Non-Stochastic Control with Bandit Feedback

Paula Gradu[1,3] **John Hallman**[1,3] **Elad Hazan**[2,3]
[1] Department of Mathematics, Princeton University
[2] Department of Computer Science, Princeton University
[3] Google AI Princeton
{pgradu,hallman,ehazan}@princeton.edu

Abstract

We study the problem of controlling a linear dynamical system with adversarial perturbations where the only feedback available to the controller is the scalar loss, and the loss function itself is unknown. For this problem, with either a known or unknown system, we give an efficient sublinear regret algorithm. The main algorithmic difficulty is the dependence of the loss on past controls. To overcome this issue, we propose an efficient algorithm for the general setting of bandit convex optimization for loss functions with memory, which may be of independent interest.

1 Introduction

The fields of Reinforcement Learning (RL), as well as its differentiable counterpart of Control, formally model the setting of learning through interaction in a reactive environment. The crucial component in RL/control that allows learning is the feedback, or reward/penalty, which the agent iteratively observes and reacts to.

While some signal is necessary for learning, different applications have different feedback to the learning agent. In many reinforcement learning and control problems it is unrealistic to assume that the learner has feedback for actions other than their own. One example is in game-playing, such as the game of Chess, where a player can observe the adversary's move for their own choice of play, but it is unrealistic to expect knowledge of the adversary's play for any possible move. This type of feedback is commonly known in the learning literature as "bandit feedback".

Learning in Markov Decision Processes (MDP) is a general and difficult problem for which there are no known algorithms that have sublinear dependence on the number of states. For this reason we look at structured MDPs, and in particular the model of control in Linear Dynamical Systems (LDS), a highly structured special case that is known to admit more efficient methods as compared to general RL.

In this paper we study learning in linear dynamical systems with bandit feedback. This generalizes the well-known Linear Quadratic Regulator to systems with only bandit feedback over any convex loss function. Further, our results apply to the non-stochastic control problem which allows for adversarial perturbations and adversarially chosen loss functions, even when the underlying linear system is unknown.

1.1 Our Results

We give the first sublinear regret algorithm for controlling a linear dynamical system with bandit feedback in the non-stochastic control model. Specifically, we consider the case in which the underlying system is linear, but has potentially adversarial perturbations (that can model deviations

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

from linearity), i.e.

$$x_{t+1} = Ax_t + Bu_t + w_t, \tag{1.1}$$

where $x_t \in \mathbb{R}^n$ is the (observed) dynamical state, $u_t \in \mathbb{R}^m$ is a learner-chosen control and $w_t \in \mathbb{R}^n$ is an adversarial perturbation. The goal of the controller is to minimize a sum of sequentially revealed adversarial cost functions $c_t(x_t, u_t)$ over the state-control pairs that it visits. More precisely, the goal of the learner in this adversarial setting is to minimize regret compared to a class of policies Π:

$$\text{Regret} = \sum_{t=1}^{T} c_t(x_t, u_t) - \min_{\pi \in \Pi} \sum_{t=1}^{T} c_t(x_t^\pi, u_t^\pi),$$

where the cost of the benchmark is measured on the counterfactual state-action sequence (x_t^π, u_t^π) that the benchmark policy in consideration visits, as opposed to the state-sequence visited by the the learner. The target class of policies we compare against in this paper are disturbance action controllers (DAC), whose control is a linear function of past disturbances plus a stabilizing linear operator over the current state $u_t = Kx_t + \sum_{i=1}^{H} M_i w_{t-i}$, for some history-length parameter H. This comparator class is known to be more general than state-feedback laws and linear dynamical controllers (LDC). This choice is a consequence of recent advances in convex relaxation for control [4, 5, 16, 32].

For the setting we consider, the controller can only observe the scalar $c_t(x_t, u_t)$, and **does not have access to the gradients or any other information about the loss**. Our main results are efficient algorithms for the non-stochastic control problem which attain the following guarantees:

Theorem 1.1 (Informal Statement)**.** *For a **known** linear dynamical system where the perturbations w_t (and convex costs c_t) are bounded and chosen by an adversary, there exists an efficient algorithm that with bandit feedback generates an adaptive sequence of controls $\{u_t\}$ for which*

$$\textit{Regret} = \widetilde{\mathcal{O}}(\mathrm{poly}(\texttt{natural-parameters})T^{3/4}).$$

This theorem can be further extended to unknown systems:

Theorem 1.2 (Informal Statement)**.** *For an **unknown** linear dynamical system where the perturbations w_t (and convex costs c_t) are bounded and chosen by an adversary, there exists an efficient algorithm that with bandit feedback generates an adaptive sequence of controls $\{u_t\}$ for which*

$$\textit{Regret} = \widetilde{\mathcal{O}}(\mathrm{poly}(\texttt{natural-parameters})T^{3/4}).$$

Techniques. To derive these results, we combine the convex relaxation technique of [4] with the non-stochastic system identification method for environments with adversarial perturbations from [16, 30]. However, the former result relies on gradient based optimization methods, and it is non-trivial to apply gradient estimation techniques in this black-box zero-order information setting. The main difficulty stems from the fact that the gradient-based methods from non-stochastic control apply to *functions with memory*, and depend on the system state going back many iterations. The natural way of creating unbiased gradient estimates, such as in [14], have no way of accounting for functions with memory.

To solve this difficulty, we introduce an efficient algorithm for the setting of *bandit convex optimization with memory*. This method combines the gradient-based methods of [6] with the unbiased gradient estimation techniques of [14]. The naive way of combining these techniques introduces time dependencies between the random gradient estimators, as a direct consequence of the memory in the loss functions. To resolve this issue, we introduce an artificial intentional delay to the gradient updates and show that this delay has only a limited effect on the overall regret.

Paper outline. After describing related work, we cover preliminaries and define notation in section 2. In section 3 we describe the algorithm for BCO with memory and the main theorem regarding its performance. We then introduce the bandit control setting in section 4, and provide algorithms for known and unknown systems in sections 5 and 6 respectively, together with relevant theoretical results. We then present experimental results in section 7.

1.2 Related Work

Reinforcement learning with bandit feedback. Online learning techniques for reinforcement learning were studied in [11] and generalized in [34]. Online learning for RL with bandit feedback was studied in [24]. For general RL it is impossible to obtain regret bounds that are sublinear in the number of states, even with full feedback. This is the reason we focus on much more structured problem of control, where our regret bounds depend on the dimension despite an infinite number of states, even in the bandit setting.

Robust Control: The classical control literature deals with adversarial perturbations in the dynamics in a framework known as H_∞ control, see e.g. [33, 35]. In this setting, the controller solves for the best linear controller assuming worst case noise to come. This is different from the setting we study which minimizes regret on a per-instance basis.

Learning to control stochastic LDS: There has been a resurgence of literature on control of linear dynamical systems in the recent machine learning venues. The case of known systems was extensively studied in the control literature, see the survey [33]. Sample complexity and regret bounds for control (under Gaussian noise) were obtained in [3, 10, 2, 23, 9, 21, 20, 22]. The works of [1], [8] and [5] allow for control in LDS with adversarial loss functions. Provable control in the Gaussian noise setting via the policy gradient method was studied in [13]. These works operate in the absence of perturbations or assume that they are i.i.d. Gaussian, as opposed to adversarial which is what we consider. Other relevant work from the machine learning literature includes spectral filtering techniques for learning and open-loop control of partially observable systems [18, 7, 17].

Non-stochastic control: Regret minimization for control of dynamical systems with adversarial perturbations was initiated in the recent work of [4], who use online learning techniques and convex relaxation to obtain provable bounds for controlling LDS with adversarial perturbations. These techniques were extended in [5] to obtain logarithmic regret under stochastic noise, in [16] for the control of unknown systems, and in [32] for control of systems with partially observed states.

System identification. For the stochastic setting, several works [12, 31, 28] propose to use the least-squares procedure for parameter identification. In the adversarial setting, least-squares can lead to inconsistent estimates. For the partially observed stochastic setting, [25, 29, 31] give results guaranteeing parameter recovery using Gaussian inputs. Provable system identification in the adversarial setting was obtained in [30, 16].

2 Preliminaries

Online convex optimization with memory. The setting of online convex optimization (OCO) efficiently models iterative decision making. A player iteratively choses an action from a convex decision set $x_t \in \mathcal{K} \subseteq \mathbb{R}^d$, and suffers a loss according to an adversarially chosen loss function $f_t(x_t)$. In the bandit setting of OCO, called Bandit Convex Optimization (BCO), the only information available to the learner after each iteration is the loss value itself, a scalar, and no other information about the loss function f_t.

A variant which is relevant to our setting of control is BCO with memory. This is used to capture time dependence of the reactive environment. Here, the adversaries pick loss functions f_t with bounded memory H of our previous predictions, and as before we assume that we may observe the value but have no access to the gradient of our losses f_t. The goal is to minimize regret, defined as:

$$\text{Regret} = \mathop{\mathbb{E}}_{\mathcal{R}_\mathcal{A}} \left[\sum_{t=H}^{T} f_t(x_{t-\bar{H}:t}) \right] - \min_{x^\star \in \mathcal{K}} \sum_{t=H}^{T} f_t(x^\star, \ldots, x^\star),$$

where we denote $\bar{H} = H - 1$ and $x_{t-\bar{H}:t} = (x_{t-\bar{H}}, \ldots, x_t)$ for clarity, $x_1, \ldots, x_T$ are the predictions of algorithm $\mathcal{A}$, and $\mathcal{R}_\mathcal{A}$ represents the randomness due to the algorithm $\mathcal{A}$.

For the settings of Theorem 3.1, we assume that the loss functions f_t are convex with respect to $x_{t-\bar{H}:t}$, G-Lipschitz, β-smooth, and bounded. We can assume without loss of generality that the loss functions are bounded by 1 in order to simplify computations. In the case where the functions are

bounded by some $|f_t(x_{t-\bar{H}:t})| \leq M$, one can obtain the same results with an additional factor M in the regret bounds by dividing the gradient estimator by M.

3 An Algorithm for BCO with Memory

This section describes the main building block for our control methods: an algorithm for BCO with memory. Our algorithm takes a non-increasing sequence of learning rates $\{\eta_t\}_{t=1}^T$ and a *perturbation constant* δ, a hyperparameter associated with the gradient estimator. Note that the algorithm projects x_t onto the Minkowski subset $\mathcal{K}_\delta = \{x \in \mathcal{K} : \frac{1}{1-\delta} x \in \mathcal{K}\}$ to ensure that $y_t = x_t + \delta u_t \in \mathcal{K}$ holds.

Algorithm 1 BCO with Memory

Input: $\mathcal{K}$, T, H, $\{\eta_t\}$ and δ
Initialize $x_1 = \cdots = x_H \in \mathcal{K}_\delta$ arbitrarily
Sample $u_1, \ldots, u_H \in_{\mathbf{R}} \mathbb{S}_1^d$
Set $y_i = x_i + \delta u_i$ for $i = 1, \ldots, H$
Set $g_i = 0$ for $i = 1, \ldots, \bar{H}$
Predict y_i for $i = 1, \ldots, \bar{H}$
for $t = H, \ldots, T$ **do**
    predict y_t
    suffer loss $f_t(y_{t-\bar{H}:t})$
    store $g_t = \frac{d}{\delta} f_t(y_{t-\bar{H}:t}) \sum_{i=0}^{\bar{H}} u_{t-i}$
    set $x_{t+1} = \prod_{\mathcal{K}_\delta} \left[x_t - \eta_t \, g_{t-\bar{H}} \right]$
    sample $u_{t+1} \in_{\mathrm{R}} \mathbb{S}_1^d$
    set $y_{t+1} = x_{t+1} + \delta u_{t+1}$
end for
return

The main performance guarantee for this algorithm is given in the following theorem, whose complete proof can be found in section A of the appendix.

Theorem 3.1. *Setting step sizes $\eta_t = \Theta(t^{-3/4} H^{-3/2} d^{-1} D^{2/3} G^{-2/3} \beta^{-1/2})$ and perturbation constant $\delta = \Theta(T^{-1/4} H^{-1/2} D^{1/3} G^{-1/3})$, Algorithm 1 produces a sequence $\{y_t\}_{t=0}^T$ that satisfies:*

$$\textit{Regret} \leq \mathcal{O}\left(T^{3/4} H^{3/2} d D^{4/3} G^{2/3} \beta^{1/2}\right)$$

In particular, Regret $\leq \mathcal{O}\left(T^{3/4}\right)$.

4 Application to Online Control of LDS

In this section we introduce the setting of online bandit control and relevant assumptions, with the objective of converting our control problem to one of BCO with memory, which will allow us to use Algorithm 1 to control linear dynamical systems using only bandit feedback.

In online control, the learner iteratively observes a state x_t, chooses an action u_t, and then suffers a convex cost $c_t(x_t, u_t)$ selected by an adversary. We assume for simplicity of analysis that $x_0 = 0$. Since the adversary can set w_0 arbitrarily, this does not change the generality of this setting. Because we are working in the bandit setting, we may only observe the value of $c_t(x_t, u_t)$ and have no access to the function c_t itself. Therefore, the learner cannot apply c_t to a different set of inputs, nor take gradients over them. As such, previous approaches to non-stochastic control such as in [4, 16] are no longer viable, as these rely on the learner being capable of executing both of these operations.

Assumptions. From hereon out, we assume that the perturbations are bounded, i.e. $\|w_t\| \leq W$, and that all x_t's and u_t's are bounded such that $\|x_t\|, \|u_t\| \leq D$. We additionally bound the norm of the dynamics $\|A\| \leq \kappa_A, \|B\| \leq \kappa_B$, and assume that the cost functions c_t are G-Lipschitz and β-smooth.

As in the existing literature, we measure our performance against the class of disturbance action controllers. Before we introduce this wider policy class, let us restate the definition of strong stability:

Definition 4.1. *A linear policy is (κ, γ)-strongly stable if there exist matrices L, H satisfying $A - BK = HLH^{-1}$, such that $||L|| \leq 1 - \gamma$ and $\max\left(||K||, ||H||, ||H^{-1}||\right) \leq \kappa$.*

Definition 4.2. ***(Disturbance Action Controller)*** *A disturbance action controller is parametrized by a sequence of H matrices $M = [M^{[i]}]_{i=1}^{H}$ and a (κ, γ)-strongly stable K, and chooses actions according to $u_t = -Kx_t + \sum_{s=1}^{H} M^{[s]} w_{t-s}$.*

DAC Policy Class. We define $\mathcal{M}$ to be the set of all disturbance action controllers (for a fixed H and K) with geometrically decreasing component norms, i.e. $\mathcal{M} \doteq \{M \text{ s.t. } ||M^{[i]}|| \leq \kappa^3 \kappa_B (1-\gamma)^i\}$.

Performance metric. For algorithm $\mathcal{A}$ that goes through the states $x_0, \ldots, x_T$, selects actions $u_0, \ldots, u_T$, and observes the sequence of perturbations $w = (w_0, \ldots, w_T)$, we define the expected total cost over any randomness in the algorithm given the observed disturbances to be

$$J_T(\mathcal{A}|w) = \mathop{\mathbb{E}}_{\mathcal{A}}\left[\sum_{t=0}^{T} c_t(x_t, u_t)\right].$$

With some slight abuse of notation, we will use $J_T(M|w)$ to denote the cost of the fixed DAC policy that chooses $u_t = -Kx_t + \sum_{s=1}^{H} M^{[s]} w_{t-s}$ and observes the same perturbation sequence w. Following the literature on non-stochastic control, our metric of performance is regret, which for an algorithm $\mathcal{A}$ is defined as:

$$\text{Regret} = \sup_{w_{1:T}}\left[J_T(\mathcal{A}|w) - \min_{M \in \mathcal{M}}\left[J_T(M|w)\right]\right]. \tag{4.1}$$

5 Non-stochastic control of known systems

We now give an algorithm for controlling known time-invariant linear dynamical systems in the bandit setting. Our approach is to design a disturbance action controller and to train it using our algorithm for BCO with memory. Formally, at time t we choose the action $u_t = -Kx_t + \sum_{i=1}^{H} M_t^{[i]} w_{t-i}$ where $M_t = \{M_t^{[1]}, \ldots, M_t^{[H]}\} \in \mathbb{R}^{H \times m \times n}$ are the learnable parameters and we denote $w_t = 0, \ \forall t < 0$, for convenience. Note that K does not update over time, and only exists to make sure that the system remains stable under the initial policy.

In order to train these controllers in the bandit setting, we identify the costs $c_t(x_t, u_t)$ with a loss function with memory that takes as input the past H controllers $M_{t-\bar{H}}, \ldots, M_t$, and apply our results from Algorithm 1. We denote the corresponding Minkowski subset of $\mathcal{M}$ by $\mathcal{M}_\delta$. Our algorithm is given below, and the main performance guarantee for it is given in Theorem 5.1, whose complete proof is deferred to section B of the Appendix.

Theorem 5.1. *If we set η_t and δ as in theorem 3.1 and $H = \Theta(\log T)$, the regret incurred by Algorithm 2 satisfies*

$$\text{Regret} \leq \mathcal{O}\left(T^{3/4} \log^{5/2} T\right).$$

6 Non-stochastic control of unknown systems

We now extend our algorithm to unknown systems, yielding a controller that achieves **sublinear regret for both unknown costs and unknown dynamics** in the non-stochastic adversarial setting. The main challenge in this scenario is that we are competing with the best policy that has access to the true dynamics. Moreover, if we don't know the the system, we are also unable to deduce the true w_t's. While this initially may appear to be specially problematic for the class of disturbance action controllers, we show that it is still possible to attain sublinear regret.

Algorithm 2 Bandit Perturbation Controller

Input: $K, H, T, \{\eta_t\}, \delta$, and $\mathcal{M}$
Initialize $M_0 = \cdots = M_{\bar{H}} \in \mathcal{M}_\delta$ arbitrarily
Sample $\epsilon_0, \ldots, \epsilon_{\bar{H}} \in_{\mathbb{R}} \mathbb{S}_1^{H \times m \times n}$
Set $\widetilde{M}_i = M_i + \epsilon_i$ for $i = 0, \ldots, \bar{H}$
Set $g_i = 0$ for $i = -\bar{H}, \ldots, 0, \ldots, \bar{H}$
for $t = 0, \ldots, T$ **do**
  choose action $u_t = -Kx_t + \sum_{i=1}^{H} \widetilde{M}_t^{[i]} w_{t-i}$
  suffer loss $c_t(x_t, u_t)$
  observe new state x_{t+1}
  record $w_t = x_{t+1} - Ax_t - Bu_t$
  store $g_t = \dfrac{mnH}{\delta} c_t(x_t, u_t) \sum_{i=0}^{\bar{H}} \epsilon_{t-i}$ if $t \geq H$ else 0
  set $M_{t+1} = \Pi_{\mathcal{M}_\delta}\left[M_t - \eta_t \, g_{t-\bar{H}}\right]$
  sample $\epsilon_{t+1} \in_{\mathrm{R}} \mathbb{S}_1^{H \times m \times n}$
  set $\widetilde{M}_{t+1} = M_{t+1} + \delta \epsilon_{t+1}$
end for
return

6.1 System identification via method of moments

Our approach to control of unknown systems follows the explore-then-commit paradigm, identifying the underlying dynamics up to some desirable accuracy using random inputs in the exploration phase, followed by running Algorithm 2 on the estimated dynamics. The approximate system dynamics allow us to obtain estimates of the disturbances, thus facilitating the execution of the disturbance action controller. The procedure used to estimate the system dynamics is given in Algorithm 3.

One property we need is strong controllability, as defined by [8]. Controllability for a linear system is characterized by the ability to drive the system to any desired state through appropriate control inputs in the presence of deterministic dynamics, i.e. when the perturbations w_t are 0. For a discussion of how to relax this requirement see remark C.2 in the appendix.

Definition 6.1. *A linear dynamical system with dynamics matrices A, B is controllable with controllability index $k \geq 1$ if the matrix*

$$C_k = \left[B, AB, \ldots, A^{k-1}B\right] \in \mathbb{R}^{n \times km}$$

has full row-rank. In addition, such as system is also considered (k, κ)-strongly controllable if $\left\|(C_k C_k^\top)^{-1}\right\| \leq \kappa$.

In order to prove regret bounds in the setting of unknown systems, we must ensure that the system remains somewhat controllable during the exploration phase, which we do by introducing the following assumptions which are slightly stronger than the ones required in the known system setting:

Assumption 6.2. *We assume that the perturbation sequence is chosen at the start of the interaction, implying that this sequence w_t does not depend on the choice of u_t.*

Assumption 6.3. *The learner knows a linear controller $\mathbb{K}$ that is (κ, γ)-strongly stable for the true, but unknown, transition matrices (A, B) defining the dynamical system.*

Assumption 6.4. *The linear dynamical system $(A - B\mathbb{K}, B)$ is (k, κ)-strongly controllable.*

Note then that $\mathbb{K}$ is any stabilizing controller ensuring that the system remains controllable under the random actions, and k the controllability index of the system.

6.2 The algorithm and regret guarantee

Combining Algorithm 2 with the system identification method in Algorithm 3, we obtain the following algorithm for the control of unknown systems.

The performance guarantee for Algorithm 4 is given in the following theorem, with the proof deferred to section C of the Appendix. Note that $\hat{\delta}$ is the probability of failure for Algorithm 3.

Algorithm 3 System identification via random inputs

Input: $T_0, \mathbb{K}$
for $t = 0, \dots, T_0$ **do**
  sample $\xi_t \in_R \{\pm 1\}^m$
  choose action $u_t = -\mathbb{K}x_t + \xi_t$
  Incur loss $c_t(x_t, u_t)$, record x_t
end for
set $N_j = \frac{1}{T_0 - k} \sum_{t=0}^{T_0-k-1} x_{t+j+1}\xi_t^T$ for all j in $[k]$
Let $C_0 = (N_0, \dots N_{k-1})$, $C_1 = (N_1, \dots N_k)$
set $\hat{A} = C_1 C_0^T (C_0 C_0^T)^{-1} + N_0 K$ and $\hat{B} = N_0$

Algorithm 4 BPC with system identification

Input: $H, T_0, T, \{\eta_t\}, \delta, \mathcal{M}, \mathbb{K}, K$
Phase 1: Run Algorithm 3 with a budget of T_0 to obtain system estimates $\hat{A}$, $\hat{B}$
Phase 2: Run Algorithm 2 with the dynamics $\hat{A}, \hat{B}$ for $T - T_0$ timesteps, and $\hat{w}_{T_0} = x_{T_0+1}$

Theorem 6.5. *If our system satisfies the assumptions put forth, setting* $T_0 = \Theta\left(T^{2/3} \log \hat{\delta}^{-1}\right)$, $\hat{\delta} = \Theta(T^{-1})$, *and* η_t, δ, *and* H *as in Theorem 5.1, we have that the regret incurred by Algorithm 4 satisfies*

$$\text{Regret} \leq \mathcal{O}\left(T^{3/4} \log^{5/2} T\right).$$

7 Experimental Results

We now provide empirical results of our algorithms' performance on different dynamical systems and under various noise distributions. In all figures, we average the results obtained over 25 runs and include the corresponding confidence intervals. Our algorithm implementation is available at [26].

7.1 Control with known dynamics

We first evaluate our Algorithm 2 (BPC) while comparing to GPC [4] (which has full access to the cost functions), as well as the baseline method Linear Quadratic Regulator (LQR) [19]. For both BPC and GPC we initialize K to be the infinite-horizon LQR solution given dynamics A and B in all of the settings below in order to observe the improvement provided by the two perturbation controllers relative to the classical approach.

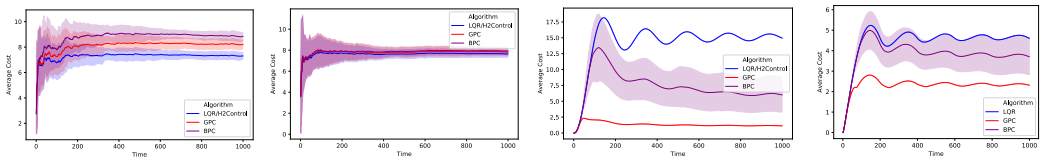
We consider four different loss functions:

1. L_2^2-norm: $c_t(x, u) = ||x||^2 + ||u||^2$ (also known as *quadratic cost*),
2. L_1-norm: $c_t(x, u) = ||x||_1 + ||u||_1$,
3. L_∞-norm: $c_t(x, u) = ||x||_\infty + ||u||_\infty$,
4. ReLU: $c_t(x, u) = ||\max(0, x)||_1 + ||\max(0, u)||_1$ (each max taken element-wise).

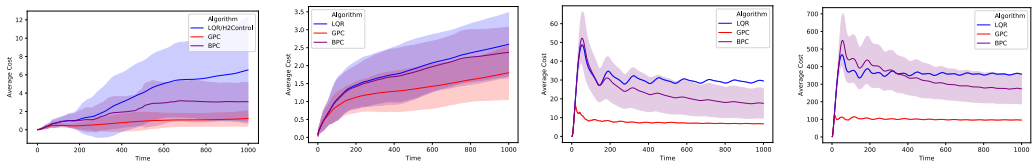
We run the algorithms on two different linear dynamical systems, the $n = 2, m = 1$ double integrator system defined by $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ and $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, as well as one additional setting on a larger LDS with $n = 5, m = 3$ for sparse but non-trivial A and B. We analyze the performance of our algorithms for the following 3 noise specifications.

1. **Sanity check.** We run our algorithms with i.i.d Gaussian noise terms $w_t \sim \mathcal{N}(0, I)$. We see that decaying learning rates allow the GPC and BPC to converge to the LQR solution which is optimal for this setup.
2. **Sinusoidal noise.** In this setup, we look at the sinusoidal $w_t = sin(t/(20\pi))$. In this correlated noise setting, the LQR policy is sub-optimal, and we see that both BPC and GPC outperform it.

3. **Gaussian random walk.** In the Gaussian random walk setting, each noise term is distributed normally, with the previous noise term as its mean, i.e. $w_{t+1} = \mathcal{N}\left(w_t, \frac{1}{T}\right)$. Since $T = 1000$, we have approximately that $w_{t+1} - w_t \sim \mathcal{N}(0, 0.3^2)$.



(a) Sanity check with fixed learning rate. (b) Sanity check with decaying learning rate. (c) Sinusoidal noise and quadratic costs. (d) Sinusoidal noise and L_1 costs.



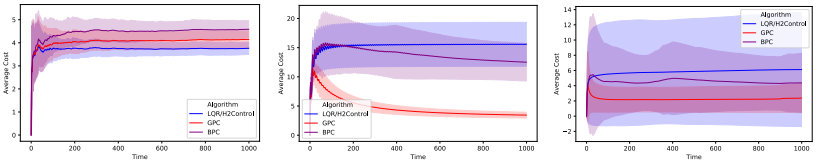
(e) Random walk noise on simple LDS and quadratic costs. (f) Random walk noise on simple LDS and L_1 costs. (g) Sinusoidal noise on complex LDS and L_∞ cost. (h) Sinusoidal noise on complex LDS and ReLU costs.

Figure 1: Known dynamics, small and large LDS setting.

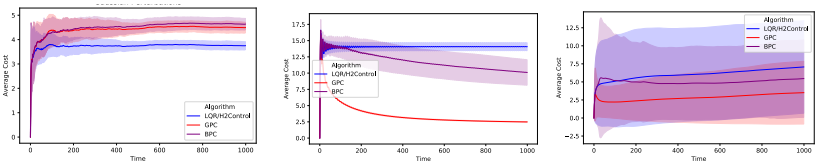
7.2 Control with unknown dynamics

Next, we evaluate Algorithm 4 on unknown dynamical systems. We obtain estimates $\hat{A}$ and $\hat{B}$ of the system dynamics using two different types of system identification methods, the first being the method described in Algorithm 3, and the second being regular linear regression based on all observations up to the current time point. We then proceed with the experiments as in the previous section, with all algorithms being given $\hat{A}$ and $\hat{B}$ instead of the true A, B. That is, LQR produces policy $\hat{K}$ based on the solution of the algebraic Riccati equation given by $\hat{A}$ and $\hat{B}$, and both BPC and GPC start from this initial $\hat{K}$ and obtain estimates of the disturbances $\hat{w}_t$ based on the approximate dynamics.

We run experiments with quadratic costs for the first LDS described in the known dynamics section, with scaled down dynamics matrices A and B such that their nuclear norm is strictly less than 1 so that the dynamical system remains stable during the identification phase. The system identification phase is repeated for each experiment and runs for $T_0 = 5000$ time steps and with initial control matrix K set to 0.



(a) Sanity check (b) Sinusoidal noise (c) Random walk noise



(d) Sanity check (e) Sinusoidal noise (f) Random walk noise

Figure 2: Unknown dynamics, the top row uses the system identification method in Algorithm 3, and the bottom row uses linear regression.

8 Conclusions and Open Questions

We have considered the non-stochastic control problem with the additional difficulty of learning with only bandit feedback. We give an efficient method with sublinear regret for this challenging problem in the case of linear dynamics based upon a new algorithm for bandit convex optimization with memory, which may be of independent interest. The application of bandit optimization to control is complicated due to time dependency issues, which required introducing an artificial delay in our online learning method.

The setting of control with general convex losses was proposed in 1987 by Tyrrell Rockafellar [27] in order to handle constraints on state and control. It remains open to add constraints (such as safety constaints) to online nonstochastic control. Other questions that remain open are quantitative: the worst case attainable regret bounds can be potentially improved to $\sqrt{T}$. The dependence on the system dimensions can also be tightened.

Broader Impact

The purpose of this work is to contribute to scientific progress in control theory. We do not expect any unethical use of this work.

Acknowledgments and Disclosure of Funding

EH acknowledges support of NSF grant # 1704860. All work was done while PG, JH and EH were employed at Google.

References

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

[2] Yasin Abbasi-Yadkori, Nevena Lazic, and Csaba Szepesvári. Model-free linear quadratic control via reduction to expert prediction. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3108–3117, 2019.

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

[4] Naman Agarwal, Brian Bullins, Elad Hazan, Sham M. Kakade, and Karan Singh. Online control with adversarial disturbances, 2019.

[5] Naman Agarwal, Elad Hazan, and Karan Singh. Logarithmic regret for online control. *arXiv preprint arXiv:1909.05062*, 2019.

[6] Oren Anava, Elad Hazan, and Shie Mannor. Online convex optimization against adversaries with memory and application to statistical arbitrage. *arXiv preprint arXiv:1302.6937*, 2013.

[7] Sanjeev Arora, Elad Hazan, Holden Lee, Karan Singh, Cyril Zhang, and Yi Zhang. Towards provable control for unknown linear dynamical systems. *International Conference on Learning Representations*, 2018. rejected: invited to workshop track.

[8] Alon Cohen, Avinatan Hassidim, Tomer Koren, Nevena Lazic, Yishay Mansour, and Kunal Talwar. Online linear quadratic control. *arXiv preprint arXiv:1806.07104*, 2018.

[9] Alon Cohen, Tomer Koren, and Yishay Mansour. Learning linear-quadratic regulators efficiently with only $\sqrt{(}t)$ regret. *arXiv preprint arXiv:1902.06223*, 2019.

[10] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. Regret bounds for robust adaptive control of the linear quadratic regulator. In *Advances in Neural Information Processing Systems*, pages 4188–4197, 2018.

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

[12] Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. Finite time identification in unstable linear systems. *Automatica*, 96:342–353, 2018.

[13] Maryam Fazel, Rong Ge, Sham M Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. *arXiv preprint arXiv:1801.05039*, 2018.

[14] Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient, 2004.

[15] Elad Hazan. The convex optimization approach to regret minimization. *Optimization for machine learning*, 2012.

[16] Elad Hazan, Sham M Kakade, and Karan Singh. The nonstochastic control problem. *arXiv preprint arXiv:1911.12178*, 2019.

[17] Elad Hazan, Holden Lee, Karan Singh, Cyril Zhang, and Yi Zhang. Spectral filtering for general linear dynamical systems. In *Advances in Neural Information Processing Systems*, pages 4634–4643, 2018.

[18] Elad Hazan, Karan Singh, and Cyril Zhang. Learning linear dynamical systems via spectral filtering. In *Advances in Neural Information Processing Systems*, pages 6702–6712, 2017.

[19] Rudolf E Kalman. On the general theory of control systems. In *Proceedings First International Conference on Automatic Control, Moscow, USSR*, 1960.

[20] Sahin Lale, Kamyar Azizzadenesheli, Babak Hassibi, and Anima Anandkumar. Logarithmic regret bound in partially observable linear dynamical systems, 2020.

[21] Sahin Lale, Kamyar Azizzadenesheli, Babak Hassibi, and Anima Anandkumar. Regret bound of adaptive control in linear quadratic gaussian (lqg) systems, 2020.

[22] Sahin Lale, Kamyar Azizzadenesheli, Babak Hassibi, and Anima Anandkumar. Regret minimization in partially observable linear quadratic control, 2020.

[23] Horia Mania, Stephen Tu, and Benjamin Recht. Certainty equivalent control of lqr is efficient. *arXiv preprint arXiv:1902.07826*, 2019.

[24] Gergely Neu, Andras Antos, András György, and Csaba Szepesvári. Online markov decision processes under bandit feedback. In *Advances in Neural Information Processing Systems*, pages 1804–1812, 2010.

[25] Samet Oymak and Necmiye Ozay. Non-asymptotic identification of lti systems from a single trajectory. In *2019 American Control Conference (ACC)*, pages 5655–5661. IEEE, 2019.

[26] Google AI Princeton. Deluca. https://github.com/MinRegret/deluca, 2020.

[27] R Tyrell Rockafellar. Linear-quadratic programming and optimal control. *SIAM Journal on Control and Optimization*, 25(3):781–814, 1987.

[28] Tuhin Sarkar and Alexander Rakhlin. Near optimal finite time identification of arbitrary linear dynamical systems. In *International Conference on Machine Learning*, pages 5610–5618, 2019.

[29] Tuhin Sarkar, Alexander Rakhlin, and Munther A Dahleh. Finite-time system identification for partially observed lti systems of unknown order. *arXiv preprint arXiv:1902.01848*, 2019.

[30] Max Simchowitz, Ross Boczar, and Benjamin Recht. Learning linear dynamical systems with semi-parametric least squares. *arXiv preprint arXiv:1902.00768*, 2019.

[31] Max Simchowitz, Horia Mania, Stephen Tu, Michael I Jordan, and Benjamin Recht. Learning without mixing: Towards a sharp analysis of linear system identification. *arXiv preprint arXiv:1802.08334*, 2018.

[32] Max Simchowitz, Karan Singh, and Elad Hazan. Improper learning for non-stochastic control, 2020.

[33] Robert F Stengel. *Optimal control and estimation*. Courier Corporation, 1994.

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

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