Proceedings of Machine Learning Research vol 120:1–10, 2020

Riccati updates for online linear quadratic control

Mohammad Akbari 13MAV1@QUEENSU.CA
Bahman Gharesifard BAHMAN.GHARESIFARD@QUEENSU.CA
Tamas Linder TAMAS.LINDER@QUEENSU.CA
Department of Mathematics and Statistics, Queen's University

Editors: A. Bayen, A. Jadbabaie, G. J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M. Zeilinger

Abstract

We study an online setting of the linear quadratic Gaussian optimal control problem on a sequence of cost functions, where similar to classical online optimization, the future decisions are made by only knowing the cost in hindsight. We introduce a modified online Riccati update that under some boundedness assumptions, leads to logarithmic regret bounds. In particular, for the scalar case we achieve the logarithmic regret without any boundedness assumption. As opposed to earlier work, the proposed method does not rely on solving semi-definite programs at each stage.[1]

Keywords: Online Learning, Regret Minimization, Linear Quadratic Gaussian Optimal Control

1. Introduction

The problem of prediction and decision making has many applications in engineering, economy and social sciences, for instance, portfolio selection (Agarwal et al., 2006; Luo et al., 2018), transportation and traffic control (Patel and Ranganathan, 2001), power engineering (Zhai et al., 2016), manufacturing and supply chain management; and it has received substantial attention in recent years, see (Anava et al., 2013), (Ross et al., 2011), and (Cesa-Bianchi and Lugosi, 2006). The subject we study in this work fits into the general theme of decision making problems, and particularly is related to online optimization. The literature on online optimization is extremely rich and its connections to many other areas of learning has been explored in recent years (Cesa-Bianchi and Lugosi, 2006; Hazan, 2016; Shalev-Shwartz, 2012; Hazan et al., 2007; Hazan and Kale, 2014; Gofer et al., 2013; Blum and Mansour, 2007).

Unlike the classical setting of online optimization, where the decisions of the learner are solely chosen according to a cost function, in many realistic scenarios the learner's decisions are inputs to a *control system*. Examples include power supply management in the presence of time-varying energy costs due to demand fluctuations and tracking of an adversarial target. In such scenarios, decisions are usually assumed to be a function of the current state which is referred to as a *policy*. As usual, the regret is defined as the difference between the accumulated costs incurred by control actions made in hindsight using previous states and the cost incurred by the best fixed admissible policy when all the cost functions are known in advance. Similar to online optimization, the objective is to design algorithms to generate policies which make the regret function grow sublinearly. Of course, the online optimization problem discussed above would reduce to the classical optimal control problem if the cost functions were available to the decision maker. Our work is closely related to the recent work of Cohen et al. (2018) where an online version of linear quadratic Gaussian

1. An extended version of this paper can be found on arXiv as (Akbari et al., 2019)

control is studied. In particular, an online gradient descent algorithm with a fixed learning rate is proposed, where in each iteration, a projection onto a bounded set of positive-definite matrices is taken, which itself relies on solving a semi-definite program. Under the assumptions that the underlying system is controllable, the cost functions are bounded, and the covariance of the disturbance is positive definite, it is proved that the regret is sublinear, and grows as $\mathcal{O}(\sqrt{T})$, where T is the time horizon. Other closely related works are (Agarwal et al., 2019) and (Agarwal et al., 2019), where the cost functions are assume to be general convex and globally Lipschitz functions. In contrast to (Cohen et al., 2018), the noise assumed in (Agarwal et al., 2019) is adversarial, and (Agarwal et al., 2019) achieves a regret bound of $\mathcal{O}((\log(T))^7)$. In these works, the generated control actions, which lead to a sublinear regret bound, are linear feedbacks which rely on a finite history of the past disturbances.

Before we state our contributions, we point out a wider set of literature related to our work. First, we note that one can think about the underlying control system as a dynamical constraint on the optimization problem. Considering control systems as constraints is also classical in the context of *model predictive control* (Garcia et al., 1989). Although we tackle dynamic constraints in this work, we should emphasize that online optimization problems with static constraints, known only in hindsight, also play a key role in various settings and have generated interest in recent years (Yu et al., 2017; Neely and Yu, 2017; Jenatton et al., 2016).

Our work is also related to the framework of Markov decision processes (MDPs), where the system transition to the next state is defined through a probability distribution. Moreover, a reward is given to the decision maker for each action at each state. This framework is classical in *reinforcement learning*, where the objective is to learn the optimal policy which yields the maximum reward (Sutton and Barto, 2018). It is also worth pointing out that there is another key role that regret minimization has played recently, bringing learning and control theory together, in the context of robust control, adaptive control, and system identification. Here, the regret enters through the lack of perfect knowledge of the model, and research efforts focus on generating algorithms for updating models in a data-driven fashion (Yang et al., 2019; Karimi and Kammer, 2017). Finally, our setting is also related to online optimization in dynamic environments (Hall and Willett, 2015), where the decisions are constrained in dynamics chosen by the environment. However, the objective of (Hall and Willett, 2015) is to study the impact of model mismatch on the overall regret, whereas in this paper the decisions are input to a control system, which impacts the way the decisions affect the future outcomes through its dynamics.

Contributions. We consider the online linear quadratic Gaussian optimal control problem, where the cost function only becomes available in hindsight. In contrast to (Cohen et al., 2018), where an online algorithm using semi-definite programming update is employed to generate the control inputs, we take a control-theoretic approach and employ a modified version of the classical Riccati update, using averaged past data, to generate control policies. Our main result is a $\mathcal{O}(\log T)$ regret bound for the online linear quadratic Gaussian optimal control problem, improving $\mathcal{O}(\sqrt{T})$ bound of Cohen et al. (2018) and $\mathcal{O}((\log(T))^7)$ of Agarwal et al. (2019) for time horizon T, under some boundedness assumption. The technical part of our result relies on characterizing the interplay between a notion of stability for the sequence of control policies and boundedness of the solutions of the proposed Riccati update; in particular, for the scalar case, we prove a stronger result that initializing the control policy to be stable is enough to guarantee boundedness of the solutions of the proposed online Riccati update.

Notation. Throughout the paper, we use the following mathematical notion. Let $\mathbb{R}$ denote the set of real numbers. We use lowercase letters for vectors and uppercase letters for matrices. We

denote by $\|\cdot\|$ the Euclidean norm on vectors and its corresponding operator norm on real matrices. We denote by $A^\top$ the transpose of matrix A. Thus $\|A\| = \sigma_{\max}(A) = \sqrt{\lambda_{\max}(A^\top A)}$, where $\sigma_{\max}(A)$ is the largest singular value of A and $\lambda_{\max}(A^\top A)$ is the largest eigenvalue of $A^\top A$. Trace of matrix A is denoted by $\mathrm{Tr}(A)$. If A is an $n \times n$ real matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$, then the spectral radius of $\rho(A)$ of A is $\rho(A) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$. We use $A \succeq B$ to indicate that $A - B$ is positive semi-definite.

2. Problem Formulation

We start by describing the general problem of online optimization in control systems. We focus on a special class of control systems where the system dynamics are linear and the cost functions are quadratic. Let us recall this setting.

2.1. Discrete-Time Linear Quadratic Gaussian Control

The discrete-time linear quadratic Gaussian (LQG) control problem is defined as follows, see for instance (Soderstrom, 2002): Let $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ be the control state and the control action at time t, respectively, with initial state x_1. The system dynamics are given by

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

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $\{w_t\}_{t\geq 1}$ are i.i.d. Gaussian noise vectors with zero mean and covariance $W \in \mathbb{R}^{n\times n}$ ($w_t \sim \mathcal{N}(0, W)$). It is assumed that the initial value is Gaussian $x_1 \sim \mathcal{N}(m, X_1)$ and is independent of the noise sequence $\{w_t\}_{t\geq 1}$. The cost incurred in each time step t is a quadratic function of the state and control action given by $x_t^\top Q_t x_t + u_t^\top R_t u_t$, where $Q_t \in \mathbb{R}^{n\times n}$ and $R_t \in \mathbb{R}^{m\times m}$ are positive-definite matrices. The total cost after T time steps is given by

$$J_T(x_1, u_1, \ldots, u_T) = \mathbb{E}\Big[x_T^\top Q_T x_T + \sum_{t=1}^{T-1} \big(x_t^\top Q_t x_t + u_t^\top R_t u_t\big)\Big].$$

We consider controllers of the form $u_t = \pi_t(x_t)$, where the function $\pi_t : \mathbb{R}^n \to \mathbb{R}^m$ is called a policy. This assumption does not restrict generality, as the optimal policy will provably be of this form (Soderstrom, 2002). It is well-known that under the assumption that the control system is stabilizable, and the cost matrices Q_t and R_t are positive-definite, the optimal policy is a stable linear feedback of the state, which will be described in Section 3.

2.2. Problem Setting

We now define the problem we study in this work, following (Cohen et al., 2018). In *online linear quadratic control*, the sequence of cost matrices $\{Q_t\}_{t\geq 1}$ and $\{R_t\}_{t\geq 1}$ are not known in advance and Q_t and R_t are only revealed after choosing the control action u_t. Since it is not possible to find the optimal policy before observing the whole sequence of cost matrices $\{Q_t\}_{t\geq 1}$ and $\{R_t\}_{t\geq 1}$, the decision maker faces a *regret*. Here, we assume that the control system (A, B) is stabilizable, and the cost matrices Q_t and R_t are positive-definite and uniformly bounded over $t \geq 1$. As the optimal policy for the system with these assumptions is given by a stable linear feedback, we use the set of stable linear feedback functions as the set of admissible policies. This setting is formally presented next.

Let $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ be the control state and controller action at time $t \geq 1$. The controller uses a linear feedback policy $u_t = -K_t x_t$ and commits to this action after observing x_t. Then the controller receives the positive-definite matrices $Q_t \in \mathbb{R}^{n\times n}$ and $R_t \in \mathbb{R}^{m\times m}$, and suffers the cost

$$J_t(K_t) = \mathbb{E}\Big[x_t^\top Q_t x_t + u_t^\top R_t u_t\Big]. \tag{2}$$

The objective is to design an algorithm to generate a sequence of policies $\{K_t\}_{t\geq 1}$ such that the regret function, which is defined as

$$\mathcal{R}(T) = \sum_{t=1}^{T} J_t(K_t) - \min_{K\in\mathcal{K}} \sum_{t=1}^{T} J_t(K),$$

where $\mathcal{K}$ is the set of stable policies, grows sublinearly in T. In other words, the average regret over time converges to zero. Before stating our main results, we provide a brief review of the iterative Riccati updates that we employ to design our main algorithm.

3. Iterative Methods for Solving the Discrete Algebraic Riccati Equation

In the classical LQG problem, where all the cost functions are known, the optimal policy can be obtained by dynamic programming, and is a linear function of the state. In particular, $u_t = -K_t x_t$, where K_t is given by the equation

$$K_t = (B^\top P_{t+1} B + R_t)^{-1} B^\top P_{t+1} A,$$

and P_{t+1} is a sequence of positive-definite matrices obtained iteratively, backwards in time, from the dynamic Riccati equation:

$$P_t = A^\top P_{t+1} A - A^\top P_{t+1} B (B^\top P_{t+1} B + R_t)^{-1} B^\top P_{t+1} A + Q_t \tag{3}$$

with the terminal condition $P_T = Q_T$.

For the infinite-horizon problem with the assumption that $Q_t = Q$ and $R_t = R$ are fixed, and under the assumptions that

1. R is positive-definite
2. the pair (A, B) is stabilizable, i.e., there exists a linear policy $\pi(x) = -Kx$ such that the closed-loop system $x_{t+1} = (A - BK)x_t$ is asymptotically stable: $\rho(A - BK) < 1$,
3. the pair (A, C), where $Q = C^\top C$, is detectable [i.e., if $u_t \to 0$ and $Cx_t \to 0$ then, $x_t \to 0$],

it is well-known that the optimal policy is unique, time invariant, and is a linear function of the state (Bertsekas, 2018), i.e., $u_t = -K^\star x_t$. Here $K^\star$ is given by

$$K^\star = (B^\top P^\star B + R)^{-1} B^\top P^\star A, \tag{4}$$

where $P^\star$ satisfies the discrete algebraic Riccati equation (DARE):

$$P^\star = A^\top P^\star A - A^\top P^\star B (B^\top P^\star B + R)^{-1} B^\top P^\star A + Q. \tag{5}$$

Moreover, P_t given by (3) converges to $P^\star$ as $t \to \infty$ (Soderstrom, 2002). By using the policy $K^\star$, we have that $x_{t+1} = (A - BK^\star)x_t + w_t$. The optimal policy $K^\star$ is guaranteed to be stable i.e. $\rho(A - BK^\star) < 1$. Here, x_t converges to a stationary distribution, i.e., x_t converges weakly to a random variable x which has the same distribution as $(A - BK^\star)x + w_t$, so that we have $\mathbb{E}[x] = \mathbb{E}[(A - BK^\star)x + w_t]$, which implies $\mathbb{E}[x] = 0$, and the covariance matrix $X = \mathbb{E}[xx^\top]$ satisfies $X = (A - BK^\star)X(A - BK^\star)^\top + W$, see e.g., (Cohen et al., 2018).

Several methods for solving DARE exist in the literature, including iterative methods (Caines and Mayne, 1970), algebraic methods (Rodman and Lancaster, 1995), and semi-definite programming (Balakrishnan and Vandenberghe, 2003). Our work is based on iterative methods, and in particular, we use the technique studied by Hewer (1971). In what follows, we modify this technique and use it for the online linear quadratic Gaussian problem. We present our algorithm after reviewing some salient properties of stable policies. Similar to (Cohen et al., 2018), we use the notion of *strong stability*, which allows us to analyze the rate of convergence of the state covariance matrices under our proposed algorithm.

4. Strong Stability

A key property that we require before introducing our algorithm is the notion of strong stability and sequential strong stability which are defined in (Cohen et al., 2018). The notion of strong stability is defined as follows.

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

Note that every (κ, γ)-strongly stable policy K is stable, since the matrices $A - BK$ and L are similar and hence $\rho(A - BK) = \rho(L) \leq (1 - \gamma)$. (Cohen et al., 2018, Lemma B.1), shows that every stable policy is (κ, γ)-strongly stable for some $\kappa > 0$ and $0 < \gamma \leq 1$. Under the assumption of (κ, γ)-strong stability of policy K, the state covariance matrices $X_t = \mathbb{E}[x_t x_t^\top]$ converge exponentially to a steady-state covariance matrix $\widehat{X}$, which satisfies

$$\widehat{X} = (A - BK)\widehat{X}(A - BK)^\top + W;$$

See (Akbari et al., 2019, Lemma A.2) for details. In order to obtain a similar result for the change of the state covariance matrices using a sequence of different (κ, γ)-strongly stable policies $\{K_t\}_{t\geq 1}$, we need to define a notion of sequential strong stability, which is presented next.

Definition 2 *A sequence of policies $\{K_t\}_{t\geq 1}$ is sequentially (κ, γ)-strongly stable, for $\kappa > 0$ and $0 < \gamma \leq 1$, if there exist sequences of matrices $\{H_t\}_{t\geq 1}$ and $\{L_t\}_{t\geq 1}$ such that*

$$A - BK_t = H_t L_t H_t^{-1}$$

for all $t \geq 1$, with the following properties:

- $\|L_t\| \leq 1 - \gamma$ *and* $\|K_t\| \leq \kappa$;
- $\|H_t\| \leq \beta$ *and* $\|H_t^{-1}\| \leq 1/\alpha$ *with* $\kappa = \beta/\alpha$ *and* $\alpha > 0$ *and* $\beta > 0$;
- $\|H_{t+1}^{-1} H_t\| \leq 1 + \gamma$.

We now proceed with some key results that we later use to ensure strong stability for the sequence of policies generated. Suppose that a sequence of positive-definite matrices P_t is generated recursively as

$$P_t = (A - BK_t)^\top P_t (A - BK_t) + \bar{Q}_t + K_t^\top \bar{R}_t K_t, \tag{6}$$

where

$$K_{t+1} = (B^\top P_t B + \bar{R}_t)^{-1} B^\top P_t A \tag{7}$$

and where $\bar{R}_t \in \mathbb{R}^{m\times m}$ and $\bar{Q}_t \in \mathbb{R}^{n\times n}$ are given positive-definite matrices for all $t \geq 1$, and K_1 is an initial stable policy. The reason for this update will become clear as part of our algorithm in Section 5. The key point we wish to make here is that under the assumption of uniform boundedness of the matrix sequence $\{P_t\}_{t\geq 1}$, and the stability of matrix K_t, for all $t \geq 1$, the sequence $\{K_t\}_{t\geq 1}$ is uniformly (κ, γ)-strongly stable, with appropriate choices of κ and γ.

Proposition 3 *Assume that for $t \geq 1$, $Q_t, R_t \succeq \mu I$ and $P_t \preceq \nu I$, where $\mu, \nu > 0$ and $\{P_t\}_{t\geq 1}$ is the sequence of matrices obtained as the solution of (6), and assume that the policy K_t given by (7) is stable for $t \geq 1$. Define $\bar{\kappa} = \sqrt{\frac{\nu}{\mu}}$. Then the sequence $\{K_t\}_{t\geq 1}$ is uniformly $(\bar{\kappa}, 1/2\bar{\kappa}^2)$-strongly stable.*

We refer to (Akbari et al., 2019) for a proof of this result. We now present a second useful result, where we show that under the additional property that the rate of changes of sequence P_t is small (which we will be able to establish for our proposed algorithm), one can obtain that the sequence $\{K_t\}_{t\geq 1}$ is sequentially strongly stable; the proof can again be found in (Akbari et al., 2019).

Proposition 4 *Assume that for $t \geq 1$, $Q_t, R_t \succeq \mu I$ and $P_t \preceq \nu I$, where $\mu, \nu > 0$ and $\{P_t\}_{t\geq 1}$ is the sequence of matrices obtained as the solution of (6), and assume that the policy K_t given by (7) is stable for $t \geq 1$. Let $\bar{\kappa} = \sqrt{\frac{\nu}{\mu}}$, and suppose that $\|P_{t+1} - P_t\| \leq \eta$ for all $t \geq 1$ for some $\eta \leq \mu/\bar{\kappa}^2$. Then the sequence $\{K_t\}_{t\geq 1}$ is sequentially $(\bar{\kappa}, 1/2\bar{\kappa}^2)$-strongly stable.*

Note the above results rely on uniform boundedness of the sequence $\{P_t\}_{t\geq 1}$, which we assume throughout the paper. However, we can show that stability of K_1 is enough to guarantee this property in the scalar case, see (Akbari et al., 2019, Proposition A.4). We believe that this property should hold only by assuming stability of K_1 for the general case, but have not been able to prove this.

5. The Online Riccati Algorithm

We outline our main algorithm in this section. Our assumptions are as follows:

Assumption 5.1 *Throughout we assume that*

- *The pair (A, B) is stabilizable.*
- *The cost matrices Q_t and R_t are positive-definite and $\mu I \preceq Q_t$, $\mu I \preceq R_t$, and $\mathrm{Tr}(Q_t) \leq \sigma$, $\mathrm{Tr}(R_t) \leq \sigma$, for some $\sigma > \mu > 0$ for all $t \geq 1$.*
- *For the noise covariance matrix W we have that $\omega = \mathrm{Tr}(W) < \infty$.*

We first provide an informal description of the algorithm; a formal description is given in Algorithm 1. We start from a stable policy K_1; the existence of K_1 is provided by the assumption of stabilizability of the control system. At each time step $t \geq 1$, the controller uses the policy $u_t = -K_t x_t$ after observing x_t, then the cost matrices Q_t and R_t are revealed, and the controller updates P_t and K_t using the average of the history of Q_ts and R_ts through (6) and (7). There is a technical step in our algorithm, which we call the "reset" step. This step allows us to show that using these updates the change of the norm of the policies is $\mathcal{O}(1/t)$, and this gives a regret bound $\mathcal{O}(\log(T))$. Before we state the algorithm, we need to elaborate on the parameters used.

Remark 5 (Parameters used in Algorithm 1) Our algorithm naturally uses parameters μ and σ, stated in Assumption 5.1. For the reset step, we also need (an estimate on) the strong stability parameters κ and γ, which are defined in Algorithm 1. Proposition 3 plays a key role in that regard, as it states that as long as we can estimate a uniform bound on the sequence P_t, we can obtain these parameters. In the scalar case, we know this uniform bound (Akbari et al., 2019); in other cases, given that the parameters are not needed in the early steps of the algorithm, one can envision that we can run our algorithm with a large estimate on this bound and adjust it if necessary. Extending the uniform boundedness of the sequence P_t to vector cases, which is an avenue of our current research, will remove this restriction all together.

Algorithm 1 Online Riccati Update

Input: The system matrices A and B, initial state x_1, time horizon T, parameters $\nu, \mu, \kappa = \sqrt{\nu/\mu}, \gamma = 1/(2\kappa^2), \sigma$

Output: A sequence of stable policies $\{K_t\}_{t=1}^T$

Initialize K_1 to be stable
for each $t = 1, 2, \cdots, T$:
  receive x_t
  use controller $u_t = -K_t x_t$ and receive Q_t and R_t
  update $\bar{R}_t = \frac{t-1}{t}\bar{R}_{t-1} + \frac{1}{t}R_t$, $\bar{Q}_t = \frac{t-1}{t}\bar{Q}_{t-1} + \frac{1}{t}Q_t$
  update P_t as the solution of

$$P_t = (A - BK_t)^\top P_t (A - BK_t) + \bar{Q}_t + K_t^\top \bar{R}_t K_t$$

  Reset:
  if $t = t^\star := \left\lceil \frac{4\kappa^3\|B\|}{\gamma\mu}(2\sigma\kappa + \frac{2\kappa^3\|B\|\sigma(1+\kappa^2)}{\gamma}) + 1 \right\rceil$:
    Initialize $\ell = 0$, $\widehat{P}_0 = P_{t^\star}$, and $\widehat{K}_0 = K_{t^\star}$
    while $\|\widehat{P}_\ell - \widehat{P}_{\ell-1}\| > (\frac{2\sigma}{\|B\|} + \frac{4\kappa^2\sigma(1+\kappa^2)}{\gamma})/t^\star$:
      $\ell \leftarrow \ell + 1$
      $\widehat{K}_\ell = (B^\top \widehat{P}_{\ell-1} B + \bar{R}_{t^\star})^{-1} B^\top \widehat{P}_{\ell-1} A$
      $\widehat{P}_\ell$ satisfies $\widehat{P}_\ell = (A - B\widehat{K}_\ell)^\top \widehat{P}_\ell (A - B\widehat{K}_\ell) + \bar{Q}_{t^\star} + \widehat{K}_\ell^\top \bar{R}_{t^\star} \widehat{K}_\ell$
    return $P_{t^\star} = \widehat{P}_\ell$
  return $K_{t+1} = (B^\top P_t B + \bar{R}_t)^{-1} B^\top P_t A$

6. Main Results

We are now in a position to state our main contribution.

Theorem 6 *Suppose that the tuple* $(A, B, \{Q_t\}_{t=1}^T, \{R_t\}_{t=1}^T, W)$ *satisfies Assumption 5.1. Suppose that the matrices* P_t *generated by Algorithm 1 are uniformly bounded. Then we have that*

$$\mathcal{R}(T) = \mathcal{O}(\log(T)).$$

Proof [Outline of the proof of Theorem 6] The proof is provided in (Akbari et al., 2019), and is quite involved. We provide a brief outline here. Our first technical result (Akbari et al., 2019, Lemma A.5) shows that Algorithm 1, as long as it is initialized at an stable policy, iteratively produces stable polices. This step is analogous to the classical result of Hewer (1971) for the case where the cost objective matrices Q_t and R_t are fixed. Recall that, by Proposition 3, stability of policies K_t is required to establish strong stability. A technical part of this proof demonstrates the reason why we need the reset step of the algorithm to ensure that the sequence of policies $\{P_{t+1} - P_t\}$ decay as m/t, for some $m > 0$. Using this and by rewriting the regret using trace products, we establish a set of bounds (Akbari et al., 2019, Lemmas A.8, A.9, A.10) which eventually yield the result. ■

Note that the assumption of (κ, γ)-strongly stability in Theorem 6 will be satisfied as long as the solutions to the online Riccati equation are uniformly bounded. In particular, we do not need this assumption for the scalar case, see (Akbari et al., 2019, Proposition A.4).

References

A. Agarwal, E. Hazan, S. Kale, and R. E. Schapire. Algorithms for portfolio management based on the Newton method. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 9–16, 2006.

N. Agarwal, B. Bullins, E. Hazan, S. Kakade, and K. Singh. Online control with adversarial disturbances. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 111–119, 2019.

N. Agarwal, E. Hazan, and K. Singh. Logarithmic regret for online control. In *Advances in Neural Information Processing Systems 32*, pages 10175–10184. 2019.

M. Akbari, B. Gharesifard, and T. Linder. An iterative Riccati algorithm for online linear quadratic control. *arXiv preprint arXiv:1912.09451*, 2019.

O. Anava, E. Hazan, S. Mannor, and O. Shamir. Online learning for time series prediction. In *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30, pages 172–184, 2013.

V. Balakrishnan and L. Vandenberghe. Semidefinite programming duality and linear time-invariant systems. *IEEE Transactions on Automatic Control*, volume 48, pages 30–41, 2003.

D. P. Bertsekas. Stable optimal control and semicontractive dynamic programming. *SIAM Journal on Control and Optimization*, volume 56, pages 231–252, 2018.

A. Blum and Y. Mansour. From external to internal regret. *Journal of Machine Learning Research*, volume 8, pages 1307–1324, 2007.

P. E. Caines and D. Q. Mayne. On the discrete time matrix Riccati equation of optimal control. *International Journal of Control*, volume 12, pages 785–794, 1970.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006. ISBN 0-521-84108-9.

A. Cohen, A. Hasidim, T. Koren, N. Lazic, Y. Mansour, and K. Talwar. Online linear quadratic control. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1029–1038, 2018.

C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: Theory and practice; a survey. *Automatica*, volume 25, pages 335–348, 1989. ISSN 0005-1098.

E. Gofer, N. Cesa-Bianchi, C. Gentile, and Y. Mansour. Regret minimization for branching experts. In *Conference on Learning Theory*, pages 618–638, 2013.

E. C. Hall and R. M. Willett. Online convex optimization in dynamic environments. *IEEE Journal of Selected Topics in Signal Processing*, volume 9, pages 647–662, 2015.

E. Hazan. Introduction to online convex optimization. *Foundation and Trends in Optimization*, volume 2, pages 157–325, 2016.

E. Hazan and S. Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research*, volume 15, pages 2489–2512, 2014.

E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, volume 69, pages 169–192, 2007.

G. Hewer. An iterative technique for the computation of the steady state gains for the discrete optimal regulator. *IEEE Transactions on Automatic Control*, volume 16, pages 382–384, 1971.

R. Jenatton, J. Huang, and C. Archambeau. Adaptive algorithms for online convex optimization with long-term constraints. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 402–411, 2016.

A. Karimi and C. Kammer. A data-driven approach to robust control of multivariable systems by convex optimization. *Automatica*, volume 85, pages 227 – 233, 2017.

H. Luo, C. Wei, and K. Zheng. Efficient online portfolio with logarithmic regret. In *Advances in Neural Information Processing Systems 31*, pages 8235–8245. Curran Associates, Inc., 2018.

M. J. Neely and H. Yu. Online convex optimization with time-varying constraints. *arXiv preprint arXiv:1702.04783*, 2017.

M. Patel and N. Ranganathan. IDUTC: an intelligent decision-making system for urban traffic-control applications. *IEEE Transactions on Vehicular Technology*, volume 50, pages 816–829, 2001.

L. Rodman and P. Lancaster. *Algebraic Riccati Equations*. Oxford Mathematical Monographs. 1995.

S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2011.

S. Shalev-Shwartz. *Online Learning and Online Convex Optimization*, volume 12 of *Foundations and Trends in Machine Learning*. Now Publishers Inc, 2012. ISBN 1601985460.

T. Soderstrom. *Discrete-Time Stochastic Systems: Estimation and Control*. Springer-Verlag, 2nd edition, 2002. ISBN 1852336498.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

Y. Yang, Z. Guo, H. Xiong, D. Ding, Y. Yin, and D. C. Wunsch. Data-driven robust control of discrete-time uncertain linear systems via off-policy reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

H. Yu, M. Neely, and X. Wei. Online convex optimization with stochastic constraints. In *Advances in Neural Information Processing Systems 30*, pages 1428–1438. 2017.

J. Zhai, Y. Li, and H. Chen. An online optimization for dynamic power management. In *2016 IEEE International Conference on Industrial Technology (ICIT)*, pages 1533–1538, 2016.