2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Change Point Detection Approach for Online Control of Unknown Time Varying Dynamical Systems

Deepan Muthirayan, Ruijie Du, Yanning Shen, and Pramod P. Khargonekar

***Abstract*— We propose a novel change point detection approach for online learning control with full information feedback (state, disturbance, and cost feedback) for unknown time-varying dynamical systems. We show that our algorithm can achieve a sub-linear regret with respect to the class of Disturbance Action Control (DAC) policies, which are a widely studied class of policies for online control of dynamical systems, for any sub-linear number of changes and very general class of systems: (i) matched disturbance system with general convex cost functions, (ii) general system with linear cost functions. Specifically, a (dynamic) regret of $\Gamma_T^{1/5}T^{4/5}$ can be achieved for these class of systems, where Γ_T is the number of changes of the underlying system and T is the duration of the control episode. That is, the change point detection approach achieves a sub-linear regret for any sub-linear number of changes, which other previous algorithms such as in [1] cannot. Numerically, we demonstrate that the change point detection approach is superior to [1] and to standard online learning approaches for time-invariant dynamical systems. Our work presents the first regret guarantee for unknown time-varying dynamical systems in terms of a stronger notion of variability like the number of changes in the underlying system. The extension of the present work to state and output feedback controllers is a subject of future work.**

I. Introduction

In recent years, there has been significant interest in the finite-time performance of learning-based control algorithms for uncertain dynamical systems. Such a control setting is broadly termed as *online control*, borrowing the notion from online learning, where a learner's performance is assessed by their ability to learn from a finite number of samples. The performance in online control is typically measured in terms of regret, which is the loss of performance using the proposed algorithm as compared with the best possible policy. Predominantly, the goal is to design algorithms that adapt to uncertainties arising from disturbances and adversarial cost function so that the regret scales sub-linearly in T, i.e., as T^α with $\alpha < 1$, where T is the duration of the control episode. Significant progress has been made in online control. For example, algorithms have been developed for control of unknown systems, with adversarial cost functions and disturbances [2]–[5], algorithms for known systems with some predictability of future disturbances [6], [7], and for unknown systems with predictability [8].

Control of uncertain systems is an extensively researched theme in control theory. Stochastic control, robust control and adaptive control are large subfields with voluminous literature that address the analysis and synthesis of control for different types of uncertainties. In particular, adaptive control comes closest to "online control" described above. While the primary focus in adaptive control is on closed-loop stability and asymptotic performance, there have been some papers on transient performance. Adaptive control has been studied for systems of all types such as linear, non-linear, and stochastic. There are many variants of adaptive control such as adaptive model predictive control, adaptive learning control, stochastic adaptive control, and robust adaptive control. These variations address the design of adaptive controllers for different variations of the basic adaptive control setting. Thus, adaptive control is a very rich and extensively studied topic. The key differences in the "online control" setting from the classical adaptive control are (a) the consideration of regret as the measure of performance and (b) in some cases the more general nature of the costs, which could be adversarial and/or unknown. Consequently, the classical adaptive control approaches can be inadequate to analyze online control problems. From a techniques point of view, progress in online control is achieved by merging tools from statistical learning, online optimization, and control theory.

A typical assumption in online control is that the system is time-invariant. In many circumstances, however, the underlying system or environment can be time-varying. While some works have studied time-varying dynamical systems [9], [10], they have been limited to quadratic cost functions. Very recently, authors of [1] explored the problem of online control of unknown time-varying linear dynamical systems for generic convex cost functions. Their work presents some impossibility results and a regret guarantee of $\widetilde{\mathcal{O}}\left(|I|\sigma_I + T^{2/3}\right)$ for any interval I, where $|I|$ denotes the length of the interval and σ_I is the square root of the average squared deviation of the system parameters in the interval I. Clearly, in their case [1], the achievability of sub-linear regret is limited to scenarios with number of changes of the underlying system within $o(T^{1/3})$. Motivated by this observation, we investigate the question, *whether sub-linear regret is achievable for any number of changes over the duration T, and under what system, information and cost structures assumptions can we achieve sub-linear guarantees.*

Contribution: Distinct from most of the prior works in online control, which study the control of time invariant dynamical systems, the present paper studies the problem of controlling a time varying dynamical system over a finite time horizon for generic convex cost functions. Specifically, a linear dynamical system with arbitrary disturbances, whose

This work is supported in part by the National Science Foundation under Grant ECCS-1839429 and ECCS-2207457. Deepan Muthirayan, Ruijie Du, Yanning Shen and Pramod P. Khargonekar are with the Department of Electrical Engineering and Computer Sciences, University of California Irvine, Irvine, CA (emails: deepan.m@uci.edu, ruijied@uci.edu, yannings@uci.edu, pramod.khargonekar@uci.edu).

system matrices can be time varying is considered. For such systems, we address the question of how to learn online and optimize when the system matrices are unknown, in addition to the cost functions and disturbances being arbitrary and unknown a priori. The goal is to design *algorithms with regret guarantees in terms of stronger notions of variability (compared to σ_I), such as the number of changes*. Towards this end, we consider the full information feedback structure, where in addition to the cost and state feedback at the end of a time step, the controller also receives disturbance as a feedback. We specifically consider the regret with respect to the class of Disturbance Action Control (DAC) policies [1], which are a widely used class of policies for online control of dynamical systems.

We propose a novel change point detection-based online control algorithm for unknown time-varying dynamical systems. We present guarantees for very general class of systems: (i) matched disturbance system with general convex cost functions, (ii) general system with linear cost functions. We show that, in both these settings, a (dynamic) regret of $\widetilde{\mathcal{O}}\left(\Gamma_T^{1/5}T^{4/5}\right)$ is achievable with a high probability, where Γ_T is the number of times the system changes in T time steps and T is the duration of the control episode. Through numerical simulations, we demonstrate that the change point detection approach is superior to a standard restart approach, the adaptive algorithm of [1], and also standard online learning approach for time-invariant dynamical systems such as [5]. Our result guarantees sub-linear regret for any sub-linear number of changes, which is an improvement over [1]. Our work presents the first regret guarantee in terms of a stronger notion of variability like the number of changes in the underlying system. The extension of our work to the setting without disturbance feedback is a subject of future work.

Notation: We denote the spectral radius of a matrix A by $\rho(A)$, the discrete time interval from m_1 to m_2 by $[m_1, m_2]$, and the sequence $(x_{m_1}, x_{m_1+1}, ..., x_{m_2})$ compactly by $x_{m_1:m_2}$. Unless otherwise specified, $\|\cdot\|$ is the 2-norm of a vector and the Frobenious norm of a matrix. We use $\mathcal{O}(\cdot)$ for the standard order notation, and $\widetilde{\mathcal{O}}(\cdot)$ denotes the order neglecting the poly-log terms in T. We denote the inner product of two vectors x and y by $\langle x, y\rangle$.

II. Problem Formulation

We consider the online control of a general linear time-varying dynamical system. Let t denote the time index, x_t, the state of the system, y_t, the output of the system that is to be controlled, u_t, the control input, w_t and e_t, the disturbance and measurement noise, and $\theta_t = [A_t, B_t]$, the time-varying system matrices. Then, the equation governing the dynamical system is given by

$$\begin{aligned} x_{t+1} &= A_t x_t + B_t u_t + B_{t,w} w_t, \\ y_t &= C_t x_t + e_t. \end{aligned} \tag{1}$$

Let $w_t \in \mathbb{R}^q$, $e_t \in \mathbb{R}^p$, $x_t \in \mathbb{R}^n$, $y_t \in \mathbb{R}^p$, and $u_t \in \mathbb{R}^m$. We assume that the sequence of system parameters $\theta_{1:T}$ is unknown to the controller. The disturbance w_t could arise from unmodeled dynamics and thus need not be stochastic. For generality, we assume that the disturbances and measurement noise are bounded and arbitrary. We denote the total duration of the control episode by T.

Like in any control problem, at any time t, the controller incurs a cost $c_t(y_t, u_t)$, which is a function of the output and the control input. In addition to the system parameters being unknown, the sequence of cost functions $c_{1:T}$ and the disturbances $w_{1:T}$ for the duration T is arbitrary and unknown a priori. We assume that the full cost function $c_t(\cdot,\cdot)$ and the disturbance w_t are revealed to the controller after its action at t. Such a feedback is typical in online control and optimization and is termed the full information feedback. The difference here compared to a standard online control formulation is the feedback of the disturbance w_t. Thus, a control policy has the following information by any time t: (i) the cost functions and the disturbances until $t-1$, $c_{1:t-1}$ and $w_{1:t-1}$, (ii) the control inputs until $t-1$, $u_{1:t-1}$, and (iii) the observations until t, $y_{1:t}$. Let Π_I denote the set of policies that satisfy this information setting.

We denote a control policy by π. The state, output, and the control input under the policy is denoted by x_t^π, y_t^π and u_t^π respectively. Given that the cost functions and disturbances are only revealed incrementally, one step at a time, the control policy will have to be adapted online as and when the controller gathers information to achieve the best performance over a period of time. Like in a standard online control problem, we characterize the performance of a control policy over a finite time by its *regret*. We denote the regret of a policy π over a duration T with respect to a policy class $\Pi_{\mathcal{M}} \in \Pi_I$ by $R_T(\pi)$:

$$R_T(\pi) = \underbrace{\sum_{t=1}^{T} c_t(y_t^\pi, u_t^\pi)}_{\text{Policy Cost}} - \underbrace{\min_{\kappa \in \Pi_{\mathcal{M}}} \sum_{t=1}^{T} c_t(y_t^\kappa, u_t^\kappa)}_{\text{comparator cost}}. \tag{2}$$

The primary goal is to design a control policy that minimizes the regret for the stated control problem. Since the regret minimization problem is typically hard, a typical goal is to design a policy that achieves sub-linear regret, i.e., a regret that scales as T^α with T, with a $\alpha < 1$ that is minimal. Such a regret scaling implies that the realized costs converge to that of the best policy from the comparator class asymptotically. Our objective is to design an adaptive policy that can track time variations and achieve sub-linear regret. We note that the regret defined above is static regret. Later, we present the extension to dynamic regret, which is a notion that is more suitable for time-varying dynamical systems.

The comparator class we consider is the class of Disturbance Action Control (DAC) policies (see [1]). A Disturbance Action Control (DAC) policy is defined as the linear feedback of the disturbances up to a certain history h. Let's denote a DAC policy by π_{DAC}. Then, the control input $u_t^{\pi_{\text{DAC}}}$

under policy π_{DAC} is given by

$$u_t^{\pi_{\mathrm{DAC}}} = \sum_{k=1}^{h} M_t^{[k]} w_{t-k}. \tag{3}$$

Here, $M_t = \left[M_t^{[1]}, \ldots, M_t^{[h]}\right]$ are the feedback gains or the disturbance gains and are the (time-varying) parameters of π_{DAC}. Here, we note that, π_{DAC} can be dynamic, i.e., it's parameters can be varying with time. Therefore, the regret defined in Eq. (2) is the notion of dynamic regret. We note that the policy is implementable with disturbance feedback. An extension to the case without the disturbance feedback can be made by using estimates of the disturbances instead. We defer the treatment without any disturbance feedback to future work. Our objective here is to optimize the parameter M online so that the regret with respect to the best DAC policy in hindsight is sub-linear.

The DAC policy is typically used in online control for regulating systems with disturbances; see [4]. The important feature of the DAC policy is that the optimization problem to find the optimal fixed disturbance gain for a given sequence of cost functions is a convex problem and is thus amenable to online optimization and online performance analysis. A very appealing feature of DAC is that, for time-invariant systems, the optimal disturbance action control for a given sequence of cost functions is very close in terms of the performance to the optimal linear feedback controller of the state; see [4]. Thus, for time-invariant systems, by optimizing the DAC online, it is possible to achieve a sub-linear regret with respect to the best linear feedback controller of the state, whose computation is a non-convex optimization problem.

For time-varying dynamical systems, as pointed out in [1, Theorem 2.1], there exist problem instances where the DAC class (with disturbance feedback) incurs a much better cost than other types of classes such as linear state or output feedback policies and vice versa. Therefore, the DAC class is not a weaker class to compete against compared to these standard classes. Moreover, as pointed out by the impossibility result [1, Theorem 3.1], it is an equally harder class to compete against in terms of regret. In this work, we focus our study on the regret minimization problem with respect to the DAC class (with disturbance feedback) and defer the treatment of other control structures to future work.

Even with the disturbance feedback, the challenge of estimating the unknown system parameters does not diminish. This is because of the presence of measurement noise and the variations itself. In the time-invariant case, following an analysis similar to [5], it can be shown that, even with disturbance feedback, only a regret of $T^{2/3}$ can be achieved with the state-of-the-art methods, which is not any better than the regret that can be achieved without disturbance feedback (see [5]). The same holds for the time-varying case. It can be shown that, what [1] can achieve for the system in Eq. (1), even with disturbance feedback, cannot be improved. Therefore, the conclusions we draw later on comparing the bounds we derive and the regret upper bound of [1] are valid. We state our other assumptions below.

Assumption 1 (System)**.** *(i) The system is stable, i.e.,* $\|C_{t+k+1}A_{t+k}\ldots A_{t+1}B_t\|_2 \leq \kappa_a\kappa_b(1-\gamma)^k,\ \forall\, k \geq 0,\ \forall\, t$, *where* $\kappa_a > 0, \kappa_b > 0$ *and* γ *is such that* $0 < \gamma < 1$, *and where* κ_a, κ_b *and* γ *are constants.* B_t *is bounded, i.e.,* $\|B_t\| \leq \kappa_b$*. (ii) The disturbance and noise* w_t *and* e_t *is bounded. Specifically,* $\|w_t\| \leq \kappa_w$*, where* $\kappa_w > 0$ *is a constant, and* $\|e_t\| \leq \kappa_e$*, where* $\kappa_e > 0$ *is a constant.*

Assumption 2 (Cost Functions)**.** *(i) The cost function* c_t *is convex* $\forall$ *t. (ii)* $\|c_t(x,u) - c_t(x',u')\| \leq LR\|z - z'\|$ *for a given* $z^\top := [x^\top, u^\top], (z')^\top := [(x')^\top, (u')^\top]$*, where* $R := \max\{\|z\|, \|z'\|, 1\}$ *and* L *is a constant. (iii) For any* $d > 0$*, when* $\|x\| \leq d$ *and* $\|u\| \leq d$*,* $\nabla_x c(x,u) \leq Gd, \nabla_u c(x,u) \leq Gd$*, where* G *is a constant.*

Remark 1 (System Assumptions). Assumption 1.(i) is the equivalent of stability assumption used in time invariant systems. Such an assumption is typically used in online control when the system is unknown; see for eg., [1], [5]. Assumption 1.(iii) that noise is bounded is necessary, especially in the non-stochastic setting [4], [5]. The assumption on cost functions is also standard [4].

Definition 1. *(i)* $\mathcal{M} := \{M = (M^{[1]}, \ldots, M^{[h]}) : \|M^{[k]}\| \leq \kappa_M\}$ *(Disturbance Action Policy Class). (ii)* $\mathcal{G} = \{G^{[1:h]} : \|G^{[k]}\|_2 \leq \kappa_a\kappa_b(1-\gamma)^{k-1}\}$*. (iii) Setting (S-1): Matched disturbance system with convex cost functions:* $B_t = B_{t,w}, C = I, e_t = 0$*. Setting (S-2): General system with linear cost functions:* $B_{t,w} = I$*, and there exists a coefficient* $\alpha_t \in \mathbb{R}^{p+m}$ *such that* $c_t(y,u) = \alpha_t^\top z$*,* $\|\alpha_t^\top\| \leq G$*.*

III. Online Learning Control Algorithm

Typically, online learning control algorithms for time-invariant dynamical systems explore first for a period of time, and then exploit, i.e., adapt or optimize the control policy. While, in the time-invariant case, this strategy results in sub-linear regret, in the time-varying case, it can be less effective. For instance, consider the case where the system remains unchanged for the duration of the exploration phase and then changes around the instant when the exploration ends. Clearly, in this case, the estimate made at the end of the exploration phase will be very distant from the underlying system parameter realized after the exploration phase and therefore not result in a sub-linear regret.

We propose an online algorithm that *continuously learns to compute an estimate of the time varying system parameters and that simultaneously optimizes the control policy online*. Our estimation algorithm combines (i) a change point detection algorithm to detect the changes in the underlying system and (ii) a regular estimation algorithm. The online algorithm runs an online optimization parallel to the estimation to optimize the parameters of the control policy, which in our case is a DAC policy.

Online Optimization: Since the cost functions and the disturbances are unknown a priori, the optimal parameter M of the DAC policy cannot be computed a priori. Rather, the parameters have to be adapted online continuously with the information gathered along the way to achieve the best

performance. Given the convexity of the cost functions and the linearity of the system dynamics, we can apply the Online Convex Optimization (OCO) framework to optimize the policy parameters online.

We call a policy that learns the DAC policy parameters online as an online DAC policy. We formally denote such a policy by $\pi_{\text{DAC}-\text{O}}$. Let the parameters estimated by $\pi_{\text{DAC}-\text{O}}$ be denoted by $M_t = \left[M_t^{[1]}, \ldots, M_t^{[h]}\right]$. Given that the parameter M_t is continuously updated, the control input $u_t^{\pi_{\text{DAC}-\text{O}}}$ can be computed by,

$$u_t^{\pi_{\text{DAC}-\text{O}}} = \sum_{k=1}^{h} M_t^{[k]} w_{t-k}. \tag{4}$$

Given that the realized cost is dependent on the past control inputs, we will have to employ an extension of the OCO framework called Online Convex Optimization with Memory (OCO-M) to optimize the parameters of the DAC policy.

For the benefit of the readers, we briefly review the online convex optimization (OCO) setting (see [11]). OCO is a game played between a player who is learning to minimize its overall cost and an adversary who is attempting to maximize the cost incurred by the player. At any time t, the player chooses a decision M_t from some convex subset $\mathcal{M}$ given by $\max_{M\in\mathcal{M}}\|M\| \leq \kappa_M$, and the adversary chooses a convex cost function $f_t(\cdot)$. As a result, the player incurs a cost $f_t(M_t)$ for its decision M_t. The goal of the player is to minimize the regret over a duration T, given by

$$R_T = \sum_{t=1}^{T} f_t(M_t) - \min_{M\in\mathcal{M}} \sum_{t=1}^{T} f_t(M).$$

The challenge is that the player does not know the cost function that the adversary will pick. Once the adversary picks a cost function, the player observes the realized cost and in some cases can also observe the full cost function. The objective of the learner is to achieve the minimal regret or at the least a sub-linear regret. We direct the readers to [11] for a more detailed exposition and the various algorithmic approaches for this problem.

The difference in the OCO-M setting is that the cost functions can be dependent on the history of past decisions up to a certain time. Let the length of the history dependence be denoted by h. The regret in the OCO-M problem is then given by

$$R_T = \sum_{t=1}^{T} f_t(M_{t-h:t}) - \min_{M\in\mathcal{M}} \sum_{t=1}^{T} f_t(M).$$

One limitation of the OCO-M framework is that it can only be applied when the length h is fixed or bounded above. In a control setting though, the cost is typically a function of the state or the output, which is dependent on the full history of decisions $M_{1:t}$, the length of which grows unbounded with the duration of the control episode. Let

$$G_t = [G_t^{[1]}, G_t^{[2]}, \ldots, G_t^{[h]}], \widetilde{G}_t = [\widetilde{G}_t^{[1]}, \widetilde{G}_t^{[2]}, \ldots, \widetilde{G}_t^{[t-1]}],$$
$$\widetilde{G}_t^{[k]} = C_t A_{t-1} \ldots A_{t-k+2} A_{t-k+1},\ \forall\ k \geq 2,\ \widetilde{G}_t^{[1]} = C_t,$$
$$G_t^{[k]} = C_t A_{t-1} \ldots A_{t-k+2} A_{t-k+1} B_{t-k},\ \forall\ k \geq 2,$$

and $G_t^{[1]} = C_t B_{t-1}$. Thus, the history of dependence increases with t and is not fixed. In order to apply the OCO-M framework, typically, a truncated output $\tilde{y}_t$ is constructed, whose dependence on the history of control inputs is limited to h time steps:

$$\tilde{y}_t^{\pi_{\text{DAC}-\text{O}}}[M_{t:t-h}|G_t, s_{1:t}] = s_t + \sum_{k=1}^{h} G_t^{[k]} u_{t-k}^{\pi_{\text{DAC}-\text{O}}},$$
$$\text{where } s_t = y_t - \sum_{k=1}^{t-1} G_t^{[k]} u_{t-k}^{\pi_{\text{DAC}-\text{O}}}.$$

Using the truncated output, a truncated cost function $\tilde{c}_t$ is constructed as

$$\tilde{c}_t(M_{t:t-h}|G_t, s_{1:t}) = c_t(\tilde{y}_t^{\pi_{\text{DAC}-\text{O}}}[M_{t:t-h}|G_t, s_{1:t}], u_t^{\pi_{\text{DAC}-\text{O}}}).$$

We denote the function $\tilde{c}_t(M_{t:t-h}|G_t, s_{1:t})$ succinctly by $\tilde{c}_t(M|G_t, s_{1:t})$ when each M_k in $M_{t:t-h}$ is equal to M. This denotes the (truncated) cost that would have been incurred had the policy parameter been fixed to M at all the past h time steps.

A standard gradient algorithm for OCO-M framework updates the decision M_t by the gradient of the function $f_t(M_{t:t-h})$ with all M_k in $M_{t:t-h}$ fixed to M_t. Using the same compact notation as above, this gradient is equal to $\partial f_t(M_t)$. An interpretation of this gradient is that, it is the gradient of the cost that would have been incurred had the policy parameter been fixed at M_t the past h time steps. We employ the same idea to update the policy parameters of the DAC policy online. The online optimization algorithm we propose updates the policy parameter M_t by the gradient of the cost function $\tilde{c}_t(M_t|G_t, s_{1:t})$ where each M_k in $M_{t:t-h}$ is fixed to M_t, i.e., as

$$M_{t+1} = \text{Proj}_{\mathcal{M}}\left(M_t - \eta \frac{\partial \tilde{c}_t(M_t|G_t, s_{1:t})}{\partial M_t}\right), \tag{5}$$

where $\mathcal{M}$ is a convex set of policy parameters.

Definition 2 (Disturbance Action Policy Class)**.** $\mathcal{M} := \{M = (M^{[1]}, \ldots, M^{[h]}) : \|M^{[k]}\| \leq \kappa_M\}$

Optimization for Dynamic Regret: The online optimization procedure described above can only fetch a sub-linear regret for static regret. To fetch a sub-linear dynamic regret, multiple online optimizers like in Eq. (5) are required to be run parallelly as in [12]. Let's index the parallel learners by i and let the parameters corresponding to the learner i be $M_{t,i}$. Just as in [12], the final parameter M_t is computed by $M_t = \sum_{i=1}^{H} p_{t,i} M_{t,i}$, where $p_{t,i}$ are a set of weights such that $\sum_{i=1}^{N} p_{t,i} = 1$ and $p_{t,i}$ are also updated online along with $M_{t,i}$s. Specifically, $p_{t,i}$ is updated by $p_{t+1,i} \propto p_{t,i} e^{-l_{t,i}(M_{t,i})}$, where $l_{t,i}(M) = \zeta\|M_{t,i} - M_{t-1,i}\| + \langle M_{t,i}, \partial\tilde{c}_t(M_t|G_t, s_{1:t})\rangle$. The $M_{t,i}$s are updated

by

$$M_{t+1,i} = \text{Proj}_{\mathcal{M}}\left(M_{t,i} - \eta_i \frac{\partial \tilde{c}_t(M_t|G_t, s_{1:t})}{\partial M_t}\right), \quad (6)$$

The complete online optimization algorithm is given in Algorithm 1.

Algorithm 1 Online Learning Control with Full Knowledge (OLC-FK) Algorithm [12, scream.control]

Input: ζ, H, Step sizes η_is, parameters $\theta_{1:T}$.
Initialize $M_{1,i} \in \mathcal{M}$ arbitrarily for all $i \in [1, H]$
Initialize $p_{1,i} \propto 1/(i^2 + i)$ for all $i \in [1, H]$
for *t = 1,...,T* **do**
  Apply $u_t^{\pi_{\text{DAC-O}}} = \sum_{k=1}^{h} M_t^{[k]} w_{t-k}$
  Observe c_t, w_t and incur cost $c_t(y_t^{\pi_{\text{DAC-O}}}, u_t^{\pi_{\text{DAC-O}}})$
  Compute: $l_{t,i} = \zeta \|M_{t,i} - M_{t-1,i}\| + \langle M_{t,i}, \partial \tilde{c}_t(M_t|G_t, s_{1:t})\rangle$ for all $i \in [1, H]$
  Update: $p_{t+1,i} \propto p_{t,i} e^{-l_{t,i}}$ for all $i \in [1, H]$
  Update: $M_{t+1,i} = \text{Proj}_{\mathcal{M}}\left(M_{t,i} - \eta_i \frac{\partial \tilde{c}_t(M_t|G_t, s_{1:t})}{\partial M_t}\right)$
end

Main Result: We state the performance of the algorithm OLC-FK formally below.

Theorem 1 (Full System Knowledge)**.** *Suppose the setting is the general setting S-2, and the cost functions are general convex functions. Then, under Algorithm 1 [12, scream.control], with* $h = \frac{\log T}{(\log(1/1-\gamma))}$, $H = \mathcal{O}(\log(T))$, $\zeta = \mathcal{O}(h^2)$, *and* $\eta_i = \mathcal{O}(2^{i-1}/\sqrt{\zeta T})$, *the regret with respect to any DAC policy* $M^{\star}_{1:T}$,

$$R_T \leq \widetilde{\mathcal{O}}\left(\sqrt{T(1+P_T)}\right), \quad (7)$$

where P_T *is the path length of the sequence* $M^{\star}_{1:T}$.

The proof follows from a standard proof for online optimization. Please see Arxiv version for the full proof.

A. Disturbance Action Control without System Knowledge

In the previous case, where the system parameters are known, the control policy parameters are optimized online through the truncated cost $\tilde{c}_t(\cdot)$, whose construction explicitly utilizes the knowledge of the underlying system parameters $G_t^{[k]}$. In this case, since the underlying system parameters are not available, we construct an estimate of the truncated state and the truncated cost by estimating the underlying system parameters $G_t^{[k]}$s. With this approach, the control policy will have to solve an online estimation problem to compute an estimate of the system parameters. Since the parameters are time-variant, the online estimation has to be run throughout, unlike the other online estimation approaches [5], [8], along with the policy optimization. Below, we describe in detail how our algorithm simultaneously performs estimation and optimizes the control policy.

Online Estimation and Optimization: The Online Learning Control with Zero Knowledge (OLC-ZK) of the system parameters has two components: (i) *a control policy* and (ii) *an online estimator* that runs in parallel to the control policy and throughout the control episode. The control policy and online optimization algorithm is similar to the online algorithm 1, except that the control policy parameters are updated through an estimate of the truncated cost function. The online estimation algorithm employs a change point detection to identify the changes in the underlying system and a standard estimation algorithm to estimate the underlying system that is restarted after every detection of change. We discuss the details of our algorithm below.

A. Online Control Policy: We use the same notation for the control policy and the control input, i.e., $\pi_{\text{DAC-O}}$ and $u_t^{\pi_{\text{DAC-O}}}$ respectively. The estimation algorithm constructs an estimate $\widehat{G}_t^{[k]}$ of the parameters $G_t^{[k]}$ of the system in Eq. (1) for $k \in [1, h]$. Thus, the estimation algorithm estimates $G_t^{[k]}$s only for a truncated time horizon (looking backwards), i.e., for $k \in [1, h]$. We describe the estimation algorithm later.

The policy $\pi_{\text{DAC-O}}$ computes the control input $u_t^{\pi_{\text{DAC-O}}}$ (zero knowledge case) by combining two terms: (i) *disturbance action control* just as in the full knowledge case and (ii) a *perturbation* for exploration. In this case, we require an additional perturbation, just as in [2], so as to be able to run the estimation parallel to the Online DAC, the control for regulating the cost. Let $\tilde{u}_t^{\pi_{\text{DAC-O}}}[M_t|w_{1:t}] = \sum_{k=1}^{h} M_t^{[k]} w_{t-k}$. Therefore, the total control input by $\pi_{\text{DAC-O}}$ is given by

$$u_t^{\pi_{\text{DAC-O}}} = \underbrace{\tilde{u}_t^{\pi_{\text{DAC-O}}}[M_t|w_{1:t}]}_{\text{DAC}} + \underbrace{\delta u_t^{\pi_{\text{DAC-O}}}}_{\text{Perturbation}}. \quad (8)$$

As in [2], we apply a Gaussian random variable as the perturbation, i.e.,

$$\delta u_t^{\pi_{\text{DAC-O}}} \sim \mathcal{N}(0, \sigma^2 I), \quad (9)$$

where σ denotes the standard deviation, and is a constant to be specified later.

In this case the policy parameters are optimized by applying OCO-M on an estimate of the truncated cost. To construct this estimate, we construct an estimate of s_t and the truncated state $\tilde{x}_t^{\pi_{\text{DAC-O}}}(\cdot)$. Given that s_t is the state response when the control inputs are zero, we estimate s_t by subtracting the contribution of the control inputs from the observed state:

$$\hat{s}_t = \sum_{k=1}^{h} \widehat{G}_t^{[k]} w_{t-k} \text{(S-1)}$$

$$\hat{s}_t = y_t^{\pi_{\text{DAC-O}}} - \sum_{k=1}^{h} \widehat{G}_t^{[k]} u_{t-k}^{\pi_{\text{DAC-O}}} \text{(S-2)}. \quad (10)$$

Then the estimate of the truncated output follows by substituting $\hat{s}_t$ in place s_t and using the estimated $\widehat{G}_t$ in place G_t:

$$\tilde{\tilde{y}}_t^{\pi_{\text{DAC-O}}}[M_{t:t-h}|\widehat{G}_t, \hat{s}_{1:t}] = \hat{s}_t + \sum_{k=1}^{h} \widehat{G}_t^{[k]} \tilde{u}_{t-k}^{\pi_{\text{DAC-O}}}. \quad (11)$$

Then, the estimate of the truncated cost is calculated as

$$\tilde{c}_t(M_{t:t-h}|\widehat{G}_t, \hat{s}_{1:t}) = c_t(\tilde{\tilde{y}}_t^{\pi_{\text{DAC-O}}}[M_{t:t-h}|\widehat{G}_t, \hat{s}_{1:t}], \tilde{u}_t^{\pi_{\text{DAC-O}}}).$$

The online update to the policy parameters is just as in Algorithm 1, i.e., by the gradient of the estimate of the truncated cost

$$M_{t+1,i} = \text{Proj}_{\mathcal{M}}\left(M_{t,i} - \eta_i \frac{\partial \tilde{c}_t(M_t|\widehat{G}_t, \hat{s}_{1:t})}{\partial M_t}\right). \quad (12)$$

A. Online Estimation: The online estimation algorithm is a combination of a change point detection algorithm and a regular estimation algorithm. The change point detection algorithm detects changes larger than a certain threshold and resets the estimation algorithm upon every detection. The estimation algorithm continuously updates the estimates using all of the data from the last reset point. This offers the online learner more flexibility as it only resets whenever there is a significant underlying change, while it continues to refine the estimate otherwise. Thus, the change point detection approach can track the time variations more optimally. This is observed to be the case in the numerical simulations.

A.1. Change Point Detection: The goal of the Change Point Detection (CPD) algorithm is to detect the underlying changes in the system reliably. To do this, the CPD algorithm runs a sequence of independent estimation algorithms each of duration $t_p = N+h$ one after the other, where the estimation algorithms are the standard least-squares estimation applied to the data collected from the respective periods of duration $t_p = N + h$. Here, t_p has to be necessarily greater than h, since computing the estimate of G_t requires at least a length of h inputs. Essentially, the CPD algorithm ignores the past and only considers the recent history to compute an estimate of the system parameters. This allows the CPD algorithm to compute a reliable estimate of the current values of the parameters of the system provided N is of adequate size and at the same time not very large. Then, provided the estimation in each period of duration N is an accurate estimate of the system parameter values in the respective periods, any change point can be detected by comparing the estimates across the different periods. More specifically, if the estimate at the end of a period is greater than a certain threshold compared to the estimate from an earlier period, we can proclaim change point detection.

We denote the index of the successive periods of duration t_p by k. We denote the start and end time of each of these periods by t_s^k and t_e^k. Therefore, it follows that $t_s^k = t_e^{k-1}$ for all k. The CPD algorithm computes the following least-squares estimate at the end of each period k

$$\widehat{G}_k^{\text{cd}} = \arg\min_{\widehat{G}} \sum_{p=t_s^k+h}^{t_e^k} \ell_p\left(\widehat{G}\right) + \lambda\|\widehat{G}\|^2,$$

$$\lambda > 0,\ \ell_p\left(\widehat{G}\right) = \|y_p^{\pi_{\text{DAC-O}}} - \sum_{l=1}^{h} \widehat{G}^{[l]} \delta u_{p-l}^{\pi_{\text{DAC-O}}}\|^2. \quad (13)$$

We denote the first period of duration t_p, after a detection, as the baseline period with index $k = 1$. By default, the very first period of duration t_p at the beginning of the control episode is also a period with index $k = 1$. The CPD algorithm **proclaims change point detection, when at the end of a period** k

$$\|\widehat{G}_k^{\text{cd}} - \widehat{G}_\ell^{\text{cd}}\|_2 > \frac{2\beta}{\sigma\sqrt{N}}, \quad \text{for any } \ \ell \ \text{ s.t. } \ 1 \le \ell < k.$$

where β is a constant to be defined later.

Algorithm 2 Online Learning Control with Change Point Detection (OLC-ZK-CPD) Algorithm

Input: Step sizes $\eta_{1:H}$, $H, \zeta, \sigma, \beta, N, h$
Initialize $M_{1,i} \in \mathcal{M}$ arbitrarily $\forall\ i \in [1, H]$, $t_d = 1, k = 1, t_s = 1, t_e = N + h$.
Initialize $p_{1,i} \propto 1/(i^2 + i)$ for all $i \in [1, H]$.
for t = *1,...,T* **do**
  Observe $y_t^{\pi_{\text{DAC-O}}}$.
  if $t == t_e$ **then**
    Estimate $\widehat{G}_k^{\text{cd}}$ according to Eq. (13).
    if $k > 1$ **then**
      if $\|\widehat{G}_k^{\text{cd}} - \widehat{G}_\ell^{\text{cd}}\|_2 > \frac{(2)\beta}{\sigma\sqrt{N}}$ *for any* $1 \le \ell < k$ **then**
        Proclaim change point detection. Set $t_d = t$. Set $k = 1$.
      else
        $k = k + 1$.
      end
    else
      $k = k + 1$.
    end
    $t_s = t_e, \quad t_e = t_s + N + h - 1$.
  end
  Compute $\widehat{G}_t$ according to Eq. (14).
  Apply $u_t^{\pi_{\text{DAC-O}}}$ from Eq. (8).
  Observe c_t, w_t and incur cost $c_t(y_t^{\pi_{\text{DAC-O}}}, u_t^{\pi_{\text{DAC-O}}})$.
  Compute: $l_{t,i} = \zeta\|M_{t,i} - M_{t-1,i}\| + \langle M_{t,i}, \partial\tilde{c}_t(M_t|\widehat{G}_t, \hat{s}_{1:t})\rangle$ for all $i \in [1, H]$
  Update: $p_{t+1,i} \propto p_{t,i} e^{-l_{t,i}}$ for all $i \in [1, H]$
  Update: $M_{t+1,i} = \text{Proj}_{\mathcal{M}}\left(M_{t,i} - \eta_i \frac{\partial\tilde{c}_t(M_t|\widehat{G}_t, \hat{s}_{1:t})}{\partial M_t}\right)$.
end

A.2. System Estimation: Upon detection of a change by the CPD algorithm, the online estimation algorithm restarts the estimation of the system parameters after a delay of h. Let t_d denote the most recent time of detection by the CPD algorithm. Then, the estimate of the system parameters for any time $t \ge t_d + 2h$ is given by

$$\widehat{G}_t = \text{Proj}_{\mathcal{G}}(\widehat{G}_t^\star), \quad \widehat{G}_t^\star = \arg\min_{\widehat{G}} \sum_{p=t_d+h}^{t-h} \ell_p\left(\widehat{G}\right) + \lambda\|\widehat{G}\|^2,$$

$$\ell_p\left(\widehat{G}\right) = \|y_p^{\pi_{\text{DAC-O}}} - \sum_{l=1}^{h} \widehat{G}^{[l]} \delta u_{p-l}^{\pi_{\text{DAC-O}}}\|^2. \quad (14)$$

Main Result: The complete algorithm for the unknown system case is shown in Algorithm 2. We state the performance of the algorithm OLC-ZK-CPD formally below.

Definition 3 (Parameters). $h = \frac{\log T}{\log(1/(1-\gamma))}$, *and*

$\beta = 2\sqrt{h}\zeta_\Delta\left(\sqrt{n\log(2)+2\log\left(\frac{2h}{\delta}\right)}+\frac{\lambda\kappa_a\kappa_b}{\gamma\zeta_\Delta\sigma\sqrt{hN}}\right)$, *where* $\zeta_\Delta = \left(R_s+\frac{\kappa_a\kappa_b\kappa_m\kappa_w h}{\gamma}+\frac{\kappa_a\kappa_b R_u}{\gamma}\right), R_u = \kappa_M\kappa_w h + 3\sigma\sqrt{m+\log(1/\delta)}$, $R_s = \frac{\kappa_a\kappa_w}{\gamma}+\kappa_e+\frac{2R_u\kappa_a\kappa_b}{\gamma}$ *and* $\delta > 0$ *is a constant.* $N = \Gamma_T^{-0.8}T^{4/5}$, $\sigma = \Gamma_T^{0.2}T^{-1/5}$.

Theorem 2 (Zero System Knowledge)**.** *Consider Algorithm 2 with the parameters given by Definition 3. Suppose* $T \geq 3$, $H = \mathcal{O}(\log(T))$, $\zeta = \mathcal{O}(h^2)$, $\eta_i = \mathcal{O}(2^{i-1}/\sqrt{\zeta T})$, $\lambda \propto \mathcal{O}(1)$, $\Gamma_T = \widetilde{O}(T^d)$, $d < 1$ *and the setting is either S-1 or S-2. Then, for* $\delta \leq 1/T$, $\widetilde{\delta}$ *arbitrarily small and* $\delta \leq \widetilde{\delta}$, *the regret with respect to any DAC policy* $M_{1:T}^\star$, $R_T \leq \widetilde{\mathcal{O}}\left(\sqrt{T(1+P_T)}+\Gamma_T^{1/5}T^{4/5}\right)$ *with probability greater than* $1-\widetilde{\delta}$, *where* P_T *is the path length of the sequence* $M_{1:T}^\star$.

Please see Arxiv version for the full proof.

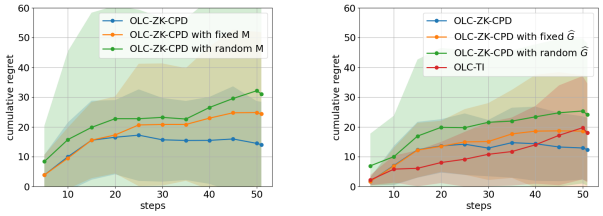
Definition 4 (Switching DAC Policy)**.** *We defining a switching DAC policy as a policy which shifts its policy parameter* M *at the instances of change in the underlying system.*

Corollary 1 (Best Switching DAC Policy)**.** *Suppose the setting is either S-1 or S-2. Then, under the parameter setting of Theorem 2, for any* $\delta \leq 1/T$, $\widetilde{\delta}$ *arbitrarily small and* $\delta \leq \widetilde{\delta}$, *the regret with respect to the best switching DAC policy* $M_{1:T}^\star$, $R_T \leq \widetilde{\mathcal{O}}\left(\Gamma_T^{1/5}T^{4/5}\right)$ *with probability greater than* $1-\widetilde{\delta}$.

This is a straightforward conclusion that follows from Theorem 2 after recognizing the fact that the number of switches of the switching policy is Γ_T.

Remark 2 (Regret Result). Minasyan et al. [1] prove an adaptive regret bound of $\widetilde{\mathcal{O}}\left(|I|\sigma_I + T^{2/3}\right)$ for any interval I of length $|I|$, where σ_I is the square root of the average squared deviation of G_t over the interval I. The key difference compared to [1] is that our result is sub-linear with respect to the number of changes Γ_T instead of σ, and we present a dynamic regret bound that is $\widetilde{\mathcal{O}}\left(\sqrt{T(1+P_T)}+\Gamma_T^{1/5}T^{4/5}\right)$. To compare with [1], lets consider the best switching policy corresponding to the switches in the underlying system. Let I be any interval where the system does not change and let $M_k^\star$ correspond to the best policy parameter for the interval k. Then, the regret achieved by [1] with respect to $M_{1:\Gamma_T}^\star$ is $\Gamma_T T^{2/3}$. The regret achieved by our algorithm is $\widetilde{\mathcal{O}}\left(\Gamma_T^{1/5}T^{4/5}\right)$, which follows from the fact that $P_T = \mathcal{O}(\Gamma_T)$. It follows that, we can achieve a sub-linear regret guarantee for $\Gamma_T = \widetilde{O}(T^d)$ for any $d < 1$, whereas the achievability of sub-linear regret in [1] is limited to scenarios with $\Gamma_T = o(T^{1/3})$.

Remark 3 (Unknown Time Variation). Algorithm 2 assumes the knowledge of total number of changes. We can extend our algorithm to the unknown time variation case by learning the optimal interval period N and optimal σ from an ensemble by using a meta-bandit algorithm on top of Algorithm 2 just as in [13]. We plan to incorporate this in our journal version.



(a) (b)

Fig. 1: Cumulative regret of OLC-ZK-CPD with (a) different M estimation and (b) different $\widehat{G}$ estimation.

IV. Numerical Experiments

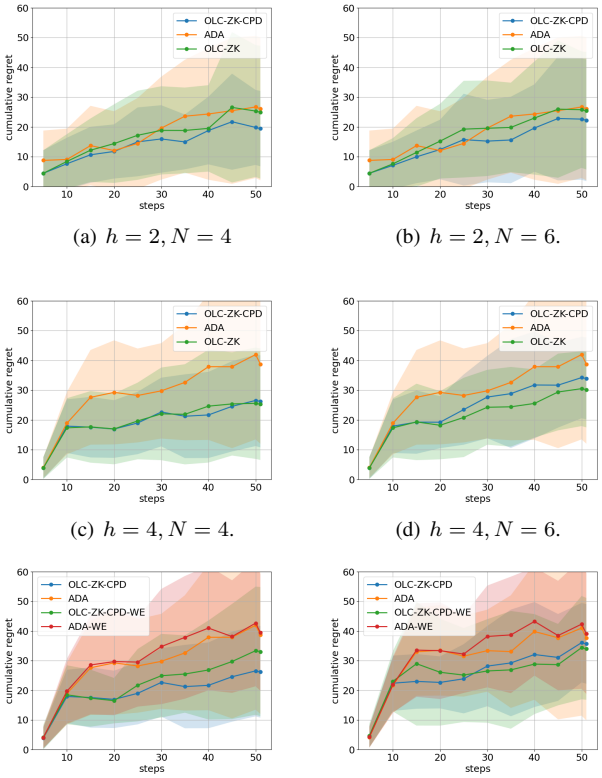
In this section, experimental results are presented for illustrating the performance of OLC-ZK-CPD.

Parameter setting: For all experiments, $\theta_t := [A_t, B_t]$ and w_t are randomly generated at each time step; C_t is randomly initialized, but is kept unchanged across all time steps: $C_{t_1} = C_{t_2}$, $\forall\ t_1, t_2 \in [1,t]$, and $e_t = 0$, $\forall t \in [1,t]$. The cost function is a quadratic function of y_t and u_t: $c_t(y_t,u_t) = y_t^T Q y_t + u_t^T R u_t$. The matrices Q and R are randomly generated positive semi-definite matrices. Experiments are averaged over 10 random runs. In each run, all the algorithms use the same Q, R, C, A_t and B_t.

Baselines: Below, we describe the baseline algorithms we compare OLC-ZK-CPD with.

- *OLC-ZK*: is the online learning algorithm where the output $\widehat{G}_k^{cd}$ is itself used as the estimate of the system parameters for the duration of the period of the next interval of the change point detection procedure. At the end of the next interval, the estimate is updated to $\widehat{G}_{k+1}^{cd}$ and so on.
- *Adaptive Estimation Algorithm (ADA)*: is Algorithm 2 with the estimation algorithm in [1], in place of the estimation approach in Algorithm 2. Essentially, in this combination, what is retained is only the policy parameter update step, with the entire estimation approach replaced by the adaptive estimation algorithm in [1].
- *OLC-TI*: is the online learning algorithm for time invariant systems [5]. In contrast to ours, which continuously explores and exploits, OLC-TI explores first and then exploits.
- *OLC-ZK-CPD with fixed* M: is the online algorithm where M is a fixed value and is not udpated. OLC-ZK-CPD with random M: the online algorithm where M is picked randomly.
- *OLC-ZK-CPD with fixed* $\widehat{G}$: is the OLC-ZK-CPD algorithm with $\widehat{G}_t$ fixed to a constant value instead of an estimator.
- *OLC-ZK-CPD with random* $\widehat{G}$: is the OLC-ZK-CPD algorithm with $\widehat{G}_t$ picked randomly.

Results: In the figures, the shaded regions represent the standard deviation for the respective algorithms. Figure 1(a) indicates that OLC-ZK-CPD has a smaller sub-linear increase in cumulative regret and smaller variance compared to the case when a fixed M or a randomly generated M is used



(a) $h = 2, N = 4$ (b) $h = 2, N = 6$.

(c) $h = 4, N = 4$. (d) $h = 4, N = 6$.

(e) $h = 4, N = 4$. (f) $h = 4, N = 6$.

Fig. 2: Performance Comparison with Baseline Algorithms for Time-Varying Dynamical Systems. OLC-ZK-CPD-WE: OLC-ZK-CPD with disturbance estimation. ADA-WE: ADA with disturbance estimation.

instead. Similarly, it can be observed from Figure 1(b) that the proposed OLC-ZK-CPD algorithm achieves a smaller sub-linear regret with smaller variance compared to the case when a fixed $\widehat{G}$ or a randomly generated $\widehat{G}$ is applied instead of Eq. (14). Most importantly, while, initially the OLC-TI algorithm is better, over time its performance worsens and converges to the OLC-ZK-CPD with an arbitrarily fixed $\widehat{G}$. This is expected as the estimate from the initial exploration phase of OLC-TI can be very different from the underlying dynamical system after a sufficiently long time and thus behave like an arbitrarily fixed $\widehat{G}$ over time. These results corroborate the effectiveness of our proposed algorithm in adapting to time variations.

In Fig. 2, we compare the performance of OLC-ZK-CPD algorithm with the other adaptive algorithms for time-varying dynamical systems such as the OLC-ZK algorithm [14] and the ADA algorithm. The plots are averaged over 10 random runs with parameters $N = [4, 6]$ for $h = 2$ and $N = [4, 6]$ for $h = 4$. In each run, all the algorithms are simulated with the same Q, R, C, A_t, and B_t. The performance of ADA is unchanged with N because it does not use the parameter N. We recall that h defines the length of the history of disturbances in the DAC policy. For $h = 2$, we find that OLC-ZK-CPD achieves a better regret compared to ADA and OLC-ZK. For $h = 4$, we find that our algorithms OLC-ZK-CPD and OLC-ZK achieve much better regret compared to ADA. Overall, we also find that the statistical deviation of OLC-ZK-CPD and OLC-ZK is lesser compared to ADA, showing that OLC-ZK-CPD and OLC-ZK are more stable in the statistical sense. We also note that, across all parameter settings, OLC-ZK-CPD achieves the best performance. In Figs. 2(e) and 2(f), we show the performance of OLC-ZK-CPD and ADA that uses an estimate $\widehat{w}_t$ in their respective control laws instead of the actual disturbance value w_t. Given that $B_{t,w} = I$ in this specific case, the disturbance estimate was calculated as $\widehat{w}_t = x_{t+1} - \widehat{A}_t x_t - \widehat{B}_t u_t$, where $\widehat{A}_t$ and $\widehat{B}_t$ were calculated according to [15] from the estimate $\widehat{G}_t$. OLC-ZK-CPD is better than ADA in this case as well.

V. Conclusion

In this work, we study the problem of online control of unknown time varying dynamical systems with arbitrary disturbances and cost functions. Our goal is to design an online adaptation algorithm that can provably achieve sub-linear regret up to any sub-linear number of changes in the underlying system. We present system, information, and cost structures along with algorithms that guarantee such results.

References

[1] E. Minasyan, P. Gradu, M. Simchowitz, and E. Hazan, "Online control of unknown time-varying dynamical systems," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[2] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "Regret bounds for robust adaptive control of the linear quadratic regulator," in *NeurIPS*, 2018.

[3] H. Mania, S. Tu, and B. Recht, "Certainty equivalence is efficient for linear quadratic control," in *Neural Information Processing Systems (NeurIPS)*, 2019.

[4] N. Agarwal, B. Bullins, E. Hazan, S. Kakade, and K. Singh, "Online control with adversarial disturbances," *International Conference on Machine Learning*, pp. 111–119, 2019.

[5] M. Simchowitz, K. Singh, and E. Hazan, "Improper learning for non-stochastic control," *Conference on Learning Theory (COLT)*, pp. 3320–3436, 2020.

[6] C. Yu, G. Shi, S. Chung, Y. Yue, and A. Wierman, "The power of predictions in online control," 2020.

[7] Y. Lin, Y. Hu, G. Shi, H. Sun, G. Qu, and A. Wierman, "Perturbation-based regret analysis of predictive control in linear time varying systems," *Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021.

[8] D. Muthirayan, J. Yuan, D. Kalathil, and P. P. Khargonekar, "Online learning for predictive control with provable regret guarantees," *arXiv preprint arXiv:2111.15041*, 2021.

[9] Y. Han, R. Solozabal, J. Dong, X. Zhou, M. Takac, and B. Gu, "Learning to control under time-varying environment," *arXiv preprint arXiv:2206.02507*, 2022.

[10] D. Baby and Y.-X. Wang, "Optimal dynamic regret in lqr control," *arXiv preprint arXiv:2206.09257*, 2022.

[11] E. Hazan, A. Rakhlin, and P. L. Bartlett, "Adaptive online gradient descent," *Advances in Neural Information Processing Systems*, pp. 65–72, 2008.

[12] P. Zhao, Y.-X. Wang, and Z.-H. Zhou, "Non-stationary online learning with memory and non-stochastic control," pp. 2101–2133, 2022.

[13] P. Zhao, L. Zhang, Y. Jiang, and Z.-H. Zhou, "A simple approach for non-stationary linear bandits," pp. 746–755, 2020.

[14] D. Muthirayan, R. Du, Y. Shen, and P. P. Khargonekar, "Adaptive control of unknown time varying dynamical systems with regret guarantees," *arXiv preprint arXiv:2210.11684*, 2022.

[15] E. Hazan, S. Kakade, and K. Singh, "The nonstochastic control problem," *Algorithmic Learning Theory*, pp. 408–421, 2020.