# Logarithmic Regret for Adversarial Online Control

**Dylan J. Foster** [1] **Max Simchowitz** [2]

## Abstract

We introduce a new algorithm for online linear-quadratic control in a known system subject to adversarial disturbances. Existing regret bounds for this setting scale as $\sqrt{T}$ unless strong stochastic assumptions are imposed on the disturbance process. We give the first algorithm with logarithmic regret for arbitrary adversarial disturbance sequences, provided the state and control costs are given by known quadratic functions. Our algorithm and analysis use a characterization for the optimal offline control law to reduce the online control problem to (delayed) online learning with approximate *advantage functions*. Compared to previous techniques, our approach does not need to control movement costs for the iterates, leading to logarithmic regret.

## 1. Introduction

Reinforcement learning and control consider the behavior of an agent making decisions in a dynamic environment in order to suffer minimal loss. In light of recent practical breakthroughs in data-driven approaches to continuous RL and control (Lillicrap et al., 2016; Mnih et al., 2015; Silver et al., 2017), there is great interest in applying these techniques in real-world decision making applications. However, to reliably deploy data-driven RL and control in physical systems such as self-driving cars, it is critical to develop principled algorithms with provable safety and robustness guarantees. At the same time, algorithms should not be overly pessimistic, and should be able to take advantage of benign environments whenever possible.

In this paper we develop algorithms for online linear-quadratic control which ensure robust worst-case performance while optimally adapting to the environment at hand. Linear control has traditionally been studied in settings where the dynamics of the environment are either governed by a well-behaved stochastic process or driven by a worst-case process to which the learner must remain robust in the $\mathcal{H}_\infty$ sense. We consider an intermediate approach introduced by Agarwal et al. (2019a) in which disturbances are non-stochastic but performance is evaluated in terms of *regret*. This benchmark forces the learner's control policy to achieve near optimal performance on any specific disturbance process encountered.

Concretely, we consider a setting in which the state evolves according to linear dynamics:

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

where $x_t \in \mathbb{R}^{d_{\mathsf{x}}}$ are states, $u_t \in \mathbb{R}^{d_{\mathsf{u}}}$ are inputs, and $A \in \mathbb{R}^{d_{\mathsf{x}} \times d_{\mathsf{x}}}$ and $B \in \mathbb{R}^{d_{\mathsf{x}} \times d_{\mathsf{u}}}$ are system matrices known to the learner. We refer to $w_t \in \mathbb{R}^{d_{\mathsf{x}}}$ as the *disturbance* (or, "noise"), which we assume is selected by an *adaptive* adversary and satisfies $\|w_t\| \le 1$; we let $\boldsymbol{w}$ refer to the entire sequence $w_{1:T}$. We consider fixed quadratic costs of the form $\ell(x,u) := x^\top R_x x + u^\top R_u u$, where $R_x, R_u \succeq 0$ are given. This model encompasses noise which is uncorrelated ($\mathcal{H}_2$), worst-case ($\mathcal{H}_\infty$), or governed by some non-stationary stochastic process. The model also approximates control techniques such as feedback linearization and trajectory tracking (Slotine & Li, 1991), where $A$ and $B$ are the result of linearizing a known nonlinear system and the disturbances arise due to systematic errors in linearization rather than from a benign noise process.

For any policy $\pi$ that selects controls based on the current state and disturbances observed so far, we measure its performance over a time horizon $T$ by

$$J_T(\pi; \boldsymbol{w}) = \sum_{t=1}^{T} \ell(x_t^\pi, u_t^\pi),$$

the total cost incurred by following $u_t = \pi_t(x_t, w_{1:t-1})$. Letting $\pi^K$ denote a state-feedback control law of the form $\pi_t^K(x) = -Kx$ for all $t$, the learning algorithm's goal is to minimize

$$\mathrm{Reg}_T = J_T(\pi^{\mathrm{alg}}; \boldsymbol{w}) - \inf_{K \in \mathcal{K}} J_T(\pi^K; \boldsymbol{w}),$$

where $\pi^{\mathrm{alg}}$ denotes the learner's policy and $\mathcal{K}$ is an appropriately defined set of stabilizing controllers. Thus, $\pi^{\mathrm{alg}}$ has low regret when its performance nearly matches the optimal

[1] Massachusetts Institute of Technology [2] UC Berkeley. Correspondence to: Dylan Foster <dylanf@mit.edu>.

*Proceedings of the 37th International Conference on Machine Learning*, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

controller $K \in \mathcal{K}$ on the specific, realized noise sequence. While the class $\mathcal{K}$ contains the optimal $\mathcal{H}_\infty$ and $\mathcal{H}_2$ control policies, we also develop algorithms to compete with a more general class of stabilizing linear controllers, which may fare better for certain noise sequences (Appendix B).

**Logarithmic regret in online control.** Agarwal et al. (2019a) introduced the adversarial LQR setting we study and provided an efficient algorithm with $\sqrt{T}$-regret. Subsequent works (Agarwal et al., 2019b; Simchowitz et al., 2020) have shown that logarithmic regret is possible when the disturbances follow a semi-adversarial process with persistent excitation. Our main result is to achieve logarithmic regret for fully adversarial disturbances, provided that costs are known and quadratic.

### 1.1. Contributions

We introduce Riccatitron (Algorithm 1), a new algorithm for online linear control with adversarial disturbances which attains polylogarithmic regret.

**Theorem 1 (informal).** Riccatitron *attains regret* $\mathcal{O}(\log^3 T)$*, where* $\mathcal{O}$ *hides factors polynomial in relevant problem parameters.*

Riccatitron has comparable computational efficiency to previous methods. We show in Appendix B that the algorithm also extends to a more general benchmark class of linear controllers with internal state, and to "tracking" loss functions of the form $\ell_t(x,u) := \ell(x - a_t, u - b_t)$. Some conceptual contributions are as follows.

**When is logarithmic regret possible in online control?** Simchowitz & Foster (2020) and Cassel et al. (2020) independently show that logarithmic regret is impossible in a minimax sense if the system matrices $(A, B)$ are unknown, even when disturbances are i.i.d. gaussian. Conversely, our result shows that if $A$ and $B$ are known, logarithmic regret is possible even when disturbances are adversarial. Together, these results paint a clear picture of when logarithmic regret is achievable in online linear control. We note, however, that our approach heavily leverages the structure of linear control with *strongly convex, quadratic* costs. We refer the to the related work section for discussion of further structural assumptions that facilitate logarithmic regret.

**Addressing trajectory mismatch.** Riccatitron represents a new approach to a problem we call *trajectory mismatch* that arises when considering policy regret in online learning problems with state. In dynamic environments, different policies inevitably visit different state trajectories. Low-regret algorithms must address the mismatch between the performance of the learner's policy $\pi^{\mathrm{alg}}$ on its own realized trajectory and the performance of each benchmark policy $\pi$ on the alternative trajectories it induces. Most algorithms with policy regret guarantees (Even-Dar et al., 2009; Zimin & Neu, 2013; Abbasi-Yadkori et al., 2013; Arora et al., 2012; Anava et al., 2015; Abbasi-Yadkori et al., 2014; Cohen et al., 2018; Agarwal et al., 2019a;b; Simchowitz et al., 2020) adopt an approach to addressing this trajectory mismatch that we refer to as "online learning with stationary costs", or OLwS. At each round $t$, the learner's adaptive policy $\pi^{\mathrm{alg}}$ commits to a policy $\pi^{(t)}$, typically from a benchmark class $\Pi$. The goal is to ensure that the iterates $\pi^{(t)}$ attain low regret on a proxy sequence of *stationary* cost functions $\pi \mapsto \lambda_t(\pi)$ that describe the loss the learner would suffer at stage $t$ under the *fictional* trajectory that would arise if she had played the policy $\pi$ at all stages up to time $t$ (or in some cases, on the corresponding steady-state trajectory as $t \to \infty$). Since the stationary cost does not depend on the learner's state, low regret on the sequence $\{\lambda_t\}$ can be obtained by feeding these losses directly into a standard online learning algorithm. To relate regret on the proxy sequence back to regret on the true sequence, most approaches use that the iterates produced by the online learner are sufficiently slow-moving.

The main technical challenge Riccatitron overcomes is that for the stationary costs that arise in our setting, no known algorithm produces iterates which move sufficiently slowly to yield logarithmic regret via OLwS (Appendix C.4). We adopt a new approach for online control we call *online learning with advantages*, or OLwA, which abandons stationary costs, and instead considers the control-theoretic advantages of actions relative to the unconstrained offline optimal policy $\pi^\star$. Somewhat miraculously, we find that these advantages remove the explicit dependence on the learner's state, thereby eliminating the issue of trajectory mismatch described above. In particular, unlike OLwS, we *do not* need to verify that the iterates produced by our algorithm change slowly.

### 1.2. Our approach: Online learning with advantages

In this section we sketch the *online learning with advantages* (OLwA) technique underlying Riccatitron. Let $\pi^\star$ denote the optimal unconstrained policy given knowledge of the entire disturbance sequence $\boldsymbol{w}$, and let $\mathbf{Q}^\star_t(x, u; \boldsymbol{w})$ be the associated Q-function (this quantity is formally defined in Definition 3). The *advantage*[1] with respect to $\pi^\star$, $\mathbf{A}^\star_t(u; x, \boldsymbol{w}) := \mathbf{Q}^\star_t(x, u; \boldsymbol{w}) - \mathbf{Q}^\star_t(w, u, \pi^\star(x); \boldsymbol{w})$, describes the difference between the total cost accumulated by selecting action $u$ in state $x$ at time $t$ and subsequently playing according to the optimal policy $\pi^\star$, versus choosing $u_t = \pi^\star_t(x; \boldsymbol{w})$ as well. By the well-known performance dif-

[1] Since we use losses rather than rewards, "advantage" refers to the advantage of $\pi^\star$ over $u$ rather than the advantage of $u$ over $\pi^\star$; the latter terminology is more common in reinforcement learning.

ference lemma (Kakade, 2003), the relative cost of a policy is *equal* the sum of the advantages under the states visited by said policy:[2]

$$J_T(\pi;\boldsymbol{w}) - J_T(\pi^\star;\boldsymbol{w}) = \sum_{t=1}^{T} \mathbf{A}_t^\star(u_t^\pi; x_t^\pi, \boldsymbol{w}). \qquad (2)$$

With this observation, the regret $\mathrm{Reg}_T(\pi^{\mathrm{alg}};\boldsymbol{w},\Pi)$ of any algorithm $\pi^{\mathrm{alg}}$ to a policy class $\Pi$ can be expressed as:

$$\sum_{t=1}^{T} \mathbf{A}_t^\star(u_t^{\mathrm{alg}}; x_t^{\mathrm{alg}}, \boldsymbol{w}) - \inf_{\pi\in\Pi} \sum_{t=1}^{T} \mathbf{A}_t^\star(u_t^\pi; x_t^\pi, \boldsymbol{w}). \qquad (3)$$

The expression (3) suggests that a reasonable approach might be to run an online learner on the functions $\pi \mapsto \mathbf{A}_t^\star(u_t^\pi; x_t^\pi, \boldsymbol{w})$. However, there are two issues. First, the advantages in the first sum are evaluated on the states $x_t^{\mathrm{alg}}$ under $\pi^{\mathrm{alg}}$, and in the second sum under the comparator trajectories $x^\pi$ (trajectory mismatch). Second, like $\pi^\star$ itself, the advantages require knowledge of all future disturbances, which are not yet known to the learner at time $t$. We show that if the control policies are parametrized using a particular optimal control law, the advantages do not depend on the state, and can be approximated using only finite lookahead.

**Theorem 2 (informal).** *For control policies $\pi$ with a suitable parametrization, the mapping $\pi \mapsto \mathbf{A}_t^\star(u_t^\pi; x_t^\pi, \boldsymbol{w})$ can be arbitrarilily-well approximated by a function $\pi \mapsto \widehat{\mathbf{A}}_{t;h}(\pi; w_{1:t+h})$ which (1) does not depend on the state, (2) can be determined by the learner at time t + h, and (3) has a simple quadratic structure.*

The "magic" behind this theorem is that the functional dependence of the unconstrained optimal policy $\pi^\star(x;\boldsymbol{w})$ on the state $x$ is linear, and does not depend $\boldsymbol{w}$ (Theorem 3). As a consequence, the state-dependent portion of $\pi^\star$ can be built into the controller parametrization, leaving only the $\boldsymbol{w}$-dependent portion up to the online learner. In light of this result, we use online learning to ensure low regret on the sequence of loss functions $f_t(\pi) := \widehat{\mathbf{A}}_{t;h}(\pi; w_{1:t+h})$; we address the fact that $f_t$ is only revealed to the learner after a delay of $h$ steps via a standard reduction (Joulani et al., 2013). We then show that for an appropriate controller parameterization $f_t(\pi)$ is exp-concave with respective to the learner's policy and hence second-order online learning algorithms attain logarithmic regret (Hazan et al., 2007).

We refer the reader to Appendix C for an in-depth overview of the OLwS framework, its relationship to OLwA, and challenges associated with using these techniques to achieve logarithmic regret.

### 1.3. Preliminaries

We consider the linear control setting in (1). For normalization, we assume $\|w_t\| \le 1\ \forall t$. We also assume $x_1 = 0$.

**Policies and trajectories.** We consider policies $\pi$ parameterized as functions of $x_t$ and $\boldsymbol{w}$ via $u_t = \pi_t(x_t;\boldsymbol{w})$. We assume that, when selecting action $u_t$ at time $t$, the learner has access to all states $x_{1:t}, u_{1:t-1}$, as well as $w_{1:t-1}$ (the latter assumption is without loss of generality by the identity $w_s = x_{s+1} - Ax_s - Bu_s$). Thus, a policy is said to be *executable* if $\pi_t(x;\boldsymbol{w})$ depends only on $x$ and $w_{1:t-1}$, i.e. $\pi(x;\boldsymbol{w}) = \pi(w; w_{1:t-1})$. For analysis purposes, we also consider *non-executable* whose value at time $t$ may depend on the entire sequence $\boldsymbol{w}$. For a policy $\pi$ and sequence $\boldsymbol{w}$, we let $x_t^\pi(\boldsymbol{w}), u_t^\pi(\boldsymbol{w})$ denote the resulting states and input trajectories (which we note depend only on $w_{1:t-1}$). For simplicity, we often write $x_t^\pi$ and $u_t^\pi$, supressing the $\boldsymbol{w}$-dependence. We shall let $\pi^{\mathrm{alg}}$ refer to the policy selected by the learner's algorithm, and use the shorthand $x_t^{\mathrm{alg}}(\boldsymbol{w})$, $u_t^{\mathrm{alg}}(\boldsymbol{w})$ to denote the corresponding trajectories. Given a class of policies $\Pi$, the regret of the policy $\pi^{\mathrm{alg}}$ is given by

$$\mathrm{Reg}_T(\pi^{\mathrm{alg}};\Pi,\boldsymbol{w}) = J_T(\pi^{\mathrm{alg}};\boldsymbol{w}) - \inf_{\pi\in\Pi} J_T(\pi;\boldsymbol{w}).$$

We consider a benchmark class of policies induced by state feedback control laws $\pi_t^K(x) = -Kx$, indexed by matrices $K \in \mathbb{R}^{d_{\mathbf{u}}\times d_{\mathbf{x}}}$.

**Linear control theory.** We say that a linear controller $K \in \mathbb{R}^{d_{\mathbf{u}} d_{\mathbf{x}}}$ is *stabilizing* if $A - BK$ is *stable*, that is $\rho(A - BK) < 1$ where $\rho(\cdot)$ denotes the spectral radius.[3] We assume the system $(A,B)$ is *stabilizable* in the sense that there exists a stabilizing controller $K$. For any stabilizable system, there is a unique positive semidefinite solution $P_\infty \succeq 0$ to the *discrete algebraic Riccati equation* (henceforth, DARE),

$$P = A^\top PA + R_x - A^\top PB(R_u + B^\top PB)^{-1}B^\top PA. \quad (4)$$

The solution $P_\infty$ to (4) is an intrinsic property of the system (1) with $(A,B)$ and characterizes the optimal infinite-horizon cost for control in the absence of noise (Bertsekas, 2005). Our algorithms and analysis make use of this parameter, as well as the corresponding optimal state feedback controller $K_\infty := (R_u + B^\top P_\infty B)^{-1}B^\top P_\infty A$. We also use the steady-state covariance matrix $\Sigma_\infty := R_u + B^\top P_\infty B$ and closed-loop dynamics matrix $A_{\mathrm{cl},\infty} := A - BK_\infty$.

**Competing with state feedback.** While $K_\infty$ represents the (asymptotically) optimal control law in the presense of uncorrelated, unbiased stochastic noise, $\pi^{K_\infty}$ may not be

[2] See Lemma D.12 in Appendix E for a general statement of the performance difference lemma. The invocation of the performance difference lemma here is slightly different from other results on online learning in MDPs such as Even-Dar et al. (2009), in that the role of $\pi$ and $\pi^\star$ is swapped.

[3] For a possibly asymmetric matrix $A$, $\rho(A) = \max\{|\lambda| \mid \lambda \text{ is an eigenvalue for } A\}$.

the optimal state feedback policy in hindsight for a given sequence of adversarial perturbations $w_t$. Hence, we compete with linear controllers that satisfy a quantitative version of the stability property.

**Definition 1** (Strong Stability (Cohen et al., 2018))**.** *We say that $A - BK \in \mathbb{R}^{d_{\mathsf{x}} \times d_{\mathsf{x}}}$ is $(\kappa, \gamma)$-strongly stable if there exists matrices $H, L \in \mathbb{R}^{d_{\mathsf{x}} \times d_{\mathsf{x}}}$ such that $A - BK = HLH^{-1}$, $\|H\|_{\mathrm{op}}\|H\|_{\mathrm{op}}^{-1} \le \kappa$ and $\|L\|_{\mathrm{op}} \le \gamma$.*

Given parameters $(\kappa_0, \gamma_0)$, we consider the benchmark class

$$\mathcal{K}_0 = \left\{\|K\|_{\mathrm{op}} \le \kappa_0 : A - BK \text{ is } (\kappa_0, \gamma_0)\text{-strongly stable}\right\}.$$

Lemma D.1 (Appendix D.1) shows that the closed-loop dynamics for $K_\infty$ are always $(\kappa_\infty, \gamma_\infty)$-strongly stable for suitable $\gamma_\infty, \kappa_\infty$. We assume that $\mathcal{K}_0$ is chosen such that $\kappa_\infty \le \kappa_0$ and $\gamma_\infty \le \gamma_0$.[4] Our algorithms minimize policy regret to the class of induced policies for $\mathcal{K}_0$:

$$\mathcal{K}_0\text{-}\mathrm{Reg}_T(\pi^{\mathrm{alg}}; \boldsymbol{w}) \coloneqq J_T(\pi^{\mathrm{alg}}; \boldsymbol{w}) - \inf_{K \in \mathcal{K}_0} J_T(\pi^K; \boldsymbol{w}).$$

**Problem parameters.** Our regret bounds depend on the following basic parameters for the LQR problem: $\Psi_\star \coloneqq \max\left\{1, \|A\|_{\mathrm{op}}, \|B\|_{\mathrm{op}}, \|R_x\|_{\mathrm{op}}, \|R_u\|_{\mathrm{op}}\right\}$, $\beta_\star \coloneqq \max\left\{1, \lambda_{\min}^{-1}(R_u), \lambda_{\min}^{-1}(R_x)\right\}$, $\Gamma_\star \coloneqq \max\left\{1, \|P_\infty\|_{\mathrm{op}}\right\}$.

**Additional notation.** We adopt non-asymptotic big-oh notation: For functions $f, g : \mathcal{X} \to \mathbb{R}_+$, we write $f = \mathcal{O}(g)$ if there exists some constant $C > 0$ such that $f(x) \le Cg(x)$ for all $x \in \mathcal{X}$. We use $\widetilde{\mathcal{O}}(\cdot)$ so suppress logarithmic dependence on system parameters, and we use $\mathcal{O}_\star(\cdot)$ to suppress *all* dependence on system parameters. For a vector $x \in \mathbb{R}^d$, we let $\|x\|$ denote the euclidean norm and $\|x\|_\infty$ denote the element-wise $\ell_\infty$ norm. For a matrix $A$, we let $\|A\|_{\mathrm{op}}$ denote the operator norm. If $A$ is symmetric, we let $\lambda_{\min}(A)$ denote the minimum eigenvalue. When $P \succ 0$ is a positive definite matrix, we let $\|x\|_P = \sqrt{\langle x, Px\rangle}$ denote the induced weighted euclidean norm. We et $\boldsymbol{w}_{t-1} = (w_{t-1}, w_{t-2}, \dots, w_1, \mathbf{0}, \mathbf{0}, \dots)$ denote a sequence of past $w$s, terminating in an infinite sequence of zeros. To simplify indexing, we let $w_s \equiv \mathbf{0}$ for $s \le 0$, so that $\boldsymbol{w}_{t-1} = (w_{t-1}, w_{t-2}, \dots)$ We also let $w_s \equiv \mathbf{0}$ for $s > T$.

### 1.4. Organization

Section 2 introduces the Riccatitron algorithm, states its formal regret guarantee, and gives an overview of the algorithm's building blocks and proof techniques. Section 3 gives a high-level proof of the key "approximate advantage" theorem used by the algorithm. Omitted proofs are deferred to Appendix E and Appendix F, and additional technical tools stated and proven in Appendix D. Appendix A gives a detailed survey of related work. Appendix B sketches extensions of Riccatitron to more general settings, and Appendix C gives a detailed survey of challenges associated with applying previous approaches to online reinforcement learning to obtain logarithmic regret in our setting.

## 2. Logarithmic regret for online linear control

Our main algorithm, Riccatitron, is described in Algorithm 1. The algorithm combines several ideas.

1. Following Agarwal et al. (2019a), we move from linear policies of the form $\pi^K(x; \boldsymbol{w}) = -Kx$, to a relaxed set of *disturbance-action* (DAP) policies of the form $\pi_t^{(M)}(x; \boldsymbol{w}) = -K_\infty x - q^M(\boldsymbol{w}_{t-1})$, where
$$q^M(\boldsymbol{w}_{t-1}) = \sum_{i=1}^m M^{[i]} w_{t-i},$$
and where $K_\infty$ is linear controller from the DARE (4).
2. We show that the optimal unconstrained policy with full knowledge of the sequence $\boldsymbol{w}$ takes the form $\pi_t^\star(x; \boldsymbol{w}) = -K_t x - q_t^\star(w_{t:T})$, where $(K_t)$ is a particular sequence of linear controllers that arises from the so-called *Riccati recursion*. We then show that for *any* policy of the form $\pi_t(x; \boldsymbol{w}) = -K_\infty - q_t(\boldsymbol{w})$—in particular, for the DAP parameterization above—the advantage functions $\mathbf{A}_t^\star(u_t^\pi; x_t^\pi, \boldsymbol{w})$ can be well approximated by simple quadratic functions of the form
$$\|q_t(\boldsymbol{w}) - q_t^\star(w_{t:T})\|_{\Sigma_\infty}^2.$$
This essentially removes the learner's state from the equation, and reduces the problem of control to that of predicting the optimal controller's bias vector $q_t^\star(w_{t:T})$. The remaining challenge is that the optimal bias vectors depend on the future disturbances, which are not available to the learner at time $t$.
3. We show that the advantages can be truncated to require only finite lookahead, thereby reducing the problem to *online learning with delayed feedback*. We then apply a reduction from delayed online learning to classical online learning (Joulani et al., 2013), which proceeds by running multiple copies of a base online learning algorithm over separate subsequences of rounds.
4. Finally—using the structure of the disturbance-action parameterization—we show that the resulting online learning problem is exp-concave. As a result, we can use a second-order online learning algorithm—either online Newton step (ONS, Hazan et al. (2007)) given in Algorithm 2, or Vovk-Azoury-Warmuth (VAW, Vovk (1998); Azoury & Warmuth (2001)) given in Algorithm 3—as our base learner to obtain logarithmic regret.

[4] This assumption only serves to keep notation compact.

**Algorithm 1** Riccatitron

```
parameters:
    Horizon $h$, DAP length $m$, radius $R$, decay factor $\gamma$.
    Online Newton parameters $\eta_{\mathrm{ons}}, \varepsilon_{\mathrm{ons}}$,
      or Vovk-Azoury-Warmuth parameter $\varepsilon_{\mathrm{vaw}}$.
initialize:
    Let $\mathcal{M}_0 \leftarrow \mathcal{M}(m,R,\gamma)$ (Eq. 5).
    Instantiate base learners $\mathsf{BL}^{(1)},\dots,\mathsf{BL}^{(h+1)}$ as either
      $\mathsf{ONS}(\varepsilon_{\mathrm{ons}},\eta_{\mathrm{ons}},\mathcal{M}_0)$ or $\mathsf{VAW}(\varepsilon_{\mathrm{vaw}},\mathcal{M}_0,\Sigma_\infty)$.
Let $\tau_t = (t-1) \mod (h+1) + 1 \in [h+1]$.
for $t = 1,\dots,T$: do
    // Predict using base learner $\tau_t$.
    Let $M_t$ denote the $k_t$-th iterate produced
      by $\mathsf{BL}^{(\tau_t)}$ where $k_t \leftarrow \lfloor t/(h+1) \rfloor$.
    Play $u_t = -K_\infty x_t - q^{M_t}(\boldsymbol{w}_{t-1})$ (Definition 2).
    Observe $x_{t+1}$ and $w_t$.
    // Update base learner $\tau_{t+1}$.
    if $t \ge h+1$ then
        // Approximate advantage from Eq. (10).
        Update $\mathsf{BL}^{(\tau_t+1)}$ with $\widehat{\mathbf{A}}_{t-h;h}(M;\boldsymbol{w}_t)$.
```

Together, these components give rise to the scheme in Algorithm 1. At time $t$, the algorithm plays the action $u_t = -K_\infty x_t - q^{M_t}(\boldsymbol{w}_{t-1})$, where $M_t$ is provided by the ONS (or VAW) instance responsible for the current round. The algorithm then observes $w_t$ and uses this to form the approximate advantage function for time $t-h$, where $h$ is the lookahead distance. The advantage is then used to update the ONS/VAW instance responsible for the next round. The main regret guarantee for this approach is as follows.

**Theorem 1.** *For an appropriate choice of parameters,* Riccatitron *ensures*

$$\mathcal{K}_0\text{-Reg}_T \le \mathcal{O}_\star(d_{\mathbf{x}} d_{\mathbf{u}} \log^3 T),$$

*where $\mathcal{O}_\star$ suppresses polynomial dependence on system parameters. Suppressing only logarithmic dependence on system parameters, the regret is at most*

$$\widetilde{\mathcal{O}}\big(d_{\mathbf{x}} d_{\mathbf{u}} \log^3 T \cdot \beta_\star^{11} \Psi_\star^{19} \Gamma_\star^{11} \kappa_0^8 (1-\gamma_0)^{-4}\big).$$

In the remainder of this section we overview the algorithmic building blocks of Riccatitron and the key ideas of the proof.

### 2.1. Disturbance-action policies

Cost functionals parametrized by state feedback controllers (e.g., $K \mapsto J_T(\pi^K;\boldsymbol{w})$) are generally non-convex (Fazel et al., 2018). To enable the use of tools from online convex optimization, we use a convex *disturbance-action* controller parameterization introduced by Agarwal et al. (2019a).

**Algorithm 2** Online Newton Step ($\mathsf{ONS}(\varepsilon,\eta,\mathcal{C},\Sigma)$)

```
parameters: Learning rate $\eta > 0$, regularization parameter $\varepsilon > 0$, convex constraint set $\mathcal{C}$.
    // OCO with exp-concave costs $f_k(z)$, where $z \in \mathcal{C} \subset \mathbb{R}^d$.
initialize: $d \leftarrow \dim(\mathcal{C})$, $z_1 \in \mathcal{C}$, $E_0 \leftarrow \varepsilon \cdot I_d$.
for $k = 1,2,\dots$: do
    Play $z_k$ and receive gradient $\nabla_k := \nabla f_k(z_k)$.
    $E_k \leftarrow E_{k-1} + \nabla_k \nabla_k^\top$.
    $\widetilde{z}_{k+1} \leftarrow z_k - \eta E_k^{-1} \nabla_k$.
    $z_{k+1} \leftarrow \arg\min_{z\in\mathcal{C}} \|z - \widetilde{z}_{k+1}\|_{E_k}^2$.
```

**Definition 2** (Disturbance-action policy (DAP))**.** *Let $M = (M^{[i]})_{i=1}^m$ denote a sequence of matrices $M^{[i]} \in \mathbb{R}^{d_{\mathbf{u}} \times d_{\mathbf{x}}}$. We define the corresponding disturbance-action policy $\pi^{(M)}$ as $\pi_t^{(M)}(x;\boldsymbol{w}) = -K_\infty x - q^M(\boldsymbol{w}_{t-1})$, where $q^M(\boldsymbol{w}_{t-1}) = \sum_{i=1}^m M^{[i]} w_{t-i}$.*

We work with DAPs for which the sequence $M$ belongs to the set

$$\mathcal{M}(m,R,\gamma) := \{M = (M^{[i]})_{i=1}^m : \|M^{[i]}\|_{\mathrm{op}} \le R\gamma^{i-1}\}, \tag{5}$$

where $m$, $R$, and $\gamma$ are algorithm parameters. We note that DAPs can be defined with general stabilizing controllers $K \ne K_\infty$, but the choice $K = K_\infty$ is critical in the design and analysis of our main algorithm.

The first lemma we require is a variant of a result of Agarwal et al. (2019a), which shows that disturbance-action policies are sufficiently rich enough to approximate all state feedback laws.

**Lemma 2.1** (Expressivity of DAP)**.** Suppose we choose our set of disturbance-action matrices as $\mathcal{M}_0 := \mathcal{M}(m, R_\star, \gamma_0)$, where $m = (1-\gamma_0)^{-1}\log((1-\gamma_0)^{-1}T)$ and $R_\star = 2\beta_\star \Psi_\star^2 \Gamma_\star \kappa_0^2$. Then for all $\boldsymbol{w}$, we have

$$\inf_{M\in\mathcal{M}_0} J_T(\pi^{(M)};\boldsymbol{w}) \le \inf_{K\in\mathcal{K}_0} J_T(\pi^K;\boldsymbol{w}) + C_{\mathrm{apx}},$$

where $C_{\mathrm{apx}} \le \mathcal{O}(\beta_\star^2 \Psi_\star^8 \Gamma_\star^2 \kappa_0^7 (1-\gamma_0)^{-2})$.

We refer the reader to Appendix E.2 for a proof. Going forward, we define

$$D_q = \widetilde{\mathcal{O}}\Big(\beta_\star^{5/2} \Psi_\star^3 \Gamma_\star^{5/2} \kappa_0^2 (1-\gamma_0)^{-1}\Big), \tag{6}$$

which serves as an upper bound on $\|q_t^M\|$ for $M \in \mathcal{M}_0$, as well as other certain other bias vector sequences that arise in the subsequent analysis. In light of Lemma 2.1, the remainder of our discussion will directly bound regret with respect to DAPs:

$$\mathcal{M}_0\text{-Reg}_T(\pi;\boldsymbol{w}) := J_T(\pi;\boldsymbol{w}) - \inf_{M\in\mathcal{M}_0} J_T(\pi^{(M)};\boldsymbol{w}). \tag{7}$$

We note in passing that DAPs are actually rich enough to compete with a broader class of linear control policies with internal state; this extension is addressed in Appendix B.2.

### 2.2. Advantages in linear control

To proceed, we adopt the OLwA paradigm, which minimizes approximations to the *advantages* (or, differences between the Q-functions) relative to the optimal unconstrained policy $\pi^\star$ given access to the entire sequence $\boldsymbol{w}$. Recalling $\ell(x,u) = \|x\|_{R_x}^2 + \|u\|_{R_u}^2$, we define the optimal controller $\pi^\star$ and associated Q-functions and advantages by induction.

**Definition 3.** *The optimal Q-function and policy at time $T$ are given by $\mathbf{Q}_T^\star(x,u;\boldsymbol{w}) = \ell(x,u)$, $\pi_T^\star(x;\boldsymbol{w}) = \min_u \mathbf{Q}_T^\star(x,u;\boldsymbol{w}) = 0$, and $\mathbf{V}_T^\star(x;\boldsymbol{w}) = \ell(x,0) = \|x\|_{R_x}^2$. For each timestep $t < T$, the optimal Q-function and policy are given by*

$$\begin{aligned}
&\mathbf{Q}_t^\star(x,u;\boldsymbol{w}) = \|x\|_Q^2 + \|u\|_R^2 + \mathbf{V}_{t+1}^\star(Ax + Bu + w_t;\boldsymbol{w}),\\
&\pi_t^\star(x;\boldsymbol{w}) = \operatorname*{arg\,min}_{u\in\mathbb{R}^{d_\mathbf{u}}} \mathbf{Q}_t^\star(x,u;\boldsymbol{w}),\\
&\mathbf{V}_t^\star(x;\boldsymbol{w}) = \min_{u\in\mathbb{R}^{d_\mathbf{u}}} \mathbf{Q}_t^\star(x,u;\boldsymbol{w}) = \mathbf{Q}_t^\star(x,\pi_t^\star(x;\boldsymbol{w});\boldsymbol{w}).
\end{aligned}$$

*The advantage function for the optimal policy is $\mathbf{A}_t^\star(u;x,\boldsymbol{w}) := \mathbf{Q}_t^\star(x,u;\boldsymbol{w}) - \mathbf{Q}_t^\star(x,\pi_t^\star(x;\boldsymbol{w});\boldsymbol{w})$.*

The advantage function $\mathbf{A}_t^\star(u;x,\boldsymbol{w})$ represents the total excess cost incurred by selecting a control $u \neq \pi_t^\star(x;\boldsymbol{w})$ at state $x$ and time $t$, assuming we follow $\pi^\star$ for the remaining rounds. We have $\mathbf{A}_t^\star(u;x,\boldsymbol{w}) \geq 0$ since, by Bellman's optimality condition, $\pi_t^\star(x;\boldsymbol{w})$ is a minimizer of $\mathbf{Q}^\star(x,u;\boldsymbol{w})$.

The advantages arise in our setting through application of the performance difference lemma (Lemma D.12), which we recall states that for any policy $\pi$, the regret to $\pi^\star$ is equal to the sum of advantages under the trajectory induced by $\pi$, i.e. $J_T(\pi;\boldsymbol{w}) - J_T(\pi^\star;\boldsymbol{w}) = \sum_{t=1}^T \mathbf{A}_t^\star(u_t^\pi;x_t^\pi,\boldsymbol{w})$. To analyze Riccatitron, we apply this identity to obtain the regret decomposition

$$\begin{aligned}
\mathcal{M}_0\text{-Reg}_T(\pi;\boldsymbol{w}) &= \sum_{t=1}^T \mathbf{A}_t^\star(u_t^\pi;x_t^\pi,\boldsymbol{w})\\
&\quad - \inf_{M\in\mathcal{M}_0}\sum_{t=1}^T \mathbf{A}_t^\star(u_t^{\pi^{(M)}};x_t^{\pi^{(M)}},\boldsymbol{w}) \qquad (8)
\end{aligned}$$

This decomposition is *exact*, and avoids the pitfalls of the usual stationary cost-based regret decomposition associated with the classical OLwS approach (cf. Appendix C). Our goal going forward will be to treat these advantages as "losses" that can be fed into an appropriate online learning algorithm to select controls. However, this approach presents three challenges: (a) the advantages for the policy $\pi$ are evaluated on the trajectory $x_t^\pi$, while the advantages for comparator are evaluated under the trajectory induced by $\pi^{(M)}$; (b) the advantage is a difference in Q-functions that considers *all* future expected reward. In particular, $\mathbf{A}_t^\star(\cdot;\cdot,\boldsymbol{w})$ depends on all future $w_t$s, including those not yet revealed to the learner; (c) the functional form of the advantages is opaque, and it is not clear that any online learning algorithm can achieve logarithmic regret even if they were able to evaluate $\mathbf{A}_t^\star$ at time $t$.

### 2.3. Approximate advantages

Our main structural result—and the starting point for Riccatitron—is the following observation. Let $\pi$ be any policy of the form $\pi_t(x;\boldsymbol{w}_{t-1}) = -K_\infty x - q^{M_t}(\boldsymbol{w}_{t-1})$, where $M_t = M_t(\boldsymbol{w}_{t-1})$ are arbitrary functions of past $w$, and where $K_\infty$ is the infinite horizon Riccati optimal controller. Then $\mathbf{A}_t^\star(u_t^\pi;x_t^\pi,\boldsymbol{w})$ is well-approximated by an *approximate advantage function* $\widehat{\mathbf{A}}_{t;h}(M;\boldsymbol{w}_{t+h})$ which (a) *does not* depend on the state, and (b) depends on only a small horizon $h$ of future disturbances, and (c) is a pure quadratic function of $M$, and thereby amenable to fast (logarithmic) rates for online learning. Let $h$ be a horizon/lookahead parameter. Defining

$$q_{\infty;h}^\star(w_{1:h+1}) := \sum_{i=1}^{h+1} \Sigma_\infty^{-1} B^\top (A_{\mathrm{cl},\infty}^\top)^{i-1} P_\infty w_i, \qquad (9)$$

the approximate advantage function is

$$\widehat{\mathbf{A}}_{t;h}(M;\boldsymbol{w}_{t+h}) := \|q^M(\boldsymbol{w}_{t-1}) - q_{\infty;h}^\star(w_{t:t+h})\|_{\Sigma_\infty}^2. \quad (10)$$

The following theorem facilitates the use of the approximate advantages.

**Theorem 2.** *Let $\pi$ be any policy of the form $\pi_t(x;\boldsymbol{w}) = -K_\infty x - q^{M_t}(\boldsymbol{w}_{t-1})$, where $M_t = M_t(\boldsymbol{w}) \in \mathcal{M}_0$. Then, by choosing $h = 2(1-\gamma_\infty)^{-1}\log(\kappa_\infty^2\beta_\star^2\Psi_\star\Gamma_\star^2T^2)$ as the horizon parameter, we have*

$$\sum_{t=1}^T \left|\mathbf{A}_t^\star(u_t^\pi;x_t^\pi,\boldsymbol{w}) - \widehat{\mathbf{A}}_{t;h}(M_t;\boldsymbol{w}_{t+h})\right| \leq C_{\mathrm{adv}},$$

*where $C_{\mathrm{adv}} = \widetilde{\mathcal{O}}\big(\beta_\star^{11}\Psi_\star^{19}\Gamma_\star^{11}\kappa_0^8(1-\gamma_0)^{-4}\log^2 T\big)$.*

The proof of this theorem constitutes a primary technical contribution of our paper, and is proven in Section 3. Briefly, the idea behind the result is to use that the optimal policy $\pi^\star$ itself satisfies $\pi_t^\star(x;\boldsymbol{w}) \approx -K_\infty x - q_{\infty;h}^\star(w_{t:t+h})$ whenever $h$ is sufficiently large and $t \leq T - \mathcal{O}_\star(\log T)$, and that $\mathbf{A}_t^\star$ has a simple quadratic structure. This characterization for is why it is essential to consider advantages with respect to the *optimal* policy $\pi^\star$, and why our DAPs use the controller $K_\infty$ as opposed to an arbitrary stabilizing controller as in Agarwal et al. (2019a).

### 2.4. Online learning with delays

An immediate consequence of Theorem 2 is that for any algorithm (in particular, Riccatitron) which selects $\pi_t(x;\boldsymbol{w}) =$

$-K_\infty x - q^{M_t}(\boldsymbol{w}_{t-1})$, the regret $\mathcal{M}_0$-$\mathrm{Reg}_T(\pi;\boldsymbol{w})$ is at most

$$\sum_{t=1}^{T}\widehat{\mathbf{A}}_{t;h}(M_t;\boldsymbol{w}_{t+h}) - \inf_{M\in\mathcal{M}_0}\sum_{t=1}^{T}\widehat{\mathbf{A}}_{t;h}(M;\boldsymbol{w}_{t+h}) + 2C_{\mathrm{adv}}. \tag{11}$$

This is simply an online convex optimization problem with $M_1,\ldots,M_T$ as iterates—the only catch is that the "loss" at time $t$, $\widehat{\mathbf{A}}_{t;h}(M_t;\boldsymbol{w}_{t+h})$, can only be evaluated after observing $w_{t:h}$, which will not be revealed to the learner until after round $t+h$. This is therefore an instance of online learning with *delays*, namely, the loss function suffered at time $t$ is only available at time $t+h+1$ (since $w_t$ is revealed at time $t+1$). To reduce the problem of minimizing regret on the approximate advantages in (11) to classical online learning without delays, we use a simple black-box reduction.

Consider a generic online convex optimization setting where, at each time $t$, the learner proposes an iterate $z_t$, then suffers cost $f_t(z_t)$ and observes $f_t$ (or some function of it). Suppose we have an algorithm for this non-delayed setting that guarantees that for every sequence, $\sum_{t=1}^{T} f_t(z_t) - \inf_{z\in\mathcal{C}}\sum_{t=1}^{T} f_t(z_t) \le R(T)$, where $R$ is increasing in $T$. Now consider the same setting with delay $h$, and let $\tau(t) = (t-1) \mod (h+1) + 1 \in [h+1]$. We use the following strategy: Make $h+1$ copies of the base algorithm. At round $t$, observe $z_t$, predict $z_t$ using the output of instance $\tau(t)$, then update instance $\tau(t+1)$ using the loss $f_{t-h}(z_{t-h})$ (which is now available).

**Lemma 2.2** (cf. Joulani et al. (2013))**.** The generic delayed online learning reduction has regret at most

$$\sum_{t=1}^{T} f_t(z_t) - \inf_{z\in\mathcal{C}}\sum_{t=1}^{T} f_t(z) \le (h+1)R(T/(h+1)),$$

where $R(T)$ is the regret of the base instance.

Lemma 2.2 shows that minimizing the regret in (11) is as easy as minimizing regret in the non-delayed setting, up to a factor of $h = \mathcal{O}_\star(\log T)$. For completeness, we provide a proof Appendix E.4. All that remains is to specify the base algorithm for the reduction.

### 2.5. Exp-concave online learning

We have reduced the problem of obtaining logarithmic regret for online control to obtaining logarithmic regret for online learning with approximate advantages of the form in (11). A sufficient condition to obtain fast rates in online learning is strong convexity of the loss Hazan (2016), but while the advantages $\widehat{\mathbf{A}}_{t;h}(M;\boldsymbol{w}_{t+h})$ are strongly convex with respect to $q^M(\boldsymbol{w})$, they are not strongly convex with respect to the parameter $M$. Itself. Fortunately, logarithmic regret can also be achieved for loss functions that satisfy a weaker condition called exp-concavity (Hazan et al., 2007; Cesa-Bianchi & Lugosi, 2006).

**Definition 4.** *A function $f : \mathcal{C} \to \mathbb{R}$ is $\alpha$-exp-concave if $\nabla^2 f(z) \succeq \alpha(\nabla f(z))(\nabla f(z))^\top$ for all $z \in \mathcal{C}$.*

Intuitively, an exp-concave function $f$ exhibits strong curvature along the directions of its gradient, which are precisely the directions along which $f$ is sensitive to change. This property holds for linear regression-type losses, as the following standard lemma (Appendix E.4) shows.

**Lemma 2.3.** Let $A \in \mathbb{R}^{d_1\times d_2}$, and consider the function $f(z) = \|Az - b\|_\Sigma^2$, where $\Sigma \succeq 0$. If we restrict to $z \in \mathbb{R}^{d_2}$ for which $f(z) \le R$, then $f$ is $(2R)^{-1}$-exp-concave.

Observe that the approximate advantage functions $\widehat{\mathbf{A}}_{t;h}(M;\boldsymbol{w}_{t+h})$ indeed have the form $f(z) = \|Az - b\|_\Sigma^2$ (viewing the map $M \mapsto q^M(\boldsymbol{w}_{t-1})$ as a linear operator), and thus satisfy exp-concavity for appropriate $\alpha > 0$. To take advantage of this property we use online Newton step (ONS, Algorithm 2), a second-order online convex optimization algorithm which guarantees logarithmic regret for exp-concave losses.

**Lemma 2.4** (Hazan (2016))**.** Suppose that $\sup_{z,z'\in\mathcal{C}}\|z-z'\| \le D$, $\sup_{z\in\mathcal{C}}\|\nabla f_t(z)\| \le G$, and that each loss $f_k$ is $\alpha$-exp-concave. Then by setting $\eta = 2\max\{4GD, \alpha^{-1}\}$ and $\varepsilon = \eta^2/D$, the online Newton step algorithm guarantees

$$\sum_{k=1}^{T} f_k(z_k) - \inf_{z\in\mathcal{C}}\sum_{k=1}^{T} f_k(z) \le 5(\alpha^{-1} + GD)\cdot d\log T.$$

**Putting everything together.** With the regret decomposition in terms of approximate advantages (Theorem 2) and the blackbox-reduction for online learning with delays (Lemma 2.2), the design and analysis of Riccatitron (Algorithm 1) is rather simple. In view of Lemma 2.1, we initialize the set $\mathcal{M}_0$ sufficiently large to compete with the appropriate state-feedback controllers (Line 2). Using Theorem 2, our goal is to obtain a regret bound for the approximate advantages in (11). In view of the delayed online learning reduction Lemma 2.2, we initialize $h+1$ base online learners (Line 2). Since the approximate advantages $\widehat{\mathbf{A}}_t$ are pure quadratics, we use online Newton step for the base learner, which ensures logarithmic regret via Lemma 2.4.

### 2.6. Sharpening the regret bound

With online Newton step as the base algorithm, Riccatitron has regret $\mathcal{O}_\star(d_{\mathbf{x}}d_{\mathbf{u}}\sqrt{d_{\mathbf{x}} \wedge d_{\mathbf{u}}}\log^3 T)$. The $d_{\mathbf{x}}d_{\mathbf{u}}$ factor comes from the hard dependence on $\dim(\mathcal{C})$ in the ONS regret bound (Lemma 2.4), while the $\sqrt{d_{\mathbf{x}} \wedge d_{\mathbf{u}}}$ factor is an upper bound on the Frobenius norm for each $M \in \mathcal{M}_0$. We can obtain improved dimension dependence by replacing ONS with a vector-valued variant of the classical Vovk-Azoury-Warmuth algorithm (VAW), described in Algorithm 3 (Appendix E.3). The VAW algorithm goes beyond the generic exp-concave online learning setting and exploits the quadratic structure of the approximate advantages. Theorem 5 in Appendix E.3 shows that its regret depends only logarithmically on the Frobenius norm of the parameter vectors, so it avoids the $\sqrt{d_{\mathbf{x}} \wedge d_{\mathbf{u}}}$ factor paid by ONS (up to a log term). This leads to a final regret bound of $\mathcal{O}_\star(d_{\mathbf{x}} d_{\mathbf{u}} \log^3 T)$ for Riccatitron. The runtime for both algorithms is identical.

The calculation for the final regret bound is carried out in Appendix E.1.

## 3. Advantages without states

We now prove the key “approximate advantage” theorem (Theorem 2) used in the analysis of Riccatitron. The roadmap for the proof is as follows:

1. In Section 3.1, we show that the unconstrained optimal policy takes the form $\pi_t^\star(x; \boldsymbol{w}) = -K_t x_t - q_t^\star(\boldsymbol{w})$, where $q_t^\star(\boldsymbol{w})$ depends on all future disturbances, and where $K_t$ is the finite-horizon solution to the Riccati recursion (Definition 5).

2. Next, Section 3.2 presents an intermediate version of the approximate advantage theorem for policies of the form $\widehat{\pi}_t(x; \boldsymbol{w}) = -K_t x_t - q^{M_t}(\boldsymbol{w}_{t-1})$. Because any such policy has the same state dependence as the optimal policy $\pi^\star$, we are able to show that $\mathbf{A}_t^\star(u_t^{\widehat{\pi}}; x_t^{\widehat{\pi}}, \boldsymbol{w})$ has *no state dependence*. Moreover, the linear structure of the dynamics and quadratic structure of the losses ensures that $\mathbf{A}_t^\star(u_t^{\widehat{\pi}}; x_t^{\widehat{\pi}}, \boldsymbol{w})$ is a quadratic of the form $\|q^{M_t}(\boldsymbol{w}_{t-1}) - q_t^\star(w_{t:T})\|_{\Sigma_t}^2$, where $\Sigma_t$ is a finite-horizon approximation to $\Sigma_\infty$, and $q_t^\star(w_{t:T})$ is the bias vector of the optimal controller.

3. Finally (Section 3.3), we use stability of the Riccati recursion to show that $q_t^\star(\boldsymbol{w})$ can be replaced with a term that depends only on $\boldsymbol{w}_{t+h}$, up to a small error. Similarly, we show that $\Sigma_t$ can be replaced by $\Sigma_\infty$ and $K_t$ by $K_\infty$.

This argument implies that a slightly modified analogue of Riccatitron which replaces infinite-horizon quantities ($K_\infty$, $\Sigma_\infty$,...) with finite-horizon analogues from the Riccati recursion attains a similar regret. We state Riccatitron with the infinite horizon analogues to simplify presentation, as well as implementation.

### 3.1. A closed form for the true optimal policy

Our first result characterizes the optimal unconstrained optimal controller $\pi^\star$ given full knowledge of the disturbance sequence $\boldsymbol{w}$, as well as the corresponding value function. To begin, we introduce a variant of the classical *Riccati recursion*.

**Definition 5** (Riccati recursion)**.** *Define $P_{T+1} = 0$ and $c_{T+1} = 0$ and consider the recursion:*

$$\begin{aligned}
P_t &= R_x + A^\top P_{t+1} A - A^\top P_{t+1} B \Sigma_t^{-1} B^\top P_{t+1} A, \\
\Sigma_t &= R_u + B^\top P_{t+1} B, \\
K_t &= \Sigma_t^{-1} B^\top P_{t+1} A, \\
c_t(w_{t:T}) &= (A - BK_t)^\top (P_{t+1} w_t + c_{t+1}(w_{t+1:T})).
\end{aligned}$$

*We also define corresponding closed loop matrices via $A_{\mathrm{cl},t} = A - BK_t$.*

For i.i.d. disturbances with $\mathbb{E}[w_t] = 0$ for all times $t$, the optimal controller is the state feedback law $\pi_t(x) = -K_t x_t$, and $K_t \to K_\infty$ as $t \to -\infty$. The following theorem shows that for arbitrary disturbances the optimal controller applies the same state feedback law, but with an extra bias term that depends on the disturbance sequence.

**Theorem 3.** *The optimal controller is given by $\pi_t^\star(x, \boldsymbol{w}) = -K_t x - q_t^\star(w_{t:T})$, where*

$$q_t^\star(w_{t:T}) = \sum_{i=t}^{T-1} \Sigma_t^{-1} B^\top \left( \prod_{j=t+1}^{i} A_{\mathrm{cl},j}^\top \right) P_{i+1} w_i. \tag{12}$$

*Moreover, for each time $t$ we have*

$$\mathbf{V}_t^\star(x; \boldsymbol{w}) = \|x\|_{P_t}^2 + 2\langle x, c_t(w_{t:T})\rangle + f_t(w_{t:T}), \tag{13}$$

*where $f_t$ is a function that does not depend on the state $x$.*

Theorem 3 is a special case of a more general result, Theorem 4, proven in Appendix D.

### 3.2. Removing the state

We now use the characterization of $\pi^\star$ to show that the advantages $\mathbf{A}_t^\star(u_t^{\widehat{\pi}}; x_t^{\widehat{\pi}}, \boldsymbol{w})$ have a particularly simple structure when we consider policies of the form $\widehat{\pi}_t(x; \boldsymbol{w}) = -K_t x_t - q_t(\boldsymbol{w}_{t-1})$, where $q_t(\boldsymbol{w})$ is an arbitrary function of $\boldsymbol{w}$. For such policies, $\mathbf{A}_t^\star$is a quadratic function which does not depend explicitly on the state.

**Lemma 3.1.** Consider a policy $\widehat{\pi}_t(x)$ of the form $\widehat{\pi}_t(x; \boldsymbol{w}) = -K_t x_t - q_t(\boldsymbol{w})$. Then, for all $x$,

$$\mathbf{A}_t^\star(\widehat{\pi}_t(x; \boldsymbol{w}); x, \boldsymbol{w}) = \|q_t(\boldsymbol{w}) - q_t^\star(w_{t:T})\|_{\Sigma_t}^2.$$

*Proof.* Since $\mathbf{Q}_t^\star(x, \cdot; \boldsymbol{w})$ is a strongly convex quadratic, and since $\pi_t^\star(x; \boldsymbol{w}) = \arg\min_{u \in \mathbb{R}^{d_{\mathbf{u}}}} \mathbf{Q}_t^\star(x, u; \boldsymbol{w})$, first-order optimality conditions imply that for any $u$,

$$\begin{aligned}
\mathbf{A}_t^\star(u; x, \boldsymbol{w}) &= \mathbf{Q}_t^\star(x, u; \boldsymbol{w}) - \mathbf{Q}_t^\star(x, \pi_t^\star(x; \boldsymbol{w}); \boldsymbol{w}) \\
&= \|u - \pi_t^\star(x; \boldsymbol{w})\|_{\nabla_u^2 \mathbf{Q}_t^\star(x, u; \boldsymbol{w})}^2.
\end{aligned}$$

A direct computation based on (13) reveals that $\nabla_u^2 \mathbf{Q}_t^\star(x, u; \boldsymbol{w}) = R + B^\top P_{t+1} B = \Sigma_t$, so that $\mathbf{A}_t^\star(u; x, \boldsymbol{w}) = \|u - \pi_t^\star(x; \boldsymbol{w})\|_{\Sigma_t}^2$. Finally, since $\pi_t^\star(x; \boldsymbol{w}) = -K_t x - q_t^\star(w_{t:T})$, we have that if $u = \widehat{\pi}_t(x; \boldsymbol{w}) = -K_t x_t - q_t(\boldsymbol{w})$, then the states in the expression $u - \pi_t^\star(x; \boldsymbol{w})$ cancel, leaving $u - \pi_t^\star(x; \boldsymbol{w}) = -(q_t(\boldsymbol{w}) - q_t^\star(w_{t:T}))$. □

### 3.3. Truncating the future and passing to infinite horizon

The next lemma—proven in Appendix F—shows that we can truncate $q^\star_t(w_{t:T})$ to only depend on disturbances at most $h$ steps in the future.

**Lemma 3.2.** For any $h \in [T]$ define a truncated version of $q^\star_t$ as follows:

$$q^\star_{t;t+h}(w_{t:t+h}) = \sum_{i=t}^{(t+h)\wedge T-1} \Sigma_t^{-1} B^\top \left( \prod_{j=t+1}^{i} A_{\mathrm{cl},j}^\top \right) P_{i+1} w_i. \tag{14}$$

Then for any $t$ such that $t + h < T - \widetilde{\mathcal{O}}(\beta_\star \Psi_\star^2 \Gamma_\star)$, setting $\bar{\gamma}_\infty = \frac{1}{2}(1 + \gamma_\infty) < 1$, we have the bound $\|q^\star_{t:t+h}(w_{t:t+h}) - q^\star_t(w_{t:T})\| \le \kappa_\infty^2 \beta_\star^2 \Psi_\star \Gamma_\star^2 (T - h) \bar{\gamma}_\infty^h$, which is geometrically decreasing in $h$.

Going forward we use that both $q^\star_t$ and $q^\star_{t:t+h}$ have norm at most $\beta_\star \Psi_\star \Gamma_\star \kappa_\infty (1 - \gamma_\infty)^{-1} =: D_{q^\star}$ (Lemma D.6). As an immediate corollary of Lemma 3.2, we approximate the advantages using finite lookahead.

**Lemma 3.3.** Consider a policy $\widehat{\pi}_t(x; \boldsymbol{w}) = -K_t x_t - q_t(\boldsymbol{w})$, and suppose that $\|q_t\| \le D_q$, where $D_q \ge D_{q^\star}$. If we choose $h = 2(1 - \gamma_\infty)^{-1} \log(\kappa_\infty^2 \beta_\star^2 \Psi_\star \Gamma_\star^2 T^2)$, we are guaranteed that

$$\sum_{t=1}^{T} \left| \mathbf{A}_t^\star(u_t^{\widehat{\pi}}; x_t^{\widehat{\pi}}, \boldsymbol{w}) - \|q_t(\boldsymbol{w}) - q^\star_{t;t+h}(w_{t:t+h})\|^2_{\Sigma_t} \right| \le C_{\mathrm{trunc}},$$

where $C_{\mathrm{trunc}} \le \widetilde{\mathcal{O}}(D_q^2 \beta_\star \Psi_\star^4 \Gamma_\star^2 (1 - \gamma_\infty)^{-1} \log T)$.

At this point, we have established an analogue of Theorem 2, except that we are still using state-action controllers $K_t$ rather than $K_\infty$, and the approximate advantages in Lemma 3.3 are using the finite-horizon counterparts of $\Sigma_\infty$ and $q_{\infty;h}$. The following lemmas show that we can pass to these infinite-horizon quantities by paying a small approximation cost.

**Lemma 3.4.** Let policies $\pi_t(x; \boldsymbol{w}) = -K_\infty x - q_t(\boldsymbol{w})$ and $\widehat{\pi}_t(x; \boldsymbol{w}) = -K_t x - q_t(\boldsymbol{w})$ be given, where $q_t$ is arbitrary but satisfies $\|q_t\| \le D_q$ for some $D_q \ge 1$. Then

$$|J_T(\widehat{\pi}, \boldsymbol{w}) - J_T(\pi, \boldsymbol{w})| \le C_{K_\infty},$$

where $C_{K_\infty} \le \widetilde{\mathcal{O}}\big(\kappa_\infty^4 \beta_\star^6 \Psi_\star^{13} \Gamma_\star^6 (1 - \gamma_\infty)^{-2} D_q^2 \cdot \log(D_q T)\big)$.

**Lemma 3.5.** Let $(q_t)_{t=1}^T$ be an arbitrary sequence with $\|q_t\| \le D_q$ for some $D_q \ge D_{q^\star}$. Then it holds that

$$\left| \sum_{t=1}^{T} \|q_t - q^\star_{t;t+h}(w_{t:t+h})\|^2_{\Sigma_t} - \|q_t - q^\star_{\infty;h}(w_{t:t+h})\|^2_{\Sigma_\infty} \right| \le C_\infty,$$

where $C_\infty \le \widetilde{\mathcal{O}}\big(D_q^2 \cdot \beta_\star^4 \Psi_\star^7 \Gamma_\star^4 \kappa_\infty^2 (1 - \gamma_\infty)^{-1} h \log(D_q T)\big)$.

Combining these results immediately yields the proof of Theorem 2; details are given in Appendix F.

## 4. Conclusion

We have presented the first efficient algorithm with logarithmic regret for online linear control with arbitrary adversarial disturbance sequences. Our result highlights the power of online learning with advantages, and we are hopeful that this framework will find broader use. Numerous questions naturally arise for future work: Does our framework extend to more general loss functions, or to more general classes of dynamical systems in control and reinforcement learning? Can our results be extended to handle partial observed dynamical systems? Can we obtain $\sqrt{T}$-regret for adversarial disturbances in unknown systems, as is possible in the stochastic regime?

### Acknowledgements

DF acknowledges the support of NSF TRIPODS award #1740751. MS is generously supported by an Open Philanthropy AI Fellowship. We thank Ali Jadbabaie for helpful discussions.

## References

Abbasi-Yadkori, Y. and Szepesvári, C. 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, Y., Bartlett, P. L., Kanade, V., Seldin, Y., and Szepesvári, C. 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, Y., Bartlett, P., and Kanade, V. Tracking adversarial targets. In *International Conference on Machine Learning*, pp. 369–377, 2014.

Agarwal, N., Bullins, B., Hazan, E., Kakade, S., and Singh, K. Online control with adversarial disturbances. In *International Conference on Machine Learning*, pp. 111–119, 2019a.

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

Anava, O., Hazan, E., and Mannor, S. Online learning for adversaries with memory: price of past mistakes. In *Advances in Neural Information Processing Systems*, pp. 784–792, 2015.

Arora, R., Dekel, O., and Tewari, A. Online bandit learning against an adaptive adversary: from regret to policy regret. In *International Conference on Machine Learning (ICML)*, pp. 1747–1754, 2012.

Azoury, K. S. and Warmuth, M. K. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, June 2001.

Bertsekas, D. P. *Dynamic Programming and Optimal Control, Vol. I.* Athena Scientific, 2005.

Cassel, A., Cohen, A., and Koren, T. Logarithmic regret for learning linear quadratic regulators efficiently. *International Conference on Machine Learning (ICML)*, 2020.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Cohen, A., Hasidim, A., Koren, T., Lazic, N., Mansour, Y., and Talwar, K. Online linear quadratic control. In *International Conference on Machine Learning*, pp. 1028–1037, 2018.

Cohen, A., Koren, T., and Mansour, Y. Learning linear-quadratic regulators efficiently with only $\sqrt{T}$ regret. In *International Conference on Machine Learning*, pp. 1300–1309, 2019.

Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. Regret bounds for robust adaptive control of the linear quadratic regulator. In *Advances in Neural Information Processing Systems*, pp. 4188–4197, 2018.

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

Faradonbeh, M. K. S., Tewari, A., and Michailidis, G. On optimality of adaptive linear-quadratic regulators. *arXiv preprint arXiv:1806.10749*, 2018.

Fazel, M., Ge, R., Kakade, S., and Mesbahi, M. Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, pp. 1467–1476, 2018.

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

Hazan, E. and Kale, S. Newtron: an efficient bandit algorithm for online multiclass prediction. In *Advances in Neural Information Processing Systems*, pp. 891–899, 2011.

Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Hazan, E., Kakade, S., and Singh, K. The nonstochastic control problem. In *Proceedings of the 31st International Conference on Algorithmic Learning Theory*, pp. 408–421. PMLR, 2020.

Joulani, P., Gyorgy, A., and Szepesvári, C. Online learning under delayed feedback. In *International Conference on Machine Learning*, pp. 1453–1461, 2013.

Kakade, S. M. *On the sample complexity of reinforcement learning*. PhD thesis, University College London (University of London), 2003.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2016.

Lincoln, B. and Rantzer, A. Relaxing dynamic programming. *IEEE Transactions on Automatic Control*, 51(8): 1249–1260, 2006.

Mania, H., Tu, S., and Recht, B. Certainty equivalence is efficient for linear quadratic control. In *Advances in Neural Information Processing Systems*, pp. 10154–10164, 2019.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529, 2015.

Orabona, F., Crammer, K., and Cesa-Bianchi, N. A generalized online mirror descent with applications to classification and regression. *Machine Learning*, 99(3):411–435, 2015.

Rakhlin, A. and Sridharan, K. Online nonparametric regression. In *Conference on Learning Theory*, 2014.

Rosenberg, A. and Mansour, Y. Online convex optimization in adversarial markov decision processes. In *International Conference on Machine Learning*, pp. 5478–5486, 2019.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

Simchowitz, M. and Foster, D. J. Naive exploration is optimal for online LQR. *International Conference on Machine Learning (ICML)*, 2020.

Simchowitz, M., Singh, K., and Hazan, E. Improper learning for non-stochastic control. *Conference on Learning Theory (COLT)*, 2020.

Slotine, J.-J. E. and Li, W. *Applied nonlinear control*. Prentice-Hall, 1991.

Vovk, V. Aggregating strategies. *Proceedings of the conference on computational Learning Theory*, 1990.

Vovk, V. A game of prediction with expert advice. In *Proceedings of the eighth annual conference on computational learning theory*, pp. 51–60. ACM, 1995.

Vovk, V. Competitive on-line linear regression. In *NIPS '97: Proceedings of the 1997 conference on advances in neural information processing systems 10*, pp. 364–370, Cambridge, MA, USA, 1998. MIT Press.

Yu, J. Y., Mannor, S., and Shimkin, N. Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research*, 34(3):737–757, 2009.

Zimin, A. and Neu, G. Online learning in episodic markovian decision processes by relative entropy policy search. In *Advances in neural information processing systems*, pp. 1583–1591, 2013.