

Adaptive Online Non-stochastic Control

Naram Mhaisen N.MHAISEN@TUDELFT.NL and **George Iosifidis** G.IOSIFIDIS@TUDELFT.NL

Faculty of Electrical Engineering, Mathematics and Computer Science.

Delft University of Technology. The Netherlands.

Editors: A. Abate, K. Margellos, A. Papachristodoulou

Abstract

We tackle the problem of Non-stochastic Control (NSC) with the aim of obtaining algorithms whose policy regret is proportional to the difficulty of the controlled environment. Namely, we tailor the Follow The Regularized Leader (FTRL) framework to dynamical systems by using regularizers that are proportional to the actual witnessed costs. The main challenge arises from using the proposed adaptive regularizers in the presence of a state, or equivalently, a memory, which couples the effect of the online decisions and requires new tools for bounding the regret. Via new analysis techniques for NSC and FTRL integration, we obtain novel disturbance action controllers (DAC) with sub-linear data adaptive policy regret bounds that shrink when the trajectory of costs has small gradients, while staying sub-linear even in the worst case.

Keywords: Non-stochastic control; Follow the Regularized Leader; Online learning.

1. Introduction

This paper tackles the Online Non-stochastic Control problem: find a policy that endures minimum cost while controlling a dynamical system whose state changes via an unknown combination of learner’s actions and external parameters. Optimal Non-stochastic control has significant applications ranging from the control of medical equipment (Suo et al., 2021) to energy management in data centers (Lee et al., 2021). This work advances the results on this fundamental problem by proposing optimal control algorithms based on adaptive online learning.

1.1. Background & Motivation

We consider a typical NSC problem with a time-slotted dynamical system (Agarwal et al., 2019a). Namely, at each time step, the controller observes the system state $\mathbf{x}_t \in \mathbb{R}^{d_x}$ and decides an action $\mathbf{u}_t \in \mathbb{R}^{d_u}$ which induces cost $c_t(\mathbf{x}_t, \mathbf{u}_t)$. Then, the system transitions to state \mathbf{x}_{t+1} . Note that the new state and cost function, at each step, are revealed to the controller *after* it commits its action. Similar to (Agarwal et al., 2019a), we study Linear Time Invariant (LTI) systems where the transition is parametrized by matrices A, B and a *disturbance* vector \mathbf{w}_t :

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{w}_t. \quad (1)$$

We allow \mathbf{w}_t to be arbitrarily set by an *adversary* that aims to manipulate the state transition, and we only restrict it to be universally upper-bounded, i.e., $\|\mathbf{w}\| \leq w$. Similarly, the adversary is allowed to select at each step *any* Lipschitz continuous convex cost function $c_t : \mathbb{R}^{d_x} \times \mathbb{R}^{d_u} \mapsto \mathbb{R}$.

The controller’s task is to deduce a (possibly non-stationary) policy that maps states to actions, $\pi : \mathbf{x} \mapsto \mathbf{u}$, from a policy class Π , leading to a trajectory of low costs $\{c_t(\mathbf{x}_t, \mathbf{u}_t)\}_{t=1}^T$. The employed performance metric in this setting is the *policy regret*, cf. (Hazan et al., 2020), which

measures the accumulated extra cost endured by the learner’s policy compared to a stationary cost-minimizing policy designed with access to all future cost functions and disturbances:

$$\mathcal{R}_T \doteq \sum_{t=1}^T c_t(\mathbf{x}_t, \mathbf{u}_t) - \min_{\pi \in \Pi} \sum_{t=1}^T c_t(\mathbf{x}_t(\pi), \mathbf{u}_t(\pi))$$

where $\mathbf{x}_t(\pi), \mathbf{u}_t(\pi)$ are the *counterfactual* state-action sequences that would have been generated under the benchmark policy, whereas $(\mathbf{x}_t, \mathbf{u}_t)$ are the *actual* state-action pair that resulted from following possibly different series of policies. A sublinear regret $\mathcal{R}_T = o(T)$ guarantees that the cost endured by the learner will converge to that of the optimal policy, i.e., $\mathcal{R}_T/T \rightarrow 0$, and the recent Gradient Perturbation Controller (GPC), proposed in (Agarwal et al., 2019a), attains indeed $\mathcal{R}_T = \mathcal{O}(T^{1/2})$. GPC’s performance is established via a reduction to the Online Convex Optimization (OCO) with Memory framework (Anava et al., 2015), which in turn established sublinear regret via a reduction to the standard OCO framework (Shalev-Shwartz, 2012).

This paper also aims to reuse results from OCO in NSC, but targets stronger guarantees than the typical $\mathcal{O}(T^{1/2})$ regret. Namely, we aim to reduce the NSC problem to the *adaptive* OCO framework, cf. (McMahan, 2017), via the Follow-The-Regularized-Leader (FTRL) algorithm, which enables *adaptive* regret bounds. Adaptive bounds depend on the observed losses $\{c_t(\cdot, \cdot)\}_{t=1}^T$, and are of the form $\mathcal{O}((\sum_{t=1}^T g_t)^{1/2})$, where $g_t \doteq \|G_t\|^2$ and $G_t \doteq \nabla c_t(\mathbf{x}_t, \mathbf{u}_t)$.¹ Hence, \mathcal{R}_T remains sub-linear in T in the worst case (since $\|G_t\| \leq g, \forall t$ for Lipschitz functions), but is much tighter when small gradients are observed, i.e., when the cost functions are “easy”². Methods with such guarantees have received significant attention as they adapt to the environment (observed costs), and hence are less conservative than their non-adaptive counterparts (Duchi et al., 2011).

Alas, such stronger adaptive regret bounds are yet to be seen in the NSC framework, and this is not a mere artifact: adaptivity in stateful systems raises fresh challenges that require new technical solutions. Specifically, due to the system’s memory, the cost at each time step depends not only on the adversary but also on the past actions of the learner. This over-time coupling, which is of unknown intensity in the adaptive case, perplexes the optimization of the action at each step, and indeed the vast majority of OCO-based control approaches employ the non-adaptive variants of the learning algorithms (see discussion in Sec. 2). In light of the above, we ask the question: *can we design algorithms with policy regret bounds that adapt to easy costs, while maintaining sub-linear regret in all cases?* We answer in the affirmative and present a learning toolbox with policy regret bounds that are adaptive w.r.t. the adversity of the environment (disturbances and cost functions).

1.2. Methodology and Contributions

The analysis approach of NSC algorithms relies on approximating the cost of the non-stationary policy $\sum_{t=1}^T c_t(\mathbf{x}_t, \mathbf{u}_t)$ by the counterfactual stationary one $\sum_{t=1}^T c_t(\mathbf{x}_t(\pi_t), \mathbf{u}_t(\pi_t))$ with a sub-linear approximation term of order $\mathcal{O}(T^{1/2})$. This is possible due to the structure of LTI systems, where the effect of far-in-the-past actions decays exponentially, deeming the cost similar to that of “OCO with bounded memory” framework (Cesa-Bianchi et al., 2013). Then, Online Gradient Descend (OGD) is used to obtain an $\mathcal{O}(T^{1/2})$ bound for the stationary policy regret, leading to a total policy regret of the same order. In our case, however, we adopt the more general FTRL framework, cf. (McMahan, 2017), which makes it no longer apparent that such a reduction is possible.

1. The gradient is w.r.t. the policy’s parametrization. Its existence and boundness depend on Π as will be detailed later.

2. For a convex function $f(\cdot)$, costs with small gradient norms induce small regret: $f(\mathbf{x}) - f(\mathbf{y}) \leq \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{y} \rangle$.

In detail, FTRL operates on the principle that at each t , the learner optimizes its decision³ M_{t+1} by minimizing the aggregate cost until t , plus a strongly convex regularization term that penalizes the deviation from M_t . It is known that when these regularization terms are adaptive (i.e., the strong convexity is incremented at each t proportionally to the witnessed cost of M_t), we get bounds proportional to the observed costs. Now, the issue is that unlike GPC where, at each t , the distance between two previous decision variables M_s and M_{s+1} , $s \in [1, t-1]$ is (i) fixed and (ii) of order $\mathcal{O}(T^{-1/2})$, FTRL's adaptive regularizers make the distance between M_s and M_{s+1} (i) dependant on the costs witnessed until s (i.e., not fixed), and (ii) independent of the horizon T (or the time t). As these two properties were essential for proving GPC policy bounds, we are faced with a new challenge that is specific to FTRL and NSC integration. However, we show that the $c_t(\mathbf{x}_t, \mathbf{u}_t)$ cost can still approximate, up to a diminishing error, the counterfactual cost of $c_t(\mathbf{x}_t(\pi_t), \mathbf{u}_t(\pi_t))$. This approximation effectively enables regularization based on the witnessed cost at t and eventually leads to the sought-after adaptive regret bound of the form $\mathcal{O}((\sum_{t=1}^T g_t)^{1/2})$.

At a more conceptual level, it is interesting to observe that GPC and similar controllers already incorporated some notion of ‘‘adaptivity’’ in the sense that the performance guarantee is with respect to a *benchmark* that depends on the observed costs. This is clearly a more refined and less conservative approach than the classical H_∞ control framework that benchmarks itself against the worst-case scenario (Karapetyan et al., 2022). In this work, we make the next step, and not only use a cost-dependent benchmark like GPC, but also the performance of the controller w.r.t. that benchmark is itself shaped by the observed costs and perturbations. This is in contrast to GPC where the difference from the benchmark (i.e., the regret bound) does not depend on the encountered costs and perturbations. This additional layer of adaptability expedites the learning rates whenever the problem allows it (i.e., smaller or less volatile gradients are encountered). While the proposed adaptive controller is designed to benefit from easy environments, it does suffer performance degradation when the environment is indeed adversarial, but only in terms of constant factors. Hence, it provides a fail-safe tool for NSC.

1.3. Notation

We denote scalars by small letters and use $h_{a:b}$ for $\sum_{s=a}^b h_s$. Vectors are denoted by bold small letters and matrices by capital letters. Both can be indexed with time via a subscript. We denote by $\{\mathbf{a}_t\}_{t=1}^T$ the set $\{\mathbf{a}_1, \dots, \mathbf{a}_T\}$, and $[T]$ denotes $\{1, 2, \dots, T\}$. When T is not relevant, we use $\{\mathbf{a}_t\}_t$. Vector/matrices elements are indexed via a superscript in parenthesis, e.g., $\mathbf{a}_t^{(i)}$. $M = [M^{[i]} | M^{[j]}]$ denotes the augmentation of $M^{[i]}$ and $M^{[j]}$. $\|\cdot\|$ denotes the ℓ_2 norm for vectors and the Frobenius norm for matrices. $\|\cdot\|_*$ is the dual norm. $\|\cdot\|_{\text{op}}$ is the matrix spectral norm (the induced ℓ_2 norm).

2. Related Work

The NSC thread was initiated in (Agarwal et al., 2019a) which designed the first policy with sub-linear regret for dynamical systems, aspiring to generalize the classical control problem to general convex cost functions and arbitrary disturbances. In essence, the system state is modeled as a limited memory and the problem is cast into the standard OCO framework, which allows recovering the OCO bounds scaled by the memory length. Follow-up works refined these results for strongly convex functions (Simchowitz, 2020; Agarwal et al., 2019b; Foster and Simchowitz, 2020); and

3. As will be detailed later, the action \mathbf{u}_t is expressed as a linear combination of the parameters M .

systems where the actions are subject to fixed or adversarially-changing constraints (Li et al., 2021; Liu et al., 2023). NSC was also extended to systems where matrices (A, B) are unknown (Hazan et al., 2020), systems with bandit feedback (Gradu et al., 2020), and time-varying systems (Gradu et al., 2023). As expected, the regret bounds deteriorate in these cases, e.g., becoming $\mathcal{O}(T^{2/3})$ for unknown systems and $\mathcal{O}(T^{3/4})$ for systems with bandit feedback. All these works provide bounds that scale with the number of time steps T , as opposed to the fully adaptive bounds presented here, which are proportional to the witnessed costs and perturbations.

The NSC framework was also investigated using variations of regret metric that e.g., compare the learner’s decisions to the optimal one for each step t , or for an interval $\mathcal{I} \in [T]$. For instance, (Zhao et al., 2022) recovered the $\mathcal{O}((TP_T)^{1/2})$ dynamic regret OCO bounds, where P_T measures the times the optimal solution changes. Nonetheless, methods with static regret guarantees of the sort discussed in this paper, are building blocks for dynamic regret algorithms via, e.g., using them as “base controllers” within meta-controllers (Gradu et al., 2023; Simchowitz et al., 2020), and hence are directly relevant for these benchmarks, too. Competitive ratio is another metric that aims to minimize the ratio in accumulated cost between the benchmark policy and that of the learner. These algorithms have different semantics from the model considered here. For example, the adversary needs to reveal information about the cost function before the learner commits to a decision (Shi et al., 2020), or the cost is assumed to be a fixed quadratic function (Goel and Hassibi, 2022). Furthermore, it was shown in (Goel et al., 2023) that having a regret guarantee against the optimal DAC policy automatically guarantees a competitive ratio up to an additive sub-linear term. A recently refreshed line of work that also considers adversarial costs and transitions in stateful systems is Adversarial MDPs (Jin et al., 2023). Unlike NSC, this line considers finite states and actions, whose sizes appear in the bounds. Thus, their algorithms are irreducible to our settings.

Adaptivity has been an important concept in OCO, and we refer the readers to the survey in (McMahan, 2017), or the remarks in (Orabona, 2023, Sec. 3.5) for details. In adaptive learning the regret bounds scale proportionally to the witnessed costs. Hence, for *easy* environments with small costs, the bounds are considerably tighter than the standard worst-case ones which assume maximum cost at each round. On the other hand, if the environment follows actually a worst-case scenario, i.e., the adversary induces costs with large gradients that fluctuate aggressively, the adaptive bounds remain sublinear but have worse constant factors. Indeed, there is a price to be paid for adaptivity. For policy regret, the only form of adaptive bounds appears in (Zhang et al., 2022, Thm. 2), which, however, contain additive constants⁴ (i.e., do not directly collapse to 0 even when all costs are 0). Additionally, the authors do not consider the DAC policy class but the actions are rather a result of a “betting” technique that is suited to the “tracking” sub-task in control. In fact, any DAC policy can be combined with their meta-algorithm to obtain guarantees on the “tracking”, and hence our works are complementary (See discussion in (Zhang et al., 2022, Sec 4.2)).

3. Preliminaries

We introduce here the technical setup of NSC tailored to the assumptions and goals of this paper. **Policy class.** We consider the class Π of Disturbance Action Controllers (DAC) that was introduced in (Agarwal et al., 2019a). A policy $\pi \in \Pi$, with memory length p , is parametrized by p matrices $M \doteq [M^{[1]} | M^{[2]} | \dots | M^{[j]} | \dots | M^{[p]}]$, and a fixed stabilizing controller K . We also define the set $\mathcal{M} \doteq \{M : \|M\| \leq \kappa_M\}$, which uses the standard bounded variable assumption. The action at a

4. These works also target the “adaptive regret” metric.

step t according to a policy $\pi_t \in \Pi$, is then calculated via:

$$\mathbf{u}_t = K\mathbf{x}_t + \sum_{j=1}^p M_t^{[j]} \mathbf{w}_{t-j}. \quad (2)$$

Strong stability. This assumption is standard in OCO-based control as it enables non-asymptotic analysis (Cohen et al., 2018, Def. 3.1). It ensures the existence of a stabilizing controller K , such that $\|(A + BK)^t\|_{\text{op}} < \kappa(1 - \delta)^t$, $\delta \in (0, 1]$, $\kappa > 0$. Here, we assume that $\|A\|_{\text{op}} \leq 1 - \delta$, which allows us to satisfy the stability assumption with K being the zero matrix. This simplification facilitates the analysis, but the obtained results are still extensible for nonzero K since K is an external parameter to NSC algorithms, see, e.g., the discussion in (Zhang et al., 2022, Remark 4.1). We also assume $\|B\| \leq \kappa_B$, while the boundedness of $\|A\|$ follows from its spectral norm bound.

Cost functions. We consider the family of general convex functions for the losses, and we denote with $G_t(M)$ the gradient matrix⁵ of the cost $c_t(\mathbf{x}, \mathbf{u})$ w.r.t. M . If the argument M is not relevant (e.g., c_t is linear) or is fixed to M_t , then we denote the gradient simply with G_t . We also use the standard l -Lipschitz assumption $|c_t(\mathbf{x}, \mathbf{y}) - c_t(\mathbf{x}', \mathbf{y}')| \leq l\|(\mathbf{x}, \mathbf{u}) - (\mathbf{x}', \mathbf{u}')\|$, $\forall t \in [T]$.

DAC rationale. The DAC class strikes a balance between efficiency and performance. Specifically, both the states and actions are convex in the optimization variables M . This can be directly seen from (2) for the actions, whereas the state can be shown to be linear by unrolling the dynamics equation in (1) (e.g., see (Agarwal et al., 2019a, Lem. 4.3)) which we adapt below to our notation:

Lemma 1 *Assuming that $\mathbf{x}_1 = 0$, and parameters M_t, \mathbf{w}_t are 0 for $t \leq 0$, the state of the system reached at $t + 1$ upon the execution of actions $\{\mathbf{u}_i\}_{i=1}^t$, derived from a DAC policy π_t , is:*

$$\mathbf{x}_{t+1} = \sum_{i=0}^t A^i \left(B \sum_{j=1}^p (M_{t-i}^{[j]} \mathbf{w}_{t-i-j}) + \mathbf{w}_{t-i} \right).$$

Clearly, $c_t(\mathbf{x}_t, \mathbf{u}_t)$ is convex in M since \mathbf{x}_t and \mathbf{u}_t are linear in variables M . This facilitates the minimization of the cost function. Note that the boundness on the norm of \mathbf{x} and \mathbf{u} ,⁶ along with the Lipschitzness of $c(\cdot, \cdot)$ implies the existence of g s.t. $\|G_t\| \leq g, \forall t$.

Besides allowing convex costs, DAC policies are expressive as they approximate the class of linear policies up to an arbitrarily small constant error ζ . This follows from the next lemma from (Hazan and Singh, 2022, Lem. 6.9).

Lemma 2 *Let $\|A\|_{\text{op}} = 1 - \delta$, $\|\mathbf{w}\| \leq w$. Then, for any linear policy $\pi^{\mathbb{L}}$ with $\|K\| \leq \kappa^{\mathbb{L}}$, and an arbitrarily small constant ζ , there is a DAC policy with $p = \lceil 1/\delta \log(\kappa^{\mathbb{L}} w / \delta \zeta) \rceil$ that achieves a cost at most $\mathcal{O}(\zeta)$ far from the cost of the linear policy: $|c_t(\mathbf{x}_t(\pi), \mathbf{u}_t(\pi)) - c_t(\mathbf{x}_t(\pi^{\mathbb{L}}), \mathbf{u}_t(\pi^{\mathbb{L}}))| = \mathcal{O}(\zeta)$.*

Policy regret. We proceed to define formally the regret of a policy. We start with the benchmark DAC policy $\pi_{\star} \in \Pi$, that is fully characterized by matrix M_{\star} which can be calculated by solving:

$$\mathbf{P}_1 : \underset{M \in \mathcal{M}}{\text{minimize}} \sum_{t=1}^T c_t(\mathbf{x}_t(\pi_{\star}), \mathbf{u}_t(\pi_{\star})), \text{ subject to } \mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{w}_t, \quad \forall t \in [T], \quad (3)$$

$$\mathbf{u}_t = K\mathbf{x}_t + \sum_{j=1}^p M^{[j]} \mathbf{w}_{t-j}, \quad \forall t \in [T]. \quad (4)$$

5. Since our decision variable is in \mathcal{M} , the gradient “vector” can be organized into a matrix.

6. The bound on the state is due to the strong stability assumption and will be explicit in the analysis.

Constraints (3) enforce the LTI dynamics of the state transition, and (4) ensure the actions are taken from a DAC policy. Clearly, this hypothetical policy can only be calculated with access to future disturbances and costs, whereas the learner, at each step t , has access only to information until $t-1$.

There are some important notation remarks in order here. Recall that π_t is the policy at step t , where the learner decides \mathbf{u}_t using M_t . The state reached at t depends on the *sequence* of policies $\pi_1, \pi_2, \dots, \pi_{t-1}$, also referred to as non-stationary policy, and we denote it with $\mathbf{x}_t(\pi_{1,\dots,t-1})$. In contrast, the *counterfactual* state reached at step t by following a stationary policy π at all steps up to t is denoted $\mathbf{x}_t(\pi)$.⁷ Similarly, the action at t would in general differ for a stationary policy π versus a non-stationary policy, i.e., $\mathbf{u}_t(\pi_t) \neq \mathbf{u}_t(\pi_{1,\dots,t})$. However, when K is the zero matrix, these vectors are equivalent. To reduce clutter when possible, we omit the argument $\pi_{1,\dots,t-1}$ from $\mathbf{x}_t(\pi_{1,\dots,t-1})$. However, when we use the hypothetical state resulting from a fixed policy π , we make this explicit by writing $\mathbf{x}_t(\pi)$. The same applies to \mathbf{u}_t . Now, we define the policy regret as the cumulative difference between the cost induced by π_* , and the cost of the non-stationary policy as:

$$\mathcal{R}_T(\pi_{1,\dots,T}, \pi_*) \doteq \sum_{t=1}^T \left(c_t(\mathbf{x}_t(\pi_{1,\dots,t-1}), \mathbf{u}_t(\pi_{1,\dots,t})) - c_t(\mathbf{x}_t(\pi_*), \mathbf{u}_t(\pi_*)) \right).$$

Because of Lemma 2, $\pi_{1,\dots,T}$ has also a regret guarantee against the best policy in the linear class. That is, there is a constant $a > 0$, that depends only on l , $\kappa^{\mathbb{L}}$, and δ , such that $\mathcal{R}_T(\pi_{1,\dots,T}, \pi_*) = \mathcal{R}_T(\pi_{1,\dots,T}, \pi_*) + a\zeta T$. Hence, the sublinear regret rate can be preserved against any linear policy by tuning ζ , which can be achieved by increasing the DAC memory parameter p (see Lemma 2). Next, we design algorithms that minimize $\mathcal{R}_T(\pi_{1,\dots,T}, \pi_*)$.

4. AdaFTRL-C

We propose an algorithm (AdaFTRL-C) for optimizing the policy parameters M_t , $t \in [T]$. The algorithm uses the FTRL update formula:

$$M_{t+1} = \operatorname{argmin}_{M \in \mathcal{M}} \left\{ \sum_{s=1}^t c_s(\mathbf{x}_s(\pi), \mathbf{u}_s(\pi)) + r_s(M) \right\}. \quad (5)$$

For the incremental regularizers $r_t(\cdot)$, we define⁸ $g_t \doteq \max(\|G_t\|, \|G_t\|^2)$ and use:

$$r_t(M) = \frac{\sigma_t}{2} \|M - M_t\|^2, \text{ where } \sigma_1 = \sigma\sqrt{g_1}, \sigma_t = \sigma(\sqrt{g_{1:t}} - \sqrt{g_{1:t-1}}). \quad (6)$$

Note that the sum of the counterfactual cost sequence in (5) is indeed a function of M since both \mathbf{x} and \mathbf{u} are written in terms of M as shown earlier. Defining the norm $\|\cdot\|_t \doteq \sqrt{\sigma_{1:t}} \|\cdot\|$ we get that the regularizer $r_t(\cdot)$ is 1- strongly convex w.r.t. norm $\|\cdot\|_t$, whose dual is $\|\cdot\|_{t,*} = \frac{1}{\sqrt{\sigma_{1:t}}} \|\cdot\|$.

Due to the cost linearization principle (see e.g., (Shalev-Shwartz, 2012, Sec. 2.4)), (5) can be solved by two efficient steps: closed form solution of an unconstrained quadratic program, followed by a Euclidean projection, essentially matching the computational complexity of GPC.

Algorithm 1 summarizes the proposed routine; AdaFTRL-C first executes an action (line 3). Then, the cost function is revealed and G_t is recorded (line 4). The system then transitions to state

7. We call $\mathbf{x}_t(\pi)$ counterfactual since it is not necessarily equal to the true state of the system $\mathbf{x}_t(\pi_{1,\dots,t-1})$.

8. Unlike the discussion in the introduction, here we have g_t as the max of the norm and its square. The necessity of this originates from lower bounds on OCO with switching costs (Gofer, 2014) and will become clear in the analysis.

\mathbf{x}_{t+1} , effectively revealing the disturbance vector \mathbf{w}_t (lines 5, 6). The strong convexity parameter is calculated (line 7) and the next action is committed through updating the policy parameters (line 8). The policy regret of this AdaFTRL-C routine is characterized by the following theorem:

Algorithm 1 Adaptive FTRL Controller (AdaFTRL-C)

Input: An intrinsically stable LTI system (A, B) , $z \doteq pw\kappa_B$, $\sigma = \sqrt{\frac{\delta^2 + 2lz}{2\delta^2 k_M^2}}$, $M_1 \in \mathcal{M}$.

Output: $\{\mathbf{u}_t\}_{t=1}^T$

```

1: for  $t = 1, 2, \dots, T$  do
2:   Use action  $\mathbf{u}_t = \sum_{j=1}^p M_t^{[j]} \mathbf{w}_{t-j}$ 
3:   Observe cost  $c_t(\mathbf{x}_t, \mathbf{u}_t)$  and record the gradient  $G_t(M_t)$ 
4:   Observe new state  $\mathbf{x}_{t+1}$ 
5:   Record the disturbance  $\mathbf{w}_t = \mathbf{x}_{t+1} - A\mathbf{x}_t - B\mathbf{u}_t$ 
6:   Update regularization parameters  $\sigma_t$  via (6).
7:   Calculate  $M_{t+1}$  via (5).
8: end for
    
```

Theorem 3 Let (A, B) be an intrinsically stable system (i.e., $\|A\|_{op} \leq 1 - \delta$) with $\|B\| \leq \kappa_B$ and $\|\mathbf{w}\| \leq w$. Let the cost $c_t(\cdot, \cdot)$ be l -Lipschitz, and assume that the DAC parametrization $M = [M^{[1]}, \dots, M^{[p]}]$ is bounded: $\|M\| \leq \kappa_M$. Define $z \doteq pw\kappa_B$ and $g_t \doteq \max(\|G_t\|, \|G_t\|^2)$. Algorithm AdaFTRL-C produces policies $\pi_{1,\dots,T}$ such that for all T , the following bound holds:

$$\mathcal{R}_T(\pi_{1,\dots,T}, \pi_\star) \leq \frac{2}{\delta} \sqrt{2\kappa_M^2 (\delta^2 + 2lz)} \sqrt{\sum_{t=1}^T g_t}.$$

Discussion. AdaFTRL-C achieves a policy regret that scales according to the *witnessed costs*⁹ gradients. This can be much tighter than the bounds of non-adaptive controllers, like GPC, in easy environments. The improvement depends on how smaller the $\|G_t\|$ values are than g (an upper bound on $\|G_t\|$). In the worst case ($\|G_t\| = g, \forall t$), our bound is worse by a constant factor compared to GPC⁹, but maintains its sub-linearity. This is expected since AdaFTRL-C is conservative in regularization. Hence, when the losses are large, its performance is slightly degraded.

Proof layout. First, we bound the difference between the learner's accumulated cost, which is induced by $\pi_{1,\dots,t}$, and the counterfactual accumulated cost had the learner followed π_t in *all* steps $1, \dots, t$ (i.e., stationary policy π_t). Second, since the cost c_t is convex in the parametrization M_t of such stationary policy, we leverage the FTRL theory to bound the regret against the parameters M_\star .

Lemma 4 Let $c_t(\mathbf{x}_t, \mathbf{u}_t)$ be the cost induced by following $\pi_{1,\dots,t}$, and $c_t(\mathbf{x}_t(\pi_t), \mathbf{u}_t(\pi_t))$ the cost induced by following the policy $\pi_t, \forall \tau \leq t$, where each π_t is derived using (5). Let $z \doteq pw\kappa_B$. Then:

$$\nu_t \doteq \left| c_t(\mathbf{x}_t, \mathbf{u}_t) - c_t(\mathbf{x}_t(\pi_t), \mathbf{u}_t(\pi_t)) \right| \leq lz \sum_{i=0}^{t-1} (1 - \delta)^i \sum_{\tau=t-i-1}^{t-1} \frac{g_\tau}{\sigma \sqrt{g_{1:\tau}}} \doteq \hat{\nu}_t.$$

Proof Since c_t is l -Lipschitz, it suffices to bound the distance between its arguments; and because it holds¹⁰ $\mathbf{u}_t(\pi_{1,\dots,t}) = \mathbf{u}_t(\pi_t)$, we only need to bound $\|\mathbf{x}_t - \mathbf{x}_t(\pi_t)\|$. From Lemma 1:

$$\mathbf{x}_t = \sum_{i=0}^{t-1} A^i \left(B \sum_{j=1}^p M_{t-i-1}^{[j]} \mathbf{w}_{t-i-j-1} + \mathbf{w}_{t-i-1} \right) = \sum_{i=0}^{t-1} A^i (B M_{t-i-1} \mathbf{w}_{t-i-1,p} + \mathbf{w}_{t-i-1}), \quad (7)$$

9. Using OGD update rule instead would result in better *worst-case* bound by a factor of $\sqrt{2g}$.

10. Recall the stability with $K = 0$ assumption. Otherwise, action deviation is reducible to state deviation from (2).

where we defined $\mathbf{w}_{t-i,p} \doteq (\mathbf{w}_{t-i-1}, \dots, \mathbf{w}_{t-i-p})$ so as to express the vector $\sum_{j=1}^p M_t^{[j]} \mathbf{w}_{t-j}$ equivalently as $M_t \mathbf{w}_{t-1,p}$. Similarly, we can write:

$$\mathbf{x}_t(\pi_t) = \sum_{i=0}^{t-1} A^i (B M_t \mathbf{w}_{t-i-1,p} + \mathbf{w}_{t-i-1}). \quad (8)$$

Note that we have M_t in (8) instead of M_{t-i-1} in (7). This is because for the counterfactual state \mathbf{x}_t , $M_{t-i-1} = M_t, \forall i < t$. Now, subtracting the above two state expressions, we have: $\|\mathbf{x}_t - \mathbf{x}_t(\pi_t)\|$

$$\begin{aligned} &\stackrel{(a)}{\leq} \sum_{i=0}^{t-1} \|A^i B (M_{t-i-1} - M_t)\|_{\text{op}} \|\mathbf{w}_{t-i-1,p}\| \stackrel{(b)}{\leq} \sum_{i=0}^{t-1} \|A\|_{\text{op}}^i \|B\|_{\text{op}} \|M_{t-i-1} - M_t\|_{\text{op}} \|\mathbf{w}_{t-i-1,p}\| \\ &\stackrel{(c)}{\leq} \sum_{i=0}^{t-1} (1-\delta)^i \|B\| \|M_{t-i-1} - M_t\| \|\mathbf{w}_{t-i-1,p}\| \stackrel{(d)}{\leq} p w \kappa_B \sum_{i=0}^{t-1} (1-\delta)^i \|M_{t-i-1} - M_t\| \\ &\stackrel{(e)}{=} z \sum_{i=0}^{t-1} (1-\delta)^i \|M_{t-i-1} - M_t\| \stackrel{(f)}{\leq} z \sum_{i=0}^{t-1} (1-\delta)^i \sum_{\tau=t-i-1}^{t-1} \frac{\|G_\tau\|}{\sigma_{1:\tau}} \stackrel{(g)}{\leq} z \sum_{i=0}^{t-1} (1-\delta)^i \sum_{\tau=t-i-1}^{t-1} \frac{g_\tau}{\sigma \sqrt{g_{1:\tau}}}, \end{aligned}$$

where (a) follows by the triangular inequality and $\|A\mathbf{x}\| \leq \|A\|_{\text{op}} \|\mathbf{x}\|$, (b) by the sub-multiplicative property of matrix norms, (c) by the bound on the spectral norm of A and $\|\cdot\|_{\text{op}} \leq \|\cdot\|$, (d) by $\|B\| \leq \kappa_B$, $\|\mathbf{w}_{t,p}\| \leq \sum_{b=1}^p \|\mathbf{w}_{t-b}\| = pw$, (e) by grouping $z \doteq pw \kappa_B$, (f) by the auxiliary Lemma 5, and finally (g) by the definition of g_t . Using l -Lipschitzness completes the proof. \blacksquare

Now that we have characterized the stationary vs. non-stationary cost deviation, we proceed into (i) bound the cost of the stationary policy (ii) bound this deviation term.

Proof of Theorem 3. By the definition of ν_t (in Lemma 4), we can directly bound the policy regret:

$$\sum_{t=1}^T \left(c_t(\mathbf{x}_t, \mathbf{u}_t) - c_t(\mathbf{x}_t(\pi_\star), \mathbf{u}_t(\pi_\star)) \right) \leq \sum_{t=1}^T \left(c_t(\mathbf{x}_t(\pi_t), \mathbf{u}_t(\pi_t)) - c_t(\mathbf{x}_t(\pi_\star), \mathbf{u}_t(\pi_\star)) \right) + \nu_{1:T}$$

For the first sum, note that $c_t(\mathbf{x}_t(\pi_t), \mathbf{u}_t(\pi_t))$ is a function of only the iterates M_t and recall that M_t are computed according to (5). Hence, we recognize this sum as simply the standard FTRL regret, which can be bounded using the known bound (McMahan, 2017, Thm. 2). Thus,

$$\mathcal{R}_T(\pi_{1,\dots,T}, \pi_\star) \leq \sum_{t=1}^T r_t(M_\star) + \frac{1}{2} \sum_{t=1}^T \|\nabla c_t(\mathbf{x}_t(\pi_t), \mathbf{u}_t(\pi_t))\|_{t,\star}^2 + \nu_{1:T}.$$

We proceed to bounding each of the three summations above.

$$\begin{aligned} (i) : \sum_{t=1}^T r_t(M_\star) &= \sum_{t=1}^T \frac{\sigma_t}{2} \|M_\star - M_t\|^2 \leq 2\kappa_M^2 \sum_{t=1}^T \sigma_t \stackrel{(a)}{\leq} 2\sigma \kappa_M^2 \sqrt{g_{1:T}}, \\ (ii) : \frac{1}{2} \sum_{t=1}^T \|\nabla c_t(\mathbf{x}_t(\pi_t), \mathbf{u}_t(\pi_t))\|_{t,\star}^2 &\leq \frac{1}{2} \sum_{t=1}^T \frac{\|G_t\|^2}{\sigma_{1:t}} \leq \frac{1}{2} \sum_{t=1}^T \frac{g_t}{\sigma \sqrt{g_{1:t}}} \stackrel{(b)}{\leq} \frac{1}{\sigma} \sqrt{g_{1:T}} \end{aligned}$$

where inequality (a) is due to the telescoping sum of σ_t , and (b) is due to the common tool (Auer et al., 2002, Lemma 3.5), or its extended version (Orabona, 2023, Lem. 4.13)). The $\nu_{1:T}$ term is

especially challenging since the standard tools from OCO do not suffice. Namely, observe that we cannot use Auer’s lemma again for $\hat{\nu}_{1:T}$, since we are adding, at each t , t many terms, each divided by a different quantity. Hence, we prove a more general result in auxiliary Lemma 6, which, we believe, is of independent interest as it can be used to provide similar bounds in learning algorithms for systems with memory. With this new technical result at hand, we get:

$$(iii) : \nu_{1:T} \stackrel{(c)}{\leq} \hat{\nu}_{1:T} = lz \sum_{t=1}^T \sum_{i=0}^{t-1} (1-\delta)^i \sum_{\tau=t-i-1}^{t-1} \frac{g_\tau}{\sigma \sqrt{g_{1:\tau}}} \stackrel{(d)}{\leq} \frac{2lz}{\sigma \delta^2} \sqrt{g_{1:T}},$$

where (c) is from the definition of $\hat{\nu}_t$ and (d) is due to using Lemma 6 with $f(x) = 1/\sqrt{x}$, $a_t = g_t$, and $a_0 = 0$. Tuning σ we get $\sigma = \sqrt{\frac{\delta^2 + 2lz}{2\delta^2 k_M^2}}$. Substituting in (i), (ii), and (iii) we get the bound. ■

5. Numerical examples & Conclusion

We conclude with numerical examples of the implications of adaptivity in NSC. We consider an LTI system with $\mathbf{x}, \mathbf{u} \in \mathbb{R}^2$, $p=10$, and hence $M \in \mathbb{R}^{2 \times 20}$. The dynamics are $A = 0.9 \times \mathbf{I}_2$, $B = \mathbf{I}_2$, with perturbation of maximum magnitude of $w = \sqrt{2}$. We consider a linear cost $c_t = \langle \theta_t, \mathbf{x}_t \rangle$, with $\theta_t \in [-10, 10]^2$ and hence $l = \sqrt{200}$. Note that for $G_t^{(i,j)}$, the coefficient of $M_t^{(i,j)}$ when $c_t(x_t(\pi), u_t(\pi))$ is written explicitly in terms of M , we have that $G_t^{(i,j)} \leq \theta_t^{(i)} \times w_t^j / 0.1 = 100$ and hence $g = 633$. θ_t and w_t are set according to different scenarios in Fig. 1. In scenario (a), the encountered environment (θ_t and w_t) are smaller than the worst-case costs (by a factor of 10). AdaFTRL-C takes advantage of this and accelerates the optimization, leading to an improved average regret by roughly the same order. In fact, (a) represents a whole class of easy environments where the witnessed costs are smaller than their maximum values, and in all these cases, AdaFTRL-C outperforms GPC. In the worst-case scenario (b), the degradation of AdaFTRL-C reaches a maximum of only 3.9 times that of GPC. Lastly, (c) serves to demonstrate that even in highly volatile environments and worst-case costs/perturbation, AdaFTRL-C closely matches the performance of GPC. Concluding, we see that adaptivity offers a lot of potential gains for an entire family of easy environments, with tolerable degradation in the (single) worst-case scenario. The implementation of AdaFTRL-C and the code to reproduce the experiments are available [here](#).

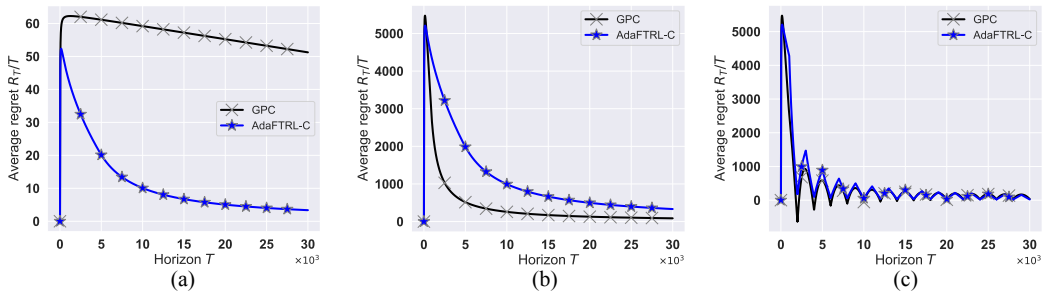


Figure 1: Average regret when (a): $\theta_t = (1, 1)$, $w_t = -(0.1, 0.1) \forall t$; (b): $\theta_t = (10, 10)$, $w_t = -(1, 1) \forall t$; (c): $\theta_t = (10, 10)$ or $-(10, 10)$ (Alternating every 1000 steps), $w_t = -(1, 1) \forall t$.

6. Auxiliary Lemmas

Lemma 5 *The distance between parameters M_{t-i-1} and M_t , with $i \leq t-1$, where each M_t is updated using (5), with regularizers (6), can be bounded as $\|M_{t-i-1} - M_t\| \leq \sum_{\tau=t-i-1}^{t-1} \frac{\|G_\tau\|}{\sigma_{1:\tau}}$.*

Proof First, note that $\|M_{t-i-1} - M_t\| \leq \sum_{\tau=t-i-1}^{t-1} \|M_\tau - M_{\tau+1}\|$. To bound each $\|M_\tau - M_{\tau+1}\|$, we leverage (McMahan, 2017, Lem. 7) which states that given a convex function $\phi_1(\cdot)$ with a minimizer $q_1 \doteq \operatorname{argmin}_q \phi_1(q)$ and a function $\phi_2(\cdot) = \phi_1(\cdot) + \psi(\cdot)$ that is a strongly convex w.r.t a norm $\|\cdot\|$ and has minimizer $q_2 \doteq \operatorname{argmin}_q \phi_2(q)$, then we can bound the two minimizers as $\|q_1 - q_2\| \leq \|b\|_*$ for some (sub)gradient $b \in \partial\psi(q_1)$. Now, we invoke this result by setting:

$$q_1 = M_\tau, \quad q_2 = M_{\tau+1}, \quad \phi_1(M) = \sum_{s=1}^{\tau-1} \left(c_s(\mathbf{x}_s(\pi), \mathbf{u}_s(\pi)) + r_s(M) \right) + r_\tau(M), \quad \text{and}$$

$$\phi_2(M) = \sum_{s=1}^{\tau-1} \left(c_s(\mathbf{x}_s(\pi), \mathbf{u}_s(\pi)) + r_s(M) \right) + r_\tau(M) + \underbrace{c_\tau(\mathbf{x}_\tau(\pi), \mathbf{u}_\tau(\pi))}_{\psi(M)}.$$

The function $\phi_2(M)$ is strongly-convex w.r.t. the norm $\|\cdot\|_\tau = \sqrt{\sigma_{1:\tau}} \|\cdot\|$, a property inherited due to containing the sum of all regularizers up to step τ , i.e., $\sum_{s=1}^\tau r_s(M)$. And the dual norm of the gradient of the above-defined $\psi(M)$ function, at each step s , is upper-bounded by $\|G_s\|_{\tau,*}$.

Finally, it suffices to observe that: (i) based on the definition of the update rule for variables M , and (ii) the fact that $r_t(M), \forall t$ is a proximal regularizer thus $r_t(M_t) = 0, \forall t$, we indeed have that $M_\tau = \operatorname{argmin}_{M \in \mathcal{M}} \phi_1(M)$ and $M_{\tau+1} = \operatorname{argmin}_{M \in \mathcal{M}} \phi_1(M) + \psi(M)$. Now, applying (McMahan, 2017, Lem. 7) we obtain $\|M_\tau - M_{\tau+1}\|_\tau \leq \|G_\tau\|_{\tau,*} \rightarrow \|M_\tau - M_{\tau+1}\| \leq \frac{\|G_\tau\|}{\sigma_{1:\tau}}$. ■

Lemma 6 Let $a_\tau \geq 0 \forall \tau$, $\delta \in (0, 1]$ and $f : [0, \infty] \mapsto [0, \infty)$ be a non-increasing function. Then:

$$\mathcal{S} \doteq \sum_{t=1}^T \sum_{i=0}^{t-1} (1-\delta)^i \sum_{\tau=t-i}^t a_\tau f(a_{0:\tau}) \leq \frac{1}{\delta^2} \int_{a_0}^{a_{1:T}} f(x) dx$$

Proof Instead of summing over all $i < t$ for each $t \in [1, T]$, the sum \mathcal{S} can be equivalently re-written by summing over all $t > i$ for each $i \in [0, T]$:

$$\mathcal{S} = \sum_{i=0}^T (1-\delta)^i \sum_{t=i+1}^T \sum_{\tau=t-i}^t a_\tau f(a_{0:\tau}) \quad (9)$$

Now we bound $\mathcal{S}' \doteq \sum_{t=i+1}^T \sum_{\tau=t-i}^t a_\tau f(a_{0:\tau})$. Let $s_t \doteq a_{0:t}$

$$\sum_{\tau=t-i}^t a_\tau f(a_{0:\tau}) \stackrel{(a)}{=} \sum_{\tau=0}^i a_{t-\tau} f(s_{t-\tau}) \stackrel{(b)}{=} \sum_{\tau=0}^i \int_{s_{t-\tau-1}}^{s_{t-\tau}} f(s_{t-\tau}) dx \stackrel{(c)}{\leq} \sum_{\tau=0}^i \int_{s_{t-\tau-1}}^{s_{t-\tau}} f(x) dx.$$

Hence, $\mathcal{S}' \leq \sum_{\tau=0}^i \sum_{t=i+1}^T \int_{s_{t-\tau-1}}^{s_{t-\tau}} f(x) dx \stackrel{(d)}{=} \sum_{\tau=0}^i \int_{s_{i-\tau}}^{s_{T-\tau}} f(x) dx \stackrel{(e)}{=} \sum_{\tau=0}^i \int_{s_0}^{s_T} f(x) dx = (i+1) \int_{s_0}^{s_T} f(x) dx.$

Where (a) follows by changing the sum index, (b) is from $\int_n^m c dx = (m-n)c, \forall n, m, c \in \mathbb{R}$, (c) from $f(\cdot)$ being non-increasing, (d) from the additive property of integrals, and (e) from expanding the integration limits and positivity of $f(\cdot)$. Substituting the bound for \mathcal{S}' back in (9) we get:

$$\mathcal{S} \leq \sum_{i=0}^T (1-\delta)^i (i+1) \int_{s_0}^{s_T} f(x) dx \leq \frac{1}{\delta^2} \int_{s_0}^{s_T} f(x) dx.$$

Where we used that $\sum_{i=0}^\infty i(1-\delta)^i \leq 1-\delta/\delta^2$ and $\sum_{i=0}^\infty (1-\delta)^i \leq 1/\delta$. ■

7. Acknowledgment

This research was supported by the National Growth Fund through the Dutch 6G flagship project “Future Network Services”.

References

- Naman Agarwal, Brian Bullins, Elad Hazan, Sham Kakade, and Karan Singh. Online control with adversarial disturbances. In *Proc. of ICML*, 2019a.
- Naman Agarwal, Elad Hazan, and Karan Singh. Logarithmic regret for online control. In *Proc. of NeurIPS*, 2019b.
- Oren Anava, Elad Hazan, and Shie Mannor. Online learning for adversaries with memory: Price of past mistakes. In *Proc. of NeurIPS*, 2015.
- Auer et al. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75, 2002. ISSN 0022-0000.
- Nicolo Cesa-Bianchi, Ofer Dekel, and Ohad Shamir. Online learning with switching costs and other adaptive adversaries. In *Proc. of NeurIPS*, 2013.
- Alon Cohen, Avinatan Hasidim, Tomer Koren, Nevena Lazic, Yishay Mansour, and Kunal Talwar. Online linear quadratic control. In *Proc. of ICML*, 2018.
- John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12(61):2121–2159, 2011.
- Dylan Foster and Max Simchowitz. Logarithmic regret for adversarial online control. In *Proc. of ICML*, 2020.
- Gautam Goel and Babak Hassibi. Competitive control. *IEEE Trans. Autom. Control*, 2022. doi: 10.1109/TAC.2022.3218769.
- Gautam Goel, Naman Agarwal, Karan Singh, and Elad Hazan. Best of both worlds in online control: Competitive ratio and policy regret. In *Proc. of LADC*, 2023.
- Eyal Gofer. Higher-order regret bounds with switching costs. In *Proc. of COLT*, 2014.
- Paula Gradu, John Hallman, and Elad Hazan. Non-stochastic control with bandit feedback. In *Proc. of NeurIPS*, 2020.
- Paula Gradu, Elad Hazan, and Edgar Minasyan. Adaptive regret for control of time-varying dynamics. In *Proc. of LADC*, 2023.
- Elad Hazan and Karan Singh. Introduction to online nonstochastic control. *arXiv:2211.09619*, 2022.
- Elad Hazan, Sham Kakade, and Karan Singh. The nonstochastic control problem. In *Proc. of ALT*, 2020.

- Tiancheng Jin, Junyan Liu, Chlo   Rouyer, William Chan, Chen-Yu We, and Haipeng Luo. No-regret online reinforcement learning with adversarial losses and transitions. *arXiv:2305.17380*, 2023.
- Aren Karapetyan, Andrea Iannelli, and John Lygeros. On the regret of h_∞ control. In *Proc. of IEEE CDC*, 2022.
- Russell Lee, Jessica Maghakian, Mohammad Hajiesmaili, Jian Li, Ramesh Sitaraman, and Zhenhua Liu. Online peak-aware energy scheduling with untrusted advice. *ACM SIGENERGY Energy Informatics Review*, 1(1):59–77, 2021.
- Yingying Li, Subhro Das, and Na Li. Online optimal control with affine constraints. In *Proc. of the AAAI*, 2021.
- Xin Liu, Zixian Yang, and Lei Ying. Online nonstochastic control with adversarial and static constraints. *arXiv:2302.02426*, 2023.
- H. Brendan McMahan. A Survey of Algorithms and Analysis for Adaptive Online Learning. *J. Mach. Learn. Res.*, 18(1):3117–3166, 2017.
- Francesco Orabona. A Modern Introduction to Online Learning. *arXiv:1912.13213*, 2023.
- Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012. ISSN 1935-8237. doi: 10.1561/22000000018.
- Guanya Shi, Yiheng Lin, Soon-Jo Chung, Yisong Yue, and Adam Wierman. Online optimization with memory and competitive control. In *Proc. of NeurIPS*, 2020.
- Max Simchowitz. Making non-stochastic control (almost) as easy as stochastic. In *Proc. of NeurIPS*, 2020.
- Max Simchowitz, Karan Singh, and Elad Hazan. Improper learning for non-stochastic control. In *Proc. of COLT*, 2020.
- Daniel Suo, Naman Agarwal, Wenhan Xia, Xinyi Chen, Udaya Ghai, Alexander Yu, Paula Gradu, Karan Singh, Cyril Zhang, Edgar Minasyan, et al. Machine learning for mechanical ventilation control. *arXiv:2102.06779*, 2021.
- Zhiyu Zhang, Ashok Cutkosky, and Ioannis Paschalidis. Adversarial tracking control via strongly adaptive online learning with memory. In *Proc. of AISTATS*, 2022.
- Peng Zhao, Yu-Xiang Wang, and Zhi-Hua Zhou. Non-stationary online learning with memory and non-stochastic control. In *Proc. of AISTATS*, 2022.