Online Linear Quadratic Control

Alon Cohen [1 2] **Avinatan Hassidim** [1 3] **Tomer Koren** [4] **Nevena Lazic** [4] **Yishay Mansour** [1 5] **Kunal Talwar** [4]

Abstract

We study the problem of controlling linear time-invariant systems with known noisy dynamics and adversarially chosen quadratic losses. We present the first efficient online learning algorithms in this setting that guarantee $O(\sqrt{T})$ regret under mild assumptions, where T is the time horizon. Our algorithms rely on a novel SDP relaxation for the steady-state distribution of the system. Crucially, and in contrast to previously proposed relaxations, the feasible solutions of our SDP all correspond to "strongly stable" policies that mix exponentially fast to a steady state.

1. Introduction

Linear-quadratic (LQ) control is one of the most widely studied problems in control theory (Anderson et al., 1972; Bertsekas, 1995; Zhou et al., 1996). It has been applied successfully to problems in statistics, econometrics, robotics, social science and physics. In recent years, it has also received much attention from the machine learning community, as increasingly difficult control problems have led to demand for data-driven control systems (Abbeel et al., 2007; Levine et al., 2016; Sheckells et al., 2017).

In LQ control, both the state and action are real-valued vectors. The dynamics of the environment are linear in the state and action, and are perturbed by Gaussian noise. The cost is quadratic in the state and control (action) vectors. The optimal control policy, which minimizes the cost, selects the control vector as a linear function of the state vector, and can be derived by solving the algebraic Ricatti equations.

The main focus of this work is control of linear systems whose quadratic costs vary in an unpredictable way. This problem may arise in settings such as building climate control in the presence of time-varying energy costs, due to energy auctions or unexpected demand fluctuations. To measure how well a control system adapts to time-varying costs, it is common to consider the notion of regret: the difference between the total cost of the controller, one that is only aware of previously observed costs, and that of the best fixed control policy in hindsight. This notion has been thoroughly studied in the context of online learning, and particularly in that of online convex optimization (Cesa-Bianchi & Lugosi, 2006; Hazan, 2016; Shalev-Shwartz, 2012). LQ control was considered in the context of regret by Abbasi-Yadkori et al. (2014), who give a learning algorithm for the problem of tracking an adversarially changing target in a system with noiseless linear dynamics.

In this paper we consider online learning with fixed, known, linear dynamics and adversarially chosen quadratic cost matrices. Our main results are two online algorithm that achieve $O(\sqrt{T})$ regret, when comparing to any fast mixing linear policy.[1] One of our algorithms is based on Online Gradient Descent (Zinkevich, 2003). The other is based on Follow the Lazy Leader (Kalai & Vempala, 2005), a variant of Follow the Perturbed Leader with only $O(\sqrt{T})$ expected number of policy switches.

Overall, our approach follows Even-Dar et al. (2009). We first show how to perform online learning in an "idealized setting", a hypothetical setting in which the learner can immediately observe the steady-state cost of any chosen control policy. We proceed to bound the gap between the idealized costs and the actual costs.

Our technique is conceptually different to most learning problems: instead of predicting a policy and observing its steady-state cost, the learner predicts a steady-state distribution and derives from it a corresponding policy. Importantly, this view allows us to cast the idealized problem as a semidefinite program which minimizes the expected costs as a function of a steady state distribution (of both states and controls). As the problem is now convex, we apply OGD and FLL to the SDP and argue about fast-mixing properties of its feasible solutions.

[1]Google Research, Tel Aviv [2]Technion—Israel Inst. of Technology [3]Bar Ilan University [4]Google Brain, Mountain View [5]Tel Aviv University. Correspondence to: Alon Cohen <alon-cohen@google.com>, Tomer Koren <tkoren@google.com>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

[1] Technically, we define the class of "strongly stable" policies that guarantee the desired fast mixing property. Conceptually, slowly mixing policies are less attractive for implementation, given their inherent gap between their long and short term cost.

For online gradient descent, we define a "sequential strong stability" property that couples consecutive control matrices, and show that it guarantees that the observed state distributions closely track those generated in the idealized setting. We then show that the sequence of policies generated by the online gradient descent algorithm satisfies this property.

In Follow the Lazy Leader, following each switch our algorithm resets the system—a process that takes a constant number of rounds, after which the cost of playing the new policy is less than its steady-state cost.

The holy grail of reinforcement learning is controlling a dynamical stochastic system under uncertainty, and clearly both MDPs and LQ control are well within this mission statement. There are obvious differences between the two models: MDPs model discrete state and action dynamics while LQ control addresses continuous linear dynamics with a quadratic cost. In this work we are inspired by methodologies from online-MDP and regret minimization to derive new results for LQ control. We believe that exploring the interface between the two will be fruitful for both sides, and holds significant potential for future RL research agenda.

1.1. Related Work

LQ control can be seen as a continuous analogue of the discrete Markov Decision Process (MDP) model. As such, our results are conceptually similar to those of Even-Dar et al. (2009), who derive regret bounds for MDPs with known dynamics and changing rewards. However, our technical approach and the derivation of our algorithms are very different than those applicable in context of MDPs.

Among the many follow-up works to Even-Dar et al. (2009), let us note Yu et al. (2009) and Abbasi et al. (2013) that propose lazy algorithms similar to our second algorithm. We remark that, compared to our $O(\sqrt{T})$ regret bounds, Abbasi-Yadkori et al. (2014) give an $O(\log^2 T)$ regret bound under much stronger assumptions.[2] Similar bounds are established by Neu & Gómez (2017) for online learning in linearly solvable MDPs, that were shown to capture appropriately discretized versions of LQ control systems (Todorov, 2009). In light of these results, it is interesting to investigate whether our bounds are tight or can actually be improved. We leave this investigation for future work.

An orthogonal line of research that has gained popularity in recent years is controlling linear quadratic systems with unknown fixed dynamics. The majority of recent papers deal with off-policy learning: either by policy gradient (Fazel et al., 2018); by estimating the transition matrices (Dean et al., 2017); or by improper learning (Hazan et al., 2017; Arora et al., 2018). In contrast to that, Abbasi-Yadkori & Szepesvári (2011) and Ibrahimi et al. (2012) present an on-policy learning algorithm with $O(\sqrt{T})$ regret.

Semidefinite programming for LQ control has been used in the past (Balakrishnan & Vandenberghe, 2003; Dvijotham et al., 2013; Lee & Hu, 2016), mostly in the context of infinite-horizon constrained LQRs (Lee & Khargonekar, 2007; Schildbach et al., 2015). In many of these formulations, one has to solve the SDP exactly to obtain a stabilizing solution; in other words, only the optimal policy is known to be stable and suboptimal policies need not be stabilizing. This is not the case in our SDP formulation, as any feasible solution is not only stable but, in fact, strongly-stable (see the formal definition in Section 3).

2. Background

2.1. Linear Quadratic Control

The standard linear quadratic (Gaussian) control problem is as follows. Let $x_t \in \mathbb{R}^d$ be the system state at time t and let $u_t \in \mathbb{R}^k$ be the control (action) taken at time t. The system transitions to the next state using linear time-invariant dynamics

$$x_{t+1} = Ax_t + Bu_t + w_t \,,$$

where w_t are i.i.d. Gaussian noise vectors with zero mean and covariance $W \succeq 0$. The cost incurred at each time point is a quadratic function of the state and control, $x_t^\intercal Q x_t + u_t^\intercal R u_t$, for positive definite matrices Q and R.

A policy is a mapping $\pi : \mathbb{R}^d \mapsto \mathbb{R}^k$ from the current state x_t to a control (i.e., an action) u_t. The cost of a policy after T time steps is

$$J_T(\pi) = \mathbb{E}\Big[\sum_{t=1}^{T} x_t^\intercal Q x_t + u_t^\intercal R u_t \Big],$$

where $u_1, \ldots, u_T$ are chosen according to π; the expectation is w.r.t. the randomness in the state transitions and (possibly) the policy. In the infinite-horizon version of the problem, the goal is to minimize the steady-state cost $J(\pi) = \lim_{T\to\infty}(1/T)J_T(\pi)$.

In the infinite-horizon setting and when the system is controllable,[3] it is well-known that the optimal policy is given by constant linear feedback $u_t = Kx_t$. For the optimal K, the dynamics are given by $x_{t+1} = (A + BK)x_t + w_t$, and K is guaranteed to be stable; a policy K is called stable if $\rho(A + BK) < 1$, where for a matrix M, $\rho(M)$ is the spectral radius of M. In this case, x_t converges to a steady-state (stationary) distribution, i.e., x_t has the

[2] Not only their setting assumes that $Q_t = Q$ and $R_t = I$ for all t for a fixed and known matrix $Q \succeq 0$, they also make non-trivial norm assumptions on the corresponding optimal control matrix $K^\star$.

[3] The system is controllable if the matrix $(B\ AB\ \cdots\ A^{d-1}B)$ has full column-rank. Under the controllability assumption, any state can be reached in at most d steps (ignoring noise).

same distribution as $(A + BK)x_t + w_t$. This implies that $\mathbb{E}[x_t] = 0$, and the covariance matrix $X = \mathbb{E}[x_t x_t^\mathsf{T}]$ satisfies $X = (A + BK)X(A + BK)^\mathsf{T} + W$.

The steady-state cost of a stable policy K with steady-state covariance X is given by $J(K) = (Q + K^\mathsf{T}RK) \bullet X$. Here $\bullet$ denotes element-wise inner product, i.e., $A \bullet B = \mathrm{Tr}(A^\mathsf{T}B)$.

2.2. Problem Setting

We consider an online setting, where a sequence of positive definite cost matrices $Q_1, \ldots, Q_T$, $R_1, \ldots, R_T$ is chosen by the environment ahead of time and unknown to the learner. We assume throughout that $\mathrm{Tr}(Q_t), \mathrm{Tr}(R_t) \le C$ for all t, for some constant $C > 0$. We assume that the dynamics (A, B) are time-invariant and known, and that the system is initialized at $x_0 = 0$. At each time step t, the learner observes the state x_t, chooses an action u_t, and suffers cost $x_t^\mathsf{T}Q_t x_t + u_t^\mathsf{T}R_t u_t$. Thereafter, the system transitions to the next state.

A (randomized) learning algorithm $\mathcal{A}$ is a mapping from x_t and the previous cost matrices $Q_0, ..., Q_{t-1}$ and $R_0, \ldots, R_{t-1}$ to a distribution over a control u_t. We define the cost of an algorithm as $J_T(\mathcal{A}) = \mathbb{E}[\sum_{t=1}^T x_t^\mathsf{T}Q_t x_t + u_t^\mathsf{T}R_t u_t]$, where $u_1, \ldots, u_T$ are chosen at random according to $\mathcal{A}$.

The goal of the learner is to minimize the regret, defined as:

$$R_T(\mathcal{A}) = J_T(\mathcal{A}) - \min_{\pi \in \Pi} J_T(\pi)\,,$$

where Π is a set of benchmark policies. In the sequel, we fix Π to be the set of all strongly stable policies; we defer the formal definition of this class of policies to Section 3 below.

3. Strong Stability

In this section we formalize the notion of a strongly stable policy and discuss some of its properties. Intuitively, a strongly stable policy is a policy that exhibits fast mixing and converges quickly to a steady-state distribution. Note that, while stable policies K (for which $\rho(A + BK) < 1$) necessarily converge to a steady-state, nothing is guaranteed regarding their rate of convergence. The following definition helps remedy that.

Definition 3.1 (Strong Stability)**.** A policy K is (κ, γ)-*strongly stable* (for $\kappa > 0$ and $0 < \gamma \le 1$) if $\|K\| \le \kappa$, and there exists matrices L and H such that $A + BK = HLH^{-1}$, with $\|L\| \le 1 - \gamma$ and $\|H\|\|H^{-1}\| \le \kappa$.

Strong-stability is a quantitative version of stability, in the sense that any stable policy is strongly-stable for some κ and γ (See Lemma B.1 in the supplementary material). Conversely, strong-stability implies stability: if K is strongly-stable then $A + BK$ is similar to a matrix L with $\|L\| < 1$, and so $\rho(A + BK) = \rho(L) \le \|L\| < 1$, i.e., K is stable. Notice that for a strongly stable K it may not be the case that $\|A + BK\| < 1$, and a non-trivial transformation $H \ne I$ may be required to make the norm smaller than one (this is indeed the case with feasible solutions to our SDP relaxation).

Strong stability ensures exponentially fast convergence to steady-state, as is made precise in the next lemma.

Lemma 3.2. For all $t = 1, 2, \ldots$ let $\widehat{X}_t$ be the state covariance matrix on round t starting from some $\widehat{X}_0 \ge 0$ and following a (κ, γ)-strongly stable policy $\pi(x) = Kx$. Then $\widehat{X}_1, \widehat{X}_2, \ldots$ approaches a steady-state covariance matrix X, and further, for all t it holds that

$$\|\widehat{X}_t - X\| \le \kappa^2 e^{-2\gamma t}\|\widehat{X}_0 - X\|.$$

This exponential convergence is true even if the policy is randomized and follows K in expectation; that is, if $\mathbb{E}[\pi(x)|x] = Kx$, and provided that $\mathrm{Cov}[\pi(x)|x]$ is finite.

Proof. Let us first analyze deterministic policies. As noted above, we know that K is stable and as a result the state covariances $\widehat{X}_t$ approach a steady-state covariance X. By definition, we have

$$\begin{aligned} \widehat{X}_{t+1} &= (A + BK)\widehat{X}_t(A + BK)^\mathsf{T} + W \qquad \forall\, t \ge 0; \\ X &= (A + BK)X(A + BK)^\mathsf{T} + W. \end{aligned}$$

Subtracting the equations and recursing, we have $\widehat{X}_t - X = (A + BK)^t(\widehat{X}_0 - X)((A + BK)^t)^\mathsf{T}$, which gives

$$\|\widehat{X}_t - X\| \le \|(A + BK)^t\|^2\|\widehat{X}_0 - X\|.$$

For further bounding the right-hand side, observe that $(A + BK)^t = HL^tH^{-1}$, thus

$$\|(A + BK)^t\| \le \|H\|\|H^{-1}\|\|L\|^t \le \kappa(1 - \gamma)^t \le \kappa e^{-\gamma t}.$$

Combining the inequalities gives the result for deterministic policies.

For randomized policies with $\mathbb{E}[u|x] = Kx$ and finite $V = \mathrm{Cov}[u|x]$, the dynamics of the state covariance take the form

$$\begin{aligned} \widehat{X}_{t+1} &= (A + BK)\widehat{X}_t(A + BK)^\mathsf{T} + BVB^\mathsf{T} + W \quad \forall\, t \ge 0; \\ X &= (A + BK)X(A + BK)^\mathsf{T} + BVB^\mathsf{T} + W. \end{aligned}$$

Since the analysis above only depends on the difference between the equations, the added BVB^T term has no effect on the convergence of X_t. Note, however, that the steady state X itself will be a function of V in general. □

Let us state one more property of strongly stable policies that will be useful in our analysis.

Lemma 3.3. Assume that K is (κ, γ)-strongly stable, and let X and U be the covariances of x and u at steady-state when following K. Then $\mathrm{Tr}(X) \le (\kappa^2/\gamma)\,\mathrm{Tr}(W)$ and $\mathrm{Tr}(U) \le (\kappa^4/\gamma)\,\mathrm{Tr}(W)$.

3.1. Sequential strong stability

We next present a stronger notion of strong stability which plays a central role in our analysis. Roughly speaking, the goal is to argue about fast mixing when following a sequence of different policies $K_1, K_2, \ldots$ (rather than a fixed policy K throughout). In this case, for any kind of mixing to take place, not only does one has to require that each policy is strongly stable, but also that the sequence is "slowly changing." This motivates the following definition.

Definition 3.4 (sequential strong stability)**.** A sequence of policies $K_1, \ldots, K_T$ is (κ, γ)-*strongly stable* (for $\kappa > 0$ and $0 < \gamma \leq 1$) if there exist matrices $H_1, \ldots, H_T$ and $L_1, \ldots, L_T$ such that $A + BK_t = H_t L_t H_t^{-1}$ for all t, with the following properties:

(i) $\|L_t\| \leq 1 - \gamma$ and $\|K_t\| \leq \kappa$;
(ii) $\|H_t\| \leq \beta$ and $\|H_t^{-1}\| \leq 1/\alpha$ with $\kappa = \beta/\alpha$;
(iii) $\|H_{t+1}^{-1} H_t\| \leq 1 + \gamma/2$.

Strongly stable sequences mix quickly, in the following sense (proof is deferred to the full version of the paper).

Lemma 3.5. Let $\pi_t(x) = K_t x$ $(t = 1, 2, \ldots)$ be a sequence of policies with respective steady-state covariance matrices $X_1, X_2, \ldots$, such that $K_1, K_2, \ldots$ is a (κ, γ)-strongly stable sequence and $\|X_t - X_{t-1}\| \leq \eta$ for all t, for some $\eta > 0$. Let $\widehat{X}_t$ be the state covariance matrix on round t starting from some $\widehat{X}_1 \succeq 0$ and following this sequence. Then

$$\|\widehat{X}_{t+1} - X_{t+1}\| \leq \kappa^2 e^{-\gamma t} \|\widehat{X}_1 - X_1\| + \frac{2\eta\kappa^2}{\gamma}.$$

The same is true even if the policies are randomized, such that $\mathbb{E}[\pi_t(x)|x] = K_t x$ and $\mathrm{Cov}[\pi_t(x)|x]$ exists and is finite.

4. SDP Relaxation for LQ control

We now present our SDP relaxation for the infinite-horizon LQ control problem. Our presentation requires the following definitions. Consider an LQ control problem parameterized by matrices A, B, Q, R and W. For any stable policy (for which a steady-state distribution exists), define

$$\mathcal{E}(\pi) = \mathbb{E}\begin{pmatrix} xx^\mathsf{T} & xu^\mathsf{T} \\ ux^\mathsf{T} & uu^\mathsf{T} \end{pmatrix}, \tag{1}$$

where x is distributed according to the steady-state distribution of π, and $u = \pi(x)$. Then, the infinite horizon cost of π is given by $J(\pi) = \left(\begin{smallmatrix} Q & 0 \\ 0 & R \end{smallmatrix}\right) \bullet \mathcal{E}(\pi)$. For a policy $\pi_K(x) = Kx$ defined by a stable control matrix K (i.e., for which $\rho(A + BK) < 1$), this matrix takes the form

$$\mathcal{E}(K) = \begin{pmatrix} X & XK^\mathsf{T} \\ KX & KXK^\mathsf{T} \end{pmatrix}, \tag{2}$$

where X is the state covariance at steady-state. (We slightly abuse notation and write $\mathcal{E}(K)$ instead of $\mathcal{E}(\pi_K)$). In this case, one also has $J(K) = J(\mathcal{E}(K)) = (Q + K^\mathsf{T} R K) \bullet X$.

4.1. The relaxation

We can now present our SDP relaxation for the LQ control problem given by (A, B, Q, R, W), which takes the form:

$$\begin{aligned} \text{minimize} \quad & J(\Sigma) = \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} \bullet \Sigma \\ \text{subject to} \quad & \Sigma_{xx} = \begin{pmatrix} A & B \end{pmatrix} \Sigma \begin{pmatrix} A & B \end{pmatrix}^\mathsf{T} + W, \\ & \Sigma \succeq 0, \quad \mathrm{Tr}(\Sigma) \leq \nu. \end{aligned} \tag{3}$$

Here, $\nu > 0$ is a parameter whose value will be determined later, and Σ is a $(d + k) \times (d + k)$ symmetric matrix that decomposes to blocks as follows:

$$\Sigma = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xu} \\ \Sigma_{xu}^\mathsf{T} & \Sigma_{uu} \end{pmatrix},$$

where Σ_{xx} is a $d \times d$ block, Σ_{uu} is $k \times k$, and Σ_{xu} is $d \times k$.

The program Eq. (3) is a relaxation in the following sense.

Lemma 4.1. For any stable policy π such that at steady-state $\mathbb{E}\|x\|^2 + \mathbb{E}\|u\|^2 \leq \nu$, the matrix $\Sigma = \mathcal{E}(\pi)$ is feasible for (3).

Proof. Let π be any stable policy and consider the matrix $\Sigma = \mathcal{E}(\pi)$. Then $\Sigma \succeq 0$ (by definition, recall Eq. (1)), and satisfies the equality constraint of (3), since if x is at steady-state and $u = \pi(x)$, then $Ax + Bu + w$ has the same distribution as x for $w \sim \mathcal{N}(0, W)$ independent of x and u, thus $\mathbb{E}[xx^\mathsf{T}] = \mathbb{E}[(Ax + Bu + w)(Ax + Bu + w)^\mathsf{T}]$; the latter is equivalent to $\Sigma_{xx} = (A \quad B)\Sigma(A \quad B)^\mathsf{T} + W$. Finally, observe that $\mathrm{Tr}(\Sigma) = \mathbb{E}\,\mathrm{Tr}(xx^\mathsf{T}) + \mathbb{E}\,\mathrm{Tr}(uu^\mathsf{T}) = \mathbb{E}\|x\|^2 + \mathbb{E}\|u\|^2$ where x, u are distributed according to the steady-state distribution of π, hence Σ satisfies the trace constraint. □

4.2. Extracting a policy

We next show that from any feasible solution to the SDP, one can extract a stable policy with the same (if not better) cost, provided that $W \succ 0$. For any feasible solution Σ for the SDP, define a control matrix as follows:

$$\mathcal{K}(\Sigma) = \Sigma_{xu}^\mathsf{T} \Sigma_{xx}^{-1}. \tag{4}$$

Note that, due to the equality constraint of the SDP, our assumption $W \succ 0$ ensures that $\Sigma_{xx} \succ 0$, thus Σ_{xx} is nonsingular and $\mathcal{K}(\Sigma)$ is well defined.

Theorem 4.2. Let Σ be any feasible solution to the SDP, and let $K = \mathcal{K}(\Sigma)$. Then the policy $\pi(x) = Kx$ is stable, and it holds that $\mathcal{E}(K) \preceq \Sigma$. In particular, $\mathcal{E}(K)$ is also feasible for the SDP and its cost is at most that of Σ.

Without the trace constraint, the theorem particularly implies that for the optimal solution $\Sigma^\star$ of the SDP, the corresponding control matrix $K^\star = \mathcal{K}(\Sigma^\star)$ is an optimal policy for the original problem, recovering a classic result in control theory.

Proof of Theorem 4.2. Our first step is to show that

$$\Sigma \succeq \Sigma' = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xx}K^\mathsf{T} \\ K\Sigma_{xx} & K\Sigma_{xx}K^\mathsf{T} \end{pmatrix}. \tag{5}$$

To see this, observe that by definition of $K = \mathcal{K}(\Sigma)$ we have

$$\Sigma = \Sigma' + \begin{pmatrix} 0 & 0 \\ 0 & \Sigma_{uu} - \Sigma_{ux}^\mathsf{T}\Sigma_{xx}^{-1}\Sigma_{ux} \end{pmatrix}.$$

Thus, it suffices to show that $\Sigma_{uu} - \Sigma_{ux}^\mathsf{T}\Sigma_{xx}^{-1}\Sigma_{ux}$ is PSD. The latter matrix is the Schur complement of Σ, and is PSD because Σ is PSD.

Next, we show that the control matrix K gives rise to a stable policy. Let us develop Eq. (3). First, since $W \succ 0$ we also have that $\Sigma_{xx} \succ 0$. Moreover, by Eq. (5),

$$\begin{aligned} \Sigma_{xx} &= (A \quad B)\Sigma(A \quad B)^\mathsf{T} + W \\ &\succeq (A + BK)\Sigma_{xx}(A + BK)^\mathsf{T} + W \\ &\succ (A + BK)\Sigma_{xx}(A + BK)^\mathsf{T}. \end{aligned}$$

Let λ and v be a (possibly complex) eigenvalue and left-eigenvector associated with $A + BK$. Then,

$$v^*\Sigma_{xx}v > v^*(A + BK)\Sigma_{xx}(A + BK)^\mathsf{T}v = |\lambda|^2 v^*\Sigma_{xx}v,$$

which, by $v^*\Sigma_{xx}v > 0$, implies $|\lambda| < 1$. This is true for all eigenvalues λ, and shows that $\rho(A + BK) < 1$, that is, K is stable.

Finally, let us show that $\mathcal{E}(K) \preceq \Sigma'$, which together with Eq. (5) would imply our claim $\mathcal{E}(K) \preceq \Sigma$. Denote by X the state covariance at steady-state when following K; then,

$$\mathcal{E}(K) = \begin{pmatrix} X & XK^\mathsf{T} \\ KX & KXK^\mathsf{T} \end{pmatrix}.$$

To establish that $\mathcal{E}(K) \preceq \Sigma'$ it is enough to show $X \preceq \Sigma_{xx}$. To this end, let $\Delta = \Sigma_{xx} - X$ and write

$$\begin{aligned} X + \Delta &\succeq (A + BK)X(A + BK)^\mathsf{T} + W \\ &\quad + (A + BK)\Delta(A + BK)^\mathsf{T} \\ &= X + (A + BK)\Delta(A + BK)^\mathsf{T}, \end{aligned}$$

from which we get $\Delta \succeq (A + BK)\Delta(A + BK)^\mathsf{T}$. Applying the latter inequality recursively, we obtain

$$\Delta \succeq (A + BK)^n\Delta((A + BK)^\mathsf{T})^n.$$

Recall that $\rho(A + BK) < 1$; thus, taking the limit as $n \to \infty$, we get $(A + BK)^n\Delta((A + BK)^\mathsf{T})^n \to 0$, which implies $\Delta \succeq 0$. This shows that $X \preceq \Sigma_{xx}$, as required.

To complete the proof observe that $\mathcal{E}(K)$ is feasible for the SDP since $\mathcal{E}(K) \preceq \Sigma$ and Σ is feasible. Furthermore, since $\left(\begin{smallmatrix} Q & 0 \\ 0 & R \end{smallmatrix}\right)$ is PSD, we have

$$J(\mathcal{E}(K)) = \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} \bullet \mathcal{E}(K) \le \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} \bullet \Sigma = J(\Sigma). \quad \square$$

4.3. Strong stability of solutions

Let us show that from a solution to the SDP one can extract a strongly stable policy.

Lemma 4.3. Assume that $W \succeq \sigma^2 I$ and let $\kappa = \sqrt{\nu}/\sigma$. Then for any feasible solution Σ for the SDP, the policy $K = \mathcal{K}(\Sigma)$ is $(\kappa, 1/2\kappa^2)$-strongly stable.

Proof. According to Theorem 4.2, the policy K is (weakly) stable and the matrix $\widehat{\Sigma} = \mathcal{E}(K)$ is feasible for the SDP. Let $X = \widehat{\Sigma}_{xx}$ be the state covariance of K at steady-state. Since $\widehat{\Sigma}$ is feasible, and since $W \succeq \sigma^2 I$, we have

$$X \succeq (A + BK)X(A + BK)^\mathsf{T} + \sigma^2 I. \tag{6}$$

In particular, this means that $X \succeq \sigma^2 I$. On the other hand, we have $\mathrm{Tr}(X) \le \mathrm{Tr}(\widehat{\Sigma}) \le \nu$, thus $X \preceq \nu I$. Overall,

$$\sigma^2 I \preceq X \preceq \nu I. \tag{7}$$

Given that X is nonsingular, we can define $L = X^{-1/2}(A + BK)X^{1/2}$. Multiplying Eq. (6) by $X^{-1/2}$ from both sides, we obtain $I \succeq LL^\mathsf{T} + \sigma^2 X^{-1} \succeq LL^\mathsf{T} + \kappa^{-2} I$. Thus $LL^\mathsf{T} \preceq (1 - \kappa^{-2})I$, so $\|L\| \le \sqrt{1 - \kappa^{-2}} \le 1 - \kappa^{-2}/2$. Also, Eq. (7) shows that $\|X^{1/2}\|\|X^{-1/2}\| \le \kappa$. It is left to establish the bound on the norm $\|K\|_\mathsf{F}$. To this end, use the fact that

$$X \bullet KK^\mathsf{T} = \mathrm{Tr}(KXK^\mathsf{T}) = \mathrm{Tr}(\widehat{\Sigma}_{uu}) \le \nu$$

together with $X \succeq \sigma^2 I$ (recall Eq. (7)) to obtain $\sigma^2\|K\|_\mathsf{F}^2 \le \nu$, that is, $\|K\|_\mathsf{F} \le \kappa$. $\square$

We can also prove an analogous statement for sequences of feasible solutions, provided that they change slowly enough (we defer the proof to the full version of the paper).

Lemma 4.4. Assume that $W \succeq \sigma^2 I$ and let $\kappa = \sqrt{\nu}/\sigma$. Let $\Sigma_1, \Sigma_2, \ldots$ be a sequence of feasible solutions of (3), and suppose that $\|\Sigma_{t+1} - \Sigma_t\| \le \eta$ for all t for some $\eta \le \sigma^2/\kappa^2$. Then the sequence $K_1, K_2, \ldots$, where $K_t = \mathcal{K}(\Sigma_t)$ for all t is $(\kappa, 1/2\kappa^2)$-strongly stable.

5. Online LQ Control

In this section we describe our gradient based algorithm for online LQ control, presented in Algorithm 1. The algorithm maintains an "ideal" steady-state covariance matrix Σ_t by performing online gradients steps directly on the SDP we formulated in Section 4 (with the linear cost functions changing from round to round). Then, a control matrix K_t is extracted from the covariance Σ_t and is used to generate a prediction.

Notice that the predictions made by the algorithm are randomly drawn from the Gaussian $\mathcal{N}(K_t x_t, V_t)$, and only follow the extracted policies $K_1, K_2, \ldots$ in expectation. This randomization step is crucial for the algorithm to exhibit fast

Algorithm 1 Online LQ Controller

Parameter: $\eta, \nu > 0$
Initialize $\Sigma_1 = I_{n\times n}$ with $n = d + k$
for $t = 1, 2, \ldots$ **do**
Receive state x_t
Compute $K_t = (\Sigma_t)_{ux}(\Sigma_t)_{xx}^{-1}$, $V_t = (\Sigma_t)_{uu} - K_t(\Sigma_t)_{xx}K_t^\intercal$
Predict $u_t \sim \mathcal{N}(K_t x_t, V_t)$; receive Q_t, R_t
Update:

$$\Sigma_{t+1} = \Pi_{\mathcal{S}}\left[\Sigma_t - \eta\left(\begin{smallmatrix} Q_t & 0 \\ 0 & R_t \end{smallmatrix}\right)\right],$$

where $\Pi_{\mathcal{S}}$ is the Frobenius-norm projection onto

$$\mathcal{S} = \left\{ \Sigma \in \mathbb{R}^{n\times n} \;\middle|\; \begin{array}{l} \Sigma \succeq 0, \quad \mathrm{Tr}(\Sigma) \le \nu, \\ \Sigma_{xx} = \begin{pmatrix} A & B \end{pmatrix} \Sigma \begin{pmatrix} A & B \end{pmatrix}^\intercal + W \end{array} \right\}$$

end for

mixing: sampling the prediction from a distribution with the right covariance ensures the observed covariance matrices converge to those generated by the algorithm, and consequently this sequence "mixes" more quickly.

For Algorithm 1 we prove the following guarantee.

Theorem 5.1. Assume that $\mathrm{Tr}(W) \le \lambda^2$ and $W \succeq \sigma^2 I$. Given $\kappa > 0$ and $0 \le \gamma < 1$, set $\nu = 2\kappa^4\lambda^2/\gamma$ and $\eta = \sigma^3/(2C\sqrt{\nu T})$. The expected regret of Algorithm 1 compared to any (κ, γ)-strongly stable control matrix $K^\star$ is at most

$$J_T(A) - J_T(K^\star) = O\left(\frac{\kappa^{10}\lambda^5}{\gamma^{2.5}\sigma^3} C\sqrt{T}\right),$$

provided that $T \ge 8\kappa^4\lambda^2/(\gamma\sigma^2)$.

We remark that the theorem (in fact, Algorithm 1 itself) tacitly assumes that the SDP defined by $\mathcal{S}$ is feasible; otherwise, the set of strongly-stable policies is empty and the statement of Theorem 5.1 is vacuous.

Proof. Fix an arbitrary (κ, γ)-strongly stable control matrix $K^\star$, and denote by $\widehat{\Sigma}_1^\star, \ldots, \widehat{\Sigma}_T^\star$ be the covariances induced by using $K^\star$ throughout. Also, let $\widehat{\Sigma}_1, \ldots, \widehat{\Sigma}_T$ be the actual observed covariance matrices induced by the algorithm. Denoting $L_t = \left(\begin{smallmatrix} Q_t & 0 \\ 0 & R_t \end{smallmatrix}\right)$, the expected regret of the algorithm can be then written as follows:

$$\begin{aligned} \sum_{t=1}^T L_t \bullet (\widehat{\Sigma}_t - \widehat{\Sigma}_t^\star) &= \sum_{t=1}^T L_t \bullet (\widehat{\Sigma}_t - \Sigma_t) \\ &\quad + \sum_{t=1}^T L_t \bullet (\Sigma_t - \Sigma^\star) \\ &\quad + \sum_{t=1}^T L_t \bullet (\Sigma^\star - \widehat{\Sigma}_t^\star). \end{aligned} \tag{8}$$

Observe that the sequence $\Sigma_1, \ldots, \Sigma_T$ generated by the algorithm is feasible for the (feasibility) SDP described by the set $\mathcal{S}$. Thanks to Lemma 4.3, for any feasible $\Sigma \in \mathcal{S}$ the corresponding control matrix $\mathcal{K}(\Sigma)$ is $(\bar{\kappa}, \bar{\gamma})$-strongly stable, for $\bar{\kappa} = \sqrt{\nu}/\sigma$ and $\bar{\gamma} = \sigma^2/2\nu$; in particular, this applies to each of the matrices Σ_t.

We proceed by bounding each of the sums on the right-hand side of Eq. (8). We start with the second term and use a well-known regret bound for the Online Gradient Descent algorithm, due to Zinkevich (2003).

Lemma 5.2. We have

$$\sum_{t=1}^T L_t \bullet (\Sigma_t - \Sigma^\star) \le \frac{4\nu^2}{\eta} + 4C^2\eta T.$$

Additionally, the Σ_t are slowly changing in the sense that, for all t,

$$\|\Sigma_{t+1} - \Sigma_t\|_{\mathrm{F}} \le 4C\eta. \tag{9}$$

We next bound the first term, now relying on Eq. (9) and the fact that the sequence of (randomized) policies chosen by Algorithm 1 is strongly stable.

Lemma 5.3. If $\eta \le \sigma^2/4C\bar{\kappa}^2$, it holds that

$$\sum_{t=1}^T L_t \bullet (\widehat{\Sigma}_t - \Sigma_t) \le \frac{16C^2\bar{\kappa}^4}{\bar{\gamma}}\eta T + \frac{4C\bar{\kappa}^4}{\bar{\gamma}}\nu.$$

Finally, the last term in Eq. (8) can be bounded using the strong stability of $K^\star$.

Lemma 5.4. For any (κ, γ)-strongly stable $K^\star$,

$$\sum_{t=1}^T L_t \bullet (\Sigma^\star - \widehat{\Sigma}_t^\star) \le 2C\frac{\kappa^4\nu}{\gamma}.$$

The theorem now follows by plugging in the bounds we established in Lemmas 5.2 to 5.4 into Eq. (8) and setting our choices of η and ν. (See the full version of the paper for details.) □

6. Oracle-based Algorithm

In this section we present a different approach that is based on Follow the Lazy Leader (Kalai & Vempala, 2005). In contrast to Algorithm 1, this approach does not require a lower bound on the noise but rather relies on occasionally performing resets, and needs a bound on the cost of this reset (this is established in the full version of the paper under reasonable assumptions). We assume access to an Oracle procedure that receives cost matrices Q, R, and parameter $\nu > 0$. It returns a control matrix K that minimizes the steady-state cost, subject to $\mathrm{Tr}(X) + \mathrm{Tr}(KXK^\intercal) \le \nu$, where X is the steady-state covariance matrix associated with K.[4]

[4] Oracle can be implemented by solving the SDP in Section 4.

Algorithm 2 Follow the Lazy Leader

Parameter: $\eta, \nu > 0$, transition matrices A, B, distribution μ.
Sample $Q_1^p \in \mathbb{R}^{d\times d}, R_1^p \in \mathbb{R}^{k\times k}$ from $d\mu$.
Set $\widehat{Q}_1 \leftarrow 0$, $\widehat{R}_1 \leftarrow 0$
for $t = 1, 2, \ldots$ **do**
  Receive state x_t.
  Compute $K_t \leftarrow \textsc{Oracle}(\widehat{Q}_t + Q_t^p, \widehat{R}_t + Q_t^p, \nu)$.
  Predict $u_t \leftarrow K_t x_t$.
  Receive Q_t, R_t.
  Update $\widehat{Q}_{t+1} = \widehat{Q}_t + Q_t$, $\widehat{R}_{t+1} = \widehat{R}_t + R_t$.
  With probability $\min\left\{1, \frac{d\mu(Q_t^p - Q_t, R_t^p - R_t)}{d\mu(Q_t^p, R_t^p)}\right\}$, set
    $Q_{t+1}^p \leftarrow Q_t^p - Q_t$.
    $R_{t+1}^p \leftarrow R_t^p - R_t$,
  else, perform reset and set
    $Q_{t+1}^p \leftarrow -Q_t^p$.
    $R_{t+1}^p \leftarrow -R_t^p$.
end for

Algorithm 2 is similar to Follow the Perturbed Leader, and in fact behaves the same in expectation. At every round t, Oracle is called using the sum of previously seen Qs and Rs plus an additional random noise, Q_t^p and R_t^p. Oracle returns a matrix K_t that is used to choose $u_t = K_t x_t$.

For the measure $d\mu$, we use the joint measure over symmetric matrices Q and R, whose upper triangle is sampled coordinate-wise i.i.d from Laplace$(1/\eta)$. The "lazyness" of the algorithm stems from $Q_1^p, \ldots, Q_T^p$ and $R_1^p, \ldots, R_T^p$ being sampled dependently over time such that the cumulative perturbed loss only changes with small probability between rounds. Consequently, the expected number of switches of K as well as the expected number of resets are only $O(\eta T)$.

The reset step in the algorithm, informally, drives the system to zero at some cost. Here we assume that B has full column-rank in which case we can reset in one step. In the full version of the paper, we show how resetting can be done over a sequence of steps under much weaker assumptions.

Observation 6.1. Suppose that B has full column-rank. Resetting the system in round t can be done by setting $u_t = -B^\dagger A x_t$, such that at the next round $x_{t+1} = w_{t+1}$. Moreover, the expected cost of the reset is at most $C\nu(1 + \|B^\dagger A\|^2)$.

For Algorithm 2 we will show the following regret bound.

Theorem 6.2. Assume that $\mathrm{Tr}(W) \le \lambda^2$, and suppose that the cost of a reset is at most C_r. Then for $\nu = 2\kappa^4\lambda^2/\gamma$, the expected regret of Algorithm 2 against any (κ, γ)-strongly-stable control matrix $K^\star$ satisfies

$$\mathbb{E}\left[J_T(A) - J_T(K^\star)\right] = O\left((d+k)^{3/4}\sqrt{C\nu(C_r + C\nu)T}\right).$$

Remark 6.3. Oracle requires that the matrices Q and R are PSD. Nonetheless, we invoke Oracle using the perturbed cumulative loss $(\hat{Q}_t + Q_t^p, \hat{R}_t + R_t^p)$ that might not be PSD, as the perturbations Q_t^p and R_t^p themselves are typically not PSD. To solve this issue, we first notice that with high-probability (Vershynin, 2010), we have $\|Q_t^p\| \le O(d/\eta)$ and $\|R_t^p\| \le O(k/\eta)$. Therefore, to guarantee that the perturbed cumulative loss is PSD, we can add an initial large pretend loss by setting $\widehat{Q}_1 = (d/\eta)I$ and $\widehat{R}_1 = (k/\eta)I$. This would contribute an $O(C\nu(d+k)/\eta)$ term to the regret which ensures that, by our choice of η, Theorem 6.2 still holds.

Proof of Theorem 6.2. Let $\widehat{X}_1, \ldots, \widehat{X}_T$ be the actual observed covariance matrices induced by Algorithm 2. Also, let $\widehat{X}_1^\star, \ldots, \widehat{X}_T^\star$ be the covariances induced by using a fixed control matrix $K^\star$ throughout. Similarly, define $X_1, \ldots, X_T$ to be the covariance matrices of the steady-state distributions induced by $K_1, \ldots, K_T$ respectively, and $X^\star$ that of $K^\star$.

As in the analysis of OGD, the expected regret can be decomposed as follows:

$$\begin{aligned}
&\sum_{t=1}^T (Q_t + K_t^\mathsf{T} R_t K_t) \bullet \widehat{X}_t - (Q_t + (K^\star)^\mathsf{T} R_t K^\star) \bullet \widehat{X}_t^\star \\
&\quad = \sum_{t=1}^T (Q_t + K_t^\mathsf{T} R_t K_t) \bullet (\widehat{X}_t - X_t) \\
&\quad + \sum_{t=1}^T (Q_t + K_t^\mathsf{T} R_t K_t) \bullet X_t - (Q_t + (K^\star)^\mathsf{T} R_t K^\star) \bullet X^\star \\
&\quad + \sum_{t=1}^T (Q_t + (K^\star)^\mathsf{T} R_t K^\star) \bullet (X^\star - \widehat{X}_t^\star). \qquad (10)
\end{aligned}$$

The second term in Eq. (10), the regret in the "idealized setting", is bounded due to Kalai & Vempala (2005). It requires the additional observation that, by Lemma 3.3, we have $\mathrm{Tr}(X^\star) + \mathrm{Tr}(K^\star X^\star (K^\star)^\mathsf{T}) \le \nu$.

Lemma 6.4. Assume $\mathrm{Tr}(Q_t), \mathrm{Tr}(R_t) \le C$ for all t. Then,

$$\mathbb{E}\left[\sum_{t=1}^T \mathrm{Tr}(X_t(Q_t + K_t^\mathsf{T} R_t K_t)) - \mathrm{Tr}(X^\star(Q_t + (K^\star)^\mathsf{T} R_t K^\star))\right] \le 8\eta C^2 \nu \sqrt{d+k}\, T + \frac{16\nu(d+k)}{\eta}.$$

Moreover, the probability that the algorithm changes K_t and performs a reset at any step t is at most $\eta C\sqrt{d+k}$.

The third term of Eq. (10) is bounded by $2C\kappa^4\nu/\gamma$ due to Lemma 5.4. It remains to bound the first term in the equation. To that end, we will next show that after the system is reset, the cost of the learner on round t is at most that of the steady-state induced by K_t.

Lemma 6.5. Suppose the learner starts playing K at state $x_{t_0} = w_{t_0}$. Then the expected cost of the learner is always less then the steady-state cost induced by K.

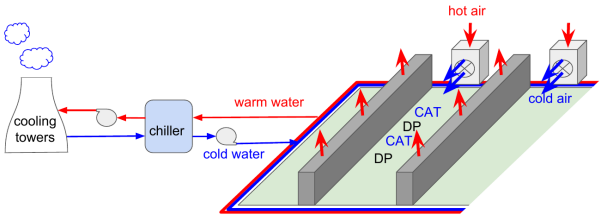

Figure 1. Data center cooling loop; see Section 7.

Proof. Let $x_{t_0} = w_{t_0}$, and recall that $x_{t+1} = (A+BK)x_t + w_t$. Let $\widehat{X}_t$ be the covariance of x_t, and X be the covariance of x at the steady-state induced by K. Then, $X_{t_0} = (A + BK_{t_0})X_{t_0}(A + BK_{t_0})^\mathsf{T} + W$.

We now show that $\widehat{X}_t \preceq X$ for all $t \geq t_0$ by induction. Indeed, for the base case $\widehat{X}_{t_0} = W \preceq (A+BK_{t_0})X_{t_0}(A+BK_{t_0})^\mathsf{T}+W = X$. Now assume that $\widehat{X}_t \preceq X_{t_0}$, that implies

$$\begin{aligned}\widehat{X}_{t+1} &= (A + BK_{t_0})\widehat{X}_t(A + BK_{t_0})^\mathsf{T} + W \\ &\preceq (A + BK_{t_0})X_{t_0}(A + BK_{t_0})^\mathsf{T} + W = X\ .\end{aligned}$$

Since $Q_t + K_t^\mathsf{T} R_t K_t$ is PSD, the expected cost of the learner at time t is $(Q_t + K_t^\mathsf{T} R_t K_t) \bullet X_t \leq (Q_t + K_t^\mathsf{T} R_t K_t) \bullet X$. □

Combining Lemmas 6.4 and 6.5 obtains the theorem (see the full version of the paper for more details). □

7. Experiments

We demonstrate our approach on the problem of regulating conditions inside a data center (DC) server floor in the presence of time-varying power costs. We learn system dynamics from a real data center, but vary the costs and run algorithms in simulation.

Fig. 1 shows a schematic of the cooling loop of a typical data center. Water is cooled to sub-ambient temperatures in the chiller and evaporative cooling towers, and then sent to multiple air handling units (AHUs) on the server floor. Server racks are arranged into rows with alternating hot and cold aisles, such that all hot air exhausts face the hot aisle. The AHUs circulate air through the building; hot air is cooled through air-water heat exchange and blown into the cold aisle, and the resulting warm water is sent back to the chiller and cooling towers. The primary goal of floor-level cooling is to control the cold aisle temperatures (CATs) and differential air pressures (DPs). The control vector includes the blower speed and water valve command for each of $n = 30$ AHUs, set every 30s. The state vector includes $2n$ temperature measurements and n pressure measurements, as well as sensor measurements and controls for the preceding time step. System noise is in part due to variability in server loads and the temperature of the chilled water.

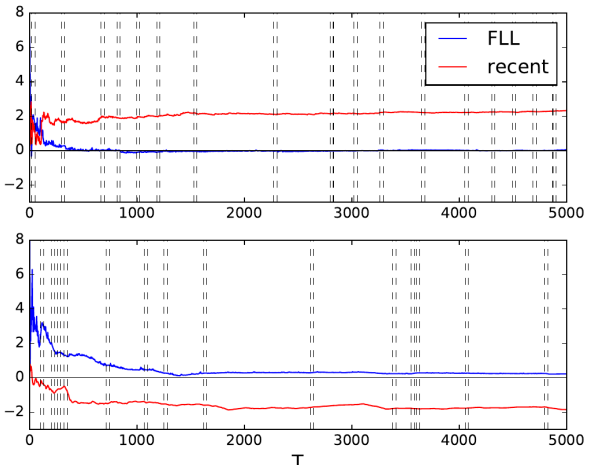

Figure 2. Normalized regret R_T/T for FLL and Recent strategies, with power costs generated uniformly (top) and by random walk (bottom). Resets occur at time steps indicated by dashed lines.

We learn a linear approximation (A, B) of the dynamics in the operating range of interest on 4h of exploratory data with controls following a random walk. We estimate the system noise covariance W as the empirical covariance of training data residuals. For the purpose of the experiment, we amplify the noise by a factor of 5. We set the diagonal coefficients of Q_t corresponding to the most recent (normalized) sensor measurements to 1 and remaining coefficients to 0, and keep $Q_t = Q$ constant throughout the experiment. We set diagonal coefficients of R_t corresponding to water usage (valve command) to 1 throughout, and all coefficients corresponding to power usage (fan speed) to r_t. We generate r_t by (a) i.i.d sampling a uniform distribution on $[0.1, 1]$, and (b) using a random walk restricted to $[0.1, 1]$ taking steps of size $0.1, -0.1, 0$ with probabilities $0.1, -0.1, 0.8$ respectively.

We run the FLL algorithm on this problem with the following modifications: we set $Q_1^p = Q$, and $R_1^p = I_k$, an upper bound on R_t. Rather than executing hard resets to 0, we perform a soft reset by running a policy K_{reset} for n steps. Here K_{reset} is similar to the next FLL policy, but based on the 1.1 times the corresponding state cost Q.

We compare the cost of FLL to that of a fixed linear controller that is based on the average of the R_t matrices, and to a *Recent* strategy which selects one of ten controllers corresponding to power costs in $r \in \{0.1, 0.2, ..., 1\}$ based on the most recently observed R_t. The normalized regret $\frac{1}{T}R_T$ of to the two strategies is shown in Fig. 2. FLL performance quickly approaches that of the fixed linear policy in both cases, and is better than the *Recent* strategy on uniform random costs. The *Recent* strategy has an advantage in the case where costs vary slowly, and empirical performance of FLL could likely be improved in this case by forgetting the old costs.

References

Abbasi, Yasin, Bartlett, Peter L, Kanade, Varun, Seldin, Yevgeny, and Szepesvári, Csaba. Online learning in markov decision processes with adversarially chosen transition probability distributions. In *Advances in neural information processing systems*, pp. 2508–2516, 2013.

Abbasi-Yadkori, Yasin and Szepesvári, Csaba. Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the 24th Annual Conference on Learning Theory*, pp. 1–26, 2011.

Abbasi-Yadkori, Yasin, Bartlett, Peter, and Kanade, Varun. Tracking adversarial targets. In *International Conference on Machine Learning*, pp. 369–377, 2014.

Abbeel, Pieter, Coates, Adam, Quigley, Morgan, and Ng, Andrew Y. An application of reinforcement learning to aerobatic helicopter flight. In *Advances in neural information processing systems*, pp. 1–8, 2007.

Abeille, Marc and Lazaric, Alessandro. Thompson sampling for linear-quadratic control problems. In *AISTATS*, 2017.

Anderson, BDO, Moore, JB, and Molinari, BP. Linear optimal control. *IEEE Transactions on Systems, Man, and Cybernetics*, (4):559–559, 1972.

Arora, Sanjeev, Hazan, Elad, Lee, Holden, Singh, Karan, Zhang, Cyril, and Zhang, Yi. Towards provable control for unknown linear dynamical systems. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BygpQlbA-. workshop track.

Åström, Karl Johan and Wittenmark, Björn. On self tuning regulators. *Automatica*, 9(2):185–199, 1973.

Auer, Peter and Ortner, Ronald. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 49–56, 2007.

Balakrishnan, Venkataramanan and Vandenberghe, Lieven. Semidefinite programming duality and linear time-invariant systems. *IEEE Transactions on Automatic Control*, 48(1):30–41, 2003.

Bertsekas, Dimitri P. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

Bittanti, Sergio and Campi, Marco C. Adaptive control of linear time invariant systems: the bet on the best principle. *Communications in Information & Systems*, 6(4):299–320, 2006.

Bradtke, Steven J. Reinforcement learning applied to linear quadratic regulation. In *Advances in neural information processing systems*, pp. 295–302, 1993.

Campi, Marco C and Kumar, PR. Adaptive linear quadratic gaussian control: the cost-biased approach revisited. *SIAM Journal on Control and Optimization*, 36(6):1890–1907, 1998.

Cesa-Bianchi, Nicolo and Lugosi, Gábor. *Prediction, learning, and games*. Cambridge university press, 2006.

Dean, Sarah, Mania, Horia, Matni, Nikolai, Recht, Benjamin, and Tu, Stephen. On the sample complexity of the linear quadratic regulator. *arXiv preprint arXiv:1710.01688*, 2017.

Dvijotham, Krishnamurthy, Todorov, Emanuel, and Fazel, Maryam. Convex control design via covariance minimization. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pp. 93–99. IEEE, 2013.

Even-Dar, Eyal, Kakade, Sham M, and Mansour, Yishay. Online markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

Fazel, Maryam, Ge, Rong, Kakade, Sham M, and Mesbahi, Mehran. Global convergence of policy gradient methods for linearized control problems. *arXiv preprint arXiv:1801.05039*, 2018.

Gao, Jim and Jamidar, Ratnesh. Machine learning applications for data center optimization. *Google White Paper*, 2014.

Hazan, Elad. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Hazan, Elad, Singh, Karan, and Zhang, Cyril. Learning linear dynamical systems via spectral filtering. In *Advances in Neural Information Processing Systems*, pp. 6705–6715, 2017.

Ibrahimi, Morteza, Javanmard, Adel, and Roy, Benjamin V. Efficient reinforcement learning for high dimensional linear quadratic systems. In *Advances in Neural Information Processing Systems 25*, pp. 2636–2644. Curran Associates, Inc., 2012.

Kalai, Adam and Vempala, Santosh. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Lee, Dong-Hwan and Hu, Jianghai. A semidefinite programming formulation of the lqr problem and its dual. 2016.

Lee, Ji-Woong and Khargonekar, Pramod P. Constrained infinite-horizon linear quadratic regulation of discrete-time systems. *IEEE Transactions on Automatic Control*, 52(10):1951–1958, 2007.

Levine, Sergey, Finn, Chelsea, Darrell, Trevor, and Abbeel, Pieter. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Lewis, Frank L and Vrabie, Draguna. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE circuits and systems magazine*, 9(3), 2009.

Neu, Gergely and Gómez, Vicenç. Fast rates for online learning in linearly solvable markov decision processes. *Proceedings of Machine Learning Research vol*, 65:1–22, 2017.

Schildbach, Georg, Goulart, Paul, and Morari, Manfred. Linear controller design for chance constrained systems. *Automatica*, 51:278–284, 2015.

Shalev-Shwartz, Shai. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Sheckells, Matthew, Garimella, Gowtham, and Kobilarov, Marin. Robust policy search with applications to safe vehicle navigation. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 2343–2349. IEEE, 2017.

Todorov, Emanuel. Efficient computation of optimal actions. *Proceedings of the national academy of sciences*, 106 (28):11478–11483, 2009.

Vershynin, Roman. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Yu, Jia Yuan, Mannor, Shie, and Shimkin, Nahum. Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research*, 34(3):737–757, 2009.

Zhou, Kemin, Doyle, John Comstock, Glover, Keith, et al. *Robust and optimal control*, volume 40. Prentice hall New Jersey, 1996.

Zinkevich, Martin. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 928–936, 2003.