A Gradient-Based Policy Update for Time-Varying Online LQR

Yi Zhang
Princeton University
y.zhang@cs.princeton.edu

Abstract

Algorithms for the online Linear Quadratic Regulator problem have largely relied on either improper learning (convex relaxation) or solving the Riccati equation exactly. While being simple and popular in practice, there have been no provable policy gradient-based methods for online LQR, due to the non-convexity of the objective in the policy matrix. In this work, we prove that, in spite of the non-convex nature of the parametrization, a certain pre-conditioned policy gradient update achieves $\tilde{O}(\sqrt{T})$ regret for the online LQR problem with unknown systems and stochastic disturbances, even if the system parameters shift adversarially by $O(1/T)$ every time step and thus may be completely different after $\Theta(T)$ steps. To the best of our knowledge, this is the first regret guarantee for time-varying online LQR.

1 Introduction

Great empirical successes of reinforcement learning witnessed recently in continuous control [12, 9, 13] have ignited an enormous interest in providing supporting theory and developing algorithms with provable guarantees. Many recent works have made remarkable progress by re-examining classical control problems via a modern algorithmic lens, and the online Linear Quadratic Regulator problem stands out as a standard benchmark for theoretical analysis, partly because it inherits rich theory from classical optimal control studies, and yet captures the core difficulty of the exploration-exploitation trade-off in reinforcement learning.

Gradient-based optimization on parameterized policy functions that map states to actions have emerged as a main driving force on the empirical side of reinforcement learning [18, 17, 3, 11], thanks to its simplicity and versatility. While its efficacy has been mathematically shown in the offline setting of LQR [7], we have yet to find a use for it in the more challenging online setting due to the commonly perceived non-convexity of the objective function with respect to the policy parameters. Indeed, despite the surprising discovery of [7], numerous works in online LQR have continued to cite such non-convexity as the motivation for improper learning or convex relaxation [4, 1, 20, 6].

In this paper, we challenge the canonical view and mathematically show that, in spite of the non-convexity, a certain pre-conditioned policy gradient update achieves $\tilde{O}(\sqrt{T})$ regret for online LQR with unknown systems and stochastic disturbances. In fact, we do so by proving a stronger statement: the same algorithm achieves $\tilde{O}(\sqrt{T})$ regret even if the (unknown) system parameters shift adversarially by $O(1/T)$ every time step, by virtue of properties unique to gradient-based methods. To the best of our knowledge, this is the first regret guarantee for online control of unknown, time-varying linear dynamical systems. We remark that this setting is challenging since the system parameters may become completely different after $\Theta(T)$ steps, and we provide reasons why previous techniques may fail to achieve $\tilde{O}(\sqrt{T})$ in black-box fashion in Section 4.

Preprint. Under review.
2 Related Works

Online LQR  Online LQR with stochastic disturbances has been studied extensively in the recent past. Assuming the system parameters and the cost matrices are known, [4] proposes a SDP-relaxation based algorithm that guarantees $O(\sqrt{T})$ regret. [2] obtains logarithmic regret with strongly convex cost functions with improper learning which learns a policy mapping disturbances (rather than states) to actions. In the case of unknown systems, [5] presents the first computation-efficient algorithm, based on the aforementioned SDP relaxation, that achieves $\tilde{O}(\sqrt{T})$ regret, and [15] shows certainty equivalence controller which solves the Riccati equation exactly based on estimated system parameters enjoys $\tilde{O}(\sqrt{T})$ as well. Recently, [19] shows the $O(\sqrt{T})$ regret obtained by naive $\epsilon$-greedy exploration and solving Riccati equation is in fact optimal for online LQR by proving a matching lower bound.

Non-stochastic Control. A great body of work has focused on linear control with non-stochastic disturbances, all of which have relied on the the same improper learning parametrization as [2]. With known systems, [1] obtains $O(\sqrt{T})$ regret with general convex losses assuming bounded adversarial disturbances, and [8] presents an algorithm with $O(\log^3(T))$ regret if the costs are quadratic. With unknown systems, [10] proposes an explore-then-commit scheme that guarantees $O(T^{3/2})$ regret. [20] extends the setting to consider partially observable systems. We remark that the non-stochastic disturbances considered by previous works are state independent and thus cannot model the system parameter shift in our setting.

Policy Gradient for Offline LQR The seminal work of [7] is the first to prove the surprising global linear convergence of policy gradient methods for the offline LQR, despite its non-convex nature, which is later improved by [14]. The key ingredient in both [7] and [14] and is the so-called Gradient Dominance property or equivalently the Polyak-Łojasiewicz property [16] of policy gradient. However, Gradient Dominance does not imply regret guarantee in general. Therefore, in spite of the discovery of [7], various online LQR literatures [4, 1, 20, 6] have continued to quote the non-convexity as one of the reasons for improper learning or convex relaxation. Even though our proposed update rule, i.e., Equation 3 has been analyzed in [7] in the offline setting, our key observation (i.e., Inequality 2) that enables its regret guarantee has not been stated or utilized previously, and our proof techniques are distinctly different. Note that our Inequality 2 implies the Gradient Dominance property, but not vice versa.

3 Background

Online LQR with Time-Varying Systems  The evolution of the state $x_t \in \mathbb{R}^d$ of a time-varying linear dynamical system is governed by the following transition:

$$x_{t+1} = A_t x_t + B_t u_t + w_t$$

where $u_t \in \mathbb{R}^k$ is the action to the system and $w_t$’s are i.i.d. from $N(0, \sigma_w^2 I_d)$. In general, system parameters $A_t, B_t$ are different across time. In the online LQR problem, at each time step, the learner chooses an action $u_t$ upon observing the state $x_t$, and then suffers the cost $c(x_t, u_t) := x_t^T Q x_t + u_t^T R u_t$ for the fixed positive definite matrices $Q$ and $R$. The goal of the learner is to minimize the sum of expected costs sequentially revealed over time, i.e., $E \left[ \sum_{t=1}^{T} c_t(x_t, u_t) \right]$. In our setting, the system parameters are unknown but the cost matrices are known to the learner.

Time-invariant Systems  When the system parameters remain constant across time, i.e. $A_t = A, B_t = B$ for all $t$, it reduces to a time-invariant linear dynamical systems. The steady-state cost of a policy $\pi : \mathbb{R}^d \to \mathbb{R}^k$ on the system $(A, B)$ is defined as

$$J_{A,B}(\pi) := \lim_{T \to \infty} \frac{1}{T} E \left[ \sum_{t=1}^{T} c_t(x^\pi_t, u^\pi_t) \right]$$

where $(x^\pi_t, u^\pi_t)_{t=0}^T$ is the trajectory generated according to the execution of $\pi$. A well-known fact in control theory is that the optimal steady-state cost can be achieved by a fixed linear function.
When the dynamics are known, such policy can be found by solving the associated Riccati equation \([19, 15]\). With an abuse of notation, we denote the steady-state cost of a linear policy \(K\) by \(J_{A,B}(K)\).

**System \((A, B)\)** Here we introduce the parenthetical notation \((A, B)\) for the time-invariant system specified by matrices \(A, B\). We emphasize the difference between the hypothetical system \((A_t, B_t)\) for any fixed \(t \in [T]\) and the actual time-varying system.

In order to achieve finite steady-state cost, it is necessary for the states to remain bounded under the execution of \(K\). In such scenario the policy \(K\) is **stabilizing** for the system or \(\rho(A + BK) < 1\). The following definition offers a quantification of stability.

**Definition 3.1 (Strong Stability).** A linear policy \(K\) is \((\kappa, \gamma)\)-strongly stable for a linear dynamical system \((A, B)\) if \(\|K\|_2 \leq \kappa\) and there exists a decomposition \(A + BK = QLQ^{-1}\) with \(\|L\|_2 \leq 1 - \gamma\) and \(\|Q\|_2, \|Q^{-1}\|_2 \leq \kappa\). A system \((A, B)\) is \((\kappa, \gamma)\)-strongly stabilizable if there exists a linear policy \(K\) that is \((\kappa, \gamma)\)-strongly stable for it.

Essential to our analysis is the notion of the steady-state covariance matrix defined for any stabilizing linear policy \(K\) for the system \((A, B)\),

\[
\Sigma_{A,B}(K) := \lim_{T \to \infty} \frac{1}{T} \mathbb{E} \left[ \sum_{t=0}^{T} x_t x_t^\top \right] = \sigma_w^2 \sum_{t=0}^{\infty} (A + BK)^t \left( (A + BK)^t \right)^\top
\]

with which the steady-state cost can be expressed as \(J_{A,B}(K) = \text{Tr}(\Sigma_{A,B}(K)(Q + K^T R K))\).

**Additional notations** \(\| \cdot \|_2\) denotes the \(\ell_2\) (or operator) norm if the argument is a vector (or matrix). For simplicity, we write \(\Sigma_t(K) := \Sigma_{A_t,B_t}(K)\), \(\Sigma_t^* := \Sigma_t(K^*)\), \(J_t(K) := J_{A_t,B_t}(K)\). When the context is clear, we drop the time indices. Furthermore, we define the inner product associated with a positive definite matrix \(M\) as \(\langle K_1, K_2 \rangle_M := \text{Tr}(MK_1^TK_2)\) and its induced norm as \(\|K\|^2_M := \langle K, K \rangle_M\).

### 4 Main Result

In this section, we present the assumptions and the main theorem.

**Assumptions** Here we make a complete list of the assumptions made in this work. Specifically, we assume there exist fixed positive constants \(C, D, \kappa, \gamma, \alpha, \beta, \sigma_w\) such that

1) **slowly varying systems:** \(\forall t \in [T]\), \(\|A_t - A_{t-1}\|_2, \|B_t - B_{t-1}\|_2 \leq \frac{C}{T}\) and \(\|A\|_2, \|B\|_2 \leq D\).
2) **stabilizability:** \(\forall t \in [T]\), the hypothetical time-invariant system \((A_t, B_t)\) is \((\kappa, \gamma)\)-strongly stabilizable.
3) **stochastic disturbances:** \(\forall t \in [T]\), \(w_t\) is drawn i.i.d. from \(N(0, \sigma_w^2 I_d)\).
4) **well-conditioned cost matrices:** \(\alpha I_d \preceq Q \preceq \beta I_d, \alpha I_k \preceq R \preceq \beta I_k\).
5) **stable initialization:** the initial policy \(K_0\) given to the learner is \((\kappa, \gamma)\)-strongly stable for the true, but unknown system \((A_0, B_0)\).

For presentation of the main theorem, we define the constants listed above, their inverses, together with the dimensions \(d, k\), as the relevant constants. We introduce three fixed polynomials \(T_{\text{min}}, c, \kappa\) and \(\bar{\kappa}\) in all relevant constants with the explicit definitions are deferred to the Appendix. We provide a proof sketch in Section 4.

**Theorem (Main).** For any \(T \geq T_{\text{min}}\), we have for the \((x_t, u_t)\)'s generated by Algorithm 1 with appropriately chosen hyper-parameters specified in Section 4 that

\[
\sum_{t=0}^{T} \mathbb{E} \left[ c(x_t, u_t) \right] - \min_{\pi \in \Pi} \sum_{t=0}^{T} \mathbb{E} \left[ c(x_t^\pi, u_t^\pi) \right] \leq O \left( \text{poly}(\text{relevant constants}) \sqrt{T \log^3 T} \right)
\]

where the comparator policy class \(\Pi\) is defined as the set of all piecewise-linear policy \(\pi\) consisting of \(\left\lfloor \frac{1}{c} \right\rfloor\) many linear policies \(K_{(i)}\)'s such that for all \(i \in [0, \ldots, \left\lfloor \frac{1}{c} \right\rfloor]\),

\[
\forall t \in [i \cdot cT, \ldots, (i+1) \cdot cT - 1], \quad \pi(x_t | t) := K_{(i)} x_t
\]

and \(K_{(i)}\) is \((\bar{\kappa}, \frac{1}{4c^2})\)-strongly stable for all system \((A_t, B_t)\)'s with \(t \in [i \cdot cT, \ldots, (i+1) \cdot cT - 1]\).
Remarks  Here we make the following remarks.

- It is conventional to compare against any fixed stable linear policy in online LQR. However, there may not exist such a policy in our setting where the system parameters may shift by $\Theta(1)$ in total of $T$ time steps. Instead, we consider a more general comparator class, consisting of piecewise-linear policies, that is guaranteed to be non-empty by carefully choosing the values of $c$ and $\bar{c}$.

- Previous techniques for online LQR with fixed, unknown systems, including improper learning, SDP relaxation and solving Riccati equation, may not obtain $\tilde{O}(\sqrt{T})$ regret in the time-varying setting, at least not in black-box fashion. To do so, they would have to use exploratory noise with variance at most $O(T^{-\frac{1}{2}})$, i.e., $\sigma_u^2 \leq O(T^{-\frac{1}{2}})$, and would require estimating the system parameters to $O(T^{-\frac{1}{2}})$ accuracy, i.e., $\epsilon_{A,B} := \|\hat{(A,B)} - (A, B)\|_2 \leq O(T^{-\frac{1}{2}})$. To see why this may not be possible, this would require the algorithm to collect data for at least $O(\sigma_u^{-2} \epsilon_{A,B}^{-2}) = \Omega(T)$ time steps, during which the system parameters may have already become off by $\Omega(1)$. On the contrary, one strength unique to gradient-based updates is they only require the variance of the estimator to be $O(1)$ as long as it is (almost) unbiased.

5 The Preconditioned Policy Gradient Update

The seminal work of [7] derives an explicit formula for the steady-state cost for time-invariant LQR. Specifically, for any stabilizing linear policies $K$ and $K^*$, they show

$$ J_{A,B}(K) - J_{A,B}(K^*) = 2\text{Tr} \left( \Sigma(K^*)(K - K^*)^\top \Sigma(K)^{-1} \nabla J_{A,B}(K) \right) - \text{Tr} \left( \Sigma(K^*)(K - K^*)^\top (R + B^\top P_K B)(K - K^*) \right) $$

(1)

for the positive definite matrix $P_K := \sum_{t=0}^\infty (A + BK)^t (Q + K^\top RK) [(A + BK)^t]^\top$ and the matrix $E_K := \Sigma_{A,B}(K)^{-1} \nabla_K J_{A,B}(K) := RK + B^\top P_K (A + BK)$ where $\nabla_K J_{A,B}(K)$ denotes the gradient of $J_{A,B}$ with respect to $K$. With the knowledge of $A, B$, the infinite sum can be efficiently well-approximated by a finite sum assuming strong stability. From Equation 1, the authors of [7] prove the so-called Gradient Dominance property [16] and combine it with smoothness of $J_{A,B}$ to show the offline global linear convergence. In general, however, Gradient Dominance does not imply regret guarantee in the online setting.

We take a completely different approach from Equation 1, and perform a relaxation\footnote{here we use the inequality $\text{Tr}(MN) \geq \lambda_{\text{min}}(M) \text{Tr}(N)$ for positive semi-definite matrices $M, N$.} to obtain:

$$ J_{A,B}(K) - J_{A,B}(K^*) \leq 2\langle K - K^*, E_K \Sigma(K^*) \rangle - \alpha \|K - K^*\|_{\Sigma(K^*)}^2 $$

(2)

where we use the assumption $R \succeq \alpha I_k$. This is reminiscent of the definition of strongly convex functions, i.e. $f(x) = f(x^*) - \langle x - x^*, \nabla f(x) \rangle \geq \frac{\alpha}{2}\|x - x^*\|_2^2$. The key differences here are two-fold: (1) Inequality 2 depends on the preconditioned gradient matrix $E_K$ while strong convexity is on the vanilla gradient $\nabla f(x)$, and (2) Inequality 2 utilizes an inner product (and its induced norm) depending the reference point $K^*$ while normally strong convexity is stated with the universal $\ell_2$ norm. Note that “Gradient Dominance” is a corollary of strong convexity but not vice versa.

Equation 2 provides a strong motivation for mimicking the Online Gradient Descent proof for strongly convex functions, but with a slight modification: we follow opposite direction of the preconditioned gradient instead of the gradient, i.e.

$$ K_{t+1} := K_t - \eta_t E_K $$

(3)

Indeed, at least for the offline optimization of $J_{A,B}$ with fixed, known system parameters, it is not difficult to show that such an update rule with appropriately chosen $\eta_t$ can achieve the $\tilde{O}(T^{-1})$ rate.

Surprisingly, we show that such an simple update rule is able to achieve $\tilde{O}(\sqrt{T})$ regret in the much more challenging setting, where we consider the time-varying online LQR problem with unknown systems. This result is enabled by one major advantage unique to gradient-based methods: the unbiased estimator of $E_K$, at each step, may tolerate constant-variance uncertainty, as in traditional stochastic convex optimization. The gradient descent-like proof is demonstrated in Section 7.3.
6 Algorithm

In this section, we describe the main algorithm and its two subroutines. From a high-level, the main algorithm performs the preconditioned gradient update to the policy $K_\tau$’s on $J_{\tau_m}(\cdot)$ using the estimated update $\hat{E}_\tau$ constructed from system parameters estimated from recent past rounds.

Concretely, the main algorithm partitions the entire $T$ horizon into $\frac{T}{m}$ many blocks of length $m$, and the blocks are indexed by $\tau$. Within the $\tau$-th block, the algorithm executes the same policy $K_\tau$ plus an exploratory Gaussian noise $\xi$ for the purpose of estimating the matrix $B$. Within the same block, we obtain a fresh estimate of the system parameters every $T_0$ rounds, by calling the subroutine Algorithm 2 which solves a linear regression. At the end of each block, we construct an estimate of the update $\hat{E}_\tau$ using the estimates, and then perform an update to the policy with learning rate $\eta_\tau$.

Algorithm 1 Preconditioned policy gradient learner for online LQR

**Input:** block length $m$, learning rate schedule $\eta_\tau$, initial policy $K_0$, horizon $H$, estimation duration $T_0$, exploration variance $\sigma^2_u$ for $\tau = 0, \ldots, \lfloor \frac{T}{m} \rfloor$

for $\tau = 0, \ldots, \lfloor \frac{T}{m} \rfloor$ do
  for $i = 0, \ldots, \lfloor \frac{mT_0}{T} \rfloor$ do
    for $t = \tau m + iT_0, \ldots, \tau m + (i + 1)T_0 - 1$ do
      Execute the action $u_t = K_\tau x_t + \xi_t$ where $\xi_t \sim N(0, \sigma^2_u I_d)$.
      Record the observed state $x_t$.
    end for
  end for
  Call Algorithm 2 on the past $T_0$ recorded states to collect an estimate $((\hat{A} + BK)_i, \hat{B}_i)$
end for

Call Algorithm 3 on the $m$ collected estimate $((\hat{A} + BK)_i, \hat{B}_i)$’s to get the update $\hat{E}_\tau$

Update the policy $K_{\tau + 1} = K_\tau - \eta_\tau \hat{E}_\tau$
end for

Algorithm 2 System estimation

**Input:** number of iterations $T_0$, recorded states $(x_0, \ldots, x_{T_0})$, injected noises $(\xi_0, \ldots, \xi_{T_0})$, exploration variance $\sigma^2_u$ Let $N := \lfloor \frac{T_0}{H} \rfloor$

return $A + BK, \hat{B} := \arg \min_{M,H} \sum_{t=1}^N \| M x_{tH - 1} + B \xi_{tH - 1} - x_{tH} \|^2$

Notably, in Algorithm 2 only uses samples that are $H$ rounds apart. As will be seen in the next section, $H$ is set to be larger than the mixing time of the system so that the returned estimates are sufficiently independent, allowing the update constructed by Algorithm 3 to be almost unbiased, which is essential to our analysis.

Algorithm 3 Update construction

**Input:** Estimated system parameters $(\hat{B}^{(i)}, (\hat{A} + BK)^{(i)})$ for $i = 0, \ldots, \lfloor \frac{m}{T_0} \rfloor$, policy $K$

Initialize $\hat{P} := \sigma^2_w (Q + K^\top RK)$

for $j = 0, \ldots, \lfloor \frac{m}{T_0} \rfloor - 2$ do
  $\hat{P} := (A + BK)^{(2j)} \hat{P} (A + BK)^{(2j+1)} + \sigma^2_w (Q + K^\top RK)$
end for

return $\hat{E} := RK + \hat{B}^{(0)} \hat{P} (A + BK)^{(1)}$
7 Analysis

In this section, we provide a proof overview of the main theorem as well as the key lemmas.

Proof Overview We first list the scale of hyper-parameters of the algorithms $m = \Theta(\sqrt{T \log^2 T})$, $T_0 = \Theta(\sqrt{T \log T})$, $H = \Theta(\log T)$, $\sigma_u^2 = T^{-\frac{4}{3}}$ and $\eta_\tau$ is specified in Section 7.3. The main challenge in our proof is to show the update rule maintains strong stability. Note that projection is not an option here, because the system parameters are unknown and the set of strongly stable policies is non-convex. We give evidence of the strong stability by proving an upper bound on the steady-state cost across time (i.e., Lemma 7.2), which is, in turn, proved with induction by utilizing the Descent Lemma 7.4 to offset the increase in cost due to system parameter shift and noise in the update estimator. Notably, Lemma 7.4 has two requirements: (1) the variance of the estimator is below a constant threshold, and (2) the learning rate is $\Omega(mT^{-1}) \leq \eta \leq O(1)$.

In order to construct the update, we need to estimate the system parameters, specifically $B_t$’s, by injecting exploratory noises. For the purpose of $O(\sqrt{T})$ regret, it is clear the variance of the noise can at most be $\sigma_u := T^{-\frac{4}{3}}$, since the exploration incurs extra cost of $T\sigma_u^2$. In order to satisfy requirement (1), $T_0$ needs to be $\Theta(H\sqrt{T})$, since the variance of the estimator scales as $O\left(\sqrt{\frac{H}{T\sigma_u^2}}\right)$.

This gives rise to $m = \Theta(\sqrt{T \log^2 T})$. Now the outer loop of Algorithm 1 is effectively a gradient-based algorithm for $\frac{J}{m} = \Theta(\sqrt{T})$ rounds. With a $1/T$-like learning rate schedule, the outer loop obtains $\log(T/m)$ regret, which leads to $m \log(T/m) = O(\sqrt{T \log^3 T})$ overall regret. However, we still need to satisfy requirement (2) that $\Omega(T^{-\frac{4}{3}}) \leq \eta \leq O(1)$. To this end, we design a cyclic learning rate schedule specified in Section 7.3, which exhibits the typical $1/T$ decay, and immediately restarts once it hits the lower bound. Note that the schedule restarts only constant times, which maintains the overall regret’s dependence on $T$.

7.1 Maintaining stability

The next lemma shows an upper bound on the steady-state cost $J_{A,B}(K)$ in fact translates into the strong stability the policy $K$.

Lemma 7.1. If $J_{A,B}(K) \leq \nu$ for a finite $\nu$ and $\kappa := \frac{1}{\sigma_u} \sqrt{\frac{T}{\eta}}$, then $K$ is $(\kappa, \frac{1}{\sigma_u})$-strongly stable for system $(A, B)$.

The steady-state cost remains bounded as long as the system shifts sufficiently slow, the learning rate magnitude is correctly set, and the variance of the update estimator is bounded. Indeed, there exist fixed polynomials $h_\epsilon, h_{\min}, h_{\max}, h_{\var}$ in relevant parameters such that the following is true.

Lemma 7.2. For $\epsilon < h_\epsilon, \eta \in [h_{\min} \cdot \epsilon, h_{\max}]$ and $\|\hat{E}_K - E_K\|_2 \leq h_{\var}$, if the system $(A, B)$ is $(\kappa, \gamma)$-strongly stabilizable, and $J_{A,B}(K) \leq 6\sigma_u^2 \beta \kappa^4 \gamma^{-1}$, and $\|A' - A\|_2, \|B' - B\|_2 \leq \epsilon$ and $K' = K - \eta \hat{E}_K$, then we have $J_{A',B'}(K') \leq 6\sigma_u^2 \beta \kappa^4 \gamma^{-1}$ as well.

The next corollary follows from a simple induction argument starting from $K_0$ which clearly satisfies the condition in Lemma 7.2 since $J_0(K_0) \leq 2\sigma_u^2 \beta \kappa^4 \gamma^{-1}$ by the assumption on initialization.

Corollary 7.3. For $T \geq \frac{mC}{h_{\min}}$, if for all $0 \leq \tau \leq \left\lfloor \frac{T}{m} \right\rfloor$, $\eta_\tau \in [h_{\min} \cdot \frac{mC}{\rho}, h_{\max}]$ and $\|\hat{E}_\tau - E_\tau\|_2 \leq h_{\var}$, then we have for all $K_\tau$’s generated by Algorithm 1 that

$$J_{m_\tau,\tau m}(K_\tau) \leq 6\sigma_u^2 \beta \kappa^4 \gamma^{-1}$$

Furthermore, $K_\tau$ is $(\sqrt{\frac{6\beta \kappa^4}{\alpha}}, \frac{\alpha \gamma}{12\beta \kappa})$-strongly stable for system $(A_m, B_{\tau m})$.

Essential to proving Lemma 7.2 is the following descent lemma for the preconditioned gradient update that asserts, there is a guaranteed amount of decrease in the steady-state cost which offsets the increase of cost due to the system shift and the noise in the estimator $\hat{E}_\tau$.

Lemma 7.4 (Descent lemma). If $\eta < \frac{1}{\|R_B + B\|_2}$ and $K' = K - \eta E_K$, then it holds that

$$J_{A,B}(K') - J_{A,B}(K^*) \leq \left(1 - \eta \frac{\alpha \sigma_u^2}{\|\Sigma(K^*)\|_2}\right) \left(J_{A,B}(K) - J_{A,B}(K^*)\right)$$

where $K^* := \arg \min_{K''} J_{A,B}(K'')$.
7.2 System Estimation

In the current and following setion, we denote \( \bar{\kappa} := \sqrt{\frac{6\beta k^4}{\alpha^2}} \) and \( H := 3\bar{\kappa}^2 \log T \). Corollary 7.3 suggests \( H \) is greater than the mixing time of the system.

**Theorem 7.5.** For \( T_0 \geq \frac{C\mu^2}{\gamma^2} (1 + 2k) H^2 h_{\text{var}}^{-1} \) and \( m \geq H \), the estimated update obtained from Algorithm 3 during the \( \tau \)-th loop in Algorithm 1 satisfies with high probability that

\[
\left\| \mathbb{E}[E_{T/\tau}] - E_T \right\|_2 \leq \frac{Cm}{T}, \quad \left\| E_T - \mathbb{E}[E_T] \right\|_2 \leq h_{\text{var}}
\]

7.3 Bounding the Regret

First we link the actual costs incurred by the algorithm to the fictitious steady-state costs by the following lemma.

**Lemma 7.6.** For \( \frac{T}{T_0} \geq \frac{C}{H^2} \), \( m > H \), and \( \eta_T \in [h_{\text{min}}, \frac{mC}{H}, h_{\text{max}}] \), then we have for the \( x_i \)'s, \( u_i \)'s, \( K_{\tau} \)'s generated from Algorithm 1 that

\[
\left| \sum_{t=0}^{T} \mathbb{E}[c(x_t, u_t)] - \left( \beta \sigma_u^2 T + m(1 + D^2 \sigma_u^2 / \sigma_w^2) \sum_{\tau=0}^{T/m} J_{\tau m}(K_{\tau}) \right) \right| \lesssim m(\sigma_u^2 + D^2 \sigma_u^2) H^2 C
\]

Now we take advantage of the power of Inequality 2 to demonstrate \( \sum_{\tau=0}^{T/m} J_{\tau m}(K_{\tau}) \) is only \( O(\log T) \). In fact, we mimic the proof of online gradient descent for strongly convex functions. Traditionally, such a proof requires the learning rate to scale as \( \frac{1}{\sqrt{T}} \). However, in our setting, this schedule may violate the requirement on the lower bound on \( \eta \), i.e. \( \eta_T \geq h_{\text{min}}, \frac{mC}{H} \) when \( T \) grows large. To reconcile the conflict, we adopt the following cyclic learning rate schedule.

**Cyclic learning rate schedule** The schedule consists of cycles of \( \left\lfloor \frac{T}{mT} \right\rfloor \) length \( T \), and within each cycle it exhibits a traditional \( \Theta \left( \frac{1}{\sqrt{T}} \right) \) annealing.

\[
\eta_T := \frac{1}{h_{\text{max}} + \alpha' \left( \tau - \left\lfloor \frac{\tau}{T} \right\rfloor \bar{T} \right)}
\]

For our purpose, we pick \( \bar{T} := \left( \frac{T}{mC} h_{\text{min}}^{-1} - h_{\text{max}}^{-1} \right) / \alpha' \) and \( \alpha' := \frac{4}{\max \{1, 256G^2 C / \sigma_w^2 \}} \) so that \( \eta_T \in [h_{\text{min}}, \frac{mC}{H}, h_{\text{max}}] \) is satisfied for all \( \tau \). The reason for choice of \( \alpha' \) shall be seen in the proof of the next lemma. Note it is essential that \( \bar{T} = \Theta \left( \frac{T}{m} \right) \), so the cyclic learning rate is equivalent to restarting the algorithm \( \frac{T}{mT} = \Theta(1) \) times without hurting the regret’s dependence on \( T \).

**Lemma 7.7 (Main Lemma).** With \( P_i := \left\lfloor i \bar{T}, \ldots, (i + 1) \bar{T} - 1 \right\rfloor \) denoting the indices within the \( i \)-th cycle, \( \eta_T \) set as above, \( T \geq \max \left\{ \frac{mC}{\bar{T}^2}, 1024h_{\text{max}}^{-1} m C \alpha^{-1} \sigma_w^{-2} \right\} \), and we have for the \( K_{\tau} \)'s generated by Algorithm 1 and \( K_{\tau} \)'s that are \( (\bar{\kappa}, \frac{1}{\sqrt{T}}) \)-strongly stable for system \( (A_{\tau m}, B_{\tau m}) \) for all \( \tau \in P_i \) that

\[
\sum_{\tau=0}^{T/m} J_{\tau m}(K_{\tau}) - \sum_{i=0}^{T/\bar{T}} \sum_{\tau \in P_i} J_{\tau m}(K_{\tau}^*) \lesssim \alpha' h_{\text{min}} \left( \bar{\kappa}^6 h_{\text{max}}^{-1} + \frac{G^2}{\alpha'} \log(T) \right)
\]

**Proof** First of all, we present a bound on the norm of \( \mathbb{E}[E_T] \). Since Corollary 7.3 suggests \( J_{\tau m}(K_{\tau}) \leq 6 \beta^2 \sigma_u^4 \bar{k}^4 \gamma^{-1} \) for all \( \tau \), from Lemma 11 in [7], and the designed deviation bound \( \left\| E_T - E_T \right\|_2 \leq h_{\text{var}} \) we obtain

\[
\left\| E_T \right\|_2 \leq \left( 6 \beta^2 \sigma_u^4 \bar{k}^4 \gamma^{-1} + 36 D^2 \beta^2 \bar{k}^8 \gamma^{-2} \right)^{1/2} \cdot dh_{\text{var}} \cdot 16 \bar{\kappa}^6 := G^2
\]
It suffices to show the regret incurred within the first cycle $\mathcal{P}_0$, and the total regret is simply bounded by the number of cycles $\frac{T}{mT} = \Theta(1)$ times that.

From Inequality 2 we have the following:

$$J_{\tau m}(K_{\tau}) - J_{\tau m}(K^*) \leq 2(K_{\tau} - K^*, E_{\tau} - \hat{E}_{\tau})_{\Sigma^*_{\tau m}} + \alpha\|K_{\tau} - K^*\|^2_{\Sigma^*_{\tau m}} + 2(K_{\tau} - K^*, E_{\tau} - \hat{E}_{\tau})_{\Sigma_{\tau m}}$$

Summing over all $\tau$'s within the first cycle $\mathcal{P}_0$, we obtain $\sum_{\tau=0}^{\tilde{T}} (J_{\tau m}(K_{\tau}) - J_{\tau m}(K^*)) \leq$ Expected Regret + Deviation

where

$$\text{Expected Regret} := \sum_{\tau=0}^{\tilde{T}} \left( \frac{\|K_{\tau} - K^*\|^2_{\Sigma^*_{\tau m}}}{\eta_\tau} - \frac{\|K_{\tau+1} - K^*\|^2_{\Sigma^*_{\tau m}}}{\eta_\tau} - \alpha\|K_{\tau} - K^*\|^2_{\Sigma_{\tau m}} \right) + G^2 \sum_{\tau=1}^{\tilde{T}} \frac{1}{\eta_\tau}$$

and

$$\text{Deviation} := \sum_{\tau=0}^{\tilde{T}} (K_{\tau} - K^*, E_{\tau} - \hat{E}_{\tau})_{\Sigma_{\tau m}}$$

We proceed to bound the expected regret and deviation respectively. The main challenge for bounding the expected regret is that the norm $\|\cdot\|_{\Sigma^*_{\tau m}}$ changes over time, invalidating the traditional telescoping. However, we show below that the strong convexity term can be used to cancel the change in norm:

$$\text{Expected Regret} \leq \frac{\|K_0 - K^*\|^2_{\Sigma_0}}{\eta_0} + \sum_{\tau=1}^{\tilde{T}} \|K_{\tau} - K^*\|^2_{\Sigma^*_{\tau m}} \left( \frac{1}{\eta_\tau} - \frac{1}{\eta_{\tau-1}} - \frac{\alpha}{4} \right) + \sum_{\tau=1}^{\tilde{T}} \|K_{\tau} - K^*\|_{\Sigma_{\tau m}}^2 G^2 \sum_{\tau=0}^{\tilde{T}} \eta_\tau$$

where we use $\|K_{\tau} - K^*\|^2_{\Sigma^*_{\tau m}} - \|K_{\tau} - K^*\|^2_{\Sigma^*_{(\tau-1)m}} \leq \|K_{\tau} - K^*\|^2_{\Sigma^*_{m}}\|\Sigma^*_{\tau m} - \Sigma^*_{(\tau-1)m}\|_2$ as well as $\|K_{\tau} - K^*\|^2_{\Sigma_{\tau m}} \leq \|K_{\tau} - K^*\|^2_{\Sigma_{m}}\|\Sigma^*_{\tau m} - \Sigma_{\tau m}\|_2$ since $\Sigma^*_{\tau m} \geq \sigma^2_{\Sigma} I_d$.

Since $K^*$ is $(\bar{\kappa}, \frac{1}{\alpha \tau^2})$ for $(A_{\tau m}, B_{\tau m})$, by Lemma ??, we know $\|\Sigma^*_{\tau m} - \Sigma^*_{(\tau-1)m}\|_2 \leq 256 \bar{\kappa}^{11} mC$. Thanks to the choice of $\alpha'$ in $\eta_\tau$, we have $\eta_{\tau-1} - \alpha/4 \leq \alpha/4$, and due to the condition on $T$, we have

$$\left\| \Sigma^*_{\tau m} - \Sigma^*_{(\tau-1)m} \right\|_2 \leq \frac{\alpha \sigma_{\Sigma}^2}{4} \leq \frac{\alpha \sigma_{\Sigma}^2}{4} \leq 0$$

Therefore, we obtain the following upper bound on the expected regret:

$$\text{Expected Regret} \lesssim h^{-1}_{\max} \|K_0 - K^*\|^2_{\Sigma_0} + \frac{G^2}{\alpha'} \log(\bar{T}) + \frac{\alpha}{2} \sum_{\tau=0}^{\tilde{T}} \|K_{\tau} - K^*\|^2_{\Sigma_{\tau m}}$$

On the other hand, with details deferred to the Appendix, we bound the deviation by a standard martingale argument to obtain Deviation $\lesssim \frac{G}{2} \sum_{\tau=0}^{\tilde{T}} \|K_{\tau} - K^*\|^2_{\Sigma^*_{\tau m}}$.

Therefore, we conclude with high probability the regret within one cycle is bounded as follows:

$$\text{Expected Regret} + \text{Deviation} \lesssim h^{-1}_{\max} \|K_0 - K^*\|^2_{\Sigma_0} + \frac{G^2}{\alpha'} \log(\bar{T}) \leq \kappa^6 h^{-1}_{\max} + \frac{G^2}{\alpha'} \log(\bar{T})$$

Since the algorithm has $\frac{T}{mT} \leq \alpha' h_{\min}$ many cycles, multiplying the above quantity by $\alpha' h_{\min}$ gives the total regret.

#### 8 Conclusion

In this paper, we propose a preconditioned policy gradient update for the online LQR problem with unknown, time-varying systems. This algorithm achieves $O(\sqrt{T})$ regret against strongly stable piecewise-linear policies with constantly many segments when the system parameters shift by $O(T^{-1})$ every time step. As a future direction, it would be nice to be able to compare against the optimal sequence of actions in hindsight as well as to handle a faster rate of system shift.
References


