Proceedings of Machine Learning Research vol 144:1-13, 2021

Generating Adversarial Disturbances for Controller Verification

Udaya Ghai[1]∗ UGHAI@CS.PRINCETON.EDU
David Snyder[2]∗ DASNYDER@PRINCETON.EDU
Anirudha Majumdar[2,3] ANI.MAJUMDAR@PRINCETON.EDU
Elad Hazan[1,3] EHAZAN@CS.PRINCETON.EDU
∗ *Equal contribution*
1 *Department of Computer Science, Princeton University*
2 *Department of Mechanical and Aerospace Engineering, Princeton University*
3 *Google AI Princeton*

Abstract

We consider the problem of generating maximally adversarial disturbances for a given controller assuming only blackbox access to it. We propose an online learning approach to this problem that *adaptively* generates disturbances based on control inputs chosen by the controller. The goal of the disturbance generator is to minimize *regret* versus a benchmark disturbance-generating policy class, i.e., to maximize the cost incurred by the controller as well as possible compared to the best possible disturbance generator *in hindsight* (chosen from a benchmark policy class). In the setting where the dynamics are linear and the costs are quadratic, we formulate our problem as an online trust region (OTR) problem with memory and present a new online learning algorithm (*MOTR*) for this problem. We prove that this method competes with the best disturbance generator in hindsight (chosen from a rich class of benchmark policies that includes linear-dynamical disturbance generating policies). We demonstrate our approach on two simulated examples: (i) synthetically generated linear systems, and (ii) generating wind disturbances for the popular PX4 controller in the AirSim simulator. On these examples, we demonstrate that our approach outperforms several baseline approaches, including H_∞ disturbance generation and gradient-based methods.

Keywords: Adversarial Disturbances, Controller Verification, Online Learning

1. Introduction

We consider the problem of certifying the safety and correct operation of control algorithms in the context of robotics, as understood by a measure of the worst-case system performance in the presence of uncertainty and disturbances. Motivated by this challenge, we consider the following idealized problem.

Consider a control system given by $x_{t+1} = f(x_t, u_t, w_t)$, with state $x \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$, control input $u \in \mathbb{R}^{d_u}$, and disturbance $w \in \mathbb{R}^{d_w}$. Suppose we are provided *blackbox access* to a controller for this system, i.e., we do not have access to the software that defines the controller, but can observe the closed-loop system's behavior by choosing disturbance values. The controller may be arbitrarily complex (e.g., adaptive, nonlinear, stateful, etc.). Our goal is to generate disturbances w_t that *maximize* a specified cost $\sum_{t=0}^{\infty} c(x_t, u_t)$ incurred by the controller.

Statement of Contributions. We present an online learning approach for tackling the problem of generating disturbances for dynamical systems in order to maximize the cost incurred by a controller

Figure 1: Quadrotor in AirSim Mountains Environment. We consider the problem of generating adversarial disturbances (e.g., wind gusts) for a given controller.

assuming only blackbox access to it. The key idea behind our approach is to leverage techniques from online learning (see e.g. Hazan (2019)) to *adaptively* choose disturbances based on control inputs chosen by the controller. Determining the optimal disturbance for a given controller with online blackbox access is computationally infeasible in general. We thus consider *regret minimization* versus a benchmark disturbance-generating policy class. Since our goal is to maximize the cost of the controller, the natural formulation of our problem in online learning is non-convex. Online non-convex optimization does not admit efficient algorithms in general.

To overcome this challenge, we consider the case when the system is linear and the costs are quadratic. In this case we formulate our problem as a special case of non-convex optimization, namely an online trust region (OTR) problem with memory. We then present a new *online trust region with memory* algorithm (*MOTR*) with optimal regret guarantees, which may be of independent interest. Using this technique, we prove that our method competes with the best disturbance-generating policy in hindsight from a reference class. This reference class includes all state-feedback linear-dynamical policies (Def. 3).

We demonstrate our approach on two simulated examples: (i) synthetically generated linear systems, and (ii) generating wind disturbances for the highly-popular PX4 controller (Meier et al., 2015) in the physically-realistic AirSim drone simulator (Shah et al., 2017) (Fig. 1). We compare our approach to several baseline methods, including gradient-based methods and an H_∞ disturbance generator. For linear systems, H_∞ is a Nash equilibrium solution to the offline disturbance problem; however, this does not hold for the case of time-varying costs or when the controller deviates from an H_∞ paradigm. We demonstrate the ability of our method to adaptively generate disturbances that outperform these baselines.

An extended version of this paper is available online (Ghai et al., 2020) and contains proofs and implementation details. References to the Appendix correspond to this extended version.

1.1. Related Work

Regret minimization for online control. There is a large body of work within the control theory literature on synthesizing robust and adaptive controllers (see, e.g., Stengel (1994); Zhou et al. (1996)). The most relevant work for our purposes is online control with low *regret*. In classical control theory, the disturbances are assumed to be i.i.d. Gaussian and the cost functions are known

ahead of time. In the *online* LQR setting (Abbasi-Yadkori and Szepesvári, 2011; Dean et al., 2018; Mania et al., 2019; Cohen et al., 2018), a fully-observed linear dynamic system is driven by i.i.d. Gaussian noise and the learner incurs a quadratic state and input cost. Recent algorithms (Mania et al., 2019; Cohen et al., 2019, 2018) attain $\sqrt{T}$ regret for this online setting, and are able to cope with changing loss functions. Agarwal et al. (2019a) consider the more general and challenging setting of *non-stochastic control* in which the disturbances are adversarially chosen, and the cost functions are arbitrary convex costs. The key insight behind this result is an improper controller parameterization, known as disturbance-action control, coupled with advances in online convex optimization with memory due to Anava et al. (2015). Non-stochastic control was extended to the setting of unknown systems and partial observability (Hazan et al., 2020; Simchowitz et al., 2020).

In contrast to the work mentioned above, we consider the problem of generating adaptive *disturbances* that *maximize* the cumulative cost incurred by a given controller. This shift in problem formulation introduces fundamental technical challenges. In particular, the primary challenge is the *non-convexity* associated with the cost maximization problem. Providing regret guarantees (from the point of view of the disturbance generator) in this non-convex setting constitutes one of they key technical contributions of this work.

Adversarial reinforcement learning. The literature on generating disturbances for control systems is sparse compared to the body of work on synthesizing robust controllers. There has been recent work on generating adversarial policies for agents trained using reinforcement learning (motivated by a long line of work on generating adversarial examples for supervised learning models (Goodfellow et al., 2015)). These results consistently suggest that for high-dimensional problems in RL settings, non-adversarially-trained agents can be directly harmed in training (Behzadan and Munir, 2017) and are highly susceptible to multiple adversarial failure modes (Huang et al., 2017; Gleave et al., 2020). The latter problem motivates Vinitsky et al. (2020) to train agents against an ensemble of adversaries to generate a more robust learned policy. In this vein, Mandlekar et al. (2017) integrates the ideas into a robust training algorithm that allows for noise perturbations in the standard control formulation. However, in contrast to our work, none of the above robust training protocols make theoretical guarantees about the performance of their trained agent, whereas we are able to obtain explicit regret guarantees for the performance of our adversarial agent.

A parallel line of investigation uses sampling-based techniques for probabilistic safety assurance (Sinha et al., 2020a) by actively seeking samples of rare events to estimate the probability of failure modes. This is extended in Sinha et al. (2020b) to include online methods. In particular, they generate models of multiple adversaries (akin to Vinitsky et al. (2020)) and then use online learning to 'decompose' their observed adversary into elements of their set of modeled adversaries, choosing robust actions accordingly. This differs from our work in the optimization paradigm. In particular, they require Monte Carlo sampling of multiple trajectories, repeated over subsequent updates of the environment distribution parameters, in order to obtain guarantees. Our guarantees are 'within-trajectory,' in that we learn and compete with a class of disturbance generators within a single trajectory, rather than by optimizing over many simulations.

Online learning and the trust region problem. We make extensive use of techniques from the field of online learning and regret minimization in games (Cesa-Bianchi and Lugosi, 2006; Hazan et al., 2016). Most relevant to our work is the literature on online non-convex optimization (Agarwal et al., 2019b; Suggala and Netrapalli, 2019), and online convex optimization with memory (Anava et al., 2015). The problem of maximizing a general quadratic function subject to Euclidean norm

constraint is known as the Trust Region (TR) problem, which originated in applying Newton's method to non-convex optimization. Despite the non-convex objective, TR is known to be solvable in polynomial time via a semi-definite relaxation (Ben-Tal and Teboulle, 1996), and also allows for accelerated gradient methods (Hazan and Koren, 2015).

2. Setting and Background

2.1. Notation

For vectors, we use the notation $\|x\|_Q = \sqrt{x^\top Q x}$ for a weighted euclidean norm, where Q is a positive definite matrix. For matrices, we use $\|M\|_F$ to denote the Frobenius norm of matrix M and $\|M\|$ to denote the spectral norm.

2.2. Setting

We consider a nonstochastic linear time-invariant (LTI) system defined by the following equation:

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

where $x_t \in \mathbb{R}^{d_x}$ is the state, $u_t \in \mathbb{R}^{d_u}$ is the control input, and $w_t \in \mathbb{R}^{d_w}$ is the adversarially chosen disturbance. We assume the state and disturbance dimensions are the same and $C = I$ for the remainder of this exposition, but the results still hold in full generality. At time t, a quadratic cost $c_t(\cdot, \cdot)$ is revealed and the controller suffers cost $c_t(x_t, u_t)$. As the disturbances are adversarially chosen, the trajectory, and thus the costs are determined by this. In this model, a disturbance generator $\mathcal{A}$ is a (possibly randomized) mapping from all previous states and actions to a disturbance vector. As the goal is to produce worst-case disturbances, the cost c_t for the controller is a reward for $\mathcal{A}$. The states produced by $\mathcal{A}$ with controls u_t are denoted $x_t^{\mathcal{A}}$ and the total reward is denoted

$$J_T(\mathcal{A}) = \sum_{t=1}^{T} c_t(x_t^{\mathcal{A}}, u_t) = \sum_{t=1}^{T} \|x_t^{\mathcal{A}}\|_{Q_t}^2 + \|u_t\|_{R_t}^2 \ .$$

For a randomized generator, we consider the expected reward. We use a *regret* notion of performance, where the goal is to play a disturbance sequence $\{w_t\}_{t=1}^T$ such that the reward is competitive with reward corresponding to the disturbances played by the best fixed disturbance generator π, chosen in hindsight from a comparator class Π.

$$\text{Regret}_T(\mathcal{A}) = \max_{\pi \in \Pi} J_T(\pi) - J_T(\mathcal{A}).$$

2.3. Comparator class

For our comparator class, we consider a bounded set of Control-disturbance Generators, defined as follows.

Definition 1 *A Control-disturbance Generator (CDG), $\pi(M)$ is specified by parameters $M = (M^{[1]}, \ldots, M^{[H]})$, along with a bias*[1] *w_0, where the disturbance w_t played at state x_t is defined as*

$$w_t = \sum_{i=1}^{H} M^{[i]} u_{t-i} + w_0 \ . \tag{1}$$

1. The bias term is not included in the remainder of the theoretical work for simplicity, but equivalent results can be proved including bias.

Definition 2 *Let* $\Pi_{D,H} = \{\pi(M) : M \in \mathbb{R}^{Hd_x \times d_u}, \|M\|_F \leq D\}$ *be the set of CDGs with history* H *and size* D*. We also use the shorthand* $M \in \Pi_{D,H}$*.*

A CDG is the equivalent of a Disturbance-action controller (DAC) (Agarwal et al., 2019a) where the roles of actions and disturbances have been swapped. It has been shown that DACs can approximate Linear Dynamic Controllers (LDCs), a powerful generalization of linear controllers. As such, we analogously define Linear Dynamic Disturbance Generators (LDDGs), which likewise are approximated by CDGs.

Definition 3 (Linear Dynamic Disturbance Generator) *A* linear dynamic disturbance generator π *is a linear dynamical system* $(A_\pi, B_\pi, C_\pi, D_\pi)$ *with internal state* $s_t \in \mathbb{R}^{d_\pi}$*, input* $x_t \in \mathbb{R}^{d_x}$*, and output* $w_t \in \mathbb{R}^{d_w}$ *that satisfies*

$$s_{t+1} = A_\pi s_t + B_\pi x_t,\ w_t = C_\pi s_t + D_\pi x_t\ .$$

2.4. Assumptions

We make the following assumptions requiring an agent to play bounded controls and requiring bounded system and cost matrices:

Assumption 4 (Bounded controls) *The control sequence is bounded so* $\|u_t\| \leq C_u$*.*

Assumption 5 (Stabilizable Dynamics) *Consider the dynamics tuple* $\{A, B, C, u(x,t)\}$*. We assume that* $A = HLH^{-1}$*, with* $\|L\| \leq 1-\gamma$*, matrix* A *having condition number* $\|H\|\|H^{-1}\| \leq \kappa$ *and* $\|A\|, \|B\|, \|C\| \leq \beta$*.*

Note that if A is not open-loop stable, but the pair (A, B) is stabilizable, there exists a matrix K^* such that $\tilde{A} = A - BK^*$ satisfies the above criterion, and we can equivalently analyze the system $\{\tilde{A}, B, u^*(x,t)\}$, where $u^*(x,t) = u(x,t) + K^*x$. This transformation explains the generality of Assumption 5. Importantly, we do not require knowledge of $u(x,t)$ in this transformation.

Assumption 6 (Bounded costs) *The cost matrices have bounded spectral norm,* $\|Q_t\|, \|R_t\| \leq \xi$*.*

Following Chen and Hazan (2020), we use $\mathcal{L}$ to denote the complexity of the system and comparator class where $\mathcal{L} = d_x + d_u + d_w + D + C_u + \beta + \kappa + \xi + \gamma^{-1}$. Here, d_x, d_u, d_w are the state, action, and disturbance dimensions respectively. C_u bounds the magnitude of controls. The comparator class is $\Pi_{H,D}$ with $H = \lceil \gamma^{-1} \log(\kappa\xi T) \rceil$, in order to capture LDDGs. β and ξ are spectral norm bounds on system matrices and costs respectively. The condition number and decay of dynamics are κ and γ respectively.

3. Online Trust Region With Memory

This section describes our main building block for the adversarial disturbance generator: an online non-convex learning problem called *online trust region (OTR) with memory*. In Sec. 3.1 we provide background on the trust region problem. Subsequently, in Sec. 3.2 we formally introduce the online trust region with memory setting.

3.1. Trust Region Problem

We show that the cost of a CDG can be approximated closely by a nonconvex quadratic. Optimizing the cost of the policy is then a trust region problem, a well-studied quadratic optimization over a Euclidean ball. The interest in this problem stems from the fact that it is one of the most basic non-convex optimization problems that admits "hidden-convexity" — a property that allows efficient algorithms that converge to a global solution.

Definition 7 *A trust region problem is defined by a tuple* (P, p, D) *with* $P \in \mathbb{R}^{d\times d}$, $p \in \mathbb{R}^d$, *and* $D > 0$ *as the following mathematical optimization problem*

$$\max_{\|z\| \le D} \quad z^\top P z + p^\top z.$$

We can define a condition number for a trust region problem as follows.

Definition 8 *The condition number for a trust region instance* (P, p, D) *is* $\kappa = \frac{\lambda}{\mu}$, *where* $\lambda = \max(2(\|P\|_2 + \|p\|_2), D, 1)$ *and* $\mu = \min(D, 1)$.

Note that a trust region problem can be solved in polynomial time by conversion to an equivalent convex optimization problem (see e.g. Ben-Tal and Teboulle (1996)).

Theorem 9 *Let* (P, p, D) *be a trust region problem with condition number* κ. *There exists an algorithm TrustRegion such that TrustRegion*(P, p, D, ϵ) *produces* z_a *with* $\|z_a\| \le D$ *such that*

$$z_a^\top P z_a + p^\top z_a \ \ge \ \max_{\|z\| \le D} z^\top P z + p^\top z - \epsilon \,,$$

and runs in time $\mathrm{poly}(d, \log \kappa, \log \frac{1}{\varepsilon})$.

3.2. Online trust region with memory

Consider the setting of online learning, where an algorithm $\mathcal{A}$ predicts a point z_t with $\|z_t\| \le D$. We use the shorthand, $z_{t:H} = (z_{t-H+1}, \ldots, z_t) \in \mathbb{R}^{dH}$ for the concatenation of the H last points. The algorithm then receives feedback from an adversarially chosen quadratic reward function $f_t : \mathbb{R}^{dH} \to \mathbb{R}$ of the last H decisions, parameterized by $P_t \in \mathbb{R}^{dH\times dH}$ and $p_t \in \mathbb{R}^{dH}$. The reward function is defined as

$$f_t(z') = z'^\top P_t z' + p_t^\top z'. \tag{2}$$

The reward function acting on a single point is also useful, so we define $g_t : \mathbb{R}^d \to \mathbb{R}$ with

$$g_t(z) = f_t(z, \ldots z) = (z, \ldots z)^\top P_t (z, \ldots z) + p_t^\top (z, \ldots z) := z^\top C_t z + d_t^\top z. \tag{3}$$

The reward earned in round t is $f_t(z_{t:H})$. The goal of the online player is to minimize the expected regret, compared to playing the single best point in hindsight:

$$\mathrm{Regret}(\mathcal{A}) = \max_{\|z\| \le D} \sum_{t=H}^{T} g_t(z) - \mathbb{E}[\sum_{t=H}^{T} f_t(z_{t:H})] \,. \tag{4}$$

Here the expectation is over randomness of the algorithm. In App. B, we provide a polynomial-time $O(H^{\frac{3}{2}} T^{\frac{1}{2}})$ regret algorithm for the online trust region with memory. Below is an informal statement of this result (See Thm. 15 for the full result).

Theorem 10 *Suppose elements of matrices P_t and elements of p_t bounded. Alg. 3, with suitable parameterization, will incur expected regret at most*

$$\textit{Regret}(T) \;\leq\; O(D^2 d^{5/2} H^{3/2} \sqrt{T}) \;,$$

and the runtime of the algorithm for each iteration will be $\mathrm{poly}(d, H, \log D, \log T)$.

The algorithm works by applying an extension of nonconvex Follow-the-Perturbed-Leader (FPL) (Agarwal et al., 2019b; Suggala and Netrapalli, 2019) to functions with memory (see App. A). In the OTR with memory setting, the perturbed subproblems that need to be solved in each iteration are trust region problems, so they can be solved in polynomial time.

4. Algorithm and Main Theorem

Our algorithm (*MOTR*; Alg. 1) is an application of Alg. 3 in the Appendix for the OTR with memory problem applied to approximations of the costs of playing a CDG, $\pi(M)$. Let $m = \mathrm{vec}(M)$ be a flattened version of the CDG matrix M. Our approximate cost, $g_t(m) = c_t(y_t(M), u_t)$ is the cost of an approximate state from a truncated rollout starting at $y_{t-H}(M) = 0$, with

$$y_{s+1}(M) = Ay_s(M) + Bu_s + C\big(\sum_{i=1}^{H} M^{[i]} u_{s-i} + w_0\big) \;. \tag{5}$$

Algorithm 1 Memory Online Trust Region (*MOTR*) Generator

Input: Rounds T, system parameters (A, B, C), noise parameter η, history H
Define $u_s = 0$ for $s \leq 0$.
Initialize $M_0 \in \Pi_{D,H}$ randomly.
$S_0 = 0_{Hd_x d_u, Hd_x d_u}, s_0 = 0_{Hd_x d_u}$.
for $t = 0$ **to** T **do**
  Observe quadratic reward c_t and earn $c_t(x_t, u_t) = \|x_t\|^2_{Q_t} + \|u_t\|^2_{R_t}$
  Generate disturbance $w_t = \sum_{i=1}^{H} M_t^{[i]} u_{t-i}$
  Observe control u_t and update state $x_{t+1} = Ax_t + Bu_t + w_t$
  Define $g_t(m) = c_t(y_t(M), u_t)$ where

$$y_{s+1}(M) = \begin{cases} Ay_s(M) + Bu_s + C\big(\sum_{i=1}^{H} M^{[i]} u_{s-i} + w_0\big) & s \geq t-h \\ 0 & \text{otherwise} \end{cases}$$

  Define $S_t = S_{t-1} + (\nabla^2 g_t)(0)$ and $s_t = s_{t-1} + (\nabla g_t)(0)$ [see (13) and (14) in App. D]
  Generate random vector $\sigma_t \in \mathbb{R}^{Hd_x d_u}$ such that $\sigma_{t,i} \sim \mathrm{Exp}(\eta)$
  Update $m_{t+1} \leftarrow \mathrm{TrustRegion}(S_t, s_t - \sigma_t, D, \frac{1}{T})$
  Reshape $M_{t+1} \leftarrow \mathrm{reshape}(m_{t+1}, \mathbb{R}^{Hd_x \times d_u})$
end for

In App. C, we show that g_t is a quadratic function of m, and that y_t is an accurate approximation of x_t due to the stabilizability of the dynamics. In App. D, we combine the regret bound for Alg. 3 in Thm. 15 with the approximation guarantee on y_t from Lem. 19, yielding the following theorem.

Theorem 11 *Suppose Assumptions 4, 5, 6 hold; then Alg. 1 suffers regret at most* $\widetilde{O}(\mathrm{poly}(\mathcal{L})\sqrt{T})$.

5. Experiments

We evaluate the performance of the disturbance generator *MOTR* defined in Alg. 1 across two settings. These consist of (1) general (randomly generated) linear systems of varying modal behavior, and (2) a thirteen-dimensional rigid-body model for a quadrotor drone in the AirSim simulation environment (Shah et al., 2017). To evaluate our algorithm in each setting, we compare its performance with the performance of several baseline disturbance generators, against several different controllers. In each setting, the *MOTR* algorithm is able to outperform the baselines.

5.1. Baseline Generators and Controllers

We compare the *MOTR* algorithm with five baseline generators. Sinusoidal and Gaussian noise are standard within control theory, and form the first two generator classes. A random directional generator (a fixed-norm equivalent of the Gaussian generator) is third. The dynamic game formulation of the H_∞ control problem (Basar and Bernhard, 2008; Bernhard, 1991) yields a Nash equilibrium disturbance generator. The final baseline is a first-order online gradient ascent (OGA) policy, which does not provide theoretical guarantees in this nonconvex setting. OGA produces disturbances via a CDG, with the M learned via gradient ascent on the instantaneous cost.

For the experimental settings in which the true dynamics are linear, the disturbance generators are tested across three controllers: a standard LQR controller, a H_∞ infinite-horizon optimal controller, and an adaptive gradient perturbation controller (GPC) (Agarwal et al., 2019a). In each setting, true, fixed system costs ($\|x\|^2 + \|u\|^2$) were provided to the algorithms. In the AirSim experiment, the two nonlinear controllers tested are the Pixhawk PX4 controller (Meier et al., 2015), which is one of the most popular controllers used by quadrotors in practice, and a pre-tuned PID controller that is defined by the AirSim environment.

5.2. Notes on Implementation

In order to ensure fair comparisons across the baselines, the actions chosen by each disturbance generator except the Gaussian are normalized to ensure that the available disturbance 'budget' does not vary across generators. Thus, the sinusoid and random generators are essentially choosing directions within the state space. The Gaussian generator is specified so that its average disturbance norm will be slightly higher than the norm bound in expectation. The frequency and initial phase vector of each sinusoid generator are optimized offline against the open-loop system dynamics. Further details of the implementation of *MOTR* are deferred to App. E.2

5.3. Experiment 1: General Linear Systems

Here, a randomly generated set of 11 linear systems are tested for each controller-generator pair over 10 initial conditions. For each system, $A \in \mathbb{R}^{4\times4}$, $B, C \in \mathbb{R}^{4\times2}$. We define the cost of a trajectory to be equal to the cumulative average cost over the time horizon. For each controller-generator pair, we average the costs over the 10 initial conditions, and then normalize each generator's average cost for a given controller to lie in the range $[0, 1]$, where a higher value indicates stronger performance. These costs are aggregated across the 11 systems and scaled to the best-performing disturbance generator (for the given controller). The results are shown in Tab. 1.

There are several important points to note. First, against an H_∞ controller, the H_∞ disturbance generator is a Nash equilibrium solution, so it is expected that *MOTR* will recover but not exceed

	LQR	GPC	H_∞
MOTR	$\mathbf{1.000 \pm 0.006}$	$\mathbf{1.000 \pm 0.017}$	0.997 ± 0.038
OGA	0.918 ± 0.128	0.897 ± 0.142	0.998 ± 0.038
H_∞	0.980 ± 0.035	0.949 ± 0.107	$\mathbf{1.000 \pm 0.039}$
Random	0.328 ± 0.097	0.323 ± 0.106	0.552 ± 0.112
Sine	0.564 ± 0.297	0.540 ± 0.301	0.767 ± 0.264
Gaussian	0.444 ± 0.160	0.438 ± 0.174	0.744 ± 0.196

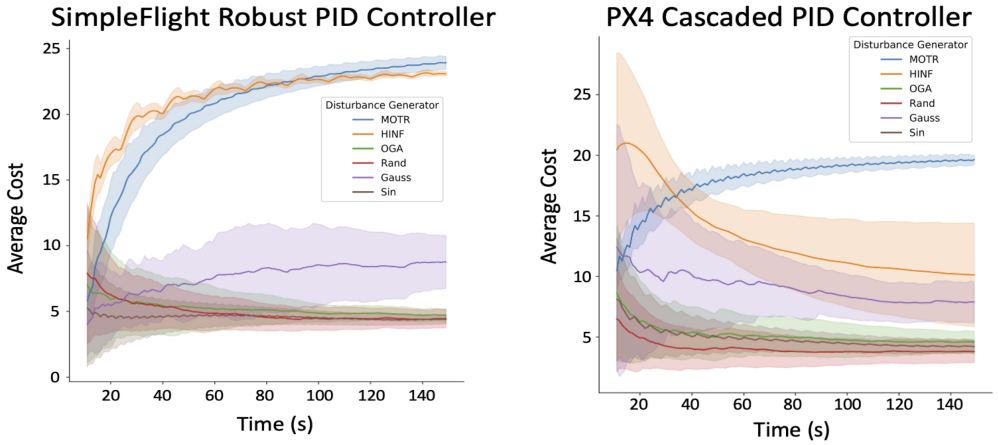
Table 1: Performance results for disturbance generators aggregated over randomized linear systems. The time horizon is $T = 200$, with 11 systems and 10 seeds per system.

that performance. In addition to *MOTR* strongly outperforming the Random, Sinusoid, and Gaussian generators in each setting, we see that against adaptive controllers like GPC, *MOTR* also begins to outperform the H_∞ disturbances. In the presence of model misspecification and cost mismatch, we expect this phenomenon may become more pronounced.

5.4. Experiment 2: Rigid-Body Drone in AirSim

Testing *MOTR* within the AirSim environment allows an empirical test of several key elements of the algorithmic performance, including (1) scaling to higher system dimension, (2) generalizability to nonlinear dynamics about linearized reference conditions, and (3) performance with an accurate but low-dimensional model of the disturbance-to-state transfer function.

The model follows the traditional rigid-body, 6 degree-of-freedom (6DOF) model for air vehicle dynamics, but uses quaternions instead of Euler angles, yielding a 13-dimensional state representation. The nonlinear dynamics are propagated about a nominal hover flight condition. This condition was linearized numerically using a least squares regression procedure on simulator data. There are four inputs, corresponding to the motor commands for each propeller of the quadcopter, and three disturbance channels, corresponding to the North-East-Down (NED) coordinates of the inertial wind vector.



(*a*) (*b*)

Figure 2: (a) Results for the SimpleFlight robust PID controller. Both H_∞ and *MOTR* perform well in this setting. Results are taken over 14 seeds per generator (b) Results for the popular PX4 controller. Note that the non-adaptive H_∞ policy is attenuated, unlike *MOTR*. Further, the first-order online method struggles in this setting. Results are taken over 15 seeds per generator.

The controllers utilized in the simulator include a 'SimpleFlight' AirSim controller and the PX4 autopilot, which is incorporated into AirSim's software-in-the-loop PX4 stack. Each of these controllers is a nonlinear PID controller, and we note that the PX4 is one of the most commonly used autopilots for quadcopter drones.

We present the results of the simulations in Fig. 2. An important feature of the simulations was the presence of a clear best strategy for large disturbances, corresponding to updrafts and downdrafts. However, because the wind magnitudes are constrained, the PX4 controller is able to adaptively attenuate this behavior. As such, the H_∞ generator, while nearly as strong as *MOTR* on the SimpleFlight simulations, suffered against PX4. *MOTR*, however, was able to adapt in the PX4 setting and thus maintain strong disturbance performance.

6. Conclusions

We have studied the problem of generating the worst possible disturbances for a given controller. This is a challenging non-convex problem, which we pose in the framework of online learning. We describe a novel method based on regret minimization in non-convex games with provable guarantees. Our experimental results demonstrate the ability of our approach to outperform various baselines including gradient-based methods and an H_∞ disturbance generator.

This work raises many intriguing questions: can this approach be generalized to dynamics that are unknown, non-linear, partially observable, admit bandit feedback and/or time-varying? Recent results in non-stochastic control suggest the feasibility of these directions (Chen and Hazan, 2020; Gradu et al., 2020a,b; Simchowitz et al., 2020). Another promising direction is to establish lower bounds on regret for the disturbance generation problem and find algorithms that match these lower bounds. Finally, in the vein of adversarial reinforcement learning, the inclusion of adversarial disturbances may prove a useful tool in synthesizing robust learned controllers, as having access to a an adaptive, online disturbance generation mechanism might enhance robustness and regularize worst-case behavior.

Acknowledgments

We thank Karan Singh for enlightening discussion. Elad Hazan is partially supported by NSF award # 1704860 as well as the Google corporation. Anirudha Majumdar was partially supported by the Office of Naval Research [Award Number: N00014-18- 1-2873]. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-2039656. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

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.

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, Alon Gonen, and Elad Hazan. Learning in non-convex games with an optimization oracle. In *Conference on Learning Theory*, pages 18–29, 2019b.

Oren Anava, Elad Hazan, and Shie Mannor. Online learning for adversaries with memory: price of past mistakes. In *Advances in Neural Information Processing Systems*, pages 784–792, 2015.

Tamer Basar and Pierre Bernhard. *H-Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Modern Birkhauser Classics. Birkhauser Basel, 2 edition, 2008. ISBN 978-0-8176-4756-8. doi: 10.1007/978-0-8176-4757-5. URL https://www.springer.com/gp/book/9780817647568.

Vahid Behzadan and Arslan Munir. Vulnerability of Deep Reinforcement Learning to Policy Induction Attacks. *arXiv:1701.04143 [cs]*, January 2017. URL http://arxiv.org/abs/1701.04143. arXiv: 1701.04143 version: 1.

Aharon Ben-Tal and Marc Teboulle. Hidden convexity in some nonconvex quadratically constrained quadratic programming. *Mathematical Programming*, 72(1):51–63, 1996. doi: 10.1007/BF02592331. URL https://doi.org/10.1007/BF02592331.

Pierre Bernhard. A lecture on the game theoretic approach to H-infinity optimal control. 1991. URL http://www-sop.inria.fr/members/Pierre.Bernhard/publications/ber91c.pdf.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Xinyi Chen and Elad Hazan. Black-box control for linear dynamical systems. *ArXiv*, abs/2007.06650, 2020.

Alon Cohen, Avinatan Hassidim, Tomer Koren, Nevena Lazic, Yishay Mansour, and Kunal Talwar. Online Linear Quadratic Control. *arXiv:1806.07104 [cs, stat]*, June 2018. URL http://arxiv.org/abs/1806.07104. arXiv: 1806.07104.

Alon Cohen, Tomer Koren, and Yishay Mansour. Learning linear-quadratic regulators efficiently with only $\sqrt{T}$ regret. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1300–1309, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/cohen19b.html.

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.

Udaya Ghai, David Snyder, Anirudha Majumdar, and Elad Hazan. Generating adversarial disturbances for controller verification, 2020. URL https://irom-lab.princeton.edu/wp-content/uploads/2020/11/AdversarialControlL4DC.pdf.

Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial Policies: Attacking Deep Reinforcement Learning. *arXiv:1905.10615 [cs, stat]*, February 2020. URL http://arxiv.org/abs/1905.10615. arXiv: 1905.10615.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. *arXiv:1412.6572 [cs, stat]*, March 2015. URL http://arxiv.org/abs/1412.6572. arXiv: 1412.6572.

Paula Gradu, John Hallman, and Elad Hazan. Non-Stochastic Control with Bandit Feedback. *arXiv:2008.05523 [cs, math, stat]*, August 2020a. URL http://arxiv.org/abs/2008.05523. arXiv: 2008.05523.

Paula Gradu, Elad Hazan, and Edgar Minasyan. Adaptive Regret for Control of Time-Varying Dynamics. *arXiv:2007.04393 [cs, math, stat]*, July 2020b. URL http://arxiv.org/abs/2007.04393. arXiv: 2007.04393.

Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

Elad Hazan and Tomer Koren. A linear-time algorithm for trust region problems. *Mathematical Programming*, 158(1-2):363–381, Jul 2015. ISSN 1436-4646. doi: 10.1007/s10107-015-0933-y.

Elad Hazan, Sham M. Kakade, and Karan Singh. The Nonstochastic Control Problem. *arXiv:1911.12178 [cs, stat]*, January 2020. URL http://arxiv.org/abs/1911.12178. arXiv: 1911.12178.

Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial Attacks on Neural Network Policies. *arXiv:1702.02284 [cs, stat]*, February 2017. URL http://arxiv.org/abs/1702.02284. arXiv: 1702.02284 version: 1.

Ajay Mandlekar, Yuke Zhu, Animesh Garg, Li Fei-Fei, and Silvio Savarese. Adversarially Robust Policy Learning: Active construction of physically-plausible perturbations. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3932–3939, Vancouver, BC, September 2017. IEEE. ISBN 978-1-5386-2682-5. doi: 10.1109/IROS.2017.8206245.

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

Lorenz Meier, Dominik Honegger, and Marc Pollefeys. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 6235–6240. IEEE, 2015.

Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017. URL https://arxiv.org/abs/1705.05065.

Max Simchowitz, Karan Singh, and Elad Hazan. Improper Learning for Non-Stochastic Control. *arXiv:2001.09254 [cs, math, stat]*, June 2020. URL http://arxiv.org/abs/2001.09254. arXiv: 2001.09254.

Aman Sinha, Matthew O'Kelly, John Duchi, and Russ Tedrake. Neural Bridge Sampling for Evaluating Safety-Critical Autonomous Systems. *arXiv:2008.10581 [cs, stat]*, August 2020a. URL http://arxiv.org/abs/2008.10581. arXiv: 2008.10581.

Aman Sinha, Matthew O'Kelly, Hongrui Zheng, Rahul Mangharam, John Duchi, and Russ Tedrake. FormulaZero: Distributionally Robust Online Adaptation via Offline Population Synthesis. *arXiv:2003.03900 [cs, stat]*, August 2020b. URL http://arxiv.org/abs/2003.03900. arXiv: 2003.03900.

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

Arun Sai Suggala and Praneeth Netrapalli. Online Non-Convex Learning: Following the Perturbed Leader is Optimal. *arXiv:1903.08110 [cs, math, stat]*, September 2019. URL http://arxiv.org/abs/1903.08110. arXiv: 1903.08110.

George Thomas. Identification of transfer functions for wind-induced pressures on prismatic buildings. August 1996. URL https://ttu-ir.tdl.org/handle/2346/20680. Publisher: Texas Tech University.

Eugene Vinitsky, Yuqing Du, Kanaad Parvate, Kathy Jang, Pieter Abbeel, and Alexandre Bayen. Robust Reinforcement Learning using Adversarial Populations. *arXiv:2008.01825 [cs, stat]*, September 2020. URL http://arxiv.org/abs/2008.01825. arXiv: 2008.01825.

Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and Optimal Control*. Prentice-Hall, Inc., USA, 1996. ISBN 0134565673.