Online Robust Control of Nonlinear Systems with Large Uncertainty

Dimitar Ho Caltech
Hoang M. Le Microsoft Research
John Doyle Caltech
Yisong Yue Caltech

Abstract

Robust control is a core approach for controlling systems with performance guarantees that are robust to modeling error, and is widely used in real-world systems. However, current robust control approaches can only handle small system uncertainty, and thus require significant effort in system identification prior to controller design. We present an online approach that robustly controls a nonlinear system under large model uncertainty. Our approach is based on decomposing the problem into two sub-problems, "robust control design" (which assumes small model uncertainty) and "chasing consistent models", which can be solved using existing tools from control theory and online learning, respectively. We provide a learning convergence analysis that yields a finite mistake bound on the number of times performance requirements are not met and can provide strong safety guarantees, by bounding the worst-case state deviation. To the best of our knowledge, this is the first approach for online robust control of nonlinear systems with such learning theoretic and safety guarantees. We also show how to instantiate this framework for general robotic systems, demonstrating the practicality of our approach.

1 Introduction

We study the problem of online control for nonlinear systems with large model uncertainty under the requirement to provide upfront control-theoretic guarantees for the worst case online performance. Algorithms with such capabilities can enable us to (at least partially) sidestep undertaking laborious system identification tasks prior to robust controller design. Our goal is to design frameworks that can leverage existing approaches for online learning (to ensure fast convergence (Bubeck et al., 2020)) and robust control (which have control-theoretic guarantees under small model uncertainty (Zhou and Doyle, 1998)), in order to exploit the vast literature of prior work and simplify algorithm design. To this end, we introduce a class of problems, which we refer to as *online control* with *mistake guarantees* (OC-MG). To the best of our knowledge, this is the first rigorous treatment of online robust control of nonlinear systems under large uncertainty.

1.1 Problem Statement

Consider controlling a discrete-time *nonlinear dynamical system* with system equations:

$$x_{t+1} = f^*(t, x_t, u_t), \quad f^* \in \mathcal{F}, \tag{1}$$

where $x_t \in \mathcal{X}$ and $u_t \in \mathcal{U}$ denote the system state and control input at time step t and $\mathcal{X} \times \mathcal{U}$ denotes the state-action space. We assume that f^* is an *unknown* function and that we only know of an *uncertainty set* $\mathcal{F}$ which contains the true f^*.

Large uncertainty setting. We impose no further assumptions on $\mathcal{F}$ and explicitly allow $\mathcal{F}$ to represent arbitrarily large model uncertainties.

Control objective. The control objective is specified as a sequence $\boldsymbol{\mathcal{G}} = (\mathcal{G}_0, \mathcal{G}_1, \dots)$ of binary cost functions $\mathcal{G}_t : \mathcal{X} \times \mathcal{U} \mapsto \{0, 1\}$, where each function $\mathcal{G}_t$ encodes a desired condition per time-step t: $\mathcal{G}_t(x_t, u_t) = 0$ means the state x_t and input u_t meet the requirements at time t. $\mathcal{G}_t(x_t, u_t) = 1$ means that some desired condition is violated at time t and we will say that the system made a *mistake* at t. The performance metric of system trajectories $\boldsymbol{x} := (x_0, x_1, \dots)$ and $\boldsymbol{u} := (u_0, u_1, \dots)$ is the sum of incurred cost $\mathcal{G}_t(x_t, u_t)$ over the time interval $[0, \infty)$ and we denote this the *total number of mistakes*:

$$\#\text{ mistakes of } \boldsymbol{x}, \boldsymbol{u} = \sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t). \tag{2}$$

For a system state-input trajectory $(\boldsymbol{x}, \boldsymbol{u})$ to achieve an objective $\boldsymbol{\mathcal{G}}$, we want the above quantity to be finite, i.e.: eventually the system stops making mistakes and meets the requirements of the objective for all time.

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

Algorithm design goal. The goal is to design an online decision rule $u_t = \mathcal{A}(t, x_t, \ldots, x_0)$ such that regardless of the unknown $f^* \in \mathcal{F}$, we are guaranteed to have finite or even explicit upper-bounds on the total number of mistakes (2) of the online trajectories. Thus, we require a strong notion of robustness: $\mathcal{A}$ can control any system (1) with the certainty that the objective $\boldsymbol{\mathcal{G}}$ will be achieved after finitely many mistakes. It is suitable to refer to our problem setting as *online control* with *mistake guarantees*.

1.2 Motivation and related work

How can we design control algorithms for dynamical systems with strong guarantees without requiring much a-priori information about the system?

This question is of particular interest in safety-critical settings involving real physical systems, which arise in engineering domains such as aerospace, industrial robotics, automotive, energy plants (Vaidyanathan et al., 2016). Frequently, the designer of control policies faces two major challenges: guarantees and uncertainty.

Guarantees. Control policies can only be deployed if it can be certified in advance that the policies will meet desired performance requirements online. This makes the mistake guarantee w.r.t. objective $\boldsymbol{\mathcal{G}}$ a natural performance metric, as $\boldsymbol{\mathcal{G}}$ can incorporate control specifications such as tracking, safety and stability that often arise in practical nonlinear dynamical systems. The online learning for control literature mostly focused on linear systems or linear controllers (Dean et al., 2018; Simchowitz et al., 2018; Hazan et al., 2020; Chen and Hazan, 2020), with some emerging work on online control of nonlinear systems. One approach is incorporate stability into the neural network policy as part of RL algorithm (Donti et al., 2021). Alternatively, the parameters of nonlinear systems can be transformed into linear space to leverage linear analysis (Kakade et al., 2020; Boffi et al., 2020). These prior work focus on sub-linear regret bound, which is not the focus of our problem setup. We note that regret is not necessarily the only way to measure performance. For example, competitive ratio is an alternative performance metric for online learning to control (Goel and Wierman, 2019; Goel et al., 2019; Shi et al., 2020). In addition, our mistake guarantee requirement is stricter than the no-regret criterion and more amenable to control-theoretic guarantees. Specifically, (fast) convergence of $\frac{1}{T}\sum_{t=0}^{T} \mathcal{G}_t(x_t, u_t) \to 0$ does not imply the total number of mistakes $\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t)$ is bounded. We provide additional discussion on the large and growing literature on learning control for linear systems, as well as adaptive control techniques from control community in Appendix F.

Large uncertainty. Almost always, the dynamics of the real system are not known exactly and one has to resort to an approximate model. The most common approach in online learning for control literature (Dean et al., 2017) is to perform system identification (Ljung, 1999), and then use tools from robust control theory (Zhou and Doyle, 1998). Robust controller synthesis can provide policies with desired guarantees, if one can obtain an approximate model which is "provably close enough" to the real system dynamics. However, estimating a complex system to a desired accuracy level quickly becomes intractable in terms of computational and/or sample complexity. In the adversarial noise setting, system identification of simple linear systems with precision guarantees can be NP-hard (Dahleh et al., 1993). General approaches for nonlinear system identification with precision guarantees are for the most part not available (recently Mania et al. (2020) analyzed sample complexity under stochastic noise).

1.3 Overview of our approach

An alternative to SysID+control: using only rough models, learn to control online. While accurate models of real systems are hard to obtain, it is often easy to provide more qualitative or rough models of the system dynamics *without* requiring system identification. Having access to a rough system description, we aim to learn to control the real system from online data and provide control-theoretic guarantees on the online performance in advance.

Rough models as compactly parametrizable uncertainty sets. In practice, we never have the exact knowledge of f^* in advance. However, for engineering applications involving physical systems, the functional form for f^* can often be derived through first principles and application-specific domain knowledge. Conceptually, we can view the unknown parameters of the functional form as conveying both the 'modeled dynamics' and 'unmodeled (adversarial) disturbance' components of the ground truth f^* in the system $x_{t+1} = f^*(t, x_t, u_t)$. The range of unknown parameters will form a compact parameter set K, which in turn determines the size of model uncertainty set $\mathcal{F}$.

Definition 1.1 (Compact parametrization)**.** A tuple $(\mathbb{T}, \mathsf{K}, d)$ is a *compact parametrization* of $\mathcal{F}$, if (K, d) is a compact metric space and if $\mathbb{T} : \mathsf{K} \mapsto 2^{\mathcal{F}}$ is a mapping such that $\mathcal{F} \subset \bigcup_{\theta \in \mathsf{K}} \mathbb{T}(\theta)$.

We will work with candidate parameters $\theta \in \mathsf{K}$ of the system. Intuitively, consider a (non-unique) compact parameterization K of the uncertain model set $\mathcal{F}$, i.e., there exists a mapping $\mathbb{T} : \mathsf{K} \mapsto \mathcal{F}$ such that for each parameter $\theta \in \mathsf{K}, \mathbb{T}[\theta]$ represents a set of candidate models, ideally with small uncertainty. The uncertainty will be reflected more precisely by whether the system is robustly controllable if the true parameter *were* θ.

For concreteness, we give several simple examples of possible parametrizations K for different systems:

1. *Linear time-invariant system*: linear system with matrices A, B perturbed by bounded disturbance sequence $\boldsymbol{w} \in \ell_\infty$, $\|\boldsymbol{w}\|_\infty \le \eta$:
$$f^*(t, x, u) = Ax + Bu + w_t. \tag{3}$$
K can describe known bounds for $\theta = (A, B, \eta)$.
2. *Nonlinear system, linear parametrization*: nonlinear system, where dynamics are a weighted sum of nonlinear functions ψ_i perturbed by bounded disturbance sequence $\boldsymbol{w} \in \ell_\infty$, $\|\boldsymbol{w}\|_\infty \le \eta$:
$$f^*(t, x, u) = \sum_{i=1}^{M} a_i \psi_i(x, u) + w_t. \tag{4}$$
K can represent known bounds on $\theta = (\{a_i\}, \eta)$.
3. *Nonlinear system, nonlinear parametrization*: nonlinear system, with function g parametrized by fixed parameter vector $p \in \mathbb{R}^m$ (e.g., neural networks), perturbed by bounded disturbance sequence $\boldsymbol{w} \in \ell_\infty$, $\|\boldsymbol{w}\|_\infty \le \eta$:
$$f^*(t, x, u) = g(x, u; p) + w_t. \tag{5}$$
K can represent known bounds on $\theta = (p, \eta)$.

In these examples, the set $\mathcal{F}$ can be summarized as:
$$\mathcal{F} = \{f_\theta(x, u, w_t) \text{ with } \|\boldsymbol{w}\|_\infty \le \eta \text{ and } \theta \in \mathsf{K}\}, \tag{6}$$
where f_θ denotes one of functional forms on the right-hand side of equation (3), (4) or (5).

Online robust control algorithm. Given a compact parametrization $(\mathbb{T}, \mathsf{K}, d)$ for the uncertainty set $\mathcal{F}$, we design meta-algorithm $\mathcal{A}_\pi(\texttt{SEL})$ (Algorithm 1) that controls the system (1) online by invoking two sub-routines π and $\texttt{SEL}$ in each time step.

- *Consistent model chasing.* Procedure $\texttt{SEL}$ receives a finite data set $\mathcal{D}$, which contains state and input observations, and returns a parameter $\theta \in \mathsf{K}$. The design criterion is for each time t, the procedure $\texttt{SEL}$ selects θ_t such that the set of models $\mathbb{T}[\theta_t]$ stays "consistent" with $\mathcal{D}_t$, i.e., candidate models can *explain* the past data.
- *Robust oracle.* Procedure π receives a posited system parameter $\theta \in \mathsf{K}$ as input and returns a control policy $\pi[\theta] : \mathbb{N} \times \mathcal{X} \mapsto \mathcal{U}$ which can be evaluated at time t to compute a control action $u_t = \pi[\theta](t, x_t)$ based on the current state x_t. The control policy design is a robust control procedure, in the sense that $\pi[\theta]$ achieves $\mathcal{G}$ **if** $f^* \in \mathbb{T}[\theta]$ (which may not be the case at given time step).

Theoretical result. We will clarify the consistency and robustness requirements of the sub-routines π and $\texttt{SEL}$ in the next section. For now, we present an informal finite mistake guarantees for the online control scheme $\mathcal{A}_\pi(\texttt{SEL})$ from Algorithm 1.

Algorithm 1 Meta-Implementation of $\mathcal{A}_\pi(\texttt{SEL})$ for (OC-MG)

Require: procedures π and $\texttt{SEL}$
Initialization: $\mathcal{D}_0 \leftarrow \{\}$, x_0 is set to initial condition ξ_0
1: **for** $t = 0, 1, \ldots$ **to** ∞ **do**
2: $\quad \mathcal{D}_t \leftarrow$ append $(t, x_t, x_{t-1}, u_{t-1})$ to $\mathcal{D}_{t-1}$ (if $t \ge 1$) ▷ update online history of observations
3: $\quad \theta_t \leftarrow \texttt{SEL}[\mathcal{D}_t]$ ▷ present online data to $\texttt{SEL}$, get posited parameter θ_t
4: $\quad u_t \leftarrow \pi[\theta_t](t, x_t)$ ▷ query π for policy $\pi[\theta_t]$ and evaluate it
5: $\quad x_{t+1} \leftarrow f^*(t, x_t, u_t)$ ▷ system transitions with unknown f^* to next state
6: **end for**

Theorem (*Informal*)**.** *For any (adversarial) $f^* \in \mathcal{F}$, the online control scheme $\mathcal{A}_\pi(\texttt{SEL})$ described in Algorithm 1 guarantees a-priori that the trajectories $\boldsymbol{x}$, $\boldsymbol{u}$ will achieve the objective $\mathcal{G}$ after finitely many mistakes. The number of mistakes $\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t)$ is at most*
$$M_\rho^\pi \left(\frac{2 * \gamma * \textit{diameter}\ (\mathsf{K})}{\textit{oracle robustness margin}\ \rho} + 1 \right),$$
and the state $\|x_t\|$ is never larger than
$$\exp(\alpha_\pi * \gamma * \textit{diameter}\ (\mathsf{K}))(\|x_0\| + C_\pi),$$
where M_ρ^π denotes the worst case total mistakes of $\rho-$robust oracle π (under true parameter), α_π, C_π are "robustness" constants of π and γ is the "competitive ratio" of $\texttt{SEL}$.

This approach brings several attractive qualities:

- *Generality.* The result applies to a wide range of problem settings. The objective $\mathcal{G}$ and uncertainty set $\mathcal{F}$ serve as a flexible abstraction to represents a large class of dynamical systems and control-theoretic performance objectives.
- *Robust guarantees in the large uncertainty setting.* Our result applies in settings where only rough models are available. As an example, we can use the result to provide guarantees in control settings with unstable uncertain nonlinear systems where stabilizing policies are **not** known a-priori.
- *Decoupling algorithm design for learning and control.* The construction of the "robust oracle" π and the consistent model chasing procedure $\texttt{SEL}$ can be addressed with existing tools from control and learning. More generally, this perspective enables to decouple for the first time learning and control problems into separate robust control and online learning problems.

2 Main result

As summarized in Algorithm 1, the main ingredients of our approach are a robust control oracle π that returns a robust controller under posited system parameters, and an online algorithm $\texttt{SEL}$ that chases parameter sets

that are consistent with the data collected so far. Here we expand on the desired conditions of π and SEL.

2.1 Required conditions on procedure π

Online control with oracle under idealized conditions. Generally, procedure π is a map $\mathsf{K} \mapsto \mathcal{C}$ from parameter space K to the space $\mathcal{C} := \{\kappa : \mathbb{N} \times \mathcal{X} \mapsto \mathcal{U}\}$ of all (non-stationary) control policies of the form $u_t = \kappa(t, x_t)$. A desired property of π as an *oracle* is that π returns controllers that satisfy $\boldsymbol{\mathcal{G}}$ if the model uncertainty *were* small. In other words, if the uncertainty set $\mathcal{F}$ *were* contained in the set $\mathbb{T}[\theta]$, then control policy $\pi[\theta]$ could guarantee to achieve the objective $\boldsymbol{\mathcal{G}}$ with finite mistake guarantees. Further, in an idealized setting where the true parameter *were* known exactly, the oracle should return policy such that the system performance is robust to some level of bounded noise – which is a standard notion of *robustness*. We make this design specification more precise below and discuss how to instantiate the oracle in Section 3.

Idealized problem setting. Let θ^* be a parameter of the true dynamics f^*, and assume that online we have access to noisy observations $\boldsymbol{\theta} = (\theta_0, \theta_1, \dots)$, where each measurement θ_t is ρ-close to θ^*, under metric d. The online control algorithm queries π at each time-step and applies the corresponding policy $\pi[\theta_t]$. The resulting trajectories obey the equations:

$$x_{t+1} = f^*(t, x_t, u_t), \quad u_t = \pi[\theta_t](t, x_t) \tag{7a}$$

$$\theta_t \text{ s.t.: } d(\theta_t, \theta^*) \le \rho, \quad \text{where } f^* \in \mathbb{T}[\theta^*] \tag{7b}$$

To facilitate later discussion, define the set of all feasible trajectories of the dynamic equations (7) as the *nominal trajectories* $\mathcal{S}_{\mathcal{I}}[\rho; \theta]$ of the oracle:

Definition 2.1. For a time-interval $\mathcal{I} = [t_1, t_2] \subset \mathbb{N}$ and fixed $\theta \in \mathsf{K}$, let $\mathcal{S}_{\mathcal{I}}[\rho; \theta]$ denote the set of all pairs of finite trajectories $x_{\mathcal{I}} := (x_{t_1}, \dots, x_{t_2})$, $u_{\mathcal{I}} := (u_{t_1}, \dots, u_{t_2})$ which for $\theta^* = \theta$, satisfy conditions (7) with some feasible f^* and sequence $(\theta_{t_1}, \dots, \theta_{t_2})$.

Design specification for oracles. We will say that π is ρ-robust for some objective $\boldsymbol{\mathcal{G}}$, if all trajectories in $\mathcal{S}_{\mathcal{I}}[\rho; \theta]$ achieve $\boldsymbol{\mathcal{G}}$ after finitely many mistakes. We distinguish between robustness and uniform robustness, which we define precisely below:

Definition 2.2 (robust oracle)**.** For each $\rho \ge 0$ and $\theta \in \mathsf{K}$, define the quantity $m^{\pi}_{\rho}(\theta)$ as

$$m^{\pi}_{\rho}(\theta) := \sup_{\mathcal{I}=[t,t'] \,:\, t<t'} \; \sup_{(x_{\mathcal{I}}, u_{\mathcal{I}}) \in \mathcal{S}_{\mathcal{I}}[\rho;\theta]} \sum_{t \in \mathcal{I}} \mathcal{G}_t(x_t, u_t)$$

If $m^{\pi}_0(\theta) < \infty$ for all $\theta \in \mathsf{K}$, we call π an *oracle* for $\boldsymbol{\mathcal{G}}$ w.r.t. parametrization $(\mathbb{T}, \mathsf{K}, d)$. In addition, we say an oracle π is (uniformally) ρ-robust, if the corresponding property below holds:

$$\rho\text{-robust:} \quad m^{\pi}_{\rho}(\theta) < \infty \text{ for all } \theta \in \mathsf{K}$$

$$\text{uniformly } \rho\text{-robust:} \quad M^{\pi}_{\rho} := \sup_{\theta \in \mathsf{K}} m^{\pi}_{\rho}(\theta) < \infty$$

If it exists, M^{π}_{ρ} is the *mistake constant* of π.

π is a robust oracle for $\boldsymbol{\mathcal{G}}$ if the property in definition above holds for some $\rho > 0$ and we refer to ρ as the robustness margin. The mistake constant M^{π}_{ρ} can be viewed as a robust offline benchmark: it quantifies how many mistakes we would make in the worst-case, if we could use the oracle π under idealized conditions, i.e., described by (7).

2.2 Required conditions on procedure SEL

We now describe required conditions for SEL, and will discuss specific strategies to design SEL in Section 4.

Let $\mathbb{D}$ be the space of all tuples of time-indexed data points $d_i = (t_i, x_i^+, x_i, u_i)$:

$$\mathbb{D} := \{(d_1, \dots, d_N) \mid d_i \in \mathbb{N} \times \mathcal{X} \times \mathcal{X} \times \mathcal{U},\ N < \infty\},$$

Generally procedure SEL takes as an input a data set $\mathcal{D} \in \mathbb{D}$ and outputs a parameter $\texttt{SEL}[\mathcal{D}] \in \mathsf{K}$.

Intuitively, given a data set $\mathcal{D} = (d_1, \dots, d_N)$ of tuples $d_i = (t_i, x_i^+, x_i, u_i)$, *any* candidate $f \in \mathcal{F}$ which satisfies $x_i^+ = f(t_i, x_i, u_i)$ for all $1 \le i \le N$ is *consistent* with $\mathcal{D}$. We will extend this definition to parameters: $\theta \in \mathsf{K}$ is a consistent parameter for $\mathcal{D}$, if $\mathbb{T}[\theta]$ contains at least one function f which is consistent with $\mathcal{D}$. Correspondingly, we will define for some data $\mathcal{D}$, the set of all consistent parameters as $\mathsf{P}(\mathcal{D})$:

Definition 2.3 (Consistent Sets)**.** For all $\mathcal{D} \in \mathbb{D}$, define the set $\mathsf{P}(\mathcal{D})$ as:

$$\mathsf{P}(\mathcal{D}) := \text{closure}\big(\,\{\text{all } \theta \in \mathsf{K} \text{ such that } (9)\}\,\big), \tag{8}$$

$$\exists f \in \mathbb{T}(\theta) : \forall (t, x^+, x, u) \in \mathcal{D} : x^+ = f(t, x, u). \tag{9}$$

Chasing conditions for selection SEL. The design specification for subroutine SEL is to output a consistent parameter $\theta = \texttt{SEL}[\mathcal{D}]$ for given data set $\mathcal{D}$, provided such a parameter $\theta \in \mathsf{K}$ exists. In addition, we require that for a stream of data $\mathcal{D}_t = (d_1, \dots, d_t)$ collected from the same system $f \in \mathcal{F}$, the sequence of parameters $\theta_t = \texttt{SEL}[\mathcal{D}_t]$ posited by SEL satisfies certain convergence properties.

Definition 2.4 (consistent model chasing)**.** Let $\mathcal{D}_t = (d_1, \dots, d_t)$ be a stream of data generated by:

$$x_{t+1} = f(t, x_t, u_t) \qquad x_0 = \xi_0,\ f \in \mathcal{F}$$
$$d_t = (t, x_t, x_{t-1}, u_{t-1}).$$

and let $\theta_t = \texttt{SEL}[\mathcal{D}_t]$ define a parameter sequence $\boldsymbol{\theta}$ returned by SEL. We say that SEL is *chasing consistent models* if $\theta_t \in \mathsf{P}(\mathcal{D}_t)$, $\forall t$ and $\lim_{t\to\infty} \theta_t \in \mathsf{P}(\mathcal{D}_\infty)$ holds regardless of initial condition ξ_0, input sequence $\boldsymbol{u}$ or $f \in \mathcal{F}$. We further say SEL is γ-*competitive* if

$$\sum_{t=1}^{\infty} d(\theta_t, \theta_{t-1}) \le \gamma \max_{\theta_0 \in \mathsf{K}} d(\mathsf{P}(\mathcal{D}_\infty), \theta_0),$$

holds for a fixed constant $\gamma > 0$, which we call the *competitive ratio*. ($d(\mathsf{S}, p) := \inf_{q \in \mathsf{S}} d(q, p)$)

2.3 Main theorem

Assuming for now that π and SEL meet the required specifications, we can provide the overall guarantees for the algorithm. Let $(\mathbb{T}, \mathsf{K}, d)$ be a compact parametrization of a given uncertainty set $\mathcal{F}$. Let π be robust per Definition 2.2 and SEL return consistent parameters per Definition 2.4. We apply the online control strategy $\mathcal{A}_\pi(\texttt{SEL})$ described in Algorithm 1 to system $x_{t+1} = f^*(t, x_t, u_t)$ with unknown dynamics $f^* \in \mathcal{F}$ and denote $(\boldsymbol{x}, \boldsymbol{u})$ as the corresponding state and input trajectories. The mistakes will be bounded as follows:

Theorem 2.5. *Assume that SEL chases consistent models and that π is an oracle for an objective $\boldsymbol{\mathcal{G}}$. Then the following mistake guarantees hold:*

(i) If π is robust then $(\boldsymbol{x}, \boldsymbol{u})$ always satisfy:

$$\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) < \infty.$$

(ii) If π is uniformly ρ-robust and SEL is γ-competitive, then $(\boldsymbol{x}, \boldsymbol{u})$ obey the inequality:

$$\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) \leq M_\rho^\pi \left(\tfrac{2\gamma}{\rho} \mathrm{diam}(\mathsf{K}) + 1 \right).$$

Theorem 2.6. *If π is (α, β)-single step robust* [1] *and SEL is γ-competitive, $\|x_t\|_t$ is always bounded by*

$$\|x_t\| \leq e^{\alpha\gamma \mathrm{diam}(\mathsf{K})} \left(e^{-t}\|x_0\| + \beta \frac{e}{e-1} \right) \tag{10}$$

Theorem 2.5 can be invoked on any learning and control method that instantiates $\mathcal{A}_\pi(\texttt{SEL})$. It offers a set of sufficient conditions to verify whether a learning agent $\mathcal{A}_\pi(\texttt{SEL})$ can provide mistake guarantees: We need to show that w.r.t. some compact parametrization $(\mathbb{T}, \mathsf{K}, d)$ of the uncertainty set $\mathcal{F}$, π operates as a robust oracle for some objective $\boldsymbol{\mathcal{G}}$, and that SEL satisfies strong enough chasing properties. Theorem 2.6 provides a general safety guarantee for $\mathcal{A}_\pi(\texttt{SEL})$ without requiring π to be an oracle for any particular objective $\boldsymbol{\mathcal{G}}$.

Theorem 2.5 also suggests a design philosophy of decoupling the learning and control problem into two separate problems while retaining the appropriate guarantees: (1) design a robust oracle π for a specified control goal $\boldsymbol{\mathcal{G}}$; and (2) design an online selection procedure SEL that satisfies the chasing properties defined in Definition 2.4.

We discuss in section 3 how addressing (1) is a pure robust control problem and briefly overview the main available methods. Designing procedures SEL with the properties stated in Definition 2.4 poses a new class of online learning problems. We propose in Section 4 a reduction of SEL to the well-known nested convex body chasing problem, which enables design and analysis of competitive SEL procedures.

[1] see appendix A

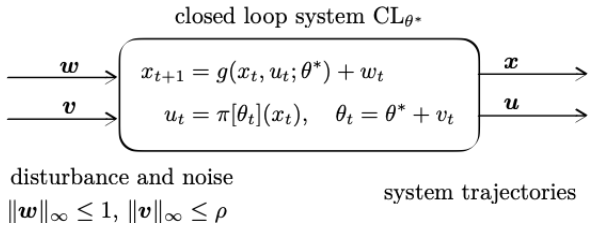

Figure 1: closed loop CL_{θ^*}: An idealized setting in which we have noisy measurements of the true θ^*.

2.4 Extensions

In the appendix section A we present the full version of the main result which considers a broader definition for robust oracle and chasing properties (see definition A.2 and A.6).

Oracle policies with memory. The results assume that π returns static policies of the type $(t, x) \mapsto u$. However, this assumption is only made for ease of exposition. The results of Theorem 2.5 also hold in the case where π returns policies which have an internal state, as long as we can define the internal state to be shared among all oracle policies, i.e.: as part of the oracle implementation online, we update the state z_t at each step t according to some fixed update rule h:

$$z_t = h(t, z_{t-1}, x_t, u_t, \ldots, x_0, u_0),$$

and control policies $\pi[\theta]$, $\theta \in \mathsf{K}$ are maps $(t, x, z) \mapsto u$ which we evaluate at time t as $u_t = \pi[\theta](t, x_t, z_t)$.

3 Robust Control Oracle

Designing robust oracles π introduced in Definition 2.2 (extended version in A.2) can be mapped to well-studied problems in robust control theory. We use a simplified problem setting to explain this correspondence.

Consider a class of system of the form $x_{t+1} = g(x_t, u_t; \theta^*) + w_t$, $\|w_t\| \leq 1$, where θ^* is an unknown system parameter which lies in a known compact set $\mathsf{K} \subset \mathbb{R}^m$. We represent the uncertainty set as $\mathcal{F} = \cup_{\theta \in \mathsf{K}} \mathbb{T}[\theta]$ with $\mathbb{T}[\theta] := \{f^* : t, x, u \mapsto g(x, u; \theta) + w_t \mid \|\boldsymbol{w}\|_\infty \leq 1\}$ Let $\pi : \mathsf{K} \mapsto \mathcal{C}$ be a procedure which returns state feedback policies $\pi[\theta] : \mathcal{X} \mapsto \mathcal{U}$ for a given $\theta \in \mathsf{K}$. Designing an uniformly ρ-robust oracle π can be equivalently viewed as making the closed loop system (described by (7)) of the idealized setting robust to disturbance and noise. For the considered example, the closed loop is depicted in Figure 1 and is represented by CL_θ^* which maps system perturbations $(\boldsymbol{w}, \boldsymbol{v})$ to corresponding system trajectories $\boldsymbol{x}$, $\boldsymbol{u}$. We call π a uniformly ρ-robust oracle if the cost-performance (measured as $\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t)$) of the closed loop CL_θ^* is robust to

disturbances of size 1 and noise of size ρ for any $\theta^* \in \mathsf{K}$. For any noise $\|\boldsymbol{v}\|_\infty \leq \rho$ and disturbance $\|\boldsymbol{w}\|_\infty \leq 1$, the performance cost has to be bounded as:

$$\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) \leq M_\rho^\pi \quad \text{or} \quad \sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) \leq m_\rho^\pi(\|x_0\|),$$

for some fixed constant M_ρ^π or fixed function m_ρ^π, in case we can only establish local properties. Now, if we identify the cost functions $\mathcal{G}_t$ with their level sets $\mathsf{S}_t := \{(x,u) \mid \mathcal{G}_t(x,u) = 0\}$, we can phrase the former equivalently as a form of robust trajectory tracking problem or a set-point control problem (if $\mathcal{G}_t$ is time independent) (Khalil and Grizzle, 2002). It is common in control theory to provide guarantees in the form of convergence rates (finite-time or exponential convergence) on the tracking-error; these guarantees can be directly mapped to M_ρ^π and $m_\rho^\pi(\,\cdot\,)$.

Available methods for robust oracle design. Many control methods exist for robust oracle design. Which method to use depends on the control objective $\boldsymbol{\mathcal{G}}$, the specific application, and the system class (linear/nonlinear/hybrid, etc.). For a broad survey, see (Zhou et al., 1996; Spong and Vidyasagar, 1987; Spong, 1992a) and references therein. We characterize two general methodologies (which can also be combined):

- *Robust stability analysis focus:* In an initial step, we use analytical design principles from robust nonlinear and linear control design to propose an oracle $\pi[\theta](x)$ in closed-form for all θ and x. In a second step we prove robustness using analysis tools such as for example *Input-to-State Stability* (ISS) stability analysis (Jiang et al., 1999) or robust set invariance methods (Rakovic et al., 2006; Rakovic and Baric, 2010)).
- *Robust control synthesis:* If the problem permits, we can also directly address the control design problem from a computational point of view, by formulating the design problem as an optimization problem and compute for a control law with desired guarantees directly. This can happen partially online, partially offline. Some common nonlinear approaches are robust (tube-based) MPC (Mayne et al., 2011; Borrelli et al., 2017), SOS-methods (Parrilo, 2000),(Aylward et al., 2008), Hamilton-Jacobi reachability methods (Bansal et al., 2017).

There are different advantages and disadvantages to both approaches and it is important to point out that robust control problems are not always tractably solvable. See (Blondel and Tsitsiklis, 2000; Braatz et al., 1994) for simple examples of robust control problems which are NP-hard. The computational complexity of robust controller synthesis tends to increase (or even be potentially infeasible) with the complexity of the system of interest; it also further increases as we try optimize for larger robustness margins ρ.

The dual purpose of the oracle. In our framework, access to a robust oracle is a necessary prerequisite to design learning and control agents $\mathcal{A}_\pi(\texttt{SEL})$ with mistake guarantees. However this is a mild assumption and is often more enabling than restrictive. First, it represents a natural way to ensure well-posedness of the overall learning and control problem; If robust oracles cannot be found for an objective, then the overall problem is likely intrinsically hard or ill-posed (for example necessary fundamental system properties like stabilizability/ detectability are not satisfied).

Second, the oracle abstraction enables a modular approach to robust learning and control problems, and directly leverages existing powerful methods in robust control: Any model-based design procedure π which works well for the small uncertainty setting (i.e.: acts as a robust oracle) can be augmented with an online chasing algorithm SEL (with required chasing properties) to provide robust control performance (in the form of mistake guarantees) in the large uncertainty setting via the augmented algorithm $\mathcal{A}_\pi(\texttt{SEL})$.

4 Chasing consistent models

To provide mistake guarantees for the Algorithm $\mathcal{A}_\pi(\texttt{SEL})$, the procedure SEL must satisfy the "chasing property" as defined in Definition 2.4.

4.1 Chasing consistent models competitively

Assume that at each timestep $1 \leq k \leq T$, we are given a data point $d_k = (t_k, x_k^+, x_k, u_k)$, where $t_k \in \mathbb{N}$, $x_k^+, x_k \in \mathcal{X}$, $u_k \in \mathcal{U}$ and assume that at time $t = 0$, it is known that each data point d_k will satisfy the equation $x_k^+ = f^*(t_k, x_k, u_k)$ for some fixed, but unknown dynamics $f^* \in \mathcal{F}$. Denote $\mathcal{D}_k := (d_1, \ldots, d_k)$ as the tuple of collected observation until time k. Given a compact parametrization $(\mathbb{T}, \mathsf{K}, d)$ of $\mathcal{F}$, we are looking for a chasing procedure SEL which given a stream of online data $\mathcal{D}_t = (d_1, \ldots, d_t)$, finds as quickly as possible a parameter θ^* consistent with $\mathcal{D}_t$.[2]

We quantify performance using competitiveness, which is a common performance objective in the design of online learning algorithms (Koutsoupias and Papadimitriou, 2000; Borodin and El-Yaniv, 2005; Chen et al., 2018; Goel and Wierman, 2019; Shi et al., 2020; Yu et al., 2020). Starting with some initial θ_0, a sequence of parameters $\theta_1, \ldots, \theta_T$ returned by SEL will be evaluated by its *total moving cost* $\sum_{j=1}^{T} d(\theta_j, \theta_{j-1})$. For each time T, we benchmark against the best parameter selection $\theta_1, \ldots, \theta_T$ in hindsight, that is the sequence with smallest moving cost assuming we know

[2]Notice that if we know that the data $\mathcal{D}_T$ is collected from a single trajectory, we can add the conditions $t_{k+1} = t_k + 1$ and $x_{k+1} = x_k^+$ as known constraints.

the set $\mathsf{P}(\mathcal{D}_T)$. It is clear that the best possible choice for $k \geq 1$ is to choose θ_k as some constant parameter: $\theta^* \in \arg\min_{\theta' \in \mathsf{P}(\mathcal{D}_T)} d(\theta_0, \theta')$. We denote the corresponding optimal offline cost as $\mathrm{OPT}(\mathcal{D}_T; \theta_0) = \min_{\theta' \in \mathsf{P}(\mathcal{D}_T)} d(\theta_0, \theta')$ and evaluate it at the worst possible starting condition θ_0 to define the offline benchmark $\mathrm{OPT}^*(\mathcal{D}_T) = \max_{\theta_0 \in \mathsf{P}(\mathcal{D}_0)} \mathrm{OPT}(\mathcal{D}_T; \theta_0)$.

Definition 4.1 (Competitive consistent model chasing)**.** Let $J_T(\mathcal{D}_T; \theta_0)$ denote the total moving cost of the online algorithm for a data sequence $\mathcal{D}_T$ and initial selection θ_0. We call an algorithm γ-competitive for consistent model chasing in the parametrization $(\mathbb{T}, \mathsf{K}, d)$, if for any data sequence $\mathcal{D}_T$, sequence length $T \geq 0$ and $\theta_0 \in \mathsf{K}$, the cost $J_T(\mathcal{D}_T; \theta_0)$ is bounded as:

$$J_T(\mathcal{D}_T; \theta_0) \leq \gamma \mathrm{OPT}^*(\mathcal{D}_T). \tag{11}$$

4.2 Reduction to chasing convex bodies

The main difficulty in selecting the parameters θ_t to solve CMC competitively is that, for any time $t < T$, we cannot guarantee to select a parameter θ_t which is guaranteed to lie in the future consistent set $\mathsf{P}(\mathcal{D}_T)$. However, a sequence of consistent sets is always nested, i.e. $\mathsf{K} \supset \mathsf{P}(\mathcal{D}_1) \cdots \supset \mathsf{P}(\mathcal{D}_T)$. This inspires a competitive procedure for the CMC problem, through a reduction to a known online learning problem.

Under the following Assumption 4.2 we can reduce the CMC problem to a well-known problem of *nested convex body chasing* (NCBC) (Bubeck et al., 2020).

Assumption 4.2. Given a compact parametrization $(\mathbb{T}, \mathsf{K}, d)$ of the uncertainty set $\mathcal{F}$, the consistent sets $\mathsf{P}(\mathcal{D})$ are always convex for any data set $\mathcal{D} \in \mathbb{D}$.

This assumption is valid for instance for the general class of robotic manipulation problems (section 5).

Nested convex body chasing (NCBC). In NCBC, we have access to online nested sequence $\mathsf{S}_0, \mathsf{S}_1, \ldots, \mathsf{S}_T$ of convex sets in some metric space $(\mathcal{M}, d)$ (i.e.: $\mathsf{S}_t \subset \mathsf{S}_{t-1}$). The learner selects at each time t a point p_t from S_t. The goal in competitive NCBC is to produce $p_1, \ldots, p_T$ online such that the total moving cost $\sum_{j=1}^{T} d(p_j, p_{j-1})$ at time T is competitive with the offline-optimum, i.e. there is some $\gamma > 0$ s.t. $\sum_{j=1}^{T} d(p_j, p_{j-1}) \leq \gamma \mathrm{OPT}_T$, where $\mathrm{OPT}_T := \max_{p_0 \in \mathsf{S}_0} \min_{p \in \mathsf{S}_T} d(p, p_0)$.

Remark 1. NCBC is a special case of the more general convex body chasing (CBC) problem, first introduced by (Friedman and Linial, 1993), which studied competitive algorithms for metrical goal systems.

Let the sequence of convex consistent sets $\mathsf{P}(\mathcal{D}_t)$ be the corresponding S_t of the NCBC problem, any γ-competitive agent $\mathcal{A}$ for the NCBC problem can instantiate a γ-competitive selection for competitive model-chasing, as summarized in the following reduction:

Algorithm 2 γ-competitive CMC selection $\texttt{SEL}_{\mathrm{NCBC}}$

Require: γ-competitive NCBC algorithm $\mathcal{A}_{\mathrm{NCBC}}$, consistent set map $\mathsf{P} : \mathbb{D} \mapsto 2^{\mathsf{K}}$
1: **procedure** $\texttt{SEL}_{\mathrm{NCBC}}(t, x^+, x, u)$
2: $\quad \mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup (t, x^+, x, u)$
3: $\quad \mathsf{S}_t \leftarrow \mathsf{P}(\mathcal{D}_t)$ ▷ construct/update new consistent set
4: $\quad$ present set S_t to $\mathcal{A}_{\mathrm{NCBC}}$
5: $\quad \mathcal{A}_{\mathrm{NCBC}}$ chooses $\theta_t \in \mathsf{S}_t$
6: $\quad$ **return** θ_t
7: **end procedure**

Proposition 4.3. *Consider the setting of Assumption 4.2. Then any γ-competitive algorithm for NCBC in metric space (K, d) instantiates via Algorithm 2 a γ-competitive CMC procedure $\texttt{SEL}_{\mathrm{NCBC}}$ for the parametrization $(\mathbb{T}, \mathsf{K}, d)$.*

Simple competitive NCBC-algorithms in euclidean space $\mathbb{R}^n$. When (K, d) is a compact euclidean finite dimensional space, recent exciting progress on the NCBC problem provides a variety of competitive algorithms (Argue et al., 2019, 2020; Bubeck et al., 2020; Sellke, 2020) that can instantiate competitive selections per Algorithm 2.

We highlight two simple instantiations based on results in (Argue et al., 2019), and (Bubeck et al., 2020). Both algorithms can be tractably implemented in the setting of Assumption 4.2. The selection criteria for $\texttt{SEL}_p(\mathcal{D}_t)$ and $\texttt{SEL}_s(\mathcal{D}_t)$ is defined as:

$$\texttt{SEL}_p(\mathcal{D}_t) := \arg\min_{\theta \in \mathsf{P}(\mathcal{D}_t)} \|\theta - \texttt{SEL}_p(\mathcal{D}_{t-1})\|, \tag{12a}$$

$$\texttt{SEL}_s(\mathcal{D}_t) := s(\mathsf{P}(\mathcal{D}_t)), \tag{12b}$$

where $\texttt{SEL}_p$ defines simply a greedy projection operator and where $\texttt{SEL}_s$ selects according to the *Steiner-Point* $s(\mathsf{P}(\mathcal{D}_t))$ of the consistent set $\mathsf{P}(\mathcal{D}_t)$ at time t.

Definition 4.4 (Steiner Point)**.** For a convex body K, the Steiner point is defined as the following integral over the $n-1$ dimensional sphere $\mathbb{S}^{n-1}$:

$$s(K) = n \int_{v \in \mathbb{S}^{n-1}} \max_{x \in \mathsf{K}} \langle v, x \rangle v dv. \tag{13}$$

Remark 2. As shown in Bubeck et al. (2020), the Steiner point can be approximated efficiently by solving randomized linear programs. We take this approach for our later empirical validation in section 6.

The competitive analysis presented in (Bubeck et al., 2020) can be easily adapted to establish that $\texttt{SEL}_p$ and $\texttt{SEL}_s$ are competitive for the CMC problem:

Corollary 4.5 (of Theorem 1.3 (Argue et al., 2019), and Theorem 2.1 (Bubeck et al., 2020))**.** *Assume K is a compact convex set in $\mathbb{R}^n$ and $d(x, y) := \|x - y\|_2$. Then, the procedures $\texttt{SEL}_p$ and $\texttt{SEL}_s$ are competitive (CMC)-algorithms with constants γ_p and γ_s:*

$$\gamma_p = (n-1) n^{\frac{n+1}{2}}, \qquad \gamma_s = \frac{n}{2}. \tag{14}$$

4.3 Constructing consistent sets online

Constructing consistent sets $\mathsf{P}(\mathcal{D}_t)$ online can be addressed with tools from set-membership identification. For a large collection of linear and nonlinear systems, the sets $\mathsf{P}(\mathcal{D})$ can be constructed efficiently online. Such methods have been developed and studied in the literature of set-membership identification, for a recent survey see (Milanese et al., 2013). Moreover it is often possible to construct $\mathsf{P}(\mathcal{D})$ as an intersection of finite half-spaces, allowing for tractable representations as LPs. To see a particularly simple example, consider the following nonlinear system with some unknown parameters $\alpha^* \in \mathbb{R}^M$ and η^*, where w_t is a vector with entries in the interval $[-\eta^*, \eta^*]$:

$$x_{t+1} = \sum_{i=1}^{M} \alpha_i^* \psi_i(x_t, u_t) + w_t, \tag{15}$$

where $\psi_i : \mathcal{X} \times \mathcal{U} \mapsto \mathcal{X}$ are M known nonlinear functions. If we represent the above system as an uncertain system $\mathbb{T}$ with parameter $\theta^* = [\alpha^*; \eta^*]$, it is easy to see that the consistent sets $\mathsf{P}(\mathcal{D})$ for some data $\mathcal{D} = \{(x_i^+, x_i, u_i) \,|\, 1 \le i \le H\}$ of H observations takes the form of a polyhedron:

$$\mathsf{P}(\mathcal{D}) = \{\theta = [\alpha; \eta] \mid \text{s.t. (16) for all } 1 \le i \le H\},$$

defined by the inequalities:

$$[\psi_1(x_i, u_i), \dots, \psi_M(x_i, u_i)]\alpha \le x_i^+ + \mathbb{1}\eta, \tag{16a}$$

$$[\psi_1(x_i, u_i), \dots, \psi_M(x_i, u_i)]\alpha \ge x_i^+ - \mathbb{1}\eta. \tag{16b}$$

We can see that any linear discrete-time system can be put into the above form (15). Moreover, as shown in section 5 the above representation also applies for a large class of (nonlinear) robotics system.

5 Examples

Here we demonstrate two illustrative examples of how to instantiate our approach: learning to stabilize of a scalar linear system, and learning to track a trajectory on a fully actuated robotic system. Detailed derivations are provided in the Appendix E.

5.1 Control of uncertain scalar linear system

Consider the basic setting of controlling a scalar linear system with unknown parameters and bounded disturbance $|w_k| \le \eta < 1$:

$$x_{k+1} = \alpha^* x_k + \beta^* u_k + w_k =: f^*(k, x_k, u_k),$$

with the goal to reach the interval $\mathcal{X}_\mathcal{T} = [-1, 1]$ and remain there. Equivalently, this goal can be expressed as the objective $\boldsymbol{\mathcal{G}} = (\mathcal{G}_0, \mathcal{G}_1, \dots)$ with cost functions:

$$\mathcal{G}_t(x, u) := \begin{cases} 0, & \text{if } |x| \le 1 \\ 1, & \text{else} \end{cases}, \quad \forall t \ge 0,$$

since "reaching and remaining in $\mathcal{X}_\mathcal{T}$" is equivalent to achieving $\boldsymbol{\mathcal{G}}$ within finite mistakes. The true parameter $\theta^* = (\alpha^*, \beta^*)$ lies in the set $\mathsf{K} = [-a, a] \times [1, 1 + 2b_\Delta]$ with known $a > 0$, $\eta < 1$ and $b_\Delta > 0$.

Parametrization of uncertainty set. We describe the uncertainty set $\mathcal{F} = \cup_{\theta \in \mathsf{K}} \mathbb{T}[\theta]$ through the compact parametrization $(\mathbb{T}, \mathsf{K}, d)$ with parameter space (K, d), $d(x, y) := \|x - y\|$, $\|x\| := |x_1| + a|x_2|$ and the collection of models:

$$\mathbb{T}[\theta] := \{t, x, u \mapsto \theta_1 x + \theta_2 u + w_t \mid \|\boldsymbol{w}\|_\infty \le \eta\}.$$

Robust oracle π. We use the simple deadbeat controller: $\pi[\theta](t, x) := -(\theta_1/\theta_2)x$. It can be easily shown that π is a locally ρ-uniformly robust oracle for $\boldsymbol{\mathcal{G}}$ for any margin in the interval $(0, 1 - \eta)$.

Construction of consistent sets via LPs. The consistent sets $\mathsf{P}(\mathcal{D}_t)$ are convex, can be constructed online and are the intersection of K with $2t$ halfspaces:

$$\{\theta \in \mathsf{K} \,|\, \text{s.t.: } \forall i < t : |\theta_1 x_i + \theta_2 u_i - x_{i+1}| \le \eta\},$$

Competitive $\mathtt{SEL}_s$ via NCBC. We instantiate $\mathtt{SEL}_s$ using Algorithm 2 with Steiner point selection. From Corollary 4.5, we have $\gamma_s = \frac{n}{2} = 1$.

Mistake guarantee for $\mathcal{A}_\pi(\mathtt{SEL}_s)$ The extension of the results in Theorem A.12, (ii) apply and we obtain for $\mathcal{A}_\pi(\mathtt{SEL}_s)$ and the stabilization objective $\boldsymbol{\mathcal{G}}$, the following mistake guarantee:

$$\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) \le 2e\varepsilon^2 + \varepsilon, \quad \varepsilon = 2(a + b_\Delta).$$

The above inequality shows that the worst-case total number of mistakes grows quadratically with the size of the initial uncertainty ε in the system parameters. Notice, however that the above inquality holds for arbitrary large choices of a and b_Δ. Thus, $\mathcal{A}_\pi(\mathtt{SEL}_s)$ gives finite mistake guarantees for this problem setting for arbitrarily large system parameter uncertainties.

5.2 Trajectory following in robotic systems

We consider uncertain fully-actuated robotic systems. A vast majority of robotic systems can be modeled via the robotic equation of motion (Murray, 2017):

$$\mathbf{M}_\eta(q)\ddot{q} + \mathbf{C}_\eta(q, \dot{q})\dot{q} + \mathbf{N}_\eta(q, \dot{q}) = \tau + \tau_d, \tag{17}$$

where $q \in \mathbb{R}^n$ is the multi-dimensional generalized coordinates of the system, $\dot{q}$ and $\ddot{q}$ are its first and second (continuous) time derivatives, $\mathbf{M}_\eta(q), \mathbf{C}_\eta(q, \dot{q}), \mathbf{N}_\eta(q, \dot{q})$ are matrix and vector-value functions that depend on the parameters $\eta \in \mathbb{R}^m$ of the robotic system, i.e. η comes from parametric physical model. Often, τ is the control action (e.g., torques and forces of actuators), which acts as input of the system. Disturbances and other uncertainties present in the system can be modeled as additional torques $\tau_d \in \mathbb{R}^n$ perturbing the equations. The disturbances are bounded as $|\tau_d(t)| \le \omega$, where $\omega \in \mathbb{R}^n$ and the inequality is entry-wise.

Consider a system with unknown η^*, ω^*, where the

parameter $\theta^* = [\eta^*; \omega^*]$ is known to be contained in a bounded set K. Our goal is to track a desired trajectory q_d, given as a function of time $q_d : \mathbb{R} \mapsto \mathbb{R}^n$, within ϵ precision, i.e.: Denoting $x = [q^\top, \dot{q}^\top]^\top$ as the state vector and $x_d = [q_d^\top, \dot{q}_d^\top]^\top$ as the desired state, we want the system state trajectory $x(t)$ to satisfy:

$$\limsup_{t\to\infty} \|x(t) - x_d(t)\| \leq \epsilon. \tag{18}$$

As common in practice, we assume we can observe the sampled measurements $x_k := x(t_k)$, $x_k^d := x^d(t_k)$ and apply a constant control action (zero-order-hold actuation) $\tau_k := \tau(t_k)$ at the discrete time-steps $t_k = kT_s$ with small enough sampling-time T_s to allow for continuous-time control design and analysis.

Control objective $\mathcal{G}^\epsilon$. We phrase trajectory tracking as a control objective $\mathcal{G}^\varepsilon$ with the cost functions:

$$\mathcal{G}_k^\epsilon(x,u) := \begin{cases} 0, & \text{if } \|x - x_k^d\| \leq \epsilon \\ 1, & \text{else} \end{cases}, \quad \forall k \geq 0.$$

Robust oracle design. We outline in Appendix G how to design a control oracle using established methods for robotic manipulators (Spong, 1992b).

Constructing consistent sets. For many robotic systems (for example robot manipulators), one can derive from first principles (Murray, 2017) that the left-hand-side of (53) can be factored into a $n \times m$ matrix of known functions $\mathbf{Y}(q, \dot{q}, \ddot{q})$ and a constant vector $\eta \in \mathbb{R}^m$. We can then construct consistent sets at each time t as polyhedrons of the form:

$$\mathsf{P}(\mathcal{D}_t) = \left\{ \theta \in \mathsf{K} \in \mathbb{R}^{m+n} \middle| \forall k \leq t : \ \mathbf{A}_k \theta \leq \mathbf{b}_k \right\}, \tag{19}$$

where $\mathbf{A}_k = \mathbf{A}_k(x_k, \tau_k)$ and $\mathbf{b}_k = \mathbf{b}_k(x_k, \tau_k)$ are a matrix and vector of "features" constructed from $u_k = \tau_k$ and x_t via the known functional form of $\mathbf{Y}$.

Designing π and $\texttt{SEL}$. We outline in Appendix E how to design a robust oracle based on a well-established robust control method for robotic manipulators proposed in (Spong, 1992b). Since the consistent sets are convex and can be constructed online, we can implement procedures $\texttt{SEL}_p$ and $\texttt{SEL}_s$ defined in (12) as competitive algorithms for the CMC problem.

Mistake guarantee for $\mathcal{A}_\pi(\texttt{SEL}_{p/s})$ The resulting online algorithm $\mathcal{A}_\pi(\texttt{SEL}_p)$ or $\mathcal{A}_\pi(\texttt{SEL}_s)$ guarantees finiteness of the total number of mistakes $\sum_{k=0}^\infty \mathcal{G}_k^\epsilon(x_k, \tau_k)$ which implies the desired tracking behavior $\limsup_{k\to\infty} \|x - x^d\| \leq \epsilon$. Moreover, if we can provide a bound M on the mistake constant $M_\rho^\pi < M$, we obtain from Theorem 2.5, (ii) the mistake guarantee:

$$\sum_{k=0}^{\infty} \mathcal{G}_k^\epsilon(x_k, \tau_k) \leq M(\tfrac{2L}{\rho} + 1),$$

which bounds the number of times the system could have a tracking error larger than ϵ.

$\pi[\theta^*]$	0	0.4	0.99	1	1
$\mathcal{A}_\pi(\texttt{SEL})$	0	0.2	0.8	0.95	1
T	$3\,s$	$6\,s$	$12\,s$	$30\,s$	$50\,s$

Table 1: Fraction of experiments completing the swing up before time T: ideal policy $\pi[\theta^*]$ vs. $\mathcal{A}_\pi(\texttt{SEL})$

6 Empirical Validation

We illustrate the practical potential for of our approach on a challenging cart-pole swing-up goal from limited amount of interaction. Compared to the standard cart-pole domain that is commonly used in RL (Brockman et al., 2016a), we introduce modifications motivated by real-world concerns in several important ways:

1. *Goal specification*: the goal is to swing up and balance the cart-pole from a down position, which is significantly harder than balancing from the up-right position (the standard RL benchmark).
2. *Realistic dynamics*: we use a high-fidelity continuous-time nonlinear model, with noisy measurements of discrete-time state observations.
3. *Safety*: cart position has to be kept in a bounded interval for all time. In addition, acceleration should not exceed a specified maximum limit.
4. *Robustness to structured adversarial disturbance*: We evaluate on 900 uncertainty settings, each with a different θ^* reflecting mass, length, and friction. The tuning parameter remains the same for all experiments. This robustness requirement amounts to a generalization goal in contemporary RL.
5. *Other constraints*: no system reset is allowed during learning (i.e., a truly continual goal).

Our introduced modification make this goal significantly more challenging from both online learning and adaptive control perspective. Table 2 summarizes the results over 900 different parameter conditions (corresponding to 900 adversarial settings). See Appendix G for additional description of our setup and results.

We employ well-established techniques to synthesize model-based oracles. The expert controllers are a hybrid combination of a linear state-feedback LQR around the upright position, a so-called energy-based swing-up controller (See (Åström and Furuta, 2000)) and a control barrier-function to satisfy the safety constraints. As also described in (Dulac-Arnold et al., 2019), adding constraints on state and acceleration makes learning the swing-up of the cart-pole a significantly harder goal for state-of-the art learning and control algorithms.

Table 2 compares the online algorithm to the corresponding ideal oracle policy $\pi[\theta^*]$ shows that the online controller is only marginally slower.

References

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

Yasin Abbasi-Yadkori, Nevena Lazic, and Csaba Szepesvári. Regret bounds for model-free linear quadratic control. *arXiv preprint arXiv:1804.06021*, 2018.

Naman Agarwal, Brian Bullins, Elad Hazan, Sham M Kakade, and Karan Singh. Online control with adversarial disturbances. *arXiv preprint arXiv:1902.08721*, 2019a.

Naman Agarwal, Elad Hazan, and Karan Singh. Logarithmic regret for online control. In *Advances in Neural Information Processing Systems*, pages 10175–10184, 2019b.

A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, Aug 2017. ISSN 1558-2523. doi: 10.1109/TAC.2016.2638961.

Brian DO Anderson, Thomas Brinsmead, Daniel Liberzon, and A Stephen Morse. Multiple model adaptive control with safe switching. *International journal of adaptive control and signal processing*, 15(5):445–470, 2001.

CJ Argue, Sébastien Bubeck, Michael B Cohen, Anupam Gupta, and Yin Tat Lee. A nearly-linear bound for chasing nested convex bodies. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 117–122. SIAM, 2019.

CJ Argue, Anupam Gupta, Guru Guruganesh, and Ziye Tang. Chasing convex bodies with linear competitive ratio. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1519–1524. SIAM, 2020.

Karl Johan Åström and Katsuhisa Furuta. Swinging up a pendulum by energy control. *Automatica*, 36 (2):287–295, 2000.

Erin M Aylward, Pablo A Parrilo, and Jean-Jacques E Slotine. Stability and robustness analysis of nonlinear systems via contraction metrics and sos programming. *Automatica*, 44(8):2163–2170, 2008.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2242–2253. IEEE, 2017.

Franco Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

Vincent D Blondel and John N Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36(9):1249–1274, 2000.

Nicholas M Boffi, Stephen Tu, and Jean-Jacques E Slotine. Regret bounds for adaptive nonlinear control. *arXiv preprint arXiv:2011.13101*, 2020.

Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. cambridge university press, 2005.

Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

R. P. Braatz, P. M. Young, J. C. Doyle, and M. Morari. Computational complexity of /spl mu/ calculation. *IEEE Transactions on Automatic Control*, 39(5): 1000–1002, 1994. doi: 10.1109/9.284879.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016a.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016b.

Sébastien Bubeck, Bo'az Klartag, Yin Tat Lee, Yuanzhi Li, and Mark Sellke. Chasing nested convex bodies nearly optimally. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1496–1508. SIAM, 2020.

Niangjun Chen, Gautam Goel, and Adam Wierman. Smoothed online convex optimization in high dimensions via online balanced descent. In *Conference On Learning Theory (COLT)*, 2018.

Xinyi Chen and Elad Hazan. Black-box control for linear dynamical systems. *arXiv preprint arXiv:2007.06650*, 2020.

Alon Cohen, Avinatan Hasidim, Tomer Koren, Nevena Lazic, Yishay Mansour, and Kunal Talwar. Online linear quadratic control. In *International Conference on Machine Learning*, pages 1029–1038, 2018.

Martin Corless and George Leitmann. Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems. *IEEE Transactions on Automatic Control*, 26(5):1139–1144, 1981.

Munther A Dahleh, Theodore V Theodosopoulos, and John N Tsitsiklis. The sample complexity of worst-case identification of fir linear systems. *Systems & control letters*, 20(3):157–166, 1993.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, pages 1–47, 2017.

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

Priya L Donti, Melrose Roderick, Mahyar Fazlyab, and J Zico Kolter. Enforcing robust control guarantees within neural network policies. In *International Conference on Learning Representations (ICLR)*, 2021.

R. M. Dudley. Universal donsker classes and metric entropy. *The Annals of Probability*, 15(4):1306–1326, 1987. ISSN 00911798. URL `http://www.jstor.org/stable/2244004`.

Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

Claude-Nicolas Fiechter. Pac adaptive control of linear systems. In *Proceedings of the tenth annual conference on Computational learning theory*, pages 72–80, 1997.

Randy Freeman and Petar V Kokotovic. *Robust nonlinear control design: state-space and Lyapunov techniques*. Springer Science & Business Media, 2008.

Joel Friedman and Nathan Linial. On convex body chasing. *Discrete & Computational Geometry*, 9(3): 293–321, 1993.

Gautam Goel and Adam Wierman. An online algorithm for smoothed regression and lqr control. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Gautam Goel, Yiheng Lin, Haoyuan Sun, and Adam Wierman. Beyond online balanced descent: An optimal algorithm for smoothed online optimization. *Advances in Neural Information Processing Systems*, 32:1875–1885, 2019.

T. Gurriet, M. Mote, A. D. Ames, and E. Feron. An online approach to active set invariance. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 3592–3599, Dec 2018. doi: 10.1109/CDC.2018.8619139.

Elad Hazan, Sham M Kakade, and Karan Singh. The nonstochastic control problem. In *Conference on Algorithmic Learning Theory (ALT)*, 2020.

Joao P Hespanha, Daniel Liberzon, and A Stephen Morse. Overcoming the limitations of adaptive control by means of logic-based switching. *Systems & control letters*, 49(1):49–65, 2003.

Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, et al. Stable baselines. *GitHub repository*, 2018.

Dimitar Ho and John C Doyle. Robust model-free learning and control without prior knowledge. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 4577–4582. IEEE, 2019.

Petros Ioannou and Bariş Fidan. *Adaptive control tutorial*. SIAM, 2006.

Petros A Ioannou and Jing Sun. *Robust adaptive control*. Courier Corporation, 2012.

Zhong-Ping Jiang, Eduardo Sontag, and Yuan Wang. Input-to-state stability for discrete-time nonlinear systems. *IFAC Proceedings Volumes*, 32(2):2403 – 2408, 1999. 14th IFAC World Congress 1999, Beijing, Chia, 5-9 July.

Sham Kakade, Akshay Krishnamurthy, Kendall Lowrey, Motoya Ohnishi, and Wen Sun. Information theoretic regret bounds for online nonlinear control. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ, 2002.

Elias Koutsoupias and Christos H Papadimitriou. Beyond competitive analysis. *SIAM Journal on Computing*, 30(1):300–317, 2000.

Miroslav Krstic, Petar V. Kokotovic, and Ioannis Kanellakopoulos. *Nonlinear and Adaptive Control Design*. John Wiley & Sons, Inc., 1995.

Xiangbin Liu, Hongye Su, Bin Yao, and Jian Chu. Adaptive robust control of nonlinear systems with dynamic uncertainties. *International Journal of Adaptive Control and Signal Processing*, 23(4):353–377, 2009.

Lennart Ljung. System identification: theory for the user. *PTR Prentice Hall, Upper Saddle River, NJ*, 28, 1999.

Horia Mania, Michael I Jordan, and Benjamin Recht. Active learning for nonlinear system identification with guarantees. *arXiv preprint arXiv:2006.10277*, 2020.

David Q Mayne, Erric C Kerrigan, EJ Van Wyk, and P Falugi. Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, 21(11):1341–1353, 2011.

Mario Milanese, John Norton, Hélène Piet-Lahanier, and Éric Walter. *Bounding approaches to system identification*. Springer Science & Business Media, 2013.

Joseph Moore and Russ Tedrake. Adaptive control design for underactuated systems using sums-of-squares optimization. In *2014 American Control Conference*, pages 721–728. IEEE, 2014.

Richard M Murray. *A mathematical introduction to robotic manipulation*. CRC press, 2017.

Kim-Doang Nguyen and Harry Dankowicz. Adaptive control of underactuated robots with unmodeled dynamics. *Robotics and Autonomous Systems*, 64:84–99, 2015.

Romeo Ortega and Mark W. Spong. Adaptive motion control of rigid robots: A tutorial. *Automatica*, 25 (6):877 – 888, 1989. ISSN 0005-1098.

Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

Marios M Polycarpou and Petros A Ioannou. A robust adaptive nonlinear control design. In *1993 American Control Conference*, pages 1365–1369. IEEE, 1993.

Saša V Rakovic and Miroslav Baric. Parameterized robust control invariant sets for linear systems: Theoretical advances and computational remarks. *IEEE Transactions on Automatic Control*, 55(7):1599–1614, 2010.

SV Rakovic, AR Teel, DQ Mayne, and A Astolfi. Simple robust control invariant tubes for some classes of nonlinear discrete time systems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6397–6402. IEEE, 2006.

Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1976.

Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer Science & Business Media, 2013.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

Mark Sellke. Chasing convex bodies optimally. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1509–1518. SIAM, 2020.

Guanya Shi, Yiheng Lin, Soon-Jo Chung, Yisong Yue, and Adam Wierman. Online optimization with memory and competitive control. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Max Simchowitz, Horia Mania, Stephen Tu, Michael I Jordan, and Benjamin Recht. Learning without mixing: Towards a sharp analysis of linear system identification. In *Conference On Learning Theory*, pages 439–473. PMLR, 2018.

Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ, 1991.

M. Spong and M. Vidyasagar. Robust linear compensator design for nonlinear robotic control. *IEEE Journal on Robotics and Automation*, 3(4):345–351, 1987.

M. W. Spong. On the robust control of robot manipulators. *IEEE Transactions on Automatic Control*, 37(11):1782–1786, 1992a.

M. W. Spong. On the robust control of robot manipulators. *IEEE Transactions on Automatic Control*, 37(11):1782–1786, 1992b. doi: 10.1109/9.173151.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite, 2018.

Russ Tedrake. Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation. (Course Notes for MIT 6.832), 2020. Downloaded on 2020-03-30 from http://underactuated.mit.edu.

Sundarapandian Vaidyanathan, Christos Volos, et al. *Advances and applications in nonlinear control systems*. Springer, 2016.

Bin Yao and Masayoshi Tomizuka. Robust adaptive nonlinear control with guaranteed transient performance. In *Proceedings of 1995 American Control Conference-ACC'95*, volume 4, pages 2500–2504. IEEE, 1995.

Chenkai Yu, Guanya Shi, Soon-Jo Chung, Yisong Yue, and Adam Wierman. The power of predictions in online control. 2020.

Kemin Zhou and John Comstock Doyle. *Essentials of robust control*, volume 104. Prentice hall Upper Saddle River, NJ, 1998.

Kemin Zhou, John C Doyle, and Keith Glover. Robust and optimal control. 1996.

Ingvar Ziemann and Henrik Sandberg. Regret Lower Bounds for Unbiased Adaptive Control of Linear Quadratic Regulators. working paper or preprint, February 2020. URL `https://hal.archives-ouvertes.fr/hal-02404014`.

A Full version of main results

A.1 Overview of main definitions and notational conventions

Notation. 2^{S} denotes the set of all subsets of a set S. A *compact parametrization* $(\mathbb{T}, \mathsf{K}, d)$ of an uncertainty set $\mathcal{F}$ consists of the compact metric space (K, d) and mapping $\mathbb{T} : \mathsf{K} \mapsto 2^{\mathcal{F}}$ such that: $\mathcal{F} \subset \bigcup_{\theta \in \mathsf{K}} \mathbb{T}(\theta)$. An *objective* $\boldsymbol{\mathcal{G}}$ is a sequence of $\{0,1\}$-valued functions $(\mathcal{G}_0, \mathcal{G}_1, \dots)$. $\mathbb{D}$ is the space of finite data sets, $\mathcal{C}$ the space of time-varying state-feedback policies and $d_H : 2^{\mathsf{K}} \times 2^{\mathsf{K}} \mapsto \mathbb{R}^+$ is the Hausdorff metric:

$$\mathbb{D} := \{(d_1, \dots, d_N) \mid d_i \in \mathbb{N} \times \mathcal{X} \times \mathcal{X} \times \mathcal{U},\ N < \infty\},$$
$$\mathcal{C} := \{\mathbb{N} \times \mathcal{X} \mapsto \mathcal{U}\}$$
$$d_H(\mathsf{S}, \mathsf{S}') := \max\left\{ \max_{x \in \mathsf{S}} d(x, \mathsf{S}'),\ \max_{y \in \mathsf{S}'} d(y, \mathsf{S}) \right\} \quad \text{for sets } \mathsf{S}, \mathsf{S}' \subset \mathsf{K}$$

A data sequence $\boldsymbol{\mathcal{D}} = (\mathcal{D}_1, \mathcal{D}_2, \dots)$ is a data stream if the data sets $\mathcal{D}_t = (d_1, \dots, d_t)$ are formed from a sequence of observations $(d_1, d_2, \dots)$. Procedure π is a map $\mathsf{K} \mapsto \mathcal{C}$ expects a parameter $\theta \in \mathsf{K}$ as input and returns policy $\pi[\theta]$. Procedure $\texttt{SEL}$ is a map $\mathbb{D} \mapsto \mathsf{K}$ which takes as input a data set $\mathcal{D} \in \mathbb{D}$ and returns a parameter $\theta = \texttt{SEL}[\mathcal{D}]$. For a fixed choice of $\texttt{SEL}$ and π, we instantiate $\mathcal{A}_\pi(\texttt{SEL})$ as the online algorithm described in Algorithm 1. For a time-interval $\mathcal{I} = [t_1, t_2] \subset \mathbb{N}$, $(x_{\mathcal{I}},\ u_{\mathcal{I}})$ represent the tuples $(x_{t_1}, \dots, x_{t_2})$, $(u_{t_1}, \dots, u_{t_2})$. For a fixed objective $\boldsymbol{\mathcal{G}}$, the set of admissable states at time t are $\mathsf{X}_t = \{x \mid \exists u' :\ \mathcal{G}_t(x, u') = 0\}$.

Definition A.1. For a time-interval $\mathcal{I} = [t_1, t_2] \subset \mathbb{N}$ and fixed $\theta^* \in \mathsf{K}$, the set $\mathcal{S}_{\mathcal{I}}[\rho; \theta^*]$ is defined as

$$\mathcal{S}_{\mathcal{I}}[\rho; \theta^*] := \left\{ (x_{\mathcal{I}}, u_{\mathcal{I}}) \;\middle|\; \begin{array}{c} \exists f \in \mathbb{T}(\theta^*) \exists (\theta_{t_1}, \dots, \theta_{t_2}) \subset \mathsf{B}_\rho(\theta^*) : \forall k \in \mathcal{I} : \\ u_k = \pi[\theta_k](k, x_k),\ \text{and } x_{k+1} = f(k, x_k, u_k),\ \text{if } k \neq t_2 \end{array} \right\}$$

Definition. A function $f : X \mapsto Y$ is a selection of the set-valued map $F : X \mapsto 2^Y$, if $\forall x \in X :\ f(x) \in F(x)$.

A.1.1 Robust oracles

ρ-robust	$\forall \theta \in \mathsf{K} :\ \sup_{\gamma \geq 0} m^\pi_\rho(\gamma; \theta) < \infty$
uniformly ρ-robust	$M^\pi_\rho := \sup_{\gamma \geq 0, \theta \in \mathsf{K}} m^\pi_\rho(\gamma; \theta) < \infty$
locally ρ-robust	$\forall \gamma \geq 0,\ \theta \in \mathsf{K} :\ m^\pi_\rho(\gamma; \theta) < \infty$
locally uniformly ρ-robust	$\forall \gamma \geq 0 :\ M^\pi_\rho(\gamma) := \sup_{\theta \in \mathsf{K}} m^\pi_\rho(\gamma; \theta) < \infty$

Table 2: Different notions of robustness of an oracle π

Definition A.2. Equip $\mathcal{X}$ with some norm $\|\cdot\|$ and consider a fix parameterization $(\mathbb{T}, \mathsf{K}, d)$ for $\mathcal{F}$. Define the quantity $m^\pi_\rho(\gamma; \theta)$ for each $\gamma \geq 0$, $\rho \geq 0$ and $\theta \in \mathsf{K}$ as

$$m^\pi_\rho(\gamma; \theta) := \sup_{\mathcal{I} = [t, t'] :\, t < t'} \ \sup_{(x_{\mathcal{I}}, u_{\mathcal{I}}) \in \mathcal{S}_{\mathcal{I}}[\rho; \theta], \|x_t\| \leq \gamma} \sum_{t \in \mathcal{I}} \mathcal{G}_t(x_t, u_t)$$

If $m^\pi_0(\gamma; \theta)$ is finite everywhere, we call π an *oracle* for $\boldsymbol{\mathcal{G}}$ and call it (locally) (uniformally) ρ-robust, if the corresponding property presented in Table 2 is satisfied. If it exists, M^π_ρ will be called the *mistake constant* or *mistake function* (for local properties) of π.

Corollary (Def.A.2)**.** *Let procedure π be a uniformly ρ-robust oracle for $\boldsymbol{\mathcal{G}}$ in $(\mathbb{T}, \mathsf{K}, d)$. Let $\boldsymbol{x}, \boldsymbol{u}$ be some trajectory and denote its restriction to some time-interval $\mathcal{I}$ as $(x_{\mathcal{I}}, u_{\mathcal{I}})$. If $(x_{\mathcal{I}}, u_{\mathcal{I}}) \in \bigcup_{\theta \in \mathsf{K}} \mathcal{S}_{\mathcal{I}}[\rho; \theta]$, then $\sum_{t \in \mathcal{I}} \mathcal{G}_t(x_t, u_t) \leq M^\pi_\rho$. If instead, π is locally uniformly ρ-robust, the previous inequality changes to $\sum_{t \in \mathcal{I}} \mathcal{G}_t(x_t, u_t) \leq M^\pi_\rho(\|x_{t_0}\|)$, $t_0 := \min(\mathcal{I})$.*

Definition A.3. For an objective $\boldsymbol{\mathcal{G}}$, we call a ρ-robust oracle π *cost invariant*, if for all $\theta \in \mathsf{K}$ and $t \geq 0$ the following holds

- For all $x \in \mathsf{X}_t$ holds $\mathcal{G}_t(x, \pi[\theta](t, x)) = 0$.
- For all $x \in \mathsf{X}_t$, $f \in \mathbb{T}[\theta]$ and θ' s.t. $d(\theta', \theta) \leq \rho$, holds $f(t, x, \pi[\theta'](t, x)) \in \mathsf{X}_{t+1}$,

Remark 3. The above condition is related to the well-known notion of *positive set/tube invariance* in control theory (Blanchini, 1999): The above condition requires that the oracle policies $\pi[\theta]$ can ensure for their nominal model $\mathbb{T}[\theta]$ the following closed loop condition: $x_t \in \mathsf{X}_t \implies x_{t+1} \in \mathsf{X}_{t+1}$, $\forall t$.

Corollary (Def. A.3)**.** *Assume that π is cost invariant as in Def.A.3 and that π is uniformly ρ-robust. Let $\boldsymbol{x}, \boldsymbol{u}$ be some trajectory and denote its restriction to some time-interval $\mathcal{I} = [t_1, t_2]$ as $(x_{\mathcal{I}}, u_{\mathcal{I}})$. If $(x_{\mathcal{I}}, u_{\mathcal{I}}) \in \bigcup_{\theta \in \mathsf{K}} \mathcal{S}_{\mathcal{I}}[\rho; \theta]$, then the following holds:*

(i) If $s \in \mathcal{I}$ s.t. $\mathcal{G}_s(x_s, u_s) = 0$ and $s < t_2$, then $\mathcal{G}_{s+1}(x_{s+1}, u_{s+1}) = 0$.
(ii) For all s in the range $t_1 + M_\rho^\pi \le s \le t_2$ holds $\mathcal{G}_s(x_s, u_s) = 0$.
(iii) If $\mathcal{G}_{t_1}(x_{t_1}, u_{t_1}) = 0$, then $\sum_{t\in\mathcal{I}} \mathcal{G}_t(x_t, u_t) = 0$
(iv) If $\mathcal{G}_{t_2}(x_{t_2}, u_{t_2}) = 1$, then $|\mathcal{I}| \le M_\rho^\pi$.

Proof. (i): $\mathcal{G}_s(x_s, u_s) = 0$, implies $x_s \in \mathsf{X}_s$. The condition $(x_\mathcal{I}, u_\mathcal{I}) \in \mathcal{S}_\mathcal{I}[\rho; \theta^*]$ means that $x_{s+1} = f(s, x_s, \kappa(x_s))$, for some $f \in \mathbb{T}[\theta^*]$ and some policy $\pi[\theta_s]$, with $d(\theta_s, \theta^*) \le \rho$. Due to the cost-invariant property, we conclude $x_{s+1} \in \mathsf{X}_{s+1}$ and since we pick $u_{s+1} = \pi[\theta_s](s+1, x_{s+1})$ we are also guaranteed that $\mathcal{G}_{s+1}(x_{s+1}, u_{s+1}) = 0$.
(ii): If there exists an s s.t. $t_1 + M_\rho^\pi \le s \le t_2$ then the interval size is bounded below as $|\mathcal{I}| > M_\rho^\pi$. The claim follows by using property (i) and that $\sum_{t\in\mathcal{I}} \mathcal{G}_t(x_t, u_t) \le M_\rho^\pi$.
(iii): Follows from (i). (iv): by contraposition of (ii) follows $t_2 < t_1 + M_\rho^\pi$ and $|\mathcal{I}| = t_2 - t_1 + 1 \le M_\rho^\pi$. □

Definition A.4. $\pi : \mathsf{K} \mapsto \mathcal{C}$ is (α, β)-single step robust in the space $(\mathcal{X}, \|\cdot\|)$ if for any 2-time-steps nominal trajectory $(x_{t+1}, x_t), (u_{t+1}, u_t) \in \mathcal{S}_{[t,t+1]}[\rho; \theta]$ holds $\|x_{t+1}\| \le \alpha\rho\|x_t\| + \beta$.

The above property requires that in the idealized setting (7), we can uniformly bound the single-step growth of the state by a scalar linear function in the noise-level ρ and the previous state norm. Equivalently, we can explicitly write out condition in Definition A.4 as:

$$\exists \alpha, \beta > 0 : \quad \forall \theta, \theta' \in \mathsf{K}, x \in \mathcal{X}, f \in \mathbb{T}[\theta], t \ge 0 : \ \|f(t, x, \pi[\theta'](t, x))\| \le \alpha d(\theta, \theta')\|x\| + \beta.$$

A.1.2 Consistent sets

Definition A.5. Given a compact parameterization $(\mathbb{T}, \mathsf{K}, d)$ of the uncertainty set $\mathcal{F}$ and a data set $\mathcal{D} = (d_1, \dots, d_N)$, $d_k = (t_k, x_k^+, x_k, u_k)$ define the consistent set map $\mathsf{P} : \mathbb{D} \mapsto 2^\mathsf{K}$ as

$$\mathsf{P}(\mathcal{D}) := \text{closure}\Big(\bigcap_{(t,x^+,x,u)\in\mathcal{D}} \big\{ \theta \in \mathsf{K} \ \big| \ \exists f \in \mathbb{T}(\theta) \text{ s.t. } : \ \ x^+ = f(t, x, u) \ \big\} \Big). \tag{20}$$

Corollary (Def. A.5)**.** *Assume $\boldsymbol{\mathcal{D}}$ is a data stream with at least one consistent $f \in \mathcal{F}$ Then, the following holds for the sequence of consistent sets $\mathsf{P}(\boldsymbol{\mathcal{D}}) = (\mathsf{P}(\mathcal{D}_1), \mathsf{P}(\mathcal{D}_2), \dots)$:*

(i) The sequence of consistent sets is nested in K, i.e: $\mathsf{P}(\mathcal{D}_t) \supset \mathsf{P}(\mathcal{D}_{t+1})$
(ii) $\mathsf{P}(\mathcal{D}_\infty) := (\bigcap_{k=1}^\infty \mathsf{P}(\mathcal{D}_k)) \cap \mathsf{K}$ is non-empty
(iii) If $\theta_t \in \mathsf{P}(\mathcal{D}_t)$, and $\lim_{t\to\infty} \theta_t = \theta_\infty$, then $\theta_\infty \in \mathsf{P}(\mathcal{D}_\infty)$.

Proof. The first property is clear since $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{d_t\}$. For the second, notice that $\mathsf{P}(\mathcal{D}_t)$ is always a non-empty compact set and recall the basic real analysis fact (Rudin et al., 1976): The intersection of a nested sequence of non-empty compact sets is always non-empty (Rudin et al., 1976). To verify property (iii), notice that nestedness implies that the sub-sequence $(\theta_T, \theta_{T+1}, \dots)$ is contained in $\mathsf{P}(\mathcal{D}_T)$; Since $\mathsf{P}(\mathcal{D}_T)$ is closed, we have that $\theta_\infty \in \mathsf{P}(\mathcal{D}_T)$. We chose T arbitrary, so we conclude that $\theta_\infty \in \mathsf{P}(\mathcal{D}_T)$ for all $T \ge 0$, i.e.: $\theta_\infty \in \mathsf{P}(\mathcal{D}_\infty)$. □

A.1.3 Consistent model chasing conditions

Definition A.6. Let $\texttt{SEL} : \mathbb{D} \mapsto \mathsf{K}$ be a selection of P. Let $\boldsymbol{\mathcal{D}} = (\mathcal{D}_1, \mathcal{D}_2, \dots)$ be an online data stream and let $\boldsymbol{\theta}$ be a sequence defined for each time t as $\theta_t = \texttt{SEL}[\mathcal{D}_t]$. Assume that there always exists an $f \in \mathcal{F}$ consistent with $\boldsymbol{\mathcal{D}}$ and consider the following statements:

(A) $\theta^* = \lim_{t\to\infty} \theta_t$ exists.
(B) $\lim_{t\to\infty} d(\theta_t, \theta_{t-1}) = 0$.
(C) γ-*competitive*: $\sum_{t=t_1+1}^{t_2} d(\theta_t, \theta_{t-1}) \le \gamma\, d_H(\mathsf{P}(\mathcal{D}_{t_2}), \mathsf{P}(\mathcal{D}_{t_1}))$ holds for any $t_1 < t_2$.
(D) (γ, T)-*weakly competitive*: $\sum_{t=t_1+1}^{t_2} d(\theta_t, \theta_{t-1}) \le \gamma\, d_H(\mathsf{P}(\mathcal{D}_{t_2}), \mathsf{P}(\mathcal{D}_{t_1}))$ holds for any $t_2 - t_1 \le T$.

We will say that SEL is a *consistent model chasing* (CMC) algorithm, if one of the above *chasing* properties can be guaranteed for any pair of $\boldsymbol{\mathcal{D}}$ and corresponding $\boldsymbol{\theta}$.

Corollary A.7. *Then following implications hold between the properties of Definition 2.4:*

$$\begin{array}{ccc} \text{(C)} & \Rightarrow & \text{(A)} \\ \Downarrow & & \Downarrow \\ \text{(D)} & \Rightarrow & \text{(B)} \end{array}$$

The reverse (and any other) implications between the properties do not hold in general.

A.2 Main theorems

Assuming that π and SEL meet the required specifications, we can provide the overall guarantees for the algorithm. Let $(\mathbb{T}, \mathsf{K}, d)$ be a compact parametrization of a given uncertainty set $\mathcal{F}$. Let π be robust per Definition 2.2 and SEL return consistent parameters per Definition 2.4. We apply the online control strategy $\mathcal{A}_\pi(\texttt{SEL})$ described in Algorithm 1 to system $x_{t+1} = f^*(t, x_t, u_t)$ with unknown dynamics $f^* \in \mathcal{F}$ and denote $(\boldsymbol{x}, \boldsymbol{u})$ as the corresponding state and input trajectories.

A.2.1 Finite mistake guarantees

Theorem A.8. *Assume that SEL is CMC-algorithm with chasing property (A) and that π is an oracle for an objective $\boldsymbol{\mathcal{G}}$. Then the following mistake guarantees hold:*

(i) If π is robust then $(\boldsymbol{x}, \boldsymbol{u})$ always satisfies: $\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) < \infty$.

(ii) If π is uniformly ρ-robust and SEL is γ-competitive, then $(\boldsymbol{x}, \boldsymbol{u})$ obey the inequality:

$$\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) \leq M_\rho^\pi \left(\tfrac{2\gamma}{\rho} \mathrm{diam}(\mathsf{K}) + 1 \right).$$

The strong competitiveness property is not necessary if instead stronger conditions on the oracle can be enforced: Theorem A.9 states that if the oracle π is cost-invariant we can weaken the assumptions on SEL and still provide finite mistake guarantees.

Theorem A.9. *Assume that SEL is CMC-algorithm with chasing property (B) and that π is an uniformly ρ-robust, cost-invariant oracle for an objective $\boldsymbol{\mathcal{G}}$. Then the following mistake guarantees holds:*

(i) $(\boldsymbol{x}, \boldsymbol{u})$ always satisfies: $\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) < \infty$.

(ii) If SEL is (γ, T)-weakly competitive, then $(\boldsymbol{x}, \boldsymbol{u})$ obey the following inequality, with N denoting the packing number of K (see definition below):

$$\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) \leq M_\rho^\pi \left(N(\mathsf{K}, r^*) + 1 \right), \quad r^* := \frac{1}{2} \frac{\rho}{\gamma} \frac{T}{M_\rho^\pi + T}$$

Definition A.10 ((Dudley, 1987))**.** Let $(\mathcal{M}, d)$ be a metric space and $\mathsf{S} \subset \mathcal{M}$ a compact set. For $r > 0$, define $N(\mathsf{S}, r)$ as the *r-packing number* of S:

$$N(\mathsf{S}, r) := \max \left\{ n \in \mathbb{N} \,\middle|\, \exists \theta_1, \ldots, \theta_N \in \mathsf{S} \text{ s.t. } d(\theta_i, \theta_j) > r \text{ for all } 1 \leq i, j \leq n,\ i \neq j \right\}.$$

We discuss in section 3 how robust oracle design is a pure robust control problem and briefly overview the main available methods. Designing procedures SEL with the properties stated in Definition 2.4 poses a new class of online learning problems. We propose in Section 4 a reduction of SEL to the well-known nested convex body chasing problem, which enables design and analysis of competitive SEL procedures.

Theorem 2.5 and A.9 can be invoked on any learning and control method that instantiates $\mathcal{A}_\pi(\texttt{SEL})$. It offers a set of sufficient conditions to verify whether a learning agent $\mathcal{A}_\pi(\texttt{SEL})$ can provide mistake guarantees: We need to show that w.r.t. some compact parametrization $(\mathbb{T}, \mathsf{K}, d)$ of the uncertainty set $\mathcal{F}$, π operates as a robust oracle for some objective $\boldsymbol{\mathcal{G}}$, and that SEL satisfies strong enough chasing properties.

Theorem 2.5 also suggests a design philosophy of decoupling the learning and control problem into two separate problems while retaining the appropriate guarantees: (1) design a robust oracle π for a specified control goal $\boldsymbol{\mathcal{G}}$; and (2) design an online selection procedure SEL that satisfies the chasing properties defined in Definition 2.4.

A.2.2 A worst-case bound on the state norm

Regardless of the objective $\boldsymbol{\mathcal{G}}$, we can provide worst-case state norm guarantees for $\mathcal{A}_\pi(\texttt{SEL})$ in the normed space $(\mathcal{X}, \|\cdot\|)$, if SEL is a competitive or a weakly competitive CMC algorithm and π provides sufficient robustness guarantees for a single-time step transition:

Theorem A.11. *Assume that $\pi : \mathsf{K} \mapsto \mathcal{C}$ is (α, β)-single step robust in the space $(\mathcal{X}, \|\cdot\|)$. Then, the following state bound guarantees hold:*

(i) If `SEL` *is a γ-competitive CMC algorithm, then:*

$$\forall t: \quad \|x_t\| \leq e^{\alpha\gamma\phi(\mathsf{K})}\left(e^{-t}\|x_0\| + \beta\frac{e}{e-1}\right).$$

(i) If `SEL` *is a (γ, T)-weakly competitive CMC algorithm, then:*

$$\|\boldsymbol{x}\|_\infty \leq \inf_{0<\mu<1}\left(1+(\alpha\phi(K))^{n^*}\right)\max\{\tfrac{\beta}{1-\mu}, \|x_0\|\} + \beta\sum_{k=0}^{n^*}(\alpha\phi(K))^k$$

where $n^* = N(\mathsf{K}, \frac{\mu}{\alpha\gamma})$ and $\phi(\mathsf{K})$ denotes the diameter of K.

A.3 Mistake guarantees with locally robust oracles

Theorem A.12 (Corollary of Thm. 2.5)**.** *Consider the setting and assumptions of Thm.2.5 and Thm.A.9, but relax the oracle robustness requirements to corresponding local versions and enforce the additional oracle assumption stated in Theorem A.11.*

Then all guarantees of Theorem 2.5 (i),(ii) and Theorem A.9 (i),(ii) still hold, if we replace M_ρ^π in Theorem (ii) and Theorem (ii) respectively by $M_\rho^\pi(\gamma_\infty)$ and $M_\rho^\pi(\gamma_\infty^w)$ with the constants:

$$\gamma_\infty = e^{\alpha\gamma\phi(\mathsf{K})}\left(\|x_0\| + \beta\frac{e}{e-1}\right)$$

$$\gamma_\infty^w = \inf_{0<\mu<1}\left(1+(\alpha\phi(K))^{n^*}\right)\max\{\tfrac{\beta}{1-\mu}, \|x_0\|\} + \beta\sum_{k=0}^{n^*}(\alpha\phi(K))^k$$

where $n^* = N(\mathsf{K}, \frac{\mu}{\alpha\gamma})$ and $\phi(\mathsf{K})$ denotes the diameter of K.

B Additional discussion

B.1 The dual purpose of the robust oracle assumption

A principled way to leverage non-adaptive control methods in adaptive settings. The range of existing control methods one has available for a given control problem becomes quickly limiting, if a system model with small uncertainty can not be obtained. For example, a majority of control algorithms which are designed for small uncertainty problem settings, can not be easily extended to apply to settings with large uncertainty in the system dynamics; one is forced to either perform system identification to lower the model uncertainty or search for a different algorithm (such as more specialized adaptive control methods).

The oracle abstraction π provides a remedy for this, by allowing to leverage domain knowledge and application-specific expertise of non-adaptive control literature and apply it in problem settings with large model uncertainty, where online adaptation is a necessity. Any model-based design procedure π which works well for the small uncertainty setting (i.e.: acts as a robust oracle) can be augmented with an online chasing algorithm `SEL` (with required chasing properties) to provide robust control performance (in the form of mistake guarantees) in the large uncertainty setting via the augmented algorithm $\mathcal{A}_\pi(\texttt{SEL})$.

Notice that although our provided learning guarantees do depend on the robustness-margin constant ρ, finite mistakes are guaranteed regardless of how small the margins ρ is; we merely require ρ to be non-zero. This is an important distinction, especially if we approach the above robust control problem computationally, where optimizing for larger robustness margins δ quickly leads to computationally hard or even infeasible problems. On the other hand, designing controllers which have *some* non-zero robust margin δ is in most cases not a hard problem (except for ill-posed dynamical systems and control problems, where necessary fundamental system properties like controllability/stabilizability detectability are not satisfied).

B.2 Available methods for robust oracle design.

There are many existing control methods applicable for robust oracle design. Which method to use, depends on the control objective $\boldsymbol{\mathcal{G}}$, the particular application setting and the system class (linear/nonlinear/hybrid, etc.). For a broad survey, see (Zhou et al., 1996), (Freeman and Kokotovic, 2008), (Spong and Vidyasagar, 1987), (Spong, 1992a) and references therein. However, we can characterize two general methodologies (which can also be combined):

- *Robust stability analysis focus:* In an initial step, we use analytical design principles from robust nonlinear and linear control design to propose an oracle $\pi[\theta](x)$ in closed-form for all θ and x. This initial design step usually can readily guarantee mistake guarantees for the objective $\mathcal{G}$ if we assume no perturbations $(\boldsymbol{w}, \boldsymbol{v})$. In a second step, we use robust stability analysis (for example *Input-to-State Stability* (Jiang et al., 1999), ℓ_∞-small-gain type analysis, robust set invariance methods (Rakovic et al., 2006; Rakovic and Baric, 2010)) to verify the desired robust stability guarantee, i.e. whether the controller achieves the control goal in presence of the specified perturbations.
- *Robust control synthesis:* If the problem permits, we can also directly address the control design problem from a computational point of view, by formulating the design problem as an optimization problem and compute for a control law with desired guarantees directly. This can happen partially online, partially offline. Some common nonlinear approaches are robust (tube-based) MPC (Mayne et al., 2011; Borrelli et al., 2017), SOS-methods (Parrilo, 2000),(Aylward et al., 2008), Hamilton-Jacobi reachability methods (Bansal et al., 2017).

There are different advantages and disadvantages to both approaches: Pursuing the analysis based approach often requires less computational power (especially analytical methods) in comparison to synthesis-based approaches; on the other hand, with a "design-then-analyze" approach it might be hard to obtain a desired robustness margin ρ, whereas robust control synthesis often allows to explicitly design for a specified desired robustness margin ρ.

Remark 4. It is important to point out, that robust control problems are not always tractably solvable. See (Blondel and Tsitsiklis, 2000; Braatz et al., 1994) for simple examples of robust control problems which are NP-hard. The computational complexity of problems of robust controller synthesis tends to increase (or even be potentially infeasible) with the complexity of the system of interest; it also further increases, the more we try optimize for larger robustness margins ρ.

C Proof of Theorem 2.5 and A.9

Setup. *Let $\mathcal{T} = (\mathbb{T}, \mathsf{K}, d)$ be a compact parametrization of the uncertainty set $\mathcal{F}$ corresponding to the partially unknown system $x_{t+1} = f^*(t, x_t, u_t), f^* \in \mathcal{F}$. Given two procedures of the type π and* `SEL`*, we apply the online control strategy $\mathcal{A}_\pi(\texttt{SEL})$ described in Meta-Algorithm 1 to the system. The corresponding online state and input trajectories, denoted $(\boldsymbol{x}, \boldsymbol{u})$, follow the equations:*

$$x_{t+1} = f^*(t, x_t, u_t) \tag{21a}$$

$$u_t = \pi[\theta_t](t, x_t), \tag{21b}$$

$$\theta_t = \texttt{SEL}[d_t, \dots, d_1], \text{ where } d_t = (t, x_t, x_{t-1}, u_{t-1}) \tag{21c}$$

Assume we are given an objective $\boldsymbol{\mathcal{G}}$. The next results state conditions on π and `SEL`*, which are sufficient for bounding a-priori the corresponding worst-case online cost $\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t)$. The condition on π depends on $\boldsymbol{\mathcal{G}}$ and $\mathcal{T}$, while the condition on* `SEL` *depends only on $\mathcal{T}$.*

Theorem C.1. *Assume that* `SEL` *chases consistent models and that π is an oracle for an objective $\boldsymbol{\mathcal{G}}$. Then the following mistake guarantees hold:*

(i) If π is robust then $(\boldsymbol{x}, \boldsymbol{u})$ always satisfy: $\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) < \infty$.

(ii) If π is uniformly ρ-robust and `SEL` *is γ-competitive, then $(\boldsymbol{x}, \boldsymbol{u})$ obey the inequality:*

$$\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) \le M_\rho^\pi \Big(\tfrac{2\gamma}{\rho} \mathrm{diam}(\mathsf{K}) + 1 \Big).$$

Theorem C.2. *Assume that* `SEL` *chases consistent models weakly and that π is an uniformly ρ-robust, cost-invariant oracle for an objective $\boldsymbol{\mathcal{G}}$. Then the following mistake guarantees hold:*

(i) $(\boldsymbol{x}, \boldsymbol{u})$ always satisfy: $\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) < \infty$.

(ii) If `SEL` *is (γ, T)-weakly competitive, then $(\boldsymbol{x}, \boldsymbol{u})$ obey the inequality:*

$$\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) \le M_\rho^\pi \left(N(\mathsf{K}, r^*) + 1 \right), \quad r^* := \frac{1}{2} \frac{\rho}{\gamma} \frac{T}{M_\rho^\pi + T}$$

where N is defined in Definition A.10 as the r^*-packing number of K.

C.1 Part (i) - Theorem C.1 and C.2

Theorem. *Assume that* SEL *chases consistent models and that π is an oracle for $\boldsymbol{\mathcal{G}}$. Then $(\boldsymbol{x}, \boldsymbol{u})$ always satisfy: $\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) < \infty$.*

Proof. Denote the online data at time t as the tuple $\mathcal{D}_t := (d_1, \ldots, d_t)$. Recall Corollary A.5(ii) to verify that $\mathsf{P}(\mathcal{D}_\infty)$ is non-empty. Since $\theta_\infty \in \mathsf{P}_\infty$, there exists some $f' \in \mathbb{T}(\theta_\infty)$ such that the trajectories satisfy for all time $t \geq 0$ the dynamics $x_{t+1} = f'(t, x_t, u_t)$. Since $\theta_t \to \theta_\infty$, there exists a time T, such that for all $t \geq T$, $d(\theta_t, \theta_\infty) < \rho$, i.e.: for $t \geq T$, we apply policies $\pi(t; \theta_t)$ with parameters ρ-close to θ_∞. Per definition, this tells us that for the time interval $\mathcal{I} = [T, \infty)$, the tail of the trajectory $x_\mathcal{I}, u_\mathcal{I}$ is contained in $\mathcal{S}_\mathcal{I}[\rho; \theta_\infty]$. Since we assume that π is a ρ-robust oracle for $\boldsymbol{\mathcal{G}}$, (Definition A.2), we can conclude $\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) \leq \sum_{t=T}^{\infty} \mathcal{G}_t(x_t, u_t) \leq M$ for some finite number M. □

Theorem. *Assume that* SEL *chases consistent models weakly and that π is an uniformly ρ-robust, cost-invariant oracle for $\boldsymbol{\mathcal{G}}$. Then, $(\boldsymbol{x}, \boldsymbol{u})$ always satisfy: $\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) < \infty$.*

Proof. Pick an arbitrary $0 < \varepsilon < \rho$ and some $T \geq M_\rho^\pi$. There exists $N > 0$ such that $\forall t \geq N$: $\operatorname{dist}(\mathsf{P}(\mathcal{D}_t), \theta_t) \leq \varepsilon/4$ and $d(\theta_t, \theta_{t-1}) \leq \varepsilon/(2T)$. Pick an arbitrary time-step $s > N+T$, there exists a $\theta'_s \in \mathsf{P}(\mathcal{D}_s)$ such that $d(\theta'_s, \theta_s) \leq \varepsilon/4$. Consider now the timesteps t in the time-window $\mathcal{I}_s = [s - T, s - 1]$. Per assumption and triangle inequality, we have for all $t \in \mathcal{I}_s$:

$$d(\theta'_s, \theta_t) \leq d(\theta'_s, \theta_s) + \sum_{j=t}^{s-1} d(\theta_{j+1}, \theta_j) \leq \frac{\varepsilon}{4} + (s-t)\frac{\varepsilon}{2T} \leq \frac{\varepsilon}{4} + T\frac{\varepsilon}{2T} \leq \varepsilon < \rho.$$

Since $\theta'_s \in \mathsf{P}(\mathcal{D}_\infty)$, the truncation $(\boldsymbol{x}_{\mathcal{I}_s}, \boldsymbol{u}_{\mathcal{I}_s})$ is contained in the set $\mathcal{S}_{\mathcal{I}_s}[\rho; \theta'_s]$. Since we picked T to be larger than the mistake constant M_ρ^π, we can use Corollary A.3 to conclude that the cost at time s has to be zero, i.e.: $\mathcal{G}_s(x_s, u_s) = 0$. Finally, since s was chosen arbitrary in the interval $[N + T + 1, \infty)$, we know that $\sum_{t=N+T+1}^{\infty} \mathcal{G}_t(x_t, u_t) = 0$. The total cost is therefore $\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) = \sum_{t=0}^{N+T} \mathcal{G}_t(x_t, u_t)$ and is finite.

□

C.2 Part (ii) - Theorem C.1 and C.2

Theorem. *Assume that procedure* SEL *chases consistent models in $\mathcal{T}$ and is γ-competitive and that procedure π is a uniformly ρ-robust, cost-invariant oracle for $\boldsymbol{\mathcal{G}}$. Then, the total number of mistakes is guaranteed to be bounded above by:*

$$\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) \leq M_\rho^\pi(2\tfrac{\gamma}{\rho} d_H(\mathsf{K}, \mathsf{P}(\mathcal{D}_\infty)) + 1) \leq M_\rho^\pi(2\tfrac{\gamma}{\rho}\operatorname{diam}(\mathsf{K}) + 1)$$

Proof. The parameter sequence $\boldsymbol{\theta}$ provided by SEL satisfies $\theta_t \in \mathsf{P}(\mathcal{D}_t)$, $\forall t$ and $\sum_{t=1}^{T} d(\theta_t, \theta_{t-1}) \leq \gamma d_H(\mathsf{K}, \mathsf{P}(\mathcal{D}_T))$. Set $t_0 = 0$ and construct the index-sequence $t_0, t_1, t_2, \ldots, t_N$ as follows:

$$t_k := \begin{cases} \min\left\{t \leq T \;\middle|\; t > t_{k-1} \text{ and } d(\theta_t, \theta_{t_{k-1}}) > \frac{1}{2}\rho\right\} & \text{, if } k > 1 \\ 0 & \text{, if } k = 0 \end{cases} \tag{22}$$

until for some N, the condition $t \leq T, t > t_N, d(\theta_t, \theta_{t_N}) > \frac{1}{2}\rho$ becomes infeasible and we terminate the construction. Define the intervals $\mathcal{I}_k := [t_k, \bar{t}_k]$, where $\bar{t}_k := t_{k+1} - 1$ for $k < N$ and $\bar{t}_N = T$. The intervals $\mathcal{I}_0, \ldots, \mathcal{I}_N$ are a non-overlapping cover of the time-interval $[0, T]$:

$$\bigcup_{0 \leq k \leq N} \mathcal{I}_k = [0, T], \quad \mathcal{I}_k \cap \mathcal{I}_{k-1} = \emptyset, \forall k : 1 \leq k \leq N.$$

Let $(a_0, a_1, \ldots, a_N)$ and $(b_0, b_1, \ldots, b_N)$ be the parameters selected at the start and end of each interval $\mathcal{I}_k$, respectively: $a_k := \theta_{t_k}$ and $b_k := \theta_{\bar{t}_k}$. Per construction, we know that

$$d(a_k, \theta_t) \leq \tfrac{1}{2}\rho \text{ for all } t \in \mathcal{I}_k \tag{23}$$

$$d(a_k, a_{k-1}) > \tfrac{1}{2}\rho \text{ for all } 1 \leq k \leq N. \tag{24}$$

Inequality (23) states that $d(a_k, b_k) \leq \frac{1}{2}\rho$ and implies via triangle inequality that for all $t \in \mathcal{I}_k$ holds

$$d(\theta_t, b_k) \leq d(\theta_t, a_k) + d(a_k, b_k) \leq \rho.$$

Since we picked $b_k = \theta_{\bar{t}_k}$ and the procedure SEL assures $\theta_{\bar{t}_k} \in \mathsf{P}(\mathcal{D}_{\bar{t}_k})$, it means that for some $f' \in \mathbb{T}[b_k]$, the partial trajectory $(x_{\mathcal{I}_k}, u_{\mathcal{I}_k})$ satisfies the following equations for the time-steps $t \in \mathcal{I}_k$:

$$x_{t+1} = f'(t, x_t, u_t), \quad u_t = \pi[\theta_t](t, x_t). \tag{25}$$

We can therefore conclude that $(x_{\mathcal{I}_k}, u_{\mathcal{I}_k}) \in \mathcal{S}_{\mathcal{I}_k}[\rho; b_k]$. Now, Corollary A.3 applies and we conclude that $\sum_{t\in\mathcal{I}_k} \mathcal{G}_t(x_t, u_t) \le M_\rho^\pi$. Therefore we can bound the total mistakes as:

$$\sum_{t=0}^{T} \mathcal{G}_t(x_t, u_t) = \sum_{k=0}^{N} \sum_{j\in\mathcal{I}_k} \mathcal{G}_j(x_j, u_j) \le M_\rho^\pi (N+1). \tag{26}$$

We can now bound N using the γ-competitiveness chasing property. Recalling (24), we obtain the chain of inequalities

$$\tfrac{1}{2}\rho N \le \sum_{k=1}^{N} d(a_k, a_{k-1}) \le \sum_{k=0}^{N-1} \sum_{t\in\mathcal{I}_k} d(\theta_t, \theta_{t+1}) \le \sum_{t=1}^{T} d(\theta_t, \theta_{t-1}) \le \gamma d_H(\mathsf{K}, \mathsf{P}(\mathcal{D}_T)).$$

which lead to the bound $N \le 2\frac{\gamma}{\rho} d_H(\mathsf{K}, \mathsf{P}(\mathcal{D}_T))$. We substitute this into (26) to obtain the desired bound on the total number of mistakes:

$$\begin{aligned}\sum_{t=0}^{T} \mathcal{G}_t(x_t, u_t) &\le M_\rho^\pi (2\tfrac{\gamma}{\rho} d_H(\mathsf{K}, \mathsf{P}(\mathcal{D}_T)) + 1) \\ &\le M_\rho^\pi (2\tfrac{\gamma}{\rho} d_H(\mathsf{K}, \mathsf{P}(\mathcal{D}_\infty)) + 1) \le M_\rho^\pi (2\tfrac{\gamma}{\rho} \mathrm{diam}(\mathsf{K}) + 1)\end{aligned} \tag{27}$$

We can take the limit $T \to \infty$ and arrive at the desired result. □

Theorem. *Assume procedure SEL is (γ, T)-weakly competitive for some $\gamma > 0$, $T \ge 1$ and that procedure π is a uniform ρ-robust, cost-invariant oracle for $\boldsymbol{\mathcal{G}}$. Then, the total number of mistakes is guaranteed to be bounded above by:*

$$\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) \le M_\rho^\pi \left(N(\mathsf{K}, r^*) + 1\right), \quad \textit{with } r^* := \frac{1}{2}\frac{\rho}{\gamma}\frac{T}{M_\rho^\pi + T}$$

Proof. Denote $\boldsymbol{x}, \boldsymbol{u}$ to be some fixed online trajectories and denote $\boldsymbol{\theta}$ as the corresponding parameter sequence selected by procedure SEL. The sequence $\boldsymbol{\theta}$ satisfies $\theta_t \in \mathsf{P}(\mathcal{D}_t)$ and $\sum_{t=t_1+1}^{t_2} d(\theta_t, \theta_{t-1}) \le \gamma d(\mathsf{P}(\mathcal{D}_{t_2}), \mathsf{P}(\mathcal{D}_{t_1}))$ for all $t_2 - t_1 \le T$. For some time-step $\tau > 0$, we derive bounds on the mistakes $\sum_{t=0}^{\tau} \mathcal{G}_t(x_t, u_t)$. Set $t_0 = 0$ and construct the index-sequence $t_0, t_1, t_2, \ldots, t_N$ as follows:

$$t_k := \begin{cases} \min\left\{t \le \tau \;\middle|\; t > t_{k-1} \text{ and } d(\theta_t, \theta_{t_{k-1}}) > \frac{1}{2}\rho\right\} & \text{, if } k > 1 \\ 0 & \text{, if } k = 0 \end{cases} \tag{28}$$

until for some N, the condition $t \le \tau, t > t_N, d(\theta_t, \theta_{t_N}) > \frac{1}{2}\rho$ becomes infeasible and we terminate the construction. Define the intervals $\mathcal{I}_k := [t_k, \bar{t}_k]$, where $\bar{t}_k := t_{k+1} - 1$ for $k < N$ and $\bar{t}_N = \tau$. The intervals $\mathcal{I}_0, \ldots, \mathcal{I}_N$ are a non-overlapping cover of the time-interval $[0, \tau]$:

$$\bigcup_{0\le k\le N} \mathcal{I}_k = [0, \tau], \quad \mathcal{I}_k \cap \mathcal{I}_{k-1} = \emptyset, \forall k : 1 \le k \le N.$$

Let $(a_0, \ldots, a_N)$ and $(b_0, \ldots, b_N)$ be the parameters selected at the start and end of each interval $\mathcal{I}_k$, respectively: $a_k := \theta_{t_k}$ and $b_k := \theta_{\bar{t}_k}$. Per construction, we know that

$$d(a_k, \theta_t) \le \tfrac{1}{2}\rho \text{ for all } t \in \mathcal{I}_k \tag{29}$$

$$d(a_k, a_{k-1}) > \tfrac{1}{2}\rho \text{ for all } 1 \le k \le N \tag{30}$$

Inequality (23) states that $d(a_k, b_k) \le \frac{1}{2}\rho$ and implies via triangle inequality that for all $t \in \mathcal{I}_k$ holds

$$d(\theta_t, b_k) \le d(\theta_t, a_k) + d(a_k, b_k) \le \rho.$$

Since we picked $b_k = \theta_{\bar{t}_k}$ and the procedure SEL assures $\theta_{\bar{t}_k} \in \mathsf{P}(\mathcal{D}_{\bar{t}_k})$, it means that for some $f_k \in \mathbb{T}[b_k]$, the partial trajectory $(x_{\mathcal{I}_k}, u_{\mathcal{I}_k})$ satisfies the following equations for the time-steps $t \in \mathcal{I}_k$:

$$x_{t+1} = f'(t, x_t, u_t), \quad u_t = \pi[\theta_t](t, x_t) \tag{31}$$

We can therefore conclude that $(x_{\mathcal{I}_k}, u_{\mathcal{I}_k}) \in \mathcal{S}_{\mathcal{I}_k}[\rho; b_k]$. We apply Corollary A.3 to conclude that

$$\sum_{t\in\mathcal{I}_k} \mathcal{G}_t(x_t, u_t) \le M_\rho^\pi \tag{32}$$

for each $k \in \{0, \dots, N\}$. Now, define $\mathbb{S}$ as the following collection of intervals

$$\mathbb{S} := \{\mathcal{I}_k \mid \mathcal{G}_{t_{k+1}}(x_{t_{k+1}}, u_{t_{k+1}}) = 1\}, \tag{33}$$

i.e.: all intervals $\mathcal{I}_k$ where at the start of the *next* interval $\mathcal{I}_{k+1}$ the cost is 1. Combining this with the former bound (32), we can decompose the total mistake sum as

$$\begin{aligned}\sum_{t=0}^{T} \mathcal{G}_t(x_t, u_t) &= \sum_{t\in\mathcal{I}_0} \mathcal{G}_t(x_t, u_t) + \sum_{\mathcal{I}_j\in\mathbb{S}} \sum_{t\in\mathcal{I}_{j+1}} \mathcal{G}_t(x_t, u_t) + \underbrace{\sum_{\mathcal{I}_j\notin\mathbb{S}} \sum_{t\in\mathcal{I}_{j+1}} \mathcal{G}_t(x_t, u_t)}_{0} \\ &= \sum_{t\in\mathcal{I}_0} \mathcal{G}_t(x_t, u_t) + \sum_{\mathcal{I}_j\in\mathbb{S}} \sum_{t\in\mathcal{I}_{j+1}} \mathcal{G}_t(x_t, u_t) \le M_\rho^\pi \left(|\mathbb{S}| + 1\right) \end{aligned} \tag{34}$$

Notice that the last term $\sum_{\mathcal{I}_j\notin\mathbb{S}} \sum_{t\in\mathcal{I}_{j+1}} \mathcal{G}_t(x_t, u_t)$ in the first equation is zero because $\mathcal{I}_j \notin \mathbb{S}$ implies that the next interval $\mathcal{I}_{j+1}$ start with zero cost; due to the cost-invariance property Corollary A.3 it follows that $\sum_{t\in\mathcal{I}_{j+1}} \mathcal{G}_t(x_t, u_t) = 0$. The remainder of the proof is concerned with bounding the cardinality of the collection $\mathbb{S}$.

Bounding $|\mathbb{S}|$: We know that for each l in the range $1 \le l \le |\mathcal{I}_k|$, there exists at least one sub-interval $\mathcal{I}'_l \subset \mathcal{I}_k$, $|\mathcal{I}'_l| = l$ of length l, such that

$$\sum_{t\in\mathcal{I}'} d(\theta_t, \theta_{t+1}) > \tfrac{1}{2}\rho \tfrac{l}{(|\mathcal{I}_k|+l)}. \tag{35}$$

The above has to be true, since otherwise we would contradict (30):

- Let $\mathcal{I}'_1, \dots, \mathcal{I}'_m$, $m = \lceil |\mathcal{I}_k|/l \rceil, |\mathcal{I}'_i| = l, \mathcal{I}'_i \subset \mathcal{I}_k$ be an overlapping cover of $\mathcal{I}_k$, then

$$d(a_k, a_{k+1}) \le \sum_{t\in\mathcal{I}_k} d(\theta_{t+1}, \theta_t) \le \sum_{j=1}^{m} \sum_{t\in\mathcal{I}'_j} d(\theta_{t+1}, \theta_t) \le \tfrac{1}{2}\rho \left\lceil \tfrac{|\mathcal{I}_k|}{l} \right\rceil \tfrac{l}{|\mathcal{I}_k|+l} \tag{36}$$

$$\le \tfrac{1}{2}\rho \left(\tfrac{|\mathcal{I}_k|}{l} + 1 \right) \tfrac{l}{|\mathcal{I}_k|+l} = \tfrac{1}{2}\rho, \qquad \text{(recall that } a_{k+1} = \theta_{\bar{t}_k+1}) \tag{37}$$

  which is a contradiction to (30)

Hence, we can always pick a sequence of sub-intervals $[\tau_k^l, \overline{\tau}_k^l] = \mathcal{I}_k^{(l)} \subset \mathcal{I}_k$ (either of length l or identical to $\mathcal{I}_k$ if $|\mathcal{I}_k| \le l$) such that

$$\sum_{t\in\mathcal{I}_k^{(l)}} d(\theta_t, \theta_{t+1}) > \tfrac{1}{2}\rho \tfrac{l}{|\mathcal{I}_k|+l} \tag{38}$$

Notice that if $|\mathcal{I}_k| \le l$, we pick $\mathcal{I}_k^{(l)} = \mathcal{I}_k$, and therefore the above inequality is vacuously true since, $\sum_{t\in\mathcal{I}_k} d(\theta_t, \theta_{t+1}) \ge d(a_k, a_{k+1}) > \frac{1}{2}\rho \ge \frac{1}{2}\rho \frac{l}{|\mathcal{I}_k|+l}$.

Now, the (γ, T)-weak competitiveness property ensures that for all k and all $t \ge \overline{\tau}_k$ holds:

$$\begin{aligned} & \tfrac{1}{2}\rho \tfrac{T}{|\mathcal{I}_k|+T} < \sum_{t\in\mathcal{I}_k^{(T)}} d(\theta_t, \theta_{t+1}) \le \gamma d_H(\mathsf{P}(\mathcal{D}_{\tau_k}), \mathsf{P}(\mathcal{D}_{\overline{\tau}_k})) \le \gamma d_H(\mathsf{P}(\mathcal{D}_{\tau_k}), \mathsf{P}(\mathcal{D}_t)) \\ \Leftrightarrow \quad & d_H(\mathsf{P}(\mathcal{D}_{\tau_k}), \mathsf{P}(\mathcal{D}_t)) > \tfrac{1}{2}\tfrac{\rho}{\gamma}\tfrac{T}{|\mathcal{I}_k|+T}. \end{aligned} \tag{39}$$

where $d_H(\mathsf{P}(\mathcal{D}_{\tau_k}), \mathsf{P}(\mathcal{D}_{\overline{\tau}_k})) \le d_H(\mathsf{P}(\mathcal{D}_{\tau_k}), \mathsf{P}(\mathcal{D}_t))$ follows from nestedness. From now on, we use the abbreviation P_t to refer to the sets $\mathsf{P}(\mathcal{D}_t)$.

Recall the definition $\mathbb{S} := \{\mathcal{I}_k \mid \mathcal{G}_{t_{k+1}}(x_{t_{k+1}}, u_{t_{k+1}}) = 1\}$ and let k_j denote the j-th interval that belongs to $\mathbb{S}$, i.e.: $\mathcal{I}_{k_j} \subset \mathbb{S}$. Define S_j as a subsequence of $\mathsf{P}_1, \mathsf{P}_2, \dots$ as follows:

$$\mathsf{S}_j := \begin{cases} \mathsf{P}_{\bar{t}_{k_j}} & \text{if } x_{\bar{t}_{k_j}} \in \mathsf{X}_{\bar{t}_{k_j}} \\ \mathsf{P}_{\tau^H_{k_j}} & \text{if } x_{\bar{t}_{k_j}} \notin \mathsf{X}_{\bar{t}_{k_j}} \end{cases} \tag{40}$$

We will show that this collection $P = \{\mathsf{S}_1, \mathsf{S}_2, \dots\}$ of sets S_j is a $\frac{1}{2}\frac{\rho}{\gamma}\frac{T}{M_\rho^\pi+T}$ - separated set in the metric space $(2^{\mathsf{K}}, d_H)$ via the following inequality:

$$\forall j < i : d_H(\mathsf{S}_j, \mathsf{S}_i) > \tfrac{1}{2}\tfrac{\rho}{\gamma}\tfrac{T}{M_\rho^\pi+T}.$$

This is proven below:

- Recall SEL is defined to always pick $\theta_t \in \mathsf{P}(\mathcal{D}_t)$ and π is ρ-uniformly robust and cost-invariant. Due to the SEL property, there always exists a function $f' \in \mathbb{T}[\theta_{t_{k+1}}]$ such that $x_{t_{k+1}} = f'(\bar{t}_k, x_{\bar{t}_k}, \pi[\theta_{\bar{t}_k}](\bar{t}_k, x_{\bar{t}_k}))$. On the other hand, because of the π property, the statement $x_{t_{k+1}} \notin \mathsf{X}_{t_{k+1}}$ implies that one of the following two has to hold at time $\bar{t}_k$:
  1. Assume $x_{\bar{t}_k} \in \mathsf{X}_{\bar{t}_k}$, then it has to hold that $d(\theta_{\bar{t}_k}, \theta_{t_{k+1}}) > \rho$. Notice due to (γ, H)-w.c. property, that $d(\theta_{\bar{t}_k}, \theta_{t_{k+1}}) \leq \gamma d_H(\mathsf{P}_{\bar{t}_k}, \mathsf{P}_{t_{k+1}})$ which gives us
  $$d_H(\mathsf{P}_{\bar{t}_k}, \mathsf{P}_{t_{k+1}}) > \tfrac{1}{2}\tfrac{\rho}{\gamma}$$
  2. Assume $x_{\bar{t}_k} \notin \mathsf{X}_{\bar{t}_k}$, then via Corollary A.3, it follows that $|\mathcal{I}_k| \leq M_\rho^\pi$, which then implies that $d_H(\mathsf{P}_{\tau_k^T}, \mathsf{P}_{t_{k+1}}) > \frac{1}{2}\frac{\rho}{\gamma}\frac{T}{|\mathcal{I}_k|+T} \geq \frac{1}{2}\frac{\rho}{\gamma}\frac{T}{M_\rho^\pi+T}$.
- Taking the minimum of both cases we can see that $d_H(\mathsf{S}_j, \mathsf{P}_{t_{k_j+1}}) > \frac{1}{2}\frac{\rho}{\gamma}\frac{T}{M_\rho^\pi+T}$. Due to nestedness, it holds for $i > j$ that $\mathsf{S}_i \subset \mathsf{P}_{t_{k_j+1}} \subset \mathsf{S}_j$. Thus, it holds $d_H(\mathsf{S}_j, \mathsf{S}_i) \geq d_H(\mathsf{S}_j, \mathsf{P}_{t_{k_j+1}})$ and we arrive at the separation condition:
  $$\forall j < i : \; d_H(\mathsf{S}_j, \mathsf{S}_i) > \tfrac{1}{2}\tfrac{\rho}{\gamma}\tfrac{T}{M_\rho^\pi+T}.$$

We conclude that $|\mathbb{S}| = |P|$ is bounded by the packing number $N(\mathsf{K}, \frac{1}{2}\frac{\rho}{\gamma}\frac{T}{M_\rho^\pi+T})$ and by substituting in the bound (34) and taking the limit $T \to \infty$, we get the total number of mistakes as:

$$\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) \leq M_\rho^\pi \left(N(\mathsf{K}, \tfrac{1}{2}\tfrac{\rho}{\gamma}\tfrac{T}{M_\rho^\pi+T}) + 1 \right)$$

A tighter bound. We can define $\mathsf{K}^\circ = \mathsf{K} \setminus \operatorname{int}(\mathsf{P}(\mathcal{D}_\infty))$ and $\mathsf{S}_j^\circ = \mathsf{S}_j \setminus \operatorname{int}(\mathsf{P}(\mathcal{D}_\infty))$ and notice that S_j° is non-empty for all j: S_j° and $\mathsf{P}(\mathcal{D}_\infty)$ are closed, so $\mathsf{S}_j \subset \mathsf{P}(\mathcal{D}_\infty)$ implies that S_j° contains at least the boundary of $\mathsf{P}(\mathcal{D}_\infty)$. Moreover, we can verify that the corresponding collection $P^\circ = \{\mathsf{S}_1^\circ, \mathsf{S}_2^\circ, \dots\}$ of sets S_j^o is still a $\frac{1}{2}\frac{\rho}{\gamma}\frac{H}{M_\rho^\pi+H}$-separated set in the compact metric space $(2^{\mathsf{K}^\circ}, d_H)$. Therefore we can improve the previous mistake guarantee and state the tighter inequality:

$$\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) \leq M_\rho^\pi \left(N(\mathsf{K} \setminus \operatorname{int}(\mathsf{P}(\mathcal{D}_\infty)), \tfrac{1}{2}\tfrac{\rho}{\gamma}\tfrac{H}{M_\rho^\pi+H}) + 1 \right)$$

□

D Proof of Theorem A.11 and Theorem A.12

D.1 Worst-case state bound

Theorem. *Assume that for procedure $\pi : \mathsf{K} \mapsto \mathcal{C}$ there are constants $\alpha, \beta > 0$, such that $\forall \theta, \theta' \in \mathsf{K}, x \in \mathcal{X}, f \in \mathbb{T}[\theta] : \; \|f(t, x, \pi[\theta'](t, x))\| \leq \alpha d(\theta, \theta')\|x\| + \beta$. The following state bound guarantees hold:*

(i) If SEL is a γ-competitive CMC algorithm, then:

$$\forall t : \quad \|x_t\| \leq e^{\alpha\gamma\phi(\mathsf{K})} \left(e^{-t}\|x_0\| + \beta\frac{e}{e-1} \right).$$

(i) If SEL is a (γ, T)-weakly competitive CMC algorithm, then:

$$\|\boldsymbol{x}\|_\infty \leq \inf_{0<\mu<1} \left(1 + (\alpha\phi(K))^{n^*}\right) \max\{\tfrac{\beta}{1-\mu}, \|x_0\|\} + \beta\sum_{k=0}^{n^*}(\alpha\phi(K))^k$$

where $n^* = N(\mathsf{K}, \frac{\mu}{\alpha\gamma})$ and $\phi(\mathsf{K})$ denotes the diameter of K.

Proof. Part 1: In each time-step t, it holds $x_{t+1} = f(t, x_t, \pi[t, \theta_t])$ for some $f \in \mathbb{T}(\theta_{t+1})$. Therefore, the following inequality holds at each time-step:

$$\|x_{t+1}\| = \|f(t, x_t, \pi[\theta_t](t, x_t))\| \leq \alpha d(\theta_{t+1}, \theta_t)\|x_t\| + \beta \tag{41}$$

We apply Lemma D.1 with the substitution $s_t := \|x_t\|$, $\delta_t := \alpha d(\theta_{t+1}, \theta_t)$ and $c := \beta$, to obtain

$$\|x_t\| \leq e^{\alpha L}\left(e^{-t}\|x_0\| + \beta\frac{e}{e-1}\right) \tag{42}$$

Part 2: We follow the proof technique used in the main result of (Ho and Doyle, 2019) to prove boundedness. Given a closed-loop trajectory $\boldsymbol{x}$, at each time-step t holds $x_{t+1} = f(t, x_t, \pi[\theta_t](t, x_t))$ for some $f \in \mathbb{T}(\theta_{t+1})$. Take an arbitrary time step T. Define $\mathbb{I}$ the set of all indeces $k < T$ for which holds $\|x_{k+1}\| > \mu\|x_k\| + \beta$. Notice that for each $k \notin \mathbb{I}$ holds $\|x_{k+1}\| \leq \mu\|x_k\| + \beta$ while for $k \in \mathbb{I}$ we have at least the inequality $\|x_{k+1}\| \leq \alpha\text{diam}(\mathsf{K})\|x_k\| + \beta$. Now, for each $k \in \mathbb{I}$ holds

$$\mu\|x_k\| + \beta < \|x_{k+1}\| = \|f(k, x_k, \pi[\theta_k](k, x_k))\| \leq \alpha d(\theta_{k+1}, \theta_k)\|x_k\| + \beta$$

which leads to $d(\theta_{k+1}, \theta_k) > \frac{\mu}{\alpha}$. Now, the (γ, T) weak competitive property ensures that $d(\theta_k, \theta_{k+1}) \leq \gamma d_H(\mathsf{P}(\mathcal{D}_k), \mathsf{P}(\mathcal{D}_{k+1}))$. We can therefore conclude that:

$$\tfrac{\mu}{\alpha\gamma} < \tfrac{1}{\gamma}d(\theta_k, \theta_{k+1}) \leq d_H(\mathsf{P}(\mathcal{D}_k), \mathsf{P}(\mathcal{D}_j)), \quad \text{for } j > k \tag{43}$$

Hence, $\{\mathsf{P}(\mathcal{D}_k) \mid k \in \mathbb{I}\}$ is a $\frac{\mu}{\alpha\gamma}$-separated set in K. Therefore $|\mathbb{I}| \leq N(\mathsf{K}, \frac{\mu}{\alpha\gamma})$. Recall again that for each $k \notin \mathbb{I}$ holds $\|x_{k+1}\| \leq \mu\|x_k\| + \beta$ while for $k \in \mathbb{I}$ it holds $\|x_{k+1}\| \leq \alpha\text{diam}(\mathsf{K})\|x_k\| + \beta$. Following the same arguments as in the appendix of (Ho and Doyle, 2019), we obtain the presented $\|\cdot\|_\infty$-bound on $\boldsymbol{x}$. □

Lemma D.1. *Let $\boldsymbol{s} = (s_0, s_1, \dots)$, $\boldsymbol{\delta} = (\delta_0, \delta_1, \dots)$ be non-negative scalar sequences such that $s_{k+1} \leq \delta_k s_k + c$, with $c \geq 0$ and $\sum_{t=0}^{\infty}\delta_t \leq L$. Then s_t is bounded by:*

$$s_t \leq e^L\left(e^{-t}s_0 + c\tfrac{e}{e-1}\right).$$

Proof. Using comparison lemma and Lemma D.2 we can bound s_t as

$$s_t \leq \prod_{k=0}^{t-1}\delta_k s_0 + c\left(1 + \sum_{j=1}^{t-1}\prod_{k=j}^{t-1}\delta_k\right) \leq e^{-t}e^L s_0 + c\left(1 + \sum_{j=1}^{t-1}e^{-j}\right)e^L \tag{44}$$

$$\leq e^{-t}e^L s_0 + ce^L\sum_{j=0}^{\infty}e^{-j} \leq e^{-t}e^L s_0 + c\frac{e^{L+1}}{e-1} \tag{45}$$

□

Lemma D.2. *Let $\boldsymbol{\delta} = (\delta_0, \delta_1, \dots)$ be a non-negative scalar sequence such that $\sum_{t=0}^{\infty}\delta_t \leq L$, then $\prod_{j=0}^{t-1}\delta_j \leq e^{-t}e^L$*

Proof. Recall the basic fact $1 + x \leq e^x$. Then

$$\prod_{j=0}^{t-1}\delta_j = \prod_{j=0}^{t-1}(1 + (\delta_j - 1)) \leq \prod_{j=0}^{t-1}\exp(\delta_j - 1) = \exp\left(\sum_{j=0}^{t-1}(\delta_j - 1)\right) = \exp(L - t) = e^{-t}e^L$$

□

D.2 Mistake guarantees with locally robust oracles

Theorem (Corollary of Thm. 2.5)**.** *Consider the setting and assumptions of Thm.2.5 and Thm.A.9, but relax the oracle robustness requirements to corresponding local versions and enforce the additional oracle assumption stated in Theorem A.11.*

Then all guarantees of Theorem 2.5 (i),(ii) and Theorem A.9 (i),(ii) still hold, if we replace M_ρ^π in Theorem (ii) and Theorem (ii) respectively by $M_\rho^\pi(\gamma_\infty)$ and $M_\rho^\pi(\gamma_\infty^w)$ with the constants:

$$\gamma_\infty = e^{\alpha\gamma\phi(\mathsf{K})}\left(\|x_0\| + \beta\frac{e}{e-1}\right)$$

$$\gamma_\infty^w = \inf_{0<\mu<1}\left(1 + (\alpha\phi(K))^{n^*}\right)\max\{\tfrac{\beta}{1-\mu}, \|x_0\|\} + \beta\sum_{k=0}^{n^*}(\alpha\phi(K))^k$$

where $n^* = N(\mathsf{K}, \frac{\mu}{\alpha\gamma})$ and $\phi(\mathsf{K})$ denotes the diameter of K.

Proof. The results are obtained by combining Theorem A.11 with Corollary of Def.A.2 and repeating the proofs of each part of Theorem 2.5, however this time, replacing M_ρ^π by $M_\rho^\pi(\gamma_\infty)$ and substituting γ_∞ with the corresponding bounds of Theorem A.11. □

E Examples

Here we demonstrate couple examples of how the framework presented in Section 2 can be applied to analyze and design learning and control agents with worst-case guarantees. We start by providing a solution to the scalar linear system described. In the later section we discuss implications for learning and control of uncertain robotic systems: We show a simple design for $\mathcal{A}_\pi(\texttt{SEL})$ agents with finite mistake guarantees; we use well-known control methods in robotics to design robust oracles π and couple it with SEL selections based on competitive (NCBC) algorithms.

E.1 Control of uncertain scalar linear system

Let's consider the very basic problem setting, where we are given an unknown scalar linear system

$$x_{k+1} = \alpha^* x_k + \beta^* u_k + w_k =: f^*(k, x_k, u_k),$$

s.t. $|w_k| \le \gamma^* \le \eta < 1$ and $\alpha^* \in [-a, a]$, $\beta^* \in [1, 1+2b_\Delta]$, and our goal is to reach the target interval $\mathcal{X}_\mathcal{T} = [-1, 1]$ and remain there. We can equivalently phrase this as to achieve the objective $\boldsymbol{\mathcal{G}} = (\mathcal{G}_0, \mathcal{G}_1, \dots)$ with cost functions

$$\mathcal{G}_t(x, u) := \begin{cases} 0, & \text{if } |x| \le 1 \\ 1, & \text{else} \end{cases}, \quad \forall t \ge 0$$

after finitely many mistakes.

Compact parametrization of uncertainty set. We define our parameter space as $\mathsf{K} = [-a, a] \times [1, 1+2b_\Delta]$, and define the true parameter as $\theta^* = (\theta_x^*, \theta_u^*) = (\alpha^*, \beta^*)$ and parametrize the uncertainty set as $\mathcal{F} = \cup_{\theta \in \mathsf{K}} \mathbb{T}[\theta]$ with

$$\mathbb{T}[\theta] := \{t, x, u \mapsto \theta_x x + \theta_u u + w_t \mid \|\boldsymbol{w}\|_\infty \le \eta\}$$

We choose the metric as $d(\theta, \theta') := |\theta_x - \theta_x'| + a|\theta_u - \theta_u'|$. The diameter of the metric space (K, d) is $\phi(\mathsf{K}) = d((-a, 1), (a, 1+2b_\Delta)) = 2(a + b_\Delta)$.

A locally uniformly robust oracle. As an oracle, we take the simple deadbeat controller: $\pi[\theta](t, x) := -(\theta_x/\theta_u)x$. It can be easily shown that π is a locally ρ-uniformly robust oracle for $\boldsymbol{\mathcal{G}}$ for any margin in the interval $(0, \overline{\rho})$, $\overline{\rho} := 1 - \eta$, by noticing the inequality:

$$|x_{t+1}| \le |\theta_x^* x_t + \theta_u^* \pi[\theta_t](t, x_t)| + \eta = |((\theta_x^* - \theta_{x,t}) - (\theta_u^* - \theta_{u,t})\tfrac{\theta_{x,t}}{\theta_{u,t}})x_t| + \eta \tag{46}$$

$$\le (|\theta_x^* - \theta_{x,t}| + |\theta_u^* - \theta_{u,t}||\tfrac{\theta_{x,t}}{\theta_{u,t}}|)|x_t| + \eta \le d(\theta^*, \theta_t)|x_t| + \eta \tag{47}$$

To obtain the mistake function M_ρ^π for a fixed $\rho \in (0, 1-\eta)$, notice that if $d(\theta^*, \theta_t) \le \rho$, then

$$|x_{t+1}| \le \rho|x_t| + \eta = \rho|x_t| + (1-\rho)\tfrac{1}{1-\rho}\eta \quad \Leftrightarrow \quad |x_{t+1}| - \tfrac{\eta}{1-\rho} \le \rho(|x_t| - \tfrac{\eta}{1-\rho}) \tag{48}$$

$$\Rightarrow \quad |x_t| \le \tfrac{\eta}{1-\rho} + \rho^t(|x_0| - \tfrac{\eta}{1-\rho}) \tag{49}$$

Notice that

$$\tfrac{\eta}{1-\rho} + \rho^t|x_0| < 1 \Leftrightarrow t > \tfrac{\log(|x_0|)}{\log(\rho^{-1})} + \underbrace{\tfrac{\log(1-\rho) - \log(1-\rho-\eta)}{\log(\rho^{-1})}}_{c(\rho)}$$

which implies the mistake function

$$M_\rho^\pi(\gamma) \le \tfrac{\log(\gamma)}{\log(\rho^{-1})} + c(\rho) \tag{50}$$

Construction of consistent sets via LPs. The consistent set for the data set $\mathcal{D}_N$ of N observed system transitions (x_i^+, x_i, u_i) can be written as an intersection of K with $2N$ halfspaces:

$$\mathsf{P}(\mathcal{D}_N) = \left\{\theta \in \mathsf{K} \mid \text{s.t.: } \forall 1 \le i \le N : \; x_i^+ - \eta \le \theta_x x_i + \theta_u u_i \le x_i^+ + \eta\right\},$$

It can be constructed online and is convex.

Competitive consistent model chasing via steiner point. Assumption 4.2 applies and we construct a competitive CMC-algorithm by using algorithms for competitive NCBC. Assume we use the steiner point and

denote the selection procedure $\texttt{SEL}_s$ as in (12). $\texttt{SEL}_s$ is a $\frac{n}{2} = 1$-competitive CMC algorithm in euclidean space and since the euclidean norm is bounded above by the 1-norm, $\texttt{SEL}_s$ is also 1-competitive w.r.t. the metric space (K, d).

Mistake guarantee for $\mathcal{A}_\pi(\texttt{SEL}_s)$ We apply the extension of the results in Theorem A.12, (ii). It is easy to see that our π satisfies the extra condition with $\alpha = 1$, $\beta = \eta$. Assuming $|x_0| = 0$, the constant γ_∞ takes the value

$$\gamma_\infty = e^{\alpha\phi(\mathsf{K})}(\|x_0\| + \beta\tfrac{e}{e-1}) = \tfrac{\eta e}{e-1}e^{\phi(\mathsf{K})}. \tag{51}$$

For ease of exposition, assume that $\eta = e^{-1}$ and that we picked $\rho = e^{-1}$. This gives us $M^\pi_\rho(\gamma_\infty) = \phi(\mathsf{K}) - \log(e-2)$ and substituting all constants gives us a finite mistake guarantee for the objective $\boldsymbol{\mathcal{G}}$:

$$\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t) \le M^\pi_\rho(\gamma_\infty)\Big(\tfrac{2L}{\rho} + 1\Big) \approx \phi(\mathsf{K})(1 + 2e\phi(\mathsf{K})) = 8e(a + b_\Delta)^2 + 2(a + b_\Delta) \tag{52}$$

The above inequality shows that the worst-case total number of mistakes grows quadratically with the size of the initial uncertainty in the system parameters θ_x and θ_u. Notice, however that the above inquality holds for arbitrary large choices of a and b_Δ. Thus, $\mathcal{A}_\pi(\texttt{SEL}_s)$ gives finite mistake guarantees for this problem setting for arbitrarily large system parameter uncertainties.

E.2 Learning to follow a trajectory for a class of robotic systems

Consider a general case of online control of uncertain fully-actuated robotic systems. A vast majority of robotic systems can be modeled via the robotic equation of motion (Murray, 2017):

$$\mathbf{M}_\eta(q)\ddot{q} + \mathbf{C}_\eta(q, \dot{q})\dot{q} + \mathbf{N}_\eta(q, \dot{q}) = \tau + \tau_d \tag{53}$$

where $q \in \mathbb{R}^n$ is the multi-dimensional generalized coordinates of the system, $\dot{q}$ and $\ddot{q}$ are its first and second (continuous) time derivatives, $\mathbf{M}_\eta(q), \mathbf{C}_\eta(q, \dot{q}), \mathbf{N}_\eta(q, \dot{q})$ are matrix and vector-value functions that depend on the parameters $\eta \in \mathbb{R}^m$ of the robotic system, i.e. η comes from parametric physical model. Often, τ is the control action (e.g., torques and forces of actuators), which acts as input of the system. Disturbances and other uncertainties present in the system can be modeled as additional torques $\tau_d \in \mathbb{R}^n$ perturbing the equations. Moreover, one can derive from first principles (Murray, 2017), that for many robotic systems (for example robot manipulators) the following two properties hold:

$$\dot{\mathbf{M}}_\eta(q) - 2\mathbf{C}_\eta(q, \dot{q}) \text{ is skew-symmetric} \tag{54a}$$

$$\mathbf{M}_\eta(q)\ddot{q} + \mathbf{C}_\eta(q, \dot{q})\dot{q} + \mathbf{N}_\eta(q, \dot{q}) = \mathbf{Y}(q, \dot{q}, \ddot{q})\eta = \tau + \tau_d \tag{54b}$$

The second equation says that the left-hand-side of equation (53) can always be factored into a $n \times m$ matrix of known functions $\mathbf{Y}(q, \dot{q}, \ddot{q})$ and a constant vector $\eta \in \mathbb{R}^m$. Assume that the disturbances are bounded as at each time t, as $|\tau_d(t)| \le \omega$ where $\omega \in \mathbb{R}^n$ and where the inequality is to be read entry-wise. Consider that we are given a system with unknown η^*, ω^*, where the parameter $\theta^* = [\eta^*; \omega^*]$ is known to be contained in a bounded set K. Assume that our goal is to track a desired trajectory q_d, given as a function of time $q_d : \mathbb{R} \mapsto \mathbb{R}^n$, within ϵ precision, i.e.: Denoting $x = [q^\top, \dot{q}^\top]^\top$ as the state vector and $x_d = [q_d^\top, \dot{q}_d^\top]^\top$ as the desired state, we want the system state trajectory $x(t)$ to satisfy:

$$\limsup_{t\to\infty} \|x(t) - x_d(t)\| \le \epsilon \tag{55}$$

As common in practice, we assume we can observe the sampled measurements $x_k := x(t_k)$, $x^d_k := x^d(t_k)$ and apply a constant control action (zero-order-hold actuation) $\tau_k := \tau(t_k)$ at the discrete time-steps $t_k = kT_s$ with small enough sampling-time T_s to allow for continuous-time control design and analysis.

Control objective $\boldsymbol{\mathcal{G}}^\epsilon$. We phrase trajectory tracking as a control objective $\boldsymbol{\mathcal{G}}^\varepsilon$ with the cost functions

$$\mathcal{G}^\epsilon_k(x, u) := \begin{cases} 0, & \text{if } \|x - x^d_k\| \le \epsilon \\ 1, & \text{else} \end{cases}, \quad \forall k \ge 0,$$

which we wish to achieve online with finite mistake guarantees against the uncertainty set $\mathcal{F}$:

$$\mathcal{F} = \bigcup_{\theta\in\mathsf{K}} \mathbb{T}[\theta], \text{ where } \mathbb{T}[\theta] := \{k, x_k, \tau_k \mapsto f^*(x_k, \tau_k, \tau_d(\,\cdot\,); \theta) \mid \tau_d : [0, T_s] \mapsto \mathbb{R}^n, \|\tau_d\|_\infty \le \omega\},$$

The function f^* denotes the discretized dynamics of (53) w.r.t. the sampling time T_s.

Robust oracle design. We outline how to design a robust oracle based on a well-established robust control method for robotic manipulators proposed in (Spong, 1992b). Define v, a and r as the quantities

$$v = \dot{q}_d - \Lambda \dot{q}, \quad a = \dot{v}, \quad r = \dot{\tilde{q}} + \Lambda \tilde{q}, \quad \tilde{q} = q - q_d \tag{56}$$

and denote $\mathbf{Y}'(q, \dot{q}, v, a)$ as the corresponding $n \times m$ matrix which allows the factorization:

$$\mathbf{M}_\eta(q)a + \mathbf{C}_\eta(q, \dot{q})v + \mathbf{N}_\eta(q, \dot{q}) = \mathbf{Y}'(q, \dot{q}, v, a)\eta \tag{57}$$

Based on the control law presented in (Spong, 1992b), we define the oracle $\pi[\theta](k, x_k)$ for $x = [q; \dot{q}]$ and $\theta = [\eta; \omega]$ through the equations:

$$\pi[\theta](k, x_k) = \mathbf{Y}'(q_k, \dot{q}_k, v_k, a_k)(\eta + u_k) - K_\omega r_k, \quad u = \begin{cases} -\rho \frac{\mathbf{Y}'^\top r_k}{\|\mathbf{Y}'^\top r_k\|_2} & \text{if } \|\mathbf{Y}'^\top r_k\|_2 > \varepsilon \\ -\frac{\rho}{\varepsilon} \mathbf{Y}'^\top r_k & \text{if } \|\mathbf{Y}'^\top r_k\|_2 \leq \varepsilon \end{cases} \tag{58}$$

where $\Lambda, K_\omega \succ 0$ are diagonal positive definite design and where ρ, ε are design variables. Following the the analysis in (Spong, 1992b) and (Corless and Leitmann, 1981) one can design a suiting gain K_ω in terms of ω, such that π is a uniformly ρ robust oracle for $\boldsymbol{\mathcal{G}}^\epsilon$ in the compact parametrization $(\mathbb{T}, \mathsf{K}, d)$.

Remark 5. The analysis in (Spong, 1992b) shows that uniform ultimate boundedness properties of the tracking error $\tilde{x} = [q - q_d; \dot{q} - \dot{q}_d]$ are preserved, if we replace η in equation (58) with some perturbation $\eta + \delta(t)$, $\|\delta(t)\|_2 \leq \rho$ for all t. In (Spong, 1992b), the disturbance τ_d is assumed zero, i.e. the $\omega = 0$ case, and the gain K_0 is left as a tuning variable. However, with standard Lyapunov arguments the analysis of (Spong, 1992b) can be extended to consider the non-zero disturbance case and specify gains K_ω for each ω such that the above oracle π becomes a uniformly ρ robust oracle for the above objective $\boldsymbol{\mathcal{G}}^\epsilon$: For each ω, increase the gain K_ω until the uniform ultimate boundedness guarantee implies the desired ϵ-tracking behavior described by $\boldsymbol{\mathcal{G}}^\epsilon$.

Constructing consistent sets. The linear factorization property (54b) can be exploited to construct convex consistent sets. Denote $\mathbb{T}$ as an uncertain robotic system (54) with some convex compact uncertainty K in euclidean space $(\mathbb{R}^{m+n}, \|\cdot\|_2)$. Recall that we parameterize the bound on the disturbance by $\omega \in \mathbb{R}^n$, i.e., $|\tau_d| \leq \omega$ holds entry-wise and that our system parameter is represented by $\theta = [\eta^\top, w^\top]^\top \in \mathsf{K}$. At the sampled time-steps t_k, equations (54b) says that the measurements $q_k, \dot{q}_k, \ddot{q}_k, \tau_k$ enforce the following entry-wise condition on consistent parameters η and ω:

$$\tau_k - \omega \leq \mathbf{Y}(q_k, \dot{q}_k, \ddot{q}_k)\eta \leq \tau_k + \omega. \tag{59}$$

In matrix form, the consistent set is captured via the following relationship:

$$\underbrace{\begin{bmatrix} \mathbf{Y}(q_k, \dot{q}_k, \ddot{q}_k) & -\mathbf{I}_n \\ -\mathbf{Y}(q_k, \dot{q}_k, \ddot{q}_k) & -\mathbf{I}_n \end{bmatrix}}_{\mathbf{A}_k} \begin{bmatrix} \eta \\ \omega \end{bmatrix} \leq \underbrace{\begin{bmatrix} \tau_k \\ -\tau_k \end{bmatrix}}_{\mathbf{b}_k} \tag{60}$$

Consequently, we have a concrete construction of consistent set at each time t:

$$\mathsf{P}(\mathcal{D}_t) = \left\{ \theta = \begin{bmatrix} \eta \\ w \end{bmatrix} \in \mathbb{R}^{m+n} \middle| \mathbf{A}_t \begin{bmatrix} \eta \\ w \end{bmatrix} \leq \mathbf{b}_t \right\} \cap \mathsf{P}(\mathcal{D}_{t-1}), \quad \mathsf{P}(\mathcal{D}_0) = \mathsf{K} \tag{61}$$

where $\mathbf{A}_k = \mathbf{A}_k(x_k, u_k)$ and $\mathbf{b}_k = \mathbf{b}_k(x_k, u_k)$ are matrix and vector of "features" constructed from current control policy and state at time t via the known functional form of $\mathbf{Y}$.

Designing a competitive chasing selection. The above consistent sets are simply an intersection of half-spaces, hence we are in the setting of Assumption 4.2 and we can instantiate competitive selections from the (NCBC) competitive greedy and Steiner point algorithm algorithms:

- **Greedy projection.** $\texttt{SEL}_p$ selects $\theta_t = \texttt{SEL}_p(\mathcal{D}_t)$ as the solution to the following convex optimization problem, which can be solved efficiently:

$$\begin{aligned} \theta_t = \underset{\theta \in \mathbb{R}^p \cap \mathsf{P}_0}{\arg\min} \quad & \frac{1}{2} \|\theta - \theta_{t-1}\|^2, \\ \text{s.t:} \quad & \mathbf{A}_i \theta \leq \mathbf{b}_i, \forall i = 1, \ldots, t. \end{aligned}$$

- **Steiner-Point.** Alternatively, $\texttt{SEL}_s$ outputs the Steiner point of the polyhedron $\mathsf{P}_\mathbb{T}(\mathcal{D}_t)$, which in principle requires calculating an integral over multi-dimensional sphere. Fortunately, as shown in (Bubeck et al., 2020), the Steiner point can be approximated efficiently by solving randomized linear programs. (We take this approach in our empirical validation.)

Algorithm 3 design of $\mathcal{A}_\pi(\texttt{SEL}_{p/s})$ for ϵ-trajectory tracking for fully actuated robots

for $t = 0, T_s, \ldots, kT_s$ **to** ∞ **do**
  measure $q_k, \dot{q}_k, \ddot{q}_k$
  update polyhedron $\mathsf{P}(\mathcal{D}_k)$ as in (61)
  select according to (12a) or (12b) ▷ selection $\texttt{SEL}_p$ or $\texttt{SEL}_s$
  choose $\tau_k = \mathbf{Y}'(q_k, \dot{q}_k, v_k, a_k)(\eta_k + u_k) - K_{\omega_k} r_k$ using (56), (58) ▷ use oracle $\pi(x_k; \theta_k)$
end for

Mistake guarantee for $\mathcal{A}_\pi(\texttt{SEL}_{p/s})$ Since π is a robust oracle for $\boldsymbol{\mathcal{G}}$ and both $\texttt{SEL}_p$ and $\texttt{SEL}_s$ are chasing consistent models in $(\mathbb{T}, \mathsf{K}, \|\cdot\|)$ with property (D-L) for some $L > 0$, our result Theorem 2.5 tells us that $\mathcal{A}_\pi(\texttt{SEL}_p)$ and $\mathcal{A}_\pi(\texttt{SEL}_s)$ guarantees upfront finiteness of the total number of mistakes $\sum_{k=0}^{\infty} \mathcal{G}_k^\epsilon(x_k, \tau_k)$, which implies the desired tracking behavior guarantee $\limsup_{k\to\infty} \|x - x^d\| \le \epsilon$. Moreover, if we can provide a bound M on the mistake constant $M_\rho^\pi < M$, we obtain from Theorem 2.5, (ii) an explicit performance bound for the tracking performance in the form of the mistake guarantee

$$\sum_{k=0}^{\infty} \mathcal{G}_k^\epsilon(x_k, \tau_k) \le M(\tfrac{2L}{\rho} + 1).$$

F Additional Discussion on Related Work

Online learning of optimal control for linear systems. Many recent learning approaches for control of dynamical systems have focused on the setting of linear optimal control: One is given a linear system and the control objective is to minimize a specified a cost functional. To relate our problem setting to other approaches in this field, we can paraphrase our problem as an optimal control problem by viewing $\mathcal{G}_t(x_t, u_t)$ as stage-cost functions. Our approach provides a solution to the above problem in the general nonlinear system setting, and is applicable even when the initial uncertainty K is so large that there does not exist a so-called "initially stabilizing policy" (a policy $\pi_0(x)$ which guarantees boundedness of the total cost $\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t)$).

Many recent learning and control approaches have focused on the case of Linear Quadratic Regulator (LQR) (Fiechter, 1997; Abbasi-Yadkori and Szepesvári, 2011; Dean et al., 2017, 2018; Cohen et al., 2018), or linear dynamical system with convex costs (Agarwal et al., 2019a,b; Hazan et al., 2020). Our work is instead suitable as well for the nonlinear control setting. In addition, even when restricted to the linear system setting, recent line of work on online learning for control differ from our approach in the following aspects:

- Performance Criteria: We focus on bounding the total cost $\sum_{t=0}^{\infty} \mathcal{G}_t(x_t, u_t)$ as defined in section 1 of the main paper. Our notion of control objective is natural to define in many applications, e.g., most popular robotic goals can be formulated as driving the systems towards a desirable set or trajectories. This differs from, but not incompatible with, the cost metric formulation that is often seen in optimal control and online learning for control work. Specifically, previous effort on learning LQR has been on improving the regret bound of the learning algorithm (Dean et al., 2017, 2018; Abbasi-Yadkori and Szepesvári, 2011; Abbasi-Yadkori et al., 2018; Hazan et al., 2020), Bounding the regret on the average cost, which is natural for LQR, is not sufficient to guarantee control goal completion. In section H, we include additional examples of how we can view our setting under cost minimization formulation, and how sub-linear regret is not sufficient to guarantee performing the control goal. Similarly, no-regret online learning of system parameters does not imply successful goal completion. Instead, we need a stricter notion of convergence than sub-linear regret, such as our version of mistake bound. Regarding control-theoretic guarantee, such as stability, for nonlinear adaptive control: *convergence of average cost does not imply stability of the system.* As stability is often a key objective, this requires a modification of the cost objective, and also a stricter notion of convergence than sub-linear regret.
- Approach: Our proposed approach does not depend on accurate online system identification, which is the focus of several recent work in learning for LQR (Fiechter, 1997; Dean et al., 2017; Hazan et al., 2020; Cohen et al., 2018). As we consider parametric uncertainty, it is plausible to also adopt system identification approach for the non-linear control settings. However, online system identification with arbitrarily small error is known to be very challenging. As shown by (Dahleh et al., 1993), the sample complexity for identifying linear systems under bounded adversarial noises can be exponential in the worst case.
- Assumptions about parameter uncertainty: Some previous work in linear systems (Cohen et al., 2018; Dean et al., 2018; Hazan et al., 2020) assume knowledge of a stabilizing controller $\pi_{\text{safe}} : \mathcal{X} \mapsto \mathcal{U}$ for the true

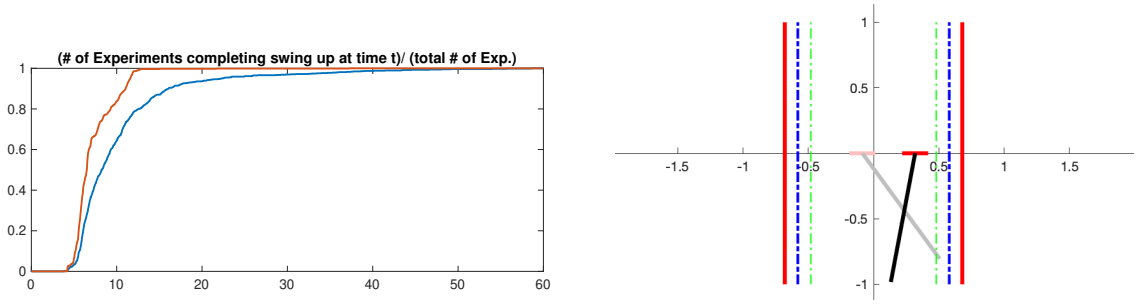

Figure 2: (Left) Percentage of runs that completed swing up goal before time t: perfect candidate controller $\pi[\theta^*]$ (red) vs. online algorithm (blue). (Right) Experiment setup.

unknown system parameter θ^*. In our setting, we do not require such an assumption but merely that the uncertainty set K is compact and that for each $\theta \in \mathsf{K}$ we can find a feasible robust policy $\pi[\theta]$ that meets the desired control objective. The assumption of access to oracle class that yields guarantee *if the posited system were correct* is reasonable and realistic from a control theory perspective.

Robust Adaptive Nonlinear Control. Naturally, our problem setting is of great interest to the control community, and have had a relatively long history (Polycarpou and Ioannou, 1993; Yao and Tomizuka, 1995; Liu et al., 2009; Krstic et al., 1995; Ioannou and Fidan, 2006). Yet, most of traditional adaptive control approaches can not be applied to the general problem setting we consider without making restrictive assumptions. As an example, in contrast to most adaptive control methods, our framework applies to non-feedback linearizable nonlinear system (see also discussion in Section G.1.2). Furthermore, many adaptive control techniques can not build on top of methods from other areas of control, like robust control theory, but rather propose separate control algorithms for each problem setting. Also, robust stability analysis and thorough empirical validation is for most methods largely unavailable. In fact, most relevant empirical results are only presented for arguably much simpler settings than the cart-pole swing-up problem, which is considered in this work. In addition, we highlight the two methodological distinctions:

- Our proposed technique takes a significantly different perspective from traditional adaptive control approach. We provide a modular framework, which allows to combine robust control tools with online learning algorithms to provide desired guarantees online. We unify the treatment of both uncertain system parameters and unknown disturbance via the construction of confidence sets of candidate systems that are consistent with the historical collected observations. The estimation of such consistent sets is also easily attainable for most robotic systems and allows for nonasymptotic convergence guarantee. Among relevant adaptive control literature, perhaps most closely related to ours is Multi-Model Adaptive Control (MMAC) from (Anderson et al., 2001). The MMAC principle needs to run a high-dimensional Multi-Estimation routine online, which requires the design of nonlinear observers (with the Matching and Detectability property - see (Hespanha et al., 2003)) for a sufficiently dense covering set of the parameter space. A general construction of such a family is only shown for linear systems (See (Hespanha et al., 2003) and references therein) and it is not clear whether designing a tractable Multi-Estimator for the cart-pole system is possible.

G Supplementary Details - Empirical Validation

We will demonstrate that the presented method can learn very challenging control tasks efficiently from limited amount of data online. We will demonstrate this on the problem of learning to swing up an inverted pendulum on a cart, also often referred to as the cart-pole system. Yet, in contrast to prior empirical results on this system, we will consider a much more challenging setting by incorporating additional constraints representing necessary restrictions that arise when interacting with real physical systems.

G.1 A hard nonlinear control task to learn: Swinging up the cart-pole system

Here, we will discuss the cart-pole model and the control task we will be tasking $\mathcal{A}_\pi(\texttt{SEL})$ to learn. In addition, we will highlight why this particular control task is a very challenging one.

A realistic model of the cart-pole.

We will model the cart-pole system in a way that is consistent with how a learning agent $\mathcal{A}$ would be deployed on

a real dynamical system $\mathcal{M}$: The state $x(\tau)$ of the dynamical system $\mathcal{M}$ evolves in continuous-time τ according to its continuous-time system equations (usually in the form of an ODE); With respect to some fixed sampling T_s, the agent $\mathcal{A}$ receives observations at discrete-time instances $\tau_k = kT_s$ and decides actions that are kept fixed for time-window $[kT_s, (k+1)T_s]$ between sampling instances. This is depicted below in Figure 3.

Given a discrete-time agent $\mathcal{A}$ and dynamical system $\mathcal{M}$ with state x and input u with continuous model $\frac{dx}{d\tau}(\tau) = g(x(\tau), u(\tau))$ in continuous time τ

At time instances $\tau = 0, T_s, \ldots, kT_s, \ldots$:
$\mathcal{A}$ collects observation $x(kT_s)$
$\mathcal{A}$ selects action u_k that is frozen for the time window $[kT_s, (k+1)T_s]$
$x([kT_s, (k+1)T_s]) \leftarrow$ system $\mathcal{M}$ evolves $\frac{dx}{d\tau}(\tau) = g(x(\tau), u(\tau))$ from $x(kT_s)$ with input held constant $u([kT_s, (k+1)T_s]) = u_k$.

Figure 3: The usual interaction protocol with real-world dynamical systems

The cart-pole system is governed by the following nonlinear differential equations:

$$(m_c + m_p)\ddot{x}_c + m_p l\ddot{\theta}_p \cos\theta_p - m_p l\dot{\theta}_p^2 \sin\theta_p - b_x\dot{x}_c = f_u \tag{62a}$$

$$m_p l\ddot{x}_c \cos\theta_p + m_p l^2\ddot{\theta}_p + m_p gl \sin\theta_p - b_\theta\dot{\theta} = 0 \tag{62b}$$

We are using the notation in (Tedrake, 2020) and refer to the same reference for detailed derivations. The variables x_c and θ_p stand for cart position and pole angle in counterclockwise direction and f_u represents the force exerted on the cart. Furthermore, $\theta = 0$ denotes the downward position. The uncertain parameters are (m_c, m_p, l, b_x, b_θ) and represent cart and pole mass, pole length, friction coefficients b_x and b_θ, respectively. $g = 9.81$ is the gravity constant.

Furthermore, (65) can be converted into the input affine standard form

$$\dot{x} = F(x) + g(x)u \tag{63}$$

where $x = [x_c, \theta_p, \dot{x}_c, \dot{\theta}_p]^T$, $u = f_u$, (see (Tedrake, 2020) for description of $F(x)$ and $g(x)$). We will choose a **sampling-time** $\tau_s = 0.02$sec ($1/\tau_s = 50$Hz) that would be easy to realize with current technology. Under the above interaction protocol Figure 3, we can abstract the transitions between discrete-time instances $x(kT_s), u(kT_s) \to x((k+1)T_s)$ as a discrete-time system. Thus, we can equivalently represent the system (63) at sampling times through the discrete-time system

$$x_t = \phi_{T_s}(x_{t-1}, u_{t-1}), \quad \phi_{T_s}(x, u) := \alpha(T_s), \text{ s.t. : } \dot{\alpha} = F(\alpha, u), \alpha(0) = x, \tag{64}$$

where we will denote $x_t := x(tT_s)$ and $u_t := u(tT_s)$ ($t \in \mathbb{N}$) to be samples of the continuous-time signals $x(\tau)$, $u(\tau)$ at time $t\tau_s$. We assume that **only noisy measurements** $\hat{x}_t = x_t + n_t$ are obtainable, where we assume n_t to be bounded noise. Furthermore, we will **approximate knowledge of the derivative** $\dot{x}_t$ simply as $(\hat{x}_t - \hat{x}_{t-1})/T_s$. Knowledge of the exact transient function $\phi_{T_s}(\,\cdot\,,\,\cdot\,)$ is usually not avaiable and approximations are necessary. In simulations, we will **accurately approximate** $\phi_{T_s}(\,\cdot\,,\,\cdot\,)$ by using a Runge-Kutta method of order 4 with a fix step-size chosen an order of magnitude smaller than T_s. This is in contrast to for example OpenAI Gym (Brockman et al., 2016b), a popular implementation of the cart-pole environment which approximates (64) using only the forward euler method and does not model the real physical cart-pole system accurately.

The control task. We will consider the swing up control task: Starting off with $x(0) = 0$, $\theta(0) = \pi$, i.e. the pole in the **downward** position and cart position in the center, choose the force u_t to bring the pole to the upward position $\theta = 0$ and cart position at $x = 0$ and balance the system there. More specifically, by balancing we mean to keep the state x within a target set $\mathcal{X}_\mathcal{G} = [-\varepsilon_x, \varepsilon_x] \times [-\varepsilon_\theta, \varepsilon_\theta] \times [-\varepsilon_{\dot{x}}, \varepsilon_{\dot{x}}] \times [-\varepsilon_{\dot{\theta}}, \varepsilon_{\dot{\theta}}]$ of allowed tolerance, i.e. reaching and staying close to the upright cart-pole position.

G.1.1 A challenging nonlinear control problem

Despite being argueably one of the most popular examples of a nonlinear control problem, the cart-pole system belongs to a class of nonlinear systems that is particularly hard to control: It is **nonlinear**, **non-state feedback linearizable** and **non-minimum phase**. As a consequence, the following standard nonlinear control approaches can **not** be applied:

1. State-feedback linearization (Slotine et al., 1991; Khalil and Grizzle, 2002; Sastry, 2013)

2. Input-output linearization (Khalil and Grizzle, 2002; Sastry, 2013): Due to non-minimum phasedness, it's hard to find a "stable" output

3. Linearization: No linear controller can swing up and balance the cart-pole.

4. Backstepping (Krstic et al., 1995): The dynamics do not conform to the so-called "strict-feedback" form.

Remark 6. A related control task is the "balancing cart pole" task in the RL literature (See OpenAIGym as an example (Brockman et al., 2016b)). The "balancing task" is concerned with balancing the cart-pole system, **but** allowing the cart-pole system to start from the **upward** position. This is a much easier task. As an example of a distinguishing factor: a linear LQR controller can accomplish the balancing task, while no linear controller can perform the swingup task.

G.1.2 A challenging adaptive nonlinear control problem

The previous control challenges hold, even if the model of the cart-pole is perfectly known. If we consider the online learning problem of the swingup problem, these difficulties are only exacerbated: Most standard nonlinear adaptive control techniques can not be directly applied to the problem. As an example, the following popular methods can not be used: MRAC (Ioannou and Fidan, 2006), (Ioannou and Sun, 2012), nonlinear adaptive back-stepping (Krstic et al., 1995), adaptive-sliding mode control (Slotine et al., 1991), computed torque based methods like in (Ortega and Spong, 1989). On the other hand, the more recent approach of multi-model adaptive control (MMAC) (Anderson et al., 2001) in principle covers the cart-pole swingup problem, yet it is not clear how to exactly instatiate the approach presented in (Anderson et al., 2001), as one requires a design of a suiting nonlinear observer and switching logic for the cart-pole which is not clear how to do.

The cartpole system can be also described to be a so-called *underactuated* robotic system. This system class often inherits the same difficulties as the one mentioned above and this recent work (Moore and Tedrake, 2014) (Nguyen and Dankowicz, 2015) is discussing the difficulties and some recent progress towards adaptive control for underactuated systems. Nevertheless, both methods do not offer solutions for the adaptive swingup problem of the cart-pole.

To the best of our efforts, we have not been able to find any empirical results in the adaptive control literature that show how to learn to swingup the cart-pole system with control-theoretic guarantees.

G.1.3 A challenging reinforcement learning problem

To characterize the difficulty of the cart-pole swing-up task as a learning problem, we tried to estimate its sampling complexity in the context of reinforcement learning and compare it to other common control tasks often used in reinforcement learning benchmark. As a proxy for the sample complexity, we tuned a state-of-the-art reinforcement learning algorithm to learn the cart-pole swing up task in as few as possible samples. We setup the swingup task as an episodic RL problem with $40\,s$ episode length, $T_s = 0.02$ and used a smooth sigmoid based dense reward function from (Tassa et al., 2018). As a simulation environment we used the original OpenAIGym (Brockman et al., 2016b) cart-pole environment and modified it to fit the swingup problem we consider.

As a representative of a state-of-the-art RL method, we used the widely successful PPO (Schulman et al., 2017) and obtained the learning curve presented in Figure 4. The learning curve presented in Figure 4 is the best run after approximately 200 iterations of hyperparameter tuning and modifications and it shows that despite being given a dense reward function, PPO2 needed over $3 * 10^6$ time-steps, to learn how to swingup and over $4.5 * 10^6$ time-steps, to find an optimal policy that swings up the fastest. At a sampling-time of $T_s = 0.02\,s$, the corresponding needed interaction time with the system is 16.5 hours and 25 hours, respectively. In comparison, the traditional cart-pole balancing task (balancing the cart-pole where each episode starts from the **top** position) only requires PPO2 around 50000 time-steps to learn. In the original paper (Schulman et al., 2017), PPO has been tested on other common RL benchmark tasks *HalfCheetah-v1*, *Hopper-v1*, *InvertedDoublePendulum-v1*, *InvertedPendulum-v1*, *Reacher-v1*, *Swimmer-v1*, *Walker2d-v1* and has been shown to learn good policies in much fewer than 10^6 samples. In comparison to those RL tasks, our cart-pole swing-up tasks requires (after hyperparameter tuning and optimization) PPO to take significantly more samples before it can learn a good policy that achieves the swing-up task.

Hyperparameters and necessary modifications. We used the PPO2 implementation of stable baselines (Hill et al., 2018) and arrived at the hyperparameters presented in Table G.1.3. In addition, we had to perform additional modifications to obtain the result in Figure 4:

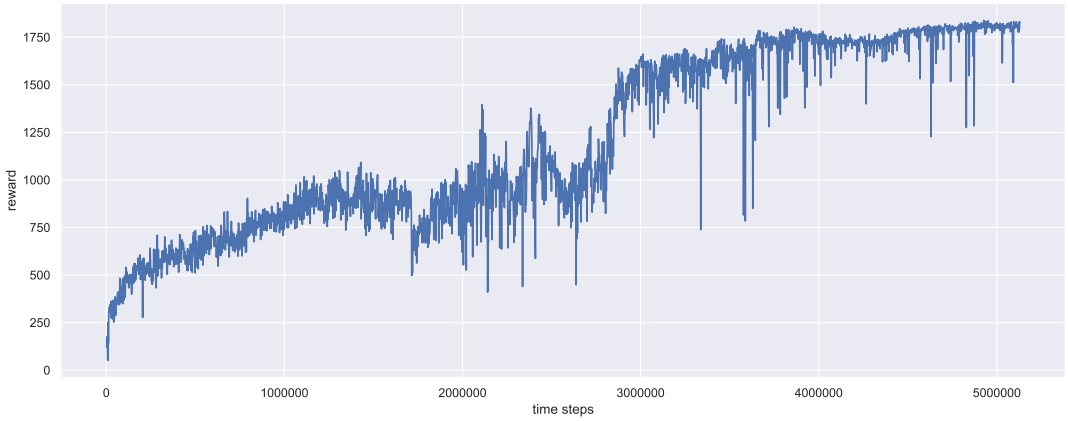

Figure 4: *Learning curve of PPO2 for cartpole swingup:* The max episodic reward is 2000 and a satisfactory swingup performance can be obtain starting at around a reward of 1500.

Table 3: Learning rate schedule based on episodic reward

episodic reward	700	1300	1400	1500	1600	1800	2001
learning rate	1e-3	1e-3	1e-4	1e-4	1e-4	1e-4	5e-5

1. Instead of the observations $x = [x_c, \theta_p, \dot{x}_c, \dot{\theta}_p]$, we give PPO the observation vector $x' = [x_c, \cos(\theta_p), \sin(\theta_p), \dot{x}_c, \dot{\theta}_p]$
2. We set $\beta_1 = 0$ of the ADAM optimizer in tensorflow
3. We scheduled the learning rate based on the episodic reward according to Table G.1.3.

G.2 Our validation setting: cart-pole swing-up problem subject to safety constraints

To show that our online learning approach allows to incorporate control theoretic guarantees useful for applications with real physical systemsm, we considered a harder version of the cart-pole swingup task that requires additional safety specifications to be met. We will start, by motivating our chosen setup:

Real world challenges. Developping an online learning agent for controlling a real physical systems comes with unique difficult challenges:

1. *Safety concerns impose heavy restrictions on online exploration*: In order to guarantee safety during operation, in most cases (Dulac-Arnold et al., 2019), strong restrictions must be imposed on the actions of the learning agent. Most commonly, safety constraints can be formulated in terms of hard and soft constraints enforced on the state x_t and input u_t of the system. To enforce these safety specifications online, usually one deploys a form of safety controller that is overseeing the proposed actions of the learning agent and overrides them, if they are deemed unsafe w.r.t. the safety specifications. A common approach is to implement the safety controller using control barrier functions (Ames et al., 2017), (Gurriet et al., 2018). This necessary practice causes a major challenge particularly for online learning, since exploration is potentially greatly hindered.
2. *System resets are costly or impossible*: In many robotics application, like autonomous systems, online learning is very desirable, since resets are costly to realize or sheer impossible. Example: UAVs, autonomous cars, robots in close contact with humans.
3. *Online data is limited and noisy*

To represent a benchmark for the above challenges, we will modify the cart-pole swingup problem by adding the above restrictions to the problem setup:

Table 4: PPO hyperparameters

Hyperparameter	Value
number of steps in epoch	8192
gamma	0.999
clip-range	0.2
num of optimization-epochs	10
GAE-lambda	0.995
num of mini-batches	8
entropy coefficient	1e-6

1. **safety**: the cart position x has to be kept in an interval $[x_{min}, x_{max}]$ for all time t (and $x_{max} - x_{min}$ is closed to pole length). In our setting we choose $x_{max} = 0.6m$.

   Remark 7. Notice that the above constraint poses a challenge for the swing up task, as the range of motion is severly limited. In some our experiment settings we even have $x_{max} < l$.

2. safety controller override: A safety controller (Control barrier function based) overrides any control actions that could bring the cartpole to close to the bounds x_{max}.
3. acceleration $\ddot{x}$ should not (or rarely) exceed an interval $[\ddot{x}_{min}, \ddot{x}_{max}]$. In our setting we choose $x_{max} = 0.5g$.

   Remark 8. It should be noticed that the constraint $\ddot{x} \leq 0.5g$ is a restrictive condition, as it can be shown in (Åström and Furuta, 2000) that this acceleration is not enough to swing-up the pole in one or two swings.

4. no system reset is allowed,
5. the input F has to lie within the bounds $[-200N, 200N]$

Moreover for our online learning setting described in Section 1, we assume the initial parameter uncertainty as described in Table. (5). Next, we describe the oracle function we used to instantiate our approach $\mathcal{A}_\pi(\texttt{SEL})$.

G.3 Modelbased oracle for cart-pole swing up

Recall the nonlinear dynamics of cart pole system:

$$\begin{aligned}(M+m)\ddot{x} - ml\ddot{\phi}\cos(\phi) + ml\dot{\phi}^2\sin(\phi) - b_x\dot{x} &= F \\ l\ddot{\phi} - g\sin(\phi) - b_\phi\dot{\phi} &= \ddot{x}\cos(\phi)\end{aligned} \tag{65}$$

Let x and $\dot{x}$ be the position and velocity of the cart and ϕ, $\dot{\phi}$ the angle and angular velocity of the pole. F is the force onto the cart pole and serves as our control input to the system. Furthermore, let $\bar{a}$ be the maximal cart acceleration allowed, and $\bar{d}$ be the maximum distance the cart is allowed to move from the center.

As a first step, we will perform so called partial feedback linearization. Let $F_d(\ddot{x}, \dot{x}, \phi, \dot{\phi}, \dot{x})$ be the force F we need to apply at time t in order to achieve a desired cart acceleration of $\ddot{x}_d$. Multiply the second equation of (65) by $ml\cos(\phi)$ and add it to the first, to see that F_d has to be chosen as:

$$F_d(\ddot{x}_d, \dot{x}, \phi, \dot{\phi}, \dot{x}) = (M + m\sin(\phi)^2)\ddot{x}_d - mg\cos(\phi)\sin(\phi) + ml\dot{\phi}^2\sin(\phi) - b_x\dot{x} \tag{66}$$

By choosing $F = F_d(\ddot{x}_d, \dot{x}, \phi, \dot{\phi}, \dot{x})$, we can now treat the desired acceleration $\ddot{x}_d$ as our new control input. With respect to our new input $\ddot{x}_d$, we can simplify the original equations (65) to

$$\ddot{x} = \ddot{x}_d \tag{67}$$

$$l\ddot{\phi} - g\sin(\phi) - b_\phi\dot{\phi} = \ddot{x}_d\cos(\phi) \tag{68}$$

The swingup controller consists now of three separate control laws that are later combined.

Around up-right position: static linear LQR controller If the pole has small enough kinetic energy and is close to the upright position, we simply choose $\ddot{x}_d$ to be LQR-state feedback controller based on the system (67) linearized around the equilibrium position $x = 0, \dot{x} = 0, \phi = 0, \dot{\phi} = 0$. The policy then takes the form

$$\ddot{x}_{d,LQR} = -K_{LQR}(\theta)z$$

Swing up-controller: energy-based controller. The Swing-Up controller is based on an elegant energy-based approach by (Åström and Furuta, 2000), in which we simply choose $\ddot{x}_d$ to control the total normalized energy

$$E(\phi, \dot{\phi}) = \frac{l}{2g}\dot{\phi}^2 + cos(\phi) \tag{69}$$

of the pole. In depth derivation can be found in (Åström and Furuta, 2000) and $\ddot{x}_d$ takes the form:

$$\ddot{x}_{d,swing} = -\mathbf{Sat}_{\bar{a}}\left(\frac{1}{2}\gamma|\cos(\phi)|(E(\phi,\dot{\phi})-1)\text{sign}(\dot{\phi}\cos(\phi))\right) \tag{70}$$

where $\mathbf{Sat}_{\bar{a}}$ is the saturation function which saturates at the max specified acceleration $\bar{a}$.

Wrapping a safety controller. As part of our oracle policy, we also use a control barrier function controller that prevents to trigger the safety policy. We do this simply by internally overriding our swing-up $\ddot{x}_{d,swing}$ or balancing $\ddot{x}_{d,LQR}$ terms, if we get too close to the boundary of $[-x_{max}, x_{max}]$. To this end, define the $B(x, \dot{x})$ as the barrier function

$$B(x, \dot{x}) = \frac{1}{2\bar{a}}\dot{x}|\dot{x}| + x$$

and define $\ddot{\phi}_{max} := \bar{a}g/l * \sin(30^\circ)$.

Full controller, including safety override. The full controller can be described below as:

Algorithm 4 oracle policy $\pi[\theta]$ under potential safety policy π_{safe} override

Input: $z = [x, \phi, \dot{x}, \dot{\phi}]$, parameters $\theta := [M, m, l, b_x, b_\theta]$, ϵ_{safe}
Output: F
if $|x| < \bar{d} - \epsilon_{safe}$ **then**
  if $|\dot{\phi}^2/\ddot{\phi}_{max}| < 60^\circ$ **and** $\cos(30^\circ/\ddot{\phi}_{max} + \phi\text{sign}(\dot{\phi})) > \cos(30^\circ)$ **and** $|-K_{LQR}(\theta)z| \leq \bar{a}$ **then**
    $\ddot{x}_d = -K_{LQR}(\theta)z$
  else
    $\ddot{x}_d = -\mathbf{Sat}_{\bar{a}}[\frac{1}{2}\gamma|\cos(\phi)|(E(\phi,\dot{\phi})-1)\text{sign}(\dot{\phi}\cos(\phi))]$
  end if
  $\ddot{x}_{d,back} = -\bar{a}\,\text{sign}(\dot{x})$
  $\lambda = \frac{|B(x,\dot{x})|}{\bar{d}-\epsilon_{safe}}$
  if $B(x, \dot{x}) \geq 0$ **then**
    $\ddot{x}_d \leftarrow (1-\lambda^2)\ddot{x}_d + \lambda^2 \min\{\ddot{x}_d, \ddot{x}_{d,back}\}$
  else
    $\ddot{x}_d \leftarrow (1-\lambda^2)\ddot{x}_d + \lambda^2 \max\{\ddot{x}_d, \ddot{x}_{d,back}\}$
  end if
  $F = F_d(\ddot{x}_d, z)$
else
  $F = \pi_{\text{safety}}(z)$
end if

The controller, switches to an LQR if the system is close to the upright position and otherwise defaults to the swing up controller that brings the pendulum to the right energy level. A correction is performed to the previous control action depending on the barrier-function value $|B(x, \dot{x})|$. As $|B(x, \dot{x})|$ gets closer to the boundary $\bar{d} - \epsilon_{safe}$ the controller prioritizes to safety and overwrites the previous planned control action. If x exceeds the buffer $\bar{d} - \epsilon_{safe}$, then a safe policy is being called, that bring the cart position back to the region $[-\bar{d} + \epsilon_{safe}, \bar{d} - \epsilon_{safe}]$.

G.4 Selection process SEL

We apply the approach presented in the main paper Section 5 to obtain polytopes of consistent parameters of P_t for the lumped parameters $p = [m_c + m_p, m_p l, b_x, l, b_\theta, \tau_{d,x}, \tau_{d,\theta}]$. We use randomized LP's (Bubeck et al., 2020), to approximate the Steiner point of the polytope P_t and select the corresponding oracle policy $\pi(\,\cdot\,; \theta_t)$ as described in the meta-algorithm 1.

Table 5: Initial Parameter Uncertainties and Range of Test Parameters

Parameter	Uncertainty	Test Parameters
M	[0.1,5]	$\{1, 2, 4\}$
m	[0.1,1]	$\{0.1, 0.2, 0.4\}$
l	[0.05,1]	$\{0.1, 0.2, 0.4, 0.6, 1.0\}$
b_x	[0,20]	$\{0, 10\}$
b_θ	[0,2]	$\{0\}$

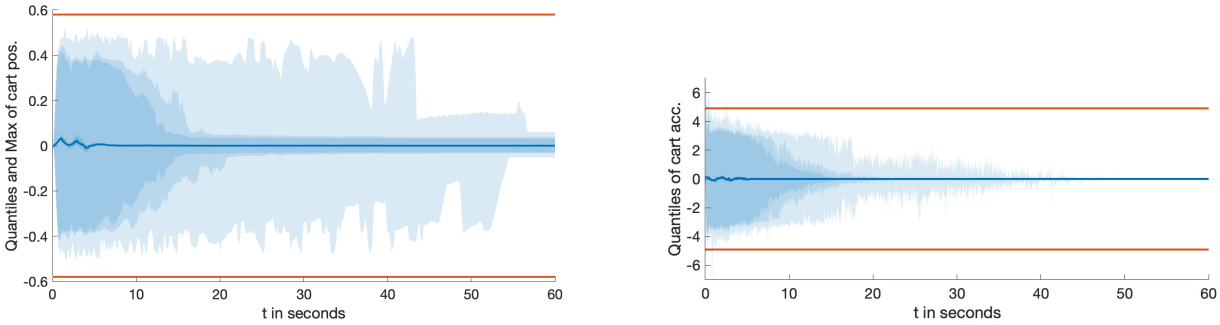

Figure 5: *Safety and Cost Quantiles over 900 Experiments*: Left: min, max, median, top/bottom 5% and 10% quantiles of cart position x_t over time t. The cart position respects the safety guarantees over all 900 experiments. Right: top/bottom 5% and 10% quantiles of cart acceleration a_t over time t. The acceleration stays well within the desired acceleration bounds $0.5g$.

G.5 Simulation Results

To test our algorithm in the adversarial setting we described in Section 1, we tested the algorithm against all θ^*, which are a combinations of the test parameters given in Table. (5) and multiple noise random seeds, resulting in a total of 900 experiments.

As previously discussed, the safety policy π_{safe} overrides unsafe actions of the learning agent and thus hinders greatly exploration during online learning. As a key benefit of our approach, we show that the performance controller is practically unhindered by the safety controller, as seen in Fig. 2. Fig. 2 shows that in all 900 experiments, the online controller was able to learn the swing up consistently **under 60 seconds in a single episode**. In fact, in 80% of the experiments, the online algorithm learns to swing up the cart pole in **under 12 seconds.** Comparing the online algorithm to the corresponding ideal true oracle controller $\pi(\,\cdot\,;\theta^*)$, shows that the online controller $\mathcal{A}_\pi(\texttt{SEL})$ is only marginally slower than $\pi(\,\cdot\,;\theta^*)$. To the best of our knowledge, it seems that there is no current other competitive algorithm for that problem setting with similar performance. In addition to fast online learning, as shown in Figure 5, our approach satisfies over all 900 experiments the desired safety bounds and acceleration bounds.

H Mistake Guarantees vs Sublinear regret

A common performance metric in online learning for control is phrased in terms of the regret $R(T)$. For our general problem setting, we show that sublinear regret does not imply finite mistake guarantees, however finite mistake guarantees do imply sublinear regret.

H.1 Regret Definition

Given a disturbance w_k and initial condition ξ, assume the ideal policy π^* would generate the trajectory x_t^*, u_t^*, while the online algorithm characterized by the sequence of policies $\{\pi_t\}$, produces x_t, u_t. The optimal total cost of $J^*(T)$ at time T is defined as

$$J^*(T) := \sum_{k=0}^{T} C(x_k^*, u_k^*).$$

Then the regret $R(T)$ usually refers to the sum of costs of the online algorithm up until time T minus the sum of costs that the optimal policy π^* would attain[3]:

$$R(T) := \sum_{k=0}^{T} C(x_k, u_k) - J^*(T). \tag{71}$$

Sub-linear regret is defined as

Definition H.1. $R(T)$ is called sublinear regret if $R(T) = o(T)$ or equivalently[4]:

$$\lim_{T\to\infty} \frac{1}{T} R(T) = \lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T} (C(x_k, u_k) - C(x_k^*, u_k^*)) = 0, \tag{72}$$

The slower $R(T)$ grows with T, (for example $O(\log(T))$), the faster convergence we can guarantee to the above limit.

H.2 Asymptotic Bounds

To the contrary, in our problem of learning and control, we are interested in the guarantees of the type:

1. For all ξ and (bounded) w_k, we have bounded cost deviation from optimal performance:
$$\sup_k |C(x_k, u_k) - C(x_k^*, u_k^*)| < \infty \tag{73}$$
2. Convergence to optimal asymptotic cost bound:[5]:
$$\limsup_{k\to\infty} C(x_k, u_k) \leq \limsup_{k\to\infty} C(x_k^*, u_k^*) \tag{74}$$
3. Quantifying a rate of convergence for (74).

The above conditions increase in strength and latter conditions imply previous ones, i.e. $(3 \Rightarrow 2 \Rightarrow 1)$.

H.3 Sub-Linear Regret does not imply asymptotic bounds

Many online learning algorithms characterize their performance in terms of regret bounds and ideally, we would expect sublinear regret as a performance metric to subsume some of the above guarantees. Unfortunately, simple derivations show that without additional assumptions, sublinear regret growth is not sufficient to show even the weakest desired condition (73). The reason for that is intrinsic to the very definition of regret and simple real analysis arguments will suffice to demonstrate that.

Abbreviate $c_k := C(x_k, u_k)$, $c_k^* := C(x_k^*, u_k^*)$ and define the sequences

$$s_k := c_k - c_k^* \tag{75}$$

$$m_k := \frac{1}{k} \sum_{j=0}^{k} (c_k - c_k^*) = \frac{1}{k} \sum_{j=1}^{k} s_k \tag{76}$$

Now, sublinear regret guarantees the convergence

$$\lim_{k\to\infty} m_k = 0$$

while the requirements from Section H.2 can be stated as

$$\sup_k |s_k| < \infty \qquad\qquad \limsup_{k\to\infty} c_k \leq \limsup_{k\to\infty} c_k^* \tag{77}$$

First, we will show

$$\lim_{k\to\infty} m_k = 0 \not\Longrightarrow \sup_k |s_k| < \infty.$$

[3]Not the most wide-spread definition, but the most suited for adaptive control setting. See for example (Ziemann and Sandberg, 2020)

[4]We will assume that the optimal policy π^* is robustly stable w.r.t. w

[5]Replacing lim sup with lim on the other hand might not be defined in the presence of disturbances

and then investigate other relationships between s_k and m_k. Below is a summary of the statements proven by counter-examples:

1. $\lim_{k\to\infty} m_k = 0 \implies \sup_k |s_k| < \infty$

   Proof. Consider s_k, where $s_{e^n} = n$ and otherwise 0. Define $\bar{n}(k) := \lfloor \log(k) \rfloor$, then
   $$m_k \leq m_{\bar{n}(k)} = \frac{1}{e^{\bar{n}(k)}} \sum_{i=1}^{\bar{n}(k)} i = \frac{\bar{n}(k)(\bar{n}(k)+1)}{2e^{\bar{n}(k)}}.$$
   This shows $\lim_{k\to\infty} m_{\bar{n}(k)} = 0$, but s_k is unbounded. □

2. For all $\varepsilon > 0$: $\sup_k s_k - \inf_k s_k < \varepsilon \implies m_k$ converges

   Proof. Define s_k as the sequence
   $$s_k = (\underbrace{\underbrace{1, \delta, 1, 1, \delta, \delta}_{6}, \underbrace{1, \ldots, 1}_{6}, \underbrace{\delta, \ldots, \delta}_{6}}_{18}, \underbrace{1, \ldots, 1}_{18}, \underbrace{\delta, \ldots, \delta}_{18}, \ldots, \underbrace{1, \ldots, 1}_{2\times 3^n}, \underbrace{\delta, \ldots, \delta}_{2\times 3^n}, \ldots), \tag{78}$$
   with $\delta = 1 - \varepsilon$. From the above pattern, it becomes apparent that the corresponding m_k satisfies for all n:
   $$m_{2\times 3^n} = 1 - \varepsilon/2 \qquad\qquad m_{4\times 3^n} = 1 - \varepsilon/4, \tag{79}$$
   hence m_k doesn't converge, yet $\sup_k s_k - \inf_k s_k < \varepsilon$. □

In conclusion, the previous examples hint at that sublinear regret and boundedness are very different properties. Statement (i) shows that sublinear regret is not sufficient for cost boundedness $c_k - \overline{c}_k$. On the contrary, (ii) shows that making the range of the sequence $c_k - \overline{c}_k$ arbitrary small is not sufficient to imply sublinear regret.

Remark 9. The above arguments still holds if we change s_k and m_k to the definitions
$$s'_k := |C(x_k, u_k) - C(x^*_k, u^*_k)| \qquad\qquad m'_k := \frac{1}{k}\sum_{j=0}^{k} |C(x_k, u_k) - C(x^*_k, u^*_k)| = \frac{1}{k}\sum_{j=1}^{k} s'_k$$