# Online Convex Optimization and Predictive Control in Dynamic Environments

Thesis by
Haoyuan Sun

In Partial Fulfillment of the Requirements for the
Degree of
Bachelor of Science

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2021
Submitted June 11, 2021

Haoyuan Sun
ORCID: 0000-0002-6203-0198

# ACKNOWLEDGEMENTS

I would like to thank my advisor Professor Adam Wierman for his invaluable advice and guidance on my academics and research. This project would not be possible without Professor Wierman's tireless support over the last three years.

I would also like to thanks Professor Jason Marden, Professor Leonard Schulman, and Professor Pablo Moscato for their help and mentorship during my time at Caltech. Lastly, I was fortunate to have collaborated with Gautam Goel, David Grimsman, Yang Hu, Yiheng Lin, Guannan Qu and Guanya Shi during my research.

# ABSTRACT

We study the performance of an online learner under a framework in which it receives partial information from a dynamic, and potentially adversarial, environment at discrete time steps. The goal of this learner is to minimize the sum of costs incurred at each time step and its performance is compared against an offline learner with perfect information of the environment.

We are interested in the scenarios where, in addition to some costs at each time step, there are some penalties or constraints on the learner's successive decisions. In the first part of this thesis, we investigate a Smoothed Online Convex Optimization (SOCO) setting where the cost functions are strongly convex and the learner pays a squared $\ell_2$ movement cost for changing decision between time steps. We shall present a lower bound on the competitive ratio of any online learner in this setting and show a series of algorithmic ideas that lead to an optimal algorithm matching this lower bound. And in the second part of this thesis, we investigate a predictive control problem where the costs are well-conditioned and the learner's decisions are constrained by a linear time-varying (LTV) dynamics but has exact prediction on the dynamics, costs and disturbances for the next $k$ time steps. We shall discuss a novel reduction from this LTV control problem to the aforementioned SOCO problem and use this to achieve a dynamic regret of $O(\lambda^k T)$ and a competitive ratio of $1 + O(\lambda^k)$ for some positive constant $\lambda < 1$.

# TABLE OF CONTENTS

*Chapter 1*

# INTRODUCTION

We will study optimization as an iterative process where the underlying environment is so complex that the solver can only a piece of the input or observation at once and commit to some decisions before rest of the information are revealed. This notion encompasses a wide of range of problems and had been extensively studied over the last several decades, e.g. the metrical task problem [1, 2], the convex body chasing problem [3], the expert problem [4], and many more.

Specifically, over a sequence of time steps $1, \dots T$, an online learner makes a decision $x_t$ and then suffers a loss according to a cost function $f_t$. This problem is called the Online Convex Optimization problem when $f_t$ are convex [5]. We are interested in settings where the online learner is additionally equipped with *prediction* and *memory*. In particular, we can use memory of previous decision points to introduce a *switching cost* (sometimes known as *movement cost*), which can be written in the form of $c_t(x_{t-1}, x_t)$ when the memory has size of one, to better capture the dynamical context of the problem [6]. And we can leverage prediction on future properties of the environment to allow the learner to make better decisions. The simplest case is to reveal the current time cost function $f_t$ before the learner is committed to a decision $x_t$. The Smoothed Online Convex Optimization (SOCO), or OCO with switch cost, problem is one example of such idea where the prediction and memory both have length of one.

We additionally can introduce more sophistication to the environment by enforcing some system dynamics where the next decision point $x_{t+1}$ is constrained to be a function of the current decision point $x_t$, an control action $u_t$, and possibly some disturbance $w_t$. Then the switching cost can be written as a function of $u_t$ and $w_t$. This framework is wildly used in the control theory community and practical algorithms such as the Model Predictive Control (MPC) use prediction to achieve good performance both in theory and practice [7].

Recently, considerable efforts have been made to prove learning guarantees on the SOCO problem and control with prediction. On a high level, we compare the total cost incurred by the online learner to the optimal cost of an offline solver which has complete access to the past and future knowledge of the environment. The SOCO problem is most often studied under the metric called competitive ratio [8–12], whereas the control problem had being studied under a variety of metrics, such as static regret [13–19], dynamic regret [20, 21] and competitive ratio [22, 23]. In this thesis, we shall present several recent results on the performance of SOCO and control with prediction. And along the performance bounds, we also illuminate some crucial connections between these two problem that had garnered substantial interests in recent years [6, 14, 15, 20, 24].

## 1.1 Smoothed Online Convex Optimization

The problem of Smoothed Online Convex Optimization (SOCO), which is a generalization of the OCO problem where an online learner plays a series of rounds labeled $1, \ldots, T$ and incur a convex cost at each round. The SOCO problem differs from OCO that the learner is additionally penalized for changing its decision between rounds. We can consider the setting as a game where an online learner plays against an adaptive adversary. In each round, the adversary picks a convex cost function $f_t : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ and presents this to the learner. After observing the cost function, the learner decides on $x_t \in \mathbb{R}^n$ and pays a *hitting cost* $f_t(x_t)$, as well as a *movement cost* $c(x_t, x_{t-1})$.

SOCO was originally proposed in the context of dynamic power management in data centers [25], where the movement cost models the wear-and-tear from re-configuring the servers. It has since then seen a wealth of applications, from speech animation to management of electric vehicle charging [26–28], and more recently applications in control [24, 29] and power systems [30, 31].

Additionally, SOCO has connections to a number of other important problems in online algorithms and learning. It has been shown that SOCO is related to online logistic regression and smoothed online maximum likelihood estimation [24]. Convex Body Chasing (CBC), introduced in [3], is related to SOCO that instead of a hitting cost function at each round, the online learner is given a convex shape which its decision point must lie on – the resulting problem would only contain a movement cost. It has been show that CBC can be reduced to SOCO [32, 33]. The problem of designing competitive algorithms for Convex Body Chasing has attracted much recent attention. e.g. [32, 34–36]. SOCO can also be viewed as a continuous version of the Metrical Task System (MTS) problem (see [1, 2, 37]). A special case of MTS is the celebrated $k-$server problem, first proposed in [38], and has received significant attention in recent years (see [39, 40]).

Given these connections, the design and analysis of algorithms for SOCO and related problems has received considerable attention in the last decade. Despite the wealth of literature, there are several weaknesses to the earlier analysis of SOCO. The prior results gave constant competitive bounds in one dimension. For instance, SOCO was first studied in the scalar setting in [8], which used SOCO to model dynamic "right-sizing" in data centers and gave a 3-competitive algorithm. Later, [9] showed a 2-competitive algorithm, also in the scalar setting, that matches the lower bound for online algorithms in this setting [10]. To break the barrier on dimensions, many works had to rely on the online algorithm having access to prediction of future cost functions (see [11, 12, 25, 30]).

A breakthrough came in 2017 when [41] proposed a new algorithm, Online Balanced Descent (OBD), and showed that it is constant competitive in all dimensions in the setting where the hitting costs are locally polyhedral and movement costs are the $\ell_2$ norm. However, the class of polyhedral functions is rather restrictive and does not properly model most loss functions used in machine learning.

Here, we present the results from [24, 42] that extend OBD to more general settings. First, [24] showed that OBD is also constant competitive, specifically $3 + O(1/m)$-competitive, when the hitting costs are $m$-strongly convex and the movement costs are the squared $\ell_2$ norm (see Theorem 2.2). Then [42] gave the first lower bound on the performance of any online learner with $m$-strongly convex hitting costs and squared $\ell_2$ norm as the movement cost. As stated in Theorem 2.3, as $m$ tends to zero, the best competitive achievable by any online algorithm is at least $\Omega(m^{-1/2})$. We will show that OBD fails to match this bound in Theorem 2.5 that the competitive ratio of OBD is at least $\Omega(m^{-2/3})$ as $m$ goes to zero.

We will make a crucial observation that a much simpler greedy approach results in surprisingly good performance (see Theorems 2.1 and 2.4). This motivated us to construct the current state-of-art of SOCO algorithm, called Regularized OBD (R-OBD) [42]. In Theorem 2.6, we see that R-OBD's competitive ratio matches the performance lower bound, including the constants. Thus, R-OBD is an optimal SOCO algorithm in the setting of $m$-strongly convex hitting cost and squared $\ell_2$ movement cost. In fact, R-OBD can be generalized to a much more general model, but we will not show this result here for clarity's sake and we encourage interested readers to check out [42].

### 1.2 Predictive Control

We are also interested in the problem of predictive control in a linear time-varying (LTV) system, where the dynamics is given by $x_{t+1} = A_t x_t + B_t u_t + w_t$. Here, $x_t$ is the state, $u_t$ is the control action, and $w_t$ is the disturbance or exogenous input. At each time step $t$, the online controller incurs a time-varying state cost $f_t(x_t)$ and control cost $c_t(u_{t-1})$. Then the controller decides each action $u_t$ by making use of predictions of the next $k$ future disturbances, cost functions, and dynamical matrices, and seeks to minimize its total cost on a finite horizon $T$. In this paper, we will illustrate the connection between this problem and SOCO. We will leverage this connection to show the dynamic regret and competitive ratio of predictive controllers in this LTV setting.

Recently, a growing literature has sought to design controllers that achieve learning guarantees such as static regret [15, 19], dynamic regret [20, 43], and competitive ratio [6]. One notable line of work concerns predictive control with learning guarantees and studies how the regret and competitive ratio can be improved by changing the prediction window $k$. Prior work had mostly focused on linear time-invariant (LTI) systems [7, 20, 43, 44]. However, linear time-varying (LTV) systems have received increasing attention in recent years due to their importance in a variety of emerging applications, despite the challenges associated with analysis.

Currently, the performance guarantees in the LTV setting is poorly-understood due to a lack of progress in developing new techniques to generalize the dynamics from LTI to LTV systems and the costs from quadratic to well-conditioned functions. Specifically, the proof approaches used in previous studies on regret and competitive ratio of predictive control in LTI dynamics with quadratic costs, e.g., [7, 43, 44], require explicit representation of the cost-to-go function, optimal control actions, and algorithm's actions in terms of the system parameters. This is very difficult to generalize to non-quadratic cost functions. A promising alternative is through reductions from optimal control to online convex optimization with multi-step memory, e.g., [6, 14, 15, 20, 24]. However, such reductions usually do not work well for LTV systems because the problem must be written in control canonical form [6, 20], or because they are limited on the policy class and comparisons to static benchmarks [14, 15].

Perhaps the most prominent approach for controlling LTV systems is Model Predictive Control (MPC), also known as Receding Horizon Control [45]. Generally speaking, at each time step, an MPC-style algorithm solves a predictive trajectory for the future $k$ time steps and commit the first control action in this trajectory. MPC-style algorithms are known to work well in practice, even when the dynamics are non-linear and time-varying, e.g., [46–49]. On theoretical side, the asymptotic behaviors of MPC such as stability and convergence have been studied intensively under general assumptions on dynamics and costs [50–53]. However, non-asymptotic guarantees such as regret and competitive ratio of MPC-style policies have been limited, especially for LTV systems.

To address this shortcoming, we introduced a novel reduction between LTV control and the SOCO problem [54]. Connections between online optimization and control have received increasing attention in recent years, e.g., [6, 14, 15, 20, 24]. While existing reductions rely on the canonical form, which does not apply to LTV systems, and/or formulations of online optimization with memory of multiple prior time steps, which makes the online problem more challenging, the reduction we present here is a fundamentally different approach and do not suffer the same weaknesses.

This analysis framework based on a perturbation approach. Specifically, instead of solving for the optimal states and control actions like previous analysis in the LTI setting with quadratic costs, e.g., [43, 44], we bound how much impact an perturbation to the system parameters can have on the optimal solution. A result of this type can be shown under the SOCO setting (Theorem 3.3), where the predictive states are not limited by the system's dynamics. Then, it is shown that, under some mild assumptions, one can rewrite a predictive control problem in a SOCO problem by dividing the time steps into equal chunks and consider each chunk as one optimization problem. The resulting perturbation bound (Theorem 3.5) can be shown even when the optimal trajectory cannot be written down explicitly, which allows it to be applied in LTV systems with well-conditioned costs.

We can then provide the regret and competitive ratio results for a controller in LTV systems with time-varying dynamics and costs. Specifically, we show that an MPC-style predictive control algorithm (Algorithm 4) achieves a dynamic regret that decays exponentially with respect to the length of prediction window $k$ in the LTV system (Theorem 3.9): $O(\lambda^k T)$, where the decay rate $\lambda$ is a positive constant less than one. Also, with a variation of predictive control (Algorithm 5), we also show the first competitive bound in LTV systems with time-varying well-conditioned costs (Theorem 3.10): $1 + O(\lambda^k)$, where the decay rate $\lambda$ is identical with the one in the regret bound.

We note this analysis is not specific to the predictive control algorithm we study, and we expect it to prove useful for other controllers in future work. A limitation of this reduction framework is that it cannot handle state/control constraints. This limitation is shared by previous works [7, 20, 43, 44], and represents a challenging open question in the literature.

*Chapter 2*

# SMOOTHED ONLINE CONVEX OPTIMIZATION

## 2.1 Model and Preliminaries

An instance of Smoothed Online Convex Optimization (SOCO) played on rounds $t = 1, 2, \ldots, T$ consists of an initial point $x_0 \in \mathbb{R}^d$, a sequence of non-negative convex cost functions $f_1 \ldots f_t : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, and a movement cost $c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$. In every round, a potentially adversarial cost function $f_t$ is given to an online learner. After observing the cost function, the learner chooses an action $x_t \in \mathbb{R}^d$ and pays a cost that is the sum of the *hitting cost*, $f_t(x_t)$, and the *movement cost* (also known as the switching cost) $c(x_t, x_{t-1})$. The online learner seeks to minimize its total cost over all $T$ rounds:

$$cost(ALG) = \sum_{t=1}^{T} f_t(x_t) + c(x_t, x_{t-1}).$$

It is worth noting that, if there were no movement costs, e.g. $c(x_t, x_{t-1}) = 0$, the problem would be trivial – the learner would always pay the optimal cost simply by picking the action that minimizes the hitting cost in each round, i.e., by setting $x_t = \arg\min_x f_t(x)$. A nonzero movement cost is key to making this problem challenging. Since movement cost couples the learner's decisions across rounds, the optimal action of the learner depends on unknown future costs.

There is a long literature on SOCO, both focusing on algorithmic questions, e.g., [8, 9, 24, 41], and applications, e.g., [25–28]. The variety of applications studied means that a variety of assumptions about the movement costs have been considered. Motivated by applications to data center capacity management, movement costs have often been taken as the $\ell_1$ norm, i.e., $c(x_1, x_2) = \|x_1 - x_2\|_1$, e.g. [8, 9]. However, recently, other norms have been considered and the setting of squared $\ell_2$ movement costs has gained attention due to its use in online regression problems and connections to LQR control, among other applications (see [24, 29, 55]).

In this paper, we focus on modeling the movement cost by the squared $\ell_2$ norm, i.e. $c(x_2, x_1) = \frac{1}{2}\|x_2 - x_1\|_2^2$ and the hitting cost by $m$-strongly convex function, as defined below:

**Definition 2.1.** *A real-valued function $g : \mathbb{R}^n \to \mathbb{R}$ is called $\ell$-strongly smooth if*

$$g(y) \le g(x) + \langle \nabla g(x), y - x \rangle + \frac{\ell}{2} \|y - x\|_2^2$$

*and is called $m$-strongly convex if*

$$g(y) \ge g(x) + \langle \nabla g(x), y - x \rangle + \frac{m}{2} \|y - x\|_2^2$$

*for any $x, y \in \mathbb{R}^n$. Here $\langle \cdot, \cdot \rangle$ denotes the standard inner product of vectors.*

Finally, it is worth noting that there are more general models involving the Bregman divergence for movement costs and quasi-convex hitting costs, and results under this setting can be found in [42].

### Competitive Ratio and Regret

The typical reference point in the analysis of SOCO algorithms is the *offline optimal*, which the best solution from an omniscient player who has full knowledge of all future cost functions $\{f_t\}$ at the start. Specifically, the cost of the offline optimal is defined as

$$cost(OPT) = \min_{x_1 \dots x_T} \sum_{t=1}^{T} f_t(x_t) + c(x_t, x_{t-1}).$$

Since it is clearly impossible for an online player to match the offline optimal, our primary goal is to design online algorithms that nearly match the performance of the offline optimal algorithm. The *competitive ratio* is a common metric to measure the performance of SOCO algorithms. It is the worst-case ratio of the online learner's total costs versus the offline optimal costs. More precisely, it is defined as

$$\sup_{f_1 \dots f_T} \frac{cost(ALG)}{cost(OPT)}.$$

Another important performance measure of interest is the *regret*. The regret compares the cost of the online learner to the static optimal, which is best solution without any movement. More precisely, it is defined as:

$$cost(ALG) - \min_{x \in \mathbb{R}^d} \sum_{t=1}^{T} f_t(x)$$

A natural generalization of the regret is to allow a movement budget for the offline solution. This motivates us to define the *L-constrained dynamic regret*:

$$cost(ALG) - \min_{x_1, \dots, x_T} \sum_{t=1}^{T} f_t(x_t) + c(x_t - x_{t-1})$$

**Algorithm 1** Greedy Algorithm

```
procedure GREEDY(f_t, x_{t-1})            ▷ Procedure to select x_t
    v_t ← arg min_x f_t(x)
    x_t ← v_t
    return x_t
```

$$\text{s.t.} \sum_{t=1}^{T} \|x_t - x_{t-1}\| \leq L$$

The desirable performance of an online algorithm is typically *constant competitiveness* or *sublinear regret*, which respective mean that the competitive ratio is bounded above by a constant or the regret is $o(T)$ with respect to the total time $T$. While regret and competitive ratio both compare the online learned against a solution made with full knowledge of the cost functions, regret does not fully capture the dynamical environment that is potentially adversarial. Hence, we will focus on the competitive ratio of SOCO algorithms. Nonetheless, regret is an important metric and widely used in the online algorithm community when the environment is (nearly) static [5].

### 2.2 Motivating Algorithms

**Greedy Algorithm**

One of the simplest algorithm is to choose the minimizer $v_t = \arg\min_x f_t(x)$ at every time step (see Algorithm 1).

In the extreme case where the switching cost $c_t$ is zero, then the greedy algorithm is equivalent to the offline optimal. But when there is a switch cost, the greedy algorithm may incur a large movement when the minimizer of successive cost functions $f_t$ are far away. This is potentially problematic when the cost functions are very "flat" and the movement cost dominates. But surprisingly, the greedy algorithm can still perform quite well.

**Theorem 2.1.** *Consider hitting cost functions that are $m$-strongly convex with respect to $\ell_2$ norm and movement costs given by $\frac{1}{2}\|x_t - x_{t-1}\|_2^2$. The greedy algorithm is $(1 + \frac{4}{m})$-competitive.*

Note that Theorem 2.1 has competitive ratio of 1 as the strong convexity parameter $m$ approaches infinity. This makes sense since we can ignore movement costs when the cost functions are very steep. Also, it is worth noting that the greedy algorithm is not constant competitive in many other settings with different classes movement and hitting cost functions.

**Algorithm 2** Online Balanced Descent (OBD)

**procedure** OBD($f_t, x_{t-1}$) ▷ Procedure to select $x_t$
  $v_t \leftarrow \arg\min_x f_t(x)$
  Let $x(\ell) = \prod_{K_t^\ell}(x_{t-1})$. Initialize $\ell = f_t(v_t)$. Here $K_t^\ell = \{x | f_t(x) \leq \ell\}$.
  Increase $\ell$. Stop when $c(x(\ell), x_{t-1}) = \gamma(\ell - f_t(v_t))$.
  $x_t \leftarrow x(\ell)$.
  **return** $x_t$

**Online Balanced Descent**

Online Balanced Descent (OBD) was a major breakthrough in the performance of SOCO algorithm. OBD is formally defined in Algorithm 2. OBD works because the level sets of the cost functions $f_t$ are convex. As illustrated in Figure 2.1, in every round, OBD projects the previously chosen point $x_{t-1}$ onto a level set of the current cost function $f_t$, which is chosen so that the ratio between the hitting costs and movement costs are kept constant. Intuitively, OBD maintain a "balance" between the movement costs and hitting costs of the learner by a fixed balancing ratio which we shall call $\gamma$. For any fixed $\gamma$, the desired level set can then be efficiently selected via binary search. With an appropriate choice of $\gamma$, OBD ensures that neither the movement nor hitting cost is too high, and thus both can be compared to the offline optimal and satisfy a constant competitive ratio.

OBD has a remarkable semblance to other techniques based on gradient descent or mirror descent [41], but it notably differs that the direction of each update step is determined by the gradient of the current cost function $f_t$ rather than the previous cost function $f_{t-1}$. A more close analog of OBD is a proximal algorithm with a dynamic step size [56]. Similar to proximal algorithms, OBD iteratively projects the previously chosen point onto a level set of the cost function. And unlike traditional proximal algorithms, OBD selects the level set adaptively in each round to carefully maintain a balance between the hitting and movement costs. This interpretation will become significant when we introduce the Regularized OBD (R-OBD), in which we shift from the original geometry interpretation of OBD and instead consider it algebraically as a proximal algorithm. We will add a special regularization term in the objective of R-OBD to help enforce a "greedier" behavior of the online learner.

OBD was first introduced in [41] and was shown to have a constant and dimension-free competitive ratio under the model where the movement costs are the $\ell_2$ norm and the hitting costs are locally polyhedral, i.e. grow at least linearly with respect to the distance from the minimizer. This was the first SOCO algorithm with constant

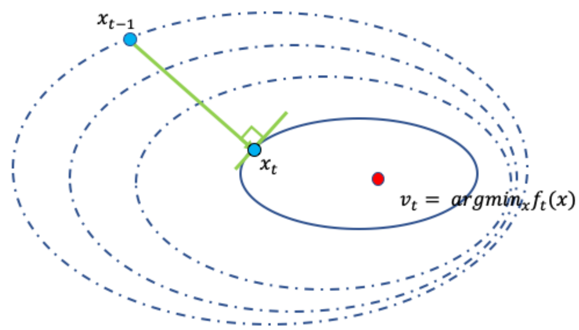


Figure 2.1: One step of the OBD algorithm. The dashed lines denote the level sets of the current hitting cost function $h_t$. The green arrow denotes the direction of the movement. The decision point $x_t$ is chosen as the projection onto an appropriately chosen level sets, which is shaded solid.

competitive beyond one-dimensional action spaces and free of predictions on future cost functions. And the same paper had shown that a variation of OBD that uses a different balance condition can achieve $O(\sqrt{TL})$ $L$-constrained regret for locally polyhedral hitting costs. And in [24], it was also shown that OBD also has a constant, dimension-free competitive ratio when movement costs are the squared $\ell_2$ norm and hitting costs costs are strongly convex. It is worth noting that [3] had shown that the competitive ratio of any online learner must grow exponentially with respect to dimension if the hitting costs can be arbitrary convex functions. But, the results regarding OBD show that having polyhedral or strong convex hitting cost functions are sufficient conditions to deliver constant competitive SOCO algorithms, independent of dimensions.

**Theorem 2.2.** *Consider hitting cost functions that are $m$-strongly convex with respect to $\ell_2$ norm and movement costs given by $\frac{1}{2}\|x_t - x_{t-1}\|_2^2$. The OBD algorithm is $(3 + O(1/m))$-competitive under the choice $\gamma = 2 + \frac{10}{m}$.*

While OBD seemingly fails to match the performance of the greedy algorithm in the limit of $m \to \infty$, we will see in the next section that the performance as $m \to 0^+$ is of greater concern and OBD can potentially outperform the greedy algorithm in this case.

## 2.3 Lower Bound

In this section, we focus on finding the least competitive ratio that can be attained by online algorithms for SOCO and to illustrate the need for improving upon OBD. Generally, [41] proves that the competitive ratio of any online algorithm is bounded below by $\Omega(\sqrt{d})$, where $d$ is the dimension of the action space. However, there are many problems of interests where better performance is possible. In particular, from the previous section, we know that when the hitting costs are $m$-strongly convex, OBD provides a dimension-free competitive ratio of $3 + O(1/m)$. We are also interested in knowing whether it is possible to achieve better performance. Before [42], there are no known lower bounds on the competitive ratio of SOCO algorithms in the strongly convex hitting cost setting. So, we will present the lower bounded from [42] that in turn motivates an optimal SOCO algorithm in this setting.

We first bring a general lower bound on the competitive ratio of SOCO algorithms when the hitting costs are strongly convex and the movement costs are the quadratic $\ell_2$ norm. And then we present lower bounds specific to the greedy algorithm and OBD, demonstrating that there exists a gap between the competitive ratios of the two algorithms and the general lower bound. In the next section, we will bring together ideas from the greedy algorithm and OBD to match our general lower bound, including constants.

We begin by stating the first lower bound for strongly convex hitting costs in SOCO.

**Theorem 2.3.** *Consider hitting cost functions that are $m$-strongly convex with respect to $\ell_2$ norm and movement costs given by $\frac{1}{2}\|x_t - x_{t-1}\|_2^2$. Any online algorithm must have a competitive ratio at least $\frac{1}{2}\left(1 + \sqrt{1 + \frac{4}{m}}\right)$.*

The proof of Theorem 2.3 can be found in Appendix A.2. Since a quadratic function is convex, we observe that, reaching a target point via one large step is incurs more movement cost than reaching it by taking many small steps. Specifically, we consider a scenario on the real line where the online learner starts at $x = 0$ and we first reveal a long sequence of "flat" cost functions whose minimizers are at zero. At this point, an online learner cannot move; otherwise we can terminate the game immediately and the resulting competitive ratio is infinity. Then, we reveal a very steep cost function with minimizer at $x = 1$ which forces the online learner to move very close to 1. In contrast, an offline algorithm with a complete knowledge of the cost functions can split the journey to $x = 1$ over many smaller steps. Since we make the cost functions, except for the last round, to be as flat as possible, the offline algorithm's saving in movement cost overshadows the additional hitting costs it had to incur, resulting in a large competitive ratio.

Importantly, the lower bound in Theorem 2.3 highlights the dependence of the competitive ratio on $m$, the strong convexity parameter. Because we cannot reveal

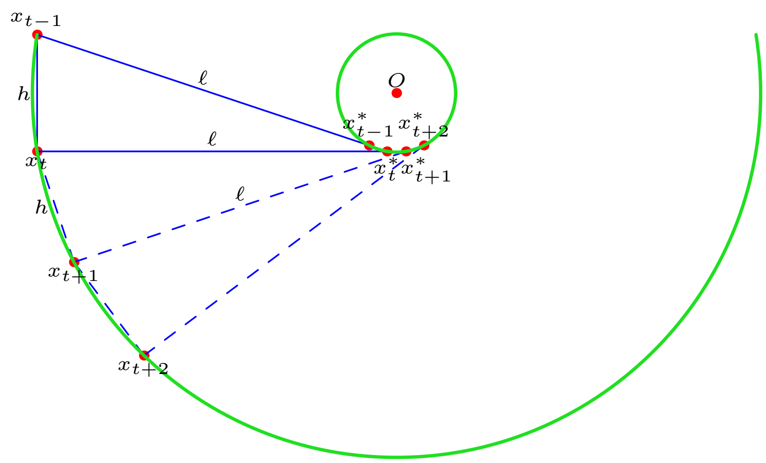


Figure 2.2: Counterexample used to prove Theorem 2.5. In the figure, $\{x_t\}$ are the choices of OBD and $\{x_t^*\}$ are the choices of the offline optimal. This figure is taken from [24].

flat cost functions when $m$ is large, the example we just described works the best in the limit as $m$ approaches 0. So we shall take a particular interest in the asymptotic behavior of the competitive ratio of SOCO algorithms for small $m$. The general lower bound is on the order of $\Omega(m^{-1/2})$ as $m \to 0^+$ and we will see that both the greedy algorithm and OBD exhibit worse competitive ratio in this case. The proofs of these statements can be found in Appendices A.3 and A.4.

**Theorem 2.4.** *Consider hitting cost functions that are $m$-strongly convex with respect to $\ell_2$ norm and movement costs given by $\frac{1}{2}\|x_t - x_{t-1}\|_2^2$. The greedy algorithm must have a competitive ratio at least $1 + \frac{4}{m}$.*

Theorem 2.4 also shows that the competitive ratio derived in Theorem 2.1 is tight. And next we examine the lower bound of OBD.

**Theorem 2.5.** *Consider hitting cost functions that are $m$-strongly convex with respect to $\ell_2$ norm and a movement costs given by $\frac{1}{2}\|x_t - x_{t-1}\|_2^2$. For any fixed $\gamma$, the competitive ratio of OBD is $\Omega(m^{-\frac{2}{3}})$ as $m \to 0^+$.*

The proof of Theorem 2.5 drew inspiration from the lower bound of projection-based algorithm in the CBC problem [3]. The OBD algorithm is similar to CBC in the sense that once the learner chooses a particular level set, it will take a decision point on this level set, which is convex. In the CBC example, the authors chose a center away from the starting point and a convex shape for the algorithm to project onto. Then for each round, the convex shape is rotated around the center so the decision points of a projection-based algorithm would trace a circle around the center. Meanwhile, an offline player would move to the center in the first round and incur no further movement cost.

While this example cannot be directly applied to OBD because the algorithm has some level of freedom in picking the convex shape in according to the cost functions, we can still construct a very similar scenario. We will instead present a sequence of cost functions whose minimizers are very close to each other. This way, we are allowed to replicate the behavior that the online learner fails to move close to the minimizers. Then an offline solution that follow the minimizers would pay a moderate movement cost upfront and incur lower hitting and movement costs for all later rounds.

As shown in Theorems 2.4 and 2.5, the greedy algorithm and OBD can achieve competitive ratio no better than $\Omega(m^{-1})$ and $\Omega(m^{-2/3})$ for small $m$, respectively. We have two takeaways from these results. First, although we were not able to show tighter competitive ratio bounds for OBD, this algorithm is still promising that it can potentially outperform the greedy algorithm. Second, if the general lower bound is in fact achievable, then there is gap between the then-state-of-art OBD and the general lower bound. And in the next section, we shall iterate on the ideas of OBD and close this gap.

## 2.4 Optimal Algorithm: Regularized Online Balanced Descent

We first observe that one weakness in OBD is its failure to follow the minimizer of the cost functions and thus accumulate a large hitting cost. Additionally, despite its simplicity, the greedy algorithm does very well in the setting of strongly convex hitting cost and squared $\ell_2$ norm movement cost. So a natural extension to OBD is to add a greedy component. The easiest way is to pick a point between OBD's decision point and the minimizer of the current cost function. This formulation, which we call greedy OBD (G-OBD), is shown to perform better than the classical OBD algorithm [42]. However, G-OBD relies on many geometric properties specific to the Euclidean norm and requires repeated projections. So, it is difficult to implement and computationally inefficient in practice. However, this idea (see Figure 2.3) is still essential to the development of our optimal algorithm, which would be stated under a *local view* of OBD.

Here, we present a *local view* of OBD that is computationally simpler and leads to

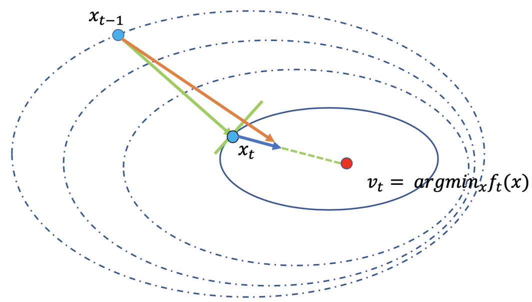


Figure 2.3: An (approximate) geometric interpretation of the R-OBD algorithm. In comparison to OBD (green arrow), R-OBD (orange) incorporates a greedy component (bleu arrow) to move close to the minimizer of the cost function $f_t$. Note that this diagram is an intuitive, not a faithful representation of R-OBD, as the actual algorithm formulated the ideas we described in a purely algebraic manner.

an algebraic counterpart to G-OBD. We consider the optimization problem

$$x_t = \arg\min_x f_t(x) + \frac{\lambda_1}{2} \|x - x_{t-1}\|^2 .$$

While it is not immediately clear on how this is related to OBD, we observe the first order optimality condition:

$$\nabla f_t(x_t) + \lambda_1 (x_t - x_{t-1}) = 0.$$

Therefore, this optimization problem is equivalent to a projection to some level set of $f_t$, where the particular level set is controlled by the parameter $\lambda_1$.

To incorporate a greedy component to the local view of OBD, we add a *regularization* term to the optimization problem. Let $v_t = \arg\min_x f_t(x)$ be the minimizer of the current cost function $f_t$, then we additionally account for the distance to $v_t$ as part of the objective function:

$$x_t = \arg\min_x f_t(x) + \frac{\lambda_1}{2} \|x - x_{t-1}\|^2 + \frac{\lambda_2}{2} \|x - v_t\|^2 .$$

This describes the update step of the regularized OBD (R-OBD) (see Algorithm 3. And we shall show that its competitive ratio improves from that of OBD and the greedy algorithm. The proof of this result can be found in Appendix A.5.

**Theorem 2.6.** *Consider hitting cost functions that are $m$-strongly convex with respect to $\ell_2$ norm and movement costs given by $\frac{1}{2} \|x_t - x_{t-1}\|_2^2$. There exists a choice $\lambda_1, \lambda_2$ such that the competitive ratio of Regularized OBD matches the lower bound proved in Theorem 2.3, i.e. the competitive ratio is at most $\frac{1}{2}\left(1 + \sqrt{1 + \frac{4}{m}}\right)$.*

**Algorithm 3** Regularized OBD (R-OBD)

```
procedure R-OBD(f_t, x_{t-1})                          ▷ Procedure to select x_t
    v_t ← arg min_x f_t(x)
    x_t ← arg min_x f_t(x) + λ_1 c(x, x_{t-1}) + λ_2 c(x, v_t)
    return x_t
```

Not only does the local view leads to a computationally simpler algorithm, but we proved that R-OBD matches the constant factors in Theorem 2.3 precisely, not just asymptotically. Further, this algorithm can macth the lower bound not just in the setting where movement costs are the squared $\ell_2$ norm, but also in the case where movement costs are Bregman divergences [42].

*Chapter 3*

# APPLICATION TO PREDICTIVE CONTROL

## 3.1 Model and Preliminaries

We consider a finite-horizon discrete-time online control problem with linear time-varying (LTV) dynamics, time-varying costs, and disturbances, namely

$$\begin{aligned} \min_{x_{0:T}, u_{0:T-1}} & \sum_{t=1}^{T} (f_t(x_t) + c_t(u_{t-1})) \\ \text{s.t. } & x_t = A_{t-1}x_{t-1} + B_{t-1}u_{t-1} + w_{t-1}, t = 1, \ldots, T, \\ & x_0 = x(0), \end{aligned} \tag{3.1}$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, and $w_t \in \mathbb{R}^n$ respectively denote the state, the control action, and the disturbance of the system at time steps $t = 1, \ldots, T$, and $x(0) \in \mathbb{R}^n$ is a given initial state. Define the tuple $\vartheta_t := (A_t, B_t, w_t, f_{t+1}, c_{t+1})$.

We assume that the algorithm has access to the exact predictions of disturbances, cost functions and dynamical matrices in the future $k$ time steps (which are time-varying); i.e., the sequence of actions and observations is

$$x_0, \vartheta_0, \vartheta_1, \ldots, \vartheta_{k-1}, u_0, \vartheta_k, u_1, \vartheta_{k+1}, \ldots, u_{T-k-1}, \vartheta_{T-1}, u_{T-k}, u_{T-k+1}, \ldots, u_{T-1}.$$

When there is no prediction ($k = 0$), the event sequence then becomes:

$$x_0, u_0, \vartheta_0, x_1, u_1, \vartheta_1, x_2, \ldots, x_{T-1}, u_{T-1}, \vartheta_{T-1}, x_T.$$

Often, experiments or observations on the dynamics can be conducted repeatedly and consistently, so we may assume all predictions are *exact*. This prediction model has been used in previous works like [25, 33, 43, 57], and is available in many real-world applications such as disturbance estimation in robotics and frequency regulation in power grids.

### Assumptions

As is standard in studies of regret and competitive ratio in linear control problems, we assume the cost functions are well-conditioned.

**Assumption 3.1** (Well-conditioned Costs)**.** *The cost functions satisfy the following constraints:*

1. *$f_t(\cdot)$ is $m_f$-strongly convex for $t = 1, \dots, T$, and $\ell_f$-strongly smooth for $t = 1, \dots, T-1$.*

2. *$c_t(\cdot)$ is both $m_c$-strongly convex and $\ell_c$-strongly smooth for $t = 1, \dots, T$.*

3. *$f_t(\cdot)$ and $c_t(\cdot)$ are twice continuously differentiable for $t = 1, \dots, T$.*

4. *$f_t(\cdot)$ and $c_t(\cdot)$ are non-negative, and $f_t(0) = c_t(0) = 0$ for $t = 1, \dots, T$.*

Note that assumptions (1) through (3) are quite common [6, 20, 24, 42, 57]. Assumption (4) is less common, but can be satisfied via re-parameterization without loss of generality. Specifically, when the minimizers of state cost $f_t$ and control cost $c_t$ are nonzero, we may perform the transformation

$$x'_t \leftarrow x_t - \arg\min_x f_t(x),\ u'_t \leftarrow u_t - \arg\min_u c_{t+1}(u),$$
$$w'_t \leftarrow w_t + A_t \arg\min_x f_t(x) + B_t \arg\min_u c_{t+1}(u).$$

Additionally, it is crucial that the dynamical system can be steered from an arbitrary initial state to an arbitrary final state via a finite sequence of permissible control actions. We call this the *controllability* of a dynamics. For linear time-invariant (LTI) systems, the full-rankness of the *controllability matrix* completely characterizes the reachability of the state space, which is generally used as a standard assumption for analysis [44, 58, 59]. This can be stated analogously for LTV systems as follows.

**Definition 3.2.** *For a dynamical system with linear time-varying dynamics $x_t = A_{t-1}x_{t-1} + B_{t-1}u_{t-1} + w_{t-1}, t = 1, \dots, T$, the transition matrix $\Phi(t_2, t_1) \in \mathbb{R}^{n\times n}$ (from time step $t_1$ to $t_2$) is defined as*

$$\Phi(t_2, t_1) := \begin{cases} A_{t_2-1}A_{t_2-2}\cdots A_{t_1} & \text{if } t_2 > t_1 \\ I & \text{if } t_2 \le t_1 \end{cases},$$

*and the controllability matrix $M(t, p) \in \mathbb{R}^{n\times(mp)}$ is defined as*

$$M(t, p) := \left[\Phi(t+p, t+1)B_t, \Phi(t+p, t+2)B_{t+1}, \dots, \Phi(t+p, t+p)B_{t+p}\right].$$

*The dynamical system is called controllable if there exists a constant $d \in \mathbb{Z}_+$, such that the controllability matrix $M(t, d)$ is of full row rank for any $t = 1, \dots, T-d$. The smallest such constant $d$ is called the controllability index of the system.*

Given the above definition, we can state the key assumption necessary for the analysis of LTV systems. We use a slightly stronger assumption than being merely controllable, which we refer to as $(d, \sigma)$-uniform controllability. It is a natural generalization of its counterpart for LTI systems (see Assumption 2 in [58], where $(d, \sigma)$ is instead named as $(\ell, \nu)$).

**Assumption 3.3.** *There exists positive constants $a$, $b$, and $b'$, such that*

$$\|A_t\| \leq a, \ \ \|B_t\| \leq b, \ \text{and } \|B_t^\dagger\| \leq b'$$

*hold for all time steps $t = 0, \ldots, T-1$, where $B_t^\dagger$ denotes the Moore–Penrose inverse of matrix $B_t$. Furthermore, there exists a positive constant $\sigma$ such that*

$$\sigma_{\min}\left(M(t, d)\right) \geq \sigma$$

*holds for all time steps $t = 0, \ldots, T-d$, where $d$ denotes the controllability index.*

Note that Assumption 3.3 implies $\sigma_{\min}(M(t, p)) \geq \sigma$ for all $p \geq d$ because appending more columns to a matrix with full row rank will not reduce its minimum singular value.

The LTV setting we consider is more general than the settings which existing results on regret and competitive ratio have assumed [15, 20, 43, 44]. We highlight the generality of this setting with the following example applications.

**Example 3.1** (Trajectory tracking in LTV systems with well-conditioned costs)**.** *Consider a trajectory tracking problem with LTV dynamics and well-conditioned costs, which generalizes the standard linear quadratic tracking problem in [43, 60] with LTI dynamics and quadratic costs. We adopt LTV dynamics $x_{t+1} = A_t x_t + B_t u_t + w_t$ and general well-conditioned cost functions $f_t(\cdot), c_t(\cdot)$ (see Assumption 3.1). With the desired trajectory $d_{1:T}$, we consider a new state $\tilde{x}_t := x_t - d_t$ and a new disturbance $\tilde{w}_t := w_t + A_t d_t - d_{t+1}$. Then the problem problem naturally fits into our setting with the new state and disturbance. Note that predictive control with LTV dynamics is practical in nonlinear systems [49, 61] because the nonlinearity could be well approximated by LTV models [61].*

**Example 3.2** (Power grid frequency regulation)**.** *Consider the frequency regulation problem in [62], where state $x = [\theta^\top, \omega^\top]^\top$ represent the status of a power plant, and power generation $p_{in} \in \mathbb{R}^n$ is the control action. The continuous-time dynamics* *is given by*

$$\underbrace{\begin{bmatrix}\dot{\theta}\\ \dot{\omega}\end{bmatrix}}_{\dot{x}(t)} = \underbrace{\begin{bmatrix}0 & I\\ -M(t)^{-1}L & -M(t)^{-1}D\end{bmatrix}}_{\hat{A}(t)} \underbrace{\begin{bmatrix}\theta\\ \omega\end{bmatrix}}_{x(t)} + \underbrace{\begin{bmatrix}0\\ M(t)^{-1}\end{bmatrix}}_{\hat{B}(t)} \underbrace{p_{in}}_{u(t)}.$$

*Here $M(t)$ denotes the rotational inertia matrix, which is time-varying and is determined by the proportion of renewable power in total power generation at time $t$; $L$ and $D$ are known system parameters. Using standard discretization techniques, we can formulate a discrete-time linear time-varying system $x_{t+1} = A_t x_t + B_t u_t + w_t$, where $A_t$ and $B_t$ are determined by $\hat{A}(t)$ and $\hat{B}(t)$. The cost functions are quadratic costs which penalizes frequency deviation [62]. Note that the controllers have accurate predictions of $A_t$ and $B_t$ in the near future because $M(t)$ can be accurately predicted [63, 64].*

**Predictive Control**

We study a classical predictive control (PC) algorithm inspired by model predictive control (MPC), in which an algorithm receives the dynamics and disturbances of the next $k$ time steps (where $k$ is called the *prediction window*), calculates the optimal solution given these predictions, and then applies the first control action of the optimal solution. We denote the PC algorithm with prediction window $k$ as $PC_k$.

Formally, At time step $t < T-k$, $PC_k$ solves the optimization problem $\tilde{\psi}_t^k(x_t, w_{t:t+k-1}; F)$. Since we need to consider horizon lengths other than $k$, for arbitrary $p \geq 1$ and time step $t$, we define the optimization problem $\tilde{\psi}_t^p(x, \zeta; F)$ as

$$\begin{aligned}\tilde{\psi}_t^p(x, \zeta; F) := \operatorname*{arg\,min}_{y_{0:p}, v_{0:p-1}} & \sum_{\tau=1}^{p} f_{t+\tau}(y_\tau) + \sum_{\tau=1}^{p} c_{t+\tau}(v_{\tau-1}) + F(y_k)\\ \text{s.t. } & y_\tau = A_{t+\tau-1} y_{\tau-1} + B_{t+\tau-1} v_{\tau-1} + \zeta_{\tau-1}, \tau = 1, \ldots, p, \\ & y_0 = x,\end{aligned} \tag{3.2}$$

where $x \in \mathbb{R}^n$ is the initial state, $\zeta \in (\mathbb{R}^n)^p$ (indexed by $0, \ldots, p-1$) is a sequence of disturbances, and $F : \mathbb{R}^n \to \mathbb{R}$ is a terminal cost function regularizing the final state. Here we additionally require that the terminal cost $F$ has the form $F(x) = \alpha(\|x\|)$, where $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a convex K-function (i.e. continuous increasing function with 0 at the origin) that is twice continuously differentiable. For each time step $\tau = 1, \ldots, k$, $y_\tau \in \mathbb{R}^n$ is the predictive state, and $v_\tau \in \mathbb{R}^m$ is the predictive control action. To make the algorithm well-defined, at time step $t = T - k$, $PC_k$ can finish

**Algorithm 4** Predictive Control ($PC_k$)

1: **for** $t = 0, 1, \ldots, T-k-1$ **do**
2: Observe current state $x_t$ and receive predictions $\vartheta_{t:t+k-1}$.
3: Solve and commit control actions $u_t := \tilde{\psi}_t^k(x_t, w_{t:t+k-1}; F)_{v_0}$.
4: At time step $t = T-k$, observe current state $x_t$ and receive predictions $\vartheta_{t:T-1}$.
5: Solve and commit control actions $u_{t:T-1} := \tilde{\psi}_t^k(x_t, w_{t:T-1}; 0)_{v_{0:k-1}}$.

the rest of the trajectory optimally by committing $u_{T-k:T-1} = \tilde{\psi}(x_{T-k}, w_{T-k:T-1}; 0)$. The pseudocode of predictive control is given in Algorithm 4.

It is also desirable to study the behavior of predictive control under some fixed terminal point. So, for prediction length $p \geq 1$ and time step $t$, we define an auxiliary optimization problem with a strict terminal constraint $y_p = z$ as follows:

$$\begin{aligned}\psi_t^p(x, \zeta, z) := \underset{y_{0:p}, v_{0:p-1}}{\arg\min} & \sum_{\tau=1}^{p} f_{t+\tau}(y_\tau) + \sum_{\tau=1}^{p} c_{t+\tau}(v_{\tau-1}) \\ \text{s.t. } & y_\tau = A_{t+\tau-1} y_{\tau-1} + B_{t+\tau-1} v_{\tau-1} + \zeta_{\tau-1}, \tau = 1, \ldots, p, \\ & y_0 = x, y_p = z,\end{aligned} \tag{3.3}$$

where the optimal value is denoted by $\iota_t^p(x, \zeta, z)$.

Throughout the paper, we use $\{(x_t, u_t)\}_{t=1}^T$ to denote the trajectory of predictive control, and use $\{(x_t^*, u_t^*)\}_{t=1}^T$ to denote the offline optimal trajectory (i.e., the optimal solution of (3.1)). In addition to the notions of strongly convex/strongly smooth functions, competitive ratio and regret we defined in the previous chapter, we will also use some standard notation in linear algebra. In particular, we use vector 2-norms and induced matrix 2-norms throughout this paper unless otherwise specified.

**Definition 3.4.** *We use the follow convention on linear algebra:*

1. $\|\cdot\|$ *denotes the (Euclidean)* 2*-norm for vectors and the induced* 2*-norm for matrices:*
$$\|v\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2},\ v \in \mathbb{R}^n$$
$$\|A\| = \sup_{v \in \mathbb{R}^n \setminus \{0\}} \frac{\|Ax\|}{\|x\|},\ A \in \mathbb{R}^{m \times n};$$
2. $\sigma(A)$ *is the collection of singular values of a matrix* $A$*, also known as the singular spectrum;*
3. $\sigma_{\min}(A)$ *denotes the smallest singular value of a matrix* $A$*;*
4. $A \succeq 0$ *indicates that a matrix* $A$ *is positive semi-definite.*

## 3.2 Relation to Online Convex Optimization

In this section we seek to study how much the solutions to (3.2) and (3.3) change with respect to some perturbations to the initial/terminal states and the disturbance sequence. This perturbation-based approach is related to the concept of *incremental stability* defined in [65], but not exactly the same because we consider the optimal trajectory in a finite horizon whereas the incremental stability focuses on asymptotic behavior over an infinite horizon. The key perturbation bound result we present in this section is Theorem 3.5, which states that if the target variable we are concerned with is the $h$-th predictive state/control input, while the perturbation occurs at the $\tau$-th time step, then the impact on the target variable is be exponentially small with respect to the time difference $|h - \tau|$.

Proving such a result directly is challenging because of the complexity of the LTV dynamical constraints in (3.2) and (3.3). Thus, we develop a novel reduction from LTV systems to fully-actuated systems, i.e., systems where the controller can steer the system to any state in the whole space $\mathbb{R}^n$ freely at every time step. This special case can be stated as a SOCO problem, which we discussed in the previous chapter. We exploit the controllability of the dynamics to analyze the LTV system in chunks of $d$ time steps. A sequence of $d$ time steps combined together can be thought as a fully-actuated system and thus we can formulate a SOCO problem, which is $(1/d)$-times as long as the original LTV system.

In the context if LTV system, the corresponding SOCO problem is an online game played by an agent against an adversary: at each time step $t$, the adversary reveals a hitting cost function $\hat{f}_t$, a switching cost function $\hat{c}_t$, and a disturbance (or exogenous input) $\hat{w}_t$. The agent picks a decision point $\hat{x}_t \in \mathbb{R}^n$, and incurs a hitting cost $\hat{f}_t(\hat{x}_t)$ and a switch cost $\hat{c}_t(\hat{x}_t, \hat{x}_{t-1}, \hat{w}_{t-1})$. The agent seeks to minimize the total cost it incurs throughout the game. The offline optimal cost is defined as the minimum cost if the agent has full knowledge of the costs and disturbances at the start of the game. Note that the switching cost also depends on the disturbances, since it would affect the actions of the controller.

More formally, observe that when the initial state $\hat{x}_0$, terminal state $\hat{x}_p$, and the disturbances $\hat{w}$ are given, the optimal $p$-step trajectory of SOCO can be obtained from the unconstrained optimization problem

$$\hat{\psi}(\hat{x}_0, \hat{w}, \hat{x}_p) := \arg\min_{\hat{x}_{1:p-1}} \sum_{\tau=1}^{p-1} \hat{f}_\tau(\hat{x}_\tau) + \sum_{\tau=1}^{p} \hat{c}_\tau(\hat{x}_\tau, \hat{x}_{\tau-1}, \hat{w}_{\tau-1}), \tag{3.4}$$

where the objective is a convex function of the decision variables $\hat{x}_{1:p-1}$.

We first present the following result due to [54], which bounds how much the perturbations of the system parameters (initial state, terminal state, and disturbances) impact the offline optimal solution.

**Theorem 3.3.** *Given a $p$-step SOCO problem with parameters $(\hat{x}_0, \hat{w}, \hat{x}_p)$. Assume $\hat{f}_\tau : \mathbb{R}^n \to \mathbb{R}$ is $\mu$-strongly convex, $\hat{c}_\tau : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}$ is convex and $\ell$-strongly smooth, and both are twice continuously differentiable for $\tau = 1, \ldots, p$, then*

$$\begin{aligned} &\left\|\hat{\psi}(\hat{x}_0, \hat{w}, \hat{x}_p)_h - \hat{\psi}(\hat{x}'_0, \hat{w}', \hat{x}'_p)_h\right\| \\ &\leq C_0 \left(\lambda_0^{h-1} \left\|\hat{x}_0 - \hat{x}'_0\right\| + \sum_{\tau=0}^{p-1} \lambda_0^{|h-\tau|-1} \left\|\hat{w}_\tau - \hat{w}'_\tau\right\| + \lambda_0^{p-h-1} \left\|\hat{x}_p - \hat{x}'_p\right\|\right) \end{aligned}$$

*for all $1 \leq h \leq p-1$, where $C_0 = (2\ell)/\mu$ and $\lambda_0 = 1 - 2 \cdot \left(\sqrt{1 + (2\ell/\mu)} + 1\right)^{-1}$.*

As a remark, Theorem 3.3 does not require the hitting cost $\hat{f}_\tau$ to be strongly smooth, nor the switching cost $\hat{c}_\tau$ to be strongly convex. This makes the assumptions on the SOCO costs $\hat{f}_\tau, \hat{c}_\tau$ weaker than that of the LTV costs $f_\tau, c_\tau$ as defined in (3.1).

We now build upon Theorem 3.3 and derive an exponentially-decaying perturbation result for LTV systems by reducing it to SOCO. As we have previously discussed, LTV systems are more difficult than SOCO because the dynamics prevent the online agent from picking the next state $x_{t+1}$ freely at a given state $x_t$. We overcome this obstacle by redefining the decision points as illustrated in Figure 3.1. Specifically, given state $x_t$ at time step $t$ as the last decision point, we then ask the online agent to decide state $x_{t+d}$ at time step $(t+d)$ rather than $x_{t+1}$ at time step $(t+1)$. Since $d$ is the controllability index, $x_{t+d}$ can be picked freely from the whole space $\mathbb{R}^n$ regardless of $x_t$. We also utilize the *principle of optimality*, e.g. if $y_{0:k}, v_{0:k-1}$ is the optimal solution to $\psi_t^k(x, \xi, z)$, then $y_{i:j}, v_{i:j-1}$ is the optimal solution to $\psi_{t+i}^{j-i}(y_i, \xi_{i:j-1}, y_j)$ for any $0 \leq i < j \leq k$. Therefore, the trajectory between time $t$ and $(t+d)$ can be recovered by solving $\psi_t^d(x_t, w_{t:t+d-1}, x_{t+d})$. So we are able to formulate a valid SOCO problem on the sequence of time steps $t, t+d, t+2d, \ldots$.

Naturally, the hitting cost at time step $(t+d)$ remains the same, while the switching cost becomes $\xi_t^d(x_t, w_{t:t+d-1}, x_{t+d})$, where the function $\xi_t^p$ is defined as

$$\xi_t^p(x, \zeta, z) := \iota_t^p(x, \zeta, z) - f_{t+p}(z). \tag{3.5}$$

Unlike the switching costs in [24, 36, 41, 42] which are explicitly defined as the $\ell_2$-distance or squared $\ell_2$-distance, the switching cost $\xi_t^p$ here is defined implicitly as

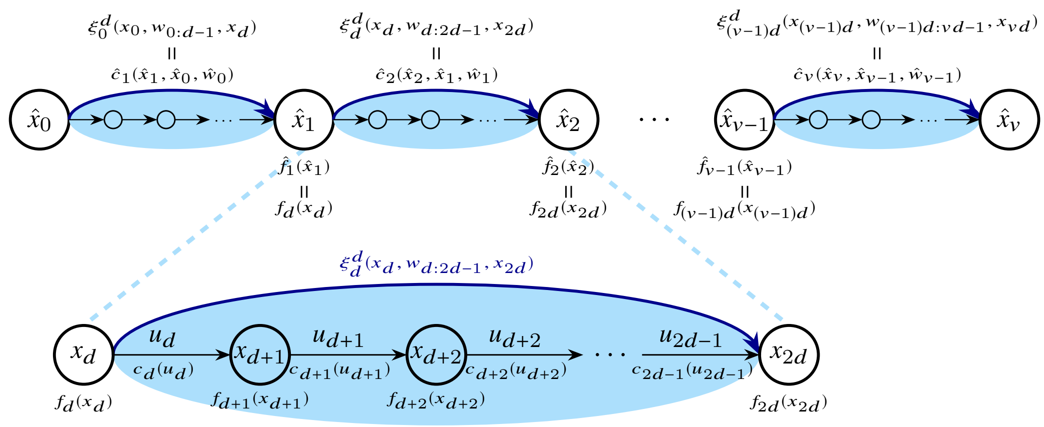


Figure 3.1: Illustration of the reduction from LTV to SOCO. Here we consider a simple example where $t = 0$ and $p = vd$. At time step 0, the agent cannot steer the system to an arbitrary target state at the next time step due to dynamical constraints. However, given $(d, \sigma)$-uniform controllability, the controller is able to enforce an arbitrary target state after $d$ time steps, which enables the transformation to a SOCO problem with a decision point in every $d$ time steps. This figure is taken from [54].

the optimal value of an optimization problem. Lemma 3.4 shows that the switching cost defined in (3.5) satisfies the requirements of Theorem 3.3, which allows us to obtain the desired perturbation bound in Theorem 3.5.

**Lemma 3.4.** *Under Assumption 3.1 and 3.3, for integer $p \geq d$, we have*

1. *$\psi_t^p(x, \zeta, z)$ is $L_1(p)$-Lipschitz in $(x, \zeta, z)$;*
2. *$\xi_t^p(x, \zeta, z)$ is convex and $L_2(p)$-strongly smooth in $(x, \zeta, z)$.*

*Here $L_1(p) = C(p)\left(1 + \ell \cdot C(p)/m_c\right), L_2(p) = \ell \cdot C(p)^2 + \ell^2 \cdot C(p)^4/m_c$, where $\ell = \max(\ell_f, \ell_c)$,*

$$C(p) = \begin{cases} O(a^{3p}) & \text{if } a > 1; \\ O(p^2) & \text{if } a = 1; \\ O(1) & \text{if } a < 1. \end{cases}$$

In Lemma 3.4, we use $O(\cdot)$ to hide quantities $a$, $b$, and $1/\sigma$; the precise expression of $C(p)$ and the proof of Lemma 3.4 can be found in [54]. Using the reduction from LTV to SOCO, we obtain a perturbation bound for the LTV systems (3.2) and (3.3) in Theorem 3.5, the proof of which can be found in Appendix B.1.

**Theorem 3.5.** *Consider the optimization problem defined in* (3.2) *and* (3.3) *and with a horizon length $p \geq d$. Under Assumptions 3.1 and 3.3, given any $(x, \zeta, z)$ and $(x', \zeta', z')$,*

$$\left\|\tilde{\psi}_t^p(x, \zeta; F)_{y_h} - \tilde{\psi}_t^p(x', \zeta'; F)_{y_h}\right\| \leq C\left(\lambda^h \|x - x'\| + \sum_{\tau=0}^{p-1} \lambda^{|h-\tau|} \left\|\zeta_\tau - \zeta'_\tau\right\|\right)$$
$$\left\|\psi_t^p(x, \zeta, z)_{y_h} - \psi_t^p(x', \zeta', z')_{y_h}\right\| \leq C\left(\lambda^h \|x - x'\| + \sum_{\tau=0}^{p-1} \lambda^{|h-\tau|} \left\|\zeta_\tau - \zeta'_\tau\right\| + \lambda^{p-h} \|z - z'\|\right)$$

*hold for all time steps $t$. Here we define $L_0 = \max_{d \leq p \leq 2d-1} L_2(p)$, and the constants are given by*

$$\lambda = \left(1 - 2\left(\sqrt{1 + (2L_0/m_c)} + 1\right)^{-1}\right)^{\frac{1}{2d-1}}, C = \frac{2L_0}{m_c} \cdot \left(1 - 2\left(\sqrt{1 + (2L_0/m_c)} + 1\right)^{-1}\right)^{-1}.$$

Theorem 3.5 allows us to bound the distance between any two trajectories so long as they can be expressed as the optimal solutions of the optimization problems (3.2) or (3.3). For example, to bound the norm of each state in the predictive trajectory $\tilde{\psi}_t^p(x, \zeta; F)$, we only need to set $x' = 0, \zeta' = 0$ in the first inequality because an all-zero trajectory can be expressed as $\tilde{\psi}_t^p(0, 0; F)$.

**Corollary 3.6** (Stability of the Optimal Trajectory)**.** *For the predicted trajectory found by solving* (3.2) *with prediction window $p \geq d$, the norm of the $h$-th predictive state is bounded above by*

$$\left\|\tilde{\psi}_t^p(x, \zeta; F)_{y_h}\right\| \leq C\left(\lambda^h \|x\| + \sum_{\tau=0}^{p-1} \lambda^{|h-\tau|} \|\zeta_\tau\|\right) \leq C\lambda^h \|x\| + \frac{2C}{1-\lambda} \sup_\tau \|\zeta_\tau\|,$$

*where $C, \lambda$ are the same constants as in Theorem 3.5.*

Another implication of Theorem 3.5 is the smoothness of the optimal cost of a $p$-step trajectory between the initial state $x$ and the terminal state $z$. Intuitively, Corollary 3.7 implies that changing the initial/terminal state will not significantly affect the optimal cost of a $p$-step trajectory between them.

**Corollary 3.7.** *For any time step $t$ and integer $p$ that satisfies $p \geq d$, function $\iota_t^p(\cdot, \zeta, \cdot)$ satisfies that*

$$\iota_t^p(x, \zeta, z) \leq (1 + \eta)\iota_t^p(x', \zeta, z') + \frac{L_0 + \ell_f}{2}\left(1 + \frac{1}{\eta}\right)\left(\|x' - x\|^2 + \|z' - z\|^2\right),$$

*$\forall x, x', \zeta, z, z'$, where $L_0$ is the same constant as in Theorem 3.5.*

### 3.3 Performance Guarantees

We now demonstrate the power of the perturbation approach in Section 3.2 by obtaining bounds on regret and competitive ratio. The key intuition behind our analysis is the following: at time step $t$, if the predictive controller with prediction window $k$ is given the knowledge of $x_t^*$ and $x_{t+k}^*$, it can fully recover the offline optimal states and control inputs for the future $k$ time steps, $x_{t+1:t+k}^*$ and $u_{t:t+k-1}^*$, from $\psi_t^k(x_t^*, w_{t:t+k-1}, x_{t+k}^*)$. However, without the knowledge of the offline optimal states, the predictive controller solves $\psi_t^k(x_t, w_{t:t+k-1}, x_{t+k})$ instead, where $x_{t+k}$ is implicitly determined by the $k$-th predictive state of $\tilde{\psi}_t^k(x_t, w_{t:t+k-1}; F)$. We overcome this gap with our perturbation approach (specifically, Theorem 3.5 and Corollary 3.6), which allows us to bound the distance between the controller's trajectory and the offline optimal trajectory. Then we utilize Corollary 3.7 to convert bounds on the distance between the trajectories to bounds on the cost $PC_k$ incurs.

#### Dynamic Regret

We first bound the dynamic regret of predictive control. For this analysis, a key observation is that the offline optimal trajectory is given by $x^* = \tilde{\psi}_0^T\left(x_0, w_{0:T-1}; 0\right)_{y_{1:T}}$. Furthermore, with the principle of optimality, the optimal trajectory starting at time step $t$ with state $x_t$ is equivalent to the trajectory of predictive control with prediction window $(T-t)$ and no terminal cost, i.e. $\tilde{\psi}_t^{T-t}\left(x_t, w_{t:T-1}; 0\right)_{y_{1:T-t}}$. To leverage this idea, we first introduce Lemma 3.8, which bounds the change in decision points against the change in prediction window $k$ by using Theorem 3.5 and Corollary 3.6.

**Lemma 3.8.** *For any positive integers $p \geq h$ and time step $t < T - p$, we have*

$$\begin{aligned}
&\left\|\tilde{\psi}_t^p\left(x_t, w_{t:t+p-1}; F\right)_{y_h} - \tilde{\psi}_t^{p+1}\left(x_t, w_{t:t+p}; F\right)_{y_h}\right\| \\
&\leq 2C\lambda^{p-h}\left(C\lambda^p \|x_t\| + \frac{2C}{1-\lambda}\sup_{0\leq\tau\leq T-1}\|w_\tau\|\right).
\end{aligned}$$

By cumulatively summing up the bounded difference in Lemma 3.8 and applying Theorem 3.5, we can show that at time step $t$, the distance between the predictive controller's next state $x_{t+1}$ and $\tilde{\psi}_t^{T-t}\left(x_t, w_{t:T-1}; 0\right)_{y_1}$ is in the order of $O(\lambda^k)$, where $\lambda$ is the decay rate of perturbation impact defined in Theorem 3.5. From here, we can derive an $O(\lambda^k)$ upper bound on the distance between the algorithm's trajectory and the offline optimal trajectory. Furthermore, under Assumption 3.3, we can conclude that at state $x_t$ and time step $t$, the predictive controller picks a near-optimal control action $u_t$. Combining these observations, along with Corollary 3.7, leads to the regret bound in Theorem 3.9.

**Theorem 3.9** (Dynamic Regret)**.** *Suppose $\|w_t\| \le D$ for some constant $D$ at each time step $t$. Let $\lambda, C, L_0$ be the decay rate and constants defined in Theorem 3.5. If prediction window $k \ge d$ is sufficiently large, such that*

$$k \ge 1 + \log\left(\frac{1}{1-\delta} \cdot C\left(\frac{2C}{1-\lambda} + \lambda\right)\right) \Big/ \log\left(\frac{1}{\lambda}\right) \tag{3.6}$$

*for some constant $\delta \in (0,1)$, then the dynamic regret of $PC_k$ is upper bounded by*

$$cost(PC_k) - cost(OPT) = O\left(\left(D + \frac{\lambda^k(\|x_0\| + D)}{\delta}\right)^2 \lambda^k T + \lambda^k \|x_0\|^2\right),$$

*where the notation hides quantities $a, b', \ell_f, \ell_c, C, 1/(1-\lambda)$ and $L_0$.*

An implication of Theorem 3.9 is that to obtain $o(1)$ dynamic regret when the norm of disturbances are uniformly upper bounded, it suffices to use a prediction window of length $\Theta(\log T)$. This parallels the result shown in [43], although in a more general setting.

**Competitive Ratio**

We now focus on bounding the competitive ratio of predictive control. Here, we study a modification of the predictive control algorithm we have considered to this point. In particular, we introduce a replan window $h$, as defined in Algorithm 5 which we denote as $PC_{(k,h)}$. This style of algorithm has been considered previously in the SOCO literature, where it has been shown to obtain a constant competitive ratio in some settings where MPC does not [12].

Our analysis approach highlights why this modification is beneficial for competitive ratio. Specifically, we obtain the competitive ratio bound by applying a potential method built upon [66]. We define the potential function as the squared distance between the algorithm's trajectory and the offline optimal trajectory, i.e., $\phi_t(x_t, x_t^*) = \|x_t - x_t^*\|^2$, which is standard in the literature [6, 24, 42]. We study how this potential function changes over time. Intuitively, we need to upper bound the increment of this potential function by the offline optimal cost to obtain a competitive ratio result. To achieve this, the algorithm needs to "move closer" to the offline optimal trajectory rather than "moving further away" from it. Recall that Theorem 3.5 gives that

$$\left\|\psi_t^k(x_t, w_{t:t+k-1}; F)_{y_h} - \psi_t^k(x_t^*, w_{t:t+k-1}; F)_{y_h}\right\| \le C\lambda^h \left\|x_t - x_t^*\right\|. \tag{3.7}$$

When the algorithm commits the first predictive state ($h = 1$), the left hand side of (3.7) might be larger than $\|x_t - x_t^*\|$ when $C\lambda > 1$. Thus, the algorithm must

**Algorithm 5** Predictive Control with Replan Window $h$ ($PC_{(k,h)}$)

1: Suppose $T = n_0 h + m_0$, where integers $n_0 \geq 0, k - h + 1 \leq m_0 \leq k$.
2: **for** $t = 0, h, \ldots, n_0(h-1)$ **do**
3: Observe current state $x_t$ and receive predictions $\vartheta_{t:t+k-1}$.
4: Solve and commit control actions $u_{t:t+h-1} := \tilde{\psi}_t^k(x_t, w_{t:t+k-1}; F)_{v_{0:h-1}}$.
5: At time step $t = n_0 h$, observe current state $x_t$ and receive predictions $\vartheta_{t:T-1}$.
6: Solve and commit control actions $u_{t:T-1} := \tilde{\psi}_t^{m_0}(x_t, w_{t:T-1}; 0)_{v_{0:m_0-1}}$.

"wait" until the right hand side of (3.7) becomes smaller than $\left\|x_t - x_t^*\right\|$. This is accomplished in Algorithm 5 via the replan window $h$.

Our main result for this section is the following competitive ratio bound for $PC_{(k,h)}$.

**Theorem 3.10.** *Let $\lambda, C, L_0$ be the decay rate and constants defined in Theorem 3.5. In Algorithm 5, if the replan window $h$ satisfies $h \geq \max\{\log((1+\varepsilon)C)/\log(1/\lambda), d\}$ for some positive constant $\varepsilon$, and the prediction window $k$ satisfies $k \geq h + d$, then it has competitive ratio*

$$1 + O\left(\varepsilon^{-1}\left(\frac{L_0 + \ell_f}{m_f}\right)^{1/2} \cdot C\lambda^{k-1-h}\right),$$

*where the notation only hides a small numerical constant.*

Note that when the constant $\varepsilon$ and the replan window $h$ are fixed, the competitive ratio is on the order of $1 + O(\rho^k)$ as the length of prediction $k$ tends to infinity. One potential line of future work is to understand if the replan window is necessary. It may be possible to either strengthen the constants given in Theorem 3.5 or improve our proof approach so as to eliminate the requirement on $h$.

# BIBLIOGRAPHY


[1] Avrim Blum and Carl Burch. “On-line learning and the metrical task system problem”. In: *Machine Learning* 39.1 (2000), pp. 35–58.

[2] Allan Borodin, Nathan Linial, and Michael E Saks. “An optimal on-line algorithm for metrical task system”. In: *Journal of the ACM* 39.4 (1992), pp. 745–763.

[3] Joel Friedman and Nathan Linial. “On convex body chasing”. In: *Discrete & Computational Geometry* 9.3 (1993), pp. 293–321.

[4] Sanjeev Arora, Elad Hazan, and Satyen Kale. “The multiplicative weights update method: a meta-algorithm and applications”. In: *Theory of Computing* 8.1 (2012), pp. 121–164.

[5] Elad Hazan et al. “Introduction to online convex optimization”. In: *Foundations and Trends in Optimization* 2.3-4 (2016), pp. 157–325.

[6] Guanya Shi et al. “Online Optimization with Memory and Competitive Control”. In: *Advances in Neural Information Processing Systems*. 2020, pp. 20636–20647.

[7] Chenkai Yu et al. “Competitive Control with Delayed Imperfect Information”. In: *arXiv preprint arXiv:2010.11637* (2020).

[8] Minghong Lin et al. “Dynamic right-sizing for power-proportional data centers”. In: *IEEE/ACM Transactions on Networking (TON)* 21.5 (2013), pp. 1378–1391.

[9] Nikhil Bansal et al. “A 2-competitive algorithm for online convex optimization with switching costs”. In: *Proceedings of the Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2015.

[10] Antonios Antoniadis and Kevin Schewior. “A tight lower bound for online convex optimization with switching costs”. In: *Proceedings of the International Workshop on Approximation and Online Algorithms*. Springer. 2017, pp. 164–175.

[11] Niangjun Chen et al. “Online convex optimization using predictions”. In: *ACM SIGMETRICS Performance Evaluation Review* 43.1 (2015), pp. 191–204.

[12] Niangjun Chen et al. “Using predictions in online optimization: Looking forward with an eye on the past”. In: *ACM SIGMETRICS Performance Evaluation Review* 44.1 (2016), pp. 193–206.

[13] Sarah Dean et al. “Regret bounds for robust adaptive control of the linear quadratic regulator”. In: *Neural Information Processing Systems (NeurIPS)*. 2018.

[14] Naman Agarwal et al. “Online control with adversarial disturbances”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 111–119.

[15] Naman Agarwal, Elad Hazan, and Karan Singh. “Logarithmic regret for online control”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[16] Alon Cohen, Tomer Koren, and Yishay Mansour. “Learning Linear-Quadratic Regulators Efficiently with only $\sqrt{T}$ Regret”. In: *International Conference on Machine Learning (ICML)*. 2019.

[17] Elad Hazan, Sham M Kakade, and Karan Singh. “The Nonstochastic Control Problem”. In: *Conference on Algorithmic Learning Theory (ALT)*. 2020.

[18] Dylan J Foster and Max Simchowitz. “Logarithmic regret for adversarial online control”. In: *arXiv preprint arXiv:2003.00189* (2020).

[19] Max Simchowitz and Dylan Foster. “Naive exploration is optimal for online LQR”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 8937–8948.

[20] Yingying Li, Xin Chen, and Na Li. “Online Optimal Control with Linear Dynamics and Predictions: Algorithms and Regret Analysis.” In: *Advances in Neural Information Processing Systems* 32 (2019).

[21] Gautam Goel and Babak Hassibi. “The Power of Linear Controllers in LQR Control”. In: *Proceedings of Machine Learning Research vol TBD* 1 (2020), p. 13.

[22] Guanya Shi et al. “Beyond No-Regret: Competitive Control via Online Optimization with Memory”. In: *arXiv preprint arXiv:2002.05318* (2020).

[23] Gautam Goel and Adam Wierman. “An online algorithm for smoothed regression and LQR control”. In: *Proceedings of Machine Learning Research* 89 (2019), pp. 2504–2513.

[24] Gautam Goel and Adam Wierman. “An Online Algorithm for Smoothed Regression and LQR Control”. In: *Proceedings of the Machine Learning Research*. Vol. 89. 2019, pp. 2504–2513.

[25] Minghong Lin et al. “Online algorithms for geographical load balancing”. In: *Proceedings of the International Green Computing Conference (IGCC)*. 2012, pp. 1–10.

[26] Taehwan Kim et al. “A decision tree framework for spatiotemporal sequence prediction”. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2015, pp. 577–586.

[27] Vinay Joseph and Gustavo de Veciana. "Jointly optimizing multi-user rate adaptation for video transport over wireless systems: Mean-fairness-variability tradeoffs". In: *Proceedings of the IEEE INFOCOM*. 2012, pp. 567–575.

[28] S. Kim and G. B. Giannakis. "An Online Convex Optimization Approach to Real-Time Energy Pricing for Demand Response". In: *IEEE Transactions on Smart Grid* 8.6 (2017), pp. 2784–2793. ISSN: 1949-3053. DOI: 10.1109/TSG.2016.2539948.

[29] Gautam Goel, Niangjun Chen, and Adam Wierman. "Thinking fast and slow: Optimization decomposition across timescales". In: *Proceedings of the IEEE Conference on Decision and Control (CDC)*. 2017, pp. 1291–1298.

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

[31] Masoud Badiei, Na Li, and Adam Wierman. "Online convex optimization with ramp constraints". In: *IEEE Conference on Decision and Control (CDC)*. 2015, pp. 6730–6736.

[32] Sebastien Bubeck et al. "Competitively Chasing Convex Bodies". In: *Proceedings of the ACM SIGACT Symposium on Theory of Computing (STOC)*. 2019.

[33] Yiheng Lin, Gautam Goel, and Adam Wierman. "Online optimization with predictions and non-convex losses". In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4.1 (2020), pp. 1–32.

[34] Nikhil Bansal et al. "Nested convex bodies are chaseable". In: *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2018, pp. 1253–1260.

[35] CJ Argue et al. "A nearly-linear bound for chasing nested convex bodies". In: *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2019, pp. 117–122.

[36] CJ Argue, Anupam Gupta, and Guru Guruganesh. "Dimension-Free Bounds for Chasing Convex Functions". In: *Conference on Learning Theory*. PMLR. 2020, pp. 219–241.

[37] Yair Bartal et al. "A Polylog(N)-competitive Algorithm for Metrical Task Systems". In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 1997, pp. 711–719. ISBN: 0-89791-888-6.

[38] Mark S Manasse, Lyle A McGeoch, and Daniel D Sleator. "Competitive algorithms for server problems". In: *Journal of Algorithms* 11.2 (1990), pp. 208–230.

[39] Sebastien Bubeck et al. “k-server via multiscale entropic regularization”. In: *Proceedings of the ACM SIGACT Symposium on Theory of Computing (STOC)*. 2018, pp. 3–16.

[40] Niv Buchbinder et al. “k-servers with a smile: online algorithms via projections”. In: *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2019, pp. 98–116.

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

[42] Gautam Goel et al. “Beyond online balanced descent: An optimal algorithm for smoothed online optimization”. In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 1875–1885.

[43] Chenkai Yu et al. “The Power of Predictions in Online Control”. In: *Advances in Neural Information Processing Systems* 33 (2020).

[44] Runyu Zhang, Yingying Li, and Na Li. “On the Regret Analysis of Online LQR Control with Predictions”. In: *arXiv preprint arXiv:2102.01309* (2021).

[45] Carlos E Garcia, David M Prett, and Manfred Morari. “Model predictive control: Theory and practice—A survey”. In: *Automatica* 25.3 (1989), pp. 335–348.

[46] Ugo Rosolia and Francesco Borrelli. “Learning model predictive control for iterative tasks. a data-driven control framework”. In: *IEEE Transactions on Automatic Control* 63.7 (2017), pp. 1883–1896.

[47] Milan Korda and Igor Mezić. “Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control”. In: *Automatica* 93 (2018), pp. 149–160.

[48] Frank Allgöwer and Alex Zheng. *Nonlinear model predictive control*. Vol. 26. Birkhäuser, 2012.

[49] Paolo Falcone et al. “A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems”. In: *2007 46th IEEE Conference on Decision and Control*. IEEE. 2007, pp. 2980–2985.

[50] Moritz Diehl, Rishi Amrit, and James B Rawlings. “A Lyapunov function for economic optimizing model predictive control”. In: *IEEE Transactions on Automatic Control* 56.3 (2010), pp. 703–707.

[51] David Angeli, Rishi Amrit, and James B Rawlings. “On average performance and stability of economic model predictive control”. In: *IEEE transactions on automatic control* 57.7 (2011), pp. 1615–1626.

[52] David Angeli, Alessandro Casavola, and Francesco Tedesco. “Theoretical advances on economic model predictive control with time-varying costs”. In: *Annual Reviews in Control* 41 (2016), pp. 218–224.

[53] Lars Grüne and Simon Pirkelmann. “Economic model predictive control for time-varying system: Performance and stability results”. In: *Optimal Control Applications and Methods* 41.1 (2020), pp. 42–64.

[54] Yiheng Lin et al. “Perturbation-based Regret Analysis of Predictive Control in Linear Time Varying Systems”. In: *arXiv preprint arXiv:2106.10497* (2021).

[55] Karl Johan Aström and Richard M Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.

[56] Neal Parikh and Stephen Boyd. “Proximal Algorithms”. In: *Foundations and Trends in Optimization* 1.3 (2014), pp. 127–239.

[57] Yingying Li, Guannan Qu, and Na Li. “Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit”. In: *IEEE Transactions on Automatic Control* (2020).

[58] Horia Mania, Stephen Tu, and Benjamin Recht. “Certainty Equivalence is Efficient for Linear Quadratic Control”. In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019.

[59] Karl Johan Åström and Richard M Murray. *Feedback systems*. Princeton university press, 2010.

[60] Brian DO Anderson and John B Moore. *Optimal Control: Linear Quadratic Methods*. Courier Corporation, Jan. 2007.

[61] Paolo Falcone et al. “Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation”. In: *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* 18.8 (2008), pp. 862–875.

[62] Patricia Hidalgo-Gonzalez et al. “Frequency Regulation using Data-Driven Controllers in Power Grids with Variable Inertia due to Renewable Energy”. In: *2019 IEEE Power & Energy Society General Meeting (PESGM)*. 2019, pp. 1–5.

[63] Madasthu Santhosh, Chintham Venkaiah, and D. M. Vinod Kumar. “Current advances and approaches in wind speed and wind power forecasting for improved renewable energy integration: A review”. In: *Engineering Reports* 2.6 (2020), e12178.

[64] Ruby Nageem and Jayabarathi R. “Predicting the Power Output of a Grid-Connected Solar Panel Using Multi-Input Support Vector Regression”. In: *Procedia Computer Science* 115 (2017), pp. 723–730.

[65] Duc N Tran, Björn S Rüffer, and Christopher M Kellett. “Incremental stability properties for discrete-time systems”. In: *Proceedings of Conference on Decision and Control (CDC)*. IEEE. 2016, pp. 477–482.

[66] Nikhil Bansal and Anupam Gupta. “Potential-function proofs for gradient methods”. In: *Theory of Computing* 15.1 (2019), pp. 1–32.

*Appendix A*

# PROOFS OF THE RESULTS FROM CHAPTER 2

Throughout the proofs we use the following notation to denote the hitting and movement costs of the online learner: $H_t := f_t(x_t)$ and $M_t := c(x_t, x_{t-1})$, where $x_t$ is the point chosen by the online algorithm at time $t$. Similarly, we denote the hitting and movement costs of the offline optimal (adversary) as $H_t^* := f_t(x_t^*)$ and $M_t^* := c(x_t^*, x_{t-1}^*)$, where $x_t^*$ is the point chosen by the offline optimal at time $t$.

The key proof technique which we will apply is called the *potential method* [66]. The goal is to bound the cost incurred by the online algorithm in each round by some constant times the per-round cost of the offline optimal. However, this is generally not possible, so we define a potential function $\phi_t$, which depends on $x_t and x_t^*$. Define $\Delta\phi_t = \phi_t - \phi_{t-1}$. Then we seek to show that

$$M_t + H_t + \Delta\phi_t \leq C(M_t^* + H_t^*)$$

for some constant $C$. Summing up over all rounds and we have the following inequality after cancellation:

$$\sum t = 1^T M_t + H_t \leq \sum t = 1^T M_t + H_t + \phi_T \leq C \cdot \left(\sum_{t=1}^{T} M_t^* + H_t^*\right),$$

which established a competitive ratio for the online algorithm.

Lastly, we note that we can assume without loss of generality that $f_t(v_t) = 0$ because shifting the cost function up by some constant can only decrease the competitive ratio.

## A.1 Proof of Theorem 2.1

This originally unpublished result was part of our work in [42].

$$\begin{aligned} H_t + M_t &= f_t(v_t) + \frac{1}{2}\|v_t - v_{t-1}\|^2 \\ &\leq \frac{1}{2}\left(\|x_t^* - v_t\| + \|x_t^* - x_{t-1}^*\| + \|x_{t-1}^* - v_{t-1}\|\right)^2 \\ &= \frac{1}{2}\Bigg(\|x_t^* - v_t\|^2 + \|x_t^* - x_{t-1}^*\|^2 + \|x_{t-1}^* - v_{t-1}\|^2 \end{aligned}$$

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

In the second step we used the triangle inequality and in fourth step we applied the AM-GM inequality (with a tuning parameter $\lambda > 0$). Adding up over all time steps, we have

$$ALG \leq \min_{\lambda > 0} \max\left(\frac{4+2\lambda^2}{m}, 1 + \frac{2}{\lambda^2}\right) OPT.$$

Since the first term is an increasing function of $\lambda$ and the second a decreasing function of $\lambda$, the optimal choice of $\lambda$ is when they are equal. The best choice is $\lambda^2 = \frac{m}{2}$, which yields competitive ratio of $1 + \frac{4}{m}$ as claimed.

### A.2 Proof of Theorem 2.3

This proof is adapted from [42].

We consider a sequence of hitting cost functions on the real line such that the algorithm stays at the starting point through time steps $t = 1, 2, \cdots, n$ and is forced to incur a huge movement cost at time step $t = n + 1$, whereas the offline adversary can pay relatively little cost by dividing the long trek between $x_0$ and $v_{n+1}$ into multiple small steps through time steps $t = 1, 2, \cdots, n + 1$.

Specifically, suppose the starting point of the algorithm and the offline adversary is $x_0 = x_0^* = 0$, and the hitting cost functions are

$$f_t(x) = \begin{cases} \frac{m}{2}x^2 & t \in \{1, 2, \cdots, n\} \\ \frac{m'}{2}(x-1)^2 & t = n + 1 \end{cases}$$

for some large parameter $m'$ that we choose later.

Suppose the algorithm first moves at time step $t_0$. If $t_0 < n + 1$, we stop the game at time step $t_0$ and compare the algorithm with an offline adversary which always stays at $x = 0$. The total cost of offline adversary is 0, but the total cost of the algorithm is non-zero. So, the competitive ratio is unbounded.

Next we consider the case where $t_0 \geq n+1$. This implies that $x_1, \ldots x_n = 0$ and $x_{n+1}$ is some non-zero point, say $x$. We see that the cost incurred by the online algorithm is

$$cost(ALG) \geq \min_{x_{n+1}}(M_{n+1} + H_{n+1}) = \min_x \left(\frac{1}{2}x^2 + \frac{m'}{2}(x-1)^2\right).$$

Notice that the right hand side tends to $\frac{1}{2}$ as $m'$ tends to infinity; specifically, we have

$$cost(ALG) \geq \min_x \left(\frac{1}{2}x^2 + \frac{m'}{2}(x-1)^2\right) = \frac{1}{2\left(1+\frac{1}{m'}\right)}. \tag{A.1}$$

Now let us consider the offline optimal. Notice that, in the limit as $m'$ tends to infinity, the offline optimal must satisfy $x_0^* = 0$ and $x_{n+1}^* = 1$; otherwise it would incur unbounded cost. Our lower bound is derived by considering the case when $m' \to \infty$ and so we constrain the adversary to satisfy the above, knowing that the adversary is not optimal for finite $m'$, i.e., $cost(ADV) \geq cost(OPT)$ with $cost(ADV) \to cost(OPT)$ as $m' \to \infty$.

Let the sequence of points the adversary chooses as $x^* = (x_0^*, x_1^*, \cdots, x_{n+1}^*) \in \mathbb{R}^{n+2}$. We compute the cost incurred by the adversary as follows where, to simplify presentation, we define $\mathcal{K}(n, y)$ to be the set $\{x \in \mathbb{R}^{n+2} \mid x_i \leq x_{i+1}, x_0 = 0, x_{n+1} = y\}$.

$$\begin{aligned} a_n &= 2 \min_{x^* \in \mathcal{K}(n,1)} \sum_{i=1}^{n+1}(H_i^* + M_i^*) \\ &= 2 \min_{x^* \in \mathcal{K}(n,1)} \left(\sum_{i=1}^{n} \frac{m}{2}(x_i^*)^2 + \sum_{i=1}^{n+1} \frac{1}{2}(x_i^* - x_{i-1}^*)^2\right). \end{aligned}$$

In words, $a_n$ is twice the minimal offline cost subject to the constraints $x_0^* = 0, x_{n+1}^* = 1$. We derive the limiting behavior of the offline costs as $n \to \infty$ in the following lemma.

**Lemma A.1.** *For $m > 0$, define*

$$a_n = 2 \min_{x^* \in \mathcal{K}(n,1)} \left(\sum_{i=1}^{n} \frac{m}{2}(x_i^*)^2 + \sum_{i=1}^{n+1} \frac{1}{2}(x_i^* - x_{i-1}^*)^2\right).$$

*Then we have* $\lim_{n\to\infty} a_n = \frac{-m+\sqrt{m^2+4m}}{2}$.

Given the lemma, the total cost of the offline adversary will be $\frac{a_n}{2}$. Finally, applying (A.1), we know $\forall n$ and $\forall m' > 0$,

$$\frac{cost(ALG)}{cost(ADV)} \geq \frac{\frac{1}{2(1+\frac{1}{m'})}}{\frac{a_n}{2}} = \frac{1}{(1+\frac{1}{m'})a_n}.$$

By taking the limit $n \to \infty$ and $m' \to \infty$ and using Lemma A.1, we obtain

$$\frac{cost(ALG)}{cost(OPT)} = \lim_{n,m' \to \infty} \frac{cost(ALG)}{cost(ADV)} \geq \left( \frac{-m + \sqrt{m^2 + 4m}}{2} \right)^{-1} = \frac{1 + \sqrt{1 + \frac{4}{m}}}{2}.$$

All that remains is to prove Lemma A.1, which describes the cost of the offline adversary in the limit as $n$ tends to infinity.

*Proof of Lemma A.1.* Using the fact that the costs are all homogeneous of degree 2, we see that for all $y \in [0,1]$, we have

$$\begin{aligned} &\min_{x^* \in \mathcal{K}(n,y)} \left( \sum_{i=1}^{n} \frac{m}{2} (x_i^*)^2 + \sum_{i=1}^{n+1} \frac{1}{2} (x_i^* - x_{i-1}^*)^2 \right) \\ = \;& y^2 \min_{x^* \in \mathcal{K}(n,1)} \left( \sum_{i=1}^{n} \frac{m}{2} (x_i^*)^2 + \sum_{i=1}^{n+1} \frac{1}{2} (x_i^* - x_{i-1}^*)^2 \right). \end{aligned} \tag{A.2}$$

The sequence $\{a_n\}, n \geq 0$ has a recursive relationship as follows:

$$\begin{aligned} a_{n+1} &= 2 \min_{x^* \in \mathcal{K}(n+1,1)} \left( \sum_{i=1}^{n+1} \frac{m}{2} (x_i^*)^2 + \sum_{i=1}^{n+2} \frac{1}{2} (x_i^* - x_{i-1}^*)^2 \right) \\ &= 2 \min_{0 \leq x \leq 1} \left( \min_{x^* \in \mathcal{K}(n,x)} \left( \sum_{i=1}^{n} \frac{m}{2} (x_i^*)^2 + \sum_{i=1}^{n+1} \frac{1}{2} (x_i^* - x_{i-1}^*)^2 \right) \right. \\ &\quad \left. + \frac{m}{2} x^2 + \frac{1}{2} (1-x)^2 \right) \\ &= 2 \min_{0 \leq x \leq 1} \left( x^2 \min_{x^* \in \mathcal{K}(n,1)} \left( \sum_{i=1}^{n} \frac{m}{2} (x_i^*)^2 + \sum_{i=1}^{n+1} \frac{1}{2} (x_i^* - x_{i-1}^*)^2 \right) \right. \\ &\quad \left. + \frac{m}{2} x^2 + \frac{1}{2} (1-x)^2 \right) \\ &= 2 \min_{0 \leq x \leq 1} \left( \frac{a_n}{2} x^2 + \frac{m}{2} x^2 + \frac{1}{2} (1-x)^2 \right) \\ &= \frac{a_n + m}{a_n + m + 1}. \end{aligned} \tag{A.3}$$

Solving the equation $x = \frac{x+m}{x+m+1}$, we find the two fixed points of the recursive relationship $a_{n+1} = \frac{a_n+m}{a_n+m+1}$ are

$$x_1 = \frac{-m + \sqrt{m^2 + 4m}}{2},$$

and

$$x_2 = \frac{-m - \sqrt{m^2 + 4m}}{2}.$$

Notice that for $i = 1, 2$, we have

$$m - (m+1)x_i = -(1 - x_i)x_i.$$

Using this property, we obtain

$$a_{n+1} - x_1 = \frac{a_n + m}{a_n + m + 1} - x_1 = \frac{(1 - x_1)a_n + m - (m+1)x_1}{a_n + m + 1} = \frac{(1 - x_1)(a_n - x_1)}{a_n + m + 1}, \tag{A.4}$$

and

$$a_{n+1} - x_2 = \frac{a_n + m}{a_n + m + 1} - x_2 = \frac{(1 - x_2)a_n + m - (m+1)x_2}{a_n + m + 1} = \frac{(1 - x_2)(a_n - x_2)}{a_n + m + 1}. \tag{A.5}$$

Notice that $a_{n+1} - x_2 > 0$. By dividing equations (A.4) and (A.5), we obtain

$$\left(\frac{a_{n+1} - x_1}{a_{n+1} - x_2}\right) = \frac{1 - x_1}{1 - x_2} \cdot \left(\frac{a_n - x_1}{a_n - x_2}\right), \forall n \geq 0.$$

Remember that $a_0 = 1$. Therefore we have

$$\left(\frac{a_n - x_1}{a_n - x_2}\right) = \left(\frac{1 - x_1}{1 - x_2}\right)^n \left(\frac{a_0 - x_1}{a_0 - x_2}\right) = \left(\frac{1 - x_1}{1 - x_2}\right)^{n+1}.$$

Rearranging this equation, we get

$$a_n = \left(1 - \left(\frac{1 - x_1}{1 - x_2}\right)^{n+1}\right)^{-1} \left(x_1 - x_2 \cdot \left(\frac{1 - x_1}{1 - x_2}\right)^{n+1}\right).$$

Since $0 < \left(\frac{1-x_1}{1-x_2}\right) < 1$, we have

$$\lim_{n \to \infty} a_n = x_1 = \frac{-m + \sqrt{m^2 + 4m}}{2}. \tag{A.6}$$

□

### A.3 Proof of Theorem 2.4

This originally unpublished result was part of our work in [42].

We will give an example in one dimension. We consider the cost functions defined as $f_t(x) = \frac{m}{2}(x - v_t)^2$ for minimizer defined as:

$$\begin{cases} v_t = -\frac{1}{2} - \frac{2}{m} & t \text{ is even} \\ v_t = \frac{1}{2} + \frac{2}{m} & t \text{ is odd} \end{cases}$$

Then the distance between the greedy algorithm's successive choices is $1 + 4/m$. So, the greedy algorithm incurs a cost of $\frac{1}{2}(1 + 4/m)^2$ at each time step.

If we instead choose $x_t = -1/2$ for even $t$ and $x_t = 1/2$ for odd $t$, then the distance between successive points is 1. And we incur a hitting cost of $\frac{m}{2}(\frac{2}{m})^2 = 2/m$ at each time step. Hence, the total cost we incur at each time step is $\frac{1}{2}(1 + 4/m)$.

Therefore the competitive ratio is at least:

$$\frac{\frac{1}{2}(1 + 4/m)^2}{\frac{1}{2}(1 + 4/m)} = 1 + \frac{4}{m}$$

### A.4 Proof of Theorem 2.5

This proof is adapted from [42].

Our proof of Theorem 2.5 relies on case of the balancing parameter $\gamma$: when $\gamma \in o(1/m)$ and $\gamma \in \Omega(1/m)$.

**Case 1:** $\gamma \in o(1/m)$.

We will rely on Lemmas A.2 and A.4.

**Lemma A.2.** *If $\gamma = o(1/m)$, the competitive ratio of OBD is $\Omega(1/(\gamma m))$ when $m \to 0^+$.*

*Proof.* Our approach is to construct a scenario where OBD is forced to move along the circumference of a large circle, but the offline adversary moves along the circumference of a much smaller circle (see Figure 2.2). The adversary is hence able to pay much smaller movements costs, forcing the competitive ratio to be large.

We propose a series of costs which force OBD to move in a circle. The idea is to construct a cost function so that, at the end of every round, the relative positions of the OBD algorithm, the offline adversary, and the minimizer are fixed. Since OBD is memoryless, we can simply input this function arbitrarily many times and the positions of OBD and the offline adversary will trace out a pair of concentric circles (see Figure 2.2).

Suppose that, at the start of a round, OBD is at the point $A$. Let $\ell$ be the distance between OBD and the adversary. Consider a right triangle $ABC$ such that $|AB| = h = \sqrt{\gamma m}\ell$, the offline adversary is at some point $D$ on the hypotenuse $AC$ and $|AD| = |BC| = \ell$ (see Figure A.1). Let us introduce a coordinate system such that

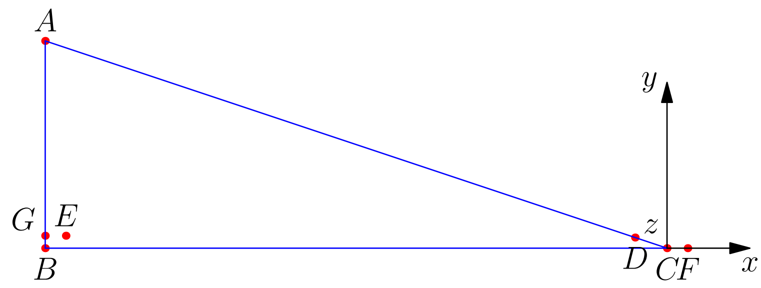


Figure A.1: *In the right triangle $\triangle ABC$, $\angle ABC = 90^o$, $|BC| = \ell$, $|AB| = h = \sqrt{\gamma m}\ell$. Point $D$ is on the line segment $AC$ such that $|AD| = \ell$. OBD starts at point $A$ and selects point $E$. The offline adversary starts at point $D$ and selects point $F$. $G$ is the projection point of $E$ on line segment $AB$.*

the origin lies at $C$, the $x$-axis contains $BC$ and the $y$-axis is parallel to $AB$, such that the positive part of the axis lies on the same side of $BC$ as the segment $AC$. Our goal is to construct a cost function which forces OBD towards $B$. This will preserve the relative positions of OBD and the adversary, since we assumed that they were a distance $\ell$ away at the start of the round. Consider the costs $g(u) = \frac{m}{2}\|u - C\|^2$, $h(u) = \alpha \cdot d(u, BC)$ where $d(u, BC)$ is the distance from the point $u$ to the line passing through $B$ and $C$ and $\alpha > 0$ is a parameter we will pick later. Define $f_t(u) = h(u) + g(u)$. Notice that $f_t$ is $m$-strongly convex because it is the sum of an $m$-strongly convex function and a convex function. Intuitively, when $\alpha$ is large, the function $f_t$ is infinity outside of the line $BC$ but is equal to $g(u) = \frac{m}{2}\|u - C\|^2$ when restricted to points $u$ on the line. After observing the cost $f_t$, OBD will pick some new point $E$.

The following lemma highlights that $E$ can be driven arbitrarily close to $B$ by taking $\alpha$ to be sufficiently large.

**Lemma A.3.** *Let $\varepsilon > 0$, and suppose $\alpha$ is picked to that $\alpha > \frac{hm\ell^2}{\varepsilon^2}$. Then the point $E$ picked by OBD satisfies $|EB| < \epsilon$.*

We instruct the adversary to pick the point $F$ on the line $BC$ (the $x$-axis) such that $EF = \ell$ (see Figure A.1). Notice that $|CF| = |BF| - |BC| \leq |BE| + |EF| - |BC| = |EB| + \ell - \ell < \varepsilon$, where we used the triangle inequality. Let $z = |DC|$. We see that the total cost incurred by the offline adversary is

$$M_t^* + H_t^* = \frac{1}{2}|DF|^2 + \frac{m}{2}|CF|^2 \leq \frac{1}{2}(|DC| + |CF|)^2 + \frac{m}{2}|CF|^2 \leq \frac{1}{2}(z+\varepsilon)^2 + \frac{m\varepsilon^2}{2},$$

where we applied the triangle inequality.

Notice that $h = |AB| = \sqrt{|AC|^2 - |BC|^2}$ by the Pythagorean theorem (recall that $ABC$ is a right triangle). Since $|AC| = \ell + z$ and $|BC| = \ell$, we see that $h = \sqrt{2z\ell + z^2}$. Hence the movement cost incurred by the OBD is

$$M_t \geq \frac{1}{2}(h - \varepsilon)^2 = \frac{1}{2}(\sqrt{2z\ell + z^2} - \varepsilon)^2.$$

Hence the ratio of the costs is

$$\frac{M_t + H_t}{M_t^* + H_t^*} \geq \frac{M_t}{M_t^* + H_t^*} \geq \frac{\frac{1}{2}(\sqrt{2z\ell + z^2} - \varepsilon)^2}{\frac{1}{2}(z + \varepsilon)^2 + \frac{m\varepsilon^2}{2}}.$$

Since the limit of this expression as $\varepsilon \to 0$ is $\frac{2z\ell + z^2}{z^2}$, for sufficiently small $\varepsilon$ this will be at least $\frac{1}{2}\frac{2z\ell+z^2}{z^2} \geq \frac{\ell}{z}$. Since $z = \sqrt{h^2 + \ell^2} - \ell$ and $h = \sqrt{\gamma m}\ell$, the ratio of costs is at least

$$\frac{\ell}{\sqrt{\gamma m \ell^2 + \ell^2} - \ell} = \frac{1}{\sqrt{\gamma m + 1} - 1} = \frac{\sqrt{\gamma m + 1} + 1}{\gamma m} \geq \frac{2}{\gamma m}.$$

Now, we describe the whole process. When $t = 1$, the hitting cost function is $f_1(x) = \frac{m}{2}\|x\|_2^2$. While OBD stays at $x = 0$, the adversary moves to the point $(\ell, 0)$; it incurs a one-time cost of $M_1^* + H_1^* = \frac{1}{2}\ell^2 + \frac{m}{2}\ell^2$. On all subsequent steps $t = 2 \ldots T$, we repeatedly apply the construction, which forces OBD to move in a circle. The one-time cost incurred by the adversary to setup the game is negligible in the limit as $T$ is large, and the per-round ratio of costs is $\Omega(\frac{1}{\gamma m})$, so the competitive ratio is also $\Omega(\frac{1}{\gamma m})$ as claimed. □

The key technical lemma used in the proof is Lemma A.3, and we now provide a proof of that result.

*Proof of Lemma A.3.* Suppose $\alpha > \frac{hm\ell^2}{\varepsilon^2}$. We first show that OBD selects the point $E$ strictly contained by the $\frac{m}{2}\ell^2$-level set, which is the one $B$ lies on. First observe that the point $B$ satisfies the balance condition: $\frac{1}{2}|AB|^2 = \gamma\frac{m}{2}|BC|^2$, because we constructed $ABC$ so that $|AB| = h = \sqrt{\gamma m}\ell$ and $|BC| = \ell$. However, the point $B$ is not necessarily a projection of $A$ onto any level set of $f_t$. If OBD projected onto the level set which $B$ lies on, it would incur less cost than if it moved to $B$; however then the balance condition would be violated. To restore the balance condition, we must increase the movement cost while decreasing the hitting cost – which means we must move to a strictly smaller level set, say the $\frac{m}{2}l_1^2$-level set, where $l_1 < l$.


Figure A.2: *Balance condition at time step $t$ in Lemma A.4. Starting from $x_{t-1}$, OBD picks $x_t$ after observing the hitting cost function $f_t(x) = \frac{m}{2}(x - t)^2$.*

Let $E_y$ denote the $y$-coordinate of $E$, using the coordinate system we define in the proof of Lemma A.2. Notice that $E_y = \frac{g(E)}{\alpha}$, since $g(E)$ was defined to be the vertical distance to the $x$-axis times $\alpha$. Since $g(E) \le f_t(E)$, we see that $E_y \le \frac{f_t(E)}{\alpha} = \frac{ml_1^2}{2\alpha} \le \frac{ml^2}{2\alpha}$, where we used the fact that $E$ lies on the $\frac{m}{2}\ell_1^2$ level set and $\ell_1 \le \ell$. By the balance condition, $\frac{1}{2}|AE|^2 = \frac{\gamma m}{2} l_1^2 \le \frac{\gamma m}{2} l^2 = \frac{1}{2}h^2$. Let $G$ be the point with coordinates $(B_x, E_y)$. Applying the Pythagorean theorem successively to the right triangle $BEG$ and the right triangle $AEG$, we see that

$$\begin{aligned} |EB|^2 = |E_x - B_x|^2 + E_y^2 &\le (|AE|^2 - (|AB| - E_y)^2) + E_y^2 \\ &\le (|AB|^2 - (|AB| - E_y)^2) + E_y^2 \le 2h \cdot E_y \le h\frac{ml^2}{\alpha}, \end{aligned} \tag{A.7}$$

where we used the fact that $|AB| \ge |AE|$ and $|AB| = h$. Since we picked $\alpha > \frac{hm\ell^2}{\varepsilon^2}$, we see that $|EB| < \varepsilon$.

□

Now we move on to the next technical lemma in the proof of Theorem 2.5.

**Lemma A.4.** *When $\gamma = o(\frac{1}{m})$, the competitive ratio of OBD is $\Omega(\sqrt{\frac{\gamma}{m}})$.*

*Proof.* We consider a sequence of cost functions on the real line such that the OBD algorithm moves far away from the starting point, incurring significant movement costs, whereas the offline adversary could pay relatively little cost by staying at the starting point. More specifically, we consider the sequence of hitting cost functions $f_t(x) = \frac{m}{2}(x - t)^2, t = 1, 2, \cdots, n$. The value of $n$ will be picked later. We assume the starting point is at zero.

Notice that by the balance condition we always have $M_t = \gamma H_t$, so $\frac{1}{2}\|x_t - x_{t-1}\|^2 = \gamma\frac{m}{2}\|x_t - t\|^2$. We can rearrange this expression to obtain $\frac{x_t - x_{t-1}}{t - x_t} = \sqrt{\gamma m}$. Define

$$\lambda = \frac{x_t - x_{t-1}}{t - x_{t-1}} = \frac{\sqrt{\gamma m}}{1 + \sqrt{\gamma m}}.$$

We obtain the recursive equation $x_t = x_{t-1} + (t - x_{t-1})\lambda$ with initial condition $x_0 = 0$. Solving this equation, we obtain $x_t = t - \frac{1-\lambda}{\lambda}(1 - (1 - \lambda)^t)$.

Suppose we picked $n$ to be $= \lceil \frac{1}{\lambda} \rceil$. By assumption, $\gamma = o(\frac{1}{m})$; hence in the limit as $m$ tends to zero, $\lambda$ also tends to zero. Notice that $x_n = n - \frac{1-\lambda}{\lambda}(1-(1-\lambda)^n) \geq \frac{1}{\lambda}\frac{1}{2e} - (1-\frac{1}{e}) \geq \frac{1}{6\lambda}$ for sufficiently small $\lambda$. Here we used the fact that $(1-\lambda)^{\frac{1}{\lambda}} \to e^{-1}$.

Suppose the next cost function is $f_{n+1}(x) = m'x^2$. Notice that if the offline adversary simply stays at zero throughout the game, the total cost it incurs would be

$$cost(ADV) = \frac{m}{2}(1^2 + 2^2 + \cdots + n^2) \leq \frac{mn^3}{2} = \Theta\left(\frac{m}{\lambda^3}\right) = \Theta\left(\frac{1}{\sqrt{\gamma^3 m}}\right).$$

In the last step, we used the fact that $\lambda$ tends to $\sqrt{\gamma m}$ when $\gamma = o(\frac{1}{m})$ and $m$ tends to zero.

If we pick $m'$ large enough that OBD is forced to incur movement cost at least $\frac{1}{2}(\frac{x_n}{2})^2$, the total cost incurred by OBD is

$$cost(OBD) \geq \frac{1}{2}\left(\frac{x_n}{2}\right)^2 = \Theta\left(\frac{1}{\lambda^2}\right) = \Theta\left(\frac{1}{\gamma m}\right).$$

Putting these facts together, we see that the competitive ratio is at least $\Theta(\sqrt{\frac{\gamma}{m}})$. $\square$

By combining Lemma A.2 and Lemma A.4, we know the competitive ratio is at least $\max\left(\frac{C_1}{\gamma m}, C_2\sqrt{\frac{\gamma}{m}}\right)$ for some positive constants $C_1, C_2$. Notice that function $\frac{C_1}{\gamma m}$ is monotonically decreasing in $\gamma$ and $C_2\sqrt{\frac{\gamma}{m}}$ is monotonically increasing in $\gamma$. So the quality is maximized when $\frac{C_1}{\gamma m} = C_2\sqrt{\frac{\gamma}{m}}$, which yields get $\gamma = \left(\frac{C_1}{C_2}\right)^{\frac{2}{3}} m^{-\frac{1}{3}}$. Therefore,

$$\max\left\{\frac{C_1}{\gamma m}, C_2\sqrt{\frac{\gamma}{m}}\right\} \geq C_1^{\frac{1}{3}} C_2^{\frac{2}{3}} m^{-\frac{2}{3}} = \Theta(m^{-\frac{2}{3}}).$$

**Case 2:** $\gamma \in \Omega(1/m)$.

We shall show that in the case, the competitive ratio of OBD is $\Omega\left(\frac{1}{m}\right)$.

Since $\gamma = \Omega(\frac{1}{m})$, we can assume there exists $C > 0$ such that $\gamma \geq C/m$. We again consider a situation such that the OBD algorithm moves far away from the starting point, incurring significant movement cost, whereas the offline adversary could pay relatively little cost by staying at the starting point. More specifically, suppose the starting point is zero and the first cost function is $f_1(x) = \frac{m}{2}(1-x)^2$. Suppose the adversary stays at zero. The cost incurred by the adversary will be

$$cost(ADV) = \frac{m}{2}.$$

Notice that by the balance condition ($M_t = \gamma H_t$), the point $x_1$ picked by OBD satisfies $\frac{x_1^2}{2} = \gamma \frac{m}{2}(1 - x_1)^2$. So the cost incurred by OBD is lower bounded by

$$cost(OBD) \geq M_1 = \frac{1}{2}\left(\frac{\sqrt{\gamma m}}{1 + \sqrt{\gamma m}}\right)^2 \geq \frac{1}{2}\left(\frac{\sqrt{C}}{1 + \sqrt{C}}\right)^2.$$

Since $C$ is a positive constant, the competitive ratio of OBD is lower bounded by $\frac{OBD}{ADV} = \Theta\left(\frac{1}{m}\right)$.

By combining the two cases, we conclude that the competitive ratio of OBD is at least $\Theta(m^{-\frac{2}{3}})$ when $m \to 0^+$.

## A.5 Proof of Theorem 2.6

This proof is adapted from [42].

This result follows from the more general bound in Theorem A.5 below, which describes the competitive ratio of Algorithm 3 as a function of $\lambda_1, \lambda_2$.

**Theorem A.5.** *Consider hitting cost functions that are $m$-strongly convex with respect to $\ell_2$ norm and movement costs given by $\frac{1}{2}\|x_t - x_{t-1}\|_2^2$. Regularized-OBD (Algorithm 3 with $h(x) = \frac{1}{2}\|x\|_2^2$) with parameters $1 \geq \lambda_1 > 0, \lambda_2 \geq 0$ has competitive ratio at most*

$$\max\left(\frac{m + \lambda_2}{\lambda_1} \cdot \frac{1}{m}, 1 + \frac{\lambda_1}{\lambda_2 + m}\right).$$

Notice that Theorem 2.6 follows immediately by setting $\frac{m+\lambda_2}{\lambda_1} = \frac{m}{2}\left(1 + \sqrt{1 + \frac{4}{m}}\right)$ in Theorem A.5.

Before proving Theorem A.5, we first prove a teechnical lemma which gives a lower bound of the value of hitting cost as a function of the distance to the minimizer.

**Lemma A.6.** *If $f : \mathcal{X} \to \mathbb{R}$ is a $m$-strongly convex function with respect to some norm $\|\cdot\|$, and $v$ is the minimizer of $f$ (i.e. $v = \arg\min_{x \in \mathcal{X}} f(x)$), then we have $\forall x \in \mathcal{X}$,*

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

*Proof.* By the definition of $m$-strongly convex, we obtain that $\forall \alpha \in (0, 1)$,

$$f(\alpha x + (1 - \alpha)v) \leq \alpha f(x) + (1 - \alpha) f(v) - \frac{m}{2}\alpha(1 - \alpha)\|x - v\|^2. \tag{A.8}$$

Notice that $f(v) \le f(\alpha x + (1-\alpha)v)$. Combining this with inequality (A.8), we obtain that $\forall \alpha \in (0,1)$,

$$f(v) \le \alpha f(x) + (1-\alpha) f(v) - \frac{m}{2}\alpha(1-\alpha)\|x-v\|^2 .$$

Rearranging the terms, we observe that $\forall \alpha \in (0,1)$,

$$f(x) \ge f(v) + \frac{m}{2}(1-\alpha)\|x-v\|^2 .$$

Therefore

$$f(x) \ge \lim_{\alpha \to 0^+} \left( f(v) + \frac{m}{2}(1-\alpha)\|x-v\|^2 \right) = f(v) + \frac{m}{2}\|x-v\|^2 .$$

□

Now we return to the proof of Theorem A.5.

*Proof of Theorem A.5.* In the proof, we use the property of strongly convex to derive an inequality in the form of $H_t + M_t + \Delta\phi \le C(H_t^* + M_t^*)$, where $\Delta\phi$ is the change in potential and $C$ is an upper bound for the competitive ratio.

Throughout the proof, we use $\|\cdot\|$ to denote $\ell_2$ norm.

Notice that when $h(x) = \frac{1}{2}\|x\|^2$, the update rule in Algorithm 3 is:

$$x_t = \arg\min_x f_t(x) + \frac{\lambda_1}{2}\|x - x_{t-1}\|^2 + \frac{\lambda_2}{2}\|x - v_t\|^2 .$$

For convenience, we define

$$F_t(x) = f_t(x) + \frac{\lambda_1}{2}\|x - x_{t-1}\|^2 + \frac{\lambda_2}{2}\|x - v_t\|^2 .$$

Since $f_t(x)$ is $m$-strongly convex, $\frac{\lambda_1}{2}\|x - x_{t-1}\|^2$ is $\lambda_1$-strongly convex, and $\frac{\lambda_2}{2}\|x - v_t\|^2$ is $\lambda_2$-strongly convex, $F_t(x)$ is $(m+\lambda_1+\lambda_2)-$strongly convex. Since $x_t = \arg\min_x F_t(x)$, by Lemma A.6, we obtain

$$F_t(x_t^*) \ge F_t(x_t) + \frac{m+\lambda_1+\lambda_2}{2}\left\|x_t^* - x_t\right\|^2 ,$$

which implies

$$\begin{aligned}
&H_t + \lambda_1 M_t + \frac{m+\lambda_1+\lambda_2}{2}\left\|x_t^* - x_t\right\|^2 \\
\le\ & H_t + \lambda_1 M_t + \frac{\lambda_2}{2}\|x - v_t\|^2 + \frac{m+\lambda_1+\lambda_2}{2}\left\|x_t^* - x_t\right\|^2 \\
\le\ & H_t^* + \frac{\lambda_1}{2}\left\|x_t^* - x_{t-1}\right\|^2 + \frac{\lambda_2}{2}\left\|x_t^* - v_t\right\|^2 .
\end{aligned} \tag{A.9}$$

We define the potential function as $\phi(x_t, x_t^*) = \frac{m+\lambda_1+\lambda_2}{2} \left\|x_t^* - x_t\right\|^2$ and $\Delta\phi = \phi(x_t, x_t^*) - \phi(x_{t-1}, x_{t-1}^*)$. We then can rewrite inequality (A.9) as

$$H_t + \lambda_1 M_t + \Delta\phi \leq \left(H_t^* + \frac{\lambda_2}{2}\left\|x_t^* - v_t\right\|^2\right) + \frac{\lambda_1}{2}\left\|x_t^* - x_{t-1}\right\|^2 - \frac{m+\lambda_1+\lambda_2}{2}\left\|x_{t-1}^* - x_{t-1}\right\|^2. \tag{A.10}$$

Additionally

$$\begin{aligned}
&\frac{\lambda_1}{2}\left\|x_t^* - x_{t-1}\right\|^2 - \frac{m+\lambda_1+\lambda_2}{2}\left\|x_{t-1}^* - x_{t-1}\right\|^2 \\
&\leq \frac{\lambda_1}{2}\left(\left\|x_t^* - x_{t-1}^*\right\| + \left\|x_{t-1}^* - x_{t-1}\right\|\right)^2 - \frac{m+\lambda_1+\lambda_2}{2}\left\|x_{t-1}^* - x_{t-1}\right\|^2 && \text{(A.11a)} \\
&= \frac{\lambda_1}{2}\left\|x_t^* - x_{t-1}^*\right\|^2 + \lambda_1\left\|x_t^* - x_{t-1}^*\right\| \cdot \left\|x_{t-1}^* - x_{t-1}\right\| - \frac{m+\lambda_2}{2}\left\|x_{t-1}^* - x_{t-1}\right\|^2 \\
&\leq \frac{\lambda_1}{2}\left\|x_t^* - x_{t-1}^*\right\|^2 + \frac{\lambda_1^2}{2(m+\lambda_2)}\left\|x_t^* - x_{t-1}^*\right\|^2 + \frac{m+\lambda_2}{2}\left\|x_{t-1}^* - x_{t-1}\right\|^2 \\
&\quad - \frac{m+\lambda_2}{2}\left\|x_{t-1}^* - x_{t-1}\right\|^2 && \text{(A.11b)} \\
&= \frac{\lambda_1(\lambda_1+\lambda_2+m)}{2(\lambda_2+m)}\left\|x_{t-1}^* - x_{t-1}^*\right\|^2 \\
&= \lambda_1\left(1 + \frac{\lambda_1}{\lambda_2+m}\right)M_t^*,
\end{aligned}$$

where we apply the triangle inequality in line (A.11a) and AM-GM in line (A.11b).

Combining inequalities (A.10) and (A.11), we obtain

$$H_t + \lambda_1 M_t + \Delta\phi \leq \left(H_t^* + \frac{\lambda_2}{2}\left\|x_t^* - v_t\right\|^2\right) + \lambda_1\left(1 + \frac{\lambda_1}{\lambda_2+m}\right)M_t^*. \tag{A.12}$$

And since $f_t(x)$ is $m$-strongly convex, we have

$$\frac{\lambda_2}{2}\left\|x_t^* - v_t\right\|^2 \leq \frac{\lambda_2}{m}H_t^*.$$

Substituting the above identity into inequality (A.12) yields

$$H_t + \lambda_1 M_t + \Delta\phi \leq \frac{m+\lambda_2}{m}H_t^* + \lambda_1\left(1 + \frac{\lambda_1}{m+\lambda_2}\right)M_t^*. \tag{A.13}$$

Using inequality (A.13), we obtain

$$H_t + M_t + \frac{1}{\lambda_1}\Delta\phi \leq \frac{H_t + \lambda_1 M_t + \Delta\phi}{\lambda_1} \leq \frac{m+\lambda_2}{\lambda_1 m}H_t^* + \left(1 + \frac{\lambda_1}{m+\lambda_2}\right)M_t^*.$$

Theorem A.5 follows from summing the above inequality over all timesteps $t$. $\square$

*Appendix B*

# PROOFS OF THE RESULTS FROM CHAPTER 3

## B.1 Proof of Theorem 3.5

This proof is adapted from [54].

The proof of Theorem 3.5 is based on the decision-point transformation introduced in Section 3.2.

Recall that $d$ denotes the controllability index, which has been defined in Definition 3.2. To show the perturbation bound of $\psi_t^p(\cdot,\cdot,\cdot)_{y_h}$, suppose $h$ and $p$ satisfy $qd \le h < (q+1)d$ and $p = sd + r$, where $q, s, r \in \mathbb{N}$ and $0 \le r < d$. Now we shall select the decision points as

$$y_0, y_d, \cdots, y_{(q-1)d}, y_h, y_{(q+2)d}, \cdots, y_{(s-1)d}, y_p,$$

which are also denoted by $y_{i_0}, \cdots, y_{i_{s-1}}$ for simplicity. Since the distance of any consecutive decision points falls in $[d, 2d)$, we can apply Lemma 3.4 to bound the strong smoothness of switching costs. In the transformed SOCO problem, the disturbance input of the $(\tau-1)$-th time period is a vector $\bar{w}_{\tau-1} = \zeta_{i_{\tau-1}:i_\tau-1} \in \mathbb{R}^{n\times(i_\tau - i_{\tau-1})}$. Each stage cost $\xi_t^{i_\tau - i_{\tau-1}}(x_{i_{\tau-1}}, \bar{w}_{\tau-1}, x_{i_\tau})$ is convex and $L_2(i_\tau - i_{\tau-1})$-strongly smooth by Lemma 3.4, and is thus $L_0$-strongly smooth by definition. Recall that the solution of the transformed SOCO problem is denoted by $\hat{\psi}(x_t, \zeta, x_{t+p})$. Then by Theorem 3.3 we have

$$\begin{aligned}
&\left\|\psi_t^p(x,\zeta,z)_{y_h} - \psi_t^p(x',\zeta',z')_{y_h}\right\| \\
=\ &\left\|\hat{\psi}(x,\zeta,z)_q - \hat{\psi}(x',\zeta',z')_q\right\| \\
\le\ &C_0\left(\lambda_0^{q-1}\left\|x-x'\right\| + \sum_{\tau=0}^{s-2}\lambda_0^{|q-\tau|-1}\left\|\bar{w}_\tau - w'_\tau\right\| + \lambda_0^{(s-1)-q-1}\left\|z-z'\right\|\right) \\
=\ &C_0\left(\lambda_0^{q-1}\left\|x-x'\right\| + \sum_{\tau=0}^{s-2}\lambda_0^{|q-\tau|-1}\sum_{j=i_\tau}^{i_{\tau+1}-1}\left\|\zeta_j - \zeta'_j\right\| + \lambda_0^{(s-1)-q-1}\left\|z-z'\right\|\right) \\
\le\ &\frac{C_0}{\lambda_0}\left(\lambda^{i_q - i_0}\left\|x-x'\right\| + \sum_{\tau=0}^{s-2}\sum_{j=i_\tau}^{i_{\tau+1}-1}\lambda^{|j-i_q|}\left\|\zeta_j - \zeta'_j\right\| + \lambda^{i_{s-1}-i_q}\left\|z-z'\right\|\right) \\
=\ &C\left(\lambda^h\left\|x-x'\right\| + \sum_{\tau=0}^{p-1}\lambda^{|h-\tau|}\left\|\zeta_\tau - \zeta'_\tau\right\| + \lambda^{p-h}\left\|z-z'\right\|\right).
\end{aligned}$$

The last inequality holds because each interval is of length at most $(2d-1)$. Here the constants are

$$C_0 = \frac{2L_0}{m_c},\ \lambda_0 = 1 - 2\cdot\left(\sqrt{1+(2L_0/m_c)}+1\right)^{-1},$$
$$C = C_0/\lambda_0 = \frac{2L_0}{m_c}\left(1 - 2\cdot\left(\sqrt{1+(2L_0/m_c)}+1\right)^{-1}\right)^{-1},$$
$$\lambda = \left(1 - 2\left(\sqrt{1+(2L_0/m_c)}+1\right)^{-1}\right)^{\frac{1}{2d-1}}.$$

The proof of the perturbation bound of $\psi_t^p(\cdot,\cdot,\cdot)_{y_h}$ is quite similar. The only difference lies in the terminal cost, which can be addressed with the addition of a fixed auxiliary state. Specifically, we append $x_{\text{aux}} = 0$ to the end of the decision point sequence, and define a zero transition cost to the auxiliary state $\hat{c}_s(x_{t+p}, \bar{w}_{s-1}, x_{\text{aux}}) \equiv 0$ (note that $\hat{c}_s$ is trivially convex and $L_0$-strongly smooth). Denote the solution of the modified version of transformed SOCO problem by $\hat{\psi}'(x_t, \zeta, x_{\text{aux}})$, then by the same argument as above, we have

$$\begin{aligned}\left\|\tilde{\psi}_t^p(x,\zeta;F)_{y_h} - \tilde{\psi}_t^p(x',\zeta';F)_{y_h}\right\| &= \left\|\hat{\psi}'(x,\zeta,0)_q - \hat{\psi}'(x',\zeta',0)_q\right\| \\ &\le \cdots \le C\left(\lambda^h \|x - x'\| + \sum_{\tau=0}^{p-1}\lambda^{|h-\tau|}\left\|\zeta_\tau - \zeta'_\tau\right\|\right),\end{aligned}$$

where the constants are the same as previously defined. This finishes the proof of Theorem 3.5.

## B.2 Proof of Corollary 3.6

This proof is adapted from [54].

*Proof of Corollary 3.6.* Note that $\tilde{\psi}_t^p(0,0;F)_{y_h} = 0$. By Theorem 3.5, we see that

$$\begin{aligned}\left\|\tilde{\psi}_t^p(x,\zeta;F)_{y_h}\right\| &= \left\|\tilde{\psi}_t^p(x,\zeta;F)_{y_h} - \tilde{\psi}_t^p(0,0;F)_{y_h}\right\| \\ &\le C\left(\lambda^h\|x\| + \sum_{\tau=0}^{p-1}\lambda^{|h-\tau|}\|\zeta_\tau\|\right) \\ &\le C\lambda^h\|x\| + \frac{2C}{1-\lambda}\sup_\tau\|\zeta_\tau\|,\end{aligned}$$

where the last inequality holds because

$$\sum_{\tau=0}^{p-1}\lambda^{|h-\tau|} \le \frac{2}{1-\lambda}.$$

□

### B.3 Proof of Corollary 3.7

This proof is adapted from [54].

Before showing Corollary 3.7, we first show a property of strongly smooth functions.

**Lemma B.1.** *Suppose function $g : \mathbb{R}^n \to \mathbb{R}_+$ is convex, $\ell$-strongly smooth, and continuously differentiable. For all $x, y \in \mathbb{R}^n$ and $\eta > 0$, we have*

$$g(x) \le (1+\eta)g(y) + \frac{\ell}{2}\left(1+\frac{1}{\eta}\right)\|x-y\|^2 .$$

*Proof of Lemma B.1.*

$$\begin{aligned} g(x) - g(y) &\le \langle \nabla g(y), x-y\rangle + \frac{\ell}{2}\|x-y\|^2 \\ &\le \frac{\eta}{2\ell}\|\nabla g(y)\|^2 + \frac{\ell}{2\eta}\|x-y\|^2 + \frac{\ell}{2}\|x-y\|^2 \\ &\le \eta g(y) + \frac{\ell}{2}\left(1+\frac{1}{\eta}\right)\|x-y\|^2 . \end{aligned}$$

where the second inequality follows from the generalized mean inequality and the last inequality holds because

$$0 \le g\left(y - \frac{\nabla g(y)}{\ell}\right) \le g(y) - \left\langle \nabla g(y), \frac{\nabla g(y)}{\ell}\right\rangle + \frac{\ell}{2}\left\|\frac{\nabla g(y)}{\ell}\right\|^2 = g(y) - \frac{1}{2\ell}\|\nabla g(y)\|^2$$

□

Now we come back to the proof of Corollary 3.7.

*Proof of Corollary 3.7.* When $d \le p \le 2d-1$, since $\xi_t^p(x,\zeta,z)$ is $L_0$-strongly smooth by Lemma 3.4, we know

$$\iota_t^p(x,\zeta,z) = \xi_t^p(x,\zeta,z) + f_{t+p}(z)$$

is $(L_0+\ell_f)$-strongly smooth. Therefore, by Lemma B.1, we obtain that

$$\iota_t^p(x,\zeta,z) \le (1+\eta)\iota_t^p(x',\zeta,z') + \frac{L_0+\ell_f}{2}\left(1+\frac{1}{\eta}\right)\left(\|x'-x\|^2 + \|z'-z\|^2\right).$$

When $p = 2d$, let $x_1 := \psi_t^p(x,\zeta,z)_{y_d}$, and we obtain that

$$\begin{aligned} \iota_t^p(x,\zeta,z) &= \iota_t^d(x,\zeta_{0:d-1},x_1) + \iota_{t+d}^d(x_1,\zeta_{d:2d-1},z) \\ &\le (1+\eta)\iota_t^d(x',\zeta_{0:d-1},x_1) + \frac{L_0+\ell_f}{2}\left(1+\frac{1}{\eta}\right)\|x-x'\|^2 \end{aligned}$$

$$+ (1+\eta)\iota_{t+d}^{d}(x_1, \zeta_{d:2d-1}, z') + \frac{L_0+\ell_f}{2}\left(1+\frac{1}{\eta}\right)\|z-z'\|^2$$
$$\le (1+\eta)\iota_t^{p}\left(x', \zeta, z'\right) + \frac{L_0+\ell_f}{2}\left(1+\frac{1}{\eta}\right)\left(\|x'-x\|^2 + \|z'-z\|^2\right).$$

When $p > 2d$, let $x_1 := \psi_t^p(x,\zeta,z)_{y_d}, x_2 := \psi_t^p(x,\zeta,z)_{y_{p-d}}$, and we obtain that

$$\begin{aligned}
\iota_t^p(x,\zeta,z) &= \iota_t^d(x,\zeta_{0:d-1},x_1) + \iota_{t+d}^{p-2d}(x_1,\zeta_{d:p-d-1},x_2) + \iota_{t+p-d}^{d}(x_2,\zeta_{p-d:p-1},z) \\
&\le (1+\eta)\iota_t^d(x',\zeta_{0:d-1},x_1) + \frac{L_0+\ell_f}{2}\left(1+\frac{1}{\eta}\right)\|x-x'\|^2 \\
&\quad + \iota_{t+d}^{p-2d}(x_1,\zeta_{d:p-d-1},x_2) \\
&\quad + (1+\eta)\iota_{t+p-d}^{d}(x_2,\zeta_{p-d:p-1},z') + \frac{L_0+\ell_f}{2}\left(1+\frac{1}{\eta}\right)\|z-z'\|^2 \\
&\le (1+\eta)\iota_t^{p}\left(x',\zeta,z'\right) + \frac{L_0+\ell_f}{2}\left(1+\frac{1}{\eta}\right)\left(\|x'-x\|^2 + \|z'-z\|^2\right).
\end{aligned}$$

□

### B.4 Proof of Lemma 3.8

This proof is adapted from [54].

For simplicity, we will use the shorthand notations

$$\tilde{\psi}_t^p(x;F) := \tilde{\psi}_t^p(x, w_{t:t+p-1}; F) \text{ and } \psi_t^p(x,z) := \psi_t^p(x, w_{t:t+p-1}, z)$$

throughout the proof, since the indices of the disturbances can be inferred from the starting time $t$ and horizon $p$. We also define

$$z := \tilde{\psi}_t^p\left(x_t;F\right)_{y_p}, z' := \tilde{\psi}_t^{p+1}\left(x_t;F\right)_{y_p}.$$

Then it is straightforward to see that

$$\left\|\tilde{\psi}_t^p\left(x_t;F\right)_{y_h} - \tilde{\psi}_t^{p+1}\left(x_t;F\right)_{y_h}\right\| = \left\|\psi_t^p\left(x_t,z\right)_{y_h} - \psi_t^p\left(x_t,z'\right)_{y_h}\right\| \tag{B.1a}$$
$$\le C\lambda^{p-h}\|z-z'\| \tag{B.1b}$$
$$\le 2C\lambda^{p-h}\left(C\lambda^p\|x_t\| + \frac{2C}{1-\lambda}D\right). \tag{B.1c}$$

where we use the definition of $\psi$ and $\tilde{\psi}$ in (B.1a), Theorem 3.5 in (B.1b), and Corollary 3.6 in (B.1c).

### B.5 Proof of Theorem 3.9

This proof is adapted from [54].

Throughout the proof, we will use $\{(\hat{x}_t, \hat{u}_t)\}$ to denote the trajectory of predictive control with prediction window $T$ ($PC_T$). Recall that $\{(x_t, u_t)\}$ denotes the trajectory of predictive control with prediction window $k$ ($PC_k$), and $\{(x_t^*, u_t^*)\}$ denotes the offline optimal trajectory ($OPT$), i.e., the optimal solution of (3.1).

For simplicity, we will use the shorthand notations

$$\tilde{\psi}_t^p(x; F) := \tilde{\psi}_t^p(x, w_{t:t+p-1}; F) \text{ and } \psi_t^p(x, z) := \psi_t^p(x, w_{t:t+p-1}, z)$$

throughout the proof.

First, we shall introduce a consequence of Corollary 3.6 and Lemma 3.8 that bounds the trajectory of predictive control, the proof of which can be found in [54].

**Lemma B.2** (Input State Stability)**.** *Under the same condition as Lemma 3.9 and Equation* (3.6)*, the norm of each state $x_t$ is upper bounded by*

$$\|x_t\| \le \begin{cases} \frac{C}{\delta} \cdot (1-\delta)^{\max(0,t-k)} \|x_0\| + \frac{2C}{\delta(1-\lambda)}\left(1 + \frac{2C}{1-\lambda}\right) D & \text{if } 0 < t \le T-k \\ \frac{C^2}{\delta} \cdot (1-\delta)^{T-2k} \lambda^{t+k-T} \|x_0\| + \left(\frac{2C^2}{\delta(1-\lambda)}\left(1 + \frac{2C}{1-\lambda}\right) + \frac{2C}{1-\lambda}\right) D & \text{if } T-k < t \le T. \end{cases}$$

*Proof of Theorem 3.9.* By Lemmas 3.8 and B.2, we also see that for $t \le T-k$,

$$\begin{aligned} \left\|\tilde{\psi}_t^k(x_t; F)_{y_1} - \tilde{\psi}_t^{T-t}(x_t; F)_{y_1}\right\| &\le \sum_{p=k}^{T-t} \left\|\tilde{\psi}_t^p(x_t; F)_{y_1} - \tilde{\psi}_t^{p+1}(x_t; F)_{y_1}\right\| \\ &\le \sum_{p=k}^{\infty} 2C\lambda^{p-1}\left(C\lambda^p \|x_t\| + \frac{2C}{1-\lambda} D\right) \\ &= \frac{2C^2}{\lambda(1-\lambda^2)} \cdot \lambda^{2k} \|x_t\| + \frac{4C^2}{\lambda(1-\lambda)^2} \cdot \lambda^k D \\ &= O\left(\left(D + \frac{\lambda^k(\|x_0\| + D)}{\delta}\right)\lambda^k\right). \end{aligned} \tag{B.2}$$

We further obtain that for $t \le T-k$,

$$\begin{aligned} \|x_t - \hat{x}_t\| = {} & \left\|x_t - \tilde{\psi}_0^T(x_0; F)\right\| \\ \le {} & \left\|x_t - \tilde{\psi}_{t-1}^{T-t+1}(x_{t-1}; F)_{y_1}\right\| + \sum_{i=1}^{t-1} \left\|\tilde{\psi}_{t-i}^{T-t+i}(x_{t-i}; F)_{y_i} - \tilde{\psi}_{t-i-1}^{T-t+i+1}(x_{t-i-1}; F)_{y_{i+1}}\right\| \\ \le {} & \left\|x_t - \tilde{\psi}_{t-1}^{T-t+1}(x_{t-1}; F)_{y_1}\right\| + \sum_{i=1}^{t-1} C\lambda^i \left\|x_{t-i} - \tilde{\psi}_{t-i-1}^{T-t+i+1}(x_{t-i-1}; F)_{y_1}\right\| \end{aligned} \tag{B.3a}$$

$$= O\left(\left(D + \frac{\lambda^k(\|x_0\| + D)}{\delta}\right)\lambda^k\right), \tag{B.3b}$$

where in (B.3a), we use Theorem 3.5 and the fact that $\tilde{\psi}_{t-i-1}^{T-t+i+1}(x_{t-i-1})_{y_{i+1}}$ can be written as

$$\tilde{\psi}_{t-i-1}^{T-t+i+1}(x_{t-i-1}; F)_{y_{i+1}} = \tilde{\psi}_{t-i}^{T-t+i}\left(\tilde{\psi}_{t-i-1}^{T-t+i+1}(x_{t-i-1}; F)_{y_1}; F\right)_{y_i};$$

in (B.3b), we use (B.2) and the following observations

$$\left\|x_{t-i} - \tilde{\psi}_{t-i-1}^{T-t+i+1}(x_{t-i-1}; F)_{y_1}\right\| = \left\|\tilde{\psi}_{t-i-1}^{k}\left(x_{t-i-1}; F\right)_{y_1} - \tilde{\psi}_{t-i-1}^{T-t+i+1}(x_{t-i-1}; F)_{y_1}\right\|,$$

$$1 + \sum_{i=1}^{t-1} C\lambda^i \le 1 + \frac{C}{1-\lambda} = O(1).$$

By Corollary 3.6 and triangle inequality, we see that

$$\left\|x_T^* - \hat{x}_T\right\| \le 2C\lambda^T\|x_0\| + \frac{4CD}{1-\lambda}.$$

Then by Theorem 3.5, the following holds for all $t \le T - k$:

$$\left\|x_t^* - \hat{x}_t\right\| = \left\|\psi_0^T(x_0, x_T^*) - \psi_0^T(x_0, \hat{x}_T)\right\| \le C\lambda^k\left(2C\lambda^T\|x_0\| + \frac{4CD}{1-\lambda}\right).$$

Combining this inequality with (B.3) gives

$$\left\|x_t - x_t^*\right\| = O\left(\left(D + \frac{\lambda^k(\|x_0\| + D)}{\delta}\right)\lambda^k\right), \ \forall t \le T - k. \tag{B.4}$$

Since

$$(u_t - u_t^*) = B_t^\dagger\left((x_{t+1} - x_{t+1}^*) - A_t(x_t - x_t^*)\right),$$

we have

$$\left\|u_t - u_t^*\right\| \le b'\left(\left\|x_{t+1} - x_{t+1}^*\right\| + a\left\|x_t - x_t^*\right\|\right).$$

Therefore, by Corollary 3.7, for any $\eta > 0$ we have

$$\begin{aligned}
&\iota_t^1(x_t, x_{t+1}) - (1+\eta)\iota_t^1(x_t^*, x_{t+1}^*) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(B.5)}\\
=\ &\left(f_{t+1}(x_{t+1}) - (1+\eta)f_{t+1}(x_{t+1}^*)\right) + \left(c_{t+1}(u_t) - (1+\eta)c_{t+1}(u_t^*)\right)\\
\le\ &\frac{1}{2}\left(1 + \frac{1}{\eta}\right)\left(\ell_f\left\|x_{t+1} - x_{t+1}^*\right\|^2 + \ell_c\left\|u_t - u_t^*\right\|^2\right)\\
\le\ &\frac{1}{2}\left(1 + \frac{1}{\eta}\right)\left(\ell_f + 2(b')^2\ell_c\right)\left\|x_{t+1} - x_{t+1}^*\right\|^2 + \frac{1}{2}\left(1 + \frac{1}{\eta}\right)2a^2(b')^2\ell_c\left\|x_t - x_t^*\right\|^2\\
\le\ &\left(1 + \frac{1}{\eta}\right)\cdot\frac{L_4}{2}\left(\left\|x_t - x_t^*\right\|^2 + \left\|x_{t+1} - x_{t+1}^*\right\|^2\right),
\end{aligned}$$

where

$$L_4 := \ell_f + 2(b')^2\ell_c + 2a^2(b')^2\ell_c.$$

Then, for any $\eta > 0$, we obtain the following inequality:

$$\begin{aligned}
&cost(PC_k) - (1+\eta)cost(OPT)\\
=& \left(\sum_{t=0}^{T-k-1} \iota_t^1(x_t, x_{t+1}) + \iota_{T-k}^k(x_{T-k}, x_T)\right) - (1+\eta)\left(\sum_{t=0}^{T-k-1} \iota_t^1(x_t^*, x_{t+1}^*) + \iota_{T-k}^k(x_{T-k}^*, x_T^*)\right)\\
=& \sum_{t=0}^{T-k-1}\left(\iota_t^1(x_t, x_{t+1}) - (1+\eta)\iota_t^1(x_t^*, x_{t+1}^*)\right) + \left(\iota_{T-k}^k(x_{T-k}, x_T) - (1+\eta)\iota_{T-k}^k(x_{T-k}^*, x_T^*)\right)\\
\le& \sum_{t=0}^{T-k-1}\left(\iota_t^1(x_t, x_{t+1}) - (1+\eta)\iota_t^1(x_t^*, x_{t+1}^*)\right) + \left(\iota_{T-k}^k(x_{T-k}, x_T^*) - (1+\eta)\iota_{T-k}^k(x_{T-k}^*, x_T^*)\right) && \text{(B.6a)}\\
\le& \left(1+\frac{1}{\eta}\right)\cdot\frac{L_4}{2}\sum_{t=0}^{T-k-1}\left(\left\|x_t - x_t^*\right\|^2 + \left\|x_{t+1} - x_{t+1}^*\right\|^2\right) + \left(1+\frac{1}{\eta}\right)\cdot\frac{L_0+\ell_f}{2}\left\|x_{T-k} - x_{T-k}^*\right\|^2 && \text{(B.6b)}\\
=& \left(1+\frac{1}{\eta}\right)\cdot L_4\sum_{t=0}^{T-k-1}\left\|x_t - x_t^*\right\|^2 + \left(1+\frac{1}{\eta}\right)\cdot\frac{L_4+L_0+\ell_f}{2}\left\|x_{T-k} - x_{T-k}^*\right\|^2\\
\le& \left(1+\frac{1}{\eta}\right)O\left(\left(D + \frac{\lambda^k(\|x_0\|+D)}{\delta}\right)^2\lambda^k T\right), && \text{(B.6c)}
\end{aligned}$$

where we use the fact that our algorithm $PC_k$ plans optimally after time step $T-k$ in (B.6a); we also use (B.5) and Corollary 3.7 in (B.6b), and (B.3) in (B.6c).

To bound the optimal cost, we consider a suboptimal controller inspired by the decision-point transformation, where the controller forces the states $x_d, x_{2d}, \cdots, x_{(v-1)d}$, and $x_{vd+r}$ to be 0 ($d$ is the controllability index, and $T = vd + r$). The cost of this suboptimal control is determined by the transformed transition cost $\xi_t^p(\cdot,\cdot,\cdot)$ between each pair of consecutive decision points. By strong smoothness of $\xi_t^p(\cdot,\cdot,\cdot)$ proven in Lemma 3.4, we have

$$\xi_t^p(x,\zeta,0) \le \frac{1}{2}L_2(p)\left(\|\zeta\|^2 + \|x\|^2\right) \le \frac{L_0 D^2}{2}p + \frac{L_0}{2}\|x\|^2,$$

where $L_0 = \max_{d\le p\le 2d-1} L_2(p)$. These inequalities add up to

$$cost(OPT) \le \xi_0^d(x_0, w_{0:d-1}, 0) + \sum_{\tau=1}^{v-2}\xi_{\tau d}^d(0, w_{\tau d:(\tau+1)d-1}, 0) + \xi_{(t-1)d}^{d+r}(0, w_{(v-1)d:T-1}, 0)$$

$$
\begin{aligned}
&\leq \frac{L_0 D^2}{2} T + \frac{L_0}{2} \|x_0\|^2 \\
&= O(D^2 T + \|x_0\|^2).
\end{aligned}
$$

Hence $cost(OPT) = O(D^2 T + \|x_0\|^2)$. Now we can take $\eta = \Theta(\lambda^k)$ in (B.6) to get a regret bound of

$$
cost(PC_k) - cost(OPT) = O\left(\left(D + \frac{\lambda^k(\|x_0\| + D)}{\delta}\right)^2 \lambda^k T + \lambda^k \|x_0\|^2\right).
$$

□

### B.6 Proof of Theorem 3.10

This proof is adapted from [54].

To simplify the notation, we still omit the disturbance sequence $w_{t:t+k-1}$ in $\tilde{\psi}_t^k$ and $\psi_t^k$ throughout the proof. At each time step $t$, we will use $x_t/u_t$ to denote the state/input of $PC_{(k,h)}$ algorithm and use $x_t^*/u_t^*$ to denote the state/input of the offline optimal. We define

$$
\begin{aligned}
H_t &:= f_t(x_t), M_t := c_t(u_{t-1}), \\
H_t^* &:= f_t(x_t^*), M_t^* := c_t(u_{t-1}^*).
\end{aligned}
$$

Let $\tilde{x}_{t+k} := \tilde{\psi}_t^k(x_t^*; F)_{y_k}, \bar{x}_{t+k} = \tilde{\psi}_t^k(x_t^*; 0)_{y_k}$.

If $t \leq T - k, t \equiv 0(\text{mod } h)$, we have

$$
\begin{aligned}
&\left\|x_{t+h} - x_{t+h}^*\right\|^2 \\
&= \left\|\tilde{\psi}_t^k(x_t, F)_{y_h} - \psi_t^k(x_t^*, x_{t+k}^*)_{y_h}\right\|^2 \\
&\leq \left(\left\|\tilde{\psi}_t^k(x_t, F)_{y_h} - \tilde{\psi}_t^k(x_t^*, F)_{y_h}\right\| + \left\|\tilde{\psi}_t^k(x_t^*, F)_{y_h} - \psi_t^k(x_t^*, x_{t+k}^*)_{y_h}\right\|\right)^2 && \text{(B.7a)} \\
&\leq (1+\epsilon)\left\|\tilde{\psi}_t^k(x_t, F)_{y_h} - \tilde{\psi}_t^k(x_t^*, F)_{y_h}\right\|^2 + \left(1 + \frac{1}{\epsilon}\right)\left\|\tilde{\psi}_t^k(x_t^*, F)_{y_h} - \psi_t^k(x_t^*, x_{t+k}^*)_{y_h}\right\|^2 && \text{(B.7b)} \\
&\leq (1+\epsilon)\left\|\tilde{\psi}_t^k(x_t, F)_{y_h} - \tilde{\psi}_t^k(x_t^*, F)_{y_h}\right\|^2 + \left(1 + \frac{1}{\epsilon}\right)\left\|\psi_t^k(x_t^*, \tilde{x}_{t+k})_{y_h} - \psi_t^k(x_t^*, x_{t+k}^*)_{y_h}\right\|^2 && \text{(B.7c)} \\
&\leq (1+\epsilon) C^2 \lambda^{2h} \left\|x_t - x_t^*\right\|^2 + C^2 \lambda^{2(k-1-h)} \cdot \left(1 + \frac{1}{\epsilon}\right)\left\|x_{t+k}^* - \tilde{x}_{t+k}\right\|^2 && \text{(B.7d)} \\
&\leq \frac{1}{1+\epsilon}\left\|x_t - x_t^*\right\|^2 + C^2 \lambda^{2(k-1-h)} \cdot \left(1 + \frac{1}{\epsilon}\right)\left\|x_{t+k}^* - \tilde{x}_{t+k}\right\|^2, && \text{(B.7e)}
\end{aligned}
$$

where we use the triangle inequality in (B.7a), the AM-GM inequality in (B.7b), the definition of $\tilde{x}_{t+k}$ in (B.7c), Theorem 3.5 in (B.7d), and the assumption on $h$ in (B.7e).

Since the objective function is $m_f$-strongly convex in variables $x_{t+1:t+k}$, we see that

$$\left\|x_{t+k}^* - \bar{x}_{t+k}\right\|^2 \le \frac{2}{m_f} \sum_{\tau=1}^{k} (H_{t+\tau}^* + M_{t+\tau}^*). \tag{B.8}$$

Since $f_{t+k}$ is $m_f$-strongly convex, we also see that $\left\|x_{t+k}^*\right\|^2 \le \frac{2}{m_f} H_{t+k}^*$. Recall that the terminal cost $F(x_{t+k}) = \alpha(\|x_{t+k}\|)$, where $\alpha$ is a K-function. By the definition of $\tilde{\psi}$, we see that $\|\tilde{x}_{t+k}\| \le \|\bar{x}_{t+k}\|$. Therefore, we obtain that

$$\begin{aligned}
\left\|x_{t+k}^* - \tilde{x}_{t+k}\right\|^2 &\le 2\left\|\tilde{x}_{t+k}\right\|^2 + 2\left\|x_{t+k}^*\right\|^2 && \text{(B.9a)}\\
&\le 2\left\|\bar{x}_{t+k}\right\|^2 + 2\left\|x_{t+k}^*\right\|^2 && \text{(B.9b)}\\
&\le 4\left\|\bar{x}_{t+k} - x_{t+k}^*\right\|^2 + 6\left\|x_{t+k}^*\right\|^2 && \text{(B.9c)}\\
&\le \frac{8}{m_f}\sum_{\tau=1}^{k}(H_{t+\tau}^* + M_{t+\tau}^*) + \frac{12}{m_f}H_{t+k}^* && \text{(B.9d)}\\
&\le \frac{20}{m_f}\sum_{\tau=1}^{k}(H_{t+\tau}^* + M_{t+\tau}^*),
\end{aligned}$$

where we use Cauchy-Schwarz inequality in (B.9a) and (B.9c), $\|\tilde{x}_{t+k}\| \le \|\bar{x}_{t+k}\|$ in (B.9b), and (B.8) in (B.9d).

Suppose $T = n_0 \cdot h + m_0$, where $n_0 \in \mathbb{Z}_+$ and $k - h + 1 \le m_0 \le k$. By summing up inequality (B.7) for $t = 0, h, 2h, \ldots, (n-1)h$, we obtain that

$$\begin{aligned}
\sum_{i=1}^{n_0}\left\|x_{ih} - x_{ih}^*\right\|^2 &\le C^2\lambda^{2(k-1-h)} \cdot \frac{(1+\epsilon)^2}{\epsilon^2} \cdot \sum_{i=1}^{n_0-1}\left\|x_{ih+k}^* - \tilde{x}_{ih+k}\right\|^2\\
&\le C^2\lambda^{2(k-1-h)} \cdot \frac{(1+\epsilon)^2}{\epsilon^2} \cdot \frac{20}{m_f} \cdot cost(OPT),
\end{aligned} \tag{B.10}$$

where we use (B.9) in the last inequality.

Therefore, we can show that, for all $\eta > 0$:

$$\begin{aligned}
&cost(PC_{(k,h)}) - (1+\eta)cost(OPT)\\
={}& \left(\sum_{i=0}^{n_0-1}\iota_{ih}^{h}(x_{ih}, x_{(i+1)h}) + \iota_{n_0h}^{m_0}(x_{n_0h}, x_T)\right) - (1+\eta)\left(\sum_{i=0}^{n_0-1}\iota_{ih}^{h}(x_{ih}^*, x_{(i+1)h}^*) + \iota_{n_0h}^{m_0}(x_{n_0h}^*, x_T^*)\right)
\end{aligned}$$

$$= \sum_{i=0}^{n_0-1} \left( \iota_{ih}^{h}(x_{ih}, x_{(i+1)h}) - (1+\eta)\iota_{ih}^{h}(x_{ih}^*, x_{(i+1)h}^*) \right) + \left( \iota_{n_0 h}^{m_0}(x_{nh}, x_T) - (1+\eta)\iota_{n_0 h}^{m_0}(x_{nh}^*, x_T^*) \right)$$

$$\leq \sum_{i=0}^{n_0-1} \left( \iota_{ih}^{h}(x_{ih}, x_{(i+1)h}) - (1+\eta)\iota_{ih}^{h}(x_{ih}^*, x_{(i+1)h}^*) \right) + \left( \iota_{n_0 h}^{m_0}(x_{nh}, x_T^*) - (1+\eta)\iota_{n_0 h}^{m_0}(x_{nh}^*, x_T^*) \right) \tag{B.11a}$$

$$\leq \left(1 + \frac{1}{\eta}\right) \cdot \frac{L_0 + \ell_f}{2} \sum_{i=0}^{n_0-1} \left( \left\| x_{ih} - x_{ih}^* \right\|^2 + \left\| x_{(i+1)h} - x_{(i+1)h}^* \right\|^2 \right)$$

$$+ \left(1 + \frac{1}{\eta}\right) \cdot \frac{L_0 + \ell_f}{2} \left\| x_{n_0 h} - x_{n_0 h}^* \right\|^2 \tag{B.11b}$$

$$= \left(1 + \frac{1}{\eta}\right) \cdot (L_0 + \ell_f) \sum_{i=1}^{n_0} \left\| x_{ih} - x_{ih}^* \right\|^2$$

$$\leq \left(1 + \frac{1}{\eta}\right) \cdot (L_0 + \ell_f) \cdot C^2 \lambda^{2(k-1-h)} \cdot \frac{20(1+\epsilon)^2}{m\epsilon^2} \cdot cost(OPT), \tag{B.11c}$$

where we use the fact that the PC algorithm (with replan window $h$) plans optimally after time step $nh$ in (B.11a); we also use Corollary 3.7 in (B.11b), and (B.10) in (B.11c).

By setting $\eta \sim \epsilon^{-1} \left( \frac{L_0 + \ell_f}{m} \right)^{\frac{1}{2}} \cdot C\lambda^{k-1-h}$, we see that the competitive ratio of the $PC_{(k,h)}$ algorithm (with replan window $h$) is in the order of

$$1 + O\left( \epsilon^{-1} \left( \frac{L_0 + \ell_f}{m} \right)^{\frac{1}{2}} \cdot C\lambda^{k-1-h} \right).$$