Online Optimization with Feedback Delay and Nonlinear Switching Cost

WEICI PAN, Stony Brook University, United States
GUANYA SHI, California Institute of Technology, United States
YIHENG LIN, California Institute of Technology, United States
ADAM WIERMAN, California Institute of Technology, United States

We study a variant of online optimization in which the learner receives k-round *delayed feedback* about hitting cost and there is a multi-step nonlinear switching cost, i.e., costs depend on multiple previous actions in a nonlinear manner. Our main result shows that a novel Iterative Regularized Online Balanced Descent (iROBD) algorithm has a constant, dimension-free competitive ratio that is $O(L^{2k})$, where L is the Lipschitz constant of the switching cost. Additionally, we provide lower bounds that illustrate the Lipschitz condition is required and the dependencies on k and L are tight. Finally, via reductions, we show that this setting is closely related to online control problems with delay, nonlinear dynamics, and adversarial disturbances, where iROBD directly offers constant-competitive online policies.

CCS Concepts: • **Computing methodologies** → **Online learning settings**.

Additional Key Words and Phrases: online learning; online optimization; online control

ACM Reference Format:
Weici Pan, Guanya Shi, Yiheng Lin, and Adam Wierman. 2022. Online Optimization with Feedback Delay and Nonlinear Switching Cost. *Proc. ACM Meas. Anal. Comput. Syst.* 6, 1, Article 17 (March 2022), 34 pages. https://doi.org/10.1145/3508037

1 INTRODUCTION

We study a variant of online convex optimization (OCO) with feedback delay and nonlinear switching (movement) cost. In recent years, the problem of online convex optimization with (linear) switching cost has received considerable attention, e.g., [4, 7, 11, 15, 31] and the references therein. In this setting an online learner iteratively picks an action y_t and then suffers a convex hitting cost $f_t(y_t)$ and a (linear) switching cost $c(y_t, y_{t-1}, \cdots, y_{t-p})$, depending on current and previous p actions. This type of online optimization with memory has deep connections to convex body chasing [5, 9, 10, 27] and has wide applications in areas such as power systems [6, 19, 21], electric vehicle charging [14, 19], cloud computing [11, 12, 25], and online control [16, 20, 22, 24, 31].

Our work aims to generalize the online convex optimization literature in two directions, motivated by two limitations of the classical setting that prevent applications of the results in some important situations. First, in the classical setting, the online learner observes the hitting cost function f_t *before* picking the action y_t. However, in many applications, such as trajectory tracking problems in robotics, f_t is revealed after a multi-round delay due to communication and process delays,

Authors' addresses: Weici Pan, weici.pan@stonybrook.edu, Stony Brook University, Stony Brook, New York, United States; Guanya Shi, gshi@caltech.edu, California Institute of Technology, Pasadena, United States; Yiheng Lin, yihengl@caltech.edu, California Institute of Technology, Pasadena, United States; Adam Wierman, adamw@caltech.edu, California Institute of Technology, Pasadena, United States.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
© 2022 Copyright held by the owner/author(s).
2476-1249/2022/3-ART17
https://doi.org/10.1145/3508037

i.e., multiple rounds of actions must be taken *without* feedback on their hitting costs. Delay is known to be very challenging in practice and [31] shows that even *one-step* delay requires non-trivial algorithmic modifications. The impact of multi-round delay has been recognized as a challenging open question for the design of online algorithms [18, 28, 31] and broadly in applications. For example, [30] highlights that a three-step delay (around 30 milliseconds) can already cause catastrophic crashes in drone tracking control using a standard controller without algorithmic adjustments for delay.

Second, the classical online convex optimization setting allows only *linear* forms of switching cost functions, where the switching cost c is some (squared) norm of a linear combination of current and previous actions, e.g., $y_t - y_{t-1}$ [13, 15, 16] or $y_t - \sum_{i=1}^{p} C_i y_{t-i}$ [31]. However, in many practical scenarios the costs to move from y_{t-1} to y_t are non-trivial nonlinear functions. For example, consider $y_t \in \mathbb{R}$ as the vertical velocity of a drone in a velocity control task. Hovering the drone (i.e., holding the position such that $y_t = 0, \forall t$) is not free, due to gravity. In this case, the cost to move from y_{t-1} to y_t is $(y_t - y_{t-1} + g(y_{t-1}))^2$ where the nonlinear term $g(y_{t-1})$ accounts for the gravity and aerodynamic drag [32]. Such non-linearities create significant algorithmic challenges because (i) in contrast to the linear setting, small movement between decisions does not necessarily imply small switching cost (e.g., the aforementioned drone control example), and (ii) a small error in a decision can lead to large non-linear penalties in the switching cost in future steps, which is further amplified by the multi-round delay. Addressing such challenges is well-known to be a challenging open question for the design of online algorithms.

Additional motivation for our focus on delay and nonlinear switching cost comes from the emerging connection between online convex optimization and online control. The notion of a switching cost in online optimization parallels the control cost in optimal control theory in that both characterize the cost to steer the state y_t. Inspired by this analogy, recent papers have used similar algorithms and techniques in the two settings, e.g., [1, 22], and shown reductions between online control and online convex optimization [1, 16, 20, 24, 29, 31, 33, 36]. These results are highly provocative – suggesting a deep connection – but also limited in terms of the generality of the control settings that can be considered. All the existing results focus on linear dynamical systems without delay. The question of how general a connection can be made between online control and online convex optimization remains unanswered. As we show in this paper, the incorporation of delay and nonlinear switching costs into online convex optimization significantly generalizes the set of control problems that can be reduced to online optimization.

1.1 Contributions

This paper addresses the three open questions highlighted above. We provide the first competitive algorithm for online convex optimization with feedback delay and nonlinear switching costs, and show a reduction between a class of nonlinear online control models with delay and online convex optimization with feedback delay and nonlinear switching cost.

More specifically, we propose a novel setting of online optimization where the hitting cost suffers k-round delayed feedback and the switching cost is nonlinear. This setting generalizes prior work on online convex optimization with switching costs (e.g., [15, 31]). In this setting, we propose a new algorithm, Iterative Regularized Online Balanced Descent (iROBD) and prove that it maintains a dimension-free constant competitive ratio that is $O(L^{2k})$, where L is the Lipschitz constant of the non-linear switching cost and k is the delay. This is the first constant competitive algorithm in the case of either feedback delay or nonlinear switching cost and we show, via lower bounds, that the dependencies on both L and k are tight. These lower bounds further serve to emphasize how the algorithmic difficulties increase as the length of delay becomes longer.

The design of iROBD deals with the k-round delay via a novel iterative process of estimating the unknown cost function optimistically, i.e., iteratively assuming that the unknown cost functions will lead to minimal cost for the algorithm. This approach is different than a one-shot optimistic approach focused on the whole trajectory of unknown cost functions, and the iterative nature is crucial for bounding the competitive ratio. In particular, the key idea to our competitive ratio proof is to bound the error that accumulates in the iterations by leveraging a Lipschitz property on the nonlinear component of the switching cost. This analytic approach is novel and a contribution in its own right.

Finally, we show that iROBD is constant competitive for the control of linear dynamical systems with squared costs and general adversarial disturbances as well as a class of nonlinear dynamics, via a novel reduction between such systems and online optimization with feedback delay and nonlinear switching costs. This reduction represents a significant generalization of the results in [16, 31], which each have significant limitations on the dynamics where they apply. Our new reduction highlights that state disturbances can be connected to delay and nonlinear dynamics can be connected to nonlinear switching costs; thus highlighting the difficulties each creates for competitive control.

2 MODEL AND PRELIMINARIES

Online convex optimization with memory has emerged as an important and challenging area with a wide array of applications, see [1, 3, 10, 13, 15, 23] and the references therein. Many results in this area have focused on the case of online optimization with switching costs (movement costs), a form of one-step memory, e.g., [10, 13, 15], though some papers have focused on more general forms of memory, e.g., [1, 3]. In this paper we, for the first time, study the impact of feedback delay and nonlinear switching cost in online optimization with switching costs.

An instance consists of a convex action set $\mathcal{K} \subset \mathbb{R}^d$, an initial point $y_0 \in \mathcal{K}$, a sequence of non-negative convex cost functions $f_1, \cdots, f_T : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, and a switching cost $c : \mathbb{R}^{d\times(p+1)} \to \mathbb{R}_{\geq 0}$. To incorporate feedback delay, we consider a situation where the online learner only knows the geometry of the hitting cost function at each round, i.e., f_t, but that the minimizer of f_t is revealed only after a delay of k steps, i.e., at time $t + k$. This captures practical scenarios where the form of the loss function or tracking function is known by the online learner, but the target moves over time and measurement lag means that the position of the target is not known until some time after an action must be taken. To incorporate nonlinear (and potentially nonconvex) switching costs, we consider the addition of a known nonlinear function δ from $\mathbb{R}^{d\times p}$ to $\mathbb{R}^d$ to the structured memory model introduced previously. Specifically, we have

$$c(y_{t:t-p}) = \frac{1}{2}\|y_t - \delta(y_{t-1:t-p})\|^2, \tag{1}$$

where we use $y_{i:j}$ to denote either $\{y_i, y_{i+1}, \cdots, y_j\}$ if $i \leq j$, or $\{y_i, y_{i-1}, \cdots, y_j\}$ if $i > j$ throughout the paper. Additionally, we use $\|\cdot\|$ to denote the 2-norm of a vector or the spectral norm of a matrix.

In summary, we consider an online agent that interacts with the environment as follows:

(1) The adversary reveals a function h_t, which is the geometry of the t^{th} hitting cost, and a point v_{t-k}, which is the minimizer of the $(t-k)^{\text{th}}$ hitting cost. Assume that h_t is m-strongly convex and l-strongly smooth, and that $\arg\min_y h_t(y) = 0$.
(2) The online learner picks y_t as its decision point at time step t after observing h_t, v_{t-k}.
(3) The adversary picks the minimizer of the hitting cost at time step t: v_t.
(4) The learner pays hitting cost $f_t(y_t) = h_t(y_t - v_t)$ and switching cost $c(y_{t:t-p})$ of the form (1).

The goal of the online learner is to minimize the total cost incurred over T time steps, $cost(ALG) = \sum_{t=1}^{T} f_t(y_t) + c(y_{t:t-p})$, with the goal of (nearly) matching the performance of the offline optimal algorithm with the optimal cost $cost(OPT)$. The performance metric used to evaluate an algorithm is typically the *competitive ratio* because the goal is to learn in an environment that is changing dynamically and is potentially adversarial. Formally, the competitive ratio (R) of the online algorithm is defined as the worst-case ratio between the total cost incurred by the online learner and the offline optimal cost: $R(ALG) = \sup_{f_{1:T}} \frac{cost(ALG)}{cost(OPT)}$.

Other than competitive ratio, there are two different types of regret that are often used as the performance metric in online learning. One is dynamic regret (competitive difference), which is defined as $\sup_{f_{1:T}} cost(ALG) - cost(OPT)$. Similar to competitive ratio, dynamic regret competes against the offline optimal thus it is also a global guarantee. However, results for the dynamic regret depend on the path-length or variation budget [20, 21], not just system properties. Bounding the competitive ratio is typically more challenging, and a dynamic regret bound can be implied by a competitive ratio bound while the converse is not true in general. On the other hands, (static) regret competes against the best fixed action y^* in hindsight [3], which is a weaker guarantee than competitive ratio/difference. In particular, when used in online control problems, the comparator in regret corresponds to the best static policy in a specific policy class, typically the linear policy class [1, 2]. However, for online control problems with delay, time-varying or nonlinear dynamics, the optimal policy is often nonlinear and time-varying, so regret may not be a proper metric in these cases. Especially, the suboptimality gap between the best static policy and the true offline optimal policy could be arbitrarily large [31].

It is important to emphasize that the online learner decides y_t based on the knowledge of the previous decisions $y_1 \cdots y_{t-1}$, the geometry of cost functions $h_1 \cdots h_t$, and the delayed feedback on the minimizer $v_1 \cdots v_{t-k}$. Thus, the learner has perfect knowledge of cost functions $f_1 \cdots f_{t-k}$, but incomplete knowledge of $f_{t-k+1} \cdots f_t$ (recall that $f_t(y) = h_t(y - v_t)$).

Both feedback delay and nonlinear switching cost add considerable difficulty for the online learner compared to versions of online optimization studied previously. Delay hides crucial information from the online learner and so makes adaptation to changes in the environment more challenging. As the learner makes decisions it is unaware of the true cost it is experiencing, and thus it is difficult to track the optimal solution. This is magnified by the fact that nonlinear switching costs increase the dependency of the variables on each other. It further stresses the influence of the delay, because an inaccurate estimation on the unknown data, potentially magnifying the mistakes of the learner.

The impact of feedback delay has been studied previously in online learning settings without switching costs, with a focus on regret, e.g., [18, 28]. However, in settings with switching costs the impact of delay is magnified since delay may lead to not only more hitting cost in individual rounds, but significantly larger switching costs since the arrival of delayed information may trigger a very large chance in action. To the best of our knowledge, we give the first competitive ratio for delayed feedback in online optimization with switching costs.

We illustrate a concrete example application of our setting in the following.

Example 2.1 (Drone tracking problem). Consider a drone with vertical speed $y_t \in \mathbb{R}$. The goal of the drone is to track a sequence of desired speeds $y_1^d, \cdots, y_T^d$ with the following tracking cost:

$$\sum_{t=1}^{T} \frac{1}{2}(y_t - y_t^d)^2 + \frac{1}{2}(y_t - y_{t-1} + g(y_{t-1}))^2, \tag{2}$$

where $g(y_{t-1})$ accounts for the gravity and the aerodynamic drag. One example is $g(y) = C_1 + C_2 \cdot |y| \cdot y$ where $C_1, C_2 > 0$ are two constants [32]. Note that the desired speed y_t^d is typically sent from a remote computer/server. Due to the communication delay, at time step t the drone only knows $y_1^d, \cdots, y_{t-k}^d$.

Algorithm 1 ROBD [15]

Parameter: $\lambda_1 \geq 0, \lambda_2 \geq 0$
for $t = 1$ **to** T **do**
  Input: Hitting cost function f_t, previous decision points $y_{t-p:t-1}$
  $v_t \leftarrow \arg\min_y f_t(y)$
  $y_t \leftarrow \arg\min_y f_t(y) + \lambda_1 c(y, y_{t-1:t-p}) + \frac{\lambda_2}{2}\|y - v_t\|_2^2$
  Output: y_t
end for

This example is beyond the scope of existing results in online optimization, e.g., [15, 16, 31], because of (i) the k-step delay in the hitting cost $\frac{1}{2}(y_t - y_t^d)$ and (ii) the nonlinearity in the switching cost $\frac{1}{2}(y_t - y_{t-1} + g(y_{t-1}))^2$ with respective to y_{t-1}. However, in this paper, because we directly incorporate the effect of delay and nonlinearity in the algorithm design, our algorithms immediately provide constant-competitive policies for this setting.

2.1 Related work

This paper contributes to the growing literature on online convex optimization with memory. Initial results in this area focused on developing constant-competitive algorithms for the special case of 1-step memory, a.k.a., the Smoothed Online Convex Optimization (SOCO) problem, e.g., [13, 15]. In that setting, [13] was the first to develop a constant, dimension-free competitive algorithm for high-dimensional problems. The proposed algorithm, Online Balanced Descent (OBD), achieves a competitive ratio of $3 + O(1/\beta)$ when cost functions are β-locally polyhedral. This result was improved by [15], which proposed two new algorithms, Greedy OBD and Regularized OBD (ROBD), that both achieve $1 + O(m^{-1/2})$ competitive ratios for m-strongly convex cost functions. Recently, [31] gave the first competitive analysis that holds beyond one step of memory. It holds for a form of structured memory where the switching cost is linear: $c(y_{t:t-p}) = \frac{1}{2}\|y_t - \sum_{i=1}^{p} C_i y_{t-i}\|^2$, with known $C_i \in \mathbb{R}^{d\times d}$, $i = 1, \cdots, p$. If the memory length $p = 1$ and C_1 is an identity matrix, this is equivalent to SOCO. In this setting, [31] shows that ROBD has a competitive ratio of

$$\frac{1}{2}\left(1 + \frac{\alpha^2 - 1}{m} + \sqrt{\left(1 + \frac{\alpha^2 - 1}{m}\right)^2 + \frac{4}{m}}\right), \tag{3}$$

when hitting costs are m-strongly convex and $\alpha = \sum_{i=1}^{p} \|C_i\|$.

Prior to this paper, competitive algorithms for online optimization have nearly always assumed that the online learner acts *after* observing the cost function in the current round, i.e., have zero delay. The only exception is [31], which considered the case where the learner must act before observing the cost function, i.e., a one-step delay. Even that small addition of delay requires a significant modification to the algorithm (from ROBD to Optimistic ROBD) and analysis compared to previous work.

As the above highlights, there is no previous work that addresses either the setting of nonlinear switching costs nor the setting of multi-step delay. However, the prior work highlights that ROBD is a promising algorithmic framework and our work in this paper extends the ROBD framework in order to address the challenges of delay and non-linear switching costs. Given its importance to our work, we describe the workings of ROBD in detail in Algorithm 1.

Another line of literature that this paper contributes to is the growing understanding of the connection between online optimization and adaptive control. The reduction from adaptive control to online optimization with memory was first studied in [1] to obtain a sublinear static regret

Algorithm 2 Iterative ROBD (iROBD)

1: **Parameter:** $\lambda \geq 0$
2: Initialize a ROBD instance with $\lambda_1 = \lambda$, $\lambda_2 = 0$
3: **for** $t = 1$ **to** T **do**
4: **Input:** h_t, v_{t-k}
5: Observe $f_{t-k}(y) = h_{t-k}(y - v_{t-k})$
6: $\hat{y}_{t-k} = \text{ROBD}(f_{t-k}, \hat{y}_{t-k-p:t-k-1})$
7: Initialize a temporary sequence $s_{1:t}$
8: $s_{1:t-k} \leftarrow \hat{y}_{1:t-k}$
9: **for** $i = t - k + 1$ **to** t **do**
10: $\tilde{v}_i = \arg\min_v \min_y h_i(y - v) + \lambda c(y, s_{i-1:i-p})$
11: Set $\tilde{f}_i(y) = h_i(y - \tilde{v}_i)$
12: $s_i \leftarrow \text{ROBD}(\tilde{f}_i, s_{i-p:i-1})$
13: **end for**
14: $y_t = s_t$
15: **Output:** y_t (the action at time step t)
16: **end for**

guarantee against the best linear state-feedback controller, where the approach is to consider a disturbance-action policy class with some fixed horizon. Many follow-up works adopt similar reduction techniques [2, 8, 17]. A different reduction approach using control canonical form is proposed by [20] and further exploited by [31]. Our work falls into this category. The most general results so far focus on Input-Disturbed Squared Regulators, which can be reduced to online convex optimization with structured memory (without delay or nonlinear switching costs). As we show in Section 5, the addition of delay and nonlinear switching costs leads to a significant extension of the generality of control models that can be reduced to online optimization.

3 A COMPETITIVE ALGORITHM

The main contribution of this paper is the first competitive algorithm for online convex optimization with multi-step delay and nonlinear switching costs. We introduce a new algorithm, Iterative Regularized Online Balanced Descent (iROBD, see Algorithm 2) that builds on ideas on ROBD and Optimistic ROBD in order to provide competitive guarantees in a significantly more general and challenging setting. The iROBD algorithm begins from $\hat{y}_{1:T}$, an oracle decision sequence from ROBD where there is no delay. Note that even though ROBD has two parameters λ_1 and λ_2, the latter is for practical implementation and redundant in theory. In this way, we use the setting where $\lambda_1 = \lambda$ and $\lambda_2 = 0$, denoting this implementation as ROBD(λ). The algorithmic goal is to make sure the actual decision sequence under delays $y_{1:T}$ stays close to the oracle one. Recall that at time step t, after observing h_t, v_{t-k}, the available information contains the perfect knowledge of hitting costs $f_{1:t-k}$ and the geometry of unknown hitting costs $f_{t-k+1:t}$, i.e., $h_{t-k+1:t}$. Therefore, the first step of iROBD is to recover what ROBD would do at time step $t - k$ as if it knew f_{t-k} at that time (Line 6). Given $\hat{y}_{1:t-k}$, the next step is to estimate unknown hitting costs $f_{t-k+1:t}$ (Line 7-13). To do this, iROBD initializes a temporary sequence $s_{1:t}$ which replicates the known part of the oracle sequence, i.e., $s_{1:t-k} = \hat{y}_{1:t-k}$. Then, iROBD iteratively estimates the unknown hitting costs optimistically (Line 9-13), i.e., for each $t - k + 1 \leq i \leq t$, it estimates the unknown hitting cost function such that running the ROBD algorithm on the temporary sequence would give the smallest cost. Note that, in the first loop $i = t - k + 1$, the memory $s_{i-1:i-p}$ is the same as the oracle sequence but, in later loops ($i = t - k + 2, \cdots, t$), the memory contains estimations from previous

iterations. After the last iteration ($i = t$), we take s_t as the output action/decision (Line 14). Figure 1 depicts the evolution of $s_{1:t}$ in the iterative process.

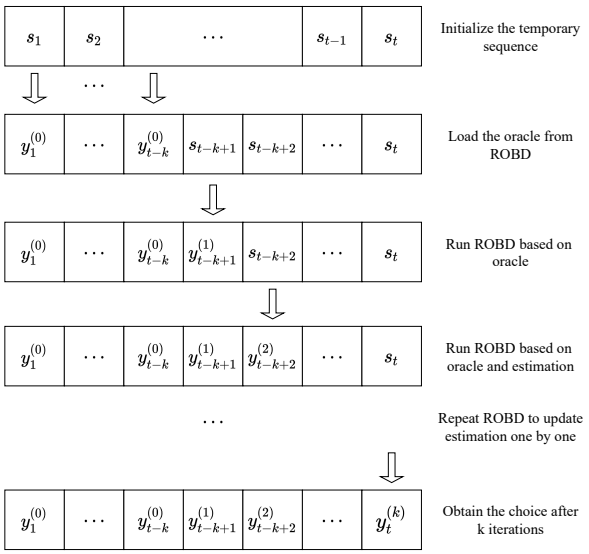

Fig. 1. The evolution of the sequence $s_{1:t}$ in iROBD.

We define some useful notations here. We use a superscript $^{(i)}$ to denote the decision of iROBD in the setting of i-step delay. For example, $y_t^{(k)}$ and $y_t^{(0)}$ are decisions of the algorithm at time t in settings of k-step delay and no delay respectively. Note that $\hat{y}_t = y_t^{(0)}$ denotes the oracle decision of ROBD without delay. Similarly, $v_t^{(k)}$ is the estimation of the algorithm on the minimizer at time t in the setting of k-step delay, while $v_t^{(0)}$ is that in the setting of no delay, that is, the exact minimizer v_t. For example, $v_t^{(1)} = \arg\min_v \min_y h_t(y-v) + \lambda c(y, y_{t-1:t-p}^{(0)})$, $v_t^{(2)} = \arg\min_v \min_y h_t(y-v) + \lambda c(y, y_{t-1}^{(1)}, y_{t-2:t-p}^{(0)})$, and $v_t^{(3)} = \arg\min_v \min_y h_t(y-v) + \lambda c(y, y_{t-1}^{(2)}, y_{t-2}^{(1)}, y_{t-3:t-p}^{(0)})$ and so on. Therefore, we have $f_t^{(i)}(y) = h_t(y - v_t^{(i)}) = f_t(y - v_t^{(i)} + v_t^{(0)})$ to be the estimated hitting cost of i-step-delay iROBD at time step t. And we can immediately get that $y_t^{(i)} = ROBD(f_t^{(i)}, y_{t-1}^{(i-1)}, y_{t-2}^{(i-2)}, \cdots)$. Moreover, let y_t^* denote the offline optimal decision at time step t.

Additionally, the following notation also denote the hitting cost and switching cost: $H_t^{(i)} := f_t(y_t^{(i)})$, and $M_t^{(i)} := c(y_{t:t-p}^{(i)})$. Similarly, $H_t^* := f_t(y_t^*)$, and $M_t^* := c(y_{t:t-p}^*)$. Our goal is to bound the competitive ratio, i.e.,

$$\frac{\sum_{t=1}^{T}\left(H_t^{(k)} + M_t^{(k)}\right)}{\sum_{t=1}^{T}\left(H_t^* + M_t^*\right)}.$$

It is worth noting that this iterative estimation process is different than a one-shot optimistic approach on the whole trajectory (which is the most natural extension of the ideas in Optimistic ROBD to multi-step delay), where the online learner optimistically estimates the unknown hitting cost function $f_{t-k+1:t}$ jointly. Intuitively, this is because a per-step greedy policy does not follow the true optimal trajectory in hindsight. This difference, and in particular the iterative nature of the algorithm, is novel and is crucial to achieving a constant competitive ratio.

In the remainder of this section, we first show a general competitive bound on iROBD in the case of nonlinear switching cost and then we probe the tightness of the general bound by considering the special case of linear switching cost.

3.1 Main Result: Nonlinear switching cost with feedback delay

Our main result (Theorem 3.1) shows that iROBD is a constant competitive algorithm under Lipschitz constraints on the nonlinear switching cost. Following the result we show that the dependence on delay and the nonlinear switching costs are both tight and that the Lipschitz constraints are necessary.

Theorem 3.1. *Suppose the hitting costs are m-strongly convex and l-strongly smooth, and the switching cost is given by $c(y_{t:t-p}) = \frac{1}{2}\|y_t - \delta(y_{t-1:t-p})\|^2$, where $\delta : \mathbb{R}^{d\times p} \to \mathbb{R}^d$. If there is a k-round-delayed feedback on the minimizers, and for any $1 \le i \le p$ there exists a constant $L_i > 0$, such that for any given $y_{t-1}, \cdots, y_{t-i-1}, y_{t-i+1}, \cdots, y_{t-p} \in \mathbb{R}^d$, we have:*

$$\|\theta(a) - \theta(b)\| \le L_i\|a - b\|, \forall a, b \in \mathbb{R}^d,$$

where $\theta(x) = \delta(y_{t-1}, \cdots, y_{t-i-1}, x, y_{t-i+1}, \cdots, y_{t-p})$, then the competitive ratio of iROBD(λ) is bounded by

$$O\left((l + 2p^2L^2)^k \max\left\{\frac{1}{\lambda}, \frac{m+\lambda}{m + (1-p^2L^2)\lambda}\right\}\right),$$

where $L = \max_i\{L_i\}$, $\lambda > 0$ and $m + (1 - p^2L^2)\lambda > 0$.

A detailed proof of Theorem 3.1 is given in Section 4. Theorem 3.1 highlights the contrasting impact of memory, feedback, and nonlinear switching cost on the competitive ratio. Interestingly, feedback delay (k) leads to an exponential degradation of the competitive ratio while memory (p) and the Lipschitz constant of the nonlinear switching cost (L) impact the competitive ratio only in a polynomial manner.

Our proof exploits the iterative structure of the algorithm, and the key idea is to bound the estimation errors that accumulate from the iterations. Note that this is something that is not necessary in the proofs of competitive ratios in ROBD and Optimistic ROBD, e.g., see [31], and leads to significant challenges.

The crucial point about the theorem above is that the competitive ratio does not depend on the time horizon T or the dimension d; it only depends on the delay, the memory structure, the convexity and smoothness of the cost functions, and the parameter the algorithm chooses. This means that iROBD is constant-competitive for OCO with feedback delay and structured memory. It is also interesting to observe that the competitive ratio is exponential in the delay k, which highlights that the bound grows quickly as delay grows. We show later that this is tight, which emphasizes the challenge delay creates for learning.

Note that, throughout our work the results are based on an assumption of time-invariant delay. If the delay is time-varying, our algorithm only needs a slight modification. See Appendix A for the detailed algorithm and theoretical justification. Another interesting variation is to consider the delay in the geometry of the hitting costs, not just the minimizer. Under the assumption that the hitting cost functions are well-conditioned, a straightforward way to generalize our algorithm to unknown geometry is the following: Assume the strongly convex constant is m, and the smooth constant is l. We can estimate the geometry is given by $\frac{m}{2}\|x\|^2$. The true cost, compared with the estimated cost, is off up to a factor of l/m. Therefore, the competitive ratio will be deteriorated by at most a factor of l/m.

The remainder of this section provides insight into the structure and tightness of Theorem 3.1. First, we highlight the case where the memory is of length one ($p = 1$) and there is no delay. This corresponds to SOCO with a nonlinear switching cost, a setting which has not been considered previously. The corollary below specializes Theorem 3.1 to this setting. Note that, because there is no delay, iROBD is simply ROBD.

Corollary 3.2. *Suppose the hitting costs are m-strongly convex and the switching cost is given by $\frac{1}{2}\|y_t - y_{t-1} - \delta(y_{t-1})\|^2$, where $\delta : \mathbb{R}^d \to \mathbb{R}^d$. If there exists a constant L such that*

$$\|\delta(a) - \delta(b)\| \leq L\|a - b\|, \ \forall a, b \in \mathbb{R}^d,$$

then the competitive ratio of ROBD(λ) is upper bounded by

$$\max\left\{\frac{1}{\lambda}, \frac{m+\lambda}{m - L(L+2)\lambda}\right\},$$

where $\lambda > 0$ and $m - L(L+2)\lambda > 0$. If $\lambda = \frac{-(m+2L+L^2)+\sqrt{(m+2L+L^2)^2+4m}}{2}$, then the upper bound is

$$\frac{1}{2}\left(1 + \frac{2L+L^2}{m} + \sqrt{\left(1 + \frac{2L+L^2}{m}\right)^2 + \frac{4}{m}}\right).$$

The result in the corollary is of particular interest because it is possible to construct a simple example showing a matching lower bound in this setting, highlighting the tightness of the analysis. Thus, the optimality of ROBD, which has been proven previously for linear switching costs [15], extends to settings with nonlinear switching costs. Notice that setting of this corollary includes many practical applications, e.g., the drone example in Example 2.1.

3.1.1 Globally or Locally Lipschitz. To discuss the necessity of the Lipschitz assumptions on the function δ in Theorem 3.1 and Corollary 3.2, we remark on the following two cases where the Lipschitz condition is violated globally or locally.

Remark 1. *Consider hitting costs that are m-strongly convex and switching costs given by $c(y_t, y_{t+1}) = \frac{1}{2}\|y_t - y_{t-1} - \delta(y_{t-1})\|^2$, where $\delta : \mathbb{R}^d \to \mathbb{R}^d$. Then there exists a δ function with Lipschitz constant L such that the competitive ratio of any online algorithm is lower bounded by*

$$\frac{1}{2}\left(1 + \frac{2L+L^2}{m} + \sqrt{\left(1 + \frac{2L+L^2}{m}\right)^2 + \frac{4}{m}}\right),$$

which exactly matches the upper bound in Corollary 3.2. One achieves this bound by setting $\delta(y) = Ly$. The derivation is included in Appendix F. Note that the lower bound is of the order L^2. Thus, as L becomes larger the competitive ratio grows unboundedly. So, if δ is very sensitive to small changes the competitive ratio can be very large.

Note that if the Lipschitz constraint in Theorem 3.1 is not satisfied then one cannot hope to obtain a constant competitive guarantee for any algorithm, even if the magnitude of δ is arbitrarily small, as we highlight below.

Having talked about a very high global Lipschitz constant, next we are going to show that the competitive ratio would explode even when the Lipschitz constant is high in a very small interval.

Remark 2. *Consider a 1-dimensional setting ($d = 1$) with hitting cost $(y_t - v_t)^2$ and switching cost $(y_t - y_{t-1} - \delta(y_{t-1}))^2$, where*

$$\delta(y) = \begin{cases} \epsilon, & y \le n\epsilon \\ -\epsilon \cdot \sin\left(\frac{\pi}{\gamma\epsilon} y - \frac{n\pi}{\gamma} - \frac{\pi}{2}\right), & n\epsilon < y \le n\epsilon + \gamma\epsilon \\ -\epsilon, & y > n\epsilon + \gamma\epsilon \end{cases}$$

with $n \in \mathbb{N}^+$ given in advance. Here, $\max_y |\delta(y)| = \epsilon$ and δ has Lipschitz constant $L = \frac{\pi}{\gamma}$, which can be unboundedly large when γ is small. In Appendix G we show that the cost of any online algorithm is no smaller than $2\epsilon^2$ in this setting. Additionally, we show that the cost of the offline optimal is no larger than $3\gamma\epsilon^2$. Thus, the competitive ratio of any online algorithm is no smaller than $\frac{2}{3\gamma}$. By taking γ arbitrarily small, the competitive ratio can become arbitrarily large. A detailed proof can be found in Appendix G.

Contrasting the Remark 1 and Remark 2, we can see that the Lipschitz assumptions in Theorem 3.1 and Corollary 3.2 are necessary to get a bounded competitive ratio, not artificial consequences of the proof approach.

3.2 Tightness: Linear switching cost

To further explore the tightness of Theorem 3.1, we now consider the special case of linear switching cost, i.e., where $\delta(y_{t-l:t-p}) = \sum_{i=1}^{p} C_i y_{t-i}$. In this setting, we can not only provide an upper bound on the performance of iROBD, but can also show a matching lower bound in terms of the dependency on delay.

Note that the case of linear switching cost is also of interest in its own right. This case corresponds to online convex optimization with feedback delay and structured memory, i.e., $c(y_{t:t-p}) = \frac{1}{2}\|y_t - \sum_{i=1}^{p} C_i y_{t-i}\|^2$, a setting that captures, for example, trajectory tracking problems in discrete time. Consider Equation (20) in Example 2 with $g(x_t) = 0$, where a vehicle is tracking some moving object with locations $v_{1:T}$. At each time step t, the vehicle measures v_t and takes a move u_t. Due to the communication and process delay from the sensor, the vehicle cannot accurately measure v_t in time. Instead, v_t is measured at time $t + k$.

Our first result here is the following upper bound.

Theorem 3.3. *Suppose the hitting costs are m-strongly convex and l-strongly smooth, and the switching cost is given by $c(y_{t:t-p}) = \frac{1}{2}\|y_t - \sum_{i=1}^{p} C_i y_{t-i}\|^2$, where $C_i \in \mathbb{R}^{d\times d}$ and $\alpha = \sum_{i=1}^{p} \|C_i\|$. If there is a k-round feedback delay, then the competitive ratio of iROBD(λ) is*

$$O\left((l + 2\alpha^2)^k \max\left\{\frac{1}{\lambda}, \frac{m+\lambda}{m + (1-\alpha^2)\lambda}\right\}\right), \tag{4}$$

where $\lambda > 0$ and $m + (1 - \alpha^2)\lambda > 0$.

A proof of Theorem 3.3 is given in Appendix C. The bound provided in this theorem resembles that in Theorem 3.1, but with C_i instead of L, making it tighter. Note that obtaining tighter results in this case requires a different proof technique.

Like in the nonlinear setting, the result for linear switching cost also displays an exponential dependency on delay. Thus, one may wonder if this dependence is a function of the algorithm or if it is fundamental. The lower bound result that follows shows that it is fundamental.

Theorem 3.4. *Consider hitting costs that are both m-strongly convex and m-strongly smooth, and switching cost given by $c(y_t, y_{t-1}) = \frac{1}{2}\|y_t - \alpha y_{t-1}\|^2$. If there is a k-round feedback delay and $\alpha > 1$, then the competitive ratio of any online algorithm is lower bounded by $\frac{m(\alpha^{2k}-1)}{\alpha^2-1}$.*

In the study of no-regret online learning *without* switching costs, delay influences regret bounds in a polynomial way, instead of exponentially [18, 28]. The contrast provided by the above result highlights that the existence of switching costs (which gives more power to the adversary) and the stronger metric (competitive ratio) makes the impact of delay significantly more dramatic. However, it is also interesting to note that the exponential impact of delay is consistent with what is proven [35] for online control in linear systems, which gives a competitive ratio lower bound $\Omega(\|A\|^k)$ for online control with k steps of delay. Nevertheless, the proof techniques in [35] cannot be applied to our setting, since [35] focuses on the setting with time-invariant cost functions, and highly relies on the structure of the underlying Linear Quadratic Regulator (LQR) online control problem.

Perhaps surprisingly, for the special case of $c(y_t, y_{t-1}) = \frac{1}{2}\|y_t - y_{t-1}\|^2$ it is possible to break through the exponential dependence on delay, as our final result of the section shows. This case corresponds to the original setting considered in the SOCO literature, e.g., [15, 16], with the addition of feedback delay.

Theorem 3.5. *Consider hitting costs that are both m-strongly convex and l-strongly smooth, and the switching costs given by $c(y_t, y_{t-1}) = \frac{1}{2}\|y_t - y_{t-1}\|^2$. When there is a k-round feedback delay, there exists an online algorithm that is $poly(k)$-competitive.*

In Sections 4.2 and 4.3, we provide proofs of Theorem 3.4 and Theorem 3.5 respectively.

4 PROOFS

In this section, we provide an overview of the proofs of the main results in Section 3. We defer the proofs of some technical lemmas needed in the analysis to the Appendix when appropriate.

We first present the proof of the $O(L^{2k})$ competitive ratio upper bound in Theorem 3.1 because it is our main result. Then, we prove the lower bound results in Theorem 3.4 and Theorem 3.5 because they establish the tightness of the dependencies on k and L in our main result Theorem 3.1.

4.1 Proof of Theorem 3.1

Intuitively, to bound the cost of iROBD with k steps of delay, we will derive relationships between its trajectory and the trajectory of ROBD (Algorithm 1), which experiences no delay and has been studied thoroughly in [15, 31]. However, while establishing such a relationship is relatively straightforward in [31], the situation becomes considerably more complicated in our setting since iROBD's trajectory can be "far away" from no-delay ROBD's trajectory after k "estimate and solve" loops (see Line 10-12 in Algorithm 2). Therefore, we adopt a novel induction-based proof, where we first reduce iROBD with k steps of delay to iROBD with less than k steps of delay, and then apply the induction hypothesis.

Following this idea, we treat the decision points of no-delay ROBD $y_t^{(0)}$ as a baseline throughout the proof. Since the cost functions are well-conditioned, it suffices to bound the difference $\|y_t^{(k)} - y_t^{(0)}\|$ in order to bound the cost incurred by k-step-delay iROBD. The impact of delays on iROBD can then be qualified by how fast the difference $\|y_t^{(k)} - y_t^{(0)}\|$ increases as the length of delay k grows.

Before the proof of Theorem 3.1, we first propose a lemma, demonstrating the cumulative nature of the error of iROBD.

Lemma 4.1. *The distance between $y_t^{(0)}$ and $y_t^{(k)}$ can be bounded by:*

$$\|y_t^{(k)} - y_t^{(0)}\|^2 \leq 8\|v_t^{(k)} - v_t^{(0)}\|^2 + 2pL^2 \sum_{i=1}^{p} \|y_{t-i}^{(k-i)} - y_{t-i}^{(0)}\|^2.$$

Lemma 4.1 bounds of the difference between the decisions of k-step-delay iROBD and no-delay ROBD by its counter parts with less steps of delays, as well as an additional error on estimating the true minimizer v_t. This additional error will be related to iROBD trajectories with fewer steps of delays later in the main proof.

Proof of Lemma 4.1. We know that, for any m-strongly convex function $g : \mathcal{X} \to \mathbb{R}$ and its minimizer v ($v = \arg\min_{x\in\mathcal{X}} g(x)$), the following inequality holds for all $x \in \mathcal{X}$:

$$g(x) \geq g(v) + \frac{m}{2}\|x - v\|^2.$$

Therefore, given $y_t^{(0)}$ =ROBD($f_t, y_{t-p:t-1}^{(0)}$) in Line 6 of Algorithm 2, that is, $y_t^{(0)} = \arg\min_y f_t(y) + \frac{\lambda}{2}\|y - \delta(y_{t-1:t-p})\|^2$, we have that

$$\begin{aligned} f_t(y_t^{(0)}) + \frac{\lambda}{2}\|y_t^{(0)} - \delta(y_{t-1:t-p}^{(0)})\|^2 + \frac{m+\lambda}{2}\|y_t^{(0)} - (y_t^{(k)} + v_t^{(0)} - v_t^{(k)})\|^2 \\ \leq f_t(y_t^{(k)} + v_t^{(0)} - v_t^{(k)}) + \frac{\lambda}{2}\|y_t^{(k)} + v_t^{(0)} - v_t^{(k)} - \delta(y_{t-1:t-p}^{(0)})\|^2. \end{aligned}$$

Similarly, since $y_t^{(k)} \leftarrow$ ROBD($f_t^{(k)}, y_{t-1}^{(k-1)}, \cdots, y_{t-k}^{(0)}, \cdots, y_{t-p}^{(0)}$) in Line 12 of Algorithm 2, we have that

$$\begin{aligned} f_t(y_t^{(k)} + v_t^{(0)} - v_t^{(k)}) + \frac{\lambda}{2}\|y_t^{(k)} - \delta(y_{t-1}^{(k-1)} \cdots y_{t-p}^{(k-p)})\|^2 + \frac{m+\lambda}{2}\|y_t^{(0)} - (y_t^{(k)} + v_t^{(0)} - v_t^{(k)})\|^2 \\ \leq f_t(y_t^{(0)}) + \frac{\lambda}{2}\|y_t^{(0)} - v_t^{(0)} + v_t^{(k)} - \delta(y_{t-1}^{(k-1)} \cdots y_{t-p}^{(k-p)})\|^2, \end{aligned}$$

where we used $f_t^{(k)}(y) = f_t(y + v_t^{(0)} - v_t^{(k)})$.

Summing the above two inequalities gives

$$\begin{aligned} &(m+\lambda)\|y_t^{(0)} - y_t^{(k)} + v_t^{(k)} - v_t^{(0)}\|^2 \\ &\quad\leq \frac{\lambda}{2}\|y_t^{(k)} + v_t^{(0)} - v_t^{(k)} - \delta(y_{t-1:t-p}^{(0)})\|^2 + \frac{\lambda}{2}\|y_t^{(0)} - v_t^{(0)} + v_t^{(k)} - \delta(y_{t-1}^{(k-1)} \cdots y_{t-p}^{(k-p)})\|^2 \\ &\qquad - \frac{\lambda}{2}\|y_t^{(0)} - \delta(y_{t-1:t-p}^{(0)})\|^2 - \frac{\lambda}{2}\|y_t^{(k)} - \delta(y_{t-1}^{(k-1)} \cdots y_{t-p}^{(k-p)})\|^2 \\ &\quad\leq \lambda\|y_t^{(0)} - y_t^{(k)} + v_t^{(k)} - v_t^{(0)}\|\|v_t^{(0)} - v_t^{(k)} + (\delta(y_{t-1}^{(k-1)} \cdots y_{t-p}^{(k-p)}) - \delta(y_{t-1:t-p}^{(0)}))\| \\ &\quad\leq \lambda\|y_t^{(0)} - y_t^{(k)} + v_t^{(k)} - v_t^{(0)}\|\left(\|v_t^{(0)} - v_t^{(k)}\| + \|\delta(y_{t-1}^{(k-1)} \cdots y_{t-p}^{(k-p)}) - \delta(y_{t-1:t-p}^{(0)})\|\right). \end{aligned}$$

Therefore, we can see that

$$\|y_t^{(0)} - y_t^{(k)} + v_t^{(k)} - v_t^{(0)}\| \leq \|v_t^{(0)} - v_t^{(k)}\| + \|\delta(y_{t-1}^{(k-1)} \cdots y_{t-p}^{(k-p)}) - \delta(y_{t-1:t-p}^{(0)})\|,$$

which implies that

$$\|y_t^{(0)} - y_t^{(k)}\| \leq 2\|v_t^{(0)} - v_t^{(k)}\| + \|\delta(y_{t-1}^{(k-1)} \cdots y_{t-p}^{(k-p)}) - \delta(y_{t-1:t-p}^{(0)})\|.$$

Take square and use the Cauchy Inequality, and then we have that

$$\|y_t^{(0)} - y_t^{(k)}\|^2 \leq 8\|v_t^{(0)} - v_t^{(k)}\|^2 + 2pL^2\sum_{i=1}^{p}\|y_{t-i}^{(k-i)} - y_{t-i}^{(0)}\|^2.$$

□

Next, we use the previous lemma to prove Theorem 3.1. Recall that in Lemma 4.1, the decision difference between k-step-delay iROBD and no-delay ROBD is bounded not only by its counter parts with less steps of delays, but also an estimation error $\|v_t^{(k)} - v_t^{(0)}\|^2$. We show that we can bound the impacts of this error on the performance using the strongly-convexity of f_t.

Proof of Theorem 3.1. Define a function $\psi : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ as

$$\psi(v) = \min_y h_t(y - v) + \lambda c(y, y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)}).$$

We know that for any function m-strongly convex $f : \mathbb{R}^d \to \mathbb{R}$, function $g : \mathbb{R}^d \to \mathbb{R}$ in the form:

$$g(x) = \min_y f(y) + \frac{\lambda}{2}\|y - x\|^2$$

is $\frac{m\lambda}{m+\lambda}$-strongly convex as a function of x (see Lemma 3 in [31]). Thus, we have

$$\begin{aligned} h_t(y_t^{(k)} - v_t^{(k)}) + \lambda c(y_t^{(k)}, y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)}) + \frac{1}{2} \cdot \frac{m\lambda}{m+\lambda}\|v_t^{(0)} - v_t^{(k)}\|^2 \\ = \psi(v_t^{(k)}) + \frac{1}{2} \cdot \frac{m\lambda}{m+\lambda}\|v_t^{(0)} - v_t^{(k)}\|^2 \leq \psi(v_t^{(0)}) \end{aligned} \tag{5a}$$

According to the definition of ψ, we can see that

$$\begin{aligned} \psi(v_t^{(0)}) &= \min_y h_t(y - v_t^{(0)}) + \lambda c(y, y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)}) \\ &\leq h_t(y_t^{(0)} - v_t^{(0)}) + \lambda c(y_t^{(0)}, y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)}) \\ &= h_t(y_t^{(0)} - v_t^{(0)}) + \frac{\lambda}{2}\|y_t^{(0)} - \delta(y_{t-1:t-p}^{(0)}) + (\delta(y_{t-1:t-p}^{(0)}) - \delta(y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)}))\|^2 \\ &\leq h_t(y_t^{(0)} - v_t^{(0)}) + \lambda\|y_t^{(0)} - \delta(y_{t-1:t-p}^{(0)})\|^2 + \lambda\|\delta(y_{t-1:t-p}^{(0)}) - \delta(y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)})\|^2, \end{aligned} \tag{6a}$$

where we have applied AM-GM inequality in Equation (6a).

Here, we have encountered terms as square of the distance between two δ functions, where $\delta(y_{t-1:t-p}^{(0)})$ corresponds to ROBD while $\delta(y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)})$ our algorithm of iROBD. All we know about the δ function is that it is Lipschitz, so by the Lipschitz condition of it, we have

$$\begin{aligned} h_t(y_t^{(0)} - v_t^{(0)}) + \lambda\|y_t^{(0)} - \delta(y_{t-1:t-p}^{(0)})\|^2 + \lambda\|\delta(y_{t-1:t-p}^{(0)}) - \delta(y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)})\|^2 \\ \leq h_t(y_t^{(0)} - v_t^{(0)}) + 2\lambda c(y_{t:t-p}^{(0)}) + \lambda(\sum_{i=1}^{p} L\|y_{t-i}^{(k-i)} - y_{t-i}^{(0)}\|)^2. \end{aligned} \tag{7}$$

And according to the Cauchy Inequality, the line above is no larger than

$$h_t(y_t^{(0)} - v_t^{(0)}) + 2\lambda c(y_{t:t-p}^{(0)}) + \lambda p L^2 \sum_{i=1}^{p} \|y_{t-i}^{(k-i)} - y_{t-i}^{(0)}\|^2. \tag{8}$$

With Equations (5a), (6a), (7) and (8), we immediately get

$$\begin{aligned} h_t(y_t^{(k)} - v_t^{(k)}) + \lambda c(y_t^{(k)}, y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)}) + \frac{1}{2} \cdot \frac{m\lambda}{m+\lambda}\|v_t^{(0)} - v_t^{(k)}\|^2 \\ \leq h_t(y_t^{(0)} - v_t^{(0)}) + 2\lambda c(y_{t:t-p}^{(0)}) + \lambda p L^2 \sum_{i=1}^{p} \|y_{t-i}^{(k-i)} - y_{t-i}^{(0)}\|^2. \end{aligned} \tag{9}$$

This inequality is important in proving because it bridges the hitting cost of no-delay ROBD, that is, $h_t(y_t^{(0)} - v_t^{(0)})$. Next, we turn to connect the hitting cost of k-step-delay iROBD, that is $h_t(y_t^{(k)} - v_t^{(0)})$.

We know that for convex and l-strongly smooth function $f : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, the inequality

$$f(y) \leq (1+\eta)f(x) + \left(1 + \frac{1}{\eta}\right) \cdot \frac{l}{2}\|x - y\|^2$$

holds for all $\eta > 0$. Observing that h is l-strongly smooth, we can conclude that, for any $\eta_{1,k} > 0$,

$$\frac{1}{1+\eta_{1,k}} h_t(y_t^{(k)} - v_t^{(0)}) \leq h_t(y_t^{(k)} - v_t^{(k)}) + \frac{l}{2\eta_{1,k}}\|v_t^{(0)} - v_t^{(k)}\|^2. \tag{10}$$

Additionally, since the function $\frac{\lambda}{2}\|y_t^{(k)} - y\|^2$ is λ-strongly smooth in y, for any $\eta_{2,k} > 0$, we have

$$\begin{aligned}
&\frac{1}{1+\eta_{2,k}} \cdot \frac{\lambda}{2}\|y_t^{(k)} - \delta(y_{t-1:t-p}^{(k)})\|^2 \\
&\quad \leq \frac{\lambda}{2}\|y_t^{(k)} - \delta(y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)})\|^2 + \frac{\lambda}{2\eta_{2,k}}\|\delta(y_{t-1:t-p}^{(k)}) - \delta(y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)})\|^2.
\end{aligned} \tag{11}$$

Substituting Equation (11) and Equation (10) into Equation (9), we have

$$\begin{aligned}
&\frac{1}{1+\eta_{1,k}} h_t(y_t^{(k)} - v_t^{(0)}) + \frac{1}{1+\eta_{2,k}} \cdot \frac{\lambda}{2}\|y_t^{(k)} - \delta(y_{t-1:t-p}^{(k)})\|^2 \\
&\quad \leq h_t(y_t^{(k)} - v_t^{(k)}) + \frac{l}{2\eta_{1,k}}\|v_t^{(0)} - v_t^{(k)}\|^2 \\
&\qquad + \frac{\lambda}{2}\|y_t^{(k)} - \delta(y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)})\|^2 + \frac{\lambda}{2\eta_{2,k}}\|\delta(y_{t-1:t-p}^{(k)}) - \delta(y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)})\|^2 \\
&\quad \leq h_t(y_t^{(0)} - v_t^{(0)}) + 2\lambda c(y_{t:t-p}^{(0)}) + \lambda p L^2 \sum_{i=1}^{p} \|y_{t-i}^{(k-i)} - y_{t-i}^{(0)}\|^2 - \frac{1}{2} \cdot \frac{m\lambda}{m+\lambda}\|v_t^{(0)} - v_t^{(k)}\|^2 \\
&\qquad + \frac{l}{2\eta_{1,k}}\|v_t^{(0)} - v_t^{(k)}\|^2 + \frac{\lambda}{2\eta_{2,k}}\|\delta(y_{t-1:t-p}^{(k)}) - \delta(y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)})\|^2 \\
&\quad \leq h_t(y_t^{(0)} - v_t^{(0)}) + 2\lambda c(y_{t:t-p}^{(0)}) + \frac{l}{2\eta_{1,k}}\|v_t^{(0)} - v_t^{(k)}\|^2 \\
&\qquad - \frac{m\lambda}{2m+2\lambda}\|v_t^{(0)} - v_t^{(k)}\|^2 + \lambda p L^2 \sum_{i=1}^{p} \|y_{t-i}^{(k-i)} - y_{t-i}^{(0)}\|^2 \\
&\qquad + \frac{\lambda}{\eta_{2,k}}\|\delta(y_{t-1:t-p}^{(k)}) - \delta(y_{t-1:t-p}^{(0)})\|^2 + \frac{\lambda}{\eta_{2,k}}\|\delta(y_{t-1}^{(k-1)}, \cdots, y_{t-p}^{(k-p)}) - \delta(y_{t-1:t-p}^{(0)})\|^2 \qquad \text{(12a)} \\
&\quad \leq h_t(y_t^{(0)} - v_t^{(0)}) + 2\lambda c(y_{t:t-p}^{(0)}) + \frac{l}{2\eta_{1,k}}\|v_t^{(0)} - v_t^{(k)}\|^2 - \frac{m\lambda}{2m+2\lambda}\|v_t^{(0)} - v_t^{(k)}\|^2 \\
&\qquad + \lambda p L^2 (1 + \frac{1}{\eta_{2,k}}) \sum_{i=1}^{p} \|y_{t-i}^{(k-i)} - y_{t-i}^{(0)}\|^2 + \frac{\lambda p L^2}{\eta_{2,k}} \sum_{i=1}^{p} \|y_{t-i}^{(k)} - y_{t-i}^{(0)}\|^2, \qquad \text{(12b)}
\end{aligned}$$

where we have applied AM-GM inequality in Equation (12a), the Lipschitz condition of δ and Cauchy inequality in Equation (12b). Now we already have the relation between the step-wise cost of k-step-delay iROBD in the left hand side and the step-wise cost of no-delay ROBD in the right hand side. The problem left is to analyze the impacts of terms of errors on estimating the minimizer and decision difference of iROBD with fewer steps of delays to the no-delay ROBD.

Summing over time and applying Lemma 4.1 gives

$$\begin{aligned}
&\sum_{t=1}^{T}\left(\frac{1}{1+\eta_{1,k}}H_t^{(k)}+\frac{\lambda}{1+\eta_{2,k}}M_t^{(k)}\right)\\
&\le\sum_{t=1}^{T}\left(H_t^{(0)}+2\lambda M_t^{(0)}\right)+\left(\frac{l}{\eta_{1,k}}-\frac{m\lambda}{m+\lambda}\right)\frac{1}{2}\sum_{t=1}^{T}\|v_t^{(0)}-v_t^{(k)}\|^2\\
&\quad+\frac{\lambda p^2L^2}{\eta_{2,k}}\sum_{t=1}^{T}\|y_t^{(k)}-y_t^{(0)}\|^2+\lambda pL^2(1+\frac{1}{\eta_{2,k}})\sum_{i=1}^{k-1}\sum_{t=1}^{T}\|y_t^{(i)}-y_t^{(0)}\|^2\\
&\le 2\sum_{t=1}^{T}\left(H_t^{(0)}+\lambda M_t^{(0)}\right)+\left(\frac{l}{\eta_{1,k}}+\frac{16\lambda p^2L^2}{\eta_{2,k}}-\frac{m\lambda}{m+\lambda}\right)\frac{1}{2}\sum_{t=1}^{T}\|v_t^{(0)}-v_t^{(k)}\|^2\\
&\quad+\sum_{j=k-1}^{1}16\lambda pL^2(1+\frac{1+2p^2L^2}{\eta_{2,k}})(1+2pL^2)^{k-1-j}\frac{1}{2}\sum_{t=1}^{T}\|v_t^{(0)}-v_t^{(j)}\|^2. \qquad (13)
\end{aligned}$$

This structure is rather complicated since it not only involves $\sum_{t=1}^{T}\|v_t^{(0)}-v_t^{(k)}\|^2$, but also $\|v_t^{(0)}-v_t^{(j)}\|^2$ for $j=1,2,\cdots,k-1$. It just corresponds to Lemma 4.1, where the distance between the choices of iROBD and ROBD consists of errors from past steps.

Finally, pick $\eta_{2,k}=\eta_k$ and $\eta_{1,k}=\frac{1+\eta_k-\lambda}{\lambda}$ so that $\frac{1}{1+\eta_{1,k}}=\frac{\lambda}{1+\eta_{2,k}}$. Additionally, denote $P(i)$ as

$$P(i)=\frac{\lambda}{1+\eta_i}\sum_{t=1}^{T}\left(H_t^{(i)}+M_t^{(i)}\right).$$

This yields the following:

$$\begin{aligned}
\frac{1}{\prod_{i=1}^{k-1}\eta_i}P(k)&\le\frac{1}{\prod_{i=1}^{k-1}\eta_i}P(k)+\frac{1}{\prod_{i=1}^{k-2}\eta_i}P(k-1)+\cdots+\frac{1}{\eta_1}P(2)+P(1)\\
&\le(1+\frac{2}{\eta_1}+\cdots+\frac{2}{\prod_{i=1}^{k-1}\eta_i})\sum_{t=1}^{T}\left(H_t^{(0)}+\lambda M_t^{(0)}\right)+\sum_{i=1}^{k-1}\frac{\lambda\sum_{t=1}^{T}\|v_t^{(0)}-v_t^{(i)}\|^2}{2\prod_{j=1}^{i-1}\eta_j}\cdot S(i)\\
&\quad+\frac{\lambda}{\prod_{i=1}^{k-1}\eta_i}\left(\frac{l}{1+\eta_k-\lambda}+\frac{16p^2L^2}{\eta_k}-\frac{m}{m+\lambda}\right)\frac{1}{2}\sum_{t=1}^{T}\|v_t^{(0)}-v_t^{(k)}\|^2,
\end{aligned}$$

where the coefficient $S(i)$ is defined as

$$S(i)=\frac{l}{1+\eta_i-\lambda}+\frac{16p^2L^2}{\eta_i}-\frac{m}{m+\lambda}+16\sum_{j=i+1}^{k}(1+\frac{1+2p^2L^2}{\eta_j})\cdot\frac{pL^2}{\eta_i}\cdot\frac{(1+2p^2L^2)^{j-i-1}}{\prod_{h=i+1}^{j-1}\eta_h}.$$

When $\eta_i=O(l+2p^2L^2)$ for $i=1,\cdots,k-1$, we can make the coefficient of $\sum_{t=1}^{T}\|v_t^{(0)}-v_t^{(i)}\|^2$ in the $S(i)$ negative for all i. Also, when $\eta_k=O(l+2p^2L^2)$, the coefficient of $\sum_{t=1}^{T}\|v_t^{(0)}-v_t^{(k)}\|^2$ in the inequality above negative. It finally leads to

$$\begin{aligned}
\sum_{i=1}^{T}\left(H_t^{(k)}+M_t^{(k)}\right)&=\frac{1+\eta_k}{\lambda}P(k)\\
&\le\frac{1+\eta_k}{\lambda}(\prod_{i=1}^{k-1}\eta_i+2\sum_{i=2}^{k-1}\prod_{j=i}^{k-1}\eta_j+2)\sum_{t=1}^{T}\left(H_t^{(0)}+\lambda M_t^{(0)}\right)
\end{aligned}$$

$$\leq O((l+2p^2L^2)^k)\max\left\{\frac{1}{\lambda}, \frac{m+\lambda}{m+(1-p^2L^2)\lambda}\right\}\sum_{i=1}^{T}\left(H_t^* + M_t^*\right).$$

The last inequality is based on the following lemma comparing the performance of ROBD to the optimal solution:

Lemma 4.2. $\sum_{t=1}^{T}(H_t^{(0)} + \lambda M_t^{(0)}) \leq \sum_{t=1}^{T}\left(H_t^* + \frac{\lambda(m+\lambda)}{m+(1-p^2L^2)\lambda}M_t^*\right)$.

Due to space constraints, we defer the proof of Lemma 4.2 to Appendix B □

4.2 Proof of Theorem 3.4

Without loss of generality, consider a 1-dimensional problem instance where the agent starts at $x_0 = 0$, and the hitting cost sequence is given by $f_t(x) = \frac{m}{2}(x - v_t)^2$ with $v_t = \alpha^{t-1}$. Note that this instance can be extended to d-dimensional space easily when $d > 1$ by letting $v_t = (\alpha^{t-1}, 0, \ldots, 0)$. Suppose the horizon T equals to the delay length k, which means the online agent cannot observe any information about the hitting costs before the game ends. We discuss the behavior of any competitive online algorithm ALG and an offline adversary ADV that moves to v_t at time step t.

Since ALG is competitive, it will stay at the origin throughout the game. This is because ALG cannot observe any information about the hitting cost functions until the end. If ALG moves at any time step, and the hitting cost sequence turns out to be $f_t(x) = \frac{m}{2}x^2, t = 1, \ldots, k$, the ratio between $cost(ALG)$ and $cost(ADV)$ will be ∞. Therefore, we see that the algorithm cost is

$$cost(ALG) = \sum_{t=1}^{k}\frac{m}{2}\cdot\alpha^{2t-2} = \frac{m(\alpha^{2k}-1)}{2(\alpha^2-1)}. \tag{14}$$

On the other hand, the total cost incurred by the offline adversary ADV is $cost(ADV) = \frac{1}{2}$, because the only cost it incurs is the switching cost at time step 1. Combining with (14), we see that

$$\frac{cost(ALG)}{cost(OPT)} \geq \frac{cost(ALG)}{cost(ADV)} = \frac{m(\alpha^{2k}-1)}{\alpha^2-1}.$$

Since this inequality holds for any competitive online algorithm ALG, we see the competitive ratio of any online algorithm is lower bounded by $\frac{m(\alpha^{2k}-1)}{\alpha^2-1}$. □

4.3 Proof of Theorem 3.5

We can assume $f_t(v_t) = 0$ without loss of generality. We consider a delayed version of a greedy, move to the minimizer (M2M) algorithm. M2M works by picking the decision point

$$x_t = \begin{cases} x_0 & \text{if } t \leq k \\ v_{t-k} & \text{if } t > k \end{cases}.$$

To simplify the notation, we define $v_0 := x_0$. The total cost incurred by M2M can be expressed as

$$cost(M2M) = \sum_{t=1}^{k} f_t(x_0) + \sum_{t=k+1}^{T} f_t(v_{t-k}) + \sum_{t=1}^{T-k}\frac{1}{2}\|v_t - v_{t-1}\|^2. \tag{15}$$

For $t \leq k$, we have

$$\begin{aligned} f_t(x_0) &\leq \frac{l}{2}\|v_t - x_0\|^2 \\ &\leq \frac{l}{2}\left(\|v_t - x_t^*\| + \sum_{\tau=1}^{t}\|x_\tau^* - x_{\tau-1}^*\|\right)^2 \end{aligned}$$

$$\leq \frac{l(t+1)}{2}\left(\|v_t - x_t^*\|^2 + \sum_{\tau=1}^{t}\|x_\tau^* - x_{\tau-1}^*\|^2\right)$$
$$\leq \frac{l(t+1)}{m}H_t^* + l(t+1)\sum_{\tau=1}^{t}M_\tau^*. \tag{16}$$

For $t > k$, we see that

$$\begin{aligned} f_t(v_{t-k}) &\leq \frac{l}{2}\|v_t - v_{t-k}\|^2 \\ &\leq \frac{l}{2}\left(\|v_t - x_t^*\| + \sum_{\tau=t-k+1}^{t}\|x_\tau^* - x_{\tau-1}^*\| + \|x_{t-k}^* - v_{t-k}\|\right)^2 \\ &\leq \frac{l(k+2)}{2}\left(\|v_t - x_t^*\|^2 + \sum_{\tau=t-k+1}^{t}\|x_\tau^* - x_{\tau-1}^*\|^2 + \|x_{t-k}^* - v_{t-k}\|^2\right) \\ &\leq \frac{l(k+2)}{m}(H_t^* + H_{t-k}^*) + l(k+2)\sum_{\tau=t-k+1}^{t}M_\tau^*. \end{aligned} \tag{17}$$

For $t \leq T - k$, we see that

$$\begin{aligned} \frac{1}{2}\|v_t - v_{t-1}\|^2 &\leq \frac{1}{2}\left(\|v_t - x_t^*\| + \|x_t^* - x_{t-1}^*\| + \|v_{t-1} - x_{t-1}^*\|\right)^2 \\ &\leq \frac{3}{2}\left(\|v_t - x_t^*\|^2 + \|x_t^* - x_{t-1}^*\|^2 + \|v_{t-1} - x_{t-1}^*\|^2\right) \\ &\leq \frac{3}{m}(H_t^* + H_{t-1}^*) + 3M_t^*. \end{aligned} \tag{18}$$

Substituting (16), (17), and (18) into (15) gives that

$$\frac{cost(M2M)}{cost(OPT)} = O(k^3).$$

□

5 CONNECTION TO ONLINE CONTROL

Deep connections between online optimization and online control have emerged in recent years. However, the reductions developed in the literature to this point, e.g., [15, 31], have applied only to limited linear control settings. In particular, the most general result so far shows that Input-Disturbed Squared Regulators (IDSR) (Equation (19) with a special form of w_t: $w_t = B\bar{w}_t$) can be reduced to online convex optimization with structured memory. Here, we highlight that the addition of feedback delay and nonlinear switching cost to online optimization with memory significantly expands the class of control problems which can be addressed. We present two different reductions, which focus on linear dynamics with general adversarial disturbance and nonlinear dynamics with delay, respectively. Both generalize prior arts significantly. In particular, we provide the first reduction as well as the first competitive control policy with either delay or nonlinear dynamics. Further generalizations of control problem classes is an interesting and important future direction.

5.1 Linear dynamics with adversarial disturbances

Our first reduction connects online optimization with delay and memory to a class of linear dynamical system that is more general than possible via reductions in prior work, e.g., in [16, 31].

Algorithm 3 Reduction to OCO with Memory and Delay

Input: Transition matrix A and control matrix B
Solver: OCO with memory p and delay p algorithm ALG
for $t = 0$ to $T - 1$ **do**
  Observe: x_t and $q_{t:t+p-1}$
  if $t > 0$ **then**
    $w_{t-1} \leftarrow x_t - Ax_{t-1} - Bu_{t-1}$
    $\zeta_{t-1} \leftarrow \psi(w_{t-1}) + \sum_{i=1}^{p} C_i \zeta_{t-1-i}$
  end if
  $f_t(y) := \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{p_i} q_{t+j} \left(y^{(i)} + \zeta_t^{(i)} + r(t+j, i, j) \right)^2$
  $h_t(y) := \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{p_i} q_{t+j} \left(y^{(i)} \right)^2$
  Work out $v_{t-p} \leftarrow \arg\min_v f_{t-p}(y)$
  Feed v_{t-p} and h_t into ALG
  Obtain the output of ALG, y_t
  $u_t \leftarrow y_t - \sum_{i=1}^{p} C_i y_{t-i}$
end for

Specifically, we consider

$$\begin{aligned} \min_{u_t} \quad & \sum_{t=1}^{T} \frac{q_t}{2} \|x_t\|^2 + \sum_{t=0}^{T-1} \frac{1}{2} \|u_t\|^2 \\ \text{s.t.} \quad & x_{t+1} = Ax_t + Bu_t + w_t, \end{aligned} \tag{19}$$

where (A, B) are in controllable canonical form and w_t is a potentially adversarial disturbance. Note that in [16] B has to be invertible and [31] only allows input disturbed systems (i.e., $x_{t+1} = Ax_t + B(u_t + w_t)$).

Algorithm 3 presents a reduction from the control problem mentioned above to online convex optimization with structured memory and feedback delay leading to the following theorem.

Theorem 5.1. *Consider the online control problem in Equation (19). Assume the coefficients $q_{t:t+p-1}$ are observable at step t. It can be converted to an instance of OCO with structured memory and feedback delay using Algorithm 3.*

A proof of Theorem 5.1 is given in Appendix D. This proof splits the disturbance into an input disturbance part, which can be dealt with using approaches in prior work, and a residual part, which leads to the k-round delay and requires a new analysis. From the details of the reduction in Algorithm 3 we can see that the cost function f_t in the resulting online optimization has a term $r(t + p_i, i, p_i)$ in it, which involves w_{t+p_i-1}. Since $p = \max\{p_i\}$, we know that f_t has a w_{t+p-1}, which is not revealed until step $t + p$ in our setting, resulting in a p-step feedback delay.

Theorem 3.3 immediately implies that iROBD provides a constant-competitive online policy for the control problem, even against adversarial disturbances. We also show the state disturbed component of w_t exactly corresponds to multi-round feedback delay in online optimization. Further, since $cost(ALG)$ and $cost(OPT)$ remain unchanged, the reduction immediately provides competitive policies for the linear system with general adversarial disturbances, based on our constant-competitive

algorithm iROBD. To state the result, we define:

$$q_{min} = \min_{0\le t\le T-1, 1\le i\le d} \sum_{j=1}^{p_i} q_{t+j},$$

$$q_{max} = \max_{0\le t\le T-1, 1\le i\le d} \sum_{j=1}^{p_i} q_{t+j}.$$

Corollary 5.2. *Consider the online control problem in Equation (19). Assume the coefficients $q_{t:t+p-1}$ are observable at step t. Let $\alpha = \sum_{i=1}^{p} \|C_i\|$. The competitive ratio of Algorithm 3, using iROBD(λ) as the solver, is*

$$O\left((q_{max} + 2\alpha^2)^p \max\left\{\frac{1}{\lambda}, \frac{q_{min}+\lambda}{q_{min} + (1-\alpha^2)\lambda}\right\}\right)$$

Note that, in this corollary, due to the structure in (A, B), the lengths of delay and memory are both p, which is also the same as the controllability index of (A, B).

5.2 Nonlinear dynamics with delay and time-varying costs

Our second reduction connects online optimization with delay and nonlinear switching cost to the following class of online nonlinear control problems:

$$\begin{aligned} \min_{u_t} \ & \sum_{t=1}^{T} f_t(x_t) + \sum_{t=0}^{T-1} \frac{1}{2}\|u_t\|^2 \\ \text{s.t.} \quad & x_{t+1} = Ax_t + u_t + g(x_t) \end{aligned} \tag{20}$$

where $\{f_t\}_{t=1}^{T}$ is time-variant well-conditioned cost (e.g., trajectory tracking cost), and $g(x_t)$ is the nonlinear dynamics term. At time step t, only $f_{1:t-k}$ is known due to communication delays. Many robotic systems can be viewed as special cases of this form, such as pendulum dynamics and quadrotor dynamics [32]. It is immediate to see that, by defining $y_t = x_t$, this online control problem can be converted into an online optimization problem with hitting cost f_t and nonlinear switching cost $c(y_t, y_{t-1}) = \frac{1}{2}\|y_t - Ay_{t-1} - g(y_{t-1})\|^2$.

In this section we present a reduction from this class of online control to online convex optimization with nonlinear switching cost and feedback delay. The reduction implies that Theorem 3.1 immediately gives that iROBD provides a constant-competitive online policy for the control problem, even against adversarial disturbances.

Remark 3. *For simplicity of presentation we consider the trajectory tracking task $f_t(x_t) = \frac{1}{2}(x_t - v_t)^\top Q_t(x_t - v_t)$, where $\{v_t\}$ is the desired trajectory to track. However, the cost itself is not necessarily quadratic. In fact, our algorithm works for general hitting costs f_t, if we know the minimizer and the geometry of the function. In other words, we need the parameters of the function to know its "shape" and the minimizer to locate the function in the space. In this general setting, we just need to modify Line 4 to Line 6 in Algorithm 4 to get the general form:*

- *Line 4: Observe x_t, v_t and the geometry of f_t.*
- *Line 5: Set the exact f_{t-k} by its geometry and minimizer v_{t-k}.*
- *Line 6: Set function h_t by the same geometry as f_t and minimizer at 0.*

With this modification, the following results still hold.

Theorem 5.3. *Consider the online control problem in Equation (20). If Q_t is observable at step t, and only the trajectory $v_{1:t-k}$ is known, i.e., there are k steps of feedback delay, then it can be converted to an instance of online optimization with switching cost and feedback delay using Algorithm 4.*

Algorithm 4 Reduction to Online Optimization with Nonlinear Switching Cost and Delay

Input: Nonlinear function $g(x)$
Solver: Online optimization with delay $k+1$ and switching cost algorithm ALG
for $t = 0$ to $T-1$ **do**
  Observe: x_t, v_{t-k} and Q_t
  Set $f_{t-k-1}(y) = \frac{1}{2}(y - v_{t-k})^T Q_{t-k}(y - v_{t-k})$
  Set $h_t(y) = \frac{1}{2} y^T Q_t y$
  $c(y, y_{t-1}) := \frac{1}{2}\|y - Ax_t - g(x_t)\|^2$
  Feed f_{t-k-1}, h_t and $c(y, y_{t-1})$ into ALG
  Obtain the output of ALG, y_t
  $u_t \leftarrow y_t - Ay_{t-1} - g(y_{t-1})$
end for

A proof of Theorem 5.3 is given in Appendix E. The reduction in Algorithm 4 results from observing that, after defining $y_t = x_t$, the online control problem in Equation (20) can be converted into an online optimization problem with hitting cost $f_t(y_t) = \frac{1}{2}(y_t - v_t)^T Q_t (y_t - v_t)$ and switching cost $c(y_t, y_{t-1}) = \frac{1}{2}\|y_t - Ay_{t-1} - g(y_{t-1})\|^2$. Note that the nonlinear switching cost comes from the nonlinear dynamics, and the delayed feedback is coming from delayed information about the target trajectory $v_{1:t}$, i.e., only $v_{1:t-k}$ is known at time step t due to communication delays.

Given that we have proven that iROBD is a constant competitive algorithm for online optimization with feedback delay and nonlinear switching costs, the reduction above immediately brings a competitive policy for class of online control problem with nonlinear dynamics and delay in Equation (20). This is because $cost(ALG)$ and $cost(OPT)$ remain unchanged in the reduction. To state this formally, suppose the smallest and largest eigenvalue of positive definite matrix Q_t is $\lambda_{min}(t)$ and $\lambda_{max}(t)$ respectively for $t = 1, \cdots, T$. Further, define $\lambda_{min} = \min_t\{\lambda_{min}(t)\}$, $\lambda_{max} = \max_t\{\lambda_{max}(t)\}$. Using this notation, we have the following corollary:

Corollary 5.4. *Consider the online control problem in Equation (20) where the Q_t is observable at step t. If $\|Ax + g(x) - Ax' - g(x')\| \le L\|x - x'\|$ for any $x, x' \in \mathbb{R}^n$, then the competitive ratio of Algorithm 4 using iROBD(λ) as the solver is upper bounded by:*

$$O\left((\lambda_{max} + 2L^2)^k \max\left\{\frac{1}{\lambda}, \frac{\lambda_{min} + \lambda}{\lambda_{min} + (1 - L^2)\lambda}\right\}\right).$$

This corollary implies that competitive control is more challenging when the system has more delay on the target trajectory (bigger k), when the cost functions are less smooth (larger λ_{max}), or if there are bad Lipschitz properties in the dynamics. These qualitative observations are consistent with those from the robust control and nonlinear control literature [34, 37].

6 CONCLUDING REMARKS

In this paper we propose a new policy, iROBD, for online optimization with feedback delay and nonlinear switching cost. We show that iROBD obtains constant competitive bound in this setting and provide reductions to online control that provide competitive bounds in that context as well. Our results are the first to characterize a competitive algorithm in settings with either multi-step delay or nonlinear switching costs, both of which are challenging and practically important factors.

Our results are general, but focus purely on worst cast bounds in the case when the algorithm does not have any information about future costs. In practice, using predictions is valuable and worst case algorithms can, at times, be overly pessimistic. Considering average case results in settings with (noisy) predictions is an important future direction. Additionally, it will be interesting

to deploy iROBD in applications with delay and nonlinear switching costs in order to evaluate the performance of the algorithm in practice.

REFERENCES

[1] Naman Agarwal, Brian Bullins, Elad Hazan, Sham M Kakade, and Karan Singh. 2019. Online control with adversarial disturbances. In *International Conference on Machine Learning (ICML)*.

[2] Naman Agarwal, Elad Hazan, and Karan Singh. 2019. Logarithmic Regret for Online Control. *Advances in Neural Information Processing Systems* 32 (2019), 10175–10184.

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

[4] Antonios Antoniadis and Kevin Schewior. 2018. A Tight Lower Bound for Online Convex Optimization with Switching Costs. In *Approximation and Online Algorithms*, Roberto Solis-Oba and Rudolf Fleischer (Eds.). Springer International Publishing, Cham, 164–175.

[5] CJ Argue, Anupam Gupta, Guru Guruganesh, and Ziye Tang. 2020. Chasing convex bodies with linear competitive ratio. In *SIAM Symposium on Discrete Algorithms (SODA)*.

[6] Masoud Badiei, Na Li, and Adam Wierman. 2015. Online convex optimization with ramp constraints. In *IEEE Conference on Decision and Control (CDC)*.

[7] Nikhil Bansal, Anupam Gupta, Ravishankar Krishnaswamy, Kirk Pruhs, Kevin Schewior, and Cliff Stein. 2015. A 2-Competitive Algorithm For Online Convex Optimization With Switching Costs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 40)*, Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim (Eds.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 96–109.

[8] Nataly Brukhim, Xinyi Chen, Elad Hazan, and Shay Moran. 2020. Online agnostic boosting via regret minimization. *arXiv preprint arXiv:2003.01150* 33 (2020), 644–654.

[9] Sébastien Bubeck, Bo'az Klartag, Yin Tat Lee, Yuanzhi Li, and Mark Sellke. 2020. Chasing nested convex bodies nearly optimally. In *SIAM Symposium on Discrete Algorithms (SODA)*.

[10] Sébastien Bubeck, Yin Tat Lee, Yuanzhi Li, and Mark Sellke. 2019. Competitively chasing convex bodies. In *ACM Symposium on Theory of Computing (STOC)*.

[11] Niangjun Chen, Anish Agarwal, Adam Wierman, Siddharth Barman, and Lachlan L.H. Andrew. 2015. Online Convex Optimization Using Predictions. *SIGMETRICS Perform. Eval. Rev.* 43, 1 (June 2015), 191–204.

[12] Niangjun Chen, Joshua Comden, Zhenhua Liu, Anshul Gandhi, and Adam Wierman. 2016. Using predictions in online optimization: Looking forward with an eye on the past. *ACM SIGMETRICS Performance Evaluation Review* 44, 1 (2016), 193–206.

[13] Niangjun Chen, Gautam Goel, and Adam Wierman. 2018. Smoothed Online Convex Optimization in High Dimensions via Online Balanced Descent. In *Proceedings of Conference On Learning Theory (COLT)*. 1574–1594.

[14] Shiyao Chen and Lang Tong. 2012. iEMS for large scale charging of electric vehicles: Architecture and optimal online scheduling. In *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 629–634.

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

[16] Gautam Goel and Adam Wierman. 2019. An online algorithm for smoothed regression and lqr control. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2504–2513.

[17] Paula Gradu, Elad Hazan, and Edgar Minasyan. 2020. Adaptive regret for control of time-varying dynamics. *arXiv preprint arXiv:2007.04393* (2020).

[18] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. 2013. Online learning under delayed feedback. In *International Conference on Machine Learning*. PMLR, 1453–1461.

[19] Seung-Jun Kim and Geogios B Giannakis. 2016. An online convex optimization approach to real-time energy pricing for demand response. *IEEE Transactions on Smart Grid* 8, 6 (2016), 2784–2793.

[20] Yingying Li, Xin Chen, and Na Li. 2019. Online Optimal Control with Linear Dynamics and Predictions: Algorithms and Regret Analysis. In *Advances in Neural Information Processing Systems*. 14858–14870.

[21] Yingying Li, Guannan Qu, and Na Li. 2018. Using predictions in online optimization with switching costs: A fast algorithm and a fundamental limit. In *2018 Annual American Control Conference (ACC)*. IEEE, 3008–3013.

[22] Yingying Li, Guannan Qu, and Na Li. 2020. Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit. *IEEE Trans. Automat. Control* (2020).

[23] Minghong Lin, Zhenhua Liu, Adam Wierman, and Lachlan LH Andrew. 2012. Online algorithms for geographical load balancing. In *Proceedings of the International Green Computing Conference (IGCC)*. 1–10.

[24] Yiheng Lin, Yang Hu, Haoyuan Sun, Guanya Shi, Guannan Qu, and Adam Wierman. 2021. Perturbation-based Regret Analysis of Predictive Control in Linear Time Varying Systems. *arXiv preprint arXiv:2106.10497* (2021).
[25] Zhenhua Liu, Iris Liu, Steven Low, and Adam Wierman. 2014. Pricing data center demand response. *ACM SIGMETRICS Performance Evaluation Review* 42, 1 (2014), 111–123.
[26] David Luenberger. 1967. Canonical forms for linear multivariable systems. *IEEE Trans. Automat. Control* 12, 3 (1967), 290–293.
[27] Mark Sellke. 2020. Chasing convex bodies optimally. In *SIAM Symposium on Discrete Algorithms (SODA)*.
[28] Ohad Shamir and Liran Szlak. 2017. Online learning with local permutations and delayed feedback. In *International Conference on Machine Learning*. PMLR, 3086–3094.
[29] Guanya Shi, Kamyar Azizzadenesheli, Soon-Jo Chung, and Yisong Yue. 2021. Meta-Adaptive Nonlinear Control: Theory and Algorithms. *Thirty-fifth Conference on Neural Information Processing Systems* (2021).
[30] Guanya Shi, Wolfgang Hönig, Xichen Shi, Yisong Yue, and Soon-Jo Chung. 2021. Neural-swarm2: Planning and control of heterogeneous multirotor swarms using learned interactions. *IEEE Transactions on Robotics* (2021).
[31] Guanya Shi, Yiheng Lin, Soon-Jo Chung, Yisong Yue, and Adam Wierman. 2020. Online Optimization with Memory and Competitive Control. In *Thirty-fourth Conference on Neural Information Processing Systems*.
[32] Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. 2019. Neural lander: Stable drone landing control using learned dynamics. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 9784–9790.
[33] Max Simchowitz, Karan Singh, and Elad Hazan. 2020. Improper learning for non-stochastic control. In *Conference on Learning Theory*. PMLR, 3320–3436.
[34] Jean-Jacques E Slotine, Weiping Li, et al. 1991. *Applied nonlinear control*. Vol. 199. Prentice hall Englewood Cliffs, NJ.
[35] Chenkai Yu, Guanya Shi, Soon-Jo Chung, Yisong Yue, and Adam Wierman. 2020. Competitive Control with Delayed Imperfect Information. *arXiv preprint arXiv:2010.11637* (2020).
[36] Chenkai Yu, Guanya Shi, Soon-Jo Chung, Yisong Yue, and Adam Wierman. 2020. The Power of Predictions in Online Control. *Advances in Neural Information Processing Systems* 33 (2020).
[37] Kemin Zhou, John Comstock Doyle, Keith Glover, et al. 1996. *Robust and optimal control*. Vol. 40. Prentice hall New Jersey.

A ITERATIVE ROBD FOR TIME-VARYING DELAY

When the feedback delay is time-varying, we assume that at time t the last revealed minimizer has the subscript of l_t. In another word, at time t, minimizers $v_{l_{t-1}+1}, v_{l_{t-1}+2}, \cdots, v_{l_t}$ are revealed. (If $l_{t-1} = l_t$, then no minimizers are revealed at time t, and Algorithm 5 will skip Line 5 to Line 8.) Thus, we have the modified Algorithm 5 for time-varying delay.

The upper bound on competitive ratio can be easily borrowed from the original problem and algorithm. Denote the maximum length of feedback delay as k. If for v_t the delay is k' and $k' < k$, then we define $v_t^{(k'+1)}, \cdots, v_t^{(k)}$ to be $v_t^{(k')}$, and $y_t^{(k'+1)}, \cdots, y_t^{(k)}$ to be $y_t^{(k')}$. In this way, all equations in the proof of Theorem 3.1 are preserved. Therefore, the upper bound in Theorem 3.1 still holds. Formally, suppose the hitting costs are m-strongly convex and l-strongly smooth, and the switching cost is given by $c(y_{t:t-p}) = \frac{1}{2}\|y_t - \delta(y_{t-1:t-p})\|^2$, where $\delta : \mathbb{R}^{d\times p} \to \mathbb{R}^d$. If the feedback on the minimizers is delayed at most for k rounds, and for any $1 \le i \le p$ there exists a constant $L_i > 0$, such that for any given $y_{t-1}, \cdots, y_{t-i-1}, y_{t-i+1}, \cdots, y_{t-p} \in \mathbb{R}^d$, we have:

$$\|\theta(a) - \theta(b)\| \le L_i\|a - b\|, \forall a, b \in \mathbb{R}^d,$$

where $\theta(x) = \delta(y_{t-1}, \cdots, y_{t-i-1}, x, y_{t-i+1}, \cdots, y_{t-p})$, then the competitive ratio of Algorithm 5 is bounded by

$$O\left((l + 2p^2L^2)^k \max\left\{\frac{1}{\lambda}, \frac{m+\lambda}{m + (1-p^2L^2)\lambda}\right\}\right),$$

where $L = \max_i\{L_i\}$, $\lambda > 0$ and $m + (1 - p^2L^2)\lambda > 0$.

Though we can derive the above results, to find the lower bound and a more accurate upper bound is still challenging.

Algorithm 5 Iterative ROBD for time-varying delay

Parameter: $\lambda \geq 0$
Initialize a ROBD instance with $\lambda_1 = \lambda$, $\lambda_2 = 0$
for $t = 1$ **to** T **do**
  Input: h_t, $v_{l_{t-1}+1:l_t}$
  for $i = l_{t-1} + 1$ **to** l_t **do**
    Observe $f_i(y) = h_i(y - v_i)$
    $\hat{y}_i = \text{ROBD}(f_i, \hat{y}_{i-p:i-1})$
  end for
  Initialize a temporary sequence $s_{1:t}$
  $s_{1:l_t} \leftarrow \hat{y}_{1:l_t}$
  for $i = l_t + 1$ **to** t **do**
    $\tilde{v}_i = \arg\min_v \min_y h_i(y - v) + \lambda c(y, s_{i-1:i-p})$
    Set $\tilde{f}_i(y) = h_i(y - \tilde{v}_i)$
    $s_i \leftarrow \text{ROBD}(\tilde{f}_i, s_{i-p:i-1})$
  end for
  $y_t = s_t$
  Output: y_t (the action at time step t)
end for

B PROOF OF LEMMA 4.2

We begin with a technical lemma, bounding the performance of the oracle decision sequence from ROBD where there is no delay:

Proof. Define $\phi_t = \frac{m+\lambda}{2}\|y_t^{(0)} - y_t^*\|^2$. Since the function

$$g_t(y) = f_t(y) + \frac{\lambda}{2}\|y - \delta(y_{t-1:t-p}^{(0)})\|^2$$

is $(m+\lambda)$-strongly convex, and ROBD selects $y_t^{(0)} = \arg\min_y g_t(y)$, we have that

$$g_t(y_t^{(0)}) + \frac{m+\lambda}{2}\|y_t^{(0)} - y_t^*\|^2 \leq g_t(y_t^*),$$

which implies that

$$H_t^{(0)} + \lambda M_t^{(0)} + \left(\phi_t - \frac{1}{p}\sum_{i=1}^{p}\phi_{t-i}\right) \leq H_t^* + \frac{\lambda}{2}\|y_t^* - \delta(y_{t-1:t-p}^{(0)})\|^2 - \frac{1}{p}\sum_{i=1}^{p}\phi_{t-i}. \tag{21}$$

Applying Jensen's inequality gives

$$\frac{1}{p}\sum_{i=1}^{p}\phi_{t-i} = \frac{m+\lambda}{2p}\sum_{i=1}^{p}\|y_{t-i}^{(0)} - y_{t-i}^*\|^2 \geq \frac{m+\lambda}{2p^2}\left(\sum_{i=1}^{p}\|y_{t-i}^{(0)} - y_{t-i}^*\|\right)^2. \tag{22}$$

Therefore, we can derive the following bound

$$\begin{aligned}
&\frac{\lambda}{2}\|y_t^* - \delta(y_{t-1:t-p}^{(0)})\|^2 - \frac{1}{p}\sum_{i=1}^{p}\phi_{t-i} \\
&\leq \frac{\lambda}{2}\|y_t^* - \delta(y_{t-1:t-p}^{(0)})\|^2 - \frac{m+\lambda}{2p^2}\left(\sum_{i=1}^{p}\|y_{t-i}^{(0)} - y_{t-i}^*\|\right)^2 \qquad (23a)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\lambda}{2}\|y_t^* - \delta(y_{t-1:t-p}^*) - (\delta(y_{t-1:t-p}^{(0)}) - \delta(y_{t-1:t-p}^*))\|^2 - \frac{m+\lambda}{2p^2}\left(\sum_{i=1}^{p}\|y_{t-i}^{(0)} - y_{t-i}^*\|\right)^2 \\
&\le \frac{\lambda}{2}\|y_t^* - \delta(y_{t-1:t-p}^*)\|^2 + \frac{\lambda}{2}\|\delta(y_{t-1:t-p}^{(0)}) - \delta(y_{t-1:t-p}^*)\|^2 \\
&\quad + \lambda\|y_t^* - \delta(y_{t-1:t-p}^*)\| \cdot \|\delta(y_{t-1:t-p}^{(0)}) - \delta(y_{t-1:t-p}^*)\| - \frac{m+\lambda}{2p^2}\left(\sum_{i=1}^{p}\|y_{t-i}^{(0)} - y_{t-i}^*\|\right)^2 \qquad (23b) \\
&\le \frac{\lambda}{2}\|y_t^* - \delta(y_{t-1:t-p}^*)\|^2 + \lambda\|y_t^* - \delta(y_{t-1:t-p}^*)\| \cdot \|\delta(y_{t-1:t-p}^{(0)}) - \delta(y_{t-1:t-p}^*)\| \\
&\quad + \frac{\lambda}{2}\left(L\sum_{i=1}^{p}\|y_{t-i}^{(0)} - y_{t-i}^*\|\right)^2 - \frac{m+\lambda}{2p^2}\left(\sum_{i=1}^{p}\|y_{t-i}^{(0)} - y_{t-i}^*\|\right)^2 \qquad (23c) \\
&= \frac{\lambda}{2}\|y_t^* - \delta(y_{t-1:t-p}^*)\|^2 + \lambda\|y_t^* - \delta(y_{t-1:t-p}^*)\| \cdot \|\delta(y_{t-1:t-p}^{(0)}) - \delta(y_{t-1:t-p}^*)\| \\
&\quad - \frac{m+\lambda(1-p^2L^2)}{2p^2}\left(\sum_{i=1}^{p}\|y_{t-i}^{(0)} - y_{t-i}^*\|\right)^2 \\
&\le \frac{\lambda}{2}\|y_t^* - \delta(y_{t-1:t-p}^*)\|^2 + \frac{\lambda^2p^2L^2}{2(m+\lambda(1-p^2L^2))}\|\|y_t^* - \delta(y_{t-1:t-p}^*)\|^2 \\
&\quad + \frac{m+\lambda(1-p^2L^2)}{2p^2L^2}\|\delta(y_{t-1:t-p}^{(0)}) - \delta(y_{t-1:t-p}^*)\|^2 - \frac{m+\lambda(1-p^2L^2)}{2p^2}\left(\sum_{i=1}^{p}\|y_{t-i}^{(0)} - y_{t-i}^*\|\right)^2 \qquad (23d) \\
&\le \frac{\lambda^2+m\lambda}{2(m+\lambda(1-p^2L^2))}\|y_t^* - \delta(y_{t-1:t-p}^*)\|^2 \\
&\quad + \frac{m+\lambda(1-p^2L^2)}{2p^2L^2}\cdot L^2\left(\sum_{i=1}^{p}\|y_{t-i}^{(0)} - y_{t-i}^*\|\right)^2 - \frac{m+\lambda(1-p^2L^2)}{2p^2}\left(\sum_{i=1}^{p}\|y_{t-i}^{(0)} - y_{t-i}^*\|\right)^2 \qquad (23e) \\
&= \frac{\lambda^2+m\lambda}{2(m+\lambda(1-p^2L^2))}M_t^*.
\end{aligned}
$$

We have applied Equation (22) in Equation (23a), AM-GM inequality in Equation (23b) and Equation (23d), the Lipschitz condition of δ in Equation (23c) and Equation (23e). In this way, we have made a connection between the last two terms of the right hand side in Equation (21), and the switching cost of the offline optimal.

Substituting the above into Equation (21) gives

$$
H_t^{(0)} + \lambda M_t^{(0)} + \left(\phi_t - \frac{1}{p}\sum_{i=1}^{p}\phi_{t-i}\right) \le H_t^* + \frac{\lambda^2+m\lambda}{2(m+\lambda(1-p^2L^2))}M_t^*. \qquad (24)
$$

Focusing on the term in parentheses, we see that

$$
\sum_{t=1}^{T}\left(\phi_t - \frac{1}{p}\sum_{i=1}^{p}\phi_{t-i}\right) = \sum_{i=0}^{p-1}\frac{p-i}{p}(\phi_{T-i} - \phi_{-i}).
$$

Since $\phi_t \geq 0, \forall t$ and $\phi_0 = \phi_{-1} = \cdots = \phi_{-p+1} = 0$, we have

$$\sum_{t=1}^{T} \left(\phi_t - \frac{1}{p} \sum_{i=1}^{p} \phi_{t-i} \right) \geq 0.$$

Now, returning to Equation (24) and summing over time gives

$$\begin{aligned} \sum_{t=1}^{T} H_t^{(0)} + \lambda M_t^{(0)} &\leq \sum_{t=1}^{T} \left(H_t^{(0)} + \lambda M_t^{(0)} \right) + \sum_{t=1}^{T} \left(\phi_t - \frac{1}{p} \sum_{i=1}^{p} \phi_{t-i} \right) \\ &\leq \sum_{t=1}^{T} \left(H_t^* + \frac{\lambda(m+\lambda)}{m + (1-p^2L^2)\lambda} M_t^* \right). \end{aligned}$$

□

With this lemma we can easily get the last inequality in the proof of Theorem 3.1.

C PROOF OF THEOREM 3.3

We prove another preliminary lemma that bounds the distance between iROBD's decision with k-step delay, $y_t^{(k)}$, and the oracle ROBD's decision without delay, $y_t^{(0)}$.

Lemma C.1. *The distance between $y_t^{(0)}$ and $y_t^{(k)}$ can be bounded by:*

$$\|y_t^{(k)} - y_t^{(0)}\|^2 \leq 8\|v_t^{(k)} - v_t^{(0)}\|^2 + 2\alpha^2 \sum_{i=1}^{k-1} \|y_{t-i}^{(k-i)} - y_{t-i}^{(0)}\|^2.$$

Proof. Let $\delta_t^{(k)} = v_t^{(0)} - v_t^{(k)}$. Since $y_t^{(0)} \leftarrow \text{ROBD}(f_t, y_{t-p,t-1}^{(0)})$,

$$\begin{aligned} f_t(y_t^{(0)}) + \frac{\lambda}{2} \left\| y_t^{(0)} - \sum_{i=1}^{p} C_i y_{t-i}^{(0)} \right\|^2 + \frac{m+\lambda}{2} \|y_t^{(0)} - y_t^{(k)} - \delta_t^{(k)}\|^2 \\ \leq f_t(y_t^{(k)} + \delta_t^{(k)}) + \frac{\lambda}{2} \left\| y_t^{(k)} + \delta_t^{(k)} - \sum_{i=1}^{p} C_i y_{t-i}^{(0)} \right\|^2. \end{aligned}$$

Also, we have $y_t^{(k)} \leftarrow \text{ROBD}(f_t^{(k)}, y_{t-1}^{(k-1)}, \cdots, y_{t-k}^{(0)}, \cdots, y_{t-p}^{(0)})$. Then

$$\begin{aligned} f_t(y_t^{(k)} + \delta_t^{(k)}) + \frac{\lambda}{2} \left\| y_t^{(k)} - C_1 y_{t-1}^{(k-1)} - \cdots - C_k y_{t-k}^{(0)} - \cdots - C_p y_{t-p}^{(0)} \right\|^2 + \frac{m+\lambda}{2} \|y_t^{(0)} - y_t^{(k)} - \delta_t^{(k)}\|^2 \\ \leq f_t(y_t^{(0)}) + \frac{\lambda}{2} \left\| y_t^{(0)} - \delta_t^{(k)} - C_1 y_{t-1}^{(k-1)} - \cdots - C_k y_{t-k}^{(0)} - \cdots - C_p y_{t-p}^{(0)} \right\|^2. \end{aligned}$$

Summing yields

$$\begin{aligned} (m+\lambda)\|y_t^{(0)} - y_t^{(k)} - \delta_t^{(k)}\|^2 &\leq \lambda \|\delta_t^{(k)} + \sum_{i=1}^{k-1} C_i (y_{t-i}^{(k-i)} - y_{t-i}^{(0)})\| \|y_t^{(0)} - y_t^{(k)} - \delta_t^{(k)}\| \\ \implies \|y_t^{(0)} - y_t^{(k)}\| &\leq 2\|\delta_t^{(k)}\| + \|\sum_{i=1}^{k-1} C_i (y_{t-i}^{(k-i)} - y_{t-i}^{(0)})\| \qquad (25a) \\ \implies \|y_t^{(0)} - y_t^{(k)}\|^2 &\leq 8\|\delta_t^{(k)}\|^2 + 2\|\sum_{i=1}^{k-1} C_i (y_{t-i}^{(k-i)} - y_{t-i}^{(0)})\|^2 \qquad (25b) \end{aligned}$$

$$\leq 8\|\delta_t^{(k)}\|^2 + 2\Big(\sum_{i=1}^{k-1}\|C_i\|\|y_{t-i}^{(k-i)} - y_{t-i}^{(0)}\|\Big)^2 \tag{25c}$$
$$\leq 8\|\delta_t^{(k)}\|^2 + 2\alpha^2\sum_{i=1}^{k-1}\|y_{t-i}^{(k-i)} - y_{t-i}^{(0)}\|^2.$$

We have used triangle inequality in Equation (25a), AM-GM inequality in Equation (25b) and Jensen inequality in Equation (25c). □

Next, we show that the distance between the action of iROBD and that of ROBD can be bounded via recursions. Then, we turn back to the proof of Theorem 2:

Theorem. *Suppose the hitting costs are m-strongly convex and l-strongly smooth, and the switching cost is given by $c(y_{t:t-p}) = \frac{1}{2}\|y_t - \sum_{i=1}^{p} C_i y_{t-i}\|^2$, where $C_i \in \mathbb{R}^{d\times d}$ and $\alpha = \sum_{i=1}^{p}\|C_i\|$. If there is a k-round feedback delay, then the competitive ratio of iROBD(λ) is*

$$O\left((l+2\alpha^2)^k \max\left\{\frac{1}{\lambda}, \frac{m+\lambda}{m+(1-\alpha^2)\lambda}\right\}\right). \tag{26}$$

Proof. In particular, define the function $\psi : \mathbb{R}^d \to \mathbb{R}^+ \bigcup\{0\}$ as

$$\psi(v) = \min_y h_t(y - v) + \lambda c(y, y_{t-1}^{(k-1)}, \cdots, y_{t-k}^{(0)}, \cdots, y_{t-p}^{(0)}).$$

We can show that ψ is $\frac{m\lambda}{m+\lambda}$-strongly convex, and $v_t^{(k)}$ minimizes it. Thus,

$$\begin{aligned}
&h_t(y_t^{(k)} - v_t^{(k)}) + \lambda c(y_t^{(k)}, y_{t-1}^{(k-1)}, \cdots, y_{t-k}^{(0)}, \cdots, y_{t-p}^{(0)}) + \frac{1}{2}\cdot\frac{m\lambda}{m+\lambda}\|v_t - v_t^{(k)}\|^2 \\
&= \psi(v_t^{(k)}) + \frac{1}{2}\cdot\frac{m\lambda}{m+\lambda}\|v_t - v_t^{(k)}\|^2 \\
&\leq \psi(v_t) \\
&= \min_y h_t(y - v_t) + \lambda c(y, y_{t-1}^{(k-1)}, \cdots, y_{t-k}^{(0)}, \cdots, y_{t-p}^{(0)}) \\
&\leq h_t(y_t^{(0)} - v_t) + \lambda c(y_t^{(0)}, y_{t-1}^{(k-1)}, \cdots, y_{t-k}^{(0)}, \cdots, y_{t-p}^{(0)}) \\
&\leq h_t(y_t^{(0)} - v_t) + \frac{\lambda}{2}\left\|y_t^{(0)} - \sum_{i=1}^{p} C_i y_{t-i}^{(0)} + \sum_{i=1}^{k-1} C_i(y_{t-i}^{(0)} - y_{t-i}^{(k-i)})\right\|^2 \\
&\leq h_t(y_t^{(0)} - v_t) + \lambda\left\|y_t^{(0)} - \sum_{i=1}^{p} C_i y_{t-i}^{(0)}\right\|^2 + \lambda\left\|\sum_{i=1}^{k-1} C_i(y_{t-i}^{(0)} - y_{t-i}^{(k-i)})\right\|^2 \qquad (27a) \\
&\leq h_t(y_t^{(0)} - v_t) + 2\lambda c(y_t^{(0)}, y_{t-1:t-p}^{(0)}) + \lambda\alpha\sum_{i=1}^{k-1}\|C_i\|\|y_{t-i}^{(k-i)} - y_{t-i}^{(0)}\|^2. \qquad (27b)
\end{aligned}$$

We have used the AM-GM inequality in Equation (27a) and Jensen's inequality in Equation (27b).

Since h is l-strongly smooth, for any $\eta_{1,k} > 0$,

$$\frac{1}{1+\eta_{1,k}} h_t(y_t^{(k)} - v_t) \leq h_t(y_t^{(k)} - v_t^{(k)}) + \frac{l}{2\eta_{1,k}}\|v_t - v_t^{(k)}\|^2. \tag{28}$$

Next, using the fact that the function $\frac{\lambda}{2}\|y_t^{(k)} - y\|^2$ is λ-strongly smooth in y, for any $\eta_{2,k} > 0$, we have

$$\begin{aligned}
&\frac{1}{1+\eta_{2,k}} \cdot \frac{\lambda}{2} \left\| y_t^{(k)} - \sum_{i=1}^{p} C_i y_{t-i}^{(k)} \right\|^2 \\
&\quad \le \frac{\lambda}{2} \left\| y_t^{(k)} - \sum_{i=1}^{k-1} C_i y_{t-i}^{(k-i)} - \sum_{i=k}^{p} C_i y_{t-i}^{(0)} \right\|^2 + \frac{\lambda}{2\eta_{2,k}} \left\| \sum_{i=1}^{k-1} C_i (y_{t-i}^{(k)} - y_{t-i}^{(k-i)}) + \sum_{i=k}^{p} C_i (y_{t-i}^{(k)} - y_{t-i}^{(0)}) \right\|^2 .
\end{aligned} \tag{29}$$

Substituting Equation (29) and Equation (28) into Equation (27), we have

$$\begin{aligned}
&\frac{1}{1+\eta_{1,k}} h_t(y_t^{(k)} - v_t) + \frac{1}{1+\eta_{2,k}} \cdot \frac{\lambda}{2} \left\| y_t^{(k)} - \sum_{i=1}^{p} C_i y_{t-i}^{(k)} \right\|^2 \\
&\quad \le h_t(y_t^{(k)} - v_t^{(k)}) + \frac{l}{2\eta_{1,k}} \|v_t - v_t^{(k)}\|^2 \\
&\qquad + \frac{\lambda}{2} \left\| y_t^{(k)} - \sum_{i=1}^{k-1} C_i y_{t-i}^{(k-i)} - \sum_{i=k}^{p} C_i y_{t-i}^{(0)} \right\|^2 + \frac{\lambda}{2\eta_{2,k}} \left\| \sum_{i=1}^{k-1} C_i (y_{t-i}^{(k)} - y_{t-i}^{(k-i)}) + \sum_{i=k}^{p} C_i (y_{t-i}^{(k)} - y_{t-i}^{(0)}) \right\|^2 \\
&\quad \le h_t(y_t^{(0)} - v_t) + 2\lambda c(y_t^{(0)}, y_{t-1:t-p}^{(0)}) + \lambda\alpha \sum_{i=1}^{k-1} \|C_i\| \|y_{t-i}^{(k-i)} - y_{t-i}^{(0)}\|^2 - \frac{1}{2} \cdot \frac{m\lambda}{m+\lambda} \|v_t - v_t^{(k)}\|^2 \\
&\qquad + \frac{l}{2\eta_{1,k}} \|v_t - v_t^{(k)}\|^2 + \frac{\lambda}{2\eta_{2,k}} \left\| \sum_{i=1}^{k-1} C_i (y_{t-i}^{(k)} - y_{t-i}^{(k-i)}) + \sum_{i=k}^{p} C_i (y_{t-i}^{(k)} - y_{t-i}^{(0)}) \right\|^2 \\
&\quad \le h_t(y_t^{(0)} - v_t) + 2\lambda c(y_t^{(0)}, y_{t-1:t-p}^{(0)}) + \frac{l}{2\eta_{1,k}} \|v_t - v_t^{(k)}\|^2 - \frac{1}{2} \cdot \frac{m\lambda}{m+\lambda} \|v_t - v_t^{(k)}\|^2 \\
&\qquad + \frac{\lambda\alpha}{\eta_{2,k}} \sum_{i=1}^{p} \|C_i\| \|y_{t-i}^{(0)} - y_{t-i}^{(k)}\|^2 + \lambda\alpha(1 + \frac{1}{\eta_{2,k}}) \sum_{i=1}^{k-1} \|C_i\| \|y_{t-i}^{(k-i)} - y_{t-i}^{(0)}\|^2 .
\end{aligned} \tag{30}$$

We have used the AM-GM inequality, the triangle inequality, and Jensen's inequality in the last step.

Summing Equation (30) over time and defining $V(k) = \frac{1}{2} \sum_{t=1}^{T} \|v_t^{(0)} - v_t^{(k)}\|^2$ yields

$$\begin{aligned}
&\min \left\{ \frac{1}{1+\eta_{1,k}}, \frac{\lambda}{1+\eta_{2,k}} \right\} \sum_{t=1}^{T} (H_t^{(k)} + M_t^{(k)}) \\
&\qquad \le 2 \sum_{t=1}^{T} (H_t^{(0)} + \lambda M_t^{(0)}) + (\frac{l}{\eta_{1,k}} - \frac{m\lambda}{m+\lambda}) \sum_{t=1}^{T} \frac{1}{2} \|v_t - v_t^{(k)}\|^2 \\
&\qquad\quad + \frac{\lambda\alpha^2}{\eta_{2,k}} \sum_{t=1}^{T} \|y_t^{(0)} - y_t^{(k)}\|^2 + \lambda\alpha^2(1 + \frac{1}{\eta_{2,k}}) \sum_{j=1}^{k-1} \sum_{t=1}^{T} \|y_t^{(j)} - y_t^{(0)}\|^2 \qquad \text{(31a)} \\
&\qquad \le 2 \sum_{t=1}^{T} (H_t^{(0)} + \lambda M_t^{(0)}) + (\frac{l}{\eta_{1,k}} - \frac{m\lambda}{m+\lambda}) \sum_{t=1}^{T} V(k)
\end{aligned}$$

$$+\lambda\alpha^2\left(\frac{1}{\eta_{2,k}}\cdot 16V(k)+\sum_{j=k-1}^{1}(\frac{1+2\alpha^2}{\eta_{2,k}}+1)(1+2\alpha^2)^{k-1-j}\cdot 16V(j)\right). \tag{31b}$$

We have applied Equation (30) in Equation (31a), and Lemma C.1 in Equation (31b). The inequality shows that, the upper bound on the cost of iROBD with delay k does not only involves the estimation error in the k^{th} iteration (i.e. $V(k)$), but also involves errors from all previous iterations of estimation (i.e. $V(j)$, $j=1,\cdots,k-1$). To understand the impact of estimation errors from different iterations, we need to analyse the cost of iROBD under different delays, from 1 to k, as a whole. To do this, define $P(k)=\min\left\{\frac{1}{1+\eta_{1,k}},\frac{\lambda}{1+\eta_{2,k}}\right\}\sum_{t=1}^{T}(H_t^{(k)}+M_t^{(k)})$, and then we have

$$\begin{aligned}\frac{1}{\prod_{i=1}^{k-1}\eta_{2,i}}P(k)\leq&\frac{1}{\prod_{i=1}^{k-1}\eta_{2,i}}P(k)+\frac{1}{\prod_{i=1}^{k-2}\eta_{2,i}}P(k-1)+\cdots+\frac{1}{\eta_{2,1}}P(2)+P(1)\\ \leq&(1+\frac{2}{\eta_{2,1}}+\cdots+\frac{2}{\prod_{i=1}^{k-1}\eta_{2,i}})\sum_{t=1}^{T}(H_t^{(0)}+M_t^{(0)})\\ &+(\frac{l}{\eta_{1,k}}-\frac{m\lambda}{m+\lambda}+\frac{16\lambda\alpha^2}{\eta_{2,k}})\frac{V(k)}{\prod_{i=1}^{k-1}\eta_{2,i}}+\sum_{j=k-1}^{1}a(j)V(j).\end{aligned} \tag{32}$$

Here the coefficient $a(j)$ is

$$a(j)=\frac{1}{\prod_{i=1}^{j-1}\eta_{2,i}}\left(\frac{l}{\eta_{1,j}}-\frac{m\lambda}{m+\lambda}+16\lambda\sum_{i=j+1}^{k}\left(\left(1+\frac{1+2\alpha^2}{\eta_{2,i}}\right)\frac{\alpha^2}{\eta_{2,j}}\prod_{q=j+1}^{i-1}\left(\frac{1+2\alpha^2}{\eta_{2,q}}\right)\right)\right). \tag{33}$$

Here $\eta_{1,i}$ and $\eta_{2,i}$ are parameters from Equation (28) and Equation (29). For $i=1,\cdots,k$, we pick $\eta_{2,i}=\eta_i$ and $\eta_{1,i}=\frac{1-\eta_i+\lambda}{\lambda}$, so that $\frac{1}{1+\eta_{1,j}}=\frac{\lambda}{1+\eta_{2,j}}$. This gives

$$a(j)=\frac{\lambda}{\prod_{i=1}^{j-1}\eta_i}\left(\frac{l}{1+\eta_j-\lambda}-\frac{m}{m+\lambda}+16\sum_{i=j+1}^{k}\left(\left(1+\frac{1+2\alpha^2}{\eta_i}\right)\frac{\alpha^2}{\eta_j}\prod_{q=j+1}^{i-1}\left(\frac{1+2\alpha^2}{\eta_q}\right)\right)\right). \tag{34}$$

When $\eta_j=O(l+2\alpha^2)$ for all j, the negative term $-\frac{m\lambda}{m+\lambda}$ will dominate in $a(j)$. Thus, we can make the coefficient $a(j)$ non-positive, and then we have

$$\begin{aligned}\frac{\lambda}{(1+\eta_k)\prod_{i=1}^{k-1}\eta_i}\sum_{t=1}^{T}(H_t^{(k)}+M_t^{(k)})=&\frac{1}{\prod_{i=1}^{k-1}\eta_i}P(k)\\ \leq&(1+\frac{2}{\eta_1}+\cdots+\frac{2}{\prod_{i=1}^{k-1}\eta_i})\sum_{t=1}^{T}(H_t^{(0)}+\lambda M_t^{(0)})\\ \leq&(1+\frac{2}{\eta_1}+\cdots+\frac{2}{\prod_{i=1}^{k-1}\eta_i})\sum_{t=1}^{T}\left(H_t^*+\frac{\lambda(m+\lambda)}{m+(1-\alpha^2)\lambda}M_t^*\right).\end{aligned} \tag{35a}$$

Finally, recall that the sequence $\{y_t^{(0)}\}$ is identical to the decisions of ROBD. Thus, Equation (35a) follows from the analysis on ROBD in [31], which shows that

$$\sum_{t=1}^{T}(H_t^{(0)}+\lambda M_t^{(0)})\leq\sum_{t=1}^{T}\left(H_t^*+\frac{\lambda(m+\lambda)}{m+(1-\alpha^2)\lambda}M_t^*\right), \tag{36}$$

Therefore, we have that

$$\sum_{t=1}^{T}(H_t^{(k)}+M_t^{(k)}) \le \frac{1+\eta_{2,k}}{\lambda}\Big(\prod_{i=1}^{k-1}\eta_i + 2\sum_{i=2}^{k-1}\prod_{j=i}^{k-1}\eta_j + 2\Big)\sum_{t=1}^{T}\left(\frac{1}{\lambda}H_t^* + \frac{m+\lambda}{m+(1-\alpha^2)\lambda}M_t^*\right),$$

which immediately leads to a bound on the competitive ratio of iROBD of $(O(l{+}2\alpha^2))^k \max\{\frac{1}{\lambda}, \frac{m+\lambda}{m+(1-\alpha^2)\lambda}\}$. □

D PROOF AND EXAMPLE OF THEOREM 5.1

In this section we will present a reduction (Algorithm 3) from the control problem mentioned in Section 5.1 to online convex optimization with structured memory and feedback delay, and then provide a proof of Theorem 5.1.

We restate here the online control problem:

$$\begin{aligned}\min_{u_t}\ & \sum_{t=1}^{T}\frac{q_t}{2}\|x_t\|^2 + \sum_{t=0}^{T-1}\frac{1}{2}\|u_t\|^2\\ \text{s.t.}\quad & x_{t+1} = Ax_t + Bu_t + w_t,\end{aligned}$$

We are going to transform the control problem above into the form of minimizing $\sum_{t=1}^{T} f_t(y_t) + c(y_{t:t-p})$. Before presenting the reduction, we introduce necessary notations for (1) canonical pair (A, B); (2) extracted matrices $\{C_i\}$; (3) accumulative disturbances $r(t, i, j)$:

In Equation (19) we have assumed pair (A, B) is in controllable canonical form. To be specific, canonical (A, B) is in the following form:

$$A = \begin{bmatrix}
0 & 1 & 0 & & 0 & & & & & & & & & & \\
0 & 0 & 1 & & 0 & & & & & & & & & & \\
\vdots & & & \ddots & & & & & & & & & & & \\
0 & 0 & 0 & \cdots & 1 & & & & & & & & & & \\
* & * & * & \cdots & * & * & * & * & \cdots & * & & \cdots & & * \\
 & & & & & 0 & 1 & 0 & & 0 & & & & & \\
 & & & & & 0 & 0 & 1 & & 0 & & & & & \\
 & & & & & \vdots & & & \ddots & & & & & & \\
 & & & & & 0 & 0 & 0 & \cdots & 1 & & & & & \\
* & * & * & \cdots & * & * & * & * & \cdots & * & & \cdots & & * \\
 & & & & & & & \vdots & & & & & & & \\
 & & & & & & & & & & 0 & 1 & 0 & & 0 \\
 & & & & & & & & & & 0 & 0 & 1 & & 0 \\
 & & & & & & & & & & \vdots & & & \ddots & \\
 & & & & & & & & & & 0 & 0 & 0 & \cdots & 1 \\
* & & & & & & & \cdots & & & * & * & * & \cdots & *
\end{bmatrix}, B = \begin{bmatrix}
0 & 0 & \cdots & \\
0 & 0 & \cdots & \\
\vdots & & & \\
0 & 0 & \cdots & \\
1 & 0 & \cdots & \\
0 & 0 & \cdots & \\
0 & 0 & \cdots & \\
\vdots & & & \\
0 & 0 & \cdots & \\
0 & 1 & 0 & \cdots \\
\vdots & & & \\
0 & 0 & \cdots & \\
0 & 0 & \cdots & \\
\vdots & & & \\
0 & 0 & \cdots & \\
0 & 0 & \cdots & 1
\end{bmatrix},$$

where each $*$ represents a (possibly) non-zero entry, and the rows of B with 1 are the same rows of A with $*$ [20, 26]. It is well-known that any controllable system can be linearly transformed to the canonical form. Denote the indices of non-zero rows in matrix B to be $\{k_1 \cdots k_d\}$, and denote the set to be $\mathcal{I}$. Define a mapping $\psi : \mathbb{R}^n \to \mathbb{R}^d$ as $\psi(x) = \left(x^{(k_1)}, \cdots, x^{(k_d)}\right)^T$. Let $p_i = k_i - k_{i-1}$ for $1 \le i \le d$, where $k_0 = 0$. The controllability index of (A, B) is defined by $p = \max\{p_i\}$ [20].

Also, define $C_i \in \mathbb{R}^{d\times d}$ for $i = 1, \cdots, p$ [1] as elements extracted from A for $1 \le i \le p$, $1 \le h \le d$ and $1 \le j \le d$,

$$C_i(h, j) = \begin{cases} A(k_h, k_j + 1 - i) & \text{if } i \le p_j \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

[1]We slightly abuse the notation C_i. In Theorem 3.3 C_i could be any matrix in $\mathbb{R}^{d\times d}$, but here C_i is a specific matrix from rearranging A.

Moreover, for $1 \le i \le d$ and $1 \le j \le p_i$ define $r(t,i,j)$ as accumulative disturbances over time on the system state:

$$r(t,i,j) = \sum_{\tau=t+1-j}^{t-1} w_\tau^{(k_i-\tau+t-j)}, \tag{38}$$

for $j \ge 2$; and $r(t,i,1) = 0$ for $j = 1$. In this way, we can turn any element of x_t into sum of $\psi(\cdot)$ and $r(t,\cdot,\cdot)$:

$$x_t^{(1-j+k_i)} = \left(\psi(x_{t-j+1})\right)^{(i)} + r(t,i,j). \tag{39}$$

The first term of the right hand side involves the system state at step $t-j+1$, while the second term involves the disturbances to the system from step $t-j+1$ to step $t-1$. This decomposition uses a different approach than that in previous work such as [31], and is the first to deal with the coupling between system state and future disturbances, which is what leads to feedback delay in the resulting online optimization problem.

Now, we can restate the theorem and begin the proof:

Theorem. *Consider the online control problem in Equation (19). Assume the coefficients $q_{t:t+p-1}$ are observable at step t. It can be converted to an instance of OCO with structured memory and feedback delay using Algorithm 3.*

Proof. Recall that we define operator $\psi : \mathbb{R}^n \to \mathbb{R}^d$ as

$$\psi(x) = \left(x^{(k_1)}, \cdots, x^{(k_d)}\right)^T. \tag{40}$$

Let $z_t = \psi(x_t)$, that is, $z_t^j = x_t^{(k_j)}$. For $i \notin \mathcal{I}$, we have $x_t^{(i)} = x_{t-1}^{i+1} + w_{t-1}^{(i)}$. In this way,

$$x_t = (z_{t-p_1+1}^{(1)} + \sum_{\tau=t+1-p_1}^{t-1} w_\tau^{(t-\tau)}, \cdots, z_t^{(1)}, \cdots, z_{t-p_d+1}^{(d)} + \sum_{\tau=t+1-p_d}^{t-1} w_\tau^{(k_d-\tau+t-p_d)}, \cdots, z_t^{(d)})^T.$$

Here $r(t,i,j) = \sum_{\tau=t+1-j}^{t-1} w_\tau^{(k_i+t-\tau-j)}$ for $j \ge 2$. When $j = 1$, $r(t,i,1) = 0$.
Notice that

$$\begin{aligned}\sum_{t=0}^{T} q_t \|x_t\|_2^2 &= \sum_{t=0}^{T} q_t \sum_{i=1}^{d} \sum_{j=1}^{p_i} \left(z_{t+1-j}^{(i)} + r(t,j,i)\right)^2 \\ &= \sum_{t=0}^{T-1} \sum_{i=1}^{d} \left(\sum_{j=1}^{p_i} q_{t+j} \left(z_{t+1}^{(i)} + r(t+j,j,i)\right)^2 \right).\end{aligned}$$

This lets us define a hitting cost

$$h_t(y) = \frac{1}{2} \sum_{i=1}^{d} \left(\sum_{j=1}^{p_i} q_{t+j} \left(y^{(i)} + r(t+j,j,i)\right)^2 \right).$$

In this way, we can transform the total cost as following:

$$\frac{1}{2} \sum_{t=0}^{T} (q_t \|x_t\|^2 + \|u_t\|^2) = \sum_{t=0}^{T-1} h_t(z_{t+1}) + \frac{1}{2}\|u_t\|^2.$$

Recall that coefficients $q_{t:t+p-1}$ are observable at time t. When picking z_{t+1}, we do not know h_t because it depends on information about $r(t+j,i,j)$, which depends on $w_{t+1:t+p-1}$.

We can also represent u_t as follows:

$$\begin{aligned} u_t =& z_{t+1} - \psi(w_t) - A(\mathcal{I}, :)x_t \\ =& z_{t+1} - \psi(w_t) - \sum_{i=1}^{p} C_i z_{t+1-i}. \end{aligned}$$

Next, we recursively define a sequence $\{y_t\}_{t\geq -p}$ as the accumulation of control actions, i.e.

$$y_t = u_t + \sum_{i=1}^{p} C_i y_{t-i}, \forall t \geq 0,$$

where $y_t = 0$ for all $t < 0$. We also recursively define a sequence $\{\zeta_t\}_{t\geq -p}$ as the accumulation of control noise, i.e.

$$\zeta_t = \psi(w_t) + \sum_{i=1}^{p} C_i \zeta_{t-i}, \forall t \geq 0,$$

where $\zeta_t = 0$ for all $t < 0$. Setting $x_0 = 0$ gives the following for all $t \geq -1$

$$z_{t+1} = y_t + \zeta_t.$$

Using the above, we can now formulate the problem as online optimization problem with memory and delay, where the hitting cost function is given by

$$f_t(y) = h_t(y + \zeta_t),$$

and the switching cost is given by

$$\frac{1}{2}\|y_t - \sum_{i=1} C_i y_{t-i}\|^2.$$

Note that h_t involves w_{t+p-1}, which is not revealed until before picking y_{t+p}. In other words, at step t, only $f_{1:t-p}$ is known then, due to the reduction structure, there is a delay of p steps. □

Example. To illustrate the reduction, we consider an example of a 2-d system with the following objective and dynamics:

$$\begin{aligned} &\min_u \sum_{t=0}^{200} \frac{1}{2}\|x_t\|^2 + \frac{1}{2}\|u_t\|^2 \\ &x_{t+1} = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix} x_t + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_t + w_t. \end{aligned}$$

There is a disturbance w_t on the system state x_t and an input u_t. In this setting, the disturbance is unknown and potentially adversarial. To begin, we write the system in the following form:

$$\begin{aligned} x_{t+1}^{(1)} &= x_t^{(2)} + w_t^{(1)} \\ x_{t+1}^{(2)} &= 2x_t^{(2)} - x_{t-1}^{(2)} + u_t + w_t^{(2)}. \end{aligned}$$

To transform the problem, we define

$$\begin{aligned} y_t &= u_t + 2y_{t-1} - y_{t-2}, \\ \zeta_t &= w_t^{(2)} + 2\zeta_{t-1} - \zeta_{t-2}, \\ y_t + \zeta_t &= x_{t+1}^{(2)}, \\ h_t(y) &= y^2 + (y + w_{t+1}^{(1)})^2. \end{aligned} \tag{41}$$

Thus, the control problem is transformed into

$$\min_{y} \sum_{t=0}^{199} h_t(y_t + \zeta_t) + \frac{1}{2}\|y_t - 2y_{t-1} - y_{t-2}\|^2.$$

Note that, from the Equation (41) we can see that, at time t, the new cost function h_t involves the disturbance from the next round, w_{t+1}, which is not revealed yet.

E PROOF OF THEOREM 5.3

For simplicity we consider the trajectory tracking task: the cost function is given by $f_t(x_t) = \frac{1}{2}(x_t - v_t)^\top Q_t(x_t - v_t)$, where $\{v_t\}$ is the desired trajectory to track.

Theorem. *Consider the online control problem in Equation (20). If Q_t is observable at step t, and only the trajectory $v_{1:t-k}$ is known, i.e., there are k steps of feedback delay, then it can be converted to an instance of online optimization with switching cost and feedback delay using Algorithm 4.*

Proof. From the dynamics we know that $u_t = x_{t+1} - Ax_t - \delta(x_t)$. Let $f_t(y) = \frac{1}{2}(y - v_t)^T Q_t(y - v_t)$. Using this we can represent the total cost as

$$\begin{aligned}
&\sum_{t=1}^{T} \frac{1}{2}(x_t - v_t)^T Q_t(x_t - v_t) + \sum_{t=0}^{T-1} \frac{1}{2}\|u_t\|^2 \\
&= \sum_{t=1}^{T} \frac{1}{2}(x_t - v_t)^T Q_t(x_t - v_t) + \sum_{t=0}^{T-1} \frac{1}{2}\|x_{t+1} - Ax_t - \delta(x_t)\|^2 \\
&= \sum_{t=1}^{T} f_t(y_t) + \frac{1}{2}\|y_t - Ay_{t-1} - \delta(y_{t-1})\|^2,
\end{aligned}$$

where $y_t = x_t$ for all t. In this way, we have formulated the problem as online optimization with delay and nonlinear switching cost. Notice that, at time step t, only $v_{1:t-k}$ is known, so there is a k-step delay on the minimizer of the hitting cost function f_t.

□

F REMARK 1

We just consider a linear case where $\delta(y) = Ly$ with $L > 0$ a constant. We prove here that the lower bound on the competitive ratio of any online algorithm in this setting matches the upper bound on the competitive ratio of iROBD, which is

$$\frac{1}{2}\left(1 + \frac{2L + L^2}{m} + \sqrt{\left(1 + \frac{2L + L^2}{m}\right)^2 + \frac{4}{m}}\right).$$

Tightness in this case is a simple application of Theorem 4 in [31], so we restate it here.

Theorem F.1. *When the hitting cost functions are m-strongly convex in feasible set $\mathcal{X}$ and the switching cost is given by $c(y_t, y_{t-1}) = \frac{1}{2}\|y_t - \alpha y_{t-1}\|^2$ for a constant $\alpha \geq 1$, the competitive ratio of any online algorithm is lower bounded by*

$$\frac{1}{2}\left(1 + \frac{\alpha^2 - 1}{m} + \sqrt{\left(1 + \frac{\alpha^2 - 1}{m}\right)^2 + \frac{4}{m}}\right).$$

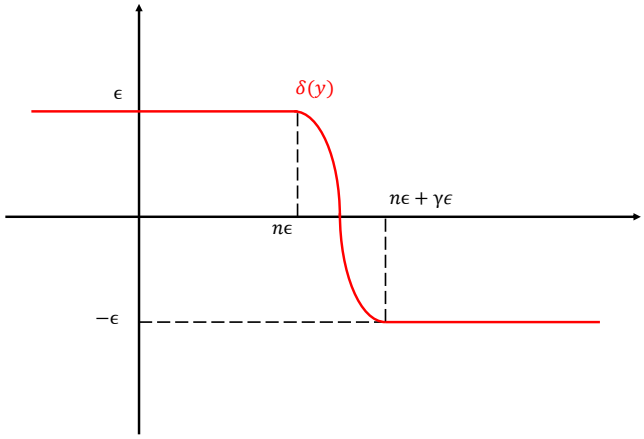

Fig. 2. Illustration of the function $\delta(y)$.

To apply this in our context, we substitute α in the theorem above with $1 + L$ in our setting. This immediately gives that a lower bound on the competitive ratio of any algorithm is

$$\frac{1}{2}\left(1 + \frac{2L + L^2}{m} + \sqrt{\left(1 + \frac{2L + L^2}{m}\right)^2 + \frac{4}{m}}\right),$$

which highlights that iROBD remains optimal in the basic linear setting.

G UNBOUNDED COMPETITIVE RATIO

In this section we show that, even when δ is small, the competitive ratio of any online algorithm can be arbitrarily large when there are nonlinear switching costs. To show this we consider the following example:

$$\sum_{t=1}^{T} (y_t - v_t)^2 + (y_t - y_{t-1} - \delta(y_{t-1}))^2.$$

Suppose the starting point of the online algorithm and the offline adversary is $y_0 = y_0^* = 0$. Let $\epsilon, \gamma > 0$ be two small numbers, and $n \in \mathbb{N}^+$. The function δ is defined as:

$$\delta(y) = \begin{cases} \epsilon, & y \le n\epsilon; \\ -\epsilon \sin\left(\frac{\pi}{\gamma\epsilon} y - \frac{n\pi}{\gamma} - \frac{\pi}{2}\right), & n\epsilon < y \le n\epsilon + \gamma\epsilon; \\ -\epsilon, & y > n\epsilon + \gamma\epsilon. \end{cases}$$

This is plotted in Figure 2. Notice that the absolute value of δ is always no larger than ϵ, and by adjusting the value of ϵ, it can be made it as small as desired.

We consider a sequence $\{v_t\}$ such that the online algorithm follows exactly the trajectory through steps $t = 1, 2, \cdots, n$ and is forced to incur a huge switching cost at step $t = n+1$ while the adversary makes use of the property of δ and departs earlier in order to achieve a much smaller cost. More specifically, for $t = 1, 2, \cdots, n+1$, the trajectory v_t is:

$$v_t = \begin{cases} t \cdot \epsilon, & t \in \{1, 2, \cdots, n\}; \\ (n-1)\epsilon, & t = n+1. \end{cases}$$

Suppose the online algorithm first chooses y_t, which does not equal v_t at step $t = t_0$. If $t_0 < n+1$, we stop the game at step t_0, and compare the online algorithm with an offline adversary which always stays chooses $y_t = v_t$. The total cost of the offline adversary is:

$$\sum_{t=1}^{t_0} (y_t - v_t)^2 + (y_t - y_{t-1} - \delta(y_{t-1}))^2 = \sum_{t=1}^{t_0} (t\epsilon - t\epsilon)^2 - (t\epsilon - (t-1)\epsilon - \epsilon)^2 = 0,$$

but the total cost of the online algorithm is non-zero. So the competitive ratio is unbounded.

Now we consider the case when the algorithm decides on $y_t = v_t = t \cdot \epsilon$ for $i = 1, \cdots, n$. In this case the cost incurred at step $n+1$ is:

$$(y_{n+1} - v_{n+1})^2 + (y_{n+1} - y_n - \delta(y_n))^2 = (y_{n+1} - (n-1)\epsilon)^2 + (y_{n+1} - (n+1)\epsilon)^2 \geq 2\epsilon^2.$$

However, consider another sequence

$$y'_t = \begin{cases} t \cdot \epsilon, & t \in \{0, 1, 2, \cdots, n-1\}; \\ n\epsilon + \gamma\epsilon, & t = n; \\ (n-1)\epsilon & t = n+1. \end{cases}.$$

In this case the cost of $y'_1, y'_2, \cdots, y'_{n+1}$ is

$$\begin{aligned} &\sum_{t=1}^{n+1} (y'_t - v_t)^2 + (y'_t - y'_{t-1} - \delta(y'_{t-1}))^2 \\ &\quad = \sum_{t=n}^{n+1} (y'_t - v_t)^2 + (y'_t - y'_{t-1} - \delta(y'_{t-1}))^2 \\ &\quad = (n\epsilon + \gamma\epsilon - n\epsilon)^2 + (n\epsilon + \gamma\epsilon - (n-1)\epsilon - \epsilon) + ((n-1)\epsilon - n\epsilon - \gamma\epsilon - (-\epsilon))^2 \\ &\quad = 3\gamma\epsilon^2. \end{aligned}$$

This cost is no smaller than the offline optimal; therefore, the competitive ratio of the online algorithm is bounded by

$$\frac{cost(ALG)}{cost(OPT)} \geq \frac{2\epsilon^2}{3\gamma\epsilon^2} = \frac{2}{3\gamma}.$$

Since $\gamma \to 0^+$, we can see the competitive ratio is unbounded.

Received October 2021; revised December 2021; accepted January 2022