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.

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 $\mathcal{G}$ 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 $\mathcal{G}$ a natural performance metric, as $\mathcal{G}$ 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:

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

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

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

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

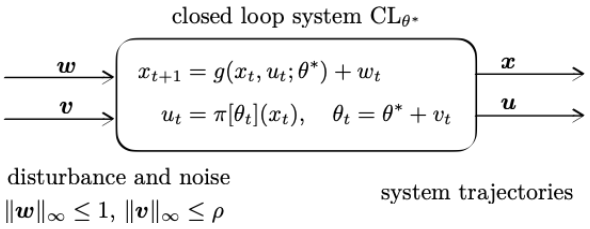

Figure 1: closed loop CL_{θ^*}: An idealized setting in which we have noisy measurements of the true θ^*.

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 $x_{t+1} = g(x_t, u_t; \theta^*) + w_t$, $\|w_t\| \le 1$, 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 $\mathbb{T}[\theta] := \{f^* : t, x, u \mapsto g(x, u; \theta) + w_t \mid \|\boldsymbol{w}\|_\infty \le 1\}$ 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 $\mathsf{P} : \mathbb{D} \mapsto 2^{\mathsf{K}}$

procedure $\texttt{SEL}_{\mathrm{NCBC}}(t, x^+, x, u)$
  $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup (t, x^+, x, u)$
  $\mathsf{S}_t \leftarrow \mathsf{P}(\mathcal{D}_t)$ ▷ construct/update new consistent set
  present set S_t to $\mathcal{A}_{\mathrm{NCBC}}$
  $\mathcal{A}_{\mathrm{NCBC}}$ chooses $\theta_t \in \mathsf{S}_t$
  return θ_t
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:

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

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

Mistake guarantee for $\mathcal{A}_\pi(\texttt{SEL}_s)$ The extension of the results in Theorem A.12, (ii) apply and we obtain for $\mathcal{A}_\pi(\texttt{SEL}_s)$ 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, $\mathcal{A}_\pi(\texttt{SEL}_s)$ 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`.