Online Optimization with Predictions and Switching Costs: Fast Algorithms and the Fundamental Limit

Yingying Li, Guannan Qu, and Na Li

***Abstract*—This paper studies an online optimization problem with switching costs and a finite prediction window. We propose two computationally efficient algorithms: Receding Horizon Gradient Descent (RHGD), and Receding Horizon Accelerated Gradient (RHAG). Both algorithms only require a finite number of gradient evaluations at each time. In addition, we show that the dynamic regrets of the proposed algorithms decay exponentially fast with the length of the prediction window. Moreover, we provide a fundamental lower bound on the dynamic regret for general online algorithms with a finite prediction window. The lower bound *almost* matches the dynamic regret of our RHAG, meaning that the performance will not be improved much even with more computation given limited prediction. This demonstrates that limited prediction information, instead of limited computational power, is the key constraint to performance in online decision making. Lastly, we present simulation results using real-world data in energy systems.**

I. INTRODUCTION

A classic online convex optimization (OCO) problem considers a decision maker interacting with an uncertain or even adversarial environment. Consider a period of T stages. At each time $t \in \{1, \ldots, T\}$, the decision maker picks an action x_t from a convex set X. Then the environment reveals a convex cost $f_t(\cdot)$. As a result, the decision maker suffers the cost $f_t(x_t)$ based on the chosen action. The goal is to minimize the total cost in T stages. Classic OCO has been studied for decades, with an focus on improving online algorithm performance measured by regrets [1]–[4].

Recent years have witnessed a growing interest in applying online optimization to real-time decision making problems, e.g. economic dispatch in power systems [5]–[7], data center scheduling [8], [9], electric vehical charging [10], [11], video streaming [12], and thermal control [13]. However, there are two features of these problems that are generally not captured by the classic OCO formulation: time coupling effect and prediction of the future uncertainties.

Time coupling effect: While classic OCO setup assumes that stage costs $f_t(x_t)$ are completely decoupled with time, in reality it is usually not the case. For example, to change actions from x_{t-1} to x_t, decision makers usually suffer a switching cost or a ramp cost $d(x_t - x_{t-1})$ [8], [9], [14], [15]. In this way, the stage cost becomes time coupled and is defined as: $C_t(x_{t-1}, x_t) := f_t(x_t) + d(x_t - x_{t-1})$.

Prediction: Classic OCO often models environment as adversary and assumes that no information is available about future cost functions. However, in most applications, there is a certain amount of prediction about the future, especially the near future. For example, in power systems, the system operator can make a good prediction about the future demand and generation [16] [17].

Recently, there are some studies from OCO community exploring the effect of prediction, but most of them do not consider time coupled stage costs [18], [19].

In contrast, there have been many control algorithms, in particular, Model Predictive Control (MPC, as known as Receding Horizon Control) [20], [21], developed for decades to handle both the prediction effect and the time coupling effect. One major focus of MPC is to design control rules to stabilize a dynamical system. Additional goals include minimizing total stage costs, as studied in economic MPC [22]–[24]. However, the classic MPC approaches require solving a large optimization problem at each time, which is usually computationally expensive. Though there have been many recent efforts in reducing the computational overhead, e.g. inexact MPC and suboptimal MPC [25]–[28], there are limited results on the efficiency loss of these algorithms, such as bounds on the dynamic regret. This is partially due to the complexity of the underlying system dynamics. Lastly, other similar online control algorithms, such as Averaging Fixed Horizon Control (AFHC) [8], [9], [29], [30], also require solving the associated optimization problems accurately, hence suffering from the same problem of high computational costs.

Contributions of this paper: In this paper we consider an OCO problem with quadratic switching costs and a prediction window W. We focus on the cases where the cost functions $f_t(x_t)$ are α-strongly convex and l-smooth. To design online algorithms for this problem, we first study the structure of offline gradient-based algorithms, which motivates the design of our online algorithm: Receding Horizon Gradient Descent (RHGD), and an accelerated online algorithm: Receding Horizon Accelerated Gradient (RHAG). Our algorithms only require $W+1$ gradient evaluations at each time step, which is more computationally efficient compared to optimization-based algorithms such as MPC and AFHC . Besides, there is a smooth interpolation between our algorithms and a classic online method in the prediction-free setting: when $W = 0$, our algorithm reduces to the classic online gradient descent in OCO [1].

We analyze the online performance of RHGD and RHAG by comparing algorithm outputs and the optimal solution in hindsight. The comparison is measured by dynamic regret. We

The work was supported by NSF 1608509, NSF CAREER 1553407, AFOSR YIP, and ARPA-E through the NODES program. Y. Li, G. Qu and N. Li are with the School of Engineering and Applied Sciences, Harvard University, 33 Oxford Street, Cambridge, MA 02138, USA (email: yingyingli@g.harvard.edu, gqu@g.harvard.edu, nali@seas.harvard.edu).

show that the dynamic regret of RHGD and RHAG i) depends linearly on *path length*, a measure of the total variation of the cost functions $f_t(\cdot)$; ii) decays exponentially with W. The decay rates depend on the convexity α, smoothness l, and the tradeoff parameter β between the cost $f_t(\cdot)$ and the switching cost. The implications of the results are twofold: i) given a fixed prediction window W, the dynamic regret is upper bounded by a constant factor of the path length; thus, if the path length is sublinear $o(T)$, the algorithms achieves a sublinear dynamic regret, $o(T)$. ii) a longer prediction window W decreases the dynamic regret exponentially; thus, the online performance of our algorithms improve significantly when given more prediction information.

Moreover, we study the fundamental limits of general online algorithms with arbitrary computation power for both the no-prediction and the with-prediction cases. When there is no prediction, we show that worst-case dynamic regret of any deterministic online algorithm is the same as the upper bound of OGD's regret up to a constant. When there is a finite prediction window W, the dynamic regret of any online algorithm, even using more computation, decays at most exponentially with W with respect to the path length. Surprisingly, this fundamental decay rate is close to the decay rate of RHAG, meaning that RHAG use the prediction in a *nearly optimal* way, even though RHAG only takes finite steps of gradient evaluations.

We also numerically compare our algorithm RHGD, RHAG with classic algorithm MPC in electricity economic dispatch problem using real-world data. Though MPC performs better than RHGD and RHAG, the dynamic regrets of RHGD and RHAG indeed decay exponentially with the length of prediction window and are comparable to MPC. Moreover, we also construct a data set where RHAG and MPC have similar online performance. This further confirms the *main message* of this paper that *increasing computation does not necessarily improve the online performance given limited prediction information*.

A. Related work

The closest literature related to this paper is online convex optimization (OCO) which we will discuss here. This paper adopts many terms from OCO to study our online decision making problem. In classic OCO, an online algorithm plays against an adversarial environment for T stages, with no prediction information for future stages, or any coupling between stages. The performance of online algorithms are usually measured by regrets. One popular regret measure is called *static regret*, which by its name, compares the algorithm performance with an optimal static action. Many algorithms have been proposed to achieve $o(T)$ static regret, which means the average regret per stage vanishes to zero when T goes to infinity. We refer readers to [1] for an overview. Notice that when the environment is not stationary, a more reasonable benchmark is the optimal actions in hindsight which change with time, and thus the *dynamic regret* has been proposed. It is straightforward that the dynamic regret is no less than the static regret. In fact, it is well-known that when the environment is changing quickly, it might be impossible to achieve a sublinear dynamic regret [31]. Nevertheless, there are many algorithms that are shown to achieve sublinear dynamic regret when the environment is not changing dramatically [2], [4], [19], [31].

There are many different ways to measure the environment variation. A commonly used measure, referred to as *path length* in this paper, is defined by the total variation of the minimizers of cost functions at each stage:

$$\text{Path length} := \sum_{t=1}^{T} \|\theta_t - \theta_{t-1}\| \leq L_T \qquad (1)$$

where $\theta_t \in \arg\min_{x_t \in X} f_t(x_t)$ is the stage optimizer in action space X at stage $t \in [T]$, and L_T is the budget of path length [2], [19]. It has been shown that online gradient descent can achieve $O(L_T)$ dynamic regret given strongly convex and smooth cost functions [2]. Therefore, sublinear regret is guaranteed by online gradient descent when the path length is $o(T)$, which means on average, θ_t gradually stabilizes as T goes to infinity. Another variation measure is defined upon the function value instead of the actions: $V_T = \sum_{t=1}^{T} \sup_{x \in X} |f_t(x) - f_{t-1}(x)|$. It is shown that an online gradient method that restarts after every a few stages can achieve $O(T^{2/3} V_T^{1/3})$ dynamic regret given convex cost functions and $O(\sqrt{V_T T})$ regret given strongly convex cost functions [31]. Moreover, these rates are shown to be optimal among all online algorithm that use one gradient feedback at each stage [31]. It has been pointed out in [19] that the path length L_T and the function variation V_T are not comparable, as there exist scenarios when either one is larger than the other. In this paper, we will adopt L_T for convenience of analysis. There are other variation measures, for the sake of limited space, we will not discuss in details here and refer readers to [2].

We also want to point out some studies from OCO community on the effect of prediction. [18] studies the effect of one-stage prediction without considering switching costs. They proposed an algorithm based on online mirrored descent and show that when the prediction error is $o(T)$, the dynamic regret will also be $o(T)$. Moreover, there are also papers on online optimization with prediction and switching costs, e.g, [8], [9], [29], [30]. For instance, [8] proposes an algorithm AFHC and show that the competitive ratio of AFHC given perfect W-stage prediction is $1 + O(\frac{1}{W})$. Besides, [30] propose another algorithm CHC and show that the dynamic regret is $O(T/W)$ given W-stage noisy prediction. As we mentioned before, these methods requires solving optimization problems exactly at each stage, different from our gradient-based methods.

Lastly, we mention that in addition to the regret analysis from OCO community, there is another way to measure the online algorithm performance: *competitive ratio*, which is defined by the ratio between the online performance and the optimal performance in hindsight. Competitive ratio analysis is commonly adopted in online algorithm problems, which need not to be convex and can be combinatorial problems. A *competitive algorithm* is an online algorithm that achieves a constant competitive ratio. Under certain assumptions, it can be shown that OCO admits competitive online algorithms [3],

[8], [9]. [3], [32], [33] studies that there is tension between low regret and constant competitive ratio. This paper only studies the dynamic regrets of the online algorithms while we leave the competitive ratio analysis as future work.

B. Notations

For vector $x \in X \subseteq \mathbb{R}^n$, norm $\|x\|$ refers to the Euclidean norm, and $\Pi_X(x)$ denotes the projection of x onto set X. We say X has a diameter D if $\forall x, y \in X$, $||x-y|| \le D$. Besides, we denote the transpose of vector x as x'. The same applies to the matrix transpose. In addition, X^T denotes the Cartesian product $X \times X \ldots \times X$ of T copies of set X. Moreover, we define $[T]$ as the set $\{1, \ldots, T\}$ for a positive integer T. For a function $f(x, y)$ of $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$. Let $\nabla f(x, y) \in \mathbb{R}^{m+n}$ be the gradient, and $\frac{\partial f}{\partial x}(x, y) \in \mathbb{R}^m$ be the partial gradient with respect to vector x. For random variable X, we define $\mathbb{E}(X)$ as the expectation and $\operatorname{var}(X)$ as the variance of the random variable. Finally, we define the big-O, big-Omega and small-o notations. For $x = (x_1, \ldots, x_k) \in \mathbb{R}^k$, we write $f(x) = O(g(x))$ as $x \to +\infty$ if there exists a constant M such that $|f(x)| \le M|g(x)|$ for any x such that $x_i \ge M$ $\forall\, i \in [k]$; we write $f(x) = \Omega(g(x))$ as $x \to +\infty$ if there exists a constant M such that $|f(x)| \ge M|g(x)|$ for any x such that $x_i \ge M$ $\forall\, i \in [k]$; and we write $f(x) = o(g(x))$ if $\lim_{x \to +\infty} f(x)/g(x) = 0$.

II. PROBLEM FORMULATION

In this paper we consider a variation of online convex optimization where the algorithm is subject to an additional switching cost on the change of actions from one stage to the next. We consider the case where a finite lookahead window of the future cost function is available at each time. This is motivated by the fact that in many applications, prediction with high precision is available for the near future, e.g. wind generation and load forecast [16] [17].

Formally, we consider online convex optimization over a finite stage horizon T. At each stage $t \in [T]$, a sequence of costs $f_t, \ldots, f_{t+W-1}$ from a *function class* is revealed to the online decision maker.[1] This W-lookahead window is "accurate" in the sense that these are the true cost functions the algorithm will experience in future stages.[2] Given this $W-$lookahead window, the decision maker needs to pick an action x_t from a set $X \subseteq \mathbb{R}^n$ which is assumed to be compact and convex with a diameter D, i.e.,

$$\|x-y\| \le D, \quad \forall\, x, y \in X.$$

Denote the decision profile over the total T stages as $x := (x_1', \ldots, x_T')' \in X^T \subseteq \mathbb{R}^{nT}$. The goal is to minimize the total cost given by

$$C_1^T(x) = \sum_{t=1}^{T} \left(f_t(x_t) + \frac{\beta}{2}\|x_t - x_{t-1}\|^2 \right) \tag{2}$$

where $x_0 \in X$ denotes the initial state of the decision and $\beta \ge 0$ is a weight parameter. We assume that X, x_0, and β are known to the decision maker beforehand. This is reasonable because they can be decided by the decision maker beforehand. Notice that here we consider quadratic switching cost functions, but the analysis can extend to other switching cost functions with properties of such as monotonicity, convexity, and smoothness.

In this paper, we consider the case where f_t is strongly convex, smooth, and with bounded gradient on X. This is formally stated in the following assumption.

Assumption 1. *For any stage $1 \le t \le T$, the cost function f_t satisfies the following conditions:*

i) α-strong convexity:

$$f(y) \ge f(x) + \langle \nabla f(x), y-x \rangle + \frac{\alpha}{2}\|y-x\|^2, \ \forall\, x, y \in \mathbb{R}^n$$

ii) l-smoothness:

$$f(y) \le f(x) + \langle \nabla f(x), y-x \rangle + \frac{l}{2}\|y-x\|^2, \ \forall\, x, y \in \mathbb{R}^n$$

iii) Bounded gradient on X,

$$\|\nabla f(x)\| \le G, \quad \forall\, x \in X.$$

We denote the class of these functions as $\mathscr{F}_X(\alpha, l, G)$.

Under this assumption, the total cost function $C_1^T(x)$ has the following property.

Lemma 1. *If $f_1, \ldots, f_T$ are from the class of $\mathscr{F}_X(\alpha, l, G)$, the total cost function $C_1^T(x)$ is α-strongly convex and is L-smooth on $\mathbb{R}^n$, where $L = l + 4\beta$.*

The proof is deferred to Appendix A. Throughout the paper, we often denote the conditional number as $Q_f := \frac{l+4\beta}{\alpha}$.

In the online decision problem we consider, the decision maker is assumed to know the function class $\mathscr{F}_X(\alpha, l, G)$, i.e., the parameters α, l, G, but the realization of the cost functions $f_1, \ldots, f_T$ happens online.[3]

A. Online Algorithms

Now we are ready to formally state our problem and define the online (deterministic) algorithms considered in this paper. Consider prediction window $W \ge 0$. When $W = 0$, the scenario reduces to the no-prediction one. Let I_t denote online information available at stage $t \ge 1$. I_t consists of all past and future predicted cost functions plus the initial knowledge of the problem:

$$I_t = \{I_0, f_1(\cdot), \ldots, f_{t-1}(\cdot), f_t(\cdot), \ldots, f_{t+W-1}(\cdot)\}, \forall t \ge 1$$

where I_0 stands for the initial knowledge of the problem, α, l, G, β and initial value x_0. An online deterministic algorithm $\mathscr{A}$ can be characterized by a series of deterministic maps $\{\mathscr{A}_t\}_{t=1}^T$ from information sets to X. Specifically, algorithm

---

[1] Strictly speaking, it should be $f_t, \ldots, f_{\min(t+W-1,T)}$

[2] Although this assumption might be unrealistic, it serves as a good benchmark to study the effect of prediction on the online decision making. We leave it as future work to handle inaccurate prediction.

[3] As shown later, the exact value of α, l, G are not necessarily needed in the proposed online algorithms. We assume the knowledge of these parameters to simplify the mathematical expositions.

$\mathscr{A}$ computes an output $x_t^{\mathscr{A}}$ based on map $\mathscr{A}_t$ and current online information I_t for all t:

$$x_t^{\mathscr{A}} = \mathscr{A}_t(I_t), \quad \forall t \in [T] \tag{3}$$

In the following, when we say $\mathscr{A}$ is an online deterministic algorithm, we mean it satisfies (3). We remark here that I_t implicitly contains all the history decisions $\{x_\tau\}_{\tau=1}^{t-1}$ because these decisions are fully determined by $I_\tau \subseteq I_t$. The goal of this paper is to design computationally efficient algorithms to minimize the overall cost (2) and to understand the fundamental limit of such online algorithms (3). We will formally define the online performance metrics in the next subsection. Notice that the only requirement imposed by (3) is that it only uses past information and prediction information to compute the decision. This feature is generally satisfied by any online algorithm that has been proposed in literature.

The problem setup has natural applications in many areas. Here we briefly discuss two application examples.

Example 1. *(Economic Dispatch in Power Systems.) Consider a power system with conventional generators and renewable energy supply. At stage t, let* $x_t = \{x_{t,i}\}_{i=1}^n$ *be the outputs of* n *generators and* X *be the set of feasible outputs. The generation cost of generator* i *is* $c^i(x_{t,i})$*. The renewable supply is* r_t *and the demand is* d_t*.*

At stage t*, the goal of economic dispatch is to reduce total generation cost while maintaining power balance:* $\sum_{i=1}^n x_{t,i} + r_t = d_t$*. Thus we incorporate imbalance penalty into the objective and consider the cost function*

$$f_t(x_t) = \sum_{i=1}^n c^i(x_{t,i}) + \xi_t(\sum_{i=1}^n x_{t,i} - r_t - d_t)^2$$

where ξ_t *is a penalty factor. In literature,* $c^i(x_{t,i})$ *is usually modeled as a quadratic function within capacity limit [5]. It is easy to see that* $f_t(x_t)$ *belongs to class* $\mathscr{F}_X(\alpha, l, G)$*.*

In addition to the costs above, ramping conventional generators also incurs significant costs, e.g. maintenance and depreciation fee. In literature, such costs are referred as ramp costs and modeled as a quadratic function of ramping rate $\frac{\beta}{2}\|x_t - x_{t-1}\|^2 := \sum_{i=1}^n \frac{\beta}{2}\|x_{i,t} - x_{i,t-1}\|^2$ *[14] [15]. As a result, the objective of economic dispatch for* T *stages is to minimize the total costs including ramp costs*

$$\min_{x_t} \sum_{t=1}^T \left(f_t(x_t) + \frac{\beta}{2}\|x_t - x_{t-1}\|^2 \right)$$

Although demand and renewable supply are uncertain and time-varying, prediction are available for a short time window [16] [17].

Example 2. *(Trajectory Tracking): Consider a simple dynamical system* $x_{t+1} = x_t + u_t$*, where* x_t *is the location of a robot,* u_t *is the control action (velocity of the robot). Let* y_t *be the location of the target at stage* t*, and the tracking error is given by* $f_t(x_t) = \frac{1}{2}\|x_t - y_t\|^2$*. There will also be an energy cost for each control action, given by* $\frac{\beta}{2}\|u_t\|^2 = \frac{\beta}{2}\|x_{t+1} - x_t\|^2$*. The objective is to minimize the sum of the tracking error and the energy loss,*

$$\min_{x_t} \sum_{t=0}^{T-1} \left(f_t(x_t) + \frac{\beta}{2}\|x_{t+1} - x_t\|^2 \right) + f_T(x_T).$$

In reality, there is usually a short lookahead window W *for the target trajectory* y_t *[34].*

B. Performance Metric: Dynamic Regret

In this paper, we adopt *dynamic regret* as the performance metric of online algorithms. Before the formal definition of dynamic regret, we introduce some useful concepts. Let $\{f_t\}_{t=1}^T$ be a cost sequence. First, given an online algorithm $\mathscr{A}$, we denotes algorithm $\mathscr{A}$'s total online cost over T stages by $C_1^T(x^{\mathscr{A}})$:

$$C_1^T(x^{\mathscr{A}}) = \sum_{t=1}^T \left(f_t(x_t^{\mathscr{A}}) + \frac{\beta}{2}\|x_t^{\mathscr{A}} - x_{t-1}^{\mathscr{A}}\|^2 \right)$$

where $x^{\mathscr{A}}$ denotes the outputs of algorithm $\mathscr{A}$ and $x_0^{\mathscr{A}} = x_0$. We remark here that $x^{\mathscr{A}}$ and $C_1^T(x^{\mathscr{A}})$ depend on the cost sequence $\{f_t\}_{t=1}^T$. For the sake of simplicity, we do not put $\{f_t\}_{t=1}^T$ into the notations of $x^{\mathscr{A}}$ and $C_1^T(x^{\mathscr{A}})$.

Secondly, we define the optimal offline total costs in hindsight by solving the offline optimization assuming $\{f_t\}_{t=1}^T$ is available,

$$C_1^T(x^*) = \min_{x \in X^T} \sum_{t=1}^T \left(f_t(x_t) + \frac{\beta}{2}\|x_t - x_{t-1}\|^2 \right)$$

where x^* represents the optimal offline actions and $x_0^* = x_0$.

Lastly, we define *path length*, which represents the variation of cost functions $\{f_t\}_{t=1}^T$, and plays an important role in the dynamic regret analysis of online algorithms [2] [31]. In this paper, we consider the *path length* of a function sequence $\{f_t\}_{t=1}^T$ as the accumulative absolute differences of stage optimizers in consecutive stages:

$$\text{Path length:} \quad \sum_{t=1}^T \|\theta_t - \theta_{t-1}\| \tag{4}$$

where $\theta_t \in \arg\min_{x_t \in X} f_t(x_t)$ is the stage optimizer at stage $t \in [T]$ and $\theta_0 = x_0$.[4] It is easy to see that the path length is within $[0, DT]$ since X has a finite diameter D.

In the following, we let $\mathscr{L}_T(L_T, \mathscr{F}_X(\alpha, l, G))$ denote the set of function sequences $\{f_t\}_{t=1}^T$ in $\mathscr{F}_X(\alpha, l, G)$ whose path length is no more than L_T:

$$\begin{aligned} &\mathscr{L}_T(L_T, \mathscr{F}_X(\alpha, l, G)) \\ &:= \{\{f_t\}_{t=1}^T \subseteq \mathscr{F}_X(\alpha, l, G) | \sum_{t=1}^T \|\theta_t - \theta_{t-1}\| \le L_T\} \end{aligned}$$

Notice that L_T serves as the budget of path length of function class $\mathscr{L}_T(L_T, \mathscr{F}_X(\alpha, l, G))$. Since path length is within $[0, DT]$, we only consider

$$0 \le L_T \le DT$$

---

[4] [2] gives an overview of possible characterizations of the variation of cost function sequences in literature.

without loss of generality. When L_T and $\mathscr{F}_X(\alpha, l, G)$ are clear from context, we write it as a short form $\mathscr{L}_T$ in the rest of the paper.

Now, we are ready to define the dynamic regret. The dynamic regret of algorithm $\mathscr{A}$ is defined by the supremum of the difference between algorithm online cost and the optimal offline cost over all function sequences $\{f_t\}_{t=1}^T$ in $\mathscr{L}_T$,

$$\text{Reg}(\mathscr{A}, \mathscr{L}_T) := \sup_{\{f_t\}_{t=1}^T \in \mathscr{L}_T} \left(C_1^T(x^{\mathscr{A}}) - C_1^T(x^*) \right) \tag{5}$$

Most literature, as well as this paper, try to design algorithms that guarantee sublinear regret when path length is sublinear in T [2], [4], [19], [31].

III. CLASSIC APPROACHES

Before presenting our algorithm, we briefly review some classic algorithms in this section. For the setting without prediction, we introduce the classic online gradient descent (OGD) and its theoretical performance. For the setting with prediction, we introduce the classic control algorithm, model predictive control (MPC) and its variants.

A. *Online Gradient Descent*

In the classic online convex optimization setting, the decision maker needs to decide x_t before f_t or any other future costs are revealed. Online Gradient Descent (OGD) chooses the action by gradient update based on the cost function f_{t-1} and the action x_{t-1} at the previous stage:

$$x_t = \Pi_X \left(x_{t-1} - \gamma \nabla f_{t-1}(x_{t-1}) \right) \tag{6}$$

At stage $t = 1$, let $x_1 = x_0$.

Though OGD is well studied in literature [8] [2] [1], to the best of our knowledge, OGD's dynamic regret for OCO with switching costs has not been stated explicitly. Thus we present it here.

Theorem 1. *Consider the set of function sequences $\mathscr{L}_T(L_T, \mathscr{F}_X(\alpha, l, G))$. Given stepsize $\gamma = 1/l$, the dynamic regret of OGD is upper bounded by:*

$$\text{Reg}(OGD, \mathscr{L}_T) \le \delta L_T$$

where $\delta = (\beta/l + 1)\frac{G}{(1-\kappa)}$, $\kappa = \sqrt{(1 - \frac{\alpha}{l})}$.

Proof. See Appendix B. □

In Section VI, we study the general lower bound of the dynamic regrets for online optimization with switching cost. When $W = 0$, the lower bound matches OGD's regret upper bound up to a constant (Theorem 3). Thus, when there is no prediction available, OGD is an effective algorithm for online optimization with switching costs. This is quite surprising because OGD only takes one gradient evaluation and one projection at each stage.

B. *Model Predictive Control and Its Variants*

When there exists a W-lookahead window, MPC is a commonly used algorithm. At each stage s, MPC solves a W-stage optimization problem:

$$\min_{X^W} C_s^{s+W-1}(x_s, \ldots, x_{s+W-1}) + T_{s+W}(x_{s+W-1}) \tag{7}$$

where

$$C_s^{s+W-1}(\cdot) = \sum_{t=s}^{s+W-1} \left(f_t(x_t) + \frac{\beta}{2}\|x_t - x_{t-1}\|^2 \right),$$

x_{s-1} is determined by previous iterations and $T_{s+W}(x_{s+W-1})$ is a terminal cost function. Let $((x_s^s)', \ldots, (x_{s+W-1}^s)')'$ denote the solution to (7). The output of MPC is x_s^s at stage s.

Though MPC enjoys much better performance than OGD thanks to prediction information, one major drawback of MPC is that it requires to solve the optimization problem (7) at each stage. This might lead to a large computational burden. Considering that OGD is an effective online algorithm for $W = 0$ by using gradient updates, a natural question is whether we can utilize prediction effectively also by gradient updates, which motivates the study of this paper.

In the rest of this paper, we will introduce two new gradient-based online algorithms, Receding Horizon Gradient Descent (RHGD) and Receding Horizon Accelerated Gradient (RHAG). We will show that they, especially RHAG, achieve almost the optimal online performance using the W-lookahead window.

Before going to our algorithm design, we would like to comment on previous efforts on reducing computational complexity of MPC. In particular, the control community has proposed several methods, e.g. inexact MPC and suboptimal MPC [25]–[28], and studied properties of stability and transient performance for trajectories converging to a steady state. However, in online optimization, optimal solutions generally do not converge. Thus, current theoretical results cannot be applied to the problem considered in this paper.

IV. RECEDING HORIZON GRADIENT BASED ALGORITHMS

In this section, we will introduce our two online algorithms: Receding Horizon Gradient Descent (RHGD) and Receding Horizon Accelerated Gradient (RHAG), and provide the dynamic regrets of these two algorithms. Both algorithms are adapted from offline gradient based algorithms: gradient descent and Nesterov's accelerated gradient descent respectively. Our online algorithms only evaluate gradients and projections for $W+1$ times at each stage, so they are more computational friendly when the projection on to set X can be computed efficiently.

A. *Receding Horizon Gradient Descent*

Before introducing RHGD, we first analyze the special structure of gradient descent for the offline optimization problem. This structure motivates our online algorithm RHGD.

1) Offline Problem and Gradient Descent: Given cost functions $f_1, \ldots, f_T$, the offline optimization problem is

$$\min_{x \in X^T} C_1^T(x) = \min_{x \in X^T} \sum_{t=1}^{T} \left(f_t(x_t) + \frac{\beta}{2} \|x_t - x_{t-1}\|^2 \right) \quad (8)$$

Apply Gradient Descent to solve (8):

$$x^{(k)} = \Pi_{X^T} \left(x^{(k-1)} - \eta \nabla C_1^T(x^{(k-1)}) \right) \quad (9)$$

where $\eta > 0$ is the stepsize, $x^{(k)}$ denotes the kth update of x whose initial value is $x^{(0)}$. Considering the update of each x_t, we can rewrite the updating rule (9) as

$$x_t^{(k)} = \Pi_X \left(x_t^{(k-1)} - \eta g_t(x_{t-1}^{(k-1)}, x_t^{(k-1)}, x_{t+1}^{(k-1)}) \right), \quad (10)$$

where $t \in [T]$, $g_t(\cdot)$ denotes the partial gradient of $C_1^T(\cdot)$ with respect to x_t, i.e. $g_t(\cdot) = \frac{\partial C_1^T}{\partial x_t}$. Moreover, due to the special structure of offline optimization (8), $g_t(\cdot)$ only depends on neighboring actions x_{t-1}, x_t, x_{t+1} and has an explicit expression:

$$g_t(x_{t-1}, x_t, x_{t+1}) = \nabla f_t(x_t) + \beta(2x_t - x_{t-1} - x_{t+1}), \; t < T$$
$$g_T(x_{T-1}, x_T, x_{T+1}) = \nabla f_T(x_T) + \beta(x_T - x_{T-1})$$

To ease the notation, we write $g_T(x_{T-1}, x_T, x_{T+1})$ even though there is no such x_{T+1} and $g_T(\cdot)$ does not depend on x_{T+1}.

Now let us consider the online scenario. The major difficulty of online optimization is the lack of future information. However, thanks to the special structure of our problem, rule (10) only needs one-step-forward information x_{t+1} to update x_t. Thus, given W-prediction, we are able to implement (10) in an online fashion, which motivates our design of RHGD.

Algorithm 1 Receding Horizon Gradient Descent

```
1: Inputs: $x_0$ (action at time 0), $W$, stepsizes $\gamma, \eta$
2: $x_1^{1-W} \leftarrow x_0$
3: for $s = 2 - W$ to $T$ do
4:     I) Initialize $x_{s+W}$.
5:     if $s + W \le T$ then
6:         $x_{s+W}^s \leftarrow \Pi_X \left( x_{s+W-1}^{s-1} - \gamma \nabla f_{s+W-1}(x_{s+W-1}^{s-1}) \right)$
7:     II) Update $x_s, \ldots, x_{s+W-1}$ backwards.
8:     for $t = \min(s + W - 1, T) : -1 : \max(s, 1)$ do
9:         $x_t^s \leftarrow \Pi_X \left( x_t^{s-1} - \eta g_t(x_{t-1}^{s-2}, x_t^{s-1}, x_{t+1}^s) \right)$
10: Ouputs: $x_t^t$ at time $t = 1, \ldots, T$.
```

2) Online Algorithm RHGD: Roughly speaking, RHGD has two parts: I) initializing each action by OGD, II) updating each action by applying gradient descent for W steps. The pseudocode is given in Algorithm 1. In the following, we will first introduce the notations, then explain the algorithm in details. In particular, we will show that our algorithm is indeed an online algorithm, in the sense that the evaluation at stage t only requires information available at stage t. Finally, we will discuss the computational overhead.

First we introduce the notations in Algorithm 1. To determine the decision x_t at stage t, RHGD computes the initial decision at stage $t - W$, which is denoted by x_t^{t-W}. Then RHGD updates the value of x_t by one gradient descent step at each stage from $s = t - W - 1$ to $s = t$. Let x_t^s denote the value of x_t at stage s for $s \in \{t - W + 1, \ldots, t\}$, where the superscript specifies the stage when the value x_t is computed. The final decision of x_t is computed at stage t, which is x_t^t, the output of RHGD. In summary, for each decision variable x_t, we have,

I): Initial value: x_t^{t-W}
II): kth update: x_t^s, where $s = t - W + k$, $k \in [W]$

Next, we explain the algorithm rules. First is the initialization rule. Recall that the prediction window at stage $t - W$ is $[t - W, t - 1]$, so $f_{t-1}(\cdot)$ is available but $f_t(\cdot)$ is unknown. Hence, we apply OGD and initialize x_t based on cost function $f_{t-1}(\cdot)$ and the initial decision $x_{t-1}^{(t-1)-W}$ computed at the previous stage:

$$x_t^{t-W} = \Pi_X \left(x_{t-1}^{(t-1)-W} - \gamma \nabla f_{t-1}(x_{t-1}^{(t-1)-W}) \right) \quad (11)$$

where $\gamma > 0$ is the stepsize and $x_1^{1-W} = x_0$.

The updating rule is by rewritting the offline gradient descent updating rule (10). Remember that x_t^s is the kth update of x_t at stage $s = t - W + k$. Similarly, $x_{t-1}^{s-2}, x_t^{s-1}, x_{t+1}^s$ are the $(k-1)$th update of x_{t-1}, x_t, x_{t+1} respectively. Therefore, for $s = t - W + 1, \ldots, t$, we can rewrite (10) as

$$x_t^s \;=\; \Pi_X \left(x_t^{s-1} - \eta g_t(x_{t-1}^{s-2}, x_t^{s-1}, x_{t+1}^s) \right) \quad (12)$$

The only remaining thing is to check that RHGD only uses available cost information at each stage. It has been shown that the initialization rule at stage t only uses cost function $f_{t-1}(\cdot)$ and the initial decision $x_{t-1}^{(t-1)-W}$ computed at the previous stage which are both available at the beginning of stage t. In the updating rule, the function $g_t(\cdot)$ is available because $f_t(\cdot)$ is predictable at stage $s = t - W + 1, \ldots, t$. In addition, previously updated decisions x_{t-1}^{s-2}, x_t^{s-1} are available because they are computed before stage s. The only tricky part is x_{t+1}^s which is also computed at stage s. To deal with this, RHGD is designed to update x_{t+1}^s before x_t^s. More specifically, at stage s, RHGD computes $x_s, \ldots, x_{s+W}$ backwards [See Line 7 and 8 in Algorithm 1]. In this way, x_{t+1}^s is available when we compute x_t^s for $t = s, \ldots, s + W - 1$.

Based on our discussion above, it is straightforward to see that RHGD and offline gradient descent have identical updating rules, as stated in Lemma 2. This relation is crucial to our theoretical analysis in Section IV-C.

Lemma 2. *Given the same stepsize η, let $x_t^{(k)}$ denote the kth update according to gradient descent, and x_t^s denote the update at stage s. Notice that x_t^s is the kth update when $s = t - W + k$. If gradient descent and RHGD shares the same initial values, i.e.,*

$$x_t^{(0)} = x_t^{t-W}, \forall t \in [T]$$

then the output of RHGD is the same as offline gradient descent after W iterations:

$$x_t^{(W)} = x_t^t, \forall t \in [T].$$

Proof. The main idea of the proof has already been discussed above. We omit the details due to space limit. □

Here, we discuss the computational overhead of RHGD at each stage $s \in [T]$. At each stage s, RHGD carries out gradient calculation and projection onto X for $W+1$ times. When the set X is simple, such as positive orthant, n-dimensional box, probability simplex, and Euclidean ball, the projection onto X admits explicit solutions which is computationally efficient. In this case, RHGD is much more computationally friendly than solving optimization exactly at each stage.

B. Receding Horizon Accelerated Gradient

RHAG is similar to RHGD except that RHAG's updating rule is based on Nesterov's accelerated gradient method. In this subsection, we will first introduce Nesterov's accelerated gradient method for offline optimization, and then present and explain RHAG.

1) Offline Problem and Nesterov's Accelerated Gradient: Nesterov's accelerated gradient method is well-known for being the optimal first order algorithm [35]. It is more complicated than gradient descent but enjoys a faster convergence rate.

Here, we apply Nesterov's accelerated gradient method to our offline problem (8) and write the updating rule for each action x_t for $t \in [T]$:

$$\begin{aligned} x_t^{(k)} &= \Pi_X\left(y_t^{(k-1)} - \eta g_t(y_{t-1}^{(k-1)}, y_t^{(k-1)} y_{t+1}^{(k-1)})\right) \\ y_t^{(k)} &= (1+\lambda)x_t^{(k)} - \lambda x_t^{(k-1)} \end{aligned} \tag{13}$$

where $\eta = 1/L,\ \lambda = \frac{1-\sqrt{\alpha\eta}}{1+\sqrt{\alpha\eta}}$ and $y_t^{(0)} = x_t^{(0)}$ is given.[5]

Notice that Nesterov's accelerated gradient method's updating rule (13) only needs one-step forward information y_{t+1} to compute x_t and y_t. This pattern is similar to that of gradient descent's updating rule. Therefore, we can use the same trick to design the online algorithm RHAG based on Nesterov's accelerated gradient method.

Algorithm 2 Receding Horizon Accelerated Gradient

```
Inputs: $x_0$ (action at time 0), $W$, $\alpha$, stepsizes $\gamma, \eta$
$y_1^{1-W} \leftarrow x_0$, $x_1^{1-W} \leftarrow x_0$
$\eta \leftarrow 1/L$, $\lambda \leftarrow \frac{1-\sqrt{\alpha\eta}}{1+\sqrt{\alpha\eta}}$
for $s = 2-W$ to $T$ do
    I) Initialize value for $x_{s+W}, y_{s+W}$.
    if $s+W \le T$ then
        $x_{s+W}^s \leftarrow \Pi_X\left(x_{s+W-1}^{s-1} - \gamma \nabla f_{s+W-1}(x_{s+W-1}^{s-1})\right)$
        $y_{s+W}^s \leftarrow x_{s+W}^s$
    II) Update $(x_s, y_s), \ldots, (x_{s+W-1}, y_{s+W-1})$ backwards
    for $t = \min(s+W-1, T) : -1 : \max(s, 1)$ do
        $x_t^s \leftarrow \Pi_X\left(y_t^{s-1} - \eta g_t(y_{t-1}^{s-2}, y_t^{s-1}, y_{t+1}^s)\right)$
        $y_t^s = (1+\lambda)x_t^s - \lambda x_t^{s-1}$
Ouputs: $x_t^t$ at time $t = 1, \ldots, T$.
```

2) Online Algorithm RHAG: We continue using the notations of RHGD: let x_t^s denote the value of x_t computed at stage s, and y_t^s denote the value of y_t computed at stage s.

Similar to RHGD, RHAG initialize the value for x_t using online gradient descent at stage $t-W$, as shown in equation (11). The initial value for y_T is given by $y_t^{t-W} = x_t^{t-W}$. The only difference lies in the updating rule. RHAG's updating rule is given below. For $s = t-W+1, \ldots, t$,

$$\begin{aligned} x_t^s &= \Pi_X\left(y_t^{s-1} - \eta g_t(y_{t-1}^{s-2}, y_t^{s-1}, y_{t+1}^s)\right) \\ y_t^s &= (1+\lambda)x_t^s - \lambda x_t^{s-1} \end{aligned}$$

By the same analysis of RHGD, x_t^s, y_t^s are the kth update of the Nesterov's accelerated gradient method where $k = s-t+W$. Hence this updating rule is identical to the offline Nesterov's accelerated gradient method's updating rule (13) upto W updates. To guarantee the availability of y_{t+1}^s when computing online updates, we apply the same trick: at each stage s, we compute $(x_s, y_s), \ldots, (x_{s+W}, y_{s+W})$ backwards. In summary, RHAG and Nesterov's accelerated gradient method have the same updating rules, as stated in Lemma 3.

Lemma 3. *Let $x_t^{(k)}$ denote the kth update according to Nesterov's accelerated gradient method, and x_t^s denote the update at stage s of RHAG. Notice that x_t^s is the kth update where $s = t-W+k$. If Nesterov's accelerated gradient method and RHAG shares the same initial values*

$$x_t^{(0)} = x_t^{t-W}, \forall s \in [T]$$

then the output of RHAG is the same as offline Nesterov's accelerated gradient method after W iterations:

$$x_t^{(W)} = x_t^t, \forall s \in [T].$$

Proof. The main idea of the proof has already been discussed above. We omit the details due to space limit. ☐

Similar to RHGD, RHAG also carries out $W+1$ gradient evaluations at each stage which is more computationally friendly than MPC especially when the projection onto X can be computed easily.

Remark 1. *The initializing rule in both RHGD and RHAG does not have to be OGD. The advantage of using OGD is that it has good theoretical performance and is easy to implement. Generally speaking, any fast online algorithm for prediction-free problem with good theoretical results can be used as the initialization rule.*

C. Performance Analysis: Dynamic Regret

Now, we provide upper bounds on dynamic regrets of RHGD and RHAG. We will show that both algorithms' performance improve exponentially with W. Moreover, RHAG enjoys better performance than RHGD. For the purpose of easy exposition, we let $x_0 = 0$ without loss of generality.

The theorem below provides upper bounds on RHGD and RHAG's dynamic regrets.

Theorem 2. *Consider the set of function sequences $\mathscr{L}_T(L_T, \mathscr{F}_X(\alpha, l, G))$. Given stepsizes $\gamma = 1/l$, $\eta = 1/L$, the dynamic regrets of RHGD and RHAG are upper bounded by*

$$\mathrm{Reg}(RHGD, \mathscr{L}_T) \le Q_f \delta (1 - \frac{1}{Q_f})^W L_T \tag{14}$$

---

[5] For simplicity, we assume L and α are known. When the parameters are unknown, [35] provides a sophisticated way to design the stepsize.

$$\mathrm{Reg}(RHAG, \mathscr{L}_T) \leq 2\delta(1-\frac{1}{\sqrt{Q_f}})^W L_T \tag{15}$$

where $\delta = (\beta/l+1)\frac{G}{(1-\kappa)}$, $\kappa = \sqrt{(1-\frac{\alpha}{l})}$, $Q_f = \frac{l+4\beta}{\alpha}$.

Before the proof, we make a few comments on the bounds.

Firstly, notice that the bound in (14) depends linearly on L_T. Thus, when cost functions fluctuate sublinearly, both RHGD and RHAG achieve sublinear regret. Moreover, in Section VI we will show that when L_T is lower bounded by a constant factor, any online algorithm's dynamic regret is at least $O(L_T)$.

Secondly, the upper bound decays exponentially fast with the prediction window W. Thus, our online algorithms' performance improves significantly by increasing the lookahead window. This means that our algorithm uses the prediction information effectively.

Finally, since $Q_f > 1$, we have

$$1-\frac{1}{Q_f} \geq 1-\frac{1}{\sqrt{Q_f}}$$

so RHAG's dynamic regret decays much faster than RHGD's, especially when Q_f is large. This means that RHAG uses prediction information much more efficiently. We will further show that RHAG provides a nearly optimal way to exploit prediction information in Section VI.

Now we are ready to prove Theorem 2.

Proof of Theorem 2: Let's first prove the bound for RHGD. Applying Lemma 2, we can convert the dynamic regret of RHGD to the objective error of offline gradient descent after W iterations

$$\mathrm{Reg}(RHGD, \mathscr{L}_T) = \sup_{\{f_t\}_{t=1}^T \in \mathscr{L}_T} C_1^T(x^{(W)}) - C_1^T(x^*)$$

where $x^{(W)} = \{x_1^{(W)}, \dots, x_T^{(W)}\}$ are gradient descent outputs after W iterations.

According to convergence rate of offline gradient descent for strongly convex and smooth functions, we have

$$\begin{aligned}&\mathrm{Reg}(RHGD, \mathscr{L}_T)\\ &\leq \sup_{\{f_t\}_{t=1}^T \in \mathscr{L}_T} \big(C_1^T(x^{(0)}) - C_1^T(x^*)\big) Q_f (1-\frac{1}{Q_f})^W\end{aligned}$$

In addition, the initial values $x_1^{(0)}, \dots, x_T^{(0)}$ are the outputs of OGD. As a result,

$$\mathrm{Reg}(OGD, \mathscr{L}_T) = \sup_{\{f_t\}_{t=1}^T \in \mathscr{L}_T} \big(C_1^T(x^{(0)}) - C_1^T(x^*)\big)$$

Thus, by applying Theorem 1 we have the upper bound for RHGD.

Similarly, for RHAG, the dynamic regret can be bounded by the error bound of offline Nesterov's accelerated gradient method after W iterations:

$$\begin{aligned}&\mathrm{Reg}(RHAG, \mathscr{L}_T)\\ &\leq 2 \sup_{\{f_t\}_{t=1}^T \in \mathscr{L}_T} \big(C_1^T(x^{(0)}) - C_1^T(x^*)\big)(1-\frac{1}{\sqrt{Q_f}})^W\end{aligned}$$

Apply OGD's regret bound in Theorem 1, we prove the upper bound of RHAG's dynamic regret.

V. LOWER BOUNDS: FUNDAMENTAL LIMITS ON DYNAMIC REGRETS

In this section, we will provide fundamental performance limits for online *deterministic* algorithms for both no-prediction case and finite-prediction window case. We consider any online deterministic algorithm, without constraints on the computational power at each stage. We show that among any deterministic online algorithms, OGD achieves an optimal regret upto a constant when there is no prediction and our algorithm RHAG is near-optimal when there is a finite prediction window under some mild conditions.

Recall the online algorithms defined in Section II-A. I_t denotes all the online information available at stage t, and an online algorithm $\mathscr{A}$ defines a map from I_t to $x_t \in X$ for all $t \in [T]$, as shown in (3). Notice that the only requirement imposed by (3) is that it only uses past information and prediction information to compute the decision. The algorithm can either use gradient-based algorithms like our RHGD and RHAG, or optimization-based algorithms such as MPC, or any other methods no matter how complicated the computation is. However, we will show that even for such a broad class of online algorithms, there are fundamental limits on the online performance for both no-prediction case and W-prediction window case, and our proposed gradient-based algorithms nearly match these limits.

A. Lower bounds

In the following, we will provide a lower bound on the dynamic regret for any online deterministic algorithm in the no prediction case.

Theorem 3 (No prediction)**.** *Consider the set of quadratic function sequences* $\mathscr{L}_T(L_T, \mathscr{F}_X(\alpha,\alpha,G))$*, where* α, G *can be any positive values. Suppose* $T \geq 1$ *and* $W = 0$*. For any online deterministic algorithm* $\mathscr{A}$*, the dynamic regret is lower bounded by:*

$$\mathrm{Reg}(\mathscr{A}, \mathscr{L}_T) \geq \tau G L_T \tag{16}$$

where $\tau = \frac{\alpha^2(1-\rho)^2}{32(\alpha+\beta)^2}$, $\rho = \frac{\sqrt{Q}_f - 1}{\sqrt{Q}_f + 1}$, *and* $Q_f = \frac{\alpha+4\beta}{\alpha}$.

Recall that the regret is defined over the supreme of the set $\mathscr{L}_T$ of function sequences:

$$\mathrm{Reg}(\mathscr{A}, \mathscr{L}_T) := \sup_{\{f_t\}_{t=1}^T \in \mathscr{L}_T} \big(C_1^T(x^{\mathscr{A}}) - C_1^T(x^*)\big)$$

Roughly speaking, the lower bound indicates that for any online algorithm $\mathscr{A}$ without prediction, there exists a sequence of functions $f_1, \dots, f_T$ from the quadratic function class $\mathscr{F}_X(\alpha,\alpha,G)$ with path length L_T such that

$$C_1^T(x^{\mathscr{A}}) - C_1^T(x^*) \geq \Omega(L_T)$$

This demonstrates that no sublinear regret is possible if the path length is linear on T. Notice that similar impossibility results have been established for online optimization without switching cost [31].

Comparing the lower bound with the upper bound of OGD in Theorem 1, we note that the upper bound of OGD matches the lower bound upto a constant term, which means that OGD with constant stepsize, as given in Section II-B, achieves a nearly optimal regret even though it only uses one step gradient calculation. We also point out that similar results are established for online optimization without switching costs [31].

The following theorem provides a lower bound for the prediction case.

Theorem 4 (W-prediction window)**.** *Consider the set of quadratic function sequences, $\mathscr{L}_T(L_T, \mathscr{F}_X(\alpha,\alpha,\alpha D))$ where α can be any positive value. Suppose $T \geq 2W$ and $W \geq 1$. For any online deterministic algorithm $\mathscr{A}$, the dynamic regret is lower bounded by:*

$$\mathrm{Reg}(\mathscr{A}, \mathscr{L}_T) \geq \begin{cases} \frac{\tau\alpha D}{3}\rho^{2W}L_T, & \text{if } L_T \geq D \\ \frac{\tau\alpha}{3}\rho^{2W}L_T^2 & \text{if } L_T < D \end{cases} \tag{17}$$

where $\rho = \frac{\sqrt{Q_f}-1}{\sqrt{Q_f}+1}$, $Q_f = \frac{\alpha+4\beta}{\alpha}$, *and* $\tau = \frac{\alpha^2(1-\rho)^2}{32(\alpha+\beta)^2}$.

Similar to no prediction case, the lower bound for W-prediction window case indicates that for any online algorithm $\mathscr{A}$ with $W \in [1, \frac{T}{2}]$ prediction, there exists a sequence of functions $f_1, \ldots, f_T$ from the quadratic function class $\mathscr{F}_X(\alpha,\alpha,\alpha D)$ with path length L_T such that

$$C_1^T(x^{\mathscr{A}}) - C_1^T(x^*) \geq \Omega(\rho^{2W}L_T)$$

when $L_T \geq D$ and

$$C_1^T(x^{\mathscr{A}}) - C_1^T(x^*) \geq \Omega(\rho^{2W}L_T^2)$$

when $L_T \leq D$. Before the proof, we would like to provide some discussion on the results.

Dependence on the prediction window $W \geq 1$.
Theorem 4 shows that when prediction window is not large, e.g., $W \leq T/2$, the dynamic regret decays at most exponentially with prediction window W. In addition, to reach a regret value R, the window W is at least:

$$W \geq \Omega((\sqrt{Q_f}-1)\log(L_T/R))$$

by $\rho^{2W} \geq \exp(-\frac{4W}{\sqrt{Q_f}-1})$.

On the other hand, by Theorem 2, we have that to reach the same regret value R, the prediction window W needed by RHAG is at most:

$$W \leq O(\sqrt{Q_f}\log(L_T/R))$$

by $(1-1/\sqrt{Q_f})^W \leq \exp(-W/\sqrt{Q_f})$.

Comparing the two requirements on W, we claim that RHAG exploits the prediction information in a nearly *optimal* way even though RHAG only carries $W+1$ steps of gradient calculation per stage. This confirms the intuition that when W is not large, the major factor that limits the online algorithm performance is the limited prediction information, not the computational power. Our results tell that the limited prediction information can be exploited by a few gradient updates. Intuitively, given limited prediction information, the online optimization problem is like an optimization with inaccurate objective. Solving an inaccurate optimization completely will not help improve the performance because the error in the objective is a leading error term. Similar arguments can be found in suboptimal MPC [28].

We also want to point out that the factor $\frac{1}{2}$ in the condition $W \leq T/2$ is not restrictive, and can be relaxed to $W \leq T/c$ for any constant $c > 1$. This relaxation will only affect the constant factor $\frac{1}{96}$ in the lower bound in Theorem 4.

We also want to briefly comment on the large prediction window scenario, especially when W is very close to T. In this scenario, the major limiting factor is not prediction information, but the computation power. Since the broad class of online algorithms defined in (3) does not restrict the computational limit, the regret can be very close to 0 when $W \approx T$. In the extreme case when $W = T$, the problem becomes an offline optimization, and the regret is equal to 0 by an infinite number of offline gradient descent steps. Since in practice, W is always small compared to T, we only study the case of small W.

Dependence on the path length L_T.
Notice that the lower bounds are different when $L_T \geq D$ and when $L_T < D$. We will first discuss each scenario one by one, then explain why lower bounds are different in these two scenarios.

When L_T is large, or $L_T \geq D$, the lower bound depends linearly on L_T. This means that given a $O(T)$ linear path length, there is no online algorithm that can achieve sublinear regret even with a finite prediction window. Notice that RHAG and RHGD's regret upper bounds also depend linearly on L_T, so we can claim these two algorithms achieve an optimal dependence on the path length L_T when $L_T \geq D$. We also point out that by definition, the path length L_T is nondecreasing with T. Thus, given a large horizon T, it is very likely that $L_T \geq D$ since D is a constant. Thus it is reasonable to say that our algorithms RHAG and RHGD are optimal with respect to L_T.

When L_T is small, i.e. $L_T \leq D$, the lower bound is proportion to L_T^2, which is smaller than $DL_T \sim O(L_T)$. A $O(L_T^2)$ bound can be achieved by a simple online algorithm where $x_t^{\mathscr{A}} = \theta_t$. This is verified by the following arguments. Since θ_t minimizes each $f_t(\cdot)$, the dynamic regret of $x_t^{\mathscr{A}} = \theta_t$ is upper bounded by the switching costs, i.e. $\sum_{t=1}^T \frac{\beta}{2}\|\theta_t - \theta_{t-1}\|^2$, and we have $\sum_{t=1}^T \|\theta_t - \theta_{t-1}\|^2 \leq L_T^2$. However, when there is no prediction, i.e., $W = 0$, this simple online algorithm can not be implemented because $f_t(\cdot)$ is not available at stage t. This roughly explains the major difference between the no prediction and prediction cases.

Now we roughly discuss the reasons behind the different lower bounds given different L_T. First, notice that L_T can be viewed as the "budget of variation" of the cost functions in the sense that $\sum_t \|\theta_t - \theta_{t-1}\| \leq L_t$, and the dynamic regret for a given algorithm $\mathscr{A}$ can be viewed as the largest possible difference between online and offline performance, i.e., $C_1^T(x^{\mathscr{A}}) - C_1^T(x^*)$, given such budget L_T. To find such "largest" possible difference is to allocate the budget L_T among different stages $t \in [T]$ to maximize $C_1^T(x^{\mathscr{A}}) - C_1^T(x^*)$. Roughly speaking, we can show that given a single-stage variation $\|\theta_t - \theta_{t-1}\| = \nu$, there exist cost functions such that the dynamic regret increases by $O(\nu^2)$ (Lemma 6).

Because $(\nu_1+\nu_2)^2 \geq \nu_1^2+\eta_2^2$, the best way to maximize the difference $C_1^T(x^{\mathscr{A}}) - C_1^T(x^*)$ is to spend all of the budget L_T at one stage. However, since X is compact with diameter D, the maximum change of θ_t at one stage is D. When $L_T \leq D$, we can follow the preceding argument to let all the variation L_T happen at one stage and receive $O(L_T^2)$ regret. When $L_T \geq D$, we let θ_t changes by D for $\frac{L_T}{D}$ stages, i.e., $\|\theta_t - \theta_{t-1}\| = D$ for $\frac{L_T}{D}$ times. The corresponding dynamic regret will be at least $O(D^2\frac{L_T}{D}) = O(L_T)$.

B. Proofs of the Lower Bounds

The proofs are based on constructions, and the main ideas behind the constructions are very similar for Theorem 3 and 4. Hence, in this subsection, we only present the proof for a special case: Theorem 4 when $L_T \geq 2D$. The remaining proof of Theorem 4 and the proof of Theorem 3 are deferred to Appendix G and H respectively.

Recall that the dynamic regret is defined by

$$\mathrm{Reg}(\mathscr{A}, \mathscr{L}_T) := \sup_{\{f_t\}_{t=1}^T \in \mathscr{L}_T} \left(C_1^T(x^{\mathscr{A}}) - C_1^T(x^*) \right)$$

To show the lower bound, we will construct a sequence $\{f_t(\cdot)\}_{t=1}^T \in \mathscr{L}_T(L_T, \mathscr{F}_X(\alpha, \alpha, \alpha D))$ for any online deterministic $\mathscr{A}$ such that

$$C_1^T(x^{\mathscr{A}}) - C_1^T(x^*) \geq \frac{\tau\alpha D}{3}\rho^{2W} L_T \tag{18}$$

The major trick in our proof is that instead of constructing a specific cost function sequence, we will construct a random sequence and show that the inequality (18) holds in expectation. Then, there must exist one realization of the random cost functions satisfying (18). The proof takes four steps:

1) Construct a random sequence $\{f_t(\cdot)\}_{t=1}^T$.
2) Characterize the optimal solution $x^* = \arg\min_{X^T} C_1^T(x)$.
3) Characterize the online algorithm output $x^{\mathscr{A}}$ using
4) Prove the lower bound for $\mathbb{E}[C_1^T(x^{\mathscr{A}}) - C_1^T(x^*)]$.

For simplicity, we consider one dimension case $X \subseteq \mathbb{R}$. Without loss of generality, we consider $x_0 = 0$ and $X = [-\frac{D}{2}, \frac{D}{2}]$ with diameter D.

Step 1: construct random $\{f_t(\cdot)\}_{t=1}^T$:

For any $\alpha > 0$, and $\beta > 0$ (when $\beta = 0$, it is trivial), we construct parameterized quadratic functions as below:

$$f_t(x_t) = \frac{\alpha}{2}(x_t - \theta_t)^2 \tag{19}$$

When $\theta_t \in X$, $f_t(x_t)$ is in the function class $\mathscr{F}_X(\alpha, \alpha, \alpha D)$, in addition, $\theta_t = \arg\min_X f_t(x_t)$.

Now, constructing $\{f_t(\cdot)\}_{t=1}^T$ becomes constructing the vector $\theta = (\theta_1, \dots, \theta_T)'$. Notice that instead of designing specific θ for each online algorithm $\mathscr{A}$, we will construct a random vector θ, as discussed below.

For each $L_T \geq 2D$, define $\Delta = \lceil T/\lfloor L_T/D \rfloor \rceil$, then divide T into $K = \lceil \frac{T}{\Delta} \rceil$ parts:

$$\underbrace{1, \dots, \Delta}, \underbrace{\Delta+1, \dots, 2\Delta}, \dots, \underbrace{(K-1)\Delta+1, \dots, T}$$

where each part has Δ stages, except that the last part may have less stages. Notice that since $0 \leq L_T \leq DT$, when $L_T \geq D$, we have $1 \leq \Delta \leq T$. Hence the construction is well-defined.

At the beginning of each part, i.e., when $t \equiv 1 \pmod{\Delta}$ for $1 \leq t \leq T$, we draw θ_t i.i.d. from distribution $\mathbb{P}(\theta_t = \frac{D}{2}) = \mathbb{P}(\theta_t = -\frac{D}{2}) = \frac{1}{2}$. For other stages in each part, we copy the parameter of the first stage of the corresponding part:

$$\theta_t = \theta_{k\Delta+1},\ k\Delta+2 \leq t \leq \min(k\Delta+\Delta, T),\ k = 0, \dots, K-1$$

We will show in the next Lemma that for each realization of θ, the path length is no more than L_T. The proof is deferred to the Appendix C.

Lemma 4. *Consider the sequence $\{f_t(x_t)\}_{t=1}^T$ where $f_t(x_t) = \frac{\alpha}{2}(x_t - \theta_t)^2$. For any $L_T \geq 2D$, define θ_t as above. Then the path length of $\{f_t(x_t)\}_{t=1}^T$ for every realization of θ_t is no more than L_T, i.e.*

$$\sum_{t=1}^T \|\theta_t - \theta_{t-1}\| \leq L_T$$

where $\theta_0 = x_0 = 0$.

Step 2: characterize x^*

For the constructed quadratic function sequence $\{f_t(\cdot)\}_{t=1}^T$ in Step 1, the optimal solution x^* admits a closed-form solution: $x^* = A\theta$. This closed-form solution specifies how x_t^* depends on the future cost functions, i.e., $\theta_{t+\tau}$ for $\tau \geq 0$. By analyzing the matrix A, we can show that the dependence decays at most exponentially. The above discussion is formally stated in the next Lemma, and proved in the appendix.

Lemma 5. *For any $\theta \in X^T$, there exists a matrix $A \in \mathbb{R}^{T\times T}$, such that $x^* = A\theta$, where $x^* = \arg\min_{X^T} C_1^T(x)$. In addition, A's entries satisfy*

$$a_{t,t+\tau} \geq \frac{\alpha}{\alpha+\beta}(1-\rho)\rho^\tau$$

where $\rho = \frac{\sqrt{Q_f}-1}{\sqrt{Q_f}+1}$, for $\tau \geq 0$ and $T \geq 1$.

Step 3: characterize $x^{\mathscr{A}}$

We claim that $x_t^{\mathscr{A}}$ is a random variable determined by $\{\theta_s\}_{s=1}^{t+W-1}$. The reason is given below.

First notice that $x^{\mathscr{A}}$ is a random variable because $\{f_t(\cdot)\}_{t=1}^T$ is random. For any online deterministic algorithm, we have

$$x_t^{\mathscr{A}} = \mathscr{A}_t(f_1, \dots, f_{t+W-1})$$

and $f_1, \dots, f_{t+W-1}$ are determined by $\theta_1, \dots, \theta_{t+W-1}$. Therefore, $x_t^{\mathscr{A}}$ is determined by $\{\theta_s\}_{s=1}^{t+W-1}$

Step 4: lower bound $\mathbb{E}[C_1^T(x^{\mathscr{A}}) - C_1^T(x^*)]$:

Consider a set of stages J defined by

$$J := \{1 \leq t \leq T-W, t+W \equiv 1 \pmod{\Delta}\}$$

It is straightforward that

$$\mathbb{E}\,\|x^{\mathscr{A}} - x^*\|^2 = \sum_{t=1}^T \mathbb{E}\,\|x_t^{\mathscr{A}} - x_t^*\|^2 \geq \sum_{t\in J} \mathbb{E}\,\|x_t^{\mathscr{A}} - x_t^*\|^2$$

If we can show i) $\mathbb{E}\,\|x_t^{\mathscr{A}} - x_t^*\|^2 \geq \frac{a_{t,t+W}^2 D^2}{4}$ for $t \in J$(Lemma 6), and ii) $|J| \geq \frac{L_T}{12D}$ (Lemma 7), then we can lower bound $\mathbb{E}\,\|x^{\mathscr{A}} - x^*\|^2$ by

$$\begin{aligned}\mathbb{E}\,\|x^{\mathscr{A}} - x^*\|^2 &\geq \sum_{t\in J} \mathbb{E}\,\|x_t^{\mathscr{A}} - x_t^*\|^2 \geq \sum_{t \in J} \frac{a_{t,t+W}^2 D^2}{4} \\ &= |J| \frac{a_{t,t+W}^2 D^2}{4} \geq \frac{L_T D}{48} \left(\frac{\alpha}{\alpha+\beta}\right)^2 (1-\rho)^2 \rho^{2W} \qquad (20)\end{aligned}$$

where the last inequality is by Lemma 5.

Then, since $C_1^T(x)$ is α-strongly convex, we have

$$\begin{aligned}\mathbb{E}[C_1^T(x^{\mathscr{A}}) - C_1^T(x^*)] &\geq \mathbb{E}\,\frac{\alpha}{2}\|x^{\mathscr{A}} - x^*\|^2 \\ &\geq \frac{\alpha D}{96}(1-\rho)^2 \left(\frac{\alpha}{\alpha+\beta}\right)^2 L_T \rho^{2W}\end{aligned}$$

where the last equality is by (20). ☐

Below are the formal statements of Lemma 6 and 7. We note that Lemma 6 provides insight on how single-stage variation affects the dynamic regret. Remember that by construction, the variation at stage $t+W$ satisfies $\|\theta_{t+W} - \theta_{t+W-1}\| \leq D$ when $t \in J$. Lemma 6 shows that such a variation results in a $O(D^2)$ difference between online action and the optimal offline action at time t, which will increase the dynamic regret by $O(D^2)$ due to strong convexity. Even though here D is the action space diameter, the result remains unchanged for any $\nu \leq D$ following the similar construction and proof arguments.

Lemma 6. *Given random θ as defined above, for any online deterministic algorithm $\mathscr{A}$, we have*

$$\mathbb{E}\,\|x_t^{\mathscr{A}} - x_t^*\|^2 \geq \frac{a_{t,t+W}^2 D^2}{4}, \quad \forall t \in J$$

Proof. See Appendix E. ☐

Lemma 7. *If $T \geq 2W$, and $L_T \geq 2D$, then*

$$|J| \geq \frac{L_T}{12D}$$

Proof. See Appendix F. ☐

VI. A NUMERICAL STUDY: ECONOMIC DISPATCH

This section presents two numerical experiments: 1) an economic dispatch problem as introduced in Example 1 using real data; 2) a special case where RHAG and MPC has similar performance.

A. *Economic Dispatch*

In this subsection, we consider an economic dispatch problem, defined in Example 1, with three conventional generators with quadratic costs given below.

$$\begin{aligned}c^1(x_{t,1}) &= 1.4(x_{t,1})^2 + 15x_{t,1} + 10 \\ c^2(x_{t,2}) &= 1.6(x_{t,2})^2 + 10x_{t,2} + 27 \\ c^3(x_{t,3}) &= 2(x_{t,3})^2 + 6x_{t,3} + 21\end{aligned}$$

Change setup and figures to beta =10, xi =0.5, upper bounds higher. Besides, we consider a high-penetration of wind supply as shown in Figure 1 (b) where the data is from [36]. Figure 1 (a) depicts the load profile of Bonneville Power Administration controlled area from January 1 to January 5 in 2017 [36]. Each stage corresponds to 5 minutes. For simplicity, we let $\xi_t = \xi = 1.2$, $\beta = 1$, the capacity of three generators be $[3\times 10^3, 2.6\times 10^3, 2.1 \times 10^3]$MW, and the starting generation amount be $[2\times 10^3, 1\times 10^3, 1\times 10^3]$MW in the simulation.

TABLE I

RUNNING TIME OF RHGD, RHAG AND MPC

W $\mathscr{A}$	5	10
RHGD	8.8781×10^{-5}	1.4923×10^{-4}
RHAG	1.0416×10^{-4}	1.9052×10^{-4}
MPC	0.0105	0.0110

Figure 1 (c) presents the dynamic regret of RHGD, RHAG and MPC in a log scale as a function of prediction window W. Notice that when $W = 0$, i.e. without prediction, RHGD and RHAG reduces to classic OGD. When W increases, the regrets of all three algorithms decay linearly in a log scale, demonstrating exponential decay rates. Moreover, RHAG decays faster than RHGD, aligned with our theoretical results in Theorem 2. Finally, even though RHAG has larger regret than MPC, to reach the same dynamic regret, the prediction window size needed by RHAG is almost three times the prediction window size needed by MPC, which matches our analysis that RHAG exploits prediction information near-optimally.

Table I compares the computational time per stage of RHGD RHAG and MPC for $W = 5, 10$. All algorithms are implemented via Matlab, and MPC uses Matlab's quadprog() solver to solve the optimization. Notice that RHGD and RHAG are significantly faster than MPC, which is intuitive because RHGD and RHAG are gradient based while MPC is optimization based.

B. *A special example*

In this subsection, we provide a special case where RHAG performs almost the same with MPC. Consider the quadratic cost functions defined on $[0, 4]$ in 16 stages:

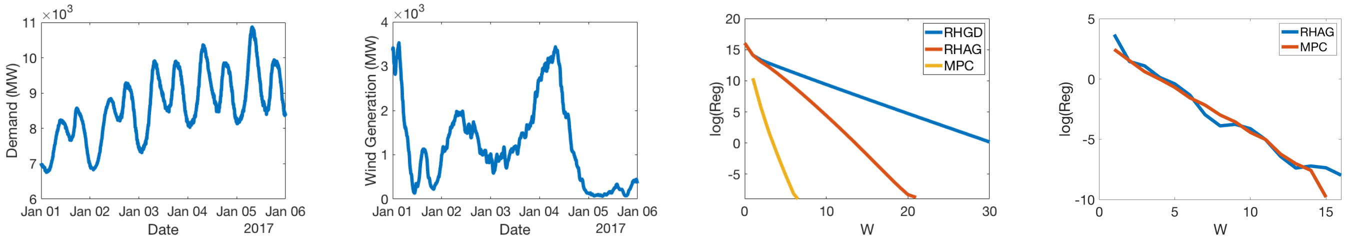
$$f_t(x_t) = 0.5(x_t - \theta_t)^2$$

where θ_t are $[0, 0, 4, 0, 0, 4, 0, 4, 0, 4, 0, 4, 4, 0, 4, 4]$. $\beta = 13$ and $x_0 = 0$. The stepsizes are based on the strong convexity factor and smoothness factor.

Figure 1 (d) compares the dynamic regret of RHAG and MPC. Here we don't compare RHGD because it has poorer performance than RHAG. Notice that RHAG has very similar performance to MPC by using much less computation than MPC. This figure reflects the main message of this paper: more computation may not improve (much) performance in the online setting.

VII. CONCLUSION

In this paper we study online convex optimization problems with switching costs and propose two computational efficient online algorithms, RHGD and RHAG. Our online algorithms only use W steps of prediction and only need $W + 1$ steps of gradient evaluation at each time step. We



(a) Demand (b) Wind Generation (c) Dynamic Regret (d) A special case

Fig. 1. (a) (b) are demand and wind generation profile every 5 minutes from January 1 to 5, 2017, from Bonneville Power Administration [36]. (c) depicts the dynamic regret of RHGD and suboptimal MPC for the economic dispatch problem. (d) shows the dynamic regret of RHAG and MPC in a small example.

show that the dynamic regret of RHGD and RHAG decay exponentially fast with the prediction window W. Moreover, RHAG's decaying rate almost matches the decay rate of the lower bound of general online algorithms, meaning that RHAG exploits prediction information near-optimally by using much less computation time. This means that in the online setting, more computation may not improve (much) performance. As a by-product, we also show that OGD is the optimal online algorithm in the online optimization with switching costs and without prediction.

There are many interesting future directions, such as i) generalizing the method to handle imperfect prediction, ii) designing and studying other computational efficient online algorithms such as suboptimal MPC, iii) studying projection-free algorithms to further reduce the computational complexity.

REFERENCES

[1] E. Hazan, *Introduction to Online Convex Optimization*, ser. Foundations and Trends(r) in Optimization Series. Now Publishers, 2016. [Online]. Available: https://books.google.com/books?id=IFxLvgAACAAJ

[2] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 7195–7201.

[3] L. Andrew, S. Barman, K. Ligett, M. Lin, A. Meyerson, A. Roytman, and A. Wierman, "A tale of two metrics: Simultaneous bounds on competitiveness and regret," in *Conference on Learning Theory*, 2013, pp. 741–763.

[4] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 647–662, 2015.

[5] B. Narayanaswamy, V. K. Garg, and T. Jayram, "Online optimization for the smart (micro) grid," in *Proceedings of the 3rd international conference on future energy systems: where energy, computing and communication meet*. ACM, 2012, p. 19.

[6] M. Moeini-Aghtaie, P. Dehghanian, M. Fotuhi-Firuzabad, and A. Abbaspour, "Multiagent genetic algorithm: an online probabilistic view on economic dispatch of energy hubs constrained by wind availability," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 2, pp. 699–708, 2014.

[7] T. Chen, Q. Ling, and G. B. Giannakis, "An online convex optimization approach to proactive network resource allocation," *IEEE Transactions on Signal Processing*, 2017.

[8] M. Lin, Z. Liu, A. Wierman, and L. L. Andrew, "Online algorithms for geographical load balancing," in *Green Computing Conference (IGCC), 2012 International*. IEEE, 2012, pp. 1–10.

[9] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 5, pp. 1378–1391, 2013.

[10] L. Gan, A. Wierman, U. Topcu, N. Chen, and S. H. Low, "Real-time deferrable load control: handling the uncertainties of renewable generation," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 3, pp. 77–79, 2014.

[11] S.-J. Kim and G. B. Giannakis, "Real-time electricity pricing for demand response using online convex optimization," in *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES*. IEEE, 2014, pp. 1–5.

[12] V. Joseph and G. de Veciana, "Jointly optimizing multi-user rate adaptation for video transport over wireless systems: Mean-fairness-variability tradeoffs," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 567–575.

[13] F. Zanini, D. Atienza, G. De Micheli, and S. P. Boyd, "Online convex optimization-based algorithm for thermal management of mpsocs," in *Proceedings of the 20th symposium on Great lakes symposium on VLSI*. ACM, 2010, pp. 203–208.

[14] M. Tanaka, "Real-time pricing with ramping costs: A new approach to managing a steep change in electricity demand," *Energy Policy*, vol. 34, no. 18, pp. 3634–3643, 2006.

[15] R. Mookherjee, B. F. Hobbs, T. L. Friesz, and M. A. Rigdon, "Dynamic oligopolistic competition on an electric power network with ramping costs and joint sales constraints," *J. Ind. Manag. Optim*, vol. 4, no. 3, pp. 425–452, 2008.

[16] https://www.misoenergy.org/MarketsOperations/RealTimeMarketData\ \/Pages/DayAheadWindForecast.aspx.

[17] http://www.pjm.com/planning/resource-adequacy-planning/ load-forecast-dev-process.aspx.

[18] A. Rakhlin and K. Sridharan, "Online learning with predictable sequences," in *Conference on Learning Theory*, 2013, pp. 993–1019.

[19] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan, "Online optimization: Competing with dynamic comparators," in *Artificial Intelligence and Statistics*, 2015, pp. 398–406.

[20] J. B. R. D. Q. Mayne and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design, 2nd Edition*. Nob Hill Publishing LLC, 2017.

[21] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999.

[22] D. Angeli, R. Amrit, and J. B. Rawlings, "On average performance and stability of economic model predictive control," *IEEE transactions on automatic control*, vol. 57, no. 7, pp. 1615–1626, 2012.

[23] L. Grüne and M. Stieler, "Asymptotic stability and transient optimality of economic mpc without terminal conditions," *Journal of Process Control*, vol. 24, no. 8, pp. 1187–1196, 2014.

[24] A. Ferramosca, D. Limon, and E. F. Camacho, "Economic mpc for a changing economic criterion for linear systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2657–2667, 2014.

[25] M. Kögel and R. Findeisen, "Stabilization of inexact mpc schemes," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 5922–5928.

[26] K. Graichen and A. Kugi, "Stability and incremental improvement of suboptimal mpc without terminal constraints," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2576–2580, 2010.

[27] D. A. Allan, C. N. Bates, M. J. Risbeck, and J. B. Rawlings, "On the inherent robustness of optimal and suboptimal nonlinear mpc," *Systems & Control Letters*, vol. 106, pp. 68–78, 2017.

[28] J. Rawlings and D. Mayne, "Postface to model predictive control: Theory and design," *Nob Hill Pub*, pp. 155–158, 2012.

[29] M. Badiei, N. Li, and A. Wierman, "Online convex optimization with ramp constraints," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015, pp. 6730–6736.

[30] N. Chen, J. Comden, Z. Liu, A. Gandhi, and A. Wierman, "Using predictions in online optimization: Looking forward with an eye on

the past," in *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*. ACM, 2016, pp. 193–206.

[31] O. Besbes, Y. Gur, and A. Zeevi, "Non-stationary stochastic optimization," *Operations research*, vol. 63, no. 5, pp. 1227–1244, 2015.

[32] A. Blum and C. Burch, "On-line learning and the metrical task system problem," *Machine Learning*, vol. 39, no. 1, pp. 35–58, 2000.

[33] N. Buchbinder, S. Chen, J. S. Naor, and O. Shamir, "Unified algorithms for online learning and competitive analysis," in *Conference on Learning Theory*, 2012, pp. 5–1.

[34] R. Rosales and S. Sclaroff, "Improved tracking of multiple humans with trajectory prediction and occlusion modeling," Boston University Computer Science Department, Tech. Rep., 1998.

[35] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[36] https://transmission.bpa.gov/Business/Operations/Wind/default.aspx.

[37] J. M. Varah, "A lower bound for the smallest singular value of a matrix," *Linear Algebra and its Applications*, vol. 11, no. 1, pp. 3–5, 1975.

[38] P. Concus, G. H. Golub, and G. Meurant, "Block preconditioning for the conjugate gradient method," *SIAM Journal on Scientific and Statistical Computing*, vol. 6, no. 1, pp. 220–252, 1985.

APPENDIX

A. Proof of Lemma 1

Proof. Remember that $C_1^T(x) = \sum_t (f_t(x_t) + \frac{\beta}{2}\|x_t - x_{t-1}\|^2)$. Since $\sum_t f_t(x_t)$ is α-strongly convex and l-smooth, we only need to study $\sum_{t=1}^T \frac{\beta}{2}\|x_t - x_{t-1}\|^2$. It can shown that the Hessian of $\sum_{t=1}^T \frac{\beta}{2}\|x_t - x_{t-1}\|^2$ has eigenvalues within $[0, 4\beta]$. So $L = l + 4\beta$, and $C_1^T(x)$ is α-strongly convex. ☐

B. Proof of Theorem 1

Before the formal proof, we introduce a supporting lemma, which upper bounds the switching cost of OGD outputs.

Lemma 8. *Given $f_t \in \mathscr{F}_X(\alpha, l, G)$ for $t = 1, \ldots, T$, and stepsize $\frac{1}{l}$, the outputs of OGD $\{x_t\}_{t=1}^T$ satisfy*

$$\sum_{t=1}^T \|x_t - x_{t-1}\|^2 \le \frac{2G}{l(1-\kappa)} \sum_{t=1}^T \|\theta_t - \theta_{t-1}\|$$

where x_1 is chosen to be $x_1 = x_0$, $\kappa = \sqrt{(1 - \frac{\alpha}{l})}$, $\theta_t := \arg\min_{x_t \in X} f_t(x_t)$ for $1 \le t \le T$ and $\theta_0 = x_0$.

Proof. Firstly, given $x_1 = x_0$, we have

$$\sum_{t=1}^T \|x_t - x_{t-1}\|^2 = \sum_{t=1}^{T-1} \|x_{t+1} - x_t\|^2$$

According to Corollary 2.2.1 (2.2.16) p.87 in [35]

$$\sum_{t=1}^{T-1} \|x_{t+1} - x_t\|^2 \le \frac{2}{l} \sum_{t=1}^{T-1} (f_t(x_t) - f_t(\theta_t)) \tag{21}$$

Since $f_t \in \mathscr{F}_X(\alpha, l, G)$, and $\|x_T - \theta_T\| \ge 0$,

$$\sum_{t=1}^{T-1} f_t(x_t) - f_t(\theta_t) \le G \sum_{t=1}^{T-1} \|x_t - \theta_t\| \le G \sum_{t=1}^{T} \|x_t - \theta_t\| \tag{22}$$

The remainder of the proof is to bound $\sum_{t=1}^T \|x_t - \theta_t\|$. By triangle inequality of Euclidean norm,

$$\sum_{t=1}^T \|x_t - \theta_t\| \le \sum_{t=1}^T (\|x_t - \theta_{t-1}\| + \|\theta_t - \theta_{t-1}\|) \tag{23}$$

From Theorem 2.2.8 p.88 in [35], we have

$$\sum_{t=2}^T \|x_t - \theta_{t-1}\| \le \kappa \sum_{t=1}^{T-1} \|x_t - \theta_t\|$$

where $\kappa = \sqrt{(1 - \frac{\alpha}{l})}$. Plug this in (23), we have

$$\begin{aligned} \sum_{t=1}^T \|x_t - \theta_t\| &\le \kappa \sum_{t=1}^{T-1} \|x_t - \theta_t\| + \sum_{t=1}^T \|\theta_t - \theta_{t-1}\| \\ &\le \kappa \sum_{t=1}^{T} \|x_t - \theta_t\| + \sum_{t=1}^T \|\theta_t - \theta_{t-1}\| \end{aligned} \tag{24}$$

where the first inequality is by $x_1 = \theta_0 = x_0$, and the second one is due to $\|x_T - \theta_T\| \ge 0$.

Regrouping the terms in (24) gives us:

$$\sum_{t=1}^T \|x_t - \theta_t\| \le \frac{1}{1-\kappa} \sum_{t=1}^T \|\theta_t - \theta_{t-1}\| \tag{25}$$

This inequality together with (21) and (22) proves the bound in the lemma. ☐

Now we are ready to prove Theorem 1.

Proof of Theorem 1:

$$\begin{aligned} &\mathrm{Reg}(OGD, \mathscr{L}_T) \le \\ &\sum_{t=1}^T \left(f_t(x_t) - f_t(x_t^*) + \beta/2\|x_t - x_{t-1}\|^2\right) \\ &\le \sum_{t=1}^T \left(f_t(x_t) - f_t(\theta_t) + \beta/2\|x_t - x_{t-1}\|^2\right) \\ &\le G \sum_{t=1}^T \|x_t - \theta_t\| + \sum_{t=1}^T \|\theta_t - \theta_{t-1}\| \frac{G\beta}{l(1-\kappa)} \\ &\le \frac{G}{(1-\kappa)} \sum_{t=1}^T \|\theta_t - \theta_{t-1}\| + \sum_{t=1}^T \|\theta_t - \theta_{t-1}\| \frac{G\beta}{l(1-\kappa)} \\ &= (\beta/l + 1)\frac{G}{(1-\kappa)} \sum_{t=1}^T \|\theta_t - \theta_{t-1}\| \le (\beta/l + 1)\frac{G}{(1-\kappa)} L_T \end{aligned}$$

The first inequality is by throwing away negative term $-\beta/2\|x_t^* - x_{t-1}^*\|^2$. The second one is because θ_t minimizes $f_t(x_t)$. The third one is from bounded gradient and Lemma 8. The last one is by (25) and combining like terms.

Then we have proved the upper bound on the dynamic regret by taking supremum on both sides of the inequality above. ☐

C. Proof of Lemma 4:

According to the construction, for any realization of θ_t, we have

$$\sum_{t=1}^T \|\theta_t - \theta_{t-1}\| = \sum_{k=0}^{K-1} \|\theta_{k\Delta+1} - \theta_{k\Delta}\| \le DK$$

In the following, we will show that $K \le L_T/D$, then the proof is done. Remember the definition of Δ:

$$\Delta = \lceil T/\lfloor L_T/D \rfloor \rceil \ge T/\lfloor L_T/D \rfloor$$

Equivalently, $\lfloor L_T/D \rfloor \geq T/\Delta$ Since $K = \lceil \frac{T}{\Delta} \rceil = \min\{i \in \mathbb{Z} |\ i \geq T/\Delta\}$, and $\lfloor L_T/D \rfloor \in \mathbb{Z}$, we have

$$K \leq \lfloor L_T/D \rfloor \leq L_T/D$$

□

D. Proof of Lemma 5

The proof takes four steps:

(I) study unconstrained optimization and show that $\tilde{x}^* = \arg\min_{\mathbb{R}^T} C_1^T(x) = A\theta$.

(II) show that the constrained optimization admits the same optimal solution: $x^* = A\theta$

(III) give closed-form expression for matrix A

(IV) lower bound the entries $a_{t,t+\tau}$ for $\tau \geq 0$ of matrix A

(I) Unconstrained optimization $\arg\min_{\mathbb{R}^T} C_1^T(x) = A\theta$**.** Remember that

$$\begin{aligned} C_1^T(x) &= \sum_{t=1}^{T} \left[f_t(x_t) + \frac{\beta}{2}\|x_t - x_{t-1}\|^2 \right] \\ &= \sum_{t=1}^{T} \left[\frac{\alpha}{2}\|x_t - \theta_t\|^2 + \frac{\beta}{2}\|x_t - x_{t-1}\|^2 \right] \end{aligned}$$

Notice that $C_1^T(x)$ is strongly convex, then the first order condition is a sufficient and necessary condition for the unconstrained optimization $\min_{\mathbb{R}^T} C_1^T(x)$:

$$\begin{aligned} &\alpha(x_t - \theta_t) + \beta(2x_t - x_{t-1} - x_{t+1}) = 0, \quad t \in [T-1] \\ &\alpha(x_T - \theta_T) + \beta(x_T - x_{T-1}) = 0 \end{aligned}$$

By $x_0 = \theta_0 = 0$ and canceling α on both sides, we can write the linear equation systems in the following matrix form:

$$Hx = \theta$$

where H is given as below:

$$H = \begin{pmatrix} 1+2\frac{\beta}{\alpha} & -\frac{\beta}{\alpha} & 0 & \cdots & 0 \\ -\frac{\beta}{\alpha} & 1+2\frac{\beta}{\alpha} & -\frac{\beta}{\alpha} & \cdots & 0 \\ 0 & -\frac{\beta}{\alpha} & 1+2\frac{\beta}{\alpha} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1+\frac{\beta}{\alpha} \end{pmatrix} \tag{26}$$

Notice that H is strictly diagonally dominant, so H is invertible. Therefore, the optimal solution to the unconstrained optimization, $\tilde{x}^* = \arg\min_{\mathbb{R}^T} C_1^T(x)$, is given by

$$\tilde{x}^* = A\theta \quad \text{where } A := H^{-1}$$

(II) The constrained optimization has the same solution. Since H is strictly diagonally dominant, then by Theorem 1 in [37], we have

$$\|A\|_\infty = \|H^{-1}\|_\infty \leq \max_{1 \leq t \leq T} \frac{1}{|h_{tt}| - \sum_{s \neq t} |h_{t,s}|} = 1$$

Besides, since H has negative off-diagonal entries and positive diagonal entries, and is strictly diagonally dominant, the inverse of H, denoted by A now, is nonnegative. Therefore, for each t, $\tilde{x}_t^*$ can be written as a convex combination of elements in X:

$$\tilde{x}_t^* = \sum_{s=1}^{T} a_{t,s}\theta_s + (1 - \sum_{s=1}^{T} a_{t,s})0$$

because $\theta_t,\ 0 \in X$. By convexity of X, we have $\tilde{x}_t^* \in X$, then naturally, $\tilde{x}^* \in X^T$. As a result, $x^* = \arg\min_{X^T} C_1^T(x) = \arg\min_{\mathbb{R}^T} C_1^T(x) = \tilde{x}^* = A\theta$.

(III) Closed form expression of A**.**
Since matrix H has many good properties, such as strictly diagonal dominance, positive diagonally entries and negative off-diagonal entries, tridiagonality, symmetry, we can find a closed-form expression for its inverse, denoted by A now, according to Theorem 2 in [38]. In particular, the entries of A are given by $a_{t,t+\tau} = \frac{\alpha}{\beta} u_t v_{t+\tau}$ for $\tau \geq 0$ where

$$\begin{aligned} u_t &= \frac{\rho}{1-\rho^2}\left(\frac{1}{\rho^t} - \rho^t\right) & v_T &= \frac{1}{-u_{T-1} + (\xi - 1)u_T} \\ v_t &= c_3 \frac{1}{\rho^{T-t}} + c_4 \rho^{T-t} & c_3 &= v_T \left(\frac{(\xi-1)\rho - \rho^2}{1-\rho^2}\right) \\ c_4 &= v_T \frac{1 - (\xi-1)\rho}{1-\rho^2} \end{aligned}$$

and $\rho = \frac{\sqrt{Q_f}-1}{\sqrt{Q_f}+1}$, $\xi = \alpha/\beta + 2$. Since A is nonnegative and u_t is apparently positive, we have $v_t \geq 0$ for all t.

(IV) Lower bound $a_{t,t+\tau}$ **for** $\tau \geq 0$**.**
We will bound u_t, v_T and $v_{t+\tau}/v_T$ separately and then combine them together for a lower bound of $a_{t,t+\tau}$ for $\tau \geq 0$.

First, we bound u_t by

$$\rho^t u_t = \frac{\rho}{1-\rho^2}(1 - \rho^{2t}) \geq \rho$$

since $t \geq 1$ and $\rho < 1$.

Next, we bound v_T in the following way:

$$\begin{aligned} \rho^{-T} v_T &= \frac{1}{(\xi-1)(1-\rho^{2T}) - (\rho - \rho^{2T-1})} \frac{1-\rho^2}{\rho} \\ &\geq \frac{1}{(\xi-1)(1-\rho^{2T})} \frac{1-\rho^2}{\rho} \geq \frac{1}{\xi - 1} \frac{1-\rho^2}{\rho} \end{aligned}$$

where $\xi = \frac{\alpha}{\beta} + 2 = \frac{2Q_f+2}{Q_f-1}$, $\rho = \frac{\sqrt{Q_f}-1}{\sqrt{Q_f}+1}$; the first inequality is by $T \geq 1$, $(\rho - \rho^{2T-1}) \geq 0$; the second inequality is by $0 < \rho < 1$.

Then, we bound $v_{t+\tau}$.

$$\begin{aligned} \rho^{T-t-\tau} \frac{v_{t+\tau}}{v_T} &= \left(\frac{(\xi-1)\rho - \rho^2}{1-\rho^2}\right) + \frac{1-(\xi-1)\rho}{1-\rho^2} \rho^{2(T-t-\tau)} \\ &\geq \left(\frac{(\xi-1)\rho - \rho^2}{1-\rho^2}\right) = \left(\frac{\rho^2 + 1 - \rho - \rho^2}{1-\rho^2}\right) = \left(\frac{1-\rho}{1-\rho^2}\right) \end{aligned}$$

where the inequality is by $1 - (\xi - 1)\rho \geq 0$, $v_T \geq 0$, and the second equality is by $\rho^2 - \xi\rho + 1 = 0$.

Finally, combining three parts together,

$$\begin{aligned} a_{t,t+\tau} &= \frac{\alpha}{\beta}\left[\rho^t u_t\right]\left[\rho^{-T} v_T\right]\left[\rho^{T-t-\tau}\frac{v_{t+\tau}}{v_T}\right]\rho^\tau \\ &\geq \frac{\alpha}{\beta}\rho \frac{1}{\xi-1}\frac{1-\rho^2}{\rho}\left(\frac{1-\rho}{1-\rho^2}\right)\rho^\tau = \frac{\alpha}{\alpha+\beta}(1-\rho)\rho^\tau \end{aligned}$$

E. Proof of Lemma 6

Proof. Define $a'_t = (a_{t,1},\dots,a_{t,T})$, $c'_t = (0,\dots,0,a_{t,t+W},\dots,a_{t,T})$, and $b'_t = a'_t - c'_t$. By Lemma 5, $x^*_t = a'_t\theta = b'_t\theta + c'_t\theta$.

$$\begin{aligned}
\mathbb{E}(x_t^{\mathscr{A}} - x_t^*)^2 &= \mathbb{E}(x_t^{\mathscr{A}} - b'_t\theta - c'_t\theta)^2\\
&= \mathbb{E}(x_t^{\mathscr{A}} - b'_t\theta)^2 + \mathbb{E}(c'_t\theta)^2 - 2\,\mathbb{E}\,c'_t\theta(x_t^{\mathscr{A}} - b'_t\theta)\\
&\ge \mathbb{E}(c'_t\theta)^2 - 2\,\mathbb{E}\,c'_t\theta(x_t^{\mathscr{A}} - b'_t\theta)\\
&= \mathbb{E}(c'_t\theta)^2 - 2\,\mathbb{E}(x_t^{\mathscr{A}} - b'_t\theta)\,\mathbb{E}\,c'_t\theta = \mathbb{E}(c'_t\theta)^2
\end{aligned}$$

where the second last equality is because $(x_t^{\mathscr{A}} - b'_t\theta)$ is determined by $\{\theta_s\}_{s=1}^{t+W-1}$, $c'_t\theta$ is determined by $\{\theta_s\}_{s=t+W}^{T}$, $\{\theta_s\}_{s=1}^{t+W-1}$ and $\{\theta_s\}_{s=t+W}^{T}$ are independent when $t \in J$; and the last equality is because $\mathbb{E}\,\theta_t = 0$ for each t.

Denote $c'_t = (c_{t,1},\dots,c_{t,T})$.

$$c'_t\theta = \sum_{s=1}^{T} c_{t,s}\theta_s = \sum_{k=0}^{K-1}\left(\sum_{s=k\Delta+1}^{\min(T,k\Delta+\Delta)} c_{t,s}\right)\theta_{k\Delta+1}$$

and $\theta_{k\Delta+1}$ are i.i.d. for $k = 0,\dots,K-1$ with zero mean and $\mathrm{var}(\theta_t) = D^2/4$. Thus,

$$\begin{aligned}
\mathbb{E}(c'_t\theta)^2 = \mathrm{var}(c'_t\theta) &= \sum_{k=0}^{K-1}\left(\sum_{s=k\Delta+1}^{\min(T,k\Delta+\Delta)} c_{t,s}\right)^2 \mathrm{var}(\theta_{k\Delta+1})\\
&= D^2/4\sum_{k=0}^{K-1}\left(\sum_{s=k\Delta+1}^{\min(T,k\Delta+\Delta)} c_{t,s}\right)^2 \ge \frac{a_{t,t+W}^2 D^2}{4}
\end{aligned}$$

where the last inequality is because c_t is nonnegative with the first $t+W-1$ entries being zero. ☐

F. Proof of Lemma 7:

Before the proof, we introduce a supportive lemma.

Lemma 9. *If $T \ge 2W$, and $L_T \ge 2D$, then*

$$\lfloor \frac{T-W}{\Delta} \rfloor \ge \frac{T-W}{2\Delta}$$

Proof. Notice that if $x \ge 1$, then $\lfloor x\rfloor \ge x/2$. Thus, all we need to show is that $\frac{T-W}{\Delta} \ge 1$, or equivalently $T - W \ge \Delta$.

Remember that $\Delta = \lceil T/\lfloor L_T/D\rfloor\rceil$. If we can show that $T - W \ge T/\lfloor L_T/D\rfloor$, then by the fact that $T-W$ is an integer, we have $T - W \ge \lceil T/\lfloor L_T/D\rfloor\rceil$.

Equivalently, we want show $\lfloor L_T/D\rfloor \ge \frac{T}{T-W}$. Notice that when $L_T \ge 2D$, we have $\lfloor L_T/D\rfloor \ge 2$. When $T \ge 2W > 0$, we have

$$\frac{T}{T-W} \le \frac{T}{T-T/2} = 2$$

Therefore, we have $\lfloor L_T/D\rfloor \ge \frac{T}{T-W}$. ☐

Proof of Lemma 7. Rewriting the definition of set J as

$$J = \{1 + W \le t \le T, \quad t \equiv 1 \pmod{\Delta}\}$$

Then we have

$$\begin{aligned}
|J| = \lceil T/\Delta\rceil - \lceil W/\Delta\rceil &\ge \lfloor \frac{T-W}{\Delta}\rfloor \ge \frac{T-W}{2\Delta}\\
&\ge \frac{1}{2}\frac{T-W}{T/\lfloor L_T/D\rfloor + 1} = \frac{1}{2}\lfloor L_T/D\rfloor\frac{T-W}{T+\lfloor L_T/D\rfloor}\\
&\ge \frac{1}{2}\lfloor L_T/D\rfloor\frac{T-T/2}{T+T} = \frac{1}{8}\lfloor L_T/D\rfloor\\
&\ge \frac{1}{8}\times\frac{2}{3}L_T/D = \frac{1}{12}L_T/D
\end{aligned}$$

where the first equality is straightforward after rewriting the set J, the first inequality is a property of floor and ceiling functons, the second inequality is by Lemma 9, the third inequality is by $\Delta = \lceil T/\lfloor L_T/D\rfloor\rceil \le T/\lfloor L_T/D\rfloor + 1$, the fourth inequality is by $T \ge 2W$ and $L_T \le DT$, and the last inequality is because $L_T/D \ge 2$, and $\lfloor x\rfloor \ge \frac{2}{3}x$ when $x \ge 2$. ☐

G. The remaining proof of Theorem 4

There are two scenarios to be discussed: $D \le L_T < 2D$, and $0 < L_T < D$. We do not consider $L_T = 0$ because it is trivial. The proof will still be based on constructing parameters for the parameterized quadratic function given by (19), but this time we will let the cost function changes only once because L_T is small.

Scenario 1: $D \le L_T < 2D$. When $W \ge 1, T \ge 2W \ge W+1$. For $1 \le t \le W$, let $\theta_t = 0$. At $t = W+1$, let θ_t following the distribution $\mathbb{P}(\theta_t = \frac{D}{2}) = \mathbb{P}(\theta_t = -\frac{D}{2}) = \frac{1}{2}$. For the rest, just copy the θ_{W+1}: $\theta_t = \theta_{W+1}$ for $W+2 \le t \le T$.

It is easy to verify that for any realization of θ, $\sum_{t=1}^{T}\|\theta_t - \theta_{t-1}\| = \|\theta_{W+1}\| = \frac{D}{2} \le \frac{L_T}{2} \le L_T$.

By Lemma 10 to be stated below, we have $\mathbb{E}\,\|x^{\mathscr{A}} - x^*\|^2 \ge \mathbb{E}\,\|x_1^{\mathscr{A}} - x_1^*\|^2 \ge \frac{a_{1,1+W}^2 D^2}{4}$. As a result, there must exist a sequence such that

$$\begin{aligned}
C_1^T(x^{\mathscr{A}}) - C_1^T(x^*) &\ge \frac{\alpha}{2}\|x^{\mathscr{A}} - x^*\|^2\\
&\ge \frac{\alpha D^2}{8}\rho^{2W}(1-\rho)^2\left(\frac{\alpha}{\alpha+\beta}\right)^2\\
&\ge \frac{\alpha D L_T}{96}\rho^{2W}(1-\rho)^2\left(\frac{\alpha}{\alpha+\beta}\right)^2
\end{aligned}$$

The proof is done.

Finally, we provide Lemma 10, which will be useful in $0 < L_T \le D$ scenario and Theorem 3's proof as well.

Lemma 10. *For any $W \ge 0$, consider the quadratic cost function* (19) *with a sequence of parameters θ satisfying: i) $\theta_1 = \cdots = \theta_W = 0$, ii) θ_{W+1} following distribution $\mathbb{P}(\theta_t = \frac{\nu}{2}) = \mathbb{P}(\theta_t = -\frac{\nu}{2}) = \frac{1}{2}$ for $0 < \nu \le D$, iii) $\theta_t = \theta_{W+1}$ for $W+2 \le t \le T$. Then, for any online algorithm $\mathscr{A}$, we have*

$$\mathbb{E}\,\|x_1^{\mathscr{A}} - x_1^*\|^2 \ge \frac{a_{1,1+W}^2\nu^2}{4}$$

Proof. The proof is very similar. Let $t = 1$. Define $a'_t = (a_{t,1},\dots,a_{t,T})$, $c'_t = (0,\dots,0,a_{t,t+W},\dots,a_{t,T})$, and $b'_t = a'_t - c'_t$. By Lemma 5, $x^*_t = a'_t\theta = b'_t\theta + c'_t\theta$.

$$\begin{aligned}
\mathbb{E}(x_t^{\mathscr{A}} - x_t^*)^2 &= \mathbb{E}(x_t^{\mathscr{A}} - b'_t\theta - c'_t\theta)^2\\
&= \mathbb{E}(x_t^{\mathscr{A}} - b'_t\theta)^2 + \mathbb{E}(c'_t\theta)^2 - 2\,\mathbb{E}\,c'_t\theta(x_t^{\mathscr{A}} - b'_t\theta)\\
&\ge \mathbb{E}(c'_t\theta)^2 - 2\,\mathbb{E}\,c'_t\theta(x_t^{\mathscr{A}} - b'_t\theta)\\
&= \mathbb{E}(c'_t\theta)^2 - 2\,\mathbb{E}(x_t^{\mathscr{A}} - b'_t\theta)\,\mathbb{E}\,c'_t\theta = \mathbb{E}(c'_t\theta)^2
\end{aligned}$$

where the second last equality is because $(x_t^{\mathscr{A}} - b_t'\theta)$ is determined by $\{\theta_s\}_{s=1}^{t+W-1}$, $c_t'\theta$ is determined by $\{\theta_s\}_{s=t+W}^{T}$, $\{\theta_s\}_{s=1}^{t+W-1}$ and $\{\theta_s\}_{s=t+W}^{T}$ are independent when $t=1$; and the last equality is because $\mathbb{E}\,\theta_t = 0$ for each t.

$$\mathbb{E}(c_1'\theta)^2 = \mathbb{E}(\sum_{s=1+W}^{T} a_{1,s}\theta_{1+W})^2 \geq \frac{\nu^2}{4} a_{1,1+W}^2$$

□

Scenario 2: $0 < L_T < D$. The proof will be same except that at $t = W+1$, let θ_t follow the distribution $\mathbb{P}(\theta_t = \frac{L_T}{2}) = \mathbb{P}(\theta_t = -\frac{L_T}{2}) = \frac{1}{2}$.

It is easy to verify that for any realization of θ, $\sum_{t=1}^{T} \|\theta_t - \theta_{t-1}\| = \|\theta_{W+1}\| = \frac{L_T}{2} \leq L_T$.

By Lemma 10, we can bound $\mathbb{E}\,\|x_1^{\mathscr{A}} - x_1^*\|^2 \geq \frac{a_{1,1+W}^2 L_T^2}{4}$. As a result, there must exist a sequence such that

$$\begin{aligned}
C_1^T(x^{\mathscr{A}}) - C_1^T(x^*) &\geq \frac{\alpha}{2}\|x^{\mathscr{A}} - x^*\|^2 \\
&\geq \frac{\alpha L_T^2}{8}\rho^{2W}(1-\rho)^2\left(\frac{\alpha}{\alpha+\beta}\right)^2 \\
&\geq \frac{\alpha}{96}(1-\rho)^2\left(\frac{\alpha}{\alpha+\beta}\right)^2 \rho^{2W} L_T^2
\end{aligned}$$

H. Proof of Theorem 3

Remember that $0 \leq L_T \leq DT$, so we will discuss two scenarios: $0 < L_T < D$, and $D \leq L_T \leq DT$ ($L_T = 0$ is trivially true), and construct different function sequences to prove the lower bound. The proof will be very similar to the proof of Theorem 4, we will first construct random sequence, then show that the lower bound holds in expectation. Without loss of generality, we let $x_0 = 0$.

Scenario 1: $0 < L_T < D$.

Construction of random costs. For each $0 < L_T < D$, we consider the following construction of $X \subseteq \mathbb{R}^2$:

$$X = [-\frac{L_T}{2}, \frac{L_T}{2}] \times [-\frac{\sqrt{D^2 - L_T^2}}{2}, \frac{\sqrt{D^2 - L_T^2}}{2}]$$

It is easy to verify that the diameter of X is D.

For any $\alpha > 0$, consider the parametrized cost function:

$$f_t(x_t, y_t; \tilde{x}_t, \tilde{y}_t) = \frac{\alpha}{2}(x_t - \tilde{x}_t)^2 + \frac{\alpha}{2}(y_t - \tilde{y}_t)^2$$

where $(\tilde{x}_t, \tilde{y}_t) \in \mathbb{R}^2$ are parameters which may be outside the action space X. It is easy to verify that $f_t(x_t, y_t; \tilde{x}_t, \tilde{y}_t)$ belongs to function class $\mathscr{F}_X(\alpha, \alpha, G)$, where $G = \alpha\sqrt{(M + D/2)^2 + D^2}$ when $\tilde{y}_t \in [-\frac{D}{2}, \frac{D}{2}]$ and $\tilde{x}_t \in [-M, M]$ for $M \geq D$.

Next, we construct two possible function sequences, and each sequence is true with probability 1/2.

Sequence 1: $\tilde{x}_1 = M$, $\tilde{x}_t = \frac{L_T}{2}$ for $t \geq 2$. $\tilde{y}_t = 0$, $t \in [T]$.

Sequence 2: $\tilde{x}_1 = -M$, $\tilde{x}_t = -\frac{L_T}{2}$ for $t \geq 2$. $\tilde{y}_t = 0$, $t \in [T]$.

where $M = D + (1 + \beta/\alpha)L_T/2$.

Let $(\theta_t, \varphi_t) = \arg\min_X f_t(x_t, y_t; \tilde{x}_t, \tilde{y}_t)$, and $(x^*, y^*) := (x_1, y_1, \ldots, x_T, y_T)' = \arg\min_{X^T} C_1^T(x, y)$. Then, for each sequence, we have

Sequence 1: $\theta_t = x_t^* = \frac{L_T}{2}$, $\varphi_t = y_t^* = 0$ for $1 \leq t \leq T$.

Sequence 2: $\theta_t = x_t^* = -\frac{L_T}{2}$, $\varphi_t = y_t^* = 0$ for $1 \leq t \leq T$.

Bound $\mathbb{E}[C_1^T(x^{\mathscr{A}}, y^{\mathscr{A}}) - C_1^T(x^*, y^*)]$**.**

By strong convexity, we have

$$\begin{aligned}
&\mathbb{E}[C_1^T(x^{\mathscr{A}}, y^{\mathscr{A}}) - C_1^T(x^*, y^*)] \\
&\geq \mathbb{E}\sum_{t=1}^{T}\left[\frac{\partial C_1^T}{\partial x_t}(x^*, y^*)(x_t^{\mathscr{A}} - x_t^*) + \frac{\partial C_1^T}{\partial y_t}(x^*, y^*)(y_t^{\mathscr{A}} - y_t^*)\right] \\
&+ \mathbb{E}\sum_{t=1}^{T}\left[\frac{\alpha}{2}\|x_t^{\mathscr{A}} - x_t^*\|^2 + \frac{\alpha}{2}\|y_t^{\mathscr{A}} - y_t^*\|^2\right] \\
&\geq \mathbb{E}\left[\frac{\partial C_1^T}{\partial x_1}(x^*, y^*)(x_1^{\mathscr{A}} - x_1^*)\right] + \mathbb{E}\left[\frac{\alpha}{2}\|x_1^{\mathscr{A}} - x_1^*\|^2\right]
\end{aligned}$$

where the last inequality is by $\frac{\partial C_1^T}{\partial y_t}(x^*, y^*) = 0$ when $t \geq 1$, and $\frac{\partial C_1^T}{\partial x_t}(x^*, y^*)(x_t^{\mathscr{A}} - x_t^*) = 0$ when $t \geq 2$.

Next, we will expand the expectation by considering two possible sequences:

$$\begin{aligned}
&\mathbb{E}[C_1^T(x^{\mathscr{A}}, y^{\mathscr{A}}) - C_1^T(x^*, y^*)] \\
&\geq \mathbb{E}\left[\frac{\partial C_1^T}{\partial x_1}(x^*, y^*)(x_1^{\mathscr{A}} - x_1^*)\right] + \mathbb{E}\left[\frac{\alpha}{2}\|x_1^{\mathscr{A}} - x_1^*\|^2\right] \\
&= \frac{1}{2}(-h)(x_1^{\mathscr{A}} - \frac{L_T}{2}) + \frac{1}{2}h(x_1^{\mathscr{A}} + \frac{L_T}{2}) + \frac{\alpha}{4}(x_1^{\mathscr{A}} - \frac{L_T}{2})^2 \\
&+ \frac{\alpha}{4}(x_1^{\mathscr{A}} + \frac{L_T}{2})^2 \geq \frac{1}{2}hL_T
\end{aligned}$$

where $h = \frac{\partial C_1^T}{\partial x_1}(x^*, y^*)$ when the costs follow Sequence 2 and $h = -\frac{\partial C_1^T}{\partial x_1}(x^*, y^*)$ when the costs follow Sequence 1. When $M = D + (1 + \beta/\alpha)\frac{L_T}{2}$, $h = \alpha D$ and

$$\mathbb{E}[C_1^T(x^{\mathscr{A}}, y^{\mathscr{A}}) - C_1^T(x^*, y^*)] \geq \frac{\alpha D}{2} L_T$$

Since $G = \alpha\sqrt{(M + D/2)^2 + D^2} \leq \alpha D\sqrt{(2 + \beta/(2\alpha))^2 + 1}$, it is easy to verify that our current lower bound by showing that

$$\begin{aligned}
\frac{\alpha D}{2} L_T &\geq \frac{GL_T}{2\sqrt{(2 + \beta/(2\alpha))^2 + 1}} \geq \frac{GL_T}{4(2 + \beta/(2\alpha))} \\
&\geq \frac{\alpha GL_T}{8(\alpha + \beta)} \geq \frac{GL_T}{32}(1-\rho)^2\left(\frac{\alpha}{\alpha+\beta}\right)^2
\end{aligned}$$

Scenario 2: $D \leq L_T \leq DT$. The proof will be identical to the proof of Theorem 4 in Section V-B except for one difference: when $W = 0$, we are able to give a better bound for $|J|$ even without the condition $L_T \geq 2D$. Notice that the condition $D \leq L_T \leq DT$ is still necessary for the construction of θ in Section V-B to be well-defined.

The bound for $|J|$ is given below.

Lemma 11. *If* $T \geq 1$*, and* $D \leq L_T \leq DT$*, then*

$$|J| \geq \frac{L_T}{4D}$$

Proof. By definition of J and $\Delta = \lceil T/\lfloor L_T/D \rfloor \rceil \leq T/\lfloor L_T/D \rfloor + 1$, when $L_T \geq D$ and $T \geq 1$, we have

$$\begin{aligned}
|J| &= \lceil \frac{T}{\Delta} \rceil \geq \frac{T}{\Delta} \geq \frac{T}{T/\lfloor L_T/D \rfloor + 1} \\
&= \lfloor L_T/D \rfloor \frac{T}{T + \lfloor L_T/D \rfloor} \geq \frac{L_T}{2D}\frac{T}{T+T} = \frac{L_T}{4D}
\end{aligned}$$

by $\lfloor x \rfloor \geq x/2$ when $x \geq 1$, and $L_T \leq DT$.

□

Then, by $G = \alpha D$, Lemma 11 and 5:

$$\mathbb{E}[C_1^T(x^{\mathscr{A}}) - C_1^T(x^*)] \geq \mathbb{E}\,\frac{\alpha}{2}\|x^{\mathscr{A}} - x^*\|^2$$

$$\geq \frac{\alpha}{2}\sum_{t\in J}\frac{a_{t,t}^2 D^2}{4} \geq \frac{GL_T}{32}(1-\rho)^2\left(\frac{\alpha}{\alpha+\beta}\right)^2$$