

On the Hardness of Learning to Stabilize Linear Systems

Xiong Zeng¹ Zexiang Liu¹ Zhe Du¹ Necmiye Ozay¹ Mario Sznaier²

Abstract—Inspired by the work of Tsiamis et al. [1], in this paper we study the statistical hardness of learning to stabilize linear time-invariant systems. Hardness is measured by the number of samples required to achieve a learning task with a given probability. The work in [1] shows that there exist system classes that are hard to learn to stabilize with the core reason being the hardness of identification. Here we present a class of systems that can be easy to identify, thanks to a non-degenerate noise process that excites all modes, but the sample complexity of stabilization still increases exponentially with the system dimension. We tie this result to the hardness of co-stabilizability for this class of systems using ideas from robust control.

I. INTRODUCTION

Learning-based control plays an increasingly important role in many application domains such as power systems [2], robotics [3], self-driving cars [4], where it might be hard to perfectly model the system and its environment. Many learning-based control algorithms assume the existence of an initial stabilizing controller in order to simplify their analysis. Such simplifying assumptions are prevalent both in model-based [5]–[11] and model-free [12]–[18] learning-based control algorithms. However, learning to stabilize is a fundamental problem in learning-based control, with several algorithms tackling this issue [19]–[25].

Understanding the fundamental limits or the corner cases of learning-to-stabilize algorithms can inform future algorithm design and is crucial for applications of these algorithms in safety-critical domains. Therefore, it is important to understand how the system properties affect the performance of the learning-to-stabilize algorithms. In particular, we are interested in the number of samples required to learn a stabilizing controller with a given probability as a performance measure. We say a class of systems is hard to learn to stabilize if this number grows exponentially with the system dimension, independent of the algorithm choice.

We focus on fully observed linear time-invariant systems and consider the task of learning a static stabilizing linear state-feedback controller from a single trajectory. In this setting, Tsiamis et al. [1] show that when the process noise is degenerate, i.e. the noise covariance matrix being singular, there are some classes of systems that are hard to learn to stabilize, by transferring the hardness of learning-to-stabilize into the hardness of system identification. The

This work is supported in part by ONR CLEVR-AI MURI (#N00014-21-1-2431) and NSF CNS Grant #1931982.

¹ Department of Electrical Engineering and Computer Science, University of Michigan Ann Arbor, MI 48105. Emails: {zengxiong, zexiang, zhedu, necmiye}@umich.edu.

² Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115. Email: msznaier@ece.neu.edu.

system classes constructed in their work are based on a (marginally) stable hard-to-stabilize pair. In this work, we significantly extend the class of systems that are hard to learn to stabilize by considering systems that are, even though close in the parameter space and generate similar state-input trajectories, not co-stabilizable with the same controller. This is achieved by a novel analysis technique that uses Ackermann’s formula to compute all stabilizing linear state-feedback gains analytically and characterize the minimal level of perturbations to the parameters that render co-stabilizability infeasible. Different from the prior work, our analysis allows us to consider system classes that may only include systems with eigenvalues strictly outside of the unit circle, for which stabilizability is arguably more critical. **Notation:** We use lower case, lower case boldface, and upper case boldface letters to denote scalars, vectors, and matrices respectively. For a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, \mathbf{M}^\top denotes its transpose, $M^{(i,j)}$ denotes its element in the i^{th} row and the j^{th} column. For a square matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$, $\mathbf{M} \succ 0$ ($\succeq 0$) denotes that \mathbf{M} is positive definite (positive semidefinite), $\rho(\mathbf{M})$ denotes its spectral radius, and $\det(\mathbf{M})$ denotes its determinant. For a vector $\mathbf{v} \in \mathbb{R}^n$, its i^{th} element is denoted by $v^{(i)}$. By $\text{poly}(\cdot)$ we denote a polynomial function of its arguments. By $\exp(\cdot)$ we denote an exponential function of its arguments. We use \mathbf{I}_n to denote the identity matrix in $\mathbb{R}^{n \times n}$. A sequence of vectors $\mathbf{x}_t, \mathbf{x}_{t+1}, \dots, \mathbf{x}_{t+N}$ is denoted by $\mathbf{x}_{t:t+N}$ for short. By convention, $\mathbf{x}_{i:j}$ is an empty set if $j < i$.

II. PROBLEM SETUP AND PRELIMINARY NOTIONS

We consider the following fully-observed discrete-time linear time-invariant (LTI) system:

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

where $\mathbf{x}_t \in \mathbb{R}^n$, $\mathbf{u}_t \in \mathbb{R}^p$, $\mathbf{w}_t \in \mathbb{R}^n$ are the state, input, and process noise at time t . For simplicity, we assume $\mathbf{x}_0 = \mathbf{0}$. The random process \mathbf{w}_t over t is zero-mean i.i.d. Gaussian, with covariance matrix $\sigma_w^2 \mathbf{I}_n$. In the remainder of the paper, we denote a system in the form (1) by the tuple (\mathbf{A}, \mathbf{B}) .

Let \mathcal{C}_n be a class of systems (\mathbf{A}, \mathbf{B}) in dimension n , parameterized by some unknown parameters.

Definition 1. A learning-to-stabilize algorithm π with respect to the class \mathcal{C}_n is a sequence of functions $\pi = \{\pi_t\}_{t=0}^N$. For $t = 0, \dots, N-1$, $\pi_t(\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t})$ specifies the probability distribution of the input $\mathbf{u}_t \in \mathbb{R}^p$ at time t , conditioned on the previous state-input trajectory $\mathbf{u}_{0:t-1}$ and $\mathbf{x}_{0:t}$. Then at $t = N$, the function π_N maps the entire state-input trajectory $\mathbf{u}_{0:N-1}$ and $\mathbf{x}_{0:N}$ to a state-feedback gain in $\mathbb{R}^{p \times n}$. This

learned state-feedback gain $\hat{\mathbf{K}}_N = \pi_N(\mathbf{u}_{0:N-1}, \mathbf{x}_{1:N})$ is called stabilizing if $\rho(\mathbf{A} + \mathbf{B}\hat{\mathbf{K}}_N) < 1$.

Intuitively, the algorithm π consists of an exploration policy in the first $N - 1$ steps and decides on the gain $\hat{\mathbf{K}}_N$ using the data generated during exploration at step N . As such, exciting the system with some open-loop persistently exciting input as in data-driven control [26], applying some i.i.d. input and computing the gain afterward using the generated data [27], or active learning policies can all be considered as special types of learning-to-stabilize algorithms.

Given a system $\mathcal{S} = (\mathbf{A}, \mathbf{B}) \in \mathcal{C}_n$ and a learning-to-stabilize algorithm π , let $\mathbb{P}_{\mathcal{S}, \pi}^N$ denote the probability measure of the input-state samples $\mathbf{u}_{0:N-1}$ and $\mathbf{x}_{1:N}$ (with $f_{\mathcal{S}, \pi}^N$ denoting the corresponding probability density function), and $\mathbb{E}_{\mathcal{S}, \pi}^N$ denote the expectation of the respective probability measure. We make the following assumptions on the class \mathcal{C}_n and the algorithm π .

Assumption 1. For all $n \geq 1$ and all $(\mathbf{A}, \mathbf{B}) \in \mathcal{C}_n$, the norm of matrices \mathbf{A}, \mathbf{B} is bounded by a positive constant M , that is, $\max_{n \geq 1, (\mathbf{A}, \mathbf{B}) \in \mathcal{C}_n} \max \{\|\mathbf{A}\|_2, \|\mathbf{B}\|_2\} \leq M$.

Assumption 2. The second moment of the norm of the input signal \mathbf{u}_t , generated by the algorithm π , is bounded by some constant $\sigma_u^2 > 0$. That is, $\mathbb{E}_{\mathcal{S}, \pi} \left[\|\mathbf{u}_t\|_2^2 \right] \leq \sigma_u^2$.

Next, we recall the definition of poly(n)-stabilizable system classes from [1]. If a class \mathcal{C}_n of discrete-time LTI systems is poly(n)-stabilizable, it is statistically easy to learn linear state-feedback controllers to stabilize systems in this class.

Definition 2 (poly(n)-stabilizable system classes [1]). Under Assumptions 1 and 2, a class \mathcal{C}_n of systems is poly(n)-stabilizable if there exists a learning-to-stabilize algorithm π such that for all confidence levels $0 \leq \delta < 1$

$$\inf_{\mathcal{S} \in \mathcal{C}_n} \mathbb{P}_{\mathcal{S}, \pi}^N (\rho(\mathbf{A} + \mathbf{B}\pi_N(\mathbf{u}_{0:N-1}, \mathbf{x}_{1:N})) < 1) > 1 - \delta, \quad (2)$$

if the sample size N satisfies $N\sigma_u^2 \geq \text{poly}(n, \log(1/\delta), M)$.

This definition essentially tells that a class is poly(n)-stabilizable if it is possible to find an algorithm that can learn a stabilizing linear state-feedback controller with high probability, even for the worst-case system in this class, as long as there are polynomially many samples in the system dimension n . Since the polynomial dependency on n is mild, we say learning to stabilize is *easy* for this class. On the other hand, being *hard* refers to a class that is not poly(n)-stabilizable.

A closely related concept is the hardness of identification [28], i.e., whether the system can be learned with ϵ accuracy using poly($n, \log(1/\delta), 1/\epsilon$) many samples. When the process noise is degenerate, by transferring the hardness of learning to stabilize into the hardness of system identification, Tsiamis et al. [1] prove that there exists a class

of systems, for which the worst-case sample complexity of learning to stabilize is at least exponential with the system dimension. Our work is complementary as we seek to answer the following question.

Problem 1. Is there a class of linear systems that are not poly(n)-stabilizable when the process noise \mathbf{w}_t is non-degenerate?

The following lemma follows directly from Definition 2.

Lemma 1. For two classes of systems \mathcal{C}_n^1 and \mathcal{C}_n^2 , if \mathcal{C}_n^1 is a subset of \mathcal{C}_n^2 and \mathcal{C}_n^1 is not poly(n)-stabilizable, neither is \mathcal{C}_n^2 .

Lemma 1 turns Problem 1 into the problem of finding a pair of systems that are not poly(n)-stabilizable. Specifically, if a pair of systems is not poly(n)-stabilizable, then any class containing this pair of systems is also not poly(n)-stabilizable.

The next two definitions are related to the co-stabilizability and distinguishability of a pair of systems.

Definition 3 (Co-stabilizability). A pair of systems $\mathcal{S}_1 = (\mathbf{A}_1, \mathbf{B}_1)$ and $\mathcal{S}_2 = (\mathbf{A}_2, \mathbf{B}_2)$ is co-stabilizable if there exists a state-feedback gain \mathbf{K} such that both $\mathbf{A}_1 + \mathbf{B}_1\mathbf{K}$ and $\mathbf{A}_2 + \mathbf{B}_2\mathbf{K}$ are stable.

Remark 1. Co-stabilization problem for two dynamical systems has been studied in robust control [29], e.g., by using the gap metric [30].

We will use KL divergence to measure the distance between the distributions of state-input trajectories generated when the same exploration policy is applied to two different systems. A small KL divergence means that it is hard to distinguish two systems.

Definition 4 (Kullback–Leibler (KL) divergence). The KL divergence between the continuous distributions \mathbb{P} and \mathbb{Q} is defined as

$$\text{KL}(\mathbb{P}, \mathbb{Q}) = \int_{-\infty}^{+\infty} p(x) \log \frac{p(x)}{q(x)} dx,$$

where $p(x)$ and $q(x)$ denote the probability densities of \mathbb{P} and \mathbb{Q} and $p(x)$ is absolutely continuous with respect to $q(x)$.

Our main insight behind constructing not poly(n)-stabilizable pairs in the next section is as follows. If we have two different systems and excite all the modes of these systems, as we increase the trajectory length N , we expect that the KL divergence between the trajectories will increase and we will be able to distinguish the systems. On the other hand, if the KL divergence remains small independent of the exploration policy, then we cannot expect the learning-to-stabilize algorithm to result in significantly different controller gains. Moreover, if these two systems are not co-stabilizable, then learning to stabilize these systems will be hard.

III. HARD TO LEARN TO STABILIZE SYSTEMS

Consider the following system of the form (1) with (\mathbf{A}, \mathbf{B}) defined parametrically as

$$\mathbf{A} = \begin{bmatrix} r & v & 0 & \cdots & 0 \\ 0 & 0 & v & \cdots & 0 \\ & & \ddots & \ddots & \\ 0 & 0 & 0 & \cdots & v \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}, \quad \mathbf{B} = \begin{bmatrix} b^{(1)} \\ 0 \\ \vdots \\ 0 \\ v \end{bmatrix} \in \mathbb{R}^n, \quad (3)$$

where $n \geq 2$, $r > 1$, $0 < v < \frac{r-1}{2}$, and $b^{(1)} \geq 0$.

Remark 2. When $b^{(1)} = -v^n/r^{n-1}$, the system in (3) is uncontrollable. To avoid this trivially hard-to-stabilize case, we let $b^{(1)} \geq 0$.

The following proposition proves that there exist two systems in the parametric family (3) differing only in $b^{(1)}$, such that for a feedback gain to be able to stabilize both systems at the same time, the difference in $b^{(1)}$ should be exponentially small in the system dimension.

Proposition 1. Let $\mathcal{S}_1 = (\mathbf{A}, \mathbf{B}_1)$, and $\mathcal{S}_2 = (\mathbf{A}, \mathbf{B}_2)$, where \mathbf{A} is as in (3), and \mathbf{B}_1 and \mathbf{B}_2 equal to \mathbf{B} in (3) with $b^{(1)} = 0$ and $b^{(1)} = m \geq 0$, respectively. Let $\mathbf{K} \in \mathbb{R}^{1 \times n}$ be any stabilizing linear state-feedback gain for \mathcal{S}_1 such that $\rho(\mathbf{A} + \mathbf{B}_1\mathbf{K}) < 1$. Let $p_1^{cl}, p_2^{cl}, \dots, p_n^{cl}$ be the eigenvalues of $\mathbf{A} + \mathbf{B}_1\mathbf{K}$ with $0 \leq |p_1^{cl}|, |p_2^{cl}|, \dots, |p_n^{cl}| < 1$. Then $\rho(\mathbf{A} + \mathbf{B}_2\mathbf{K}) < 1$ only if

$$0 \leq m < v^n \prod_{i=1}^n \frac{1 + p_i^{cl}}{r - p_i^{cl}}. \quad (4)$$

The proof, which uses Ackermann's formula (Lemma 2) to analytically compute any stabilizing feedback gain of $(\mathbf{A}, \mathbf{B}_1)$ and Jury stability test (Lemma 4) to verify the closed-loop stability of $(\mathbf{A}, \mathbf{B}_2)$ when using the stabilizing gain of the former, is given in Appendix A.

Next, we upper bound the KL divergence between the probability distributions of length N input-state trajectories generated by the two LTI systems defined in Proposition 1. Similar upper bounds of the KL divergence between two LTI systems can also be found in [1], [28], [31].

Proposition 2. Let the systems \mathcal{S}_1 , and \mathcal{S}_2 be the same as those defined in Proposition 1. Let π be any learning-to-stabilize algorithm that satisfies Assumption 2. Then, the KL divergence between $\mathbb{P}_{\mathcal{S}_1, \pi}^N$ and $\mathbb{P}_{\mathcal{S}_2, \pi}^N$ satisfies

$$\text{KL}(\mathbb{P}_{\mathcal{S}_1, \pi}^N, \mathbb{P}_{\mathcal{S}_2, \pi}^N) \leq \frac{Nm^2\sigma_u^2}{2\sigma_w^2}.$$

The proof is given in Appendix B.

The next theorem states that there exist some classes of systems with non-degenerate process noise, for which the worst-case sample complexity of learning to stabilize is at least exponential with the system dimension n .

Theorem 1. Consider \mathcal{S}_1 and \mathcal{S}_2 defined in Proposition 1, with $m = 2\left(\frac{2v}{r-1}\right)^n$. Consider any class \mathcal{C}_n of systems including \mathcal{S}_1 and \mathcal{S}_2 , which satisfies Assumption 1. Then, for all learning-to-stabilize algorithms π satisfying Assumption 2 and for all confidence levels $0 < \delta < 1/2$, the requirement

$$\inf_{\mathcal{S} \in \mathcal{C}_n} \mathbb{P}_{\mathcal{S}, \pi}^N(\rho(\mathbf{A} + \mathbf{B}\pi_N(\mathbf{u}_{0:N-1}, \mathbf{x}_{1:N})) < 1) \geq 1 - \delta \quad (5)$$

is satisfied only if

$$N \geq \frac{\sigma_w^2}{2\sigma_u^2} \left(\frac{r-1}{2v}\right)^{2n} \log \frac{1}{3\delta},$$

where $n \geq 2$, $r > 1$, and $0 < v < \frac{r-1}{2}$.

The proof of Theorem 1 can be found in Appendix C. In the proof we show that if the same algorithm π is applied to \mathcal{S}_1 and \mathcal{S}_2 , for the stabilization probability in (5) to be high for both, exponentially many samples are needed. This indicates that for any class containing \mathcal{S}_1 and \mathcal{S}_2 , polynomially many samples will not be sufficient for the satisfaction of requirement (5), therefore such classes cannot be poly(n)-stabilizable.

Comparing the systems \mathcal{S}_1 and \mathcal{S}_2 in our proof to corresponding system pairs in [1], our pairs are individually not necessarily "hard to identify" but the distance m in the parameter space between the pairs shrinks exponentially fast as we increase n . As shown in Proposition 2, the input-state trajectory distributions our pairs of systems generate look very similar; this is expected since the system parameters get closer with n . In general, one may expect if two systems are close to each other in the parameter space, they can be co-stabilized by the same controller \mathbf{K} . However, our pairs cannot be co-stabilized (as shown in Proposition 1) with a single gain \mathbf{K} although the systems are very close in parameter space, which is the main source of hardness.

Remark 3. Our proof technique can also be extended to show the hardness of learning to stabilize for classes of systems containing single-input systems with diagonal state matrices and n unstable eigenvalues in a compact range, presented in [32]. In that case, when the input vector is the all-one vector, the controllability matrix of the system is a Vandermonde matrix, which allows us to again use Ackermann's formula to obtain the explicit form of all stabilizing linear state-feedback gains. Results similar to Proposition 1 and Theorem 1 can be established in this case too.

IV. NUMERICAL EXPERIMENTS

In this section, we implement two numerical experiments, i.e., certainty equivalent linear quadratic regulator (LQR) and robust control, to show the hardness of stabilization.

A. Certainty Equivalent LQR

Since solving LQR problems always gives stabilizing controllers (under mild regularity conditions), the first experiment considers the certainty equivalent LQR control [27]. Specifically, a controller is computed by solving an LQR

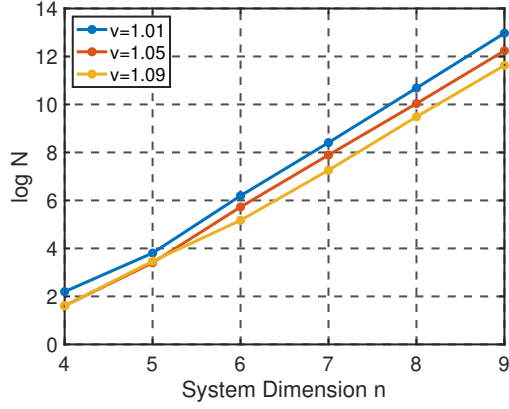


Fig. 1. Required trajectory length N for 90% stabilization rate vs. system dimension n : $\sigma_u^2 = 32$, $\sigma_w^2 = 0.005$, $v \in \{1.01, 1.05, 1.09\}$

problem using some estimated system dynamics and then applied to the ground truth system. The infinite-horizon LQR problem, simplified as $dLQR(\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R})$, is as follows.

$$\min_{\mathbf{u}_0, \mathbf{u}_1, \dots} \lim_{T \rightarrow \infty} \mathbb{E} \left[\frac{1}{T} \sum_{t=0}^T (\mathbf{x}_t^\top \mathbf{Q} \mathbf{x}_t + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t) \right] \quad (6)$$

s.t. (1)

where \mathbf{Q}, \mathbf{R} are positive semi-definite cost matrices. Its solution is given by $\mathbf{u}_t = \mathbf{K} \mathbf{x}_t$ where the controller \mathbf{K} can be computed by solving the Riccati equation. Consider the system $(\mathbf{A}, \mathbf{B}_1)$ defined in Proposition 1, and let $\mathbf{Q} = \mathbf{I}_n$, and $\mathbf{R} = 1$. Since the analysis of Theorem 1 is established on perturbing $b^{(1)}$ in \mathbf{B}_1 , we consider a simplified setting where only $b^{(1)}$ is unknown and to be estimated using the least squares estimator, which is denoted by $\hat{b}^{(1)}$. Let $\hat{\mathbf{B}}_1$ denote the matrix by replacing $b^{(1)}$ with $\hat{b}^{(1)}$ and $\hat{\mathbf{K}}$ denote the certainty equivalent controller for $(\mathbf{A}, \mathbf{B}_1)$ obtained by solving $dLQR(\mathbf{A}, \hat{\mathbf{B}}_1, \mathbf{Q}, \mathbf{R})$. We let the input $u_t \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma_u^2)$. Since there is only a single unknown parameter and its regressor u_t is independent, this system is trivially easy to identify.

For each dimension n , we run $M = 200$ independent experiments. Let $\hat{\mathbf{K}}_{i, N'}$ denote the controller obtained using the first N' data points, i.e., $\{\mathbf{u}_{0:N'-1}, \mathbf{x}_{1:N'}\}$, in the i^{th} experiment. We record the smallest trajectory length N under which at least 90% of the experiments produce stabilizing controllers, i.e.

$$N := \min \left\{ N' \in \mathbb{N} : \frac{1}{M} \sum_{i \in [M]} \mathbb{I}_{\{\rho(\mathbf{A} + \mathbf{B}_1 \hat{\mathbf{K}}_{i, N'}) < 1\}} \geq 0.9 \right\}, \quad (7)$$

where \mathbb{I} denotes the indicator function.

The results are given in Fig. 1. According to Fig. 1, we have that as the system dimension increases, the required number of samples for a given frequency of stability increases exponentially with the system dimension.

B. LMI-based Sufficient Condition for Co-stabilizability

In this section, we numerically demonstrate the hardness of co-stabilizability of $\mathcal{S}_1 = (\mathbf{A}, \mathbf{B}_1)$ and $\mathcal{S}_2 = (\mathbf{A}, \mathbf{B}_2(m))$

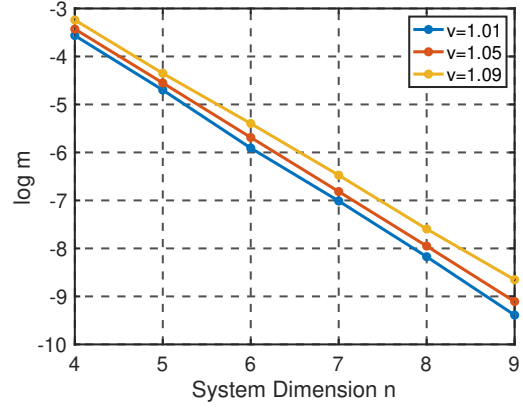


Fig. 2. $v = 1.01, 1.05, 1.09$, and $r = 3.2$. The x -axis is the system dimension n and the y -axis is the logarithm of the largest m such that the problem in (8) is feasible.

using ideas from robust control, where we leave m as a parameter. We use the following feasibility problem, which can be converted to an LMI, to check sufficient conditions of co-stabilizability.

$$\begin{aligned} & \text{find } \mathbf{K}, \mathbf{P} \\ & \text{s.t. } (\mathbf{A} + \mathbf{B}_1 \mathbf{K})^\top \mathbf{P} (\mathbf{A} + \mathbf{B}_1 \mathbf{K}) \prec \mathbf{P} \\ & \quad (\mathbf{A} + \mathbf{B}_2(m) \mathbf{K})^\top \mathbf{P} (\mathbf{A} + \mathbf{B}_2(m) \mathbf{K}) \prec \mathbf{P} \\ & \quad \mathbf{P} \succ 0 \end{aligned} \quad (8)$$

We use the bisection method to find the largest m such that the problem (8) is feasible. The results are shown in Fig. 2. According to this figure, we see that as the system dimension increases, the largest m such that the LMI optimization problem in (8) is feasible decreases exponentially with increasing system dimension, which is consistent with Eq. (4) in Proposition 1.

V. CONCLUSION AND FUTURE WORK

In this work, we identified an extended class of LTI systems that are hard to learn to stabilize with static state feedback. The main idea in constructing such examples is to find pairs of systems whose parameters become exponentially close to each other as the dimension increases, yet they are not co-stabilizable. One interesting observation is that the entries of stabilizing gains for these pairs are also growing exponentially (see, Eq. (17)). In the future, we want to investigate the ramifications of this observation in gradient-based learning algorithms used for control as in [33].

Acknowledgments: The authors would like to thank Prof. Peter Seiler of University of Michigan for some early discussions that motivated this work.

APPENDIX

A. Proof of Proposition 1

We first introduce a few lemmas used in the proof of Proposition 1. The first lemma parameterizes all stabilizing state-feedback gains for single-input controllable LTI

systems. Recall that the controllability matrix of a system (\mathbf{A}, \mathbf{B}) is defined by

$$\mathbf{Ctr}_{(\mathbf{A}, \mathbf{B})} = \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}. \quad (9)$$

Lemma 2 (Ackermann's formula [34]). *Consider the following order n single-input controllable system (\mathbf{A}, \mathbf{B}) with state feedback $\mathbf{K} \in \mathbb{R}^{1 \times n}$:*

$$\begin{cases} \mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t \\ \mathbf{u}_t = \mathbf{K}\mathbf{x}_t \end{cases}$$

Given n desired eigenvalues of $\mathbf{A} + \mathbf{B}\mathbf{K}$, the unique state feedback that achieves these closed-loop eigenvalues is:

$$\mathbf{K} = -\mathbf{e}_n^\top \mathbf{Ctr}_{(\mathbf{A}, \mathbf{B})}^{-1} \Delta^{\text{cl}}(\mathbf{A}), \quad (10)$$

where \mathbf{e}_n is the last column of the $n \times n$ identity matrix, and $\Delta^{\text{cl}}(\mathbf{A})$ is the characteristic polynomial of $\mathbf{A} + \mathbf{B}\mathbf{K}$ evaluated at \mathbf{A} .

The next lemma derives the expression of the first element of any stabilizing state-feedback gains for $(\mathbf{A}, \mathbf{B}_1)$, parameterized by the stable closed-loop poles.

Lemma 3. *For the system $(\mathbf{A}, \mathbf{B}_1)$ defined in Proposition 1 and any stabilizing state feedback $\mathbf{K} \in \mathbb{R}^{1 \times n}$, let $\{p_k^{\text{cl}}\}_{k=1}^n$ be the eigenvalues of $\mathbf{A} + \mathbf{B}_1\mathbf{K}$, with $\{p_k^{\text{cl}}\}_{k=1}^n$ all inside the unit circle. Then, the first element k_1 of the state feedback \mathbf{K} satisfies*

$$k_1 = -\frac{(r - p_1^{\text{cl}})(r - p_2^{\text{cl}}) \cdots (r - p_n^{\text{cl}})}{v^n}. \quad (11)$$

Proof. By Lemma 2, since $(\mathbf{A}, \mathbf{B}_1)$ is single-input and controllable, the state feedback \mathbf{K} satisfies

$$\mathbf{K} = -\mathbf{e}_n^\top \mathbf{Ctr}_{(\mathbf{A}, \mathbf{B}_1)}^{-1} \Delta^{\text{cl}}(\mathbf{A}), \quad (12)$$

where the characteristic polynomial $\Delta^{\text{cl}}(\mathbf{A})$ of the closed-loop system $\mathbf{A} + \mathbf{B}_1\mathbf{K}$ evaluated at \mathbf{A} is

$$\Delta^{\text{cl}}(\mathbf{A}) = \prod_{i=1}^n (\mathbf{A} - p_i^{\text{cl}} \mathbf{I}_n). \quad (13)$$

By the definition of \mathbf{A} in (3), for all $i = 1, 2, \dots, n$,

$$\mathbf{A} - p_i^{\text{cl}} \mathbf{I}_n = \begin{bmatrix} r - p_i^{\text{cl}} & v & 0 & \cdots & 0 \\ 0 & -p_i^{\text{cl}} & v & \cdots & 0 \\ & & \ddots & \ddots & \\ 0 & 0 & 0 & \cdots & v \\ 0 & 0 & 0 & \cdots & -p_i^{\text{cl}} \end{bmatrix}. \quad (14)$$

Based on (14), the element of $\Delta^{\text{cl}}(\mathbf{A})$ at the first row and the first column is

$$[\Delta^{\text{cl}}(\mathbf{A})]^{(1,1)} = (r - p_1^{\text{cl}})(r - p_2^{\text{cl}}) \cdots (r - p_n^{\text{cl}}). \quad (15)$$

Furthermore, due to the special structures of $(\mathbf{A}, \mathbf{B}_1)$, it can be shown that the last row of the inverse of the controllability matrix $\mathbf{Ctr}_{(\mathbf{A}, \mathbf{B}_1)}$ is

$$\mathbf{e}_n^\top \mathbf{Ctr}_{(\mathbf{A}, \mathbf{B}_1)}^{-1} = [v^{-n} \ 0 \ \cdots \ 0]. \quad (16)$$

Thus, according to (12), (15), and (16), the first element of the state feedback \mathbf{K} is

$$\begin{aligned} k_1 &= -v^{-n} [\Delta^{\text{cl}}(\mathbf{A})]^{(1,1)} \\ &= -\frac{(r - p_1^{\text{cl}})(r - p_2^{\text{cl}}) \cdots (r - p_n^{\text{cl}})}{v^n}. \end{aligned} \quad (17)$$

□

The next lemma provides a necessary condition for the stability of discrete-time LTI systems.

Lemma 4 (Jury stability test, Theorem 4.6 in [35]). *For the polynomial*

$$\Delta(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0 = 0,$$

with $a_n > 0$, the roots of the polynomial are inside the unit circle only if

$$\begin{cases} \Delta(1) > 0 \\ (-1)^n \Delta(-1) > 0 \end{cases}. \quad (18)$$

Now, we are ready to present the proof of Proposition 1.

Proof. Consider the two systems $(\mathbf{A}, \mathbf{B}_1)$ and $(\mathbf{A}, \mathbf{B}_2)$ in Proposition 1. Let \mathbf{K} be any stabilizing state-feedback gain of $(\mathbf{A}, \mathbf{B}_1)$. By Lemma 3, the first element of \mathbf{K} satisfies

$$k_1 = -\frac{(r - p_1^{\text{cl}})(r - p_2^{\text{cl}}) \cdots (r - p_n^{\text{cl}})}{v^n}, \quad (19)$$

where $p_1^{\text{cl}}, p_2^{\text{cl}}, \dots, p_n^{\text{cl}}$ are the eigenvalues of $\mathbf{A} + \mathbf{B}_1\mathbf{K}$ with $|p_1^{\text{cl}}|, |p_2^{\text{cl}}|, \dots, |p_n^{\text{cl}}| < 1$. Next, it can be shown that the characteristic polynomial $\Delta_n^{\text{cl}}(z)$ of $\mathbf{A} + \mathbf{B}_1\mathbf{K}$ is

$$\begin{aligned} \Delta_n^{\text{cl}}(z) &:= \det(z\mathbf{I} - \mathbf{A} - \mathbf{B}_1\mathbf{K}) \\ &= z^n - (r + vk_n)z^{n-1} \\ &\quad + \sum_{j=0}^{n-2} v^{n-j-1} (rk_{j+2} - vk_{j+1})z^j \end{aligned}. \quad (20)$$

Similarly, one can show that the characteristic polynomial $\hat{\Delta}_n^{\text{cl}}(z)$ of $\mathbf{A} + \mathbf{B}_2\mathbf{K}$ satisfies

$$\begin{aligned} \hat{\Delta}_n^{\text{cl}}(z) &:= \det(z\mathbf{I} - \mathbf{A} - \mathbf{B}_2\mathbf{K}) \\ &= z^n - (r + vk_n + k_1 m)z^{n-1} + \\ &\quad \sum_{j=0}^{n-2} v^{n-j-1} (rk_{j+2} - vk_{j+1})z^j \\ &= \Delta_n^{\text{cl}}(z) - mk_1 z^{n-1}. \end{aligned} \quad (21)$$

Thus, by (21), we have

$$\begin{cases} \hat{\Delta}_n^{\text{cl}}(1) = \Delta_n^{\text{cl}}(1) - mk_1, \\ \hat{\Delta}_n^{\text{cl}}(-1) = \Delta_n^{\text{cl}}(-1) - (-1)^{n-1} mk_1. \end{cases} \quad (22)$$

By Lemma 4, the matrix $\mathbf{A} + \mathbf{B}_2\mathbf{K}$ is stable only if

$$\begin{cases} \hat{\Delta}_n^{\text{cl}}(1) > 0 \\ (-1)^n \hat{\Delta}_n^{\text{cl}}(-1) > 0 \end{cases}. \quad (23)$$

Also, note that

$$\begin{cases} \Delta_n^{\text{cl}}(1) = \prod_{i=1}^n (1 - p_i^{\text{cl}}) \\ \Delta_n^{\text{cl}}(-1) = (-1)^n \prod_{i=1}^n (1 + p_i^{\text{cl}}) \end{cases} \quad (24)$$

Combining (22), (23), and (24), we have that $\mathbf{A} + \mathbf{B}_2\mathbf{K}$ is stable only if

$$0 \leq m < v^n \prod_{i=1}^n \frac{1 + p_i^{\text{cl}}}{r - p_i^{\text{cl}}}, \quad (25)$$

with $|p_1^{\text{cl}}|, |p_2^{\text{cl}}|, \dots, |p_n^{\text{cl}}| < 1$. \square

B. Proof of Proposition 2

For simplicity of notation, we denote $\mathbb{P}_{\mathcal{S}_i, \pi}^t$, $f_{\mathcal{S}_i, \pi}^t$, and $\mathbb{E}_{\mathcal{S}_1, \pi}^t$ by \mathbb{P}_i^t , f_i^t , and \mathbb{E}_1^t respectively, for $i = 1, 2$, and $0 \leq t \leq N$. With this notation, Proposition 2 can be proven as follows.

Proof. Starting with the definition of KL divergence (i.e., Definition 4), we have

$$\begin{aligned} \text{KL}(\mathbb{P}_1^N, \mathbb{P}_2^N) &= \mathbb{E}_1^N \left[\log \frac{f_1^N(\mathbf{u}_{0:N-1}, \mathbf{x}_{0:N})}{f_2^N(\mathbf{u}_{0:N-1}, \mathbf{x}_{0:N})} \right] \\ &= \mathbb{E}_1^N \left[\log \frac{\prod_{t=0}^N f_1^t(\mathbf{x}_t | \mathbf{x}_{0:t-1}, \mathbf{u}_{0:t-1})}{\prod_{t=0}^N f_2^t(\mathbf{x}_t | \mathbf{x}_{0:t-1}, \mathbf{u}_{0:t-1})} \right] \\ &\quad + \mathbb{E}_1^N \left[\log \frac{\prod_{t=0}^{N-1} f_1^t(\mathbf{u}_t | \mathbf{x}_{0:t}, \mathbf{u}_{0:t-1})}{\prod_{t=0}^{N-1} f_2^t(\mathbf{u}_t | \mathbf{x}_{0:t}, \mathbf{u}_{0:t-1})} \right] \\ &= \sum_{t=0}^N \mathbb{E}_1^t \left[\log \frac{f_1^t(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1})}{f_2^t(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1})} \right], \end{aligned} \quad (26)$$

where the second equality is from the properties of the conditional probability density functions and the third equality is because the exploration policies of these two systems are the same and the discrete-time LTI system has the Markovian structure.

Based on the special structure of (\mathbf{A}, \mathbf{B}) of \mathcal{S}_1 and \mathcal{S}_2 , we have the following relationships between every element of state vectors of these systems:

$$\mathcal{S}_1 : \begin{cases} x_t^{(1)} = rx_{t-1}^{(1)} + vx_{t-1}^{(2)} + w_{t-1}^{(1)} \\ x_t^{(j)} = vx_{t-1}^{(j+1)} + w_{t-1}^{(j)}, \text{ for } j = 2, \dots, n-1 \\ x_t^{(n)} = v\mathbf{u}_{t-1} + w_{t-1}^{(n)} \end{cases}, \quad (27)$$

$$\mathcal{S}_2 : \begin{cases} x_t^{(1)} = rx_{t-1}^{(1)} + vx_{t-1}^{(2)} + m\mathbf{u}_{t-1} + w_{t-1}^{(1)} \\ x_t^{(j)} = vx_{t-1}^{(j+1)} + w_{t-1}^{(j)}, \text{ for } j = 2, \dots, n-1 \\ x_t^{(n)} = v\mathbf{u}_{t-1} + w_{t-1}^{(n)} \end{cases}. \quad (28)$$

Due to (27), (28) and the fact that $w_t^{(j)}$ for $j = 1, \dots, n$ are mutually independent, we have for $i = 1$ and 2,

$$\begin{aligned} &f_i^t(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \\ &= f_i^t \left(x_t^{(1)} | x_{t-1}^{(1)}, x_{t-1}^{(2)}, \mathbf{u}_{t-1} \right) \\ &\quad \prod_{j=2}^{n-1} f_i^t \left(x_t^{(j)} | x_{t-1}^{(j+1)} \right) f_i^t \left(x_t^{(n)} | \mathbf{u}_{t-1} \right). \end{aligned} \quad (29)$$

According to (27) and (28), we also have

$$\begin{cases} f_1^t \left(x_t^{(j)} | x_{t-1}^{(j+1)} \right) = f_2^t \left(x_t^{(j)} | x_{t-1}^{(j+1)} \right) \\ \text{for } j = 2, \dots, n-1 \\ f_1^t \left(x_t^{(n)} | \mathbf{u}_{t-1} \right) = f_2^t \left(x_t^{(n)} | \mathbf{u}_{t-1} \right) \end{cases}, \quad (30)$$

and

$$\begin{cases} f_1^t \left(x_t^{(1)} | x_{t-1}^{(1)}, x_{t-1}^{(2)}, \mathbf{u}_{t-1} \right) \sim \\ \mathcal{N} \left(rx_{t-1}^{(1)} + vx_{t-1}^{(2)}, \sigma_w^2 \right) \\ f_2^t \left(x_t^{(1)} | x_{t-1}^{(1)}, x_{t-1}^{(2)}, \mathbf{u}_{t-1} \right) \sim \\ \mathcal{N} \left(rx_{t-1}^{(1)} + vx_{t-1}^{(2)} + m\mathbf{u}_{t-1}, \sigma_w^2 \right) \end{cases}, \quad (31)$$

where \mathcal{N} denotes the Gaussian distribution.

Then, (26) is equal to

$$\begin{aligned} \text{KL}(\mathbb{P}_1^N, \mathbb{P}_2^N) &= \sum_{t=1}^N \mathbb{E}_1^t \left[\log \frac{f_1^t \left(x_t^{(1)} | x_{t-1}^{(1)}, x_{t-1}^{(2)}, \mathbf{u}_{t-1} \right)}{f_2^t \left(x_t^{(1)} | x_{t-1}^{(1)}, x_{t-1}^{(2)}, \mathbf{u}_{t-1} \right)} \right] \\ &= \sum_{t=1}^N \mathbb{E}_1^t \left[\frac{\left(w_{t-1}^{(1)} + m\mathbf{u}_{t-1} \right)^2 - \left(w_{t-1}^{(1)} \right)^2}{2\sigma_w^2} \right] \\ &= \sum_{t=1}^N \mathbb{E}_1^t \left[\frac{\left(m\mathbf{u}_{t-1} \right)^2}{2\sigma_w^2} \right] \\ &\leq \frac{Nm^2\sigma_u^2}{2\sigma_w^2}, \end{aligned} \quad (32)$$

where the first equality is due to (29) and (30), the second equality is due to (27), (31), and the definition of the Gaussian distribution, the third equality is by the noise process being zero mean and $w_{t-1}^{(1)}$ and \mathbf{u}_{t-1} being independent, and the last inequality is due to Assumption 2. \square

C. Proof of Theorem 1

Before presenting the proof of Theorem 1, we first introduce Birgé's inequality, a classical inequality from information theory.

Lemma 5 (Birgé's Inequality, Theorem 4.21 in [36]). *Let Ω be a set and \mathcal{E} be a σ -algebra on the set Ω . Let $\mathbb{P}_1, \mathbb{P}_2$ be probability measures on the probability space (Ω, \mathcal{E}) and let $E_1, E_2 \in \mathcal{E}$ be disjoint events. If $1 - \delta \triangleq \min_{i=1,2} \mathbb{P}_i(E_i) \geq 1/2$ then*

$$\text{KL}(\mathbb{P}_1, \mathbb{P}_2) \geq (1 - \delta) \log \frac{1 - \delta}{\delta} + \delta \log \frac{\delta}{1 - \delta}.$$

Next, we present the proof of Theorem 1.

Proof. Given \mathcal{S}_1 and \mathcal{S}_2 , let us define two events:

$$\begin{aligned} E_1 &= \{ \mathbf{u}_{0:N-1}, \mathbf{x}_{1:N} \mid \rho(\mathbf{A} + \mathbf{B}_1\pi_N(\mathbf{u}_{0:N-1}, \mathbf{x}_{1:N})) < 1 \}, \\ E_2 &= \{ \mathbf{u}_{0:N-1}, \mathbf{x}_{1:N} \mid \rho(\mathbf{A} + \mathbf{B}_2\pi_N(\mathbf{u}_{0:N-1}, \mathbf{x}_{1:N})) < 1 \}. \end{aligned}$$

Since $m = 2 \left(\frac{2v}{r-1} \right)^n > v^n \prod_{i=1}^n \frac{1+p_i^{\text{cl}}}{r-p_i^{\text{cl}}}$ for any stable closed-loop poles, by Proposition 1, \mathcal{S}_1 and \mathcal{S}_2 cannot be co-stabilized. Hence, E_1 and E_2 are disjoint events.

Suppose (5) is true, which implies

$$\begin{cases} \mathbb{P}_{\mathcal{S}_1, \pi}^N(E_1) \geq 1 - \delta, \\ \mathbb{P}_{\mathcal{S}_2, \pi}^N(E_2) \geq 1 - \delta. \end{cases} \quad (33)$$

Therefore, we can apply Lemma 5 to obtain

$$\begin{aligned} \text{KL}(\mathbb{P}_{\mathcal{S}_1, \pi}^N, \mathbb{P}_{\mathcal{S}_2, \pi}^N) &\geq (1 - \delta) \log \frac{1 - \delta}{\delta} + \delta \log \frac{\delta}{1 - \delta} \\ &\geq \log \left(\frac{1}{3\delta} \right). \end{aligned} \quad (34)$$

According to Proposition 2, the KL divergence between $\mathbb{P}_{\mathcal{S}_1, \pi}^N$ and $\mathbb{P}_{\mathcal{S}_2, \pi}^N$ satisfies

$$\text{KL}(\mathbb{P}_{\mathcal{S}_1, \pi}^N, \mathbb{P}_{\mathcal{S}_2, \pi}^N) \leq \frac{Nm^2\sigma_u^2}{2\sigma_w^2} \leq \frac{2N\sigma_u^2}{\sigma_w^2} \left(\frac{2v}{r-1} \right)^{2n}. \quad (35)$$

Combining (35) and (34), we have that (5) holds only if

$$N \geq \frac{\sigma_w^2}{2\sigma_u^2} \left(\frac{r-1}{2v} \right)^{2n} \log \left(\frac{1}{3\delta} \right). \quad (36)$$

□

REFERENCES

- [1] A. Tsiamis, I. M. Ziemann, M. Morari, N. Matni, and G. J. Pappas, "Learning to control linear systems can be hard," in *Conference on Learning Theory*. PMLR, 2022, pp. 3820–3857.
- [2] T. Liu, Y. Song, L. Zhu, and D. J. Hill, "Stability and control of power grids," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 689–716, 2022.
- [3] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.
- [4] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 187–210, 2018.
- [5] A. Cohen, T. Koren, and Y. Mansour, "Learning linear-quadratic regulators efficiently with only \sqrt{T} regret," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1300–1309.
- [6] M. Simchowitz, K. Singh, and E. Hazan, "Improper learning for non-stochastic control," in *Conference on Learning Theory*. PMLR, 2020, pp. 3320–3436.
- [7] Y. Zheng, L. Frieri, M. Kamgarpour, and N. Li, "Sample complexity of linear quadratic gaussian (LQG) control for output feedback systems," in *Learning for dynamics and control*. PMLR, 2021, pp. 559–570.
- [8] Y. Ouyang, M. Gagrani, and R. Jain, "Posterior sampling-based reinforcement learning for control of unknown linear systems," *IEEE Transactions on Automatic Control*, vol. 65, no. 8, pp. 3600–3607, 2019.
- [9] M. K. S. Faradonbeh, A. Tewari, and G. Michailidis, "Optimism-based adaptive regulation of linear-quadratic systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1802–1808, 2020.
- [10] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "Regret bounds for robust adaptive control of the linear quadratic regulator," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [11] Y. Sattar, Z. Du, D. A. Tarzanagh, L. Balzano, N. Ozay, and S. Oymak, "Identification and adaptive control of markov jump systems: Sample complexity and regret bounds," *arXiv preprint arXiv:2111.07018*, 2021.
- [12] M. Fazel, R. Ge, S. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1467–1476.
- [13] A. Cassel, A. Cohen, and T. Koren, "Logarithmic regret for learning linear quadratic regulators efficiently," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1328–1337.
- [14] K. Zhang, B. Hu, and T. Basar, "Policy optimization for \mathcal{H}_2 linear control with \mathcal{H}_∞ robustness guarantee: Implicit regularization and global convergence," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 179–190.
- [15] Y. Tang, Y. Zheng, and N. Li, "Analysis of the optimization landscape of linear quadratic gaussian (LQG) control," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 599–610.
- [16] Y. Li, Y. Tang, R. Zhang, and N. Li, "Distributed reinforcement learning for decentralized linear quadratic control: A derivative-free policy optimization approach," *IEEE Transactions on Automatic Control*, 2021.
- [17] H. Mohammadi, A. Zare, M. Soltanolkotabi, and M. R. Jovanović, "Convergence and sample complexity of gradient methods for the model-free linear-quadratic regulator problem," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2435–2450, 2021.
- [18] Z. Yang, Y. Chen, M. Hong, and Z. Wang, "Provably global convergence of actor-critic: A case for linear quadratic regulator with ergodic cost," *Advances in neural information processing systems*, vol. 32, 2019.
- [19] Y. Abbasi-Yadkori and C. Szepesvári, "Regret bounds for the adaptive control of linear quadratic systems," in *Proceedings of the 24th Annual Conference on Learning Theory*. JMLR Workshop and Conference Proceedings, 2011, pp. 1–26.
- [20] M. K. S. Faradonbeh, A. Tewari, and G. Michailidis, "Finite-time adaptive stabilization of linear systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3498–3505, 2018.
- [21] S. Lale, K. Azizzadenesheli, B. Hassibi, and A. Anandkumar, "Reinforcement learning with fast stabilization in linear dynamical systems," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 5354–5390.
- [22] X. Chen and E. Hazan, "Black-box control for linear dynamical systems," in *Conference on Learning Theory*. PMLR, 2021, pp. 1114–1143.
- [23] Y. Hu, A. Wierman, and G. Qu, "On the sample complexity of stabilizing LTI systems on a single trajectory," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16 989–17 002, 2022.
- [24] T. Dai, M. Sznajder, and B. R. Solvas, "Data-driven quadratic stabilization of continuous lti systems," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 3965–3970, 2020.
- [25] J. Perdomo, J. Umenberger, and M. Simchowitz, "Stabilizing dynamical systems via policy gradient methods," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 274–29 286, 2021.
- [26] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2019.
- [27] H. Mania, S. Tu, and B. Recht, "Certainty equivalence is efficient for linear quadratic control," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [28] A. Tsiamis and G. J. Pappas, "Linear systems can be hard to learn," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 2903–2910.
- [29] K. Zhou and J. C. Doyle, *Essentials of robust control*. Prentice hall Upper Saddle River, NJ, 1998, vol. 104.
- [30] T. Georgiou and M. Smith, "Optimal robustness in the gap metric," *IEEE Transactions on Automatic Control*, vol. 35, no. 6, pp. 673–686, 1990.
- [31] Y. Jedra and A. Proutiere, "Sample complexity lower bounds for linear system identification," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 2676–2681.
- [32] J. Li, S. Sun, and Y. Mo, "Fundamental limit on siso system identification," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 856–861.
- [33] I. Ziemann, A. Tsiamis, H. Sandberg, and N. Matni, "How are policy gradient methods affected by the limits of control?" in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 5992–5999.
- [34] J. Ackermann, "Der entwurf linearer regelungssysteme im zustandsraum," *at-Automatisierungstechnik*, vol. 20, no. 1-12, pp. 297–300, 1972.
- [35] M. S. Fadali and A. Visioli, *Digital control engineering: analysis and design*. Academic Press, 2012.
- [36] S. Boucheron, G. Lugosi, and P. Massart, *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.