

# RANDOMIZED ALGORITHMS FOR DATA-DRIVEN STABILIZATION OF STOCHASTIC LINEAR SYSTEMS

Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, George Michailidis

## ABSTRACT

Data-driven control strategies for dynamical systems with unknown parameters are popular in theory and applications. An essential problem is to prevent stochastic linear systems becoming destabilized, due to the uncertainty of the decision-maker about the dynamical parameter. Two randomized algorithms are proposed for this problem, but the performance is not sufficiently investigated. Further, the effect of key parameters of the algorithms such as the magnitude and the frequency of applying the randomizations is not currently available. This work studies the stabilization speed and the failure probability of data-driven procedures. We provide numerical analyses for the performance of two methods: stochastic feedback, and stochastic parameter. The presented results imply that as long as the number of statistically independent randomizations is not too small, fast stabilization is guaranteed.

**Index Terms**— randomized algorithms, fast stabilization, stochastic feedback, stochastic parameter, unstable dynamics.

## 1. INTRODUCTION

Sequential decision-making strategies for stochastic linear systems is an ubiquitous model extensively used in different fields such as biology, robotics, finance, and cryptography [1, 2, 3, 4, 5]. The setting consists of a state-space dynamical model following the stochastic trajectory

$$x(t+1) = A_0x(t) + B_0u(t) + \xi(t+1). \quad (1)$$

That is, the current state vector  $x(t) \in \mathbb{R}^p$  together with the current action  $u(t) \in \mathbb{R}^r$  lead the system to the next state  $x(t+1)$ , according to (1). The linear dynamical model is being disturbed by the stochastic noise process  $\{\xi(t)\}_{t=1}^{\infty}$ . The state transition matrix  $A_0$ , and the action influence matrix  $B_0$ , both of appropriate dimensions, are *unknown*.

The objective is to analyze the performance of adaptive stabilization procedures. The focus is on randomized algorithms for stabilizing the system by designing the control actions according to the observed state sequence. Note that the

control action can cause instability of the state vector. That is due to the uncertainty of the decision maker about the dynamics matrices  $A_0, B_0$ . After the stabilization procedure, the control policy for the operating plant under consideration needs to be designed [6, 7, 8]. The real applications determine the goal being for example minimizing the input energy or regulating the state vector. Therefore, a reliable stabilization algorithm is required to be executed prior to the regulation step, and is desired to guarantee the stabilization quite fast. In general, design of effective methods for finite time stabilization has been recognized as a difficult problem [9, 10].

Next, we briefly review various algorithms proposed in the literature for stabilization. Early work consider a restricted setting where the operator norm of the closed-loop matrix is less than one [11]. Then, employing parameter estimation and policy design in an alternate manner, the system is shown to be stabilized [12]. Ensuing work establishes the theoretical guarantee for a randomized algorithm based on *stochastic feedbacks* (**SF**) [13]. In case of availability of many independent trajectories, stabilizing feedbacks can be learned using only the last sample of each trajectory [14]. Further, reinforcement learning algorithms relying on other randomization methods such as *stochastic parameters* (**SP**) [15], and input perturbation [16, 17] are presented in the literature. The former designs the stabilizing feedback for a randomly generated dynamics parameter, while the latter adds an independent dither signal to the control command. Both procedures then learn the true dynamical parameters through observing the state sequence.

This work studies the performance of **SF** and **SP** which compared to the other aforementioned methods, are able to stabilize a larger class of linear systems. In both algorithms,  $k$  random control policies are being applied to the system for episodes of equal lengths.<sup>1</sup> Although some theoretical results exist in the literature [13, 15], practicality of these stabilization algorithms is not studied. Further, analysis of **SF** and **SP** is of independent interest because the underlying stochastic systems are *unstable* and *time-varying*. This work aims to address the above gap through considering the implementation of the algorithms. We provide numerical analysis to compare the amount of time needed for the stabilization. Further, we study the learning accuracy, the failure probability,

M.K. Shirani Faradonbeh and G. Michailidis are with the Department of Statistics, University of Florida, Gainesville, FL, 32611-5585 USA (e-mail: mfaradonbeh@ufl.edu, gmichail@ufl.edu)

Ambuj Tewari is with the Department of Statistics, University of Michigan, Ann Arbor, MI 48109-1107 USA (e-mail: tewaria@umich.edu)

<sup>1</sup> $k \geq 1 + \lceil r/p \rceil$  is required, see Section 3 for details

and the effect of both the number of episodes, and the standard deviation (of the stochastic feedbacks and the stochastic parameters).

This paper is organized as follows. In Section 2 we formulate the stabilization problem this work is addressing and discuss the theoretical backgrounds. In Section 3, two randomized algorithms to stabilize the system in finite time are discussed. Then, in Section 4 the numerical analysis of the presented algorithms are provided by depicting stabilization speeds, learning accuracies, and failure probabilities.

**Notation:** For square matrix  $M$ , the largest eigenvalue of  $M$  (in magnitude) is denoted by  $\lambda_{\max}(M)$ . For vector  $v$ ,  $\|v\|$  is the Euclidean norm, and the operator norm of matrices is denoted by  $\|M\| = \sup_{\|v\|=1} \|Mv\|$ . For the notation  $\theta$ , see Definition 1. Finally,  $\mathcal{N}(\mu, \Sigma)$  denotes the normal distribution with mean vector  $\mu$  and covariance matrix  $\Sigma$ .

## 2. THEORETICAL FRAMEWORK

To set the stage, we briefly discuss the theoretical framework of the problem. First, the system needs to be stabilizable:

**Assumption 1.** *There exists a  $r \times p$  stabilizer matrix  $L$ . That is,  $|\lambda_{\max}(A_0 + B_0L)| < 1$ .*

The noise in (1) is a sequence of centered independent random vectors;  $\mathbb{E}[\xi(t)] = 0$ , satisfying the following.

**Assumption 2.** *The noise covariance matrices are positive definite;  $|\lambda_{\min}(\mathbb{E}[\xi(t)\xi(t)'])| > 0$ . Further, the noise is sub-Gaussian:  $\sup_{t \geq 1} \mathbb{E} \left[ \exp(\beta \|\xi(t)\|^2) \right] < \infty$ , for some  $\beta > 0$ .*

Given the time length  $T$ , the objective is to design the control sequence  $\{u(t)\}_{t=0}^{T-1}$  effectively. That is, one can stabilize the system utilizing the observations  $\{x(t)\}_{t=0}^T$  acquired by applying  $\{u(t)\}_{t=0}^{T-1}$ . For this purpose, we employ a model-based approach so that the algorithm learns  $A_0, B_0$  through observing the history by date:  $\mathcal{H}_t = \left\{ \{x(i)\}_{i=0}^t, \{u(i)\}_{i=0}^{t-1} \right\}$ . The subsequent argument shows that a coarse approximation suffices for stabilization. To proceed, we define a notation:

**Definition 1.** *Henceforth, for all  $A \in \mathbb{R}^{p \times p}$ ,  $B \in \mathbb{R}^{p \times r}$ , we use  $[A, B]$  and  $\theta \in \mathbb{R}^{p \times q}$  interchangeably, where  $q \equiv p + r$ .*

It is well known that the system can be stabilized by solving a Riccati equation [18]. First, let  $Q \in \mathbb{R}^{p \times p}$ ,  $R \in \mathbb{R}^{r \times r}$  be arbitrary positive definite matrices. Then, define  $K(\theta)$  as a possible solution of the algebraic Riccati equation  $K(\theta) = Q + A'K(\theta)A - A'K(\theta)B(B'K(\theta)B + R)^{-1}B'K(\theta)A$ . Accordingly, for an arbitrary dynamics parameter  $\theta$ , define the linear feedback matrix

$$L(\theta) = -(B'K(\theta)B + R)^{-1}B'K(\theta)A. \quad (2)$$

The matrix  $K(\theta)$  exists for almost all  $\theta$  (w.r.t. Lebesgue measure [19]), and  $L(\theta)$  stabilizes the system:

**Lemma 1.** [13] *There exists a fixed  $\epsilon_0 > 0$ , such that an  $\epsilon_0$ -accurate approximation ensures the stabilization. That is,  $\|\hat{\theta} - \theta_0\| \leq \epsilon_0$  implies that  $|\lambda_{\max}(A_0 + B_0L(\hat{\theta}))| < 1$ .*

## 3. ALGORITHMS

The system can be stabilized by leveraging Lemma 1, together with the recent results on *fast* identification of unstable systems [20, 17]. In the sequel, we explain two different randomization methods, and compare their performance through numerical simulations. Note that since during the stabilization period the system is presumably unstable, the state vector can grow unbounded. In order to learn the unstable

---

### Algorithm 1 : SF

---

Input: time length  $T$ , No. of episodes  $k$ , variance  $\sigma^2$ .

**for**  $i = 1, \dots, k$  **do**

**for**  $j = 1, \dots, p$  **do**

        Draw column  $j$  of  $L_i$  from  $\mathcal{N}(0, \sigma^2 I_r)$

**end for**

**for**  $(i-1) \lfloor \frac{T}{k} \rfloor \leq t < i \lfloor \frac{T}{k} \rfloor$  **do**

        Apply  $u(t) = L_i x(t)$

**end for**

    Estimate  $\hat{D}_i$  by (3)

**end for**

    Compute  $\hat{\theta}_T$  according to (4)

---

linear dynamical systems, a sufficient and necessary condition is the closed-loop *regularity* [21]. In regular matrices, all eigenspaces corresponding to the eigenvalues of magnitude larger than one are *one dimensional*. To satisfy the regularity condition, it suffices to apply stochastic feedbacks [13]. The stabilizing algorithm proposed in the above reference utilizes a few random feedback matrices to accurately learn  $\theta_0$ .

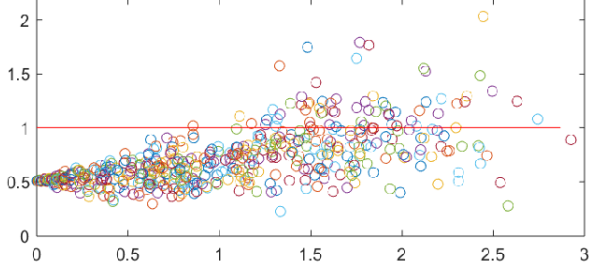
Namely, suppose that we want to stabilize the system in  $T$  time steps. The algorithm applies  $k$  statistically independent Gaussian feedback matrices  $L_1, \dots, L_k$  in episodes of the equal length  $\lfloor T/k \rfloor$ . Collecting the history  $\mathcal{H}_{k \lfloor T/k \rfloor}$ , the algorithm first estimates the closed-loop matrix  $A_0 + B_0L_i$ :

$$\hat{D}_i = \arg \min_{D \in \mathbb{R}^{p \times q}} \sum_{\ell=(i-1)\lfloor T/k \rfloor}^{i\lfloor T/k \rfloor - 1} \|x(\ell+1) - Dx(\ell)\|^2. \quad (3)$$

Then, the closed-loop estimates  $\hat{D}_1, \dots, \hat{D}_k$  are being used to estimate the true dynamics parameter  $\theta_0$  according to:

$$\hat{\theta}_T = \arg \min_{\theta \in \mathbb{R}^{p \times q}} \sum_{i=1}^k \left\| \hat{D}_i - \theta \begin{bmatrix} I_p \\ L_i \end{bmatrix} \right\|^2. \quad (4)$$

The corresponding pseudo-code is provided in Algorithm 1. To ensure that the learning procedure leads to an accurate approximation, the number of episodes  $k$  satisfies  $k \geq 1 +$



**Fig. 1:**  $\left| \lambda_{\max} \left( A_0 + B_0 L \left( \hat{\theta} \right) \right) \right|$  vs  $\left\| \hat{\theta} - \theta_0 \right\|$ .

$\lceil r/p \rceil$  [13]. The following high probability result is established for the performance of the stochastic feedback (SF) method.

**Lemma 2.** [13] Defining  $\epsilon_0$  similar to Lemma 1, let  $\hat{\theta}_T$  be the output of Algorithm 1. Then, there is  $c_0 > 0$  such that

$$\mathbb{P} \left( \left\| \hat{\theta}_T - \theta_0 \right\| > \epsilon_0 \right) \leq e^{-T c_0}.$$

Putting Lemma 1 and Lemma 2 together, we conclude that the failure probability of SF decays exponentially with  $T$ .

---

**Algorithm 2 : SP**

---

Input: time length  $T$ , No. of episodes  $k$ , variance  $\sigma^2$ .

**for**  $i = 1, \dots, k$  **do**

**for**  $j = 1, \dots, q$  **do**

    Draw column  $j$  of  $\theta_i$  from  $\mathcal{N} \left( 0, \sigma^2 I_p \right)$

**end for**

**for**  $(i-1) \lfloor \frac{T}{k} \rfloor \leq t < i \lfloor \frac{T}{k} \rfloor$  **do**

    Apply  $u(t) = L \left( \theta_i \right) x(t)$

**end for**

  Estimate  $\hat{D}_i$  by (3)

**end for**

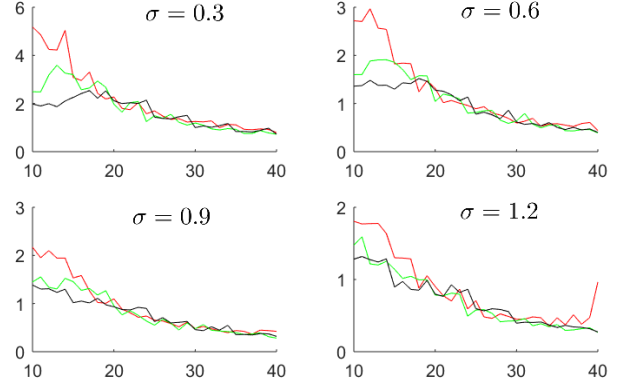
Compute  $\hat{\theta}_T$  according to (5)

---

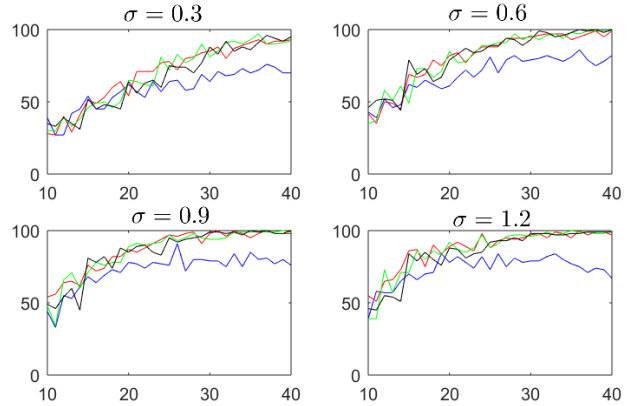
Here, we also study another randomization procedure based on stochastic parameters (SP). In this method, the algorithm draws Gaussian independent parameters  $\theta_1, \dots, \theta_k$ . Then, in each episode  $i$ , SP applies the control  $u(t) = L \left( \theta_i \right) x(t)$ , which is computed according to (2). At the end of every episode, the closed-loop matrix is estimated by (3). Finally, putting the estimates  $\hat{D}_1, \dots, \hat{D}_k$  together, SP learns  $\hat{\theta}_T$  by the minimization problem

$$\hat{\theta}_T = \arg \min_{\theta \in \mathbb{R}^{p \times q}} \sum_{i=1}^k \left\| \hat{D}_i - \theta \begin{bmatrix} I_p \\ L \left( \theta_i \right) \end{bmatrix} \right\|^2. \quad (5)$$

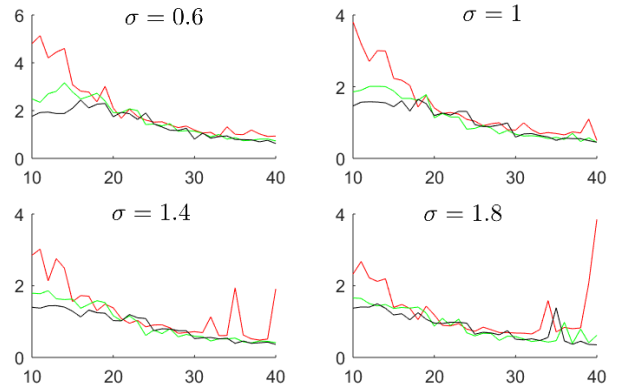
Similarly, the lower bound  $k \geq 1 + \lceil r/p \rceil$  is necessary to ensure identifiability of  $\hat{\theta}_T$ . Algorithm 2 depicts the implementation of SP.



**Fig. 2:**  $\left\| \hat{\theta}_T - \theta_0 \right\|$  vs  $T$ , where  $\hat{\theta}_T$  is given by Algorithm 1, for  $k = 3$  (red),  $k = 4$  (green), and  $k = 5$  (black).



**Fig. 3:** Percentage of stabilization by Algorithm 1 vs  $T$ , for  $k = 2$  (blue),  $k = 3$  (red),  $k = 4$  (green), and  $k = 5$  (black).



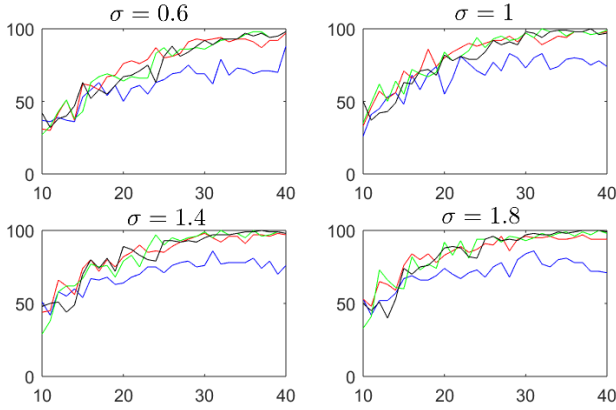
**Fig. 4:**  $\left\| \hat{\theta}_T - \theta_0 \right\|$  vs  $T$ , where  $\hat{\theta}_T$  is given by Algorithm 2, for  $k = 3$  (red),  $k = 4$  (green), and  $k = 5$  (black).

## 4. NUMERICAL ANALYSIS

In this section we simulate SF and SP for the dynamics matrices  $A_0, B_0$  in (6), and the cost matrices  $Q, R$  in (6),

$$A_0 = \begin{bmatrix} 1.07 & 0 & -0.37 \\ 0.48 & -0.88 & 0.85 \\ 0 & 0.03 & -0.92 \end{bmatrix}, \quad B_0 = \begin{bmatrix} -0.48 & 0.44 & -0.29 \\ -0.51 & 0.59 & 0.26 \\ 0.29 & 0 & -0.74 \end{bmatrix}, \quad Q = \begin{bmatrix} 1.31 & -0.17 & -0.28 \\ -0.17 & 1.14 & 0.51 \\ -0.28 & 0.51 & 5.01 \end{bmatrix}, \quad (6)$$

$$R = \begin{bmatrix} 2.01 & 0.54 & 0.77 \\ 0.54 & 1.38 & 0.42 \\ 0.77 & 0.42 & 2.38 \end{bmatrix}, \quad K(\theta_0) = \begin{bmatrix} 2.83 & 0 & -0.87 \\ 0 & 2.20 & -0.32 \\ -0.87 & -0.32 & 7.31 \end{bmatrix}, \quad L(\theta_0) = \begin{bmatrix} 0.45 & -0.19 & 0.5 \\ -0.62 & 0.35 & -0.04 \\ 0.13 & 0.06 & -0.77 \end{bmatrix}. \quad (7)$$



**Fig. 5:** Percentage of stabilization by Algorithm 2 vs  $T$ , for  $k = 2$  (blue),  $k = 3$  (red),  $k = 4$  (green), and  $k = 5$  (black).

(7). Entries of  $A_0, B_0, Q, R$  are rounded, after being randomly generated by MATLAB. For this system, solving the Riccati equation, we get  $K(\theta_0), L(\theta_0)$  as (7). Then, if one designs the feedback gain according to the true parameter  $\theta_0$ , the spectral radius of the closed-loop matrix is  $|\lambda_{\max}(A_0 + B_0 L(\theta_0))| = 0.51$ .

Fig. 1 illustrates the statement of Lemma 1. The largest eigenvalue of the closed-loop matrix is reported as a function of the error in the dynamics parameter. The scatter plot consists of different error values  $\|\hat{\theta} - \theta_0\|$ , and the resulting spectral radius  $|\lambda_{\max}(A_0 + B_0 L(\hat{\theta}))|$ .

Fig. 2, and Fig 3 plot the performance of Algorithm 1 for 100 replications, as the number of time steps  $T$  grows. The accuracy of learning the unknown dynamics parameter  $\theta_0$  is reported in Fig. 2 for different values of  $k, \sigma$ . Further, Fig. 3 graphs the number of replications in which the system is stabilized. This can be interpreted as a surrogate for the stabilization probability in Lemma 2.

Similarly, Fig. 4 and Fig. 5 illustrate the performance of Algorithm 2. The results on the learning error  $\|\hat{\theta}_T - \theta_0\|$  are depicted in Fig. 4. Then, Fig. 5 plots the percentage of the 100 replications that the linear feedback designed by the output of Algorithm 2 stabilizes the true dynamics  $\theta_0$ . Note that in both Fig. 2 and Fig. 4, the graph for  $k = 2$  is omitted since in this case the learning error is much larger than the other cases  $k = 3, 4, 5$ .

According to the above figures, as long as  $k \geq 2$ , stabilization will eventually occur. However, the case  $k = 2$  is significantly slower, and less accurate, compared to the others. All three cases  $k = 3, 4, 5$  although, exhibit similar performances. On the other hand, comparing different values of  $\sigma$ , it is clear that the magnitude of the random matrices ( $L_i$  in **SF** and  $\theta_i$  in **SP**) does not have any remarkable influence on the performance. More precisely, different values of  $\sigma$  are indifferent as long as they are not too small, or too large. If  $\sigma$  is very small, the randomization will be masked by the randomness of the noise process  $\{w(t)\}_{t=1}^{\infty}$ , and is not practical. Further, for large values of  $\sigma$ , the spectral radius of the resulting closed-loop is wildly large, and leads to an immediate explosion.

## 5. REFERENCES

- [1] P. Dorato, C. T. Abdallah, V. Cerone, and D. H. Jacobson, *Linear-quadratic control: an introduction*. Prentice Hall Englewood Cliffs, NJ, 1995.
- [2] E. Burdet, R. Osu, D. W. Franklin, T. E. Milner, and M. Kawato, “The central nervous system stabilizes unstable dynamics by learning optimal impedance,” *Nature*, vol. 414, no. 6862, p. 446, 2001.
- [3] W. Li and E. Todorov, “Iterative linear quadratic regulator design for nonlinear biological movement systems.” in *ICINCO (1)*, 2004, pp. 222–229.
- [4] M. Abeille, A. Lazaric, and X. Brokmann, “LQG for portfolio optimization,” *arXiv preprints arXiv:1611.00997*, 2016.
- [5] A. Tsiamis, K. Gatsis, and G. J. Pappas, “An information matrix approach for state secrecy,” in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 2062–2067.
- [6] L. Guo and H.-F. Chen, “The åstrom-wittenmark self-tuning regulator revisited and els-based adaptive trackers,” *IEEE Transactions on Automatic Control*, vol. 36, no. 7, pp. 802–812, 1991.
- [7] M. Ibrahimi, A. Javanmard, and B. V. Roy, “Efficient reinforcement learning for high dimensional linear

- quadratic systems,” in *Advances in Neural Information Processing Systems*, 2012, pp. 2636–2644.
- [8] M. K. S. Faradonbeh, A. Tewari, and G. Michailidis, “Optimism-based adaptive regulation of linear-quadratic systems,” *arXiv preprint arXiv:1711.07230*, 2017.
- [9] C. Li and J. Lam, “Stabilization of discrete-time non-linear uncertain systems by feedback based on ls algorithm,” *SIAM Journal on Control and Optimization*, vol. 51, no. 2, pp. 1128–1151, 2013.
- [10] L. Guo and C. Wei, “Global stability/instability of ls-based discrete-time adaptive nonlinear control,” *IFAC Proceedings Volumes*, vol. 29, no. 1, pp. 5215–5220, 1996.
- [11] S. Bittanti and M. C. Campi, “Adaptive control of linear time invariant systems: the bet on the best principle,” *Communications in Information & Systems*, vol. 6, no. 4, pp. 299–320, 2006.
- [12] Y. Abbasi-Yadkori and C. Szepesvári, “Regret bounds for the adaptive control of linear quadratic systems.” in *COLT*, 2011, pp. 1–26.
- [13] M. K. S. Faradonbeh, A. Tewari, and G. Michailidis, “Finite time adaptive stabilization of linear systems,” *IEEE Transactions on Automatic Control*, 2018.
- [14] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, “On the sample complexity of the linear quadratic regulator,” *arXiv preprint arXiv:1710.01688*, 2017.
- [15] M. K. S. Faradonbeh, A. Tewari, and G. Michailidis, “On optimality of adaptive linear-quadratic regulators,” *arXiv preprint arXiv:1806.10749*, 2018.
- [16] —, “Input perturbations for adaptive regulation and learning,” *arXiv preprint arXiv:1811.04258*, 2018.
- [17] T. Sarkar and A. Rakhlin, “How fast can linear dynamical systems be learned?” *arXiv preprint arXiv:1812.01251*, 2018.
- [18] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.
- [19] M. Abeille and A. Lazaric, “Improved regret bounds for thompson sampling in linear quadratic control problems,” in *International Conference on Machine Learning*, 2018, pp. 1–9.
- [20] M. K. S. Faradonbeh, A. Tewari, and G. Michailidis, “Finite time identification in unstable linear systems,” *Automatica*, vol. 96, pp. 342–353, 2018.
- [21] B. Nielsen, “Singular vector autoregressions with deterministic terms: Strong consistency and lag order determination,” *Discussion paper, Nuffield College*, 2009.