon$, and $\|P_1 - P_2\| \le \varepsilon$. Then, we have*

$$\|K_1 - K_2\| \le \frac{7\varepsilon\Gamma^3}{\sigma_{\min}(R)}\,.$$

*Proof.* We first compute the gradient $\nabla f_i(\mathbf{u};\mathbf{x})$ with respect to $\mathbf{u}$:

$$\nabla f_i(\mathbf{u};\mathbf{x}) = (B_i^\mathsf{T} P_i B_i + R)\mathbf{u} + B_i^\mathsf{T} P_i A_i \mathbf{x}\,.$$

Now, we observe that:

$$\|B_1^\mathsf{T} P_1 B_1 - B_2^\mathsf{T} P_2 B_2\| \le 7\Gamma^2\varepsilon \;\text{ and }\; \|B_1^\mathsf{T} P_1 A_1 - B_2^\mathsf{T} P_2 A_2\| = 7\Gamma^2\varepsilon.$$

Hence, for any vector $\mathbf{x}$ with $\|\mathbf{x}\| \le 1$, we have

$$\|\nabla f_1(\mathbf{u};\mathbf{x}) - \nabla f_2(\mathbf{u};\mathbf{x})\| \le 7\Gamma^2\varepsilon(\|\mathbf{u}\| + 1).$$

We can bound $\|\mathbf{u}_1\| \le \|K_1\|\|\mathbf{x}\| \le \|K_1\|$. Then, from Lemma 4.2.4 we obtain

$$\sigma_{\min}(R)\|(K_1 - K_2)\mathbf{x}\| = \sigma_{\min}(R)\|\mathbf{u}_1 - \mathbf{u}_2\| \le 7\Gamma^3\varepsilon.$$

□

Recall that $\Gamma_\star := 1 + \max\{\|A_\star\|, \|B_\star\|, \|P_\star\|, \|K_\star\|\}$. Now, we upper bound $\|\widehat{K} - K_\star\|$.

**Proposition 4.2.6.** *Let $\varepsilon > 0$ such that $\|\widehat{A} - A_\star\| \le \varepsilon$ and $\|\widehat{B} - B_\star\| \le \varepsilon$. Also, let $\|\widehat{P} - P_\star\| \le f(\varepsilon)$ for some function $f$ such that $f(\varepsilon) \ge \varepsilon$. Then, under Assumption 7 we have*

$$\|\widehat{K} - K_\star\| \le 7\Gamma_\star^3 f(\varepsilon). \tag{4.2.5}$$

*Let $\gamma$ be a real number such that $\rho(L_\star) < \gamma < 1$. Then, if $f(\varepsilon)$ is small enough so that the right hand side of (4.2.5) is smaller than $\frac{1-\gamma}{2\tau(L_\star,\gamma)}$, we have*

$$\tau\left(A_\star + B_\star K, \frac{1+\gamma}{2}\right) \le \tau(L_\star, \gamma).$$

*Proof.* By our assumptions $\|\widehat{A} - A_\star\|$, $\|\widehat{B} - B_\star\|$, and $\|\widehat{P} - P_\star\|$ are smaller than $f(\varepsilon)$, and $\sigma_{\min}(R) \ge 1$. Then, Lemma 4.2.5 ensures that

$$\|\widehat{K} - K_\star\| \le 7\Gamma_\star^3 f(\varepsilon).$$

Finally, when $\varepsilon$ is small enough so that the right hand side of (4.2.5) is smaller or equal than $\frac{1-\gamma}{2\tau(A_\star+B_\star K_\star,\gamma)}$, we can apply Lemma 4.2.11, presented in Section 4.2.4, to guarantee that $\|(A_\star + B_\star\widehat{K})^k\| \le \tau(A_\star + B_\star K_\star, \gamma)\left(\frac{1+\gamma}{2}\right)^k$ for all $k \ge 0$. □

In order to finish the proof of Theorem 4.2.1 we need to quantify the suboptimality gap $\widehat{J} - J_\star$ in terms of the controller mismatch $\widehat{K} - K_\star$. For a stable matrix $L$ and a symmetric matrix $M$, we let $\mathsf{dlyap}(L, M)$ denote the solution $X$ to the Lyapunov equation $L^\top X L - X + M = 0$. The following lemma offers a useful second order expansion of the average LQR cost.

**Lemma 4.2.7** (Lemma 12 of Fazel et al. [49])**.** *Let $K$ be an arbitrary static linear controller that stabilizes $(A_\star, B_\star)$. Denote $\Sigma(K) := \mathsf{dlyap}((A_\star + B_\star K)^\mathsf{T}, \sigma_w^2 I_n)$ the covariance matrix of the stationary distribution of the closed loop system $A_\star + B_\star K$. We have that:*

$$J(A_\star, B_\star, K) - J_\star = \mathbf{Tr}(\Sigma(K)(K - K_\star)^\mathsf{T}(R + B_\star^\mathsf{T} P_\star B_\star)(K - K_\star))\,. \tag{4.2.6}$$

Now, we have the necessary ingredients to complete the proof of Theorem 4.2.1. Equation 4.2.6 implies:

$$J(A_\star, B_\star, K) - J_\star \le \|\Sigma(K)\|\|R + B_\star^\mathsf{T} P_\star B_\star\|\|K - K_\star\|_F^2\,.$$

Proposition 4.2.6 states that $\widehat{K}$ stabilizes the system $(A_\star, B_\star)$ when the estimation error is small enough. More precisely, under the assumptions of Theorem 4.2.1, we have

$\tau\left(A_\star + B_\star \widehat{K}, \frac{1+\gamma}{2}\right) \le \tau(L_\star, \gamma)$. When $\widehat{L} = A_\star + B_\star \widehat{K}$ is a stable matrix we know that $\Sigma(K) = \sigma^2 \sum_{t\ge 0} (L^\top)^t L^t$. Then, by the triangle inequality we can bound

$$\|\Sigma(K)\| \le \frac{\sigma_w^2 \tau(L_\star,\gamma)^2}{1 - \left(\frac{\gamma+1}{2}\right)^2} \le \frac{4\sigma_w^2 \tau(L_\star,\gamma)^2}{1-\gamma^2}\,.$$

Recalling that $\Gamma_\star := 1 + \max\{\|A_\star\|, \|B_\star\|, \|P_\star\|, \|K_\star\|\}$, we have $\|R + B_\star^\mathsf{T} P_\star B_\star\| \le \Gamma^3$. Then,

$$\begin{aligned} J(K) - J_\star &\le 4\sigma_w^2 \Gamma^3 \frac{\tau(L_\star,\gamma)^2}{1-\gamma^2} \|K - K_\star\|_F^2 \\ &\le 4\sigma_w^2 \min\{n,p\} \Gamma^3 \frac{\tau(L_\star,\gamma)^2}{1-\gamma^2} \|K - K_\star\|^2 \\ &\le 200 \sigma_w^2 d \Gamma^9 \frac{\tau(L_\star,\gamma)^2}{1-\gamma^2} f(\varepsilon)^2, \end{aligned}$$

where we used Proposition 4.2.6 and the assumption on $f(\varepsilon)$.

### 4.2.4 Riccati perturbation theory

As discussed in Section 3.3.2, a key piece of our analysis is bounding the solutions to discrete Riccati equations as we perturb the problem parameters. Specifically, we are interested in quantities $b, L$ such that $\|\widehat{P} - P_\star\| \le L\varepsilon$ if $\varepsilon < b$, where $\varepsilon$ represents a bound on the perturbation. We note that it is not possible to find universal values $b, L$. Consider the systems $(A_\star, B_\star) = (1, \varepsilon)$ and $(\widehat{A}, \widehat{B}) = (1, 0)$; the latter system is not stabilizable and hence $\widehat{P}$ does not even exist. Therefore, $b$ and $L$ must depend on the system parameters.

While there is a long line of work analyzing perturbations of Riccati equations, we are not aware of any result that offers explicit and easily interpretable $b$ and $L$ for a fixed $(A_\star, B_\star, Q, R)$; see Konstantinov et al. [80] for an overview of this literature. In this section, we present two new results for Riccati perturbation which offer interpretable bounds. The first one expands upon the operator-theoretic proof of Konstantinov et al. [79]. In this result we assume the cost matrix $Q$ can also be perturbed, which is needed for our LQG guarantee. In order to be consistent we denote the true cost matrix by $Q_\star$ and the estimated one by $\widehat{Q}$.

**Proposition 4.2.8.** *Let $\gamma \ge \rho(L_\star)$ and also let $\varepsilon$ such that $\|\widehat{A} - A_\star\|$, $\|\widehat{B} - B_\star\|$, and $\|\widehat{Q} - Q_\star\|$ are at most $\varepsilon$. Let $\|\cdot\|_+ = \|\cdot\| + 1$. We assume that $R \succ 0$, $(A_\star, B_\star)$ is stabilizable, $(Q^{1/2}, A_\star)$ observable, and $\sigma_{\min}(P_\star) \ge 1$.*

$$\|\widehat{P} - P_\star\| \le \mathcal{O}(1)\, \varepsilon\, \frac{\tau(L_\star,\gamma)^2}{1-\gamma^2} \|A_\star\|_+^2 \|P_\star\|_+^2 \|B_\star\|_+ \|R^{-1}\|_+,$$

*as long as*

$$\varepsilon \le \mathcal{O}(1) \frac{(1-\gamma^2)^2}{\tau(L_\star,\gamma)^4} \|A_\star\|_+^{-2} \|P_\star\|_+^{-2} \|B_\star\|_+^{-3} \|R^{-1}\|_+^{-2} \min\left\{\|L_\star\|_+^{-2}, \|P_\star\|_+^{-1}\right\}.$$

We note that the assumption $\sigma_{\min}(P_\star) \geq 1$ can be made without loss of generality when the other assumptions are satisfied. When $R \succ 0$ and $(Q^{1/2}, A)$ observable, the value function matrix $P_\star$ is guaranteed to be positive definite. Then, by rescalling $Q$ and $R$ we can ensure that $\sigma_{\min}(P_\star) \geq 1$.

We now present our direct approach, which uses Assumption 8 to give a bound which is sharper for some systems $(A_\star, B_\star)$ then the one provided by Proposition 4.2.8. Recall that any controllable system is always $(\ell, \nu)$-controllable for some $\ell$ and $\nu$.

**Proposition 4.2.9.** *Let $\rho \geq \rho(A_\star)$ and also let $\varepsilon \geq 0$ such that $\|\widehat{A} - A_\star\| \leq \varepsilon$ and $\|\widehat{B} - B_\star\| \leq \varepsilon$. Let $\beta := \max\{1, \varepsilon\tau(A_\star, \rho) + \rho\}$. Under Assumptions 7 and 8 we have*

$$\|\widehat{P} - P_\star\| \leq 32\,\varepsilon\,\ell^{\frac{5}{2}}\tau(A_\star,\rho)^3\beta^{2(\ell-1)}\left(1+\frac{1}{\nu}\right)(1+\|B_\star\|)^2\|P_\star\|\frac{\max\{\|Q\|,\|R\|\}}{\min\{\sigma_{\min}(R),\underline{\sigma}(Q)\}},$$

*as long as $\varepsilon$ is small enough so that the right hand side is smaller or equal than one.*

We return to the proof of this result shortly. For now, we note that Proposition 4.2.9 can also be extended to handle perturbations in the cost matrix $Q$, as we describe in the proof. Proposition 4.2.9 requires an $(\ell, \nu)$-controllable system $(A_\star, B_\star)$, whereas Proposition 4.2.8 only requires a stabilizable system, which is a milder assumption. However, Proposition 4.2.9 can offer a sharper guarantee. For example, consider the linear system with two dimensional states ($n = 2$) given by $A_\star = 1.01 \cdot I_2$ and $B_\star = \begin{bmatrix} 1 & 0 \\ 0 & \beta \end{bmatrix}$. Both $Q$ and $R$ are chosen to be the identity matrix $I_2$. This system $(A_\star, B_\star)$ is readily checked to be $(1, \beta)$-controllable. It is also straightforward to verify that as $\beta$ tends to zero, Proposition 4.2.8 gives a bound of $\|\widehat{P} - P_\star\| = \mathcal{O}(\varepsilon/\beta^4)$, whereas Proposition 4.2.9 gives a sharper bound of $\|\widehat{P} - P_\star\| = \mathcal{O}(\varepsilon/\beta^3)$.

**Proof of Proposition 4.2.8.** Given parameters $(A, B, Q)$ ($R$ is assumed fixed throughout; $Q$ is assumed positive semidefinite throughout) we denote by $F(X, A, B, Q)$ the matrix expression

$$\begin{aligned} F(X, A, B, Q) &= X - A^\top X A + A^\top X B(R + B^\top X B)^{-1} B^\top X A - Q \\ &= X - A^\top X \left(I + BR^{-1}B^\top X\right)^{-1} A - Q\,. \end{aligned} \tag{4.2.7}$$

Then, solving the Riccati equation associated with $(A, B, Q)$ corresponds to finding the unique positive definite matrix $X$ such that $F(X, A, B, Q) = 0$. We denote by $P_\star$ the solution of the Riccati equation corresponding to the true system parameters $(A_\star, B_\star)$ and we denote by $\widehat{P}$ the solution associated with $(\widehat{A}, \widehat{B}, \widehat{Q})$. Our goal is to upper bound $\|\widehat{P} - P_\star\|$ in terms of $\varepsilon$, where $\varepsilon > 0$ such that $\|\widehat{A} - A_\star\| \leq \varepsilon$, $\|\widehat{B} - B_\star\| \leq \varepsilon$, and $\|\widehat{Q} - Q_\star\| \leq \varepsilon$.

We denote $\Delta_P = \widehat{P} - P_\star$. The proof strategy goes as follows. Given the identities $F(P_\star, A_\star, B_\star, Q_\star) = 0$ and $F(\widehat{P}, \widehat{A}, \widehat{B}, \widehat{Q}) = 0$ we construct an operator $\Phi$ such that $\Delta_P$ is its unique fixed point. Then, we show that the fixed point of $\Phi$ must have small norm when $\varepsilon$ is sufficiently small.

We denote $S_\star = B_\star R^{-1} B_\star^\top$ and $\widehat{S} = \widehat{B} R^{-1} \widehat{B}^\top$. Also, recall that $L_\star = A_\star + B_\star K_\star$. For any matrix $X$ such that $I + S_\star(P_\star + X)$ is invertible we have

$$F(P_\star + X, A_\star, B_\star, Q_\star) = X - L_\star^\top X L_\star + L_\star^\top X \left[I + S_\star(P_\star + X)\right]^{-1} S_\star X L_\star. \tag{4.2.8}$$

To check this identity one needs to add $F(P_\star, A_\star, B_\star, Q_\star)$, which is equal to zero, to the right hand side of (4.2.8) and use the identity $(I + B_\star R^{-1} B_\star^\top P_\star)^{-1} A_\star = A_\star + B_\star K_\star$. This last identity can be checked by recalling $K_\star = -(R + B_\star^\top P_\star B_\star)^{-1} B_\star^\top P_\star A_\star$ and using the matrix inversion formulat.

To write (4.2.8) more compactly we define the following two matrix operators

$$\mathcal{T}(X) = X - L_\star^\top X L_\star \quad \text{and} \quad \mathcal{H}(X) = L_\star^\top X \left(I + S_\star(P_\star + X)\right)^{-1} S_\star X L_\star.$$

Then, Equation 4.2.8 becomes $F(P_\star + X, A_\star, B_\star, Q_\star) = \mathcal{T}(X) + \mathcal{H}(X)$. Since Equation 4.2.8 is satisfied by any matrix $X$ with $I + S_\star(P_\star + X)$ invertible, the matrix equation

$$F(P_\star + X, A_\star, B_\star, Q_\star) - F(P_\star + X, \widehat{A}, \widehat{B}, \widehat{Q}) = \mathcal{T}(X) + \mathcal{H}(X) \tag{4.2.9}$$

has a unique symmetric solution $X$ such that $P_\star + X \succeq 0$. That solution is $X = \Delta_P$ because any solution of (4.2.9) must satisfy $F(P_\star + X, \widehat{A}, \widehat{B}, \widehat{Q}) = 0$.

The linear map $\mathcal{T}\colon X \mapsto X - L_\star^\top X L_\star$ has eigenvalues equal to $1 - \lambda_i \lambda_j$, where $\lambda_i$ and $\lambda_j$ are eigenvalues of the closed loop matrix $L_\star$. Since $L_\star$ is a stable matrix, the linear map $\mathcal{T}$ must be invertible. Now, we define the operator

$$\Phi(X) = \mathcal{T}^{-1}\left(F(P_\star + X, A_\star, B_\star, Q_\star) - F(P_\star + X, \widehat{A}, \widehat{B}, \widehat{Q}) - \mathcal{H}(X)\right).$$

Then, solving for $X$ in Equation 4.2.9 is equivalent to finding $X$ satisfying $P_\star + X \succeq 0$ such that $X = \Phi(X)$. Hence, $\Phi$ has a unique symmetric fixed point $X$ such that $P_\star + X \succeq 0$ and that is $X = \Delta_P$. Now, we consider the set

$$\mathcal{S}_\nu := \left\{X\colon \|X\| \le \nu,\, X = X^\top,\, P_\star + X \succeq 0\right\}$$

and we show that for an appropriately chosen $\nu$ the operator $\Phi$ maps $\mathcal{S}_\nu$ into itself and is also a contraction over the set $\mathcal{S}_\nu$. If we show these two properties, $\Phi$ is guaranteed to have a fixed point in the set $\mathcal{S}_\nu$. However, since $\Delta_P$ is the only possible fixed point of $\Phi$ in a set $\mathcal{S}_\nu$ we find $\|\Delta_P\| \le \nu$.

We denote $\Delta_A = \widehat{A} - A_\star$, $\Delta_B = \widehat{B} - B_\star$, $\Delta_Q = \widehat{Q} - Q_\star$, and $\Delta_S = \widehat{S} - S_\star$. By assumption we have $\|\Delta_A\| \le \varepsilon$, $\|\Delta_B\| \le \varepsilon$, $\|\Delta_Q\| \le \varepsilon$. Then, $\|\Delta_S\| \le 3\|B_\star\|\|R^{-1}\|\varepsilon$ because $\varepsilon \le \|B_\star\|$.

**Lemma 4.2.10.** *Suppose the matrices $X$, $X_1$, $X_2$ belong to $\mathcal{S}_\nu$, with $\nu \le \min\{1, \|S_\star\|^{-1}\}$. Furthermore, we assume that $\|\Delta_A\| \le \varepsilon$, $\|\Delta_B\| \le \varepsilon$, and $\|\Delta_Q\| \le \varepsilon$ with $\varepsilon \le \min\{1, \|B_\star\|\}$. Finally, let $\sigma_{\min}(P_\star) \ge 1$. Then*

$$\|\Phi(X)\| \le 3\frac{\tau(L_\star, \gamma)^2}{1 - \gamma^2}\left[\|L_\star\|^2 \|S_\star\| \nu^2 + \varepsilon \|A_\star\|_+^2 \|P_\star\|_+^2 \|B_\star\|_+ \|R^{-1}\|_+\right],$$

$$\|\Phi(X_1) - \Phi(X_2)\| \le 32\frac{\tau(L_\star, \gamma)^2}{1 - \gamma^2}\left[\|L_\star\|^2 \|S_\star\| \nu + \varepsilon \|A_\star\|_+^2 \|P_\star\|_+^3 \|B_\star\|_+^3 \|R^{-1}\|_+^2\right] \|X_1 - X_2\|.$$

*Proof.* We wish to upper bound $\|\Phi(X)\|$ and $\|\Phi(X_1)-\Phi(X_2)\|$ for $X$, $X_1$, and $X_2$ in $\mathcal{S}_\nu$. First, we upper bound the operator norm of the linear operator $\mathcal{T}^{-1}$, the inverse of $\mathcal{T}\colon X \mapsto X - L_\star^\top X L_\star$. Since $L_\star$ is a stable matrix, the linear map $\mathcal{T}$ must be invertible. Moreover, when $L_\star$ is stable and $X - L_\star^\top X L_\star = M$ for some matrix $M$, we know that $X = \sum_{k=0}^\infty (L_\star^k)^\top M L_\star^k$. Therefore, by the triangle inequality, the operator norm of $\mathcal{T}^{-1}$ can be upper bounded by $\|\mathcal{T}^{-1}\| \le \frac{\tau(L_\star,\rho)^2}{1-\rho^2}$. Before we proceed with the rest of the proof we check the following fact.

For any positive semidefinite matrices $M$ and $N$ of the same dimension we have

$$\|N(I+MN)^{-1}\| \le \|N\|. \tag{4.2.10}$$

To check this, we assume that $M$ and $N$ are invertible. If they are not, we can work with the matrices $M+\nu I$ and $N+\nu I$ and take the limit of $\nu$ going to zero. Then, we have $N(I+MN)^{-1} = NN^{-1}(N^{-1}+M)^{-1} = (N^{-1}+M)^{-1} \preceq N$, which proves (4.2.10).

Now, ecall that $\mathcal{H}(X) = L_\star^\top X \left(I + S_\star(P_\star + X)\right)^{-1} S_\star X L_\star$. Then, fact (4.2.10) yields

$$\|\mathcal{H}(X)\| \le \|L_\star\|^2 \|S_\star\| \|X\|^2.$$

We turn our attention to the difference $F(P_\star + X, A_\star, B_\star) - F(P_\star + X, \widehat{A}, \widehat{B})$. We use the notation $P_X$ as a shorthand for $P_\star + X$. Then, by Equation 4.2.7 we find

$$\begin{aligned} F(P_X, \widehat{A}, \widehat{B}, \widehat{Q}) - F(P_X, A_\star, B_\star, Q_\star) &= A_\star^\top P_X (I + S_\star P_X)^{-1} A_\star - \widehat{A}^\top P_X \Big(I + \widehat{S} P_X\Big)^{-1} \widehat{A} - \Delta_Q \\ &= A_\star^\top P_X (I + S_\star P_X)^{-1} \Delta_S P_X (I + \widehat{S} P_X)^{-1} A_\star - A_\star^\top P_X (I + \widehat{S} P_X)^{-1} \Delta_A \\ &\quad - \Delta_A^\top P_X (I + \widehat{S} P_X)^{-1} A_\star - \Delta_A^\top P_X (I + \widehat{S} P_X)^{-1} \Delta_A - \Delta_Q. \end{aligned} \tag{4.2.11}$$

Then,

$$\begin{aligned} &\|F(P_\star + X, \widehat{A}, \widehat{B}, \widehat{Q}) - F(P_\star + X, A_\star, B_\star, Q_\star)\| \\ &\qquad \le \|A_\star\|^2 \|P_X\|^2 \|\Delta_S\| + 2\|A_\star\| \|P_X\| \varepsilon + \|P_X\| \varepsilon^2 + \varepsilon, \end{aligned}$$

where we used fact (4.2.10). Since $X \in \mathcal{S}_\nu$, we know $\|X\| \le \nu$ and hence $\|P_X\| \le \|P_\star\| + \nu$. We assumed that $\nu \le 1/2$ and so $\|P_X\| \le \|P_\star\| + 1$. Now, we know that $\|\Delta_S\| \le 2\|B_\star\| \|R^{-1}\| \varepsilon + \|R^{-1}\| \varepsilon^2$ and since we assumed $\varepsilon \le \|B_\star\|$, we have $\|\Delta_S\| \le 3\|B_\star\| \|R^{-1}\| \varepsilon$. Therefore,

$$\|\Phi(X)\| \le \frac{\tau(L_\star,\rho)^2}{1-\rho^2} \left[\|L_\star\|^2 \|S_\star\| \nu^2 + 3\|A_\star\|_+^2 \|P_\star\|_+^2 \|B_\star\|_+ \|R^{-1}\|_+ \varepsilon\right].$$

We use fact (4.2.10), the assumption $\nu \le \|S_\star\|^{-1}$, and the definition of $\mathcal{H}$ to upper bound

$$\|\mathcal{H}(X_1) - \mathcal{H}(X_2)\| \le \|L_\star\|^2 \left[\|S_\star\|^2 \nu^2 + 2\|S_\star\| \nu\right] \|X_1 - X_2\| \le 3\|L_\star\|^2 \|S_\star\| \nu \|X_1 - X_2\|.$$

Let us denote $\mathcal{G}(X) = F(P_\star + X, \widehat{A}, \widehat{B}, \widehat{Q}) - F(P_\star + X, A_\star, B_\star, Q_\star)$. In order to upper bound $\|\mathcal{G}(X_1) - \mathcal{G}(X_2)\|$ we first upper bound the norm of $(I + S_\star P_X)^{-1}$ and $(I + \widehat{S} P_X)^{-1}$. Since $\|X\| \le \nu \le 1/2$ and since $P_\star \succeq I$, by fact 4.2.10 we get

$$\|(I + S_\star P_X)^{-1}\| = \|P_X^{-1} P_X (I + S_\star P_X)^{-1}\| \le \|P_X^{-1}\| \|P_X (I + S_\star P_X)^{-1}\| \le 2\|P_X\|.$$

Therefore, after some algebraic manipulations, we obtain

$$\|\mathcal{G}(X_1) - \mathcal{G}(X_2)\| \leq 32\varepsilon \|A_\star\|_+^2 \|P_\star\|_+^3 \|B_\star\|_+^3 \|R^{-1}\|_+^2 \|X_1 - X_2\|.$$

□

Now, we choose

$$\nu = 6\,\varepsilon\, \frac{\tau(L_\star, \gamma)^2}{1-\gamma^2} \|A_\star\|_+^2 \|P_\star\|_+^2 \|B_\star\|_+ \|R^{-1}\|_+. \tag{4.2.12}$$

Since $\varepsilon$ is assumed to be small enough, we know

$$\nu \leq \min\left\{ \frac{1-\gamma^2}{128\tau(L_\star,\gamma)^2 \|L_\star\|^2 \|S_\star\|}, \|S_\star\|^{-1}, \frac{1}{2} \right\}.$$

Then, the operator $\Phi$ satisfies $\|\Phi(X_1) - \Phi(X_2)\| \leq \frac{1}{2}\|X_1 - X_2\|$ for all $X_1$ and $X_2$ in $\mathcal{S}_\nu$. Moreover, we have $\|\Phi(X)\| \leq \nu$ for all $X \in \mathcal{S}_\nu$. Since $\nu \leq \sigma_{\min}(P_\star)$, we know that $P_\star + \Phi(X) \succeq 0$

Therefore, $\Phi$ maps $\mathcal{S}_\nu$ into itself and is a contraction over $\mathcal{S}_\nu$. Hence, $\Phi$ has a fixed point in $\mathcal{S}_\nu$ since $\mathcal{S}_\nu$ is a closed set. However, we already argued that the unique fixed point of $\Phi$ is $\Delta_P$. Therefore, $\Delta_P \in \mathcal{S}_\nu$ and $\|\Delta_P\| \leq \nu$. Proposition 4.2.8 is now proven.

**Proof of Proposition 4.2.9.** Since both noisy and noiseless LQR have the same associated Riccati equation and the same optimal controller, we can focus on the noiseless case in this section. Namely, noiseless LQR takes the form

$$\min_{\mathbf{u}} \sum_{t=0}^{\infty} \mathbf{x}_t^\top Q \mathbf{x}_t + \mathbf{u}_t^\top R \mathbf{u}_t \text{ , where } \mathbf{x}_{t+1} = A_\star \mathbf{x}_t + B_\star \mathbf{u}_t,$$

for a given initial state $\mathbf{x}_0$. Then, we know that the cost achieved by the optimal controller when the system is initialized at $\mathbf{x}_0$ is equal to $\mathbf{x}_0^\top P_\star \mathbf{x}_0$.

We denote by $J(A, B, \mathbf{x}_0, \{\mathbf{u}_t\}_{t\geq 0})$ the cost achieved on a linear system $(A, B)$ initialized at $\mathbf{x}_0$ by the input sequence $\{\mathbf{u}_t\}_{t\geq 0}$. When the input sequence is given by a time invariant linear gain matrix $K$ we slightly abuse notation and denote the cost by $J(A, B, \mathbf{x}_0, K)$. In this case, $J(A, B, \mathbf{x}_0, K) = \mathbf{x}_0^\top P \mathbf{x}_0$, where $P$ is the solution to the associated Riccati equation.

Now, let $\mathbf{x}_0$ be an arbitrary unit state vector in $\mathbb{R}^n$. Then,

$$\begin{aligned}
\mathbf{x}_0^\top \widehat{P} \mathbf{x}_0 - \mathbf{x}_0^\top P_\star \mathbf{x}_0 &= J(\widehat{A}, \widehat{B}, \mathbf{x}_0, \widehat{K}) - J(A_\star, B_\star, \mathbf{x}_0, K_\star) \\
&\leq J(\widehat{A}, \widehat{B}, \mathbf{x}_0, \{\widehat{\mathbf{u}}_t\}_{t\geq 0}) - J(A_\star, B_\star, \mathbf{x}_0, K_\star)
\end{aligned}$$

for any sequence of inputs $\{\widehat{\mathbf{u}}_t\}_{t\geq 0}$. We denote by $\hat{\mathbf{x}}_t$ the states produced by $\widehat{\mathbf{u}}_t$ on the system $(\widehat{A}, \widehat{B})$ and by $\mathbf{x}_t$ and $\mathbf{u}_t$ the states and actions obtained on the system $(A_\star, B_\star)$ when the

optimal controller $\mathbf{u}_t = K_\star \mathbf{x}_t$ is used. To prove Proposition 4.2.9 we choose a sequence of actions $\{\widehat{\mathbf{u}}_t\}_{t\geq 0}$ such that $J(\widehat{A}, \widehat{B}, \mathbf{x}_0, \{\widehat{\mathbf{u}}_t\}_{t\geq 0}) \approx J(A_\star, B_\star, \mathbf{x}_0, K_\star)$.

For any sequence of inputs $\{\widehat{\mathbf{u}}_t\}_{t\geq 0}$ such that the series defining $J(\widehat{A}, \widehat{B}, \mathbf{x}_0, \{\widehat{\mathbf{u}}_t\}_{t\geq 0})$ is absolutely convergent, we can write

$$\begin{aligned} J(\widehat{A}, \widehat{B}, \mathbf{x}_0, \widehat{K}) - J(A_\star, B_\star, \mathbf{x}_0, K_\star) = &\sum_{j=0}^{\infty}\sum_{i=0}^{\ell-1}\left[\widehat{\mathbf{x}}_{\ell j+i}^\top Q \widehat{\mathbf{x}}_{\ell j+i} - \mathbf{x}_{\ell j+i}^\top Q \mathbf{x}_{\ell j+i}\right] \\ &+ \sum_{j=0}^{\infty}\sum_{i=0}^{\ell-1}\left[\widehat{\mathbf{u}}_{\ell j+i}^\top R \widehat{\mathbf{u}}_{\ell j+i} - \mathbf{u}_{\ell j+i}^\top R \mathbf{u}_{\ell j+i}\right]. \end{aligned} \tag{4.2.13}$$

Then, the key idea is to choose a sequence of inputs $\{\widehat{\mathbf{u}}_t\}_{t\geq 0}$ such that the system $(\widehat{A}, \widehat{B})$ tracks the system $(A_\star, B_\star, K_\star)$, i.e., $\widehat{\mathbf{x}}_{\ell j} = \mathbf{x}_{\ell j}$ for any $j \geq 0$ ( $\widehat{\mathbf{x}}_0 = \mathbf{x}_0$ because both systems are initialized at the same state). This can be done because $(\widehat{A}, \widehat{B})$ is $(\ell, \tau/2)$-controllable when $(A_\star, B_\star)$ is $(\ell, \tau)$-controllable and the estimation error is sufficiently small, as shown in Lemma 4.2.12. First, we present a result that quantifies the effect of matrix perturbations on powers of matrices.

**Lemma 4.2.11.** *Let $M$ be an arbitrary matrix in $\mathbb{R}^{n\times n}$ and let $\rho \geq \rho(M)$. Then, for all $k \geq 1$ and real matrices $\Delta$ of appropriate dimensions we have*

$$\begin{aligned} \|(M+\Delta)^k\| &\leq \tau(M,\rho)(\tau(M,\rho)\|\Delta\| + \rho)^k, \\ \|(M+\Delta)^k - M^k\| &\leq k\,\tau(M,\rho)^2(\tau(M,\rho)\|\Delta\| + \rho)^{k-1}\|\Delta\|. \end{aligned}$$

*Recall that $\tau(M,\rho)$ is defined in Equation 4.2.1.*

*Proof.* This proof is a simple modification of Lemma D.1 in [41]. We replicate the argument here for completeness.

Fix an integer $k \geq 1$. Consider the expansion of $(M+\Delta)^k$ into $2^k$ terms. Label all these terms as $T_{i,j}$ for $i = 0, ..., k$ and $j = 1, ..., \binom{k}{i}$ where $i$ denotes the degree of $\Delta$ in the term (hence there are $\binom{k}{i}$ terms with a degree of $i$ for $\Delta$). Using the fact that $\|M^k\| \leq \tau(M,\rho)\rho^k$ for all $k \geq 0$, we can bound $\|T_{i,j}\| \leq \tau(M,\rho)^{i+1}\rho^{k-i}\|\Delta\|^i$. Hence by triangle inequality:

$$\begin{aligned} \|(M+\Delta)^k\| &\leq \sum_{i=0}^{k}\sum_{j}\|T_{i,j}\| \\ &\leq \sum_{i=0}^{k}\binom{k}{i}\tau(M,\rho)^{i+1}\rho^{k-i}\|\Delta\|^i \\ &= \tau(M,\rho)\sum_{i=0}^{k}\binom{k}{i}(\tau(M,\rho)\|\Delta\|)^i\rho^{k-i} \\ &= \tau(M,\rho)(\tau(M,\rho)\|\Delta\| + \rho)^k. \end{aligned}$$

To prove the first part of the lemma we follow the same argument. We find

$$\begin{aligned}\|(M+\Delta)^k - M^k\| &\leq \sum_{i=1}^{k}\sum_{j}\|T_{i,j}\| \\ &\leq \sum_{i=1}^{k}\binom{k}{i}\tau(M,\rho)^{i+1}\rho^{k-i}\|\Delta\|^i \\ &= \tau(M,\rho)\sum_{i=1}^{k}\binom{k}{i}(\tau(M,\rho)\|\Delta\|)^i\rho^{k-i} \\ &= \tau(M,\rho)\left[(\tau(M,\rho)\|\Delta\|+\rho)^k - \rho^k\right] \\ &\leq kC_M^2(\tau(M,\rho)\|\Delta\|+\rho)^{k-1}\|\Delta\|\,,\end{aligned}$$

where the last inequality follows from the mean value theorem applied to the function $z \mapsto z^k$. □

Lemma 4.2.11 quantifies the effect of a perturbation $\Delta$, applied to a matrix $M$ on the spectral radius of $M+\Delta$. We are interested in quantifying the sizes of these perturbations for all $k = 1, 2, \ldots, \ell$. Depending on $\|\Delta\|$, $M$, and $\rho$ the sum $\tau(M,\rho)\|\Delta\| + \rho$ can either be greater than one or smaller than one. For notational simplicity, in the rest of the proof we denote $\beta = \max\{1, \varepsilon\tau(A_\star,\rho) + \rho\}$. Then, we have $\|(A_\star+\Delta)^k\| \leq \tau(A_\star,\rho)\beta^{\ell-1}$ and $\|(A_\star+\Delta)^k - A_\star^k\| \leq \ell\tau(A_\star,\rho)^2\beta^{\ell-1}\varepsilon$ for all $k \leq \ell-1$ and all real matrices $\Delta$ with $\|\Delta\| \leq \varepsilon$.

We denote $C_\ell = \begin{bmatrix}B_\star & A_\star B_\star & \ldots & A_\star^{\ell-1}B_\star\end{bmatrix}$ and $\widehat{\mathcal{C}}_\ell = \begin{bmatrix}\widehat{B} & \widehat{A}\widehat{B} & \ldots & \widehat{A}^{\ell-1}\widehat{B}\end{bmatrix}$. Before presenting the next result we recall that for any block matrix $M$ with blocks $M_{i,j}$ we have $\|M\|^2 \leq \sum_{i,j}\|M_{i,j}\|^2$. The next lemma gives us control over the smallest positive singular value of the controllability matrix $\widehat{\mathcal{C}}_\ell$ in terms of the corresponding value for $\mathcal{C}_\ell$.

**Lemma 4.2.12.** *Suppose the linear $(A_\star, B_\star)$ is $(\ell,\nu)$-controllable and let $\rho$ be a real number such that $\rho \geq \rho(A_\star)$. Then, if $\|\widehat{A} - A_\star\| \leq \varepsilon$ and $\|\widehat{B} - B_\star\| \leq \varepsilon$, we have*

$$\underline{\sigma}(\widehat{C}_\ell) \geq \tau - 3\varepsilon\ell^{\frac{3}{2}}\tau(A_\star,\rho)^2\max\{1,\tau(A_\star,\rho)\|\Delta\|+\rho\}^{\ell-1}\left(\|B_\star\|+1\right).$$

*Proof.* We can write

$$\underline{\sigma}\left(\begin{bmatrix}\widehat{B} & \widehat{A}\widehat{B} & \ldots & \widehat{A}^{\ell-1}\widehat{B}\end{bmatrix}\right) = \min_{v\in\mathcal{S}^{n-1}}\left\|v^\top\begin{bmatrix}\widehat{B} & \widehat{A}\widehat{B} & \ldots & \widehat{A}^{\ell-1}\widehat{B}\end{bmatrix}\right\|.$$

Fix an arbitrary unit vector $v$ in $\mathbb{R}^p$. Then,

$$\begin{aligned}
&\left\| v^\top \begin{bmatrix} B_\star & A_\star B_\star & \dots & A_\star^{\ell-1} B_\star \end{bmatrix} - v^\top \begin{bmatrix} \widehat{B} & \widehat{A}\widehat{B} & \dots & \widehat{A}^{\ell-1}\widehat{B} \end{bmatrix} \right\| \\
&\le \left\| v^\top \begin{bmatrix} B_\star & A_\star B_\star & \dots & A_\star^{\ell-1} B_\star \end{bmatrix} - v^\top \begin{bmatrix} B_\star & \widehat{A} B_\star & \dots & \widehat{A}^{\ell-1} B_\star \end{bmatrix} \right\| \\
&\quad + \left\| v^\top \begin{bmatrix} B_\star & \widehat{A} B_\star & \dots & \widehat{A}^{\ell-1} B_\star \end{bmatrix} - v^\top \begin{bmatrix} \widehat{B} & \widehat{A}\widehat{B} & \dots & \widehat{A}^{\ell-1}\widehat{B} \end{bmatrix} \right\| \\
&\le \varepsilon \ell^{\frac{3}{2}} \tau(A_\star, \rho)^2 \beta^{\ell-1} \|B_\star\| + \varepsilon \sqrt{\ell} \tau(A_\star, \rho, \beta)^{\ell-1} \\
&\le \varepsilon \ell^{\frac{3}{2}} \tau(A_\star, \rho)^2 \beta^{\ell-1} \left( \|B_\star\| + 1 \right).
\end{aligned}$$

We used $\ell \ge 1$, $\tau(A_\star, \rho) \ge 1$, Lemma 4.2.11, and the upper bound $\|M\|^2 \le \sum_{i,j} \|M_{i,j}\|^2$ on the operator norm of a block matrix. The conclusion follows by the triangle inequality. □

Lemma 4.2.12 tells us that by the assumption made in Proposition 4.2.9 on $\varepsilon$, we have $\underline{\sigma}(\widehat{\mathcal{C}}_\ell) \ge \frac{\tau_\ell}{2}$. Hence, we know that for any $\mathbf{x}_0 \in \mathbb{R}^n$ and $\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{\ell-1} \in \mathbb{R}^p$, there exist $\widehat{\mathbf{u}}_0, \widehat{\mathbf{u}}_1, \dots, \widehat{\mathbf{u}}_{\ell-1} \in \mathbb{R}^p$ such that

$$A_\star^\ell \mathbf{x}_0 + \sum_{i=0}^{\ell-1} A_\star^i B_\star \mathbf{u}_{\ell-1-i} = \widehat{A}^\ell \mathbf{x}_0 + \sum_{i=0}^{\ell-1} \widehat{A}^i \widehat{B} \widehat{\mathbf{u}}_{\ell-1-i} \tag{4.2.14}$$

because the system $(\widehat{A}, \widehat{B})$ is controllable. This equation implies that $\widehat{\mathbf{x}}_\ell = \mathbf{x}_\ell$.

We denote the concatenation of $\mathbf{u}_i$, for $i$ from 0 to $\ell-1$ by $\mathbf{u}^{(\ell)}$. We define $\widehat{\mathbf{u}}^{(\ell)}$ analogously. Therefore, Equation (4.2.14) can be rewritten as

$$\left( A_\star^\ell - \widehat{A}^\ell \right) \mathbf{x}_0 + \left( \mathcal{C}_\ell - \widehat{\mathcal{C}}_\ell \right) \mathbf{u}^{(\ell)} = \widehat{\mathcal{C}}_\ell (\widehat{\mathbf{u}}^{(\ell)} - \mathbf{u}^{(\ell)}). \tag{4.2.15}$$

Recall that $\beta = \max\{1, \tau(A_\star, \rho)\|\Delta\| + \rho\}$. Combining Lemma 4.2.11 and the upper bound on operator norms of block matrices we find $\|\widehat{\mathcal{C}}_\ell - \mathcal{C}_\ell\| \le \varepsilon \ell^{\frac{3}{2}} \tau(A_\star, \rho)^2 \beta^{\ell-1} \left( \|B_\star\| + 1 \right)$.

We are free to choose $\widehat{\mathbf{u}}^{(\ell)}$ anyway we wish as long as Equation (4.2.15) is true. Therefore, we can choose $\widehat{\mathbf{u}}^{(\ell)}$ such that $\widehat{\mathbf{u}}^{(\ell)} - \mathbf{u}^{(\ell)}$ is perpendicular to the nullspace of $\widehat{\mathcal{C}}_\ell$. Then,

$$\begin{aligned}
\frac{\tau_\ell}{2} \|\widehat{\mathbf{u}}^{(\ell)} - \mathbf{u}^{(\ell)}\| &\le \|\widehat{\mathcal{C}}_\ell (\widehat{\mathbf{u}}^{(\ell)} - \mathbf{u}^{(\ell)})\| \le \varepsilon \ell \tau(A_\star, \rho)^2 \beta^{\ell-1} \|\mathbf{x}_0\| + \|\widehat{\mathcal{C}}_\ell - \mathcal{C}_\ell\| \|\mathbf{u}^{(\ell)}\| \\
&\le \varepsilon \ell \tau(A_\star, \rho)^2 \beta^{\ell-1} \|\mathbf{x}_0\| + \varepsilon \ell^{\frac{3}{2}} \tau(A_\star, \rho)^2 \beta^{\ell-1} \left( \|B_\star\| + 1 \right) \|\mathbf{u}^{(\ell)}\|.
\end{aligned}$$

Hence,

$$\begin{aligned}
\|\widehat{\mathbf{u}}^{(\ell)} - \mathbf{u}^{(\ell)}\| &\le \frac{2\varepsilon \ell^{\frac{3}{2}}}{\tau_\ell} \tau(A_\star, \rho)^2 \beta^{\ell-1} (\|B_\star\| + 1) \left( \|\mathbf{x}_0\| + \|\mathbf{u}^{(\ell)}\| \right) \\
&=: \eta \left( \|\mathbf{x}_0\| + \|\mathbf{u}^{(\ell)}\| \right) .
\end{aligned} \tag{4.2.16}$$

Let us consider the block Toeplitz matrix

$$\mathcal{T}_\ell = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ B_\star & 0 & 0 & \cdots & 0 \\ A_\star B_\star & B_\star & 0 & \cdots & 0 \\ \vdots & \dots & \dots & \dots & \dots \\ A_\star^{\ell-2} B_\star & A_\star^{\ell-3} B_\star & \cdots & B_\star & 0 \end{bmatrix}.$$

From Lemma 4.2.11 and the upper bound on operator norms of block matrices we have $\|\mathcal{T}_\ell - \widehat{\mathcal{T}}_\ell\| \leq \varepsilon \ell^2 \tau(A_\star, \rho)^2 \beta^{\ell-2}(\|B_\star\| + 1)$. Let $\mathbf{x}^{(\ell)}$ be the concatenation of the vectors $\mathbf{x}_0$, $\mathbf{x}_1$, $\dots$, $\mathbf{x}_{\ell-1}$. Then,

$$\mathbf{x}^{(\ell)} = \mathcal{T}_\ell \mathbf{u}^{(\ell)} + \begin{bmatrix} I_n \\ A_\star \\ \vdots \\ A^{\ell-1} \end{bmatrix} \mathbf{x}_0.$$

Hence,

$$\begin{aligned} \|\mathbf{x}^{(\ell)} - \widehat{\mathbf{x}}^{(\ell)}\| &\leq \|\mathcal{T}_\ell \mathbf{u}^{(\ell)} - \widehat{\mathcal{T}}_\ell \widehat{\mathbf{u}}^{(\ell)}\| + \varepsilon \ell^{\frac{3}{2}} \tau(A_\star, \rho)^2 \beta^{\ell-2} \|\mathbf{x}_0\| \\ &\leq \|\mathcal{T}_\ell \mathbf{u}^{(\ell)} - \widehat{\mathcal{T}}_\ell \mathbf{u}^{(\ell)}\| + \|\widehat{\mathcal{T}}_\ell \mathbf{u}^{(\ell)} - \widehat{\mathcal{T}}_\ell \widehat{\mathbf{u}}^{(\ell)}\| + \varepsilon \ell^{\frac{3}{2}} \tau(A_\star, \rho)^2 \beta^{\ell-2} \|\mathbf{x}_0\| \\ &\leq \|\mathcal{T}_\ell - \widehat{\mathcal{T}}_\ell\| \|\mathbf{u}^{(\ell)}\| + \|\widehat{\mathcal{T}}_\ell\| \|\mathbf{u}^{(\ell)} - \widehat{\mathbf{u}}^{(\ell)}\| + \varepsilon \ell^{\frac{3}{2}} \tau(A_\star, \rho)^2 \beta^{\ell-2} \|\mathbf{x}_0\| \\ &\leq \varepsilon \ell^2 \tau(A_\star, \rho)^2 \beta^{\ell-2}(\|B_\star\| + 1) \|\mathbf{u}^{(\ell)}\| + \varepsilon \ell^{\frac{3}{2}} \tau(A_\star, \rho)^2 \beta^{\ell-2} \|\mathbf{x}_0\| \\ &\quad + \ell \tau(A_\star, \rho) \beta^{\ell-2}(\|B_\star\| + 1) \|\mathbf{u}^{(\ell)} - \widehat{\mathbf{u}}^{(\ell)}\| \\ &\leq 2\varepsilon \ell^{\frac{5}{2}} \tau(A_\star, \rho)^3 \beta^{2(\ell-1)}(1 + \tau^{-1})(\|B_\star\| + 1)^2 \left[\|\mathbf{x}_0\| + \|\mathbf{u}^{(\ell)}\|\right] \\ &=: \mu \left[\|\mathbf{u}^{(\ell)}\| + \|\mathbf{x}_0\|\right]. \end{aligned} \tag{4.2.17}$$

In Equations (4.2.16) and (4.2.17) we proved that the inputs and states of the system $(\widehat{A}, \widehat{B})$ are close to the inputs and states of the system $(A_\star, B_\star)$ from time 0 to $\ell$. Since the inputs to the system $(\widehat{A}, \widehat{B})$ satisfy Equation (4.2.15), we know that $\widehat{\mathbf{x}}_{\ell j} = \mathbf{x}_{\ell j}$ for all $j$. We can repeat the same argument as above, with $\mathbf{x}_{\ell j}$ taking the place of $\mathbf{x}_0$, to show that the inputs and states of the two systems are close to each other from time $\ell j$ to $\ell(j+1)$. Let us denote by $\mathbf{x}_j^{(\ell)}$ the concatenation of the vectors $\mathbf{x}_{\ell j}$, $\mathbf{x}_{\ell j+1}$, $\dots$, $\mathbf{x}_{\ell j+\ell-1}$ and let $\mathbf{u}_j^{(\ell)}$ be defined analogously. Then,

$$\|\widehat{\mathbf{u}}_j^{(\ell)} - \mathbf{u}_j^{(\ell)}\| \leq \eta \left[\|\mathbf{u}_j^{(\ell)}\| + \|\mathbf{x}_{\ell j}\|\right], \quad \text{and} \quad \|\widehat{\mathbf{x}}_j^{(\ell)} - \mathbf{x}_j^{(\ell)}\| \leq \mu \left[\|\mathbf{u}_j^{(\ell)}\| + \|\mathbf{x}_{\ell j}\|\right]. \tag{4.2.18}$$

Now, we note that

$$\begin{aligned}\mathbf{x}_0^\top \widehat{P}\mathbf{x}_0 - \mathbf{x}_0^\top P_\star \mathbf{x}_0 &\leq \sum_{j=0}^{\infty}\sum_{i=0}^{\ell-1}\left[\widehat{\mathbf{x}}_{\ell j+i}^\top Q\widehat{\mathbf{x}}_{\ell j+i} - \mathbf{x}_{\ell j+i}^\top Q\mathbf{x}_{\ell j+i}\right] \\ &\quad + \sum_{j=0}^{\infty}\sum_{i=0}^{\ell-1}\left[\widehat{\mathbf{u}}_{\ell j+i}^\top R\widehat{\mathbf{u}}_{\ell j+i} - \mathbf{u}_{\ell j+i}^\top R\mathbf{u}_{\ell j+i}\right] \\ &\leq \sum_{j=0}^{\infty} 2\|Q\|\|\mathbf{x}_j^{(\ell)}\|\|\mathbf{x}_j^{(\ell)} - \widehat{\mathbf{x}}_j^{(\ell)}\| + \|Q\|\|\mathbf{x}_j^{(\ell)} - \widehat{\mathbf{x}}_j^{(\ell)}\|^2 \\ &\quad + \sum_{j=0}^{\infty} 2\|R\|\|\mathbf{u}_j^{(\ell)}\|\|\mathbf{u}_j^{(\ell)} - \widehat{\mathbf{u}}_j^{(\ell)}\| + \|R\|\|\mathbf{u}_j^{(\ell)} - \widehat{\mathbf{u}}_j^{(\ell)}\|^2 .\end{aligned}$$

Now, we use the upper bounds from (4.2.18). We always have $\eta \leq \mu$. Since Proposition 4.2.9 assumes $\varepsilon$ is small enough, we also have $\mu \leq 1$. Using these upper bounds, we find

$$\begin{aligned}\widehat{J} - J_\star &\leq \mu \sum_{j=0}^{\infty} 2\|Q\|\|\mathbf{x}_j^{(\ell)}\|\left[\|\mathbf{u}_j^{(\ell)}\| + \|\mathbf{x}_{\ell j}\|\right] + \|Q\|\left[\|\mathbf{u}_j^{(\ell)}\| + \|\mathbf{x}_{\ell j}\|\right]^2 \\ &\quad + \mu \sum_{j=0}^{\infty} 2\|R\|\|\mathbf{u}_j^{(\ell)}\|\left[\|\mathbf{u}_j^{(\ell)}\| + \|\mathbf{x}_{\ell j}\|\right] + \|R\|\left[\|\mathbf{u}_j^{(\ell)}\| + \|\mathbf{x}_{\ell j}\|\right]^2 .\end{aligned}$$

Then, we get $\widehat{J} - J_\star \leq 8\mu \max\{\|Q\|, \|R\|\} \sum_{j=0}^{\infty} \|\mathbf{x}_j^{(\ell)}\|^2 + \|\mathbf{u}_j^{(\ell)}\|^2$ after using the inequalities $(a+b)^2 \leq 2(a^2+b^2)$ and $2ab \leq a^2 + b^2$. Now, As long as $\|\mathbf{x}_0\| \leq 1$ we have

$$\min\{\underline{\sigma}(Q), \underline{\sigma}(R)\} \sum_{j=0}^{\infty} \|\mathbf{x}_j^{(\ell)}\|^2 + \|\mathbf{u}_j^{(\ell)}\|^2 \leq \sum_{t=0}^{\infty} \mathbf{x}_t^\top Q\mathbf{x}_t + \mathbf{u}_t^\top R\mathbf{u}_t = \mathbf{x}_0^\top P_\star \mathbf{x}_0 \leq \|P_\star\|. \quad (4.2.19)$$

Since the initial state is an arbitrary unit norm vector, our upper bound on $\mathbf{x}_0^\top(\widehat{P} - P_\star)\mathbf{x}_0$ becomes

$$\lambda_{\max}\left(\widehat{P} - P_\star\right) \leq 16\varepsilon\ell^{\frac{5}{2}}\tau(A_\star, \rho)^3\beta^{2(\ell-1)}(1+\nu^{-1})(1+\|B_\star\|)^2\|P_\star\|\frac{\max\{\|Q\|, \|R\|\}}{\min\{\underline{\sigma}(Q), \underline{\sigma}(R)\}}. \quad (4.2.20)$$

Now, we can reverse the roles of $(\widehat{A}, \widehat{B})$ and $(A_\star, B_\star)$ and repeat the same argument and obtain an upper bound on $\lambda_{\max}\left(P_\star - \widehat{P}\right)$ analogous to Equation (4.2.20), but which has $\|P_\star\|$ replaced by $\|\widehat{P}\|$ on the right hand side. However, (4.2.20) implies that $\|\widehat{P}\| \leq \|P_\star\| + 1 \leq 2\|P_\star\|$ because we assumed that $\varepsilon$ is small enough such that the right hand side of (4.2.20) is less than one, and because $P_\star \succeq I_n$. The conclusion follows.

We note that the proof can be extended to the case when the cost matrix $Q$ is also being perturbed. Moreover, the only essiential step in the argument where we used $Q \succ 0$ is (4.2.19). The goal of (4.2.19) is to upper bound $\sum_{j=0}^{\infty} \|\mathbf{x}_j^{(\ell)}\|^2 + \|\mathbf{u}_j^{(\ell)}\|^2$. This quantity can be upper bounded even when $Q$ is not positive definite, but the system $(Q^{1/2}, A)$ is observable; which is a necessary requirement for LQR on the parameters $(A, B, Q, R)$ to stabilize the system $(A, B)$.

## 4.3 Related work

For the offline LQR batch setting, Fiechter [51] proved that the sub-optimality gap $\widehat{J} - J_\star$ scales as $\mathcal{O}(\varepsilon)$ for certainty equivalent control. A crucial assumption of his analysis is that the nominal controller stabilizes the true unknown system. We give bounds on when this assumption is valid. Recently, Dean et al. [41] proposed a robust controller synthesis procedure which takes model uncertainty into account and whose suboptimality gap scales as $\mathcal{O}(\varepsilon)$. Tu and Recht [156] show that the gap $\widehat{J} - J_\star$ of certainty equivalent control scales asymptotically as $\mathcal{O}(\varepsilon^2)$; we provide a non-asymptotic analogue of this result. Fazel et al. [49] and Malik et al. [93] analyze a model-free approach to policy optimization for LQR, in which the controller is directly optimized from sampled rollouts. Malik et al. [93] showed that, after collecting $N$ rollouts, a derivative free method achieves a discounted cost gap that scales as $\mathcal{O}(1/\sqrt{N})$ or $\mathcal{O}(1/N)$, depending on the oracle model used.

In the online LQR adaptive setting it is well understood that using the certainty equivalence principle without adequate exploration can result in a lack of parameter convergence [see e.g. 14]. Abbasi-Yadkori and Szepesvári [1] showed that optimism in the face of uncertainty (OFU), when applied to online LQR, yields $\widetilde{\mathcal{O}}(\sqrt{T})$ regret.

Ibrahimi et al. [68] showed that when the underlying system is sparse, the dimension dependent constants in the regret bound can be improved. The main issue with OFU for LQR is that there are no known computationally tractable ways of implementing it. In order to deal with this, both Dean et al. [42] and Abbasi-Yadkori et al. [3] propose polynomial time algorithms for adaptive LQR based on $\varepsilon$-greedy exploration which achieve $\widetilde{\mathcal{O}}(T^{2/3})$ regret. Only recently progress has been made on offering $\widetilde{\mathcal{O}}(\sqrt{T})$ regret guarantees for computationally tractable algorithms. Abeille and Lazaric [7] show that Thompson sampling achieves $\widetilde{\mathcal{O}}(\sqrt{T})$ (frequentist) regret for the case when the state and inputs are both scalars. In a Bayesian setting Ouyang et al. [108] showed that Thompson sampling achieves $\widetilde{\mathcal{O}}(\sqrt{T})$ *expected* regret. Faradonbeh et al. [46] argue that certainty equivalence control with an epsilon-greedy-like scheme achieves $\widetilde{\mathcal{O}}(\sqrt{T})$ regret, though their work does not provide any explicit dependencies on instance parameters. Finally, Cohen et al. [37] also give an efficient algorithm based on semidefinite programming that achieves $\widetilde{\mathcal{O}}(\sqrt{T})$ regret.

The literature for LQG is less complete, with most of the focus on the estimation side. Hardt et al. [61] show that gradient descent can be used to learn a model with good predictive performance, under strong technical assumptions on the $A$ matrix. A line of work [62, 63]

has focused on using spectral filtering techniques to learn a predictive model with low regret. Beyond predictive performance, several works [110, 142, 154] show how to learn the system dynamics up to a similarity transform from input/output data. Finally, we remark that Boczar et al. [21] give sub-optimality guarantees for output-feedback of a single-input-single-output (SISO) linear system with no process noise.

A key part of our analysis involves bounding the perturbation of solutions to the discrete algebraic Riccati equation. While there is a rich line of work studying perturbations of Riccati equations [79, 80, 147, 148], the results in the literature are either asymptotic in nature or difficult to use and interpret. We clarify the operator-theoretic result of Konstantinov et al. [79] and provide an explicit upper bound on the perturbation based on their proof strategy. Also, we take a new direct approach and use an extended notion of controllability to give a constructive and simpler result. While the result of Konstantinov et al. [79] applies more generally to systems that are stabilizable, we give examples of linear systems for which our new perturbation result is tighter.

# Chapter 5

# Multi-player bandits and matching markets

We study an economic version of the multi-armed bandit (MAB) problem in which there are *multiple* agents solving a bandit problem, and there is competition—if two or more agents pick the same arm, only one of the agents is given a reward. We assume that the arms have a preference ordering over the agents—a key point of departure from the line of work on multi-player bandits with collisions [30, 34, 87, 136]—and this ordering is unknown a priori to the agents.

We are motivated by problems involving two-sided markets that link producers and consumers or workers and employers, where each side sees the other side via a recommendation system, and where there is scarcity on the supply side (for example, a restaurant has a limited number of seats, a street has a limited capacity, or a worker can attend to one task at a time). The overall goal is an economic one—we wish to find a stable matching between producers and consumers. To study the core mathematical problems that arise in such a setting, we have abstracted away the recommendation systems on the two sides, modeling them via the preference orderings and the differing reward functions. Several massive online labor and service markets can be captured by this abstraction; see the end of this section for an illustration of an application. In the context of two-sided markets the arms' preferences can be explict, e.g. when the arms represent entities in the market with their own utilities for the other side of the market, or implicit, e.g. when the arms represent resources their "preferences" encode the skill levels of the agents in securing those resources.

To determine the appropriate notions of equilibria in our multi-agent MAB model, we turn to the literature on stable matchings in two-sided markets [54, 56, 78, 120, 121]. Since its introduction by Gale and Shapley [54], the stable matching problem has had high practical impact, leading to improved matching systems for high-school admissions and labor markets [119], house allocations with existing tenants [6], content delivery networks [92], and kidney exchanges [122].

In spite of these advances, standard matching models tend to assume that entities in the market know their preferences over the other side of the market. Models that allow

unknown preferences usually assume that preferences can be discovered through one or few interactions [13], e.g., one interview per candidate in the case of medical residents market [119, 121]. These assumptions do not capture the statistical uncertainty inherent in problems where data informs preferences. We discuss related work in further detail in Section 5.3.

In contrast, our work is motivated by modern matching markets which operate at scale and require repeated interactions between the two sides of the market, leading to exploration-exploitation tradeoffs. We consider two-sided markets in which entities on one side of the market do not know their preferences over the other side, and develop matching and learning algorithms that can provably attain a stable market outcome in this setting. Our contributions are as follows:

- We introduce a new model for understanding two-sided markets in which one side of the market does not know its preferences over the other side, but is allowed multiple rounds of interaction. Our model combines work on multi-armed bandits with work on stable matchings. In particular, we define two natural notions of regret, based on stable matchings of the market, which quantify the exploration-exploitation trade-off for each individual agent.

- We extend the Explore-then-Commit (ETC) algorithm for single agent MAB to our multi-agent setting. We consider two versions of ETC: centralized and decentralized. For both versions we prove $\mathcal{O}(\log(n))$ problem-dependent upper bounds on the regret of each agent.

- In addition to the known limitations of ETC for single agent MAB, in Section 5.2.2 we discuss other issues with ETC in the multi-agent setting. To address these issues we introduce a centralized version of the well-known upper confidence bound (UCB) algorithm. We prove that centralized UCB achieves $\mathcal{O}(\log(n))$ problem-dependent upper bounds on the regret of each agent. Moreover, we show that centralized UCB is incentive compatible.

Most of the above results can be extended to the case where arms also have uncertain preferences over agents in a straightforward manner. For the sake of simplicity, we focus on the setting where one side of market initiates the exploration and leave extensions of our results to future work. The material presented in this chapter is based on the work by Liu et al. [88].

**Online labor markets** Our model is applicable to matching problems that arise in online labor markets (e.g., Upwork and Taskrabbit for freelancing, Handy for housecleaning) and online crowdsourcing platforms (e.g., Amazon Mechanical Turks). In this case, the employers, each with a stream of similar tasks to be delegated, can be modeled as the players, and the workers can be modeled as the arms. For an employer, the mean reward received from each worker when a task is completed corresponds to how well the task was completed (e.g., did the Turker label the picture correctly?). This differs for each worker due to differing

skill levels, which the employer does not know a priori and must learn by exploring different workers. A worker has preferences over different types of tasks (e.g., based on payment or prior familiarity the task) and can only work on one task at a time; hence they will pick their most preferred task to complete out of all the tasks that are offered to them.

## 5.1 Problem setting

We denote the set of $N$ agents by $\mathcal{N} = \{p_1, p_2, \ldots, p_N\}$ and the set of $K$ arms by $\mathcal{K} = \{a_1, a_2, \ldots, a_K\}$. We assume $N \leq K$. At time step $t$, each agent $p_i$ selects an arm $m_t(i)$, where $m_t \in \mathcal{K}^N$ is the vector of all agents' selections.

When multiple agents select the same arm only one agent is allowed to pull the arm, according to the arm's preferences via a mechanism we detail shortly. Then, if player $p_i$ successfully pulls arm $m_t(i)$ at time $t$, they are said to be *matched* to $m_t(i)$ at time $t$ and they receive a stochastic reward $X_{i,m_t}(t)$ sampled from a 1-sub-Gaussian distribution with mean $\mu_i(m_t(i))$.

Each arm $a_j$ has a fixed known ranking $\pi_j$ of the agents, where $\pi_j(i)$ is the rank of player $p_i$. In other words, $\pi_j$ is a permutation of $[N]$ and $\pi_j(i) < \pi_j(i')$ implies that arm $a_j$ prefers player $p_i$ to player $p_{i'}$. If two or more agents attempt to pull the same arm $a_j$, there is a *conflict* and only the top-ranked agent successfully pulls the arm to receive a reward; the other agent(s) $p_{i'}$ is said to be *unmatched* and does not receive any reward, that is, $X_{i',m_t}(t) = 0$. As a shorthand, the notation $p_i \succ_j p_{i'}$ means that arm $a_j$ prefers player $p_i$ over $p_{i'}$. When arm $a_j$ is clear from context, we simply write $p_i \succ p_{i'}$. Similarly, the notation $a_j \succ_i a_{j'}$ means that $p_i$ prefers arm $a_j$ over $a_{j'}$, i.e. $\mu_i(j) > \mu_i(j')$.

We now proceed to develop suitable notions of *regret* for the agents. Recall that when preferences are known on both sides, the goal is to attain a *stable matching*, where no pair of agent and arm prefer each other over their respective matches and hence no pair has the incentive to deviate. Given the full preference rankings of the arms and players, arm $a_j$ is called a *valid match* of player $p_i$ if there exists a stable matching according to those rankings such that $a_j$ and $p_i$ are matched.

We say $a_j$ is the *optimal match* of agent $p_i$ if it is the most preferred valid match. Similarly, we say $a_j$ is the *pessimal match* of agent $p_i$ if it is the least preferred valid match. Given complete preferences, the Gale-Shapley (GS) algorithm [54] finds a stable matching after repeated proposals from one side of the market to the other. The matching returned by the GS algorithm is always optimal for the proposing side and pessimal for the non-proposing side [78]. We denote by $\overline{m}$ and $\underline{m}$ the functions from $\mathcal{N}$ to $\mathcal{K}$ that define the optimal and pessimal matchings of the players according to the true preferences of the players and arms.

In the online matching problem where agent preferences are a priori unknown, agents aim to perform well relative to their best action in hindsight[1], as is typical in online learning.

[1]When the stable matching is unique, it is not hard to see that the best action for any agent—if all agents knew their own preferences and are maximizing reward—is to choose their unique valid match.

Thus, it is natural to define the *agent-optimal stable regret* of agent $p_i$ as

$$\overline{R}_i(n) := n\mu_i(\overline{m}(i)) - \sum_{t=1}^{n} \mathbb{E}X_{i,m_t}(t), \tag{5.1.1}$$

because when the arms' mean rewards are known the GS algorithm outputs the optimal matching $\overline{m}$. However, as we show in Section 5.2.2, there are natural algorithms which cannot achieve sublinear agent-optimal stable regret. Therefore, we also consider the *agent-pessimal stable regret*, defined as

$$\underline{R}_i(n) := n\mu_i(\underline{m}(i)) - \sum_{t=1}^{n} \mathbb{E}X_{i,m_t}(t). \tag{5.1.2}$$

When the stable matching is unique, the agent-optimal and agent-pessimal stable regrets coincide. Also, we note that when $N > K$ stable matches will not match all players with arms. Then, we denote $m(i) = \emptyset$ if player $p_i$ does not have a match in $m$ and we let $\mu_i(\emptyset)$ be the reward player $p_i$ receives when not matched. Then, the results presented below extend to this case. For simplicity, we assume $N \leq K$ throughout.

The central question of our investigation is as follows:

**How to achieve a *sequence of matchings* where all agents have *low stable regret*?**

Several interaction settings are of interest:

**Centralized:** At each time step the agents are required to send a ranking of the arms to a matching platform. Then, the platform decides the action vector $m_t$. In this work we consider two platforms. The first platform, shown in Algorithm 4, outputs a random assignment for a number of time steps and then computes the agent-optimal stable matching according to the agents' preferences. The second platform, shown in Algorithm 5, takes in the agent's preferences at each time step and outputs a stable matching between the agents and arms. Both platforms ensure that there will be no conflicts between the agents. The first platform corresponds to an explore-then-commit strategy. When the second platform is used the agents must rank arms in a way which enables exploration and exploitation. We show that ranking according to upper confidence bounds yields $\mathcal{O}(\log(n))$ agent-pessimal stable regret.

**Decentralized:** Agents observe each other's actions and the outcomes of the ensuing conflicts, but do not have a platform for coordination and communication. We can also ask what happens if, after selecting an arm, agents observe whether they lost a conflict and if they successfully pull an arm they observe their own reward, but they do not observe any other information. Both decentralized cases are interesting and we leave their study for future work.

## 5.2 Multi-agent bandits with a platform

### 5.2.1 Centralized Explore-then-Commit

In this section we give a guarantee for the explore-then-commit planner defined in Algorithm 4. At each iteration, each agent $p_i$ updates their mean reward for arm $j$ to be

$$\widehat{\mu}_{i,j}(t) = \frac{1}{T_{i,j}(t)} \sum_{s=1}^{t} \mathbb{1}\{m_s(i) = j\} X_{i,m_s}(s), \tag{5.2.1}$$

where $T_{i,j}(t) = \sum_{s=1}^{t} \mathbb{1}\{m_t(i) = j\}$ is the number of times agent $p_i$ successfully pulled arm $a_j$. At each time step, player $p_i$ ranks the arms in decreasing order according to $\widehat{\mu}_{i,j}(t)$ and sends the resulting ranking $\widehat{r}_{i,t}$ to the platform. As seen in Algorithm 4, for the first $hK$ time steps, the platform assigns players to arms cyclically, ensuring that each agent samples every arm $h$ times. We now provide a regret analysis of centralized ETC.

**Algorithm 4** Explore-then-Commit Platform.

**for** $t = 1, \dots, T$ **do**
  **if** $t \leq hK$ **then**
    $m_t(i) \leftarrow a_{t+i-1 \pmod K + 1}$, $\forall i$.
  **else if** $t = hK + 1$ **then**
    Receive rankings $\widehat{r}_{i,t}$ from all $p_i$.
    Compute agent-optimal stable matching $m_t(i)$ according to $\widehat{r}_{i,t}$ and $\pi_j$.
  **else**
    $m_t(i) \leftarrow m_{hK+1}(i)$, $\forall i$.
  **end if**
**end for**

**Algorithm 5** Gale-Shapley Platform.

**for** $t = 1, \dots, T$ **do**
  Receive rankings $\widehat{r}_{i,t}$ from all $p_i$.
  Compute agent-optimal stable matching $m_t$ according to all $\widehat{r}_{i,t}$ and $\pi_j$.
**end for**

**Theorem 5.2.1.** *Suppose all players rank arms according to the empirical mean rewards (5.2.1) and submit their rankings to the explore-then-commit platform. Let* $\overline{\Delta}_{i,j} = \mu_i(\overline{m}(i)) - \mu_i(j)$, $\overline{\Delta}_{i,\max} = \max_j \overline{\Delta}_{i,j}$, *and* $\Delta = \min_{i \in [N]} \min_{j:\, \overline{\Delta}_{i,j} > 0} \overline{\Delta}_{i,j} > 0$. *Then, the expected agent-optimal regret of player* $p_i$ *is upper bounded by*

$$\overline{R}_i(n) \leq h \sum_{j=1}^{K} \overline{\Delta}_{i,j} + (n - hK) \overline{\Delta}_{i,\max} NK \exp\left(-\frac{h\Delta^2}{4}\right).$$

*In particular, if* $h = \max\left\{1, \frac{4}{\Delta^2}\log\left(1+\frac{n\Delta^2 N}{4}\right)\right\}$, *we have*

$$\overline{R}_i(n) \leq \max\left\{1, \frac{4}{\Delta^2}\log\left(1+\frac{n\Delta^2 N}{4}\right)\right\}\sum_{j=1}^{K}\overline{\Delta}_{i,j} + \frac{4K\overline{\Delta}_{i,\max}}{\Delta^2}\log\left(1+\frac{n\Delta^2 N}{4}\right).$$

This result shows that centralized ETC achieves $\mathcal{O}(\log(n))$ agent-optimal stable regret when the number of exploration rounds is chosen appropriately. As is the case for single agent ETC, centralized ETC requires knowledge of both the horizon $n$ and the minimum gap $\Delta$ [see, e.g., 82, Chapter 6]. However, a glaring difference between the settings is that in the latter the regret of each agent scales with $1/\Delta^2$, where $\Delta$ is the minimum reward gap between the optimal match and a suboptimal arm across all agents. In other words, the regret of an agent might depend on the suboptimality gap of other agents. Example 5.2.2 shows that this dependence is real in general and not an artifact of our analysis. Moreover, while single agent ETC achieves $\mathcal{O}(\sqrt{n})$ problem-independent regret, Example 5.2.2 shows that centralized ETC does not have this desirable property. Finally, $\sum_{j=1}^{K}\overline{\Delta}_{i,j}$ could be negative for some agents. Therefore, some agents can have negative agent-optimal regret, an effect that never occurs in the single agent MAB problem.

**Example 5.2.2** (The dependence on $1/\Delta^2$ cannot be improved in general)**.** *Let* $\mathcal{N} = \{p_1, p_2\}$ *and* $\mathcal{K} = \{a_1, a_2\}$ *with true preferences:*

$$p_1 \colon a_1 \succ a_2 \qquad a_1 \colon p_1 \succ p_2$$
$$p_2 \colon a_2 \succ a_1 \qquad a_2 \colon p_1 \succ p_2.$$

*The agent-optimal stable matching is given by* $\overline{m}(1) = 1$ *and* $\overline{m}(2) = 2$. *Both* $a_1$ *and* $a_2$ *prefer* $p_1$ *over* $p_2$. *Therefore, at the end of the exploration stage* $p_1$ *is matched to their top choice arm while* $p_2$ *is matched to the remaining arm. In order for* $p_2$ *to be matched to their optimal arm,* $p_1$ *must correctly determine that they prefer* $a_1$ *over* $a_2$. *The number of exploration rounds would then have to be* $\Omega(1/\overline{\Delta}_{1,2}^2)$ *where* $\overline{\Delta}_{1,2} = \mu_1(1) - \mu_1(2)$. *Hence, when* $\overline{\Delta}_{1,2} \leq 1/\sqrt{n}$, *the regret of* $p_2$ *is* $\Omega(n\overline{\Delta}_{2,1})$. *Figure 5.1 depicts this effect empirically; we observe that a smaller gap* $\overline{\Delta}_{1,2}$ *causes* $p_2$ *to have larger regret.*

*In Figure 5.1, there are two agents and two arms. Player* $p_2$ *receives Gaussian rewards from the arms* $a_1$, $a_2$ *with means* 0 *and* 1 *respectively and variance* 1. *Player* $p_1$ *receives Gaussians rewards* $\Delta$ *and* 0 *from the arms* $a_1$ *and* $a_2$. *Both arms prefer* $p_1$ *over* $p_2$. *Figure 5.1 shows the regret of each agent as a function of* $\Delta$ *when we run centralized UCB with horizon* 400 *and average over* 100 *trials.*

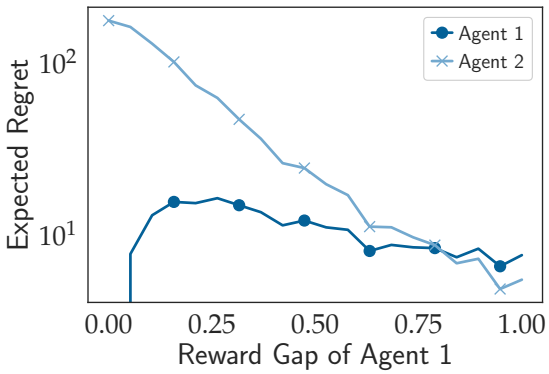


**Figure 5.1:** The empirical performance of centralized UCB in the setting described in Example 5.2.2

### 5.2.2 Centralized UCB

We saw that centralized ETC achieves $\mathcal{O}(\log(n))$ agent-optimal regret for all agents. However, centralized ETC must know the horizon $n$ and the minimum gap $\Delta$ between an optimal arm and a suboptimal arm. While knowing the horizon $n$ is feasible in certain scenarios, knowing $\Delta$ is not plausible. It is known that single agent ETC achieves $\mathcal{O}(n^{2/3})$ when the number of exploration rounds is chosen deterministically without knowing $\Delta$, and there are also known methods for adaptively choosing the number of exploration rounds so that single agent ETC achieves $\mathcal{O}(\log(n))$ [82]. However, in our setting, the $\mathcal{O}(n^{2/3})$ guarantee does not hold because the suboptimality gaps of one agent affect the regret of other agents, and the known adaptive stopping times cannot be implemented because the platform does not observe the agents' rewards. Therefore, it is necessary to find methods which do not need to know $\Delta$.

Another drawback of centralized ETC is that it requires agents to learn concurrently, i.e. players must explore randomly at the same time. Hence, even if a player knew their preferences a priori, they would still be required to explore randomly in order to guarantee low regret for all players. The Gale-Shapley Platform shown in Algorithm 5 resolves this problem, always outputting a matching that is stable—in fact, agent-optimal—according to the rankings received from the agents. We derive an upper bound on the agent-pessimal stable regret in this setting when all agents use upper confidence bounds to rank arms. In Section 5.2.3 we show this method is incentive compatible.

Before proceeding with the analysis we define more precisely the UCB method employed by each agent and also introduce several technical concepts. At each time step the platform matches agent $p_i$ with arm $m_t(i)$. Each player $p_i$ successfully pulls arm $m_t(i)$, receives reward $X_{i,m_t}(t)$, and updates their empircal mean for $m_t(i)$ as in (5.2.1). They then compute the

upper confidence bound

$$u_{i,j}(t) = \begin{cases} \infty & \text{if } T_{i,j}(t) = 0, \\ \widehat{\mu}_{i,j}(t) + \sqrt{\frac{3\log t}{2T_i(t-1)}} & \text{otherwise.} \end{cases} \tag{5.2.2}$$

Finally, each player $p_i$ orders the arms according to $u_{i,j}(t)$ and computes the ranking $\widehat{r}_{i,t+1}$ so that a higher upper confidence bound means a better rank, e.g. $\arg\max_j u_{i,j}(t)$ is ranked first in $\widehat{r}_{i,t+1}$.

Let $m$ be an injective function from the set of players $\mathcal{N}$ to the set of arms $\mathcal{K}$; hence $m$ is the matching where $m(i)$ is the match of agent $i$. Then, let $T_m(t)$ be the number of times matching $m$ is played by time $t$. We say a matching is *truly stable* if it is stable according to the true preferences induced by the mean rewards of the arms, and *non-truly stable*, otherwise. For agent $p_i$ and arm $a_\ell$ we consider the set $M_{i,\ell}$ of non-truly stable matchings $m$ such that $m(i) = \ell$. Let $\underline{\Delta}_{i,\ell} = \mu_i(\underline{m}(i)) - \mu_i(\ell)$.

Then, since any truly-stable matching yields agent-pessimal regret smaller or equal than zero for all agents, we can upper-bound the agent-pessimal regret of agent $i$ as follows:

$$\underline{R}_i(n) \leq \sum_{\ell:\, \underline{\Delta}_{i,\ell}>0} \underline{\Delta}_{i,\ell} \left( \sum_{m \in M_{i,\ell}} \mathbb{E}T_m(n) \right). \tag{5.2.3}$$

For any matching $m$ that is non-truly stable there must exist an agent $p_j$ and an arm $a_k$, different from arm $m(j)$, such that the pair $(p_j, a_k)$ is a *blocking pair* according to the true preferences $\mu$, i.e. $\mu_j(k) > \mu_j(m(j))$ and arm $a_k$ is either unmatched or $\pi_k(j) > \pi_k(m^{-1}(k))$. We say a triplet $(p_j, a_k, a_{k'})$ blocks a matching when $p_j$ is matched with $a_{k'}$ and $(p_j, a_k)$ is a blocking pair. Let $B_{j,k,k'}$ be the set of all matchings blocked by the triplet $(p_j, a_k, a_{k'})$. Given a set $S$ of matchings, we say a set $Q$ of triplets $(p_j, a_k, a_{k'})$ is a *cover* of $S$ if

$$\bigcup_{(p_j,a_k,a_{k'})\in Q} B_{j,k,k'} \supseteq S.$$

Let $\mathcal{C}(S)$ denote the set of covers of $S$. Also, let $\Delta_{j,k,k'} = \mu_j(k) - \mu_j(k')$. Now we state our result.

**Theorem 5.2.3.** *When all agents rank arms according to the upper confidence bounds (5.2.2) and submit their preferences to the Gale-Shapley Platform, the agent-pessimal regret of agent $p_i$ up to time $n$, $\underline{R}_i(n)$, is upper-bounded by*

$$\sum_{\ell:\, \underline{\Delta}_{i,\ell}>0} \underline{\Delta}_{i,\ell} \left[ \min_{Q \in \mathcal{C}(M_{i,\ell})} \sum_{\substack{(p_j,a_k,a_{k'}) \\ \in Q}} \left( 5 + \frac{6\log(n)}{\Delta^2_{j,k,k'}} \right) \right].$$

Theorem 5.2.3 offers a problem-dependent $\mathcal{O}(\log(n))$ upper bound guarantee on the agent-pessimal stable regret of each agent $p_i$. Similarly to the case of centralized ETC, the regret of one agent depends on the suboptimality gaps of other agents. However, we saw in Section 5.2.1 that centralized ETC achieves $\mathcal{O}(\log(n))$ agent-optimal stable regret, a stronger notion of regret. Example 5.2.4 shows that centralized UCB cannot yield sublinear agent-optimal stable regret in general. While centralized ETC has stronger regret guarantees, it requires knowledge of the reward gaps and of the horizon of the problem. Also, centralized ETC requires all players to have synchronized exploration rounds. UCB with the Gale-Shapley platform does not have these drawbacks.

**Example 5.2.4** (Centralized UCB does not achieve sublinear agent-optimal stable regret)**.** *Let $\mathcal{N} = \{p_1, p_2, p_3\}$ and $\mathcal{K} = \{a_1, a_2, a_3\}$, with true preferences given by:*

$$\begin{aligned} p_1 &: a_1 \succ a_2 \succ a_3 & \qquad a_1 &: p_2 \succ p_3 \succ p_1 \\ p_2 &: a_2 \succ a_1 \succ a_3 & \qquad a_2 &: p_1 \succ p_2 \succ p_3 \\ p_3 &: a_3 \succ a_1 \succ a_2 & \qquad a_3 &: p_3 \succ p_1 \succ p_2. \end{aligned}$$

*The agent-optimal stable matching is $(p_1, a_1)$, $(p_2, a_2)$, $(p_3, a_3)$. When $p_3$ incorrectly ranks $a_1 \succ a_3$ and the other two players submit their correct rankings, the Gale-Shapley Platform outputs the matching $(p_1, a_2)$, $(p_2, a_1)$, $(p_3, a_3)$. In this case $p_3$ will never correct their mistake because they never get matched with $a_1$ again, and hence their upper confidence bound for $a_1$ will never shrink. Figure 5.2 illustrates this example; the optimal regret for $p_1$ and $p_2$ is seen to be linear in $n$.*

*In Figure 5.2, the rewards of the arms for each agent are Gaussian with variance 1. The mean rewards of the arms are set so that the preference structure shown in Example 5.2.4 is satisfied. For agents $p_1$ and $p_2$, the gap in mean rewards between consecutive arms is 1. For agent $p_3$ the gap between arms $a_1$ and $a_3$ is 0.05. Figure 5.2 shows the performance of centralized UCB, averaged over 100 trials, as a function of the horizon.*

*Proof of Theorem 5.2.3.* Let $L_{j,k,k'}(n)$ be the number of times agent $p_j$ pulls arm $a_{k'}$ when the triplet $(p_j, a_k, a_{k'})$ is blocking the matching selected by the platform. Then, by definition

$$\sum_{m \in B_{j,k,k'}} T_m(n) = L_{j,k,k'}(n). \tag{5.2.4}$$

By the definition of a blocking triplet we know that if $p_j$ pulls $a_{k'}$ when $(p_j, a_k, a_{k'})$ is blocking, they must have a higher upper confidence bound for $a_{k'}$ than for $a_k$. In other words, we are trying to upper bound the expected number of times the upper confidence bound on $a_{k'}$ is higher than that of the better arm $a_k$ when we have the guarantee that each time this event occurs $a_{k'}$ is successfully pulled. Therefore, standard analysis for the single agent UCB [e.g., 29, Chap. 2] shows that

$$\mathbb{E}L_{j,k,k'}(n) \leq 5 + \frac{6\log(n)}{\Delta^2_{j,k,k'}}. \tag{5.2.5}$$

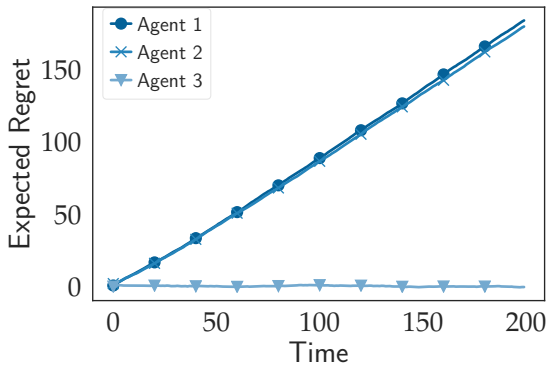


**Figure 5.2:** The empirical performance of centralized UCB in the setting described in Example 5.2.4

The conclusion follows from equations (5.2.3) and (5.2.4). □

To better understand the guarantee of Theorem 5.2.3 we consider two examples in which the markets have a special structure which enables us to simplify the upper bound on the regret. Moreover, in Corollary 5.2.7 we consider the worst case upper bound over possible coverings of matchings.

**Example 5.2.5** (Global preferences)**.** *Let* $\mathcal{N} = \{p_1, \cdots, p_N\}$ *and* $\mathcal{K} = \{a_1, \cdots, a_K\}$*. We assume the following preferences:* $p_i : a_1 \succ \cdots \succ a_K$ *and* $a_j : p_1 \succ \cdots \succ p_N$*. In other words all agents have the same ranking over arms, and all arms have the same ranking over agents. Hence, the unique stable matching is* $(p_1, a_1)$, $(p_2, a_2)$, $\ldots$, $(p_N, a_N)$*. Moreover, for any* $p_i$ *and* $a_\ell$ *we can cover the set of matchings* $M_{i,\ell}$ *with the triplets* $(p_i, a_k, a_\ell)$ *for all* $k$ *with* $1 \leq k \leq i$*. Then, Theorem 5.2.3 implies (5.2.6) once we observe that* $\Delta_{i,k,\ell} \geq \underline{\Delta}_{i,\ell}$ *for all* $k \leq i$*.*

$$\underline{R}_i(n) \leq 5i \sum_{\ell=i+1}^{K} \underline{\Delta}_{i,\ell} + \sum_{\ell=i+1}^{K} \frac{6i \log(n)}{\underline{\Delta}_{i,\ell}}. \tag{5.2.6}$$

*Figure 5.3 illustrates this example empirically, displaying the pessimal stable regret of 5 out of 20 agents. In this experiment, there are* 20 *agents and* 20 *arms. The rewards of the arms are Guassian with variance* 1*. The mean reward gap between consecutive arms is* 0.1*. Figure 5.3 shows the performance of centralized UCB, averaged over* 50 *trials, as a function of the horizon.*

*As one can see, the 1st-ranked agent has sublinear regret, consistent with (5.2.6), while the 20th-ranked agent has negative regret and our upper bound is indeed 0.*

**Example 5.2.6** (Unique pairs)**.** *Let* $\mathcal{N} = \{p_1, \cdots, p_N\}$ *and* $\mathcal{K} = \{a_1, \cdots, a_N\}$ *and assume that agent* $p_i$ *prefers arm* $a_i$ *the most and that arm* $a_i$ *prefers agent* $p_i$ *the most. Therefore, the*

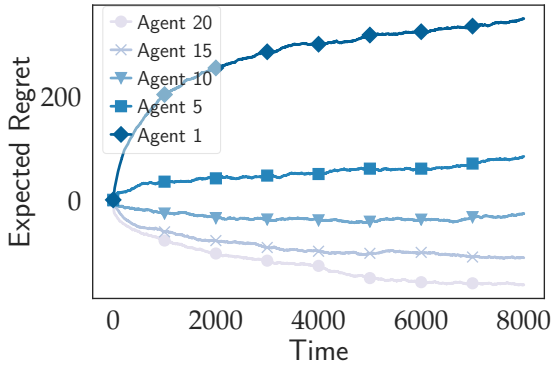


**Figure 5.3:** The empirical performance of centralized UCB in the setting described in Example 5.2.5

*unique stable matching is* $(p_1, a_1)$, $(p_2, a_2)$, $\ldots$, $(p_N, a_N)$. *Then, we can cover each set* $M_{i,\ell}$ *with the triplet* $(p_i, a_i, a_\ell)$. *Therefore, Theorem 5.2.3 implies* (5.2.7)*; note that the right-hand side is identical to the guarantee for single agent UCB:*

$$\underline{R}_i(n) \leq 5 \sum_{\ell \neq i}^{K} \underline{\Delta}_{i,\ell} + \sum_{\ell \neq i}^{K} \frac{6 \log(n)}{\underline{\Delta}_{i,\ell}}. \tag{5.2.7}$$

**Corollary 5.2.7.** *Let* $\Delta = \min_i \min_{j,j'} |\mu_i(j) - \mu_i(j')|$. *When all players follow the centralized UCB method, the regret of* $p_i$ *can be upper bounded as follows*

$$\underline{R}_i(n) \leq \max_{\ell} \Delta_{i,\ell} \left( 6NK^2 + 12 \frac{NK \log(n)}{\Delta^2} \right).$$

*Proof.* We consider the covering $(j, k, k')$ composed of all possible triples with $\mu_j(k) > \mu_j(k')$. Then, Theorem 5.2.3 implies the result because $\sum_{k' \colon \mu_j(k') < \mu_j(k)} \frac{1}{\Delta^2_{j,k,k'}} \leq \sum_{\ell=1}^{K} \frac{1}{\ell^2 \Delta^2} \leq \frac{2}{\Delta^2}$. □

### 5.2.3 Honesty and strategic behavior

Classical results show that in the agent-proposing GS algorithm, no single agent can improve their match by misrepresenting their preferences, assuming that the other agents and arms submit their true preferences [43, 118]. The result generalizes to coalition of agents. Moreover, when there is a unique stable matching, the Dubins-Freedman Theorem says that no arms or agents can benefit from misrepresenting their preferences [43].

The ETC Platform does not allows agents to choose which arms to explore. In this case, the classical results on honesty in agent-proposing GS apply; the agents are incentivized to submit the rankings according to their current mean estimates. When agents have some degree of freedom to explore over multiple rounds, it is no longer clear if any agents, or

arms, can benefit from misrepresenting their preferences in some of the rounds. In general, one agent's preferences can influence not only the matches of other agents, but also their reward estimates. One might be able to improve their regret by capitalizing on the ranking mistakes of other agents. The possibilities for long-term strategic behavior are more diverse than in the single-round setting.

In general, the optimal regret for a player can be negative if the player is on average getting rewards higher than its optimal stable arm, as seen in Figure 5.3.

We now show that when all agents except one submit their UCB-based preferences to the GS Platform, the remaining agent has an incentive to also submit preferences based on their UCBs, so long as they do not have multiple stable arms. This result is a lower bound on the optimal regret of the remaining agent, hence establishing that they have limited gains from deviating from their UCB-based preferences.

First, we establish the following lemma, which is an upper bound on the expected number of times the remaining agent can pull an arm that is better than their optimal match, regardless of what preferences they might have submitted to the platform.

**Lemma 5.2.8.** *Let $T_l^i(n)$ be the number of times an agent $i$ pulls an arm $l$ such that the mean reward of $l$ for $i$ is greater than $i$'s optimal match. Then*

$$\mathbb{E}[T_l^i(n)] \leq \min_{Q \in \mathcal{C}(M_{i,\ell})} \sum_{(j,k,k') \in Q} \left(5 + \frac{6\log(n)}{\Delta^2_{j,k,k'}}\right). \tag{5.2.8}$$

*Proof.* If agent $i$ is matched with arm $l$ in any round, the matching $m$ must be unstable according true preferences. We claim that there must exist a blocking triplet $(j, k, k')$ where $j \neq i$.

Arguing by contradiction, we suppose otherwise, that all blocking triplets in $m$ only involve agent $i$. By Theorem 4.2 in Abeledo and Rothblum [8], we can go from the matching $m$ to a $\mu$-stable matching, by iteratively *satisfying* block pairs in a 'gender consistent' order $O$. To satisfy a blocking pair $(k, j)$, we break their current matches, if any, and match $(k, j)$ to get a new matching. Doing so, agent $i$ can never get a worse match than $l$ or become unmatched as the algorithm proceeds, so the matching remains unstable—a contradiction. Hence there must exist a $j \neq i$ such that $j$ is part of a blocking triplet in $m$. In particular, agent $j$ must be submitting its UCB preferences.

The result follows from Equation (5.2.5) and the identity

$$\mathbb{E}[T_l^i(n)] = \sum_{m \in M_{i,\ell}} \mathbb{E}T_m(n).$$

□

Lemma 5.2.8 directly implies the following lower bound on the remaining agent's optimal regret.

**Proposition 5.2.9.** *Suppose all agents other than $p_i$ submit preferences according to the UCBs (5.2.2) to the GS Platform. Then the following lower bound on agent $i$'s optimal regret holds:*

$$\overline{R}_i(n) \geq \sum_{\ell:\, \overline{\Delta}_{i,l}<0} \overline{\Delta}_{i,l} \left[ \min_{Q \in \mathcal{C}(M_{i,\ell})} \sum_{(j,k,k') \in Q} \left( 5 + \frac{6\log(n)}{\Delta^2_{j,k,k'}} \right) \right].$$

Therefore, there is no sequence of preferences that an agent can submit to the GS Platform that would give them negative optimal regret greater than $\mathcal{O}(\log n)$ in magnitude. When there is a unique stable matching, Proposition 5.2.9 shows that no agent can gain significantly above and beyond the mean reward of their optimal stable arm by submitting preferences other than their UCB rankings. When there exist multiple stable matchings, however, Proposition 5.2.9 leaves open the question of whether any agent can submit a sequence of preferences that achieves super-logarithmic negative *pessimal* regret for themselves, when all other agents are playing their UCB preferences. In other words, can an agent do significantly better than its pessimal stable arm, by possibly deviating from their UCB rankings?

## 5.3 Related work

Since its introduction by Thompson [152], the stochastic multi-armed bandit problem has inspired a rich body of work spanning different settings, algorithms, and guarantees [29, 81, 82].

There has been recent interest in the MAB literature in problems with multiple, interacting players [34, 136]. In one popular formulation known as *bandits with collision*, multiple players choose from the same set of arms, and if two or more players choose the same arm, no reward is received by any player [e.g. 10, 17, 30, 87, 91, 116]. This differs from our formulation, in which arms have preferences and the most preferred player receives a reward, while the other players selecting the arm do not.

A variant of this problem is where agents have different preferences over arms. Then, Bistritz and Leshem [20]'s algorithm approximately finds the maximum matching of players to arms with $\mathcal{O}(\log(n)^{2+\kappa})$ regret. However, stable matching does not reduce to maximum matching in general, so such guarantees do not apply to matching with two-sided preferences.

The two-sided matching problem has also been studied in sequential settings. Das and Kamenica [40] performed an empirical study of a two-sided matching problem with uncertain preferences. Johari et al. [73] studied a sequential matching problem, where participants are one of several types and the goal is to learn the type of agents on one side of the market.

Ashlagi et al. [13] considered the communication and preference learning cost of stable matching. Their model formulates preference learning as querying a noiseless choice function, rather than obtaining noisy observations of one's underlying utility. Different players can query their choice functions independently; hence congestion in the preference learning stage

is not captured by this model. In many markets, obtaining information about the other side of the market itself can lead to congestion and thus the need for strategic decision. For example, Roth and Sotomayor [121, chap. 10] noted that graduating medical students go to interviews to ascertain their own preferences for hospitals, but the interviews that a student can schedule are limited. Our model begins to capture such tradeoffs by introducing statistical uncertainty in the preferences of one side of the market and providing a natural mode of interaction between the learning agents.

# Chapter 6

# Conclusion and future work

We described the challenges of empirical evaluation of RL algorithms in Chapter 2 and throughout the rest of the thesis we aimed to build a theoretical understanding of methods and problems that can elucidate the data requirements of RL, complementing empirical evaluation.

Finding the optimal exploration-exploitation tradeoff is often the main challenge in designing efficient RL algorithms. Interestingly, in the case of linear dynamical systems and LQR exploration is easy: adding small exploratory noise to the inputs is sufficient for both estimation and controller synthesis. However, LQR requires careful exploitation, i.e., a method that uses available data effectively to build controllers. We proposed two such methods, robust LQR and certainty equivalence, and we saw that certainty equivalence (the straightforward approach) surpasses robust LQR when the estimation error is small. In fact, it is known that certainty equivalence is optimal in the case of online control [140]. Nonetheless, the robust method is beneficial in a higher estimation error regime, enabling the stabilization of unknown linear systems when certainty equivalence fails.

Exploration-exploitation tradeoffs are still a concern when departing from linear dynamical systems. For example, in this thesis we considered two such departures: the identification of nonlinear dynamical systems and the assignment of players to arms in two-sided markets. While our results take us closer to understanding the fundamental limits of data-driven control, there are many limitations to the models studied here and the proposed solutions. We end with a list of open questions.

**Open questions regarding system Identification:**

- To solve trajectory planning problems we assumed access to a computational oracle. Is it possible to develop a method that has good statistical guarantees and is also computationally tractable? In practice, successful nonlinear control is often based on linearizations of the dynamics. Is it possible to quantify the sample complexity of system identification when trajectory planning is implemented using linearizations?

- Our method relies on full state observations. However, in many applications full state observations are impossible. Is it possible to obtain finite-time statistical guarantees for nonlinear system identification from partial observations?
- Our guarantee holds only when the true system being identified lies in the model class (3.3.1). When the true system is not part of the model class, how much data is needed to find the best model in class? Ross and Bagnell [117] studied this problem under a generative model.
- Only fully actuated systems can satisfy Assumption 3 with $\gamma < 1$. Is it possible to extend our result to systems that require multiple time steps to recover from disturbances?
- Assumption 4 allows only systems whose feature vectors can align with any direction. What if the feature vectors can align only with vectors in a subspace? In this case, it is not possible to recover $A_\star$ fully. However, in this case, it would not be necessary to know $A_\star$ fully in order to predict or control. Is it possible to estimate $A_\star$ only in the relevant directions?
- What if we consider infinite-dimensional feature maps $\phi$? For example, can we develop a statistical theory of learning reproducing kernel Hilbert space models of dynamical systems?

**Open questions regarding multi-player bandits:**

- We assumed access to a central matching platform. Is it possible to design robust decentralized methods that allow players to enter and exit the market asynchronously?
- In practice, methods should take advantage of context in order to achieve effective matching. Is it possible to extend our model and method to a contextual setting?
- We considered a simple reward structure. However, markets have other reward or cost components that we did not model. Is there a good way to integrate prices in our model?

# Bibliography


[1] Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret Bounds for the Adaptive Control of Linear Quadratic Systems. In *Conference on Learning Theory*, 2011.

[2] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[3] Yasin Abbasi-Yadkori, Nevena Lazić, and Csaba Szepesvári. Model-Free Linear Quadratic Control via Reduction to Expert Prediction. *arXiv:1804.06021*, 2018.

[4] Pieter Abbeel and Andrew Y Ng. Exploration and apprenticeship learning in reinforcement learning. *International Conference on Machine Learning*, pages 1–8, 2005.

[5] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y Ng. An application of reinforcement learning to aerobatic helicopter flight. In *Advances in neural information processing systems*, pages 1–8, 2007.

[6] Atila Abdulkadiroğlu and Tayfun Sönmez. House allocation with existing tenants. *Journal of Economic Theory*, 88(2):233–260, 1999.

[7] Marc Abeille and Alessandro Lazaric. Improved Regret Bounds for Thompson Sampling in Linear Quadratic Control Problems. In *International Conference on Machine Learning*, 2018.

[8] Hernan Abeledo and Uriel G. Rothblum. Paths to marriage stability. *Discrete Applied Mathematics*, 63:1–12, 10 1995.

[9] Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. pages 28–40, 2010.

[10] A. Anandkumar, N. Michael, A. K. Tang, and A. Swami. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *IEEE J.Sel. A. Commun.*, 29(4):731–745, April 2011.

[11] Brian D. O. Anderson and John B. Moore. *Optimal Control: Linear Quadratic Methods*. 2007.

[12] James Anderson and Nikolai Matni. Structured State Space Realizations for SLS Distributed Controllers. In *Allerton*, 2017.

[13] Itai Ashlagi, Mark Braverman, Yash Kanoria, and Peng Shi. Communication requirements and informative signaling in matching markets. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, EC '17, pages 263–263, New York, NY, USA, 2017. ACM.

[14] Karl J. Åström and Björn Wittenmark. On Self Tuning Regulators. *Automatica*, 9: 185–199, 1973.

[15] Karl J. Åström and Björn Wittenmark. *Adaptive Control*. 2013.

[16] Karl Johan Åström and Peter Eykhoff. System identification—a survey. *Automatica*, 7(2):123–162, 1971.

[17] Orly Avner and Shie Mannor. Concurrent bandits and cognitive radio networks. In Toon Calders, Floriana Esposito, Eyke Hüllermeier, and Rosa Meo, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 66–81, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[18] Francis Bach and Vianney Perchet. Highly-smooth zero-th order online optimization. *Conference on Learning Theory*, 2016.

[19] Sohail Bahmani and Justin Romberg. Convex programming for estimation in nonlinear recurrent models. *arXiv preprint:1908.09915*, 2019.

[20] Ilai Bistritz and Amir Leshem. Distributed multi-player bandits - a game of thrones approach. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7222–7232. Curran Associates, Inc., 2018.

[21] Ross Boczar, Nikolai Matni, and Benjamin Recht. Finite-data performance guarantees for the output-feedback control of an unknown system. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2994–2999. IEEE, 2018.

[22] Xavier Bombois, Michel Gevers, Roland Hildebrand, and Gabriel Solari. Optimal experiment design for open and closed-loop system identification. *Communications in Information and Systems*, 11(3):197–224, 2011.

[23] Francesco Borrelli, Alberto Bemporad, Michael Fodor, and Davor Hrovat. An MPC/hybrid system approach to traction control. *IEEE Transactions on Control Systems Technology*, 14(3):541–552, 2006.

[24] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[25] R. P. Braatz, Peter M. Young, John C. Doyle, and Manfred Morari. Computational Complexity of $\mu$ Calculation. *IEEE Transactions on Automatic Control*, 39(5), 1994.

[26] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI gym, 2016.

[27] Steven L Brunton, Bingni W Brunton, Joshua L Proctor, and J Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PloS one*, 11(2):e0150171, 2016.

[28] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[29] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[30] Sébastien Bubeck, Yuanzhi Li, Yuval Peres, and Mark Sellke. Non-Stochastic Multi-Player Multi-Armed Bandits: Optimal Rate With Collision Information, Sublinear Without. *arXiv e-prints*, art. arXiv:1904.12233, Apr 2019.

[31] James R. Bunch, Christopher P. Nielsen, and Danny C. Sorensen. Rank-one modification of the symmetric eigenproblem. *Numerische Mathematik*, 31(1):31–48, 1978.

[32] Eduardo F. Camacho, Daniel R. Ramírez, Daniel Limón, David Muñoz de la Peña, and Teodoro Alamo. Model predictive control techniques for hybrid systems. *Annual Reviews in Control*, 34(1):21–31, 2010.

[33] Marco C. Campi and Erik Weyer. Finite sample properties of system identification methods. *IEEE Transactions on Automatic Control*, 47(8):1329–1334, 2002.

[34] Nicolò Cesa-Bianchi, Claudio Gentile, Yishay Mansour, and Alberto Minora. Delay and cooperation in nonstochastic bandits. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 605–622, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

[35] Alessandro Chiuso and Gianluigi Pillonetto. System identification: a machine learning perspective. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:281–304, 2019.

[36] Alon Cohen, Avinatan Hassidim, Tomer Koren, Nevena Lazic, Yishay Mansour, and Kunal Talwar. Online Linear Quadratic Control. In *International Conference on Machine Learning*, 2018.

[37] Alon Cohen, Tomer Koren, and Yishay Mansour. Learning Linear-Quadratic Regulators Efficiently with only $\sqrt{T}$ Regret. *arXiv:1902.06223*, 2019.

[38] Munther A. Dahleh, Theodore V. Theodosopoulos, and John N. Tsitsiklis. The sample complexity of worst-case identification of FIR linear systems. *Systems & Control Letters*, 20(3):157–166, 1993.

[39] Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. *Conference on Learning Theory*, pages 355–366, 2008.

[40] Sanmay Das and Emir Kamenica. Two-sided bandits and the dating market. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pages 947–952, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.

[41] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *arxiv:1710.01688*, 2017.

[42] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. Regret Bounds for Robust Adaptive Control of the Linear Quadratic Regulator. In *Neural Information Processing Systems*, 2018.

[43] L. E. Dubins and D. A. Freedman. Machiavelli and the Gale-Shapley Algorithm. *The American Mathematical Monthly*, 88(7):485–494, 1981.

[44] Bogdan Dumitrescu. *Positive trigonometric polynomials and signal processing applications*. 2007.

[45] Bradley Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1), 1979.

[46] Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. Regret Analysis for Adaptive Linear-Quadratic Policies. *arXiv:1711.07230*, 2018.

[47] Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. Finite time identification in unstable linear systems. *Automatica*, 96:342–353, 2018.

[48] Salar Fattahi, Nikolai Matni, and Somayeh Sojoudi. Learning sparse dynamical systems from a single sample trajectory. *IEEE Conference on Decision and Control*, pages 2682–2689, 2019.

[49] Maryam Fazel, Rong Ge, Sham M. Kakade, and Mehran Mesbahi. Global Convergence of Policy Gradient Methods for the Linear Quadratic Regulator. In *International Conference on Machine Learning*, 2018.

[50] Eric Feron. Analysis of Robust $\mathcal{H}_2$ Performance Using Multiplier Theory. *SIAM Journal on Control and Optimization*, 35(1), 1997.

[51] Claude-Nicolas Fiechter. PAC Adaptive Control of Linear Systems. In *Conference on Learning Theory*, 1997.

[52] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *Proceedings of the ACM-SIAM symposium on Discrete algorithms*, pages 385–394, 2005.

[53] Dylan J Foster, Alexander Rakhlin, and Tuhin Sarkar. Learning nonlinear dynamical systems from a single trajectory. *Learning for Dynamics and Control*, 2020.

[54] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[55] Tobias Geyer, Georgios Papafotiou, and Manfred Morari. Hybrid model predictive control of the step-down DC–DC converter. *IEEE Transactions on Control Systems Technology*, 16(6):1112–1124, 2008.

[56] Dan Gusfield and Robert W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, Cambridge, MA, USA, 1989.

[57] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *Proceedings of the International Conference on Machine Learning*, 2017.

[58] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arxiv:1801.01290*, 2018.

[59] Peter Hall. *The Bootstrap and Edgeworth Expansion*. Springer Science & Business Media, 2013.

[60] Weiqiao Han and Russ Tedrake. Feedback design for multi-contact push recovery via LMI approximation of the piecewise-affine quadratic regulator. *IEEE-RAS International Conference on Humanoid Robotics*, pages 842–849, 2017.

[61] Moritz Hardt, Tengyu Ma, and Benjamin Recht. Gradient descent learns linear dynamical systems. *Journal of Machine Learning Research*, 19(1):1025–1068, 2018.

[62] Elad Hazan, Karan Singh, and Cyril Zhang. Learning linear dynamical systems via spectral filtering. *Advances in Neural Information Processing Systems*, 2017.

[63] Elad Hazan, Holden Lee, Karan Singh, Cyril Zhang, and Yi Zhang. Spectral filtering for general linear dynamical systems. *Advances in Neural Information Processing Systems*, pages 4634–4643, 2018.

[64] Maurice Heemels, Bart De Schutter, and Alberto Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, 2001.

[65] Nicolas Heess, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, Ali Eslami, Martin Riedmiller, et al. Emergence of locomotion behaviours in rich environments. *arxiv:1707.02286*, 2017.

[66] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. *arXiv:1709.06560*, 2017.

[67] Xia Hong, Richard J. Mitchell, Sheng Chen, Chris J. Harris, Kang Li, and George W. Irwin. Model selection approaches for non-linear system identification: a review. *International Journal of Systems Science*, 39(10):925–946, 2008.

[68] Morteza Ibrahimi, Adel Javanmard, and Benjamin Van Roy. Efficient Reinforcement Learning for High Dimensional Linear Quadratic Systems. In *Neural Information Processing Systems*, 2012.

[69] Riashat Islam, Peter Henderson, Maziar Gomrokchi, and Doina Precup. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. *arxiv:1708.04133*, 2017.

[70] Garud N. Iyengar. Robust Dynamic Programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

[71] Kevin G Jamieson, Robert Nowak, and Ben Recht. Query complexity of derivative-free optimization. pages 2672–2680, 2012.

[72] Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. *International Conference on Machine Learning*, 2020.

[73] Ramesh Johari, Vijay Kamble, and Yash Kanoria. Matching while learning. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, EC '17, pages 119–119, New York, NY, USA, 2017. ACM.

[74] Anatoli Juditsky, Håkan Hjalmarsson, Albert Benveniste, Bernard Delyon, Lennart Ljung, Jonas Sjöberg, and Qinghua Zhang. Nonlinear black-box models in system identification: mathematical foundations. *Automatica*, 31(12):1725–1750, 1995.

[75] Lydia E. Kavraki, Petr Švestka, Jean-Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[76] Mohammad Khosravi and Roy S. Smith. Convex nonparametric formulation for identification of gradient flows. *arXiv preprint:2003.12336*, 2020.

[77] Mohammad Khosravi and Roy S. Smith. Nonlinear system identification with prior knowledge of the region of attraction. *arXiv preprint:2003.12330*, 2020.

[78] Donald E. Knuth. *Stable Marriage and Its Relation to Other Combinatorial Problems*, volume 10 of *CRM Proceedings and Lecture Notes*. American Mathematical Society, 1997.

[79] Mihail M. Konstantinov, Petko Hr. Petkov, and Nicolai D. Christov. Perturbation analysis of the discrete riccati equation. *Kybernetika*, 29(1):18–29, 1993.

[80] Mihail M. Konstantinov, Da-Wei Gu, Volker Mehrmann, and Petko Hr. Petkov. *Perturbation theory for matrix equations*, volume 9. Gulf Professional Publishing, 2003.

[81] T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.*, 6(1):4–22, March 1985.

[82] Tor Lattimore and Csaba Szepesvari. *Bandit Algorithms*. Cambridge University Press (To Appear), 2019.

[83] Steven M LaValle and James J. Kuffner Jr. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001.

[84] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-End Training of Deep Visuomotor Policies. *Journal of Machine Learning Research*, 17, 2016.

[85] Weiwei Li and Emanuel Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *ICINCO (1)*, pages 222–229, 2004.

[86] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *International Conference on Learning Representations*, 2016.

[87] Keqin Liu and Qing Zhao. Distributed learning in multi-armed bandit with multiple players. *Trans. Sig. Proc.*, 58(11):5667–5681, November 2010.

[88] Lydia T Liu, Horia Mania, and Michael Jordan. Competing bandits in matching markets. In *International Conference on Artificial Intelligence and Statistics*, pages 1618–1628, 2020.

[89] Lennart Ljung. System identification: theory for the user. *Prentice Hall*, 1987.

[90] Lennart Ljung, Tianshi Chen, and Biqiang Mu. A shift in paradigm for system identification. *International Journal of Control*, 93(2):173–180, 2020.

[91] Gábor Lugosi and Abbas Mehrabian. Multiplayer bandits without observing collision information. *CoRR*, abs/1808.08416, 2018.

[92] Bruce M Maggs and Ramesh K Sitaraman. Algorithmic nuggets in content delivery. *ACM SIGCOMM Computer Communication Review*, 45(3):52–66, 2015.

[93] Dhruv Malik, Ashwin Pananjady, Kush Bhatia, Koulik Khamaru, Peter L Bartlett, and Martin J Wainwright. Derivative-Free Methods for Policy Optimization: Guarantees for Linear Quadratic Systems. *arXiv:1812.08305*, 2018.

[94] Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search of static linear policies is competitive for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1800–1809, 2018.

[95] Horia Mania, Stephen Tu, and Benjamin Recht. Certainty equivalence is efficient for linear quadratic control. In *Advances in Neural Information Processing Systems*, pages 10154–10164, 2019.

[96] Horia Mania, Michael I Jordan, and Benjamin Recht. Active learning for nonlinear system identification with guarantees. *arXiv preprint arXiv:2006.10277*, 2020.

[97] Tobia Marcucci, Robin Deits, Marco Gabiccini, Antonio Bicchi, and Russ Tedrake. Approximate hybrid model predictive control for multi-contact push recovery in complex environments. *IEEE-RAS International Conference on Humanoid Robotics*, pages 31–38, 2017.

[98] Nikolai Matni, Yuh-Shyang Wang, and James Anderson. Scalable system level synthesis for virtually localizable systems. In *IEEE Conference on Decision and Control*, 2017.

[99] J Matyas. Random optimization. *Automation and Remote control*, 26(2):246–253, 1965.

[100] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518 (7540):529, 2015.

[101] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *Proceedings of the International Conference on Machine Learning*, pages 1928–1937, 2016.

[102] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, William Paul, Michael I Jordan, and Ion Stoica. Ray: A distributed framework for emerging ai applications. *arxiv:1712.05889*, 2017.

[103] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. *arxiv:1708.02596*, 2017.

[104] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust Stochastic Approximation Approach to Stochastic Programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[105] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

[106] Andrew Y. Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger, and Eric Liang. Autonomous inverted helicopter flight via reinforcement learning. *Experimental Robotics IX, Springer*, pages 363–372, 2006.

[107] Arnab Nilim and Laurent El Ghaoui. Robust Control of Markov Decision Processes with Uncertain Transition Matrices. *Operations Research*, 53(5):780–798, 2005.

[108] Yi Ouyang, Mukul Gagrani, and Rahul Jain. Control of Unknown Linear Systems with Thompson Sampling. In *Allerton*, 2017.

[109] Samet Oymak. Stochastic gradient descent learns state equations with nonlinear activations. *Conference on Learning Theory*, 2019.

[110] Samet Oymak and Necmiye Ozay. Non-asymptotic Identification of LTI Systems from a Single Trajectory. *arXiv:1806.05722*, 2018.

[111] Samet Oymak and Necmiye Ozay. Non-asymptotic identification of LTI systems from a single trajectory. *American Control Conference*, pages 5655–5661, 2019.

[112] Fernando Paganini. Necessary and Sufficient Conditions for Robust $\mathcal{H}_2$ Performance. In *IEEE Conference on Decision and Control*, 1995.

[113] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. *arxiv:1706.01905*, 2017.

[114] Friedrich Pukelsheim. *Optimal Design of Experiments*. Wiley & Sons, 1993.

[115] Aravind Rajeswaran, Kendall Lowrey, Emanuel Todorov, and Sham Kakade. Towards generalization and simplicity in continuous control. *Advances in Neural Information Processing Systems*, 2017.

[116] Jonathan Rosenski, Ohad Shamir, and Liran Szlak. Multi-player bandits – a musical chairs approach. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 155–163, New York, New York, USA, 20–22 Jun 2016. PMLR.

[117] Stephane Ross and J. Andrew Bagnell. Agnostic system identification for model-based reinforcement learning. *International Conference on Machine Learning*, 2012.

[118] Alvin E. Roth. The economics of matching: Stability and incentives. *Mathematics of Operations Research*, 7(4):617–628, 1982.

[119] Alvin E Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of political Economy*, 92(6):991–1016, 1984.

[120] Alvin E. Roth. Deferred acceptance algorithms: history, theory, practice, and open questions. *International Journal of Game Theory*, 36(3):537–569, Mar 2008.

[121] Alvin E. Roth and Marilda A. Oliveira Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Econometric Society Monographs. Cambridge University Press, 1990. doi: 10.1017/CCOL052139015X.

[122] Alvin E Roth, Tayfun Sönmez, and M Utku Ünver. Pairwise kidney exchange. *Journal of Economic theory*, 125(2):151–188, 2005.

[123] Paat Rusmevichientong and John N Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

[124] Sadra Sadraddini and Russ Tedrake. Sampling-based polytopic trees for approximate optimal control of piecewise affine systems. *International Conference on Robotics and Automation*, pages 7690–7696, 2019.

[125] Tim Salimans, Jonathan Ho, Xi Chen, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arxiv:1703.03864*, 2017.

[126] Tuhin Sarkar and Alexander Rakhlin. Near optimal finite time identification of arbitrary linear dynamical systems. *International Conference on Machine Learning*, 2019.

[127] Tuhin Sarkar, Alexander Rakhlin, and Munther A. Dahleh. Finite-time system identification for partially observed LTI systems of unknown order. *arXiv preprint:1902.01848*, 2019.

[128] Tuhin Sarkar, Alexander Rakhlin, and Munther A. Dahleh. Nonparametric system identification of stochastic switched linear systems. *IEEE Conference on Decision and Control*, 2019.

[129] Shankar Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer, 1999.

[130] Yahya Sattar and Samet Oymak. Non-asymptotic and accurate learning of nonlinear dynamical systems. *arXiv preprint:2002.08538*, 2020.

[131] Johan Schoukens and Lennart Ljung. Nonlinear system identification: a user-oriented road map. *IEEE Control Systems Magazine*, 39(6):28–99, 2019.

[132] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. pages 1889–1897, 2015.

[133] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust Region Policy Optimization. *International Conference on Machine Learning*, 2015.

[134] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *International Conference on Learning Representations*, 2015.

[135] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arxiv:1707.06347*, 2017.

[136] S. Shahrampour, A. Rakhlin, and A. Jadbabaie. Multi-armed bandits in multi-agent networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2786–2790, March 2017.

[137] Jun Shao and Dongsheng Tu. *The Jackknife and Bootstrap*. Springer Science & Business Media, 2012.

[138] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. *Proceedings of the International Conference on Machine Learning*, 2014.

[139] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[140] Max Simchowitz and Dylan J Foster. Naive exploration is optimal for online lqr. *arXiv preprint arXiv:2001.09576*, 2020.

[141] Max Simchowitz, Horia Mania, Stephen Tu, Michael I. Jordan, and Benjamin Recht. Learning without mixing: towards a sharp analysis of linear system identification. *Conference on Learning Theory*, pages 439–473, 2018.

[142] Max Simchowitz, Ross Boczar, and Benjamin Recht. Learning linear dynamical systems with semi-parametric least squares. *Conference on Learning Theory*, pages 2714–2802, 2019.

[143] Sumeet Singh, Spencer M. Richards, Vikas Sindhwani, Jean-Jacques E. Slotine, and Marco Pavone. Learning stabilizable nonlinear dynamics with contraction-based regularization. *arXiv preprint:1907.13122*, 2019.

[144] Jonas Sjöberg, Qinghua Zhang, Lennart Ljung, Albert Benveniste, Bernard Deylon, Pierre-Yves Glorennec, Håkan Hjalmarsson, and Anatoli Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31:1691–1724, 1995.

[145] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice Hall, 1991.

[146] Eduardo Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2):346–358, 1981.

[147] Ji-guang Sun. Perturbation theory for algebraic riccati equations. *SIAM Journal on Matrix Analysis and Applications*, 19(1):39–65, 1998.

[148] Ji-guang Sun. Sensitivity analysis of the discrete-time algebraic riccati equation. *Linear algebra and its applications*, 275:595–615, 1998.

[149] Yue Sun, Samet Oymak, and Maryam Fazel. Finite sample system identification: improved rates and the role of regularization. *Learning for Dynamics and Control*, 2020.

[150] Mario Sznaier, Takeshi Amishima, Pablo A Parrilo, and Jorge Tierno. A convex approach to robust $\mathcal{H}_2$ performance analysis. *Automatica*, 38(6), 2002.

[151] Russ Tedrake. Lqr-trees: Feedback motion planning on sparse randomized trees. 2009.

[152] William R. Thompson. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 25(3-4):285–294, 12 1933.

[153] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[154] Anastasios Tsiamis and George J. Pappas. Finite sample analysis of stochastic system identification. *IEEE Conference on Decision and Control*, 2019.

[155] Anastasios Tsiamis, Nikolai Matni, and George J. Pappas. Sample complexity of kalman filtering for unknown systems. *Learning for Dynamics and Control*, 2019.

[156] Stephen Tu and Benjamin Recht. The Gap Between Model-Based and Model-Free Methods on the Linear Quadratic Regulator: An Asymptotic Viewpoint. *arXiv:1812.03565*, 2018.

[157] Stephen Tu, Ross Boczar, Andrew Packard, and Benjamin Recht. Non-Asymptotic Analysis of Robust Control from Coarse-Grained Identification. *arXiv:1707.04791*, 2017.

[158] Aad W Van Der Vaart and Jon A Wellner. *Weak Convergence and Empirical Processes*. 1996.

[159] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[160] Andrew Wagenmaker and Kevin Jamieson. Active learning for identification of linear dynamical systems. *Conference on Learning Theory*, 2020.

[161] Yuh-Shyang Wang, Nikolai Matni, and John C Doyle. A System Level Approach to Controller Synthesis. *arXiv:1610.04815*, 2016.

[162] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. *Interational Conference on Learning Representations*, 2016.

[163] John S White. The limiting distribution of the serial correlation coefficient in the explosive case. *The Annals of Mathematical Statistics*, pages 1188–1197, 1958.

[164] Geoffrey Wolfer and Aryeh Kontorovich. Minimax learning of ergodic Markov chains. *International Conference on Algorithmic Learning Theory*, 2019.

[165] Fen Wu and Andy Packard. Optimal LQG performance of linear uncertain systems using state-feedback. In *American Control Conference*, 1995.

[166] Yuhuai Wu, Elman Mansimov, Shun Liao, Roger Grosse, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *Advances in Neural Information Processing Systems*, 2017.

[167] Huan Xu and Shie Mannor. Distributionally Robust Markov Decision Processes. *Mathematics of Operations Research*, 37(2):288–300, 2012.

[168] Boyan Yordanov, Jana Tumova, Ivana Cerna, Jiří Barnat, and Calin Belta. Temporal logic control of discrete-time piecewise affine systems. *IEEE Transactions on Automatic Control*, 57(6):1491–1504, 2011.

[169] Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and Optimal Control*. 1995.

[170] Matt Zucker, Nathan Ratliff, Anca D. Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher M. Dellin, J. Andrew Bagnell, and Siddhartha S. Srinivasa. Chomp: Covariant Hamiltonian optimization for motion planning. *The International Journal of Robotics Research*, 32(9-10):1164–1193, 2013.