# Online Nonstochastic Control with Adversarial and Static Constraints

**Xin Liu** [1] **Zixian Yang** [2] **Lei Ying** [2]

## Abstract

This paper studies online nonstochastic control problems with adversarial and static constraints. We propose online nonstochastic control algorithms that achieve both sublinear regret and sublinear adversarial constraint violation while keeping static constraint violation minimal against the optimal constrained linear control policy in hindsight. To establish the results, we introduce an online convex optimization with memory framework under adversarial and static constraints, which serves as a subroutine for the constrained online nonstochastic control algorithms. This subroutine also achieves the state-of-the-art regret and constraint violation bounds for constrained online convex optimization problems, which is of independent interest. Our experiments demonstrate the proposed control algorithms are adaptive to adversarial constraints and achieve smaller cumulative costs and violations. Moreover, our algorithms are less conservative and achieve significantly smaller cumulative costs than the state-of-the-art algorithm.

## 1. Introduction

Online nonstochastic control paradigm has been widely applied in practice (Hazan & Singh, 2022; Suo et al., 2021; O'Connell et al., 2022). It is a topic of great interest in both the learning and control communities because online nonstochastic control algorithms are robust to time-varying and even adversarial environments (Agarwal et al., 2019a;b; Cohen et al., 2018; Dean et al., 2019). In an online nonstochastic control problem, the learner aims to learn a controller that minimizes the cumulative costs in a time-varying or even adversarial environment (e.g., the adversarial cost functions and disturbances). Practical control systems often operate under various constraints (e.g., safety, demand-supply, or energy constraints), which could be unpredictable and adversarial as well. For example, robots need to navigate along a collision-free path by maintaining a minimum distance from each other with complicated surroundings (e.g., pedestrians or other robots) (Brunke et al., 2022); cloud computing platforms should guarantee low-latency service for users with time-varying traffic workload (Tirmazi et al., 2020). Motivated by these applications, we focus on online nonstochastic control problems with adversarial constraints and propose online nonstochastic control algorithms to achieve the minimum cost and the best constraint satisfaction.

It is a challenging task to synthesize a safe (online) controller because of the conflicting objectives, i.e. minimizing the cumulative costs while satisfying all constraints. The traditional way to guarantee constraint satisfaction is to incorporate the constraints into Model Predictive Control (constrained MPC) (Mayne et al., 2005; Rawlings et al., 2017). However, constrained MPC often introduces overly-conservative, even infeasible, actions in the presence of system disturbances. To address the issue, a sequence of works have relaxed or softened the constraints in MPC (Zeilinger et al., 2010; Wabersich et al., 2022; Raković et al., 2023). For online nonstochatic control problems where cost functions or disturbance could be arbitrary and time-varying, it is impossible to predict the model so the constrained MPC method is not applicable. Only a few works (Nonhoff & Müller, 2021; Li et al., 2021) have considered online nonstochastic control with constraints, and they only studied "static" affine constraints on the state $x_t$ and input $u_t$, i.e., $D_x x_t \leq 0$ and $D_u u_t \leq 0$. The work by (Nonhoff & Müller, 2021) studied an online nonstochatic control problem with state and input constraints, but the system dynamics are noise/disturbance-free. The work most related to ours is (Li et al., 2021), which considers adversarial cost functions and adversarial disturbance but static affine constraints. The paper proposed a gradient descent-based control algorithm (called online gradient descent with buffer zones (OGD-BZ)) to achieve $\tilde{O}(\sqrt{T})$ regret while the static affine constraints are satisfied. However, the work assumed the affine constraints and the knowledge of slackness of the constraints so that robust optimization methods can be used to construct safe/feasible regions for the affine constraints. The method also has the issue of being over-conservative as

[1]The School of Information Science and Technology, ShanghaiTech University, Shanghai, China. [2]The Electrical Engineering and Computer Science Department, The University of Michigan, Ann Arbor, Ann Arbor, USA. Correspondence to: Xin Liu <liuxin7@shanghaiteh.edu.cn>.

*Proceedings of the $40^{th}$ International Conference on Machine Learning*, Honolulu, Hawaii, USA. PMLR 202, 2023. 

constrained MPC. Moreover, the method in (Li et al., 2021) cannot be applied to the adversarial constraints because they are unknown to the controller before an action is taken.

In this paper, we study an online nonstochastic control problem with adversarial constraints, where the cost and constraint functions are revealed after the control/action has been taken (our setting can also include static constraints). Specifically, we consider a discrete-time, linear system as follows

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

where $x_t$ is the state, $u_t$ is the control/action and $w_t$ is the adversarial noise/disturbance at time $t$. The learner takes an action $u_t$ and observe the cost function $c_t(x_t, u_t)$ and constraint functions $d_t(x_t, u_t)$ and $l(x_t, u_t)$. The goal of the learner is to minimize the cumulative costs $\sum_{t=1}^{T} c_t(x_t, u_t)$ while satisfying the constraints, where the constraint violation will be measured by three different metrics: the soft cumulative constraint violation $\sum_{t=1}^{T} d_t(x_t, u_t)$, the hard cumulative violation $\sum_{t=1}^{T} d_t^+(x_t, u_t)$, and static anytime violation $l(x_t, u_t), \forall t$, where $d_t^+(\cdot) = \max(d_t(\cdot), 0)$. We propose a class of constrained online nonstochastic control algorithms (COCA) that guarantee sublinear regret and sublinear constraint violation against the *optimal linear* controllers for the constrained problems in hindsight, where the controller knows everything apriori. Our contributions are summarized below (in Table 1):

- We propose COCA-Soft when adversarial constraints are measured using soft cumulative violation. The algorithm is based on the Lyapunov optimization method. COCA-Soft achieves $\tilde{O}(\sqrt{T})$ regret, $O(1)$ cumulative soft violation for adversarial constraints, and $o(1)$ anytime violation for static constraints.
- When considering hard cumulative violation as the metric, we propose COCA-Hard based on proximal optimization methods. COCA-Hard achieves $\tilde{O}(T^{2/3})$ regret, $\tilde{O}(T^{2/3})$ hard cumulative violation for adversarial constraints, and $\tilde{O}(T^{-1/3})$ anytime violation for static constraints.
- When the cost functions are strongly-convex, we propose COCA-Best2Worlds that integrates proximal and Lyapunov optimization methods and provides performance guarantees in terms of both soft and hard violation metrics. COCA-Best2Worlds achieves $\tilde{O}(\sqrt{T})$ regret, $O(1)$ and $\tilde{O}(\sqrt{T})$ cumulative soft and hard violation for adversarial constraints, respectively, and $o(1)$ anytime violation for static constraints.

To the best of our knowledge, all these results are new in the setting of online non-stochastic control with *adversarial* and *general static* constraints (not necessarily static affine constraints). In this paper, we focus on presenting COCA-Soft and COCA-Hard and defer COCA-Best2Worlds in (Liu et al., 2023).

### 1.1. Related work

**Online Nonstochastic Control of Dynamic System:** Online nonstochastic control leverages online learning or data-driven methods to design efficient and robust control algorithms in an adversarial environment, where both cost functions and disturbances could be adversarial (Agarwal et al., 2019a). The main idea behind online nonstochastic control is to design a disturbance-action controller by carefully synthesizing the historical disturbance through the subroutine of online convex optimization (OCO) (Hazan, 2016). The initial work (Agarwal et al., 2019a) shows the disturbance-action controller can achieve $\tilde{O}(\sqrt{T})$ regret w.r.t. the optimal linear controller in hindsight that knows all costs and disturbances beforehand. The results have been refined in (Agarwal et al., 2019b; Foster & Simchowitz, 2020) when the cost functions are strongly-convex and have been generalized to various settings block-box time-invariant system (Plevrakis & Hazan, 2020; Simchowitz et al., 2020; Cassel et al., 2022), the time-varying system in (Minasyan et al., 2021), the non-linear system (Foster et al., 2020; Ghai et al., 2022), and only with bandit feedback (Gradu et al., 2020). Further, a neural network has been used to parameterize the control policy in (Chen et al., 2022b) and the regret performance is analyzed by combining OCO algorithm (Hazan, 2016) and neural-tangle kernel (NTK) theory (Jacot et al., 2018). However, these works do not consider any adversarial or static constraints.

**Online Learning with Constraints:** Online learning with constraints has been widely studied in the literature (Mahdavi et al., 2012; Sun et al., 2017; Neely & Yu, 2017; Cao et al., 2021; Yi et al., 2021a;b; Guo et al., 2022). Existing results can be classified according to the types of constraints, e.g., static, stochastic, and adversarial constraints. We next only review the papers on adversarial constraints because they are the most related ones. The work (Sun et al., 2017) studied OCO with adversarial constraints and established $O(\sqrt{T})$ regret and $O(T^{3/4})$ soft cumulative constraint violation. The work (Yi et al., 2021b; Guo et al., 2022) considered the benchmark of hard cumulative violation and established $O(\sqrt{T})$ regret and $O(T^{3/4})$ hard violation. The performance has been further improved to be $O(\log T)$ regret and $O(\sqrt{T \log T})$ hard violation when the objective is strongly-convex (Guo et al., 2022).

**Safe Reinforcement Learning:** Safe reinforcement learning (RL) refers to reinforcement learning with safety constraints and has received great interest as well (Aswani et al., 2013; Fisac et al., 2019; García et al., 2015; Koller et al., 2018; Wabersich & Zeilinger, 2018; Cheng et al., 2019; Tir-

| Algorithms | Cost Function | Regret | Soft/Hard Adversarial Vio. | Static Vio. |
|---|---|---|---|---|
| OGD-BZ in (Li et al., 2021) | Convex | $\tilde{O}(\sqrt{T})$ | None/None | None* |
| COCA-Soft | Convex | $\tilde{O}(\sqrt{T})$ | $O(1)$/None | $o(1)$ |
| COCA-Hard | Convex | $O(T^{2/3})$ | $O(T^{2/3})$/$O(T^{2/3})$ | $O(T^{-1/3})$ |
| COCA-Best2Worlds | Strongly Convex | $\tilde{O}(\sqrt{T})$ | $O(1)$/$\tilde{O}(\sqrt{T})$ | $o(1)$ |

*Table 1.* Our contribution and related work. *(Li et al., 2021) establishes zero violation for static anytime affine constraints, which is a special case of static constraints studied in this paper. Moreover, if we use projection-based method for static affine constraints, our algorithm can also achieve zero violation.

mazi et al., 2020; Efroni et al., 2020; Ding et al., 2020; 2021; Liu et al., 2021; Amani et al., 2021; Vaswani et al., 2022; Chen et al., 2022a; Ghosh et al., 2022; Wei et al., 2022). In safe RL, The agent optimizes the policy by interacting with the environment without violating safety constraints. However, the line of safe RL requires either the knowledge of the initial safe policy or a stationary environment where the reward and cost distributions are time-invariant.

## 2. Online Nonstochastic Control with Constraints

In this section, we introduce the online nonstochastic control problem with constraints and the performance metrics for evaluating the cost and constraint satisfaction. We consider the following linear system:

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

where $x_t \in \mathbb{R}^n$ is the state, $u_t \in \mathbb{R}^m$ is the control/action and $w_t \in \mathbb{R}^n$ is the noise or disturbance at time $t$. Note $w_t$ could be even adversarial. The system parameters $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are assumed to be known. A constrained online nonstochastic control system works as follows: given the state $x_t, \forall t \in [T]$, the learner takes action $u_t$ and observes the cost function $c_t(x_t, u_t)$ and constraint functions $d_t(x_t, u_t)$ and $l(x_t, u_t)$. The system evolves to the next state $x_{t+1}$ according to the system equation. Note the noises, costs, and constraints are chosen by an oblivious adversary. Our objective is to design an optimal control policy to minimize $\sum_{t=1}^{T} c_t(x_t, u_t)$ while satisfying the constraints. Next, we introduce our baseline policy and define the performance metrics of regret and constraint satisfaction.

**Offline Control Problem:** Assuming the full knowledge of disturbance, cost functions, and constraint functions beforehand, the offline control problem is defined to be:

$$\begin{aligned}\min_{\{u_t\}} \ & \sum_{t=1}^{T} c_t(x_t, u_t) \\ \text{s.t. } & x_{t+1} = Ax_t + Bu_t + w_t, \forall t \in [T], \\ & d_t(x_t, u_t) \le 0, \\ & l(x_t, u_t) \le 0, \forall t \in [T].\end{aligned}$$

We define $K^* \in \mathbb{R}^{m\times n}$ to be the optimal linear control $u_t^{K^*} = -K^* x_t^{K^*}$ which satisfies the constraints, i.e., $K^* \in \Omega$ such that

$$\Omega = \{\pi \mid d_t(x_t^\pi, u_t^\pi) \le 0, l(x_t^\pi, u_t^\pi) \le 0, \forall t \in [T]\}.$$

**Regret:** Given the optimal linear policy $K^*$ as the baseline, the goal of the learner is to design an online nonstochastic control policy $\pi$ that minimizes the following regret

$$\mathcal{R}(T) = \sum_{t=1}^{T} c_t(x_t^\pi, u_t^\pi) - \sum_{t=1}^{T} c_t(x_t^{K^*}, u_t^{K^*}).$$

**Constraint Violation:** The control algorithm needs to obey the constraints. However, since the constraints are unknown and adversarial, some violation has to occur during learning and control. To evaluate the level of constraint satisfaction, we consider two different metrics for adversarial constraints: soft violation and hard violation:

$$\mathcal{V}_d^{soft}(T) = \sum_{t=1}^{T} d_t(x_t^\pi, u_t^\pi),$$

$$\mathcal{V}_d^{hard}(T) = \sum_{t=1}^{T} d_t^+(x_t^\pi, u_t^\pi),$$

and the anytime violation for the static constraint $l$ is

$$\mathcal{V}_l(t) = l(x_t^\pi, u_t^\pi).$$

Note soft and hard violation metrics for adversarial constraints are for different applications. For example, in cloud computing, the latency constraint is soft and the soft violation is a natural metric; however in drone control, the power constraint is hard and the hard violation is a better metric. We consider anytime violation for static constraint function $l$ because it is related to state and input constraints that needs to be satisfied anytime, resembling stability requirements. To present our algorithm, we first present several key concepts of online nonstochastic control from (Agarwal et al., 2019a).

### 2.1. Preliminary on Constrained Online Nonstochastic Control

**Definition 2.1** (Strong Stability)**.** A linear controller $K \in \mathbb{R}^{m\times n}$ is $(\kappa, \rho)$-strongly stable if there exists matrices

$A, B, U, L$ such that

$$\tilde{A} := A - BK = ULU^{-1}$$

with $\max(\|U\|_2, \|H^{-1}\|_2, \|K\|_2) \leq \kappa$ and $\|L\|_2 \leq 1 - \rho, \rho \in (0, 1]$.

Given the knowledge of system dynamics $A$ and $B$, the $(\kappa, \rho)$-strongly stable controller $K$ can be computed with semi-definite programming (SDP) (Cohen et al., 2018). Note the stable controller $K$ might not satisfy the constraints in $\Omega$, i.e., $K \notin \Omega$.

**Definition 2.2** (Disturbance-Action Policy Class (DAC))**.** A disturbance-action policy $\pi(K, \{\mathbf{M}_t\})$ with memory size $H$ is defined as follows

$$u_t = -Kx_t + \sum_{i=1}^{H} \mathbf{M}_t^{[i]} w_{t-i},$$

where $\mathbf{M}_t \in \mathcal{M}$ and $\mathcal{M} = \{\mathbf{M}_t^{[1]}, \cdots, \mathbf{M}_t^{[H]} \mid \|\mathbf{M}_t^{[i]}\| \leq a(1-\rho)^i, \mathbf{M}_t^{[i]} \in \mathbb{R}^{m \times n}, a > 0, \forall i \in [H]\}$.

The disturbance-action policy consists of a linear combination of the disturbance (the memory size is $H = \Theta(\log T)$ in the paper). Given a stable controller $K$ and by carefully choosing $\{\mathbf{M}_t\}$, $\pi(K, \{\mathbf{M}_t\})$ aims to approximate a good linear stable controller that achieves small costs and satisfies the constraints in $\Omega$. Further, we define DAC with fixed weights, which serves as an intermediate policy class and is frequently used in our analysis.

**Definition 2.3** (DAC with Fixed Weight)**.** For a DAC $\pi(K, \{\mathbf{M}_t\})$, the set of fixed weight DAC policies is

$$\mathcal{E} = \{\pi(K, \{\mathbf{M}_t\}) \mid \mathbf{M}_t = \mathbf{M}, \forall t \in [T]\}. \tag{1}$$

Let $\mathbf{M}_{s:t} := \{\mathbf{M}_s, \cdots, \mathbf{M}_t\}$ and $\tilde{A} = A - BK$. Under a policy $\pi(K, \{\mathbf{M}_t\})$ in DAC, $\Psi_{t,i}^{\pi}(\mathbf{M}_{t-H:t-1})$ is defined to be the disturbance-state transfer matrix:

$$\begin{aligned}&\Psi_{t,i}^{\pi}(\mathbf{M}_{t-H:t-1})\\ =&\tilde{A}^{i-1}\mathbb{I}(i \leq H) + \sum_{j=1}^{H} \tilde{A}^{j-1} B \mathbf{M}_{t-j}^{[i-j]} \mathbb{I}_{i-j \in [1,H]}.\end{aligned}$$

We occasionally abbreviate $\Psi_{t,i}^{\pi}(\mathbf{M}_{t-H:t-1})$ to be $\Psi_{t,i}^{\pi}$ without causing any confusion. As shown in (Agarwal et al., 2019a), under a policy $\pi$ in DAC, the state is represented by $x_t^{\pi} = \sum_{i=1}^{t} \Psi_{t,i}^{\pi} w_{t-i}$, which is equivalent to

$$x_t^{\pi} = \tilde{A}^H x_{t-H} + \sum_{i=1}^{2H} \Psi_{t,i}^{\pi} w_{t-i}.$$

By truncating the true states, we define the approximated states and actions

$$\tilde{x}_t^{\pi} = \sum_{i=1}^{2H} \Psi_{t,i}^{\pi} w_{t-i}, \;\; \tilde{u}_t^{\pi} = -K\tilde{x}_t^{\pi} + \sum_{i=1}^{H} \mathbf{M}_t^{[i]} w_{t-i}. \tag{2}$$

Further, we have the approximated cost and constraint functions in the following

$$c_t(\tilde{x}_t^{\pi}, \tilde{u}_t^{\pi}), \;\; d_t(\tilde{x}_t^{\pi}, \tilde{u}_t^{\pi}), \;\; l(\tilde{x}_t^{\pi}, \tilde{u}_t^{\pi}). \tag{3}$$

Based on the definition of approximated constraint functions, we define an approximated constraint set $\tilde{\Omega}$ such that

$$\tilde{\Omega} = \{\pi \mid d_t(\tilde{x}_t^{\pi}, \tilde{u}_t^{\pi}) \leq 0, l(\tilde{x}_t^{\pi}, \tilde{u}_t^{\pi}) \leq 0, \forall t \in [T]\}.$$

Note the states in both $\Omega$ and $\tilde{\Omega}$ are driven by the same underlying dynamics with the policy $\pi$. Intuitively, $\Omega$ and $\tilde{\Omega}$ are "close" if the approximated errors of states and actions are small. We introduce the following assumptions on cost and constraint functions.

**Assumption 2.4.** The cost $c_t(x, u)$ and constraint functions $d_t(x, u)$ and $l(x, u)$ are convex and differentiable. Let $C_0$ and $C_1$ be positive constants. As long as $\|x - x'\| \leq D$ and $\|u - u'\| \leq D$, we assume the gradients $\|\nabla_x c_t\|, \|\nabla_u c_t\|, \|\nabla_x d_t\|, \|\nabla_u d_t\|, \|\nabla_x l\|, \|\nabla_u l\|$ are bounded by $C_0 D$; we assume $c_t, d_t$, and $l$ are bounded by $C_1 D$. We assume the noises $\|w_t\| \leq W, \forall t$ are bounded.

Further, we introduce an assumption on the feasibility of the offline control problem. This assumption can be regarded as Slater's condition in the optimization literature.

**Assumption 2.5.** Let $\delta$ be a positive constant, there exists a policy $\pi \in \mathcal{E}$ such that

$$d_t(x_t^{\pi}, u_t^{\pi}) \leq -\delta, \;\; l(x_t^{\pi}, u_t^{\pi}) \leq -\delta, \; \forall t \in [T].$$

Note it is non-trivial to extend (Agarwal et al., 2019a) into a constrained setting because it requires an adaptive balance between the costs and constraints. Next, we propose our constrained control algorithms to achieve this goal.

## 3. Constrained Online Nonstochastic Control Algorithm

Given an (arbitrary) stable control policy $K$, we develop a set of online nonstochastic control policy $\pi(K, \{\mathbf{M}_t\})$ to adjust the weights of disturbance/noise $\{\mathbf{M}_t\}$ such that it achieves small regret and constraint violation. Specifically, we use constrained online learning algorithms as the subroutines of our online nonstochastic control policy $\pi(K, \{\mathbf{M}_t\})$ to optimize the weights $\{\mathbf{M}_t\}$.

According to the definition in (2), the approximated state $\tilde{x}_t^{\pi}$ and action $\tilde{u}_t^{\pi}$ are only related to the weights of the past $H$ steps $\mathbf{M}_{t-H:t} := \{\mathbf{M}_{t-H}, \cdots, \mathbf{M}_t\}$. Therefore, we denote $\tilde{c}_t(\mathbf{M}_{t-H:t}) := c_t(\tilde{x}_t^{\pi}, \tilde{u}_t^{\pi})$, $\tilde{d}_t(\mathbf{M}_{t-H:t}) := d_t(\tilde{x}_t^{\pi}, \tilde{u}_t^{\pi})$, $\tilde{l}(\mathbf{M}_{t-H:t}) := l(\tilde{x}_t^{\pi}, \tilde{u}_t^{\pi})$. To simplify notation, we further define $\tilde{c}_t(\mathbf{M}) := \tilde{c}_t(\mathbf{M}, \cdots, \mathbf{M})$ and similarly

for $\tilde{d}_t(\mathbf{M})$ and $\tilde{l}(\mathbf{M})$. We are ready to present our constrained online nonstochastic control algorithm (COCA) by using the constrained online convex optimization solver (COCO-Solver) as the subroutine.

**Constrained Online Nonstochastic Control Algorithm**

**Initialize:** a $(\kappa, \rho)$ stable controller $K$ and the proper learning rates in COCO-Solver.
**for** $t = 1, \cdots, T$, **do**
  **Observe** state $x_t$ and compute the disturbance $w_{t-1}$.
  **Apply** control $u_t = -Kx_t + \sum_{i=1}^{H} \mathbf{M}_t^{[i]} w_{t-i}$.
  **Receive** feedback including cost function $c_t(x_t, u_t)$ and constraint functions $d_t(x_t, u_t)$ and $l(x_t, u_t)$.
  **Compute** the approximated cost function $\tilde{c}_t(\cdot)$ and constraint functions $\tilde{d}_t(\cdot)$ and $\tilde{l}(\cdot)$.
  **Invoke** the **COCO-Solver**$(\mathbf{M}_t, Q_t, \tilde{c}_t(\cdot), \tilde{d}_t(\cdot), \tilde{l}(\cdot))$ to obtain $\mathbf{M}_{t+1}$ and $Q_{t+1}$.
**end for**

For COCA at time $t$, we observe the state $x_t$ and infer the previous disturbance $w_{t-1} = x_t - Ax_{t-1} - Bu_{t-1}$. The information of state and the past disturbances are used in $\pi(K, \{\mathbf{M}_t\})$ to output a control/action $u_t$. Then we observe the full information of the cost function $c_t(\cdot, \cdot)$ and constraint functions $d_t(\cdot, \cdot)$ and $l_t(\cdot, \cdot)$. Based on the feedback, we compute $\tilde{c}_t(\cdot)$, $\tilde{d}_t(\cdot)$, and $\tilde{l}(\cdot)$, and invoke COCO-Solver to optimize the weights of disturbance $\{\mathbf{M}_t\}$ for the next control period $t + 1$. Note COCO-Solver has an input variable of $Q_t$, which is designed to track the soft cumulative violation of $d_t(\cdot, \cdot)$ and is also a feedback signal to control the trade-off between the cost and soft constraint satisfaction of $d_t(\cdot, \cdot)$.

As discussed, COCO-Solver is the key to optimizing the cumulative costs while minimizing (soft or hard) constraint violations. Depending on the types of constraint violation metrics we want to optimize, COCO-Solver will be instantiated with the COCO-Soft or COCO-Hard solvers. Moreover, when the cost functions are strongly-convex, we design COCO-Best2Worlds solver that can optimize soft and hard cumulative violations simultaneously. The key in COCO-Solver is to design the proper surrogate functions to incorporate the cost and (adversarial) constraints (e.g., the Lyapunov optimization method for soft violation and the penalty-based method for hard violation). This design enables a flexible trade-off between the cost and constraint violation and renders a large policy region for optimizing costs. Therefore, COCA with dedicated solvers is less conservative compared to the existing robust optimization-based control approaches (Dean et al., 2019; Li et al., 2021), which usually construct overly-conservative regions via robust optimization and use a projection-based method to guarantee the constraints in each control period. Moreover, these approaches are infeasible to handle the adversarial constraints because they require the knowledge of future constraints. Next, we introduce these solvers and their corresponding theoretical performance, respectively.

### 3.1. COCA with COCO-Soft Solver

We instantiate COCO-Solver in COCA with the algorithm **COCO-Soft**$(\mathbf{M}_t, Q_t, \tilde{c}_t(\cdot), \tilde{d}_t(\cdot), \tilde{l}(\cdot))$. The main idea behind COCO-soft is to carefully design a control surrogate function based on the Lyapunov optimization method such that the cumulative cost and soft violation are balanced. Specifically, for the loss function, we use $\tilde{c}_t(\mathbf{M}_t) + \langle \mathbf{M} - \mathbf{M}_t, \nabla \tilde{c}_t(\mathbf{M}_t) \rangle$ to approximate $\tilde{c}_{t+1}(\mathbf{M})$. For the adversarial constraint, we use $\tilde{d}_t(\mathbf{M}_t) + \langle \mathbf{M} - \mathbf{M}_t, \nabla \tilde{d}_t(\mathbf{M}_t) \rangle$ to approximate $\tilde{d}_{t+1}(\mathbf{M})$ and the virtual queue $Q_t$ indicates the degree of the constraint violation, i.e., a large/small $Q_t$ means a large/small violation of the adversarial constraints. The product term $Q_t \langle \mathbf{M} - \mathbf{M}_t, \nabla \tilde{d}_t(\mathbf{M}_t) \rangle$ in the control surrogate function is a proxy term of $Q_t \tilde{d}_{t+1}(\mathbf{M})$. Combining with the virtual queue update, minimizing the product term is equivalent to minimize the Lyapunov drift of $Q_{t+1}^2(\mathbf{M}) - Q_t^2$. For the static constraint function $\tilde{l}(\mathbf{M})$, we directly impose the penalty factor $\eta$ to prevent the violation. In summary, the control surrogate function carefully integrates the approximated cost and constraints in $V\tilde{c}_{t+1}(\mathbf{M}) + Q_{t+1}^2(\mathbf{M}) + \eta \tilde{l}^+(\mathbf{M})$, so optimizing the surrogate function guarantees the best trade-off between the cumulative costs and constraint violations. Next, we present

**COCO-Soft**$(\mathbf{M}_t, Q_t, \tilde{c}_t(\cdot), \tilde{d}_t(\cdot), \tilde{l}(\cdot))$

**Control Decision:** Choose $\mathbf{M}_{t+1} \in \mathcal{M}$ to minimize the control surrogate function

$$V\langle \mathbf{M} - \mathbf{M}_t, \nabla \tilde{c}_t(\mathbf{M}_t) \rangle + Q_t \langle \mathbf{M} - \mathbf{M}_t, \nabla \tilde{d}_t(\mathbf{M}_t) \rangle + \eta \tilde{l}^+(\mathbf{M}) + \alpha \|\mathbf{M} - \mathbf{M}_t\|^2.$$

**Virtual Queue Update:**

$$Q_{t+1} = \left[ Q_t + \tilde{d}_t(\mathbf{M}_t) + \langle \mathbf{M}_{t+1} - \mathbf{M}_t, \nabla \tilde{d}_t(\mathbf{M}_t) \rangle + \epsilon \right]^+$$

**Output:** $\mathbf{M}_{t+1}$ and $Q_{t+1}$.

the theoretical results for COCA with the COCO-Soft solver. We only present order-wise results. The exact constants and proof can be found in Appendix C in (Liu et al., 2023).

**Theorem 3.1.** *Given a stable controller $K$, under Assumptions 2.4 and 2.5, COCA with COCO-Soft solver achieves*

$$\mathcal{R}(T) = O\left(\sqrt{T} \log^3 T\right),$$
$$\mathcal{V}_d^{soft}(T) = O(1), \mathcal{V}_l(t) = O(1/\log T), \forall t \in [T].$$

*Remark* 3.2. Theorem 3.1 implies COCA with COCO-Soft achieves similar performance as the optimal offline linear controller $K^*$ when $T$ is large. COCO-Soft only needs to solve an almost unconstrained optimization problem, which is more computationally efficient than the projection-based method in (Li et al., 2021). Moreover, if we project $\mathbf{M}_{t+1}$ into the set of $\{\mathbf{M} \mid \tilde{l}(\mathbf{M}) \leq 0\}$ instead of the penalty-based design $\eta\tilde{l}^+(\mathbf{M})$, we can also achieve zero anytime violation as in (Li et al., 2021), which is verified in Appendix C.4 in (Liu et al., 2023). Finally, we would like to mention that Lyapunov optimization with the pessimistic design in virtual queue allows COCO-Soft to achieve the best trade-off between regret and violation (please refer to Theorem 4.4 and Remark 4.5).

### 3.2. COCA with COCO-Hard Solver

We instantiate COCO-Solver in COCA with the algorithm **COCO-Hard**$(\mathbf{M}_t, Q_t, \tilde{c}_t(\cdot), \tilde{d}_t(\cdot), \tilde{l}(\cdot))$. The main idea behind COCO-hard is to capture the constraint directly in the control surrogate function with the proximal penalty-based method, which is different from the Lyapunov optimization method in COCO-Soft. Since the design for $\tilde{c}_t(\mathbf{M})$ and $\tilde{l}(\mathbf{M})$ is similar to that in COCO-Soft. We focus on the new design for taking care of the adversarial constraint. Specifically, we directly use $\tilde{d}_t^+(\mathbf{M})$ as a proxy term of $\tilde{d}_{t+1}^+(\mathbf{M})$ and impose a penalty factor $\gamma$ to prevent the violation. Therefore, the control surrogate function approximates $V\tilde{c}_{t+1}(\mathbf{M})+\gamma\tilde{d}_{t+1}^+(\mathbf{M})+\eta\tilde{l}^+(\mathbf{M})$, which directly captures the cumulative costs and (hard) constraint violation.

**COCO-Hard**$(\mathbf{M}_t, \tilde{c}_t(\cdot), \tilde{d}_t(\cdot), \tilde{l}(\cdot))$

**Control Decision:** Choose $\mathbf{M}_{t+1} \in \mathcal{M}$ to minimize the control surrogate function

$$V\langle \mathbf{M} - \mathbf{M}_t, \nabla\tilde{c}_t(\mathbf{M}_t)\rangle + \gamma\tilde{d}_t^+(\mathbf{M}) + \eta\tilde{l}^+(\mathbf{M}) + \alpha\|\mathbf{M} - \mathbf{M}_t\|^2$$

**Output:** $\mathbf{M}_{t+1}$.

Next, we present the theoretical results for COCA with the COCO-Hard solver. The detailed parameters and proof can be found in Appendix D in (Liu et al., 2023).

**Theorem 3.3.** *Given a stable linear controller $K$, under Assumptions 2.4, COCA with COCO-Hard solver achieves*

$$\begin{aligned}\mathcal{R}(T) =& O(T^{\frac{2}{3}}\log^2 T),\\ \mathcal{V}_l(t) =& O(\log T/T^{\frac{1}{3}}), \forall t \in [T],\\ \mathcal{V}_d^{hard}(T) =& O(T^{\frac{2}{3}}\log^2 T), \text{ for a large } T.\end{aligned}$$

*Remark* 3.4. COCO-Hard establishes Theorem 3.3 without a "Slater-like" Assumption 2.5. Similar as in COCO-Soft, COCO-Hard is computationally efficient and avoids the complex projection operator. Moreover, by tuning learning rates $V, \gamma, \eta$, and $\alpha$ in COCO-Hard, we are able to establish a trade-off $\mathcal{R}(T) = \tilde{O}(T^{\max\{1-\frac{c}{2},c\}})$, $\mathcal{V}_d^{hard}(T) = \tilde{O}(T^{\max\{1-\frac{c}{2},0.5\}})$, and $\mathcal{V}_l(t) = \tilde{O}(T^{-\frac{c}{2}}), \forall t \in [T]$, with $c \in [0.5, 1)$ (please refer to Appendix D.4 in (Liu et al., 2023) for details).

### 3.3. A Roadmap to Prove Theorems 3.1 and 3.3

We provide a general roadmap to prove the regret and constraint violation in Theorems 3.1 and 3.3.

**Regret analysis:** we have the following decomposition for the regret

$$\begin{aligned}\mathcal{R}(T) =& \sum_{t=1}^T c_t(x_t^\pi, u_t^\pi) - \sum_{t=1}^T c_t(x_t^{K^*}, u_t^{K^*})\\ =& \sum_{t=1}^T [c_t(x_t^\pi, u_t^\pi) - c_t(\tilde{x}_t^\pi, \tilde{u}_t^\pi)] \quad (4)\\ &+ \sum_{t=1}^T c_t(\tilde{x}_t^\pi, \tilde{u}_t^\pi) - \min_{\pi\in\tilde{\Omega}\bigcap\mathcal{E}} \sum_{t=1}^T c_t(\tilde{x}_t^\pi, \tilde{u}_t^\pi) \quad (5)\\ &+ \min_{\pi\in\tilde{\Omega}\bigcap\mathcal{E}} \sum_{t=1}^T c_t(\tilde{x}_t^\pi, \tilde{u}_t^\pi) - \sum_{t=1}^T c_t(x_t^{K^*}, u_t^{K^*}). \quad (6)\end{aligned}$$

The term in (4) is on the approximation error of cost functions, related to the approximated errors of states and actions, and is bounded by $O(1)$ in Lemma F.1 when choosing the memory size $H = \Theta(\log T)$ for a disturbance-action policy. The term in (6) is on the representation ability of a disturbance-action policy with constraints, which can also be bounded by $O(1)$ in Lemma G.1 because $K^*$ intuitively belongs to the class $\tilde{\Omega}\bigcap\mathcal{E}$. The term in (5) is the key to the regret of COCA, which depends on the regret of COCO-Solver and will be established in Theorems 4.4 and 4.6 in the next section, respectively.

**Cumulative soft/hard violation of $d_t$ function:** we have the following decomposition for the soft/hard violation of adversarial $d_t$:

$$\begin{aligned}\mathcal{V}_d^{soft}(T) =& \sum_{t=1}^T [d_t(x_t^\pi, u_t^\pi) - d_t(\tilde{x}_t^\pi, \tilde{u}_t^\pi)] + d_t(\tilde{x}_t^\pi, \tilde{u}_t^\pi)\\ \mathcal{V}_d^{hard}(T) \leq& \sum_{t=1}^T [d_t(x_t^\pi, u_t^\pi) - d_t(\tilde{x}_t^\pi, \tilde{u}_t^\pi)]^+ + d_t^+(\tilde{x}_t^\pi, \tilde{u}_t^\pi)\end{aligned}$$

The difference terms in $\mathcal{V}_d^{soft}(T)$ and $\mathcal{V}_d^{hard}(T)$ are on the approximation error of constraint functions, which are also related to the approximated errors of states and actions and are bounded by $O(1)$ in Lemma F.1; the terms $\sum_{t=1}^T d_t(\tilde{x}_t^\pi, \tilde{u}_t^\pi)$ or $\sum_{t=1}^T d_t^+(\tilde{x}_t^\pi, \tilde{u}_t^\pi)$ are on the soft or hard constraint violation of COCO-Solver, which are established in the next section.

**Anytime violation of $l$ function:** we have the following decomposition for the constraint $l$ function:

$$\mathcal{V}_l(t) = l(x_t^\pi, u_t^\pi) - l(\tilde{x}_t^\pi, \tilde{u}_t^\pi) + l(\tilde{x}_t^\pi, \tilde{u}_t^\pi).$$

Similarly, the difference term is on the anytime approximated error of constraint functions, which is bounded by $O(1/T)$ in Lemma F.1; the term of $l(\tilde{x}_t^\pi, \tilde{u}_t^\pi)$ depends on the anytime violation of COCO-Solver, which is established in the next section.

As discussed above, the key is to analyze the performance of $c_t(\tilde{x}_t^\pi, \tilde{u}_t^\pi), d_t(\tilde{x}_t^\pi, \tilde{u}_t^\pi)$ and $l(\tilde{x}_t^\pi, \tilde{u}_t^\pi)$ with COCO-Solver, where the approximated states $\tilde{x}_t^\pi$ and actions $\tilde{u}_t^\pi$ depend on the past states and actions up to the previous $H$ steps, i.e., $\mathbf{M}_{t-H:t}$. COCO-Solver is naturally implemented by a constrained online convex optimization with memory framework (COCOwM). Since COCOwM is a plug-in component of COCA, we present the analysis in a separate section and any advance in COCOwM can be directly translated to that in COCA.

# 4. COCO-Solver via Constrained Online Convex Optimization with Memory

In the standard constrained online convex optimization (COCO), the loss and constraint functions at time $t$ only depend on the current decision $\mathbf{M}_t \in \mathcal{M}$. In the constrained online convex optimization with memory (COCOwM), the loss function $f_t(\mathbf{M}_{t-H:t})$ and cost functions $g_t(\mathbf{M}_{t-H:t})$ and $h(\mathbf{M}_{t-H:t})$ at time $t$ depends on the historical decisions of $\{\mathbf{M}_{t-H:t}\}$ up to the previous $H$-steps. If we associate $f_t(\mathbf{M}_{t-H:t})$, $g_t(\mathbf{M}_{t-H:t})$, and $h(\mathbf{M}_{t-H:t})$ with $c_t(\tilde{x}_t^\pi, \tilde{u}_t^\pi), d_t(\tilde{x}_t^\pi, \tilde{u}_t^\pi)$, and $l(\tilde{x}_t^\pi, \tilde{u}_t^\pi)$, respectively, COCOwM is naturally used to optimize $\{\mathbf{M}_t\}$ and the performance of COCOwM (or COCO-Solver) can be translated to that of COCA. Similar to COCA, we define the metrics of regret and constraint violation for COCOwM.

**Offline COCOwM:** Recall for a simple notation, we define $f_t(\mathbf{M}) = f_t(\mathbf{M}, \cdots, \mathbf{M})$ and similarly for $g_t(\mathbf{M})$ and $h(\mathbf{M})$. We formulate the offline COCOwM as follows:

$$\min_{\mathbf{M}\in\mathcal{M}} \sum_{t=1}^{T} f_t(\mathbf{M}) \tag{7}$$

$$\text{s.t. } h(\mathbf{M}) \le 0,\ g_t(\mathbf{M}) \le 0, \forall t \in [T]. \tag{8}$$

Let the optimal solution to (7)-(8) be $\mathbf{M}^*$. We define the regret and constraint violations of COCOwM

$$\mathcal{R}_f(T) = \sum_{t=1}^{T} f_t(\mathbf{M}_{t-H:t}) - \sum_{t=1}^{T} f_t(\mathbf{M}^*),$$

$$\mathcal{V}_g^{soft}(T) = \sum_{t=1}^{T} g_t(\mathbf{M}_{t-H:t}),\ \mathcal{V}_g^{hard}(T) = \sum_{t=1}^{T} g_t^+(\mathbf{M}_{t-H:t})$$

$$\mathcal{V}_h(t) = h(\mathbf{M}_{t-H:t}),\ \forall t \in [T].$$

Before presenting the formal analysis of COCOwM (or COCO-Solver) algorithms, we introduce several necessary assumptions.

**Assumption 4.1.** The feasible set $\mathcal{M}$ is convex with diameter $D$ such that $\|\mathbf{M} - \mathbf{M}'\| \le D, \forall \mathbf{M}, \mathbf{M}' \in \mathcal{M}$.

**Assumption 4.2.** The loss and constraint functions are convex and Lipschitz continuous with Lipschitz constant $L$. Further, assume $h(\mathbf{M}) \le E$ and $g_t(\mathbf{M}) \le E, \forall t \in [T]$.

**Assumption 4.3.** There exists a positive constant $\xi > 0$ and $\mathbf{M} \in \mathcal{M}$ such that $h(\mathbf{M}) \le -\xi$ and $g_t(\mathbf{M}) \le -\xi, \forall t \in [T]$.

We are ready to present the theoretical results of COCO-Soft and COCO-Hard solvers introduced in Section 3.

## 4.1. Theoretical Analysis of COCO-Soft

**COCO-Soft**$(\mathbf{M}_t, Q_t, f_t(\cdot), g_t(\cdot), h(\cdot))$ optimizes $f_t(\mathbf{M}_t)$, $g_t(\mathbf{M}_t)$, and $h(\mathbf{M}_t)$, which are slightly off to the true targets $f_t(\mathbf{M}_{t-H,t})$, $g_t(\mathbf{M}_{t-H,t})$, and $h(\mathbf{M}_{t-H,t})$. Therefore, we also need to quantify the mismatches by establishing the stability terms $\|\mathbf{M}_t - \mathbf{M}_{t+1}\|$. The following theorem establishes the regret and constraint violation of COCO-Soft.

**Theorem 4.4.** *Under Assumptions 4.1-4.3, COCO-Soft algorithm achieves*

$$\begin{aligned}\mathcal{R}_f(T) =& O\left(\sqrt{T}\log^3 T\right),\\ \mathcal{V}_h(t) =& O(1/\log T), \forall t \in [T],\\ \mathcal{V}_g^{soft}(T) =& O\left(1\right),\ \textit{for a large } T.\end{aligned}$$

We outline the key steps of the proof and leave the details in Appendix C in (Liu et al., 2023). We first study the regret

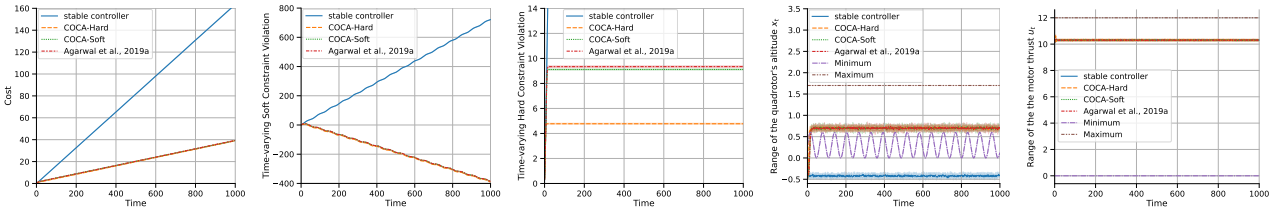
$$\begin{aligned}\mathcal{R}_f(T) \le& \sum_{t=1}^{T} |f_t(\mathbf{M}_{t-H:t}) - f_t(\mathbf{M}_t)| + f_t(\mathbf{M}_t) - f_t(\mathbf{M}^*)\\ \le& O(H^2 \sum_{t=1}^{T} \|\mathbf{M}_{t+1} - \mathbf{M}_t\|) + O(\sqrt{T}\log^3 T + T\epsilon),\end{aligned}$$

which proves the regret bound by letting the pessimistic factor $\epsilon = \Theta(\log^3 T/\sqrt{T})$ in Lemma C.1 and by Lemma C.2. Similarly, we establish the soft constraint violation

$$\begin{aligned}\mathcal{V}_g^{soft}(T) \le& \sum_{t=1}^{T} |g_t(\mathbf{M}_{t-H:t}) - g_t(\mathbf{M}_t)| + g_t(\mathbf{M}_t)\\ =& O(H^2 \sum_{t=1}^{T} \|\mathbf{M}_{t+1} - \mathbf{M}_t\| + \sqrt{T}\log^3 T - T\epsilon)\end{aligned}$$

which is $O(1)$ with $\epsilon = \Theta(\log^3 T/\sqrt{T})$ for a relatively large $T$. Finally, we have anytime violation

$$\begin{aligned}\mathcal{V}_h(t) \le& |h(\mathbf{M}_{t-H:t}) - h(\mathbf{M}_t)| + h(\mathbf{M}_t)\\ =& O(1/\log T).\end{aligned}$$




(a) Cumulative costs (b) Soft violations (c) Hard violations (d) Range of $x_t$ (e) Range of $u_t$

*Figure 1.* Experiment results for QVF control with winds blowing down $w_t \sim U(-5.5, -4.5)$. The lines are plotted by averaging over 10 independent runs. The shaded areas in Figures (a)-(c) are 95% confidence intervals and in Figure (d)-(e) are the full ranges of the data.

*Remark* 4.5. Theorem 4.4 achieves $\tilde{O}(\sqrt{T})$ regret and $O(1)$ violation, which significantly improves the best existing results of $O(\sqrt{T})$ regret and violation in (Neely & Yu, 2017), a special case of ours by letting $H = 0$. Moreover, for OCO with stochastic constraints in (Yu et al., 2017), COCO-Soft can also establish $\tilde{O}(\sqrt{T})$ regret and $O(1)$ violation against the same baseline in (Yu et al., 2017), which again improves $O(\sqrt{T})$ regret and violation in (Yu et al., 2017) (please refer to Appendix C.5 for details). The key design and analysis for such improvement is to introduce the pessimistic factor $\epsilon$ in virtual queue update and compare COCO-Soft with a "$\epsilon$-tight" baseline such that we can trade regret (the amount of $T\epsilon$) to achieve a constant soft violation $\mathcal{V}_g^{soft}(T)$.

### 4.2. Theoretical Analysis of COCO-Hard

For **COCO-Hard**, we also omit the intermediate lemmas and only present the main theorem due to the limited space. The details and proofs can be found in Appendix D in (Liu et al., 2023).

**Theorem 4.6.** *Under Assumptions 4.1-4.2, COCO-Hard algorithm achieves*

$$\begin{aligned}\mathcal{R}_f(T) =& O\left(T^{\frac{2}{3}}\log^2 T\right),\\ \mathcal{V}_h(t) =& O(\log T/T^{\frac{1}{3}}), \forall t \in [T],\\ \mathcal{V}_g^{hard}(T) =& O\left(T^{\frac{2}{3}}\log^2 T\right), \text{ for a large } T.\end{aligned}$$

Note once the regret and constraint violation of COCO solvers have been established in Theorems 4.4 and 4.6, we plug them into the roadmap in Section 3.3 and prove the performance of COCA in Theorems 3.1 and 3.3.

## 5. Experiment

In this section, we test our algorithms on a quadrotor vertical flight (QVF) control under an adversarial environment, which is modified from (Li et al., 2023). The experiment is designed to verify if our algorithms are adaptive to time-varying/adversarial constraints and we compare COCA with the stable controller without considering constraints in (Agarwal et al., 2019a). We also test our algorithms on a Heating Ventilation and Air Conditioning (HVAC) control with static affine constraints in (Li et al., 2021). The experiment is to verify if our approach is less conservative in designing COCA algorithms. The learning rates of COCA algorithms can be found in Appendix H in (Liu et al., 2023).

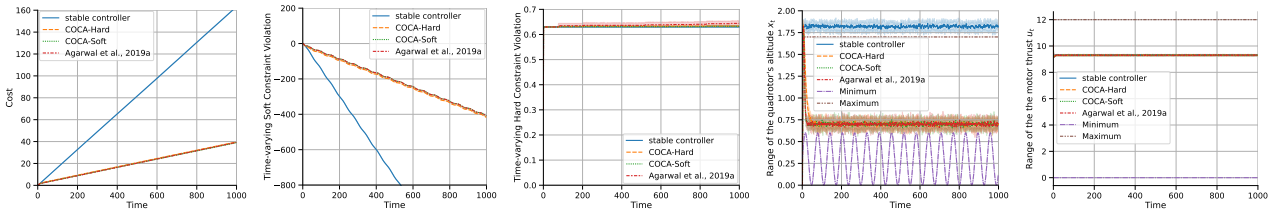
### 5.1. QVF Control

The system equation is

$$\ddot{x}_t = \frac{u_t}{m} - g - \frac{I^a \dot{x}_t}{m} + w_t,$$

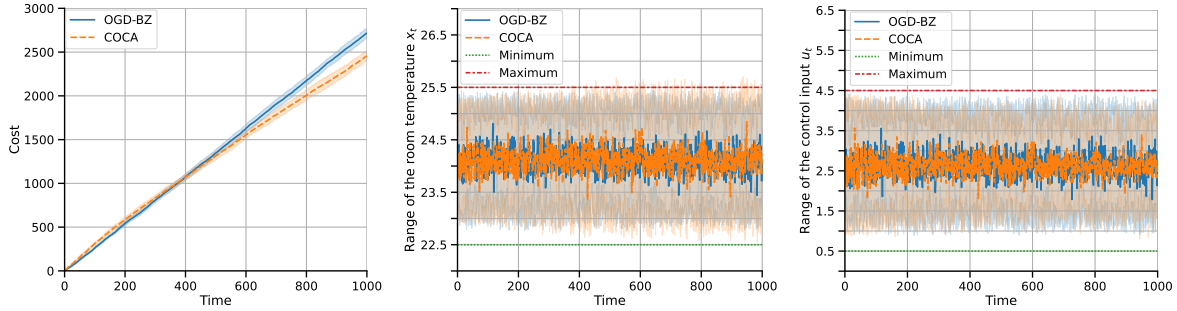
where $x_t$ is the altitude of the quadrotor, $u_t$ is the motor thrust, $m$ is the mass of the quadrotor, $g$ is the gravitational acceleration, and $I^a$ is the drag coefficient of the air resistance. Let $m = 1\text{kg}$, $g = 9.8\text{m/s}^2$, and $I^a = 0.25\text{kg/s}$. The system is discretized with $\Delta_t = 1s$. We impose time-varying constraints, $z_t \geq 0.3 + 0.3\sin(t/10)$, to emulate the complicated time-varying obstacles on the ground. The static affine constraints are $z_t \leq 1.7$ and $0 \leq v_t \leq 12$. We consider a time-varying quadratic cost function $0.1(z_t - 0.7)^2 + 0.1\dot{z}_t^2 + \chi_t(v_t - 9.8)^2$, where $\chi_t \sim U(0.1, 0.2)$. We simulate two different wind conditions $w_t \sim U(-5.5, -4.5)$ (winds blow down) and $w_t \sim U(4.5, 5.5)$ (winds blow up), respectively.

Figure 1 shows the experiment results for QVT control with winds blowing down $w_t \sim U(-5.5, -4.5)$. Figure 2 shows the experiment results for QVT control with winds blowing up $w_t \sim U(4.5, 5.5)$. These two figures show that both COCA-Soft and COCA-Hard achieve much better performance than the stable controller in (Agarwal et al., 2019a). Specifically, our algorithms have much smaller cumulative costs, near-zero static constraint violations, negative cumulative soft violations that decrease by time, and cumulative hard violations that remain unchanged small constant shortly after the initial stages. Moreover, our algorithms have small hard constraint violations while having similar costs compared with the controller in (Agarwal et al., 2019a) in both settings. These results verify our algorithms are very adaptive to the adversarial environment and achieve minimal




(a) Cumulative costs (b) Soft violations (c) Hard violations (d) Range of $x_t$ (e) Range of $u_t$

*Figure 2.* Experiment results for QVF control with winds blowing up $w_t \sim U(4.5, 5.5)$. The lines are plotted by averaging over 10 independent runs. The shaded areas in Figures (a)-(c) are 95% confidence intervals and in Figure (d)-(e) are the full ranges of the data.




(a) Cumulative costs (b) Range of $x_t$ (c) Range of $u_t$

*Figure 3.* Experiment results for HVAC Control. The lines are average of 10 runs. The lines are plotted by averaging over 10 independent runs. The shaded areas in Figure (a) are 95% confidence intervals and in Figures (b)-(c) are the full ranges of the data.

cumulative costs and best constraints satisfaction. Moreover, we observe COCA-Soft and COCA-Hard have almost identical performance on cumulative costs and soft violations, but COCA-Hard has a smaller hard violation than COCA-Soft. It justifies the penalty-based design is efficient in tackling the hard violation.

### 5.2. HVAC Control

The system equation is

$$\dot{x}_t = \frac{\theta^o - x_t}{v\zeta} - \frac{u_t}{v} + \frac{w_t + \iota}{v},$$

where $x_t$ is the room temperature, $u_t$ is the airflow rate of the HVAC system as the control input, $\theta^o$ is the outdoor temperature, $w_t$ is the random disturbance, $\iota$ represents the impact of the external heat sources, $v$ and $\zeta$ denotes the environmental parameters. Let $v = 100$, $\zeta = 6$, $\theta^o = 30°C$, and $\iota = 1.5$. Let $w_t \sim U(-1.1, 1.3)$ and we discretize the system with $\Delta_t = 60s$. Similar to (Li et al., 2021), the state and input constraints are $22.5 \leq x_t \leq 25.5$ and $0.5 \leq u_t \leq 4.5$, respectively. We specify the time-varying cost functions $c_t = 2(x_t - 24)^2 + \chi_t(u_t - 2.5)^2$ with $\chi_t \sim U(0.1, 4.0)$.

We compare COCA with OGD-BZ algorithm (COCA-Soft and COCA-Hard are exactly identical, called COCA, because there only exist static state and input constraints). Figure 3 (a)-(c) show the cumulative costs, the ranges of the room temperature $x_t$ and control input $u_t$. We observe that COCA has a significantly better cumulative cost than OGD-BZ algorithm with a near-zero constraint violation. The results verify our approach is effective in designing less-conservative COCA algorithms.

## 6. Conclusions and Extensions

In this paper, we studied online nonstochastic control problems with adversarial and static constraints. We developed COCA algorithms that minimize cumulative costs and soft or hard constraint violations. Our experiments showed the proposed algorithms are adaptive to time-varying environments and less conservative in achieving better performance. We need to mention when the costs are strongly-convex, COCA-Best2Worlds combines the Lyapunov optimization and the penalty-based methods to achieve minimal soft/hard violations for adversarial constraints. The informal results are in Table 1 and the details are in Appendix E in (Liu et al., 2023). We conclude our paper by mentioning possible future directions. The paper assumes the system dynamics is linear, known, and time-invarying. It would be interesting to extend COCA to the setting when the systems are unknown (Plevrakis & Hazan, 2020; Simchowitz et al., 2020; Cassel et al., 2022), time-varying (Minasyan et al., 2021), or even nonlinear (Foster et al., 2020; Ghai et al., 2022).

## References


Agarwal, N., Bullins, B., Hazan, E., Kakade, S., and Singh, K. Online control with adversarial disturbances. *Proceedings of the 36th International Conference on Machine Learning*, 2019a.

Agarwal, N., Hazan, E., and Singh, K. Logarithmic regret for online control. *Advances Neural Information Processing Systems (NeurIPS)*, 2019b.

Amani, S., Thrampoulidis, C., and Yang, L. Safe reinforcement learning with linear function approximation. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Aswani, A., Gonzalez, H., Sastry, S. S., and Tomlin, C. Provably safe and robust learning-based model predictive control. *Automatica*, 2013.

Brunke, L., Greeff, M., Hall, A. W., Yuan, Z., Zhou, S., Panerati, J., and Schoellig, A. P. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 2022.

Cao, X., Zhang, J., and Poor, H. V. Online stochastic optimization with time-varying distributions. *IEEE Transactions on Automatic Control*, 2021.

Cassel, A. B., Cohen, A., and Koren, T. Efficient online linear control with stochastic convex costs and unknown dynamics. In *Proceedings of Thirty Fifth Conference on Learning Theory*, 2022.

Chen, G. and Teboulle, M. Convergence analysis of a proximal-like minimization algorithm using bregman functions. *SIAM Journal on Optimization*, 1993.

Chen, L., Jain, R., and Luo, H. Learning infinite-horizon average-reward Markov decision process with constraints. In *Proceedings of the 39th International Conference on Machine Learning*, 2022a.

Chen, X., Minasyan, E., Lee, J. D., and Hazan, E. Provable regret bounds for deep online learning and control. *arXiv preprint arXiv:2110.07807*, 2022b.

Cheng, R., Orosz, G., Murray, R. M., and Burdick, J. W. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

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

Dean, S., Tu, S., Matni, N., and Recht, B. Safely learning to control the constrained linear quadratic regulator. In *2019 American Control Conference (ACC)*, 2019.

Ding, D., Zhang, K., Basar, T., and Jovanovic, M. Natural policy gradient primal-dual method for constrained markov decision processes. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020.

Ding, D., Wei, X., Yang, Z., Wang, Z., and Jovanovic, M. Provably efficient safe exploration via primal-dual policy optimization. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, 2021.

Efroni, Y., Mannor, S., and Pirotta, M. Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189*, 2020.

Fisac, J. F., Akametalu, A. K., Zeilinger, M. N., Kaynama, S., Gillula, J., and Tomlin, C. J. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 2019.

Foster, D., Sarkar, T., and Rakhlin, A. Learning nonlinear dynamical systems from a single trajectory. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, 2020.

Foster, D. J. and Simchowitz, M. Logarithmic regret for adversarial online control. ICML'20. JMLR.org, 2020.

García, J., Fern, and o Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 2015.

Ghai, U., Chen, X., Hazan, E., and Megretski, A. Robust online control with model misspecification. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, 2022.

Ghosh, A., Zhou, X., and Shroff, N. Provably efficient model-free constrained RL with linear function approximation. In *Advances in Neural Information Processing Systems*, 2022.

Gradu, P., Hallman, J., and Hazan, E. Non-stochastic control with bandit feedback. In *Advances in Neural Information Processing Systems*, 2020.

Guo, H., Liu, X., Wei, H., and Ying, L. Online convex optimization with hard constraints: Towards the best of two worlds and beyond. In *Advances in Neural Information Processing Systems*, 2022.

Hajek, B. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Ann. Appl. Prob.*, 1982.

Hazan, E. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2016.

Hazan, E. and Singh, K. Introduction to online nonstochastic control. *arXiv preprint arXiv:2111.09619*, 2022.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances Neural Information Processing Systems (NeurIPS)*, 2018.

Koller, T., Berkenkamp, F., Turchetta, M., and Krause, A. Learning-based model predictive control for safe exploration. In *2018 IEEE Conference on Decision and Control (CDC)*, 2018.

Li, Y., Das, S., and Li, N. Online optimal control with affine constraints. *AAAI Conf. Artificial Intelligence*, 2021.

Li, Y., Zhang, T., Das, S., Shamma, J., and Li, N. Safe adaptive learning for linear quadratic regulators with constraints. *Technical report*, 2023.

Liu, T., Zhou, R., Kalathil, D., Kumar, P., and Tian, C. Learning policies with zero or bounded constraint violation for constrained MDPs. In *Advances in Neural Information Processing Systems*, 2021.

Liu, X., Yang, Z., and Ying, L. Online nonstochastic control with adversarial and static constraints. *arXiv preprint arXiv:2302.02426*, 2023.

Mahdavi, M., Jin, R., and Yang, T. Trading regret for efficiency: online convex optimization with long term constraints. *The Journal of Machine Learning Research*, 2012.

Mayne, D., Seron, M., and Raković, S. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 2005.

Minasyan, E., Gradu, P., Simchowitz, M., and Hazan, E. Online control of unknown time-varying dynamical systems. In *Advances in Neural Information Processing Systems*, 2021.

Neely, M. J. Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Communication Networks*, 2010.

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

Nonhoff, M. and Müller, M. A. An online convex optimization algorithm for controlling linear systems with state and input constraints. In *2021 American Control Conference (ACC)*, 2021.

O'Connell, M., Shi, G., Shi, X., Azizzadenesheli, K., Anandkumar, A., Yue, Y., and Chung, S.-J. Neural-fly enables rapid learning for agile flight in strong winds. *Science Robotics*, 2022.

Plevrakis, O. and Hazan, E. Geometric exploration for online control. In *Advances in Neural Information Processing Systems*, 2020.

Raković, S. V., Zhang, S., Sun, H., and Xia, Y. Model predictive control for linear systems under relaxed constraints. *IEEE Transactions on Automatic Control*, 2023.

Rawlings, J., Mayne, D., and Diehl, M. *Model Predictive Control: Theory, Computation, and Design*. 2017.

Simchowitz, M., Singh, K., and Hazan, E. Improper learning for non-stochastic control. In *Proceedings of Thirty Third Conference on Learning Theory*, 2020.

Sun, W., Dey, D., and Kapoor, A. Safety-aware algorithms for adversarial contextual bandit. In *International Conference on Machine Learning*, 2017.

Suo, D., Ghai, U., Minasyan, E., Gradu, P., Chen, X., Agarwal, N., Zhang, C., Singh, K., LaChance, J., Zadjel, T., Schottdorf, M., Cohen, D., and Hazan, E. Machine learning for mechanical ventilation control. 2021.

Tirmazi, M., Barker, A., Deng, N., Haque, M. E., Qin, Z. G., Hand, S., Harchol-Balter, M., and Wilkes, J. Borg: The next generation. EuroSys, 2020.

Vaswani, S., Yang, L., and Szepesvari, C. Near-optimal sample complexity bounds for constrained MDPs. In *Advances in Neural Information Processing Systems*, 2022.

Wabersich, K. P. and Zeilinger, M. N. Linear model predictive safety certification for learning-based control. In *2018 IEEE Conference on Decision and Control (CDC)*, 2018.

Wabersich, K. P., Krishnadas, R., and Zeilinger, M. N. A soft constrained mpc formulation enabling learning from trajectories with constraint violations. *IEEE Control Systems Letters*, 2022.

Wei, H., Liu, X., and Ying, L. Triple-q: A model-free algorithm for constrained reinforcement learning with sublinear regret and zero constraint violation. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, 2022.

Yi, X., Li, X., Yang, T., Xie, L., Chai, T., and Johansson, K. Regret and cumulative constraint violation analysis for online convex optimization with long term constraints. In *International Conference on Machine Learning*. PMLR, 2021a.

Yi, X., Li, X., Yang, T., Xie, L., Chai, T., and Johansson, K. H. Regret and cumulative constraint violation analysis for distributed online constrained convex optimization. *arXiv preprint arXiv:2105.00321*, 2021b.

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

Zeilinger, M. N., Jones, C. N., and Morari, M. Robust stability properties of soft constrained mpc. In *49th IEEE Conference on Decision and Control (CDC)*, 2010.