Online Algorithms: From Prediction to Decision

Thesis by

Niangjun Chen

In Partial Fulfillment of the Requirements for the
degree of
Doctor of Philosophy

Caltech

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2018
Defended September 27, 2017

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisors, Prof. Adam Wierman and Prof. Steven Low, for their helpful guidance and advise for my research, and also for their encouragement and support when I encountered difficulties. It has truly been an honor and great pleasure working them. They not only taught me many essential skills of how to do good research, but they also showed me by their example how to be a good researcher.

I would also like to thank my many collaborators: Siddharth Barman, Krishnamurthy Dvijotham, Lingwen Gan, Xiaoqi Ren, Yorie Nakahira, Navid Azizan-Ruhi, Gautam Goel, Palma London, Ben Razon and Anish Agarwal, from Caltech, Prof. Shaolei Ren from UCI, Prof. Anshul Gandhi, Prof. Zhenhua Liu and Joshua Comden from Stony Brooks, and Prof. Lachlan Andrew from Monash University. I am fortunate to have learned many things from my collaborators, and our collaborations led to many interesting results that contributed to this thesis.

My five years in the Department of Computing and Mathematical Sciences at Caltech have been truly enjoyable. I would like to thank my friends and colleagues in CMS for creating the helpful and friendly atmosphere. I would also like to thank the helpful administrative staff in our department, especially Maria Lopez, Sydney Garstang, Christine Ortega, and Sheila Shull for keeping everything working so that I don't have to worry about anything other than my research.

Finally, I would like to thank my family, who supported me all the way through the highs and lows of my life, and my girlfriend Yorie Nakahira, whose love made this thesis possible.

ABSTRACT

Making use of predictions is a crucial, but under-explored, area of sequential decision problems with limited information. While in practice most online algorithms rely on predictions to make real time decisions, in theory their performance is only analyzed in simplified models of prediction noise, either adversarial or i.i.d. The goal of this thesis is to bridge this divide between theory and practice: to study online algorithm under more practical predictions models, gain better understanding about the value of prediction, and design online algorithms that make the best use of predictions.

This thesis makes three main contributions. First, we propose a stochastic prediction error model that generalizes prior models in the learning and stochastic control communities, incorporates correlation among prediction errors, and captures the fact that predictions improve as time passes. Using this general prediction model, we prove that Averaging Fixed Horizon Control (AFHC) can simultaneously achieve sublinear regret and constant competitive ratio in expectation using only a constant-sized prediction window, overcoming the hardnesss results in adversarial prediction models. Second, to understand the optimal use of noisy prediction, we introduce a new class of policies, Committed Horizon Control (CHC), that generalizes both popular policies Receding Horizon Control (RHC) and Averaging Fixed Horizon Control (AFHC). Our results provide explicit results characterizing the optimal use of prediction in CHC policy as a function of properties of the prediction noise, e.g., variance and correlation structure. Third, we apply the general prediction model and algorithm design framework to the deferrable load control problem in power systems. Our proposed model predictive algorithm provides significant reduction in variance of total load in the power system. Throughout this thesis, we provide both average-case analysis and concentration results for our proposed online algorithms, highlighting that the typical performance is tightly concentrated around the average-case performance.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] Niangjun Chen, Lingwen Gan, Steven H. Low, and Adam Wierman. Distributional analysis for model predictive deferrable load control. In *53rd IEEE Conference on Decision and Control*, pages 6433–6438, Dec 2014. doi: 10.1109/CDC.2014.7040398.
N.C participated in the conception of the project, formulated the problem, applied and analyzed the performance of model predictive load control algorithm and participated in the writing of the manuscript.

[2] Niangjun Chen, Anish Agarwal, Adam Wierman, Siddharth Barman, and Lachlan L.H. Andrew. Online convex optimization using predictions. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '15, pages 191–204, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3486-0. doi: 10.1145/2745844.2745854. URL http://doi.acm.org/10.1145/2745844.2745854.
N.C participated in the conception of the project, formulated the problem, applied and analyzed the performance of averaging fixed horizon control (AFHC) algorithm and participated in the writing of the manuscript.

[3] Niangjun Chen, Joshua Comden, Zhenhua Liu, Anshul Gandhi, and Adam Wierman. Using predictions in online optimization: Looking forward with an eye on the past. In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*, SIGMETRICS '16, pages 193–206, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4266-7. doi: 10.1145/2896377.2901464. URL http://doi.acm.org/10.1145/2896377.2901464.
N.C participated in the conception of the project, formulated the problem, proposed the Committed Horizon Control (CHC) algorithm to solve the problem, analyzed its performance and participated in the writing of the manuscript.

[4] Lingwen Gan, Adam Wierman, Ufuk Topcu, Niangjun Chen, and Steven H. Low. Real-time deferrable load control: Handling the uncertainties of renewable generation. In *Proceedings of the Fourth International Conference on Future Energy Systems*, e-Energy '13, pages 113–124, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2052-8. doi: 10.1145/2487166.2487179. URL http://doi.acm.org/10.1145/2487166.2487179.
N.C participated in the conception of the project, participated in the performance analysis of model predictive load control algorithm and participated in the writing of the manuscript.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Chapter 1

INTRODUCTION

Many important applications in control and communication involve making sequential decisions with limited future information, e.g., balancing electricity demand and generation in power systems, dynamic right-sizing of servers in data-centers, and adapting video encoding bit-rates in video streaming. We call such sequential decision making *online algorithms*. Improving the efficiency of online algorithms has potentially huge benefits to society, e.g., it is estimated in 2012 that the cost savings from improving the efficiency of real time power dispatch in the US amounts to $19 billions per year [67]. In practice, many important online algorithms rely on *predictions* to make real time decisions. This thesis is about designing online algorithms that make the best use of predictions.

While as separate subjects there is an abundance of literature on designing online algorithms and making predictions, it is less understood how accuracy of predictions should impact the design of online algorithms. For example, as prediction error in the future changes, how should we adapt our online algorithm? Is it possible to quantify the performance of online algorithms as a function of the prediction errors? In this thesis, we address these questions using the *online convex optimization* framework, which models many real problems in control and communications. We show that insights that can be gained from incorporating a general, realistic model of prediction noise into the analysis of online algorithms.

1.1 Online convex optimization

In an online convex optimization (OCO) problem, a learner interacts with an environment in a sequence of rounds. In round t the learner chooses an action x_t from a convex decision/action space G, and then the environment reveals a convex cost function c_t and the learner pays cost $c_t(x_t)$. An algorithm's goal is to minimize total cost over a (long) horizon T.

OCO has a rich theory and a wide range of important applications. In computer science, it is most associated with the so-called k-experts problem, an online learning problem where in each round t the algorithm chooses one of k possible actions, viewed as following the advice of one of k "experts".

In typical applications of online convex optimization in networking and cloud computing there is an additional cost in each round, termed a "switching cost", that captures the cost of changing actions during a round. Specifically, the cost is

$$\sum_{t=1}^{T} c_t(x_t) + \beta \|x_t - x_{t-1}\|, \tag{1.1}$$

where $\| \cdot \|$ is a norm (often the one-norm). This additional term makes the online problem more challenging since the optimal choice in a round then depends on future cost functions. These "smoothed" online convex optimization problems have received considerable attention in the context of networking and cloud computing applications, e.g., [56, 58–61], and are also relevant for many more traditional online convex optimization applications where, in reality, there is a cost associated with a change in action, e.g., portfolio management. *We focus on smoothed online convex optimization problems.*

1.2 Example applications

OCO is being increasingly broadly applied, and recently has become prominent in networking and cloud computing applications, including the design of dynamic capacity planning, load shifting and demand response for data centers [49, 58–60, 66], geographical load balancing of internet-scale systems [56, 82], electrical vehicle charging [29, 49], video streaming [44, 45] and thermal management of systems-on-chip [87, 88]; we briefly explain some examples in the following paragraphs. Note that for each application below, decisions need to be made without exact knowledge of the future costs.

Geographical load balancing for data centers In *geographical load balancing*, we seek to minimize the energy cost by exploiting the geographical diversity of internet-scale system. At each time t, there is $\lambda_{t,j}$ amount of traffic generated from source j. The variable $x_t \in (x_{t,s})_{s\in\{1,\dots,S\}}$ represents the number of active servers at data s at time t. The energy cost at time t, $c_t(x_t) = \sum_{s=1}^{S} c_{t,s}(x_{t,s}, \sum_j \lambda_{t,s,j})$ where $\lambda_{t,s,j}$ represents on the amount of real time traffic from source j routed to data center s at time t. Assuming the state transition cost of a server (from sleep state to active state and vice versa) is β, then the switching cost $\beta\|x_t - x_{t-1}\|_1$ represents the cost of state transition cost of servers in the internet-scale system.

Power systems economic dispatch In power systems *economic dispatch*, the goal is to satisfy the energy demand with minimum production cost. At each time t, the generators need to satisfy y_t of total demand, and $x_t = (x_{t,g})_{g\in\{1,\ldots,G\}}$ represents the amount of energy produce by generator g at time t. The cost $c_t(x_t, y_t)$ represents the power generation cost given the demand y_t and generator power output profile x_t. Assuming generator g has a ramp cost of $\beta_g > 0$, then the switching cost with appropriately defined norm $\|x_t - x_{t-1}\| = \sum_{g=1}^{G} \beta_g |x_{t,g} - x_{t-1,g}|$ captures the ramp cost due to change in generation output.

Adaptive streaming In *adaptive streaming* problem, the goal is to adapt the bit-rate representation of different segments of the video stream over varying network states, in order to maximize the playback quality while avoiding fluctuation. Here, the variable x_t represents the bit-rate request for the next segment at time t. At each time, the cost $c_t(x_t) = c_t(x_t, y_t)$ represents the utility of requesting next segment with x_t when the link condition is given by y_t, and the switching cost $\beta\|x_t - x_{t-1}\|$ penalizes rapid changing of bit-rate between consecutive video segments.

1.3 Incorporating predictions

It is no surprise that predictions are crucial to online algorithms in practice. In OCO, knowledge about future cost functions is valuable, even when noisy. However, despite the importance of predictions, we do not understand how prediction noise affects the performance (and design) of online convex optimization algorithms.

This is not due to a lack of effort. Most papers that apply OCO algorithms to networking and cloud computing applications study the impact of prediction noise, e.g., [2, 8, 59, 66]. Typically, these consider numerical simulations where i.i.d. noise terms with different levels of variability are added to the parameter being predicted, e.g., [36, 80]. While this is a valuable first step, it does not provide any *guarantees* about the performance of the algorithm with realistic prediction errors (which tend to be correlated, since an overestimate in one period is likely followed by another overestimate) and further does not help inform the *design* of algorithms that can effectively use predictions.

Though most work on predictions has been simulation based, there has also been significant work done seeking analytic guarantees. This literature can be categorized into:

(i) *Worst-case models* of prediction error typically assume that there exists a lookahead window ω such that within that window, prediction is near-perfect (*too optimistic*), and outside that window the workload is adversarial (*too pessimistic*), e.g., [13, 19, 56, 58, 61].

(ii) *Simple stochastic models* of prediction error typically consider i.i.d. errors, e.g., [16, 59, 60]. Although this is analytically appealing, it ignores important features of prediction errors, as described in the next section.

(iii) *Detailed stochastic models* of specific predictors applied for specific signal models, such as [46, 73, 74, 89]. This leads to less pessimistic results, but the guarantees, and the algorithms themselves, become *too fragile* to assumptions on the system evolution.

1.4 Performance metrics: regret vs competitive ratio

There are two main performance metrics used for online algorithms in the literature: *regret*, defined as the difference between the cost of the algorithm and the cost of the offline optimal static solution, and the *competitive ratio*, defined as the maximum ratio between the cost of the algorithm and the cost of the offline optimal (dynamic) solution. While in practice online algorithm can often perform well with respect to both regret and competitive ratio; however, as shown in Chapter 2, in adversarial setting, no algorithm can simultaneous achieve good performance in both metrics. This motivates us to look at a more realistic model for predictions. The resulting general prediction model also gives new insights into designing good online algorithms that make use of predictions (Chapter 4).

1.5 Overview of this thesis.

This thesis makes the following contributions.

Generalizing prediction model

In Chapter 2, we show in Theorem 2.1 that, for the adversarial prediction model (where predictions are exact within the prediction window and adversarial beyond), no algorithm can achieve sublinear regret and constant competitive ratio simultaneously, no matter how long the prediction window is. This motivates us to introduce a general colored noise model for studying prediction errors in online convex optimization problems. The model captures three important features of real predictors: (i) it allows for arbitrary correlations in prediction errors (e.g., both short and long range correlations); (ii) the quality of predictions decreases the further in the future we try to look ahead; and (iii) predictions about the future are updated as time passes. Furthermore, it strikes a middle ground between the worst-case and stochastic approaches. In particular, it does not make any assumptions about an underlying stochastic process or the design of the predictor. Instead, it only makes (weak) assumptions about the stochastic form of the error of the predictor; these assumptions are satisfied by many common stochastic models, e.g., the prediction error of standard Weiner filters [84] and Kalman filters [48]. Importantly, by being agnostic to the underlying stochastic process, the model allows worst-case analysis with respect to the realization of the underlying cost functions.

Bridging hardness result between regret and competitive ratio

In Chapter 3, *we show that a simple algorithm, Averaging Fixed Horizon Control (AFHC) [56], simultaneously achieves sublinear regret and a constant competitive ratio in expectation using very limited prediction*, i.e., a prediction window of size $O(1)$, in nearly all situations when it is feasible for an online algorithm to do so (Theorem 3.1). Further, we show that the performance of AFHC is tightly concentrated around its mean (Theorem 3.9). Thus, AFHC extracts the asymptotically optimal value from predictions. Additionally, our results inform the choice of the optimal prediction window size. (For ease of presentation, both Theorems 3.8 and 3.9 are stated and proven for the specific case of online LASSO – see Section 2.1 – but the proof technique can be generalized in a straightforward way.)

Importantly, Theorem 3.4 highlights that the dominant factor impacting whether the prediction window should be long or short in AFHC is not the variance of the noise, but rather the correlation structure of the noise. For example, if prediction errors are i.i.d. then it is optimal for AFHC to look ahead as far as possible (i.e., T) regardless of the variance, but if prediction errors have strong short-range dependencies then the optimal prediction window is constant sized regardless of the variance.

Previously, AFHC had only been analyzed in the adversarial model [58], and our results are in stark contrast to the pessimism of prior work. To highlight this, we prove that in the "easiest" adversarial model (where predictions are exact within the prediction window), no online algorithm can achieve sublinear regret *and* a constant competitive ratio when using a prediction window of constant size (Theorem 2.1). This contrast emphasizes the value of moving to a more realistic stochastic model of prediction error.

Optimizing the use of prediction

Receding horizon control (RHC) has a long history in the control theory literature [11, 18, 33, 50, 63, 83]. While it has been successful in many applications, [56] shows that in the adversarial prediction setting, RHC with w steps lookahead has competitive ratio of $1 + \Omega(1)$, i.e., the performance of RHC does not improve with increasing accurate prediction. The contrast between the good performance of RHC in practice against the pessimistic performance guarantee motivates us to move beyond the worst case in the performance analysis of online algorithms.

In Chapter 4, our goal is to provide new insights into the design of algorithms for OCO problems with noisy predictions. In particular, our results highlight the importance of *commitment* in online algorithms, and the significant performance gains that can be achieved by tuning the *commitment level* of an algorithm as a function of structural properties of the prediction noise such as variance and correlation structure.

In terms of commitment, receding horizon control (RHC) and averaging fixed horizon control (AFHC) represent two extreme algorithm designs: RHC commits to only one action at a time whereas AFHC averages over algorithms that commit to actions spanning the whole prediction horizon. While the non-committal nature of RHC enables quick response to improved predictions, it makes RHC *susceptible to switching costs*. On the other hand, the cautious nature of AFHC averts switching costs but makes it entirely *dependent on the accuracy of predictions*.

Motivated by these deficiencies in existing algorithm design, we introduce a new class of policies, **Committed Horizon Control** (CHC), that allows for arbitrary levels of commitment and thus subsumes RHC and AFHC. We present both average-case analysis (Theorems 4.1 and 4.6) and concentration results (Theorems 4.7) for CHC policies. In doing so, we provide the *first analysis* of RHC with noisy predictions.

Our results demonstrate that intermediate levels of commitment can provide significant reductions in cost, to the tune of more than 50% (e.g., Figure 4.4a, Figure 4.5a and Figure 4.6a). Further, our results also reveal the impact of correlation structure and variance of prediction noise on the optimal level of commitment, and provide simple guidelines on how to choose between RHC and AFHC.

These results are enabled by a key step in our proof that transforms the control strategy employed by the offline optimal algorithm, OPT to the strategy of CHC via a trajectory of intermediate strategies. We exploit the structure of our algorithm at each intermediate step to bound the difference in costs; the sum of these costs over the entire transformation then gives us a bound on the difference in costs between

OPT and CHC .

Application to smart energy systems

In Chapter 5 we focus on *direct load control* with the goal of using demand response to *reduce variations of the aggregate load*. This objective has been studied frequently in the literature, e.g., [30, 77], because reducing the variations of the aggregate load corresponds to minimizing the generation cost of the utilities. In particular, large generators with the smallest marginal costs, e.g., nuclear generators and hydro generators, have limited ramp rates, i.e., their power output cannot be adjusted too quickly. So, if load varies frequently, then it must be balanced by more expensive generators (i.e., “peakers”) that have fast ramp rate. Thus, if the load variation is reduced, then the utility can use the least expensive sources of power generation to satisfy the electricity demand.

Using the general prediction model introduced in Chapter 2, we introduce a model predictive algorithm for deferrable load control with uncertainty (Section 5.3). We perform a detailed performance analysis of our proposed algorithm. The performance analysis uses both analytic results and trace-based experiments to study (i) the reduction in expected load variance achieved via deferrable load control, and (ii) the value of using model predictive control via our algorithm when compared with static (open-loop) control. *The theorems in Section 5.4 characterize the impact of prediction inaccuracy on deferrable load control.* These analytic results highlight that as the time horizon expands, the expected load variance obtained by our proposed algorithm approaches the optimal value (Corollary 5.9). Also, as the time horizon expands, the algorithm obtains an increasing variance reduction over the optimal static algorithm (Corollary 5.12, 5.13). Furthermore, in Section 5.5 we provide trace-based experiments using data from Southern California Edison and Alberta Electric System Operator to validate the analytic results. These experiments highlight that our proposed algorithm obtains a small suboptimality under high uncertainties of renewable generation, and has significant performance improvement over the optimal static control.

Chapter 2

GENERAL PREDICTION MODEL

In this chapter, we discuss some of the shortcomings of the adversarial models for prediction error, and introduce a practical model for prediction error that generalizes models in the filtering and statistical learning community.

As OCO algorithms make their way into networking and cloud computing applications, it is increasingly clear that there is a mismatch between the pessimistic results provided by the theoretical analysis (which is typically adversarial) and the near-optimal performance observed in practice. Concretely, two main performance metrics have been studied in the literature: *regret*, defined as the difference between the cost of the algorithm and the cost of the offline optimal static solution, and the *competitive ratio*, defined as the maximum ratio between the cost of the algorithm and the cost of the offline optimal (dynamic) solution.

Within the *machine learning community*, regret has been heavily studied [40, 86, 90] and there are many simple algorithms that provide provably sublinear regret (also called "no regret"). For example, online gradient descent achieves $O(\sqrt{T})$-regret [90], even when there are switching costs [9]. In contrast, the *online algorithms community* considers a more general class of problems called "metrical task systems" (MTS) and focuses on competitive ratio [12, 13, 58]. Most results in this literature are "negative", e.g., when c_t are arbitrary, the competitive ratio grows without bound as the number of states in the decision space grows [13]. Exceptions to such negative results come only when structure is imposed on either the cost functions or the decision space, e.g., when the decision space is *one-dimensional* it is possible for an online algorithm to have a constant competitive ratio, e.g., [58]. However, *even in this simple setting no algorithms perform well for both competitive ratio and regret*. No online algorithm can have sublinear regret and a constant competitive ratio, even if the decision space is one-dimensional and cost functions are linear [9].

In contrast to the pessimism of the analytic work, applications in networking and cloud computing have shown that OCO algorithms can significantly outperform the static optimum while nearly matching the performance of the dynamic optimal, i.e., they simultaneously do well for regret and the competitive ratio. Examples include dynamic capacity management of a single data center [7, 58] and geographical load balancing across multiple data centers [56, 66, 69].

It is tempting to attribute this discrepancy to the fact that practical workloads are not adversarial. However, a more important factor is that, in reality, *algorithms can exploit relatively accurate predictions about the future*, such as diurnal variations [10, 35, 59]. But a more important contrast between the theory and application is simply that, in reality, *predictions about the future are available and accurate, and thus play a crucial role in the algorithms*. In this chapter, we argue that this mismatch is due to the pessimistic assumption on prediction errors, and propose a general model for practical predictions. Using this model, we show in Chapter 3 that there is a simple online algorithm that achives simultaneous sublinear regret and constant competitive ratio.

2.1 Problem formulation and notation

Throughout this thesis we consider online convex optimization problems with switching costs, i.e., "smoothed" online convex optimization (SOCO) problems.

The standard formulation of an online optimization problem with switching costs considers a convex decision/action space $G \subset \mathbb{R}^n$ and a sequence of cost functions $\{c_1, c_2, \ldots\}$, where each $c_t : G \to \mathbb{R}^+$ is convex. At time t, the online algorithm first chooses an action, which is a vector $x_t \in G$, the environment chooses a cost function c_t from a set C, and the algorithm pays a stage cost $c_t(x_t)$ and a switching cost $\beta||x_t - x_{t-1}||$ where $\beta \in (\mathbb{R}^+)$. Thus, the total cost of the online algorithm is defined to be

$$\text{cost}(ALG) = \mathbb{E}_x \left[\sum_{t=1}^{T} c_t(x_t) + \beta||x_t - x_{t-1}|| \right], \tag{2.1}$$

where $x_1, \ldots, x_T$ are the actions chosen by the algorithm, ALG. Without loss of generality, assume the initial action $x_0 = 0$, the expectation is over any randomness used by the algorithm, and $|| \cdot ||$ is a seminorm on $\mathbb{R}^n$.

Typically, a number of assumptions about the action space G and the cost functions c_t are made to allow positive results to be derived. In particular, the action set G is often assumed to be closed, nonempty, and bounded, where by bounded we mean that there exists $D \in \mathbb{R}$ such that for all $x, y \in G, ||x - y|| \leq D$. Further, the cost functions c_t are assumed to have a uniformly bounded subgradient, i.e., there exists $N \in \mathbb{R}^+$ such that, for all $x \in G$, $||\nabla c_t(x)|| \leq N$.

Performance Metrics

The performance of online algorithms for SOCO problems is typically evaluated via two performance metrics: *regret* and the *competitive ratio*. Regret is the dominant choice in the machine learning community and competitive ratio is the dominant choice in the online algorithms community. The key difference between these measures is whether they compare the performance of the online algorithm to the offline optimal static solution or the offline optimal dynamic solution. Specifically, the optimal offline *static* solution, is[1]

$$STA = \operatorname*{argmin}_{x \in G} \sum_{t=1}^{T} c_t(x) + \beta||x||, \tag{2.2}$$

and the optimal *dynamic* solution is

$$OPT = \operatorname*{argmin}_{(x_1,\ldots,x_T)\in G^T} \sum_{t=1}^{T} c_t(x_t) + \beta||(x_t - x_{t-1})||. \tag{2.3}$$

Definition 1. *The* ***regret*** *of an online algorithm, ALG, is less than $\rho(T)$ if the following holds:*

$$\sup_{(c_1,\ldots,c_T)\in C^T} \text{cost}(ALG) - \text{cost}(STA) \le \rho(T). \tag{2.4}$$

Definition 2. *An online algorithm ALG is said to be $\rho(T)$-**competitive** if the following holds:*

$$\sup_{(c^1,\ldots,c^T)\in C^T} \frac{\text{cost}(ALG)}{\text{cost}(OPT)} \le \rho(T). \tag{2.5}$$

The goals are typically to find algorithms with a (small) constant competitive ratio ("constant-competitive") and to find online algorithms with sublinear regret, i.e., an algorithm ALG that has regret $\rho(T)$ bounded above by some $\hat{\rho}(T) \in o(T)$; note that $\rho(T)$ may be negative if the concept we seek to learn varies dynamically. Sublinear regret is also called "no-regret", since the time-average loss of the online algorithm goes to zero as T grows.

Background

To this point, there are large literatures studying both the designs of no-regret algorithms and the design of constant-competitive algorithms. However, in general, these results tell a pessimistic story.

[1] One switching cost is incurred due to the fact that we enforce $x_0 = 0$.

In particular, on a positive note, it is possible to design simple, no-regret algorithms, e.g., *online gradient descent* (OGD) based algorithms [40, 90] and *Online Newton Step and Follow the Approximate Leader* algorithms [40]. (Note that the classical setting does not consider switching costs; however, [9] shows that similar regret bounds can be obtained when switching costs are considered.)

However, when one considers the competitive ratio, results are much less optimistic. Historically, results about competitive ratio have considered weaker assumptions, i.e., the cost functions c_t and the action set G can be nonconvex, and the switching cost is an arbitrary metric $d(x_t, x_{t-1})$ rather than a seminorm $||x_t - x_{t-1}||$. The weakened assumptions, together with the tougher offline target for comparison, leads to the fact that most results are "negative". For example, [13] has shown that any deterministic algorithm must be $\Omega(n)$-competitive given metric decision space of size n. Furthermore, [12] has shown that any randomized algorithm must be $\Omega(\sqrt{\log n/\log\log n})$-competitive. To this point, positive results are only known in very special cases. For example, [58] shows that, when G is a one-dimensional normed space, there exists a deterministic online algorithm that is 3-competitive.

Results become even more pessimistic when one asks for algorithms that perform well for both competitive ratio and regret. Note that performing well for both measures is particularly desirable for many networking and cloud computing applications where it is necessary to both argue that a dynamic control algorithm provides benefits over a static control algorithm (sublinear regret) and is near optimal (competitive ratio). However, a recent result in [9] highlights that such as goal is impossible: even when the setting is restricted to a one-dimensional normed space with linear cost functions no online algorithm can simultaneously achieve sublinear regret and constant competitive ratio [2].

2.2 Impossibility results in adversarial prediction models

The adversarial model underlying most prior work on online convex optimization has led to results that tend to be pessimistic; however, in reality, *algorithms can often use predictions about future cost functions in order to perform well.*

Knowing information about future cost functions is clearly valuable for smoothed online convex optimization problems, since it allows one to better justify whether

[2] Note that this impossibility is not the result of the regret being additive and the competitive ratio being multiplicative, as [9] proves the parallel result for competitive difference, which is an additive comparison to the dynamic optimal.

it is worth it to incur a switching cost during the current stage. Thus, it is not surprising that predictions have proven valuable in practice for such problems.

Given the value of predictions in practice, it is not surprising that there have been numerous attempts to incorporate models of prediction error into the analysis of online algorithms. We briefly expand upon the worst-case and stochastic approaches described in the introduction to motivate our approach, which is an integration of the two.

Worst-case models. Worst-case models of prediction error tend to assume that there exists a lookahead window, w, such that within that window, a perfect (or near-perfect, e.g., error bounded by ε) prediction is available. Then, outside of that window the workload is adversarial. A specific example is that, for any t the online algorithm knows $y_t, \ldots, y_{t+w}$ precisely, while $y_{t+w+1}, \ldots$ are adversarial.

Clearly, such models are both *too optimistic* about the the predictions used and *too pessimistic* about what is outside the prediction window. The result is that algorithms designed using such models tend to be too trusting of short term predictions and too wary of unknown fluctuations outside of the prediction window. Further, such models tend to underestimate the value of predictions for algorithm performance. To illustrate this, we establish the following theorem.

Theorem 2.1. *For any constant $\gamma > 0$ and any online algorithm A (deterministic or randomized) with constant lookahead w, either the competitive ratio of the algorithm is at least γ or its regret, is $\Omega(T)$. Here T is the number of cost functions in an instance.*

The above theorem focuses on the "easiest" worst-case model, i.e., where the algorithm is allowed perfect lookahead for w steps. Even in this case, an online algorithm must have super-constant lookahead in order to simultaneously have sublinear regret and a constant competitive ratio. Further, the proof (given in Appendix 2.A) highlights that this holds even in the scalar setting with linear cost functions. Thus, worst-case models are *overly pessimistic* about the value of prediction.

Stochastic models. Stochastic models tend to come in two forms: (i) i.i.d. models or (ii) detailed models of stochastic processes and specific predictors for those processes.

In the first case, for reasons of tractability, prediction errors are simply assumed to be i.i.d. mean zero random variables. While such an assumption is clearly analytically appealing, it is also quite simplistic and ignores many important features of prediction errors. For example, in reality, predictions have increasing error the further in time we look ahead due to correlation of predictions errors in nearby points in time. Further, predictions tend to be updated or refined as time passes. These fundamental characteristics of predictions cannot be captured by the i.i.d. model.

In the second case, which is common in control theory, a specific stochastic model for the underlying process is assumed and then an optimal predictor (filter) is derived. Examples here include the derivation of Weiner filters and Kalaman filters for the prediction of wide-sense stationary processes and linear dynamical systems respectively, see [46]. While such approaches avoid the pessimism of the worst-case viewpoint, they instead tend to be fragile to the underlying modeling assumptions. In particular, an online algorithm designed to use a particular filter based on a particular stochastic model lacks the robustness to be used in settings where the underlying assumptions are not valid.

2.3 Modelling practical prediction errors

A key contribution of this thesis is the development of a model for studying predictions that provides a middle ground between the worst-case and the stochastic viewpoints. The model we propose below seeks a middle ground by not making any assumption on the underlying stochastic process or the design of the predictor, but instead making assumptions only on the form of the error of the predictor. Thus, it is agnostic to the predictor and can be used in worst-case analysis with respect to the realization of the underlying cost functions.

Further, the model captures three important features of real predictors: (i) it allows for correlations in prediction errors (both short range and long range); (ii) the quality of predictions decreases the further in the future we try to look ahead; and (iii) predictions about the future are refined as time passes.

Concretely, throughout this thesis we model prediction error via the following equation:

$$y_t = y_{t|\tau} + \sum_{s=\tau+1}^{t} f(t-s)e(s). \tag{2.6}$$

Here, $y_{t|\tau}$ is the prediction of y_t made at time $\tau < t$. Thus, $y_t - y_{t|\tau}$ is the prediction error, and is specified by the summation in (2.6). In particular, the prediction error is modeled as a weighted linear combination of per-step noise terms, $e(s)$ with weights

$f(t-s)$ for some deterministic impulse function f. The key assumptions of the model are that $e(s)$ are i.i.d. with mean zero and positive definite covariance R_e; and that f satisfies $f(0) = I$ and $f(t) = 0$ for $t < 0$. Note that, as the examples below illustrate, it is common for the impulse function to decay as $f(s) \sim 1/s^{\alpha}$. As we will see later, this simple model is flexible enough to capture the prediction error that arise from classical filters on time series, and it can represent all forms of stationary prediction error by using appropriate forms of f.

Some intuition for the form of the model can be obtained by expanding the summation in (2.6). In particular, note that for $\tau = t-1$ we have

$$y_t - y_{t|t-1} = f(0)e(t) = e(t), \tag{2.7}$$

which highlights why we refer to $e(t)$ as the per-step noise.

Further, expanding the summation further gives

$$y_t - y_{t|\tau} = f(0)e(t) + f(1)e(t-1) + \ldots + f(t-\tau-1)e(\tau+1). \tag{2.8}$$

Note that the first term is the one-step prediction error $y_t - y_{t|t-1}$; the first two terms make up the two-step prediction error $y_t - y_{t|t-2}$; and so on. This highlights that predictions in the model have increasing noise as one looks further ahead in time and that predictions are refined as time goes forward.

Additionally, note that the form of (2.8) highlights that the impulse function f captures the degree of short-term/long-term correlation in prediction errors. Specifically, the form of $f(t)$ determines how important the error t steps in the past is for the prediction. Since we assume no structural form for f, complex correlation structures are possible.

Finally, unraveling the summation all the way to time zero highlights that the process y_t can be viewed as a random deviation around the predictions made at time zero, $y_{t|0} := \hat{y}_t$, which are specified externally to the model:

$$y_t = \hat{y}_t + \sum_{s=1}^{t} f(t-s)e(s). \tag{2.9}$$

This highlights that an instance of the online convex optimization problem can be specified via either the process y_t or via the initial predictions $\hat{y}_t$, and then the random noise from the model determines the other. We discuss this more when defining the notions of regret and competitive ratio we study in Section 4.1.

Examples

While the form of the prediction error in the model may seem mysterious, it is quite general, and includes many of the traditional models as special cases. For example, to recover the worst-case prediction model one can set, $\forall t, e(t) = 0$ and $\hat{y}_{t'}$ as unknown $\forall t' > t + w$ and then take the worst case over $\hat{y}$. Similarly, a common approach in robust control is to set $f(t) = \begin{cases} I, & t = 0; \\ 0, & t \neq 0 \end{cases}$, $|e(s)| < D, \forall s$ and then consider the worst case over e.

Additionally, strong motivation for it can be obtained by studying the predictors for common stochastic processes. In particular, the form of (2.6) matches the prediction error of standard Weiner filters [84] and Kalman filters [48], etc. To highlight this, we include a few brief examples below.

Example 2.2 (Wiener Filter)**.** *Let $\{y_t\}_{t=0}^T$ be a wide-sense stationary stochastic process with $\mathbb{E}[y_t] = \hat{y}_t$, and covariance $\mathbb{E}[(y_i - \hat{y}_i)(y_j - \hat{y}_j)^T] = R_y(i - j)$, i.e., the covariance matrix $R_y > 0$ of $y = [y_1\ y_2\ \dots y_T]^T$ is a Toeplitz matrix. The corresponding $e(s)$ in the Wiener filter for the process is called the "innovation process" and can be computed via the Wiener-Hopf technique [46]. Using the innovation process $e(s)$, the optimal causal linear prediction is*

$$y_{t|\tau} = \hat{y}_t + \sum_{s=1}^{\tau} \langle y_t, e(s) \rangle ||e(s)||^{-2} e(s),$$

and so the correlation function $f(s)$ as defined in (2.6) *is*

$$f(s) = \langle y_s, e(0) \rangle ||e(0)||^{-2} = R_y(s) R_e^{-1}. \tag{2.10}$$

Example 2.3 (Kalman Filter)**.** *Consider a stationary dynamical system described by the hidden state space model*

$$x'_{t+1} = Ax'_t + Bu_t, \quad y_t = Cx'_t + v_t,$$

where the $\{u_t, v_t, x_0\}$ are $m \times 1, p \times 1,$ and $n \times 1$-dimensional random variables such that

$$\left\langle \begin{bmatrix} u_i \\ v_i \\ x_0 \end{bmatrix}, \begin{bmatrix} u_j \\ v_j \\ x_0 \\ 1 \end{bmatrix} \right\rangle = \begin{bmatrix} Q\delta_{ij} & S\delta_{ij} & 0 & 0 \\ S^*\delta_{ij} & R\delta_{ij} & 0 & 0 \\ 0 & 0 & \Pi_0 & 0 \end{bmatrix}.$$

The Kalman filter for this process yields the optimal causal linear estimator $y_{t|\tau} = K[y_1^T, \dots, y_\tau^T]^T$ such that $y_{t|\tau} = \arg\min \mathbb{E}_\tau ||y_t - K'[y_1^T, \dots, y_\tau^T]^T||^2$. When t is large *and the system reaches steady state, the optimal prediction is given in the following recursive form [46]:*

$$
\begin{aligned}
x'_{t+1|t} &= Ax'_{t|t-1} + K_p e(t),\ y_{0|-1} = 0,\ e(0) = y_0, \\
e(t) &= y_t - Cx'_{t|t-1},
\end{aligned}
$$

where $K_p = (APC^* + BS)R_e^{-1}$, is the Kalman gain, and $R_e = R + CPC^*$ is the covariance of the innovation process e_t, and P solves

$$P = APA^* + CQC^* - K_p R_e K_p^*.$$

This yields the predictions

$$
\begin{aligned}
y_{t|\tau} &= \sum_{s=1}^{\tau} \langle y_t, e(s) \rangle R_e^{-1} e(s) \\
&= \sum_{s=1}^{\tau} CA^{t-s-1}(APC^* + BS)R_e^{-1} e(s).
\end{aligned}
$$

Thus, for stationary Kalman filter, the prediction error correlation function is

$$f(s) = CA^{s-1}(APC^* + BS)R_e^{-1} = CA^{s-1}K_p. \tag{2.11}$$

Performance metrics under general prediction model

A key feature of the prediction model described above is that it provides a general stochastic model for prediction errors while not imposing any particular underlying stochastic process or predictor. Thus, it generalizes a variety of stochastic models while allowing worst-case analysis.

More specifically, when studying online algorithms using the prediction model above, one could either specify the instance via y_t and then use the form of (2.6) to give random predictions about the instance to the algorithm or, one could specify the instance using $\hat{y} := y_{t|0}$ and then let the y_t be randomly revealed using the form of (2.9). Note that, of the two interpretations, the second is preferable for analysis, and thus we state our theorems using it.

In particular, our setup can be interpreted as allowing an adversary to specify the instance via the initial (time 0) predictions $\hat{y}$, and then using the prediction error model to determine the instance y_t. We then take the worst-case over $\hat{y}$. This corresponds to having an adversary with a "shaky hand" or, alternatively, letting the adversary specify the instance but forcing them to also provide unbiased initial predictions.

In this context, we study the following notions of (expected) regret and (expected) competitive ratio, where the expectation is over the realization of the prediction noise e and the measures consider the worst-case specification of the instance $\hat{y}$.

Definition 3. *We say an online algorithm ALG, has (expected)* ***regret*** *at most $\rho(T)$ if*

$$\sup_{\hat{y}} \mathbb{E}_e[\text{cost}(ALG) - \text{cost}(STA)] \leq \rho(T). \tag{2.12}$$

Definition 4. *We say an online algorithm ALG is $\rho(T)$**-competitive** (in expectation) if*

$$\sup_{\hat{y}} \frac{\mathbb{E}_e[\text{cost}(ALG)]}{\mathbb{E}_e[\text{cost}(OPT)]} \leq \rho(T). \tag{2.13}$$

Using the general model for predictions, we show in the next Chapter that, there exists online algorithm that can simultaneously achieve sublinear regret and constant competitive ratio. Overcoming the hardness result for adversarial analysis (Theorem 2.1).

2.A Proofs for Section 2.2

Proof for Theorem 2.1. For a contradiction, assume that there exists an algorithm $\mathcal{A}'$ that achieves constant competitive ratio *and* sublinear regret with constant lookahead. We can use algorithm $\mathcal{A}'$ to obtain another online algorithm $\mathcal{A}$ that achieves constant competitive ratio *and* sublinear regret *without* lookahead. This contradicts Theorem 4 of [9], and we get the claim.

Consider an instance $\{c_1, c_2, \ldots, c_T\}$ without lookahead. We simply "pad" the input with ℓ copies of the zero function $\mathbf{0}$ if $\mathcal{A}'$ has a lookahead of ℓ. That is, the input to $\mathcal{A}'$ is: $c_1, \mathbf{0}, \ldots, \mathbf{0}, c_2, \mathbf{0}, \ldots, \mathbf{0}, c_3, \mathbf{0}, \ldots$

We simulate $\mathcal{A}'$ and set the tth action of $\mathcal{A}$ equal to the $((t-1)(\ell+1)+1)$th action of $\mathcal{A}'$. Note that the optimal values of the padded instance are equal to the optimal values of the given instance. Also, by construction, $\text{cost}(\mathcal{A}) \leq \text{cost}(\mathcal{A}')$. Therefore, if $\mathcal{A}'$ achieves constant competitive ratio *and* sublinear regret then so does $\mathcal{A}$, and the claim follows. $\square$

Chapter 3

THE VALUE OF PREDICTION

In Chapter 2, we have shown that in the adversarial setting, no matter far we can lookahead, no algorithm can achieve good performance in terms of regret and competitive ratio simultaneously. This motivates us to propose a new model for prediction error

Since our focus in this thesis is on predictions, we consider a variation of the above with *parameterized* cost functions $c_t(x_t; y_t)$, where the parameter y_t is the focus of prediction. Further, except when considering worst-case predictions, we adopt a specific form of c_t for concreteness. We focus on a tracking problem where the online algorithm is trying to do a "smooth" tracking of y_t and pays a least square penalty each round.

$$\text{cost}(ALG) = \mathbb{E}_x \left[\sum_{t=1}^{T} \frac{1}{2} ||y_t - Kx_t||_2^2 + \beta ||x_t - x_{t-1}||_1. \right], \tag{3.1}$$

where the target $y_t \in \mathbb{R}^m$, and $K \in \mathbb{R}^{m \times n}$ is a (known) linear map that transforms the control variable into the space of the tracking target. Let $K^\dagger$ be the Moore-Penrose pseudoinverse of K.

We focus on this form because it represents an online version of the LASSO (Least Absolute Shrinkage and Selection Operator) formulation, which is widely studied in a variety of contexts, e.g., see [20, 21, 26, 81] and the references therein. Typically in LASSO the one-norm regularizer is used to induce sparsity in the solution. In our case, this corresponds to specifying that a good solution does not change too much, i.e., $x_t - x_{t-1} \neq 0$ is infrequent. Importantly, the focus on LASSO, i.e., the two-norm loss function and one-norm regularizer, is simply for concreteness and ease of presentation. Our proof technique generalizes (at the expense of length and complexity).

We assume that $K^T K$ is invertible and that the static optimal solution to (3.1) is positive. Neither of these is particularly restrictive. If K has full column rank then $K^T K$ is invertible. This is a reasonable, for example, when the dimensionality of the action space G is small relative to the output space. Note that typically K is designed, and so it can be chosen to ensure these assumptions are satisfied. Additionally if K is invertible, then it no longer appears in the results provided.

Finally, it is important to highlight a few contrasts between the cost function in (3.1) and the typical assumptions in the online convex optimization literature. First, note that the feasible action set $G = \mathbb{R}^n$ is unbounded. Second, note that gradient of c_t can be arbitrarily large when y_t and Kx_t are far apart. Thus, both of these are relaxed compared to what is typically studied in the online convex optimization literature. We show in Section 3.2 that, we can have sublinear regret even in this relaxed setting.

Recall that we assume the prediction model in (2.6):

$$y_t = y_{t|\tau} + \sum_{s=\tau+1}^{t} f(t-s)e(s).$$

Naturally, the form of the correlation structure plays a crucial role in the performance results we prove. But, the detailed structure is not important, only its effect on the aggregate variance. Specifically, the impact of the correlation structure on performance is captured through the following two definitions, which play a prominent role in our analysis. First, for any $w > 0$, let $||f_w||^2$ be the two norm of prediction error covariance over $(w+1)$ steps of prediction, i.e.,

$$||f_w||^2 = \mathrm{tr}(\mathbb{E}[\delta y_w \delta y_w^T]) = \mathrm{tr}(R_e \sum_{s=0}^{w} f(s)^T f(s)), \tag{3.2}$$

where $\delta y_w^T = y_{t+w} - y_{t+w|t-1} = \sum_{s=t}^{t+w} f(t+w-s)e(s)$. The derivation of (3.2) is found in the proof of Theorem 3.4.

Second, let $F(w)$ be the two norm square of the projected cumulative prediction error covariance, i.e.,

$$F(w) = \sum_{t=0}^{w} \mathbb{E}||KK^{\dagger}\delta y_w||^2 = \mathrm{tr}(R_e \sum_{s=0}^{w}(w-s+1)f(s)^T KK^{\dagger} f(s)). \tag{3.3}$$

Note that $KK^{\dagger}$ is the orthogonal projector onto the range space of K. Therefore it is natural that the definitions are over the induced norm of $KK^{\dagger}$ since any action chosen from the space F can only be mapped to the range space of K i.e. no algorithm, online or offline, can track the portion of y that falls in the null space of K.

Examples

1. For Wierner filter shown in Example 2.2, we have

$$||f_w||^2 = \frac{1}{R_e}\sum_{s=0}^{w} R_y(s)^2 \text{ and } F(w) = \frac{1}{R_e}\sum_{s=0}^{w}(w-s+1)R_y(s)^2.$$

2. For the Kalman filter shown in Example 2.3, we have

$$||f_w||^2 = \sum_{s=0}^{w} \mathrm{tr}(R_e(CA^{s-1}K_p)^T KK^\dagger(CA^{s-1}K_p)) \text{ and}$$
$$F(w) = \sum_{s=0}^{w}(w-s+1)\mathrm{tr}(R_e(CA^{s-1}K_p)^T KK^\dagger(CA^{s-1}K_p)).$$

Our proofs bound the competitive ratio through an analysis of the competitive difference, which is defined as follows.

Definition 5. *We say an online algorithm ALG has (expected)* ***competitive difference*** *at most $\rho(T)$ if*

$$\sup_{\hat{y}} \mathbb{E}_e\left[\mathrm{cost}(ALG) - \mathrm{cost}(OPT)\right] \le \rho(T). \tag{3.4}$$

Note that these expectations are with respect to the prediction noise, $(e(t))_{t=1}^{T}$, and so $\mathrm{cost}(OPT)$ is also random. Note also that when $\mathrm{cost}(OPT) \in \Omega(\rho(T))$ and ALG has competitive difference at most $\rho(T)$, then the algorithm has a constant (bounded) competitive ratio.

3.1 Averaging fixed horizon control

A wide variety of algorithms have been proposed for online convex optimization problems. Given the focus of this thesis on predictions, the most natural choice of an algorithm to consider is Receding Horizon Control (RHC), a.k.a., Model Predictive Control (MPC).

There is a large literature in control theory that studies RHC/MPC algorithms, e.g., [33, 63] and the references therein; and thus RHC is a popular choice for online optimization problems when predictions are available, e.g., [11, 18, 50, 83]. However, recent results have highlighted that while RHC can perform well for one-dimensional smoothed online optimization problems, it does not perform well (in the worst case) outside of the one-dimension case. Specifically, the competitive ratio of RHC with perfect lookahead w is $1 + O(1/w)$ in the one-dimensional setting, but is $1 + \Omega(1)$ outside of this setting, i.e., the competitive ratio does not decrease to 1 as the prediction window w increases [56].

In contrast, a promising new algorithm, Averaging Fixed Horizon Control (AFHC), proposed by [56] in the context of geographical load balancing, maintains good performance in high-dimensional settings, i.e., maintains a competitive ratio of

$1 + O(1/w)$[1]. Thus, in this chapter, we focus on AFHC. Our results highlight that AFHC extracts the asymptotically optimal value from predictions, and so validates this choice.

As the name implies, AFHC averages the choices made by Fixed Horizon Control (FHC) algorithms. In particular, AFHC with prediction window size $(w+1)$ averages the actions of $(w+1)$ FHC algorithms.

Algorithm 3.1 Fixed Horizon Control

Let $\Omega_k = \{i : i \equiv k \mod (w+1)\} \cap [-w, T]$ for $k = 0, \dots, w$. Then $FHC^{(k)}(w+1)$, the kth FHC algorithm is defined in the following manner. At timeslot $\tau \in \Omega_k$ (i.e., before c_τ is revealed), choose actions $x^{(k)}_{FHC,t}$ for $t = \tau, \dots \tau + w$ as follows: If $t \leq 0$, $x^{(k)}_{FHC,t} = 0$. Otherwise, let $x_{\tau-1} = x^{(k)}_{FHC,\tau-1}$, and let $(x^{(k)}_{FHC,t})_{t=\tau}^{\tau+w}$ be the vector that solves

$$\min_{x_\tau,\dots,x_{\tau+w}} \sum_{t=\tau}^{\tau+w} \hat{c}_t(x_t) + \beta||(x_t - x_{t-1})||,$$

where $\hat{c}_t(\cdot)$ is the prediction of the future cost $c_t(\cdot)$ for $t = \tau, \dots, \tau + w$.

Note that in the classical OCO with $(w+1)$-lookahead setting, $\hat{c}_t(\cdot)$ is exactly equal to the true cost $c(\cdot)$. Each FHC$^{(k)}(w+1)$ can be seen as a length $(w+1)$ fixed horizon control starting at position k. Given $(w+1)$ versions of FHC, AFHC is defined as the following:

Algorithm 3.2 Averaging Fixed Horizon Control

At timeslot $t \in 1, ..., T$, $AFHC(w+1)$ sets

$$x_{AFHC,t} = \frac{1}{w+1} \sum_{k=0}^{w} x^{(k)}_{FHC,t}. \tag{3.5}$$

3.2 Average-case analysis

We first consider the average-case performance of AFHC (in this section), and then consider distributional analysis (in Section 3.3). We focus on the tracking problem in (3.1) for concreteness and conciseness, though our proof techniques generalize. Note that, unless otherwise specified, we use $|| \cdot || = || \cdot ||_2$.

[1] Note that this result assumes that the action set is bounded, i.e., for all feasible action x, y, there exists $D > 0$, such that $||x - y|| < D$, and that there exists $e_0 > 0$, s.t. $c_t(0) \geq e_0, \forall t$. The results we prove in this chapter make neither of these assumptions.

Our main result shows that AFHC can simultaneously achieve sublinear regret and a constant competitive ratio using only a *constant-sized* prediction window in nearly all cases that it is feasible for an online algorithm to do so. This is in stark contrast with Theorem 2.1 for the worst-case prediction model.

Theorem 3.1. *Let w be a constant. AFHC($w+1$) is constant-competitive whenever* $\inf_{\hat{y}} \mathbb{E}_e[OPT] = \Omega(T)$ *and has sublinear regret whenever* $\inf_{\hat{y}} \mathbb{E}_e[STA] \geq \alpha_1 T - o(T)$*, for* $\alpha_1 = 4V + 8B^2$*, where*

$$V = \frac{\beta||K^{\dagger}||_1||f_w|| + 3\beta^2||(K^T K)^{-1}\mathbb{1}|| + F(w)/2}{w+1} \tag{3.6}$$

$$B = \beta||(K^T)^{\dagger}\mathbb{1}||, \tag{3.7}$$

and $||M||_1$ denotes the induced 1-norm of a matrix M.

Theorem 3.1 imposes bounds on the expected costs of the dynamic and static optimal in order to guarantee a constant competitive ratio and sublinear regret. These bounds come about as a result of the noise in predictions. In particular, prediction noise makes it impossible for an online algorithm to achieve sublinear expected cost, and thus makes it infeasible for an online algorithm to compete with dynamic and static optimal solutions that perform too well. This is made formal in Theorems 3.2 and 3.3, which are proven in Appendix 3.A. Recall that R_e is the covariance of an estimation error vector, $e(t)$.

Theorem 3.2. *Any online algorithm ALG that chooses x_t using only (i) internal randomness independent of $e(\cdot)$ and (ii) predictions made up until time t, has expected cost* $\mathbb{E}_e[\text{cost}(ALG)] \geq \alpha_2 T + o(T)$*, where* $\alpha_2 = \frac{1}{2}||R_e^{1/2}||^2_{KK^{\dagger}}$.

Theorem 3.3. *Consider an online algorithm ALG such that* $\mathbb{E}_e[\text{cost}(ALG)] \in o(T)$*. The actions, x_t, of ALG can be used to produce one-step predictions $y'_{t|t-1}$, such that mean square of the one-step prediction error is smaller than that for $y_{t|t-1}$, i.e.,* $\mathbb{E}_e||y_t - y'_{t|t-1}||^2 \leq \mathbb{E}_e||y_t - y_{t|t-1}||^2$*, for all but sublinearly many t.*

Theorem 3.2 implies that it is impossible for any online algorithm that uses extra information (e.g., randomness) independent of the prediction noise to be constant competitive if $\mathbb{E}_e[\text{cost}(OPT)] = o(T)$ or to have sublinear regret if $\mathbb{E}_e[\text{cost}(STA)] \leq (\alpha_2 - \varepsilon)T + o(T)$, for $\varepsilon > 0$.

Further, Theorem 3.3 states that if an online algorithm does somehow obtain asymptotically smaller cost than possible using only randomness independent of the prediction error, then it must be using more information about future y_t than is available from the predictions. This means that the algorithm can be used to build a better predictor.

Thus, the consequence of Theorems 3.2 and 3.3 is the observation that the condition in Theorem 3.1 for the competitive ratio is tight and the condition in Theorem 3.1 for regret is tight up to a constant factor, i.e., α_1 versus α_2. (Attempting to prove matching bounds here is an interesting, but very challenging, open question.)

In the remainder of the section, we outline the analysis needed to obtain Theorem 3.1, which is proven by combining Theorem 3.4 bounding the competitive difference of AFHC and Theorem 3.8 bounding the regret of AFHC. The analysis exposes the importance of the correlation in prediction errors for tasks such as determining the optimal prediction window size for AFHC. Specifically, the window size that minimizes the performance bounds we derive is determined not by the quality of predictions, but rather by how quickly error correlates, i.e., by $||f_\omega||^2$.

Proof of Theorem 3.1

The first step in our proof of Theorem 3.1 is to bound the competitive difference of AFHC. This immediately yields a bound on the competitive ratio and, since it is additive, it can easily be adapted to bound regret as well.

The main result in our analysis of competitive difference is the following. This is the key both to bounding the competitive ratio and regret.

Theorem 3.4. *The competitive difference of $AFHC(w+1)$ is $O(T)$ and bounded by:*

$$\sup_{\hat{y}} \mathbb{E}_e[\text{cost}(AFHC) - \text{cost}(OPT)] \leq VT, \tag{3.8}$$

where V is given by (3.6)

Theorem 3.4 implies that the competitive ratio of $AFHC$ is bounded by a constant when $\text{cost}(OPT) \in \Omega(T)$.

The following corollary of Theorem 3.4 is obtained by minimizing V with respect to w.

Corollary 3.5. *For AFHC, the prediction window size that minimizes the bound in Theorem 3.4 on competitive difference is a finite constant (independent of T) if $F(T) \in \omega(T)$ and is T if there is i.i.d noise*[2]*.*

[2] Specifically $f(0) = I, f(t) = 0 \forall t > 0$

The intuition behind this result is that if the prediction model causes noise to correlate rapidly, then a prediction for a time step too far into the future will be so noisy that it would be best to ignore it when choosing an action under AFHC. However, if the prediction model is nearly independent, then it is optimal for AFHC to look over the entire time horizon, T, since there is little risk from aggregating predictions. Importantly, notice that the quality (variance) of the predictions is not determinant, only the correlation.

Theorem 3.4 is proven using the following lemma (proven in the appendix) by taking expectation over noise.

Lemma 3.6. *The cost of $AFHC(w+1)$ for any realization of y_t satisfies*

$$\text{cost}(AFHC) - \text{cost}(OPT) \leq \frac{1}{w+1}\sum_{k=0}^{w}\sum_{\tau\in\Omega_k}\left(\beta||x^*_{\tau-1} - x^{(k)}_{\tau-1}||_1 + \sum_{t=\tau}^{\tau+w}\frac{1}{2}||y_t - y_{t|\tau-1}||^2_{KK^\dagger}\right).$$

Next, we use the analysis of the competitive difference in order to characterize the regret of AFHC. In particular, to bound the regret we simply need a bound on the gap between the dynamic and static optimal solutions.

Lemma 3.7. *The suboptimality of the offline static optimal solution STA can be bounded below on each sample path by*

$$\begin{aligned}&\text{cost}(STA) - \text{cost}(OPT)\\ \geq&\frac{1}{2}\left(\sqrt{\sum_{t=1}^{T}||y_t - \bar{y}||^2_{KK^\dagger}} - 2B\sqrt{T})\right)^2 - 2B^2T - C\end{aligned}$$

where $\bar{y} = \frac{\sum_{t=1}^{T} y_t}{T}$, $B = \beta||(K^T)^\dagger\mathbb{1}||_2$ *and* $C = \frac{\beta^2\mathbb{1}^T(K^TK)^{-1}\mathbb{1}}{2T}$.

Note that the bound above is in terms of $||(y_t - \bar{y})||^2_{KK^\dagger}$, which can be interpreted as a measure of the variability y_t. Specifically, it is the projection of the variation onto the range space of K.

Combining Theorem 3.4 with Lemma 3.7 gives a bound on the regret of AFHC, proven in Appendix 3.A.

Theorem 3.8. *$AHFC$ has sublinear expected regret if*

$$\inf_{\hat{y}} \mathbb{E}_e\sum_{t=1}^{T}||KK^\dagger(y_t - \bar{y})||^2 \geq (8V + 16B^2)T,$$

where V and B are defined in (3.6) *and* (3.7).

Finally, we make the observation that, for all instances of y:

$$\begin{aligned}
\text{cost}(STA) &= \frac{1}{2}\sum_{t=1}^{T} ||y_t - Kx||^2 + \beta||x||_1 \\
&\geq \frac{1}{2}\sum_{t=1}^{T} ||(I - KK^{\dagger})y_t + KK^{\dagger}y_t - Kx||^2 \\
&= \frac{1}{2}\sum_{t=1}^{T} ||(I - KK^{\dagger})y_t||^2 + \frac{1}{2}||KK^{\dagger}y_t - Kx||^2 \\
&\geq \frac{1}{2}||KK^{\dagger}(y_t - \bar{y})||^2.
\end{aligned}$$

Therefore, by Theorem 3.8, we have the condition of the Theorem.

3.3 Concentration bounds

The previous section shows that AFHC performs well in expectation, but it is also important to understand the distribution of the cost under AFHC. In this section, we show that, with a mild additional assumption on the prediction error $e(t)$, the event when there is a large deviation from the expected performance bound proven in Theorem 3.4 decays exponentially fast.

The intuitive idea behind the result is the observation that the competitive difference of AFHC is a function of the uncorrelated prediction error $e(1), \ldots, e(T)$ that does not put too much emphasis on any one of the random variables $e(t)$. This type of function normally has sharp concentration around its mean because the effect of each $e(t)$ tends to cancel out.

For simplicity of presentation, we state and prove the concentration result for AFHC for the one dimensional tracking cost function

$$\frac{1}{2}\sum_{t=1}^{T}(y_t - x_t)^2 + \beta|x_t - x_{t-1}|.$$

In this case, $R_e = \sigma^2$, and the correlation function $f : \mathbb{N} \to R$ is a scalar valued function. The results can all be generalized to the multidimensional setting.

Additionally, for simplicity of presentation, we assume (for this section only) that $\{e(t)\}_{t=1}^{T}$ are uniformly bounded, i.e., $\exists\epsilon > 0$, s.t. $\forall t$, $|e(t)| < \epsilon$. Note that, with additional effort, the boundedness assumption can be relaxed to the case of $e(t)$ being subgaussian, i.e., $\mathbb{E}[\exp(e(t)^2/\epsilon^2)] \leq 2$, for some $\epsilon > 0$.[3]

To state the theorem formally, let VT be the upper bound of the expected competitive difference of $AFHC$ in (3.8). Given $\{\hat{y}_t\}_{t=1}^T$, the competitive difference of $AFHC$ is a random variable that is a function of the prediction error $e(t)$. The following theorem shows that the probability that the cost of $AFHC$ exceeds that of OPT by much more than the expected value VT decays rapidly.

Theorem 3.9. *The probability that the competitive difference of AFHC exceeds VT is exponentially small, i.e., for any $u > 0$:*

$$\begin{aligned}&\mathbb{P}(\text{cost}(AFHC) - \text{cost}(OPT) > VT + u)\\ \leq &\exp\left(\frac{-u^2}{8\epsilon^2 \frac{\beta^2 T}{(w+1)\sigma^2}||f_w||^2}\right) + \exp\left(\frac{-u^2}{16\epsilon^2\lambda(2\frac{T}{w+1}F(w) + u)}\right)\\ \leq &2\exp\left(\frac{-u^2}{a+bu}\right),\end{aligned}$$

where $||f_w||^2 = (\sum_{t=0}^{w}|f(t)|^2)$, the parameter λ of concentration

$$\lambda \leq \sum_{t=0}^{w}(w-t)f(t)^2 = \frac{1}{\sigma^2}F(w),$$

and $a = 8\epsilon^2[T/(w+1)]\max(\frac{\beta^2}{\sigma^2}||f_w||^2, 4\lambda F(w))$, $b = 16\epsilon^2\lambda$.

The theorem implies that the tail of the competitive difference of $AFHC$ has a Bernstein type bound. The bound decays much faster than the normal large deviation bounds obtained by bounding moments, i.e., Markov Inequality or Chebyshev Inequality. This is done by more detailed analysis of the structure of the competitive difference of $AFHC$ as a function of $e = (e(1), \ldots, e(T))^T$.

Note that smaller values of a and b in Theorem 3.9 imply a sharper tail bound. We can see that smaller $||f_w||$ and smaller $F(w)$ implies the tail bound decays faster. Since higher prediction error correlation implies higher $||f_w||$ and $F(w)$, Theorem 3.9 quantifies the intuitive idea that the performance of AFHC concentrates more tightly around its mean when the prediction error is less correlated.

[3] This involves more computation and worse constants in the concentration bounds. Interested readers are referred to Theorem 12 and the following remark of [14] for a way to generalize the concentration bound for the switching cost (Lemma 3.10), and Theorem 1.1 of [72] for a way to generalize the concentration bound for prediction error (Lemma 3.14).

Proof of Theorem 3.9

To prove Theorem 3.9, we start by decomposing the bound in Lemma 3.6. In particular, Lemma 3.6 gives

$$\text{cost}(AFHC) - \text{cost}(OPT) \le g_1 + g_2, \tag{3.9}$$

where

$$g_1 = \frac{1}{w+1} \sum_{k=0}^{w} \sum_{\tau \in \Omega_k} \beta |x^*_{\tau-1} - x^{(k)}_{\tau-1}|$$

represents loss due to the switching cost, and

$$g_2 = \frac{1}{w+1} \sum_{k=0}^{w} \sum_{\tau \in \Omega_k} \sum_{t=\tau}^{\tau+w} \frac{1}{2}(y_t - y_{t|\tau-1})^2$$

represents the loss due to the prediction error.

Let $V_1 = \frac{3\beta^2 T}{w+1} + \frac{\beta T}{w+1}||f_w||_2$, and $V_2 = \frac{T}{2(w+1)}F(w)$. Note that $VT = V_1 + V_2$. Then, by (3.9),

$$\begin{aligned} &\mathbb{P}(\text{cost}(AFHC) - \text{cost}(OPT) > u + VT) \\ \le\ &\mathbb{P}(g_1 > u/2 + V_1 \text{ or } g_2 > u/2 + V_2) \\ \le\ &\mathbb{P}(g_1 > u/2 + V_1) + \mathbb{P}(g_2 > u/2 + V_2). \end{aligned} \tag{3.10}$$

Thus, it suffices to prove concentration bounds for the loss due to switching cost, g_1, and the loss due to prediction error, g_2, deviating from V_1 and V_2 respectively. This is done in the following. The idea is to first prove that g_1 and g_2 are functions of $e = (e(1), \ldots, e(T))^T$ that are not "too sensitive" to any of the elements of e, and then apply the method of bounded difference [64] and Log-Sobolev inequality [53]. Combining (3.10) with Lemmas 3.10 and 3.14 below will complete the proof of Theorem 3.9.

Bounding the loss due to switching cost This section establishes the following bound on the loss due to switching:

Lemma 3.10. *The loss due to switching cost has a sub-Gaussian tail: for any $u > 0$,*

$$\mathbb{P}(g_1 > u + V_1) \le \exp\left(\frac{-u^2}{2\epsilon^2\beta^2\frac{T}{w+1}(||f_w||)^2}\right). \tag{3.11}$$

To prove Lemma 3.10, we introduce two lemmas. Firstly, we use the first order optimality condition to bound g_1 above by a linear function of $e = (e(1), \ldots, e(T))^T$ using the following lemma proved in the Appendix.

Lemma 3.11. *The loss due to switching cost can be bounded above by*

$$g_1 \leq \frac{3\beta^2 T}{w+1} + \frac{\beta}{w+1} \sum_{k=0}^{w} \sum_{\tau \in \Omega_k} \left| \sum_{s=1 \vee (\tau - w - 2)}^{\tau - 1} f(\tau - 1 - s) e(s) \right|. \tag{3.12}$$

Let $g_1'(e)$ be the second term of g_1. Note that the only randomness in the upper bound (3.12) comes from g_1'.

Lemma 3.12. *The expectation of $g_1'(e)$ is bounded above by*

$$\mathbb{E}_e g_1'(e) \leq \frac{\beta T}{w+1} ||f_w||_2.$$

With Lemma 3.12, we can reduce (3.11) to proving a concentration bound on $g_1'(e)$, since

$$\mathbb{P}(g_1 > u + V_1) \leq \mathbb{P}(g_1' - \mathbb{E} g_1'(e) \leq u). \tag{3.13}$$

To prove concentration of $g_1'(e)$, which is a function of a collection of independent random variables, we use the method of bounded difference, i.e., we bound the difference of $g_1'(e)$ where one component of e is replaced by an identically-distributed copy. Specifically, we use the following lemma, the one-sided version of one due to McDiarmid:

Lemma 3.13 ([64], Lemma 1.2)**.** *Let $X = (X_1, \ldots, X_n)$ be independent random variables and Y be the random variable $f(X_1, \ldots, X_n)$, where function f satisfies*

$$|f(x) - f(x_k')| \leq c_k$$

whenever x and x_k' differ in the kth coordinate. Then for any $t > 0$,

$$\mathbb{P}(Y - \mathbb{E}Y > t) \leq \exp\left(\frac{-2t^2}{\sum_{k=1}^n c_k^2}\right).$$

Proof of Lemma 3.10. Let $e = (e(1), \ldots, e(T))^T$, and $e_k' = (e(1), \ldots, e'(k), \ldots, e(T))^T$ be formed by replacing $e(k)$ with an independent and identically distributed copy $e'(k)$. Then

$$\begin{aligned} |g_1(e) - g_1(e_k')| &\leq \frac{1}{w+1} \beta \sum_{m=0}^{w} |f(m)(e(k) - e'(k))| \\ &\leq \frac{2}{w+1} \epsilon \beta \sum_{m=0}^{w} |f(m)| =: c_k. \end{aligned}$$

Therefore,

$$\sum_{k=1}^{T} c_k^2 = \frac{4\epsilon^2\beta^2 T}{(w+1)^2}\left(\sum_{m=0}^{w}|f(m)|\right)^2 \leq 4\epsilon^2\beta^2\frac{T}{(w+1)\sigma^2}||f_w||^2.$$

By Lemma 4.9,

$$\mathbb{P}(g_1'(e) - \mathbb{E}g_1'(e) > u) \leq \exp\left(\frac{-u^2}{2\epsilon^2\beta^2\frac{T}{(w+1)\sigma^2}(||f_w||)^2}\right).$$

Substituting this into (3.13) and (3.11) finishes the proof. □

Bounding the loss due to prediction error In this section we prove the following concentration result for the loss due to correlated prediction error.

Lemma 3.14. *The loss due to prediction error has Berstein type tail: for any $u > 0$,*

$$\mathbb{P}(g_2 > u + V_2) \leq \exp\left(\frac{-u^2}{8\epsilon^2\lambda(\frac{T}{w+1}F(w) + u)}\right). \tag{3.14}$$

To prove Lemma 3.14, we characterize g_2 as a convex function of e in Lemma 3.15. We then show that this is a *self-bounding* function. Combining convexity and self-bounding property of g_2, Lemma 5.23 makes use of the convex Log-Sobolev inequality to prove concentration of g_2.

Lemma 3.15. *The expectation of g_2 is $\mathbb{E}g_2 = V_2$, and g_2 is a convex quadratic form of e. Specifically, there exists a matrix $A \in \mathbb{R}^{T\times T}$, such that $g_2 = \frac{1}{2}||Ae||^2$. Furthermore, the spectral radius of λ of AA^T satisfies $\lambda \leq F(w)$.*

Hence, (3.14) is equivalent to a concentration result of g_2:

$$\mathbb{P}(g_2 > V_2 + u) = \mathbb{P}(g_2 - \mathbb{E}g_2 > u).$$

The method of bounded difference used in the previous section is not good for a quadratic function of e because the uniform bound of $|g_2(e) - g_2(e_k')|$ is too large since

$$|g_2(e) - g_2(e_k')| = \frac{1}{2}|(e - e'(k))^T A^T A(e + e'(k))|,$$

where the $(e + e'(k))$ term has T non-zero entries and a uniform upper bound of this will be in $\Omega(T)$. Instead, we use the fact that the quadratic form is self-bounding. Let $h(e) = g_2(e) - \mathbb{E}g_2(e)$. Then

$$\begin{aligned} ||\nabla h(e)||^2 &= ||A^T Ae||^2 = (Ae)^T(AA^T)(Ae) \\ &\leq \lambda(Ae)^T(Ae) = 2\lambda[h(e) + \mathbb{E}V_2]. \end{aligned}$$

We now introduce the concentration bound for a self-bounding function of a collection of random variables. The proof uses the convex Log-Sobolev inequality [53].

Lemma 3.16. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and random variable X be supported on $[-d/2, d/2]^n$. If $\mathbb{E}[f(X)] = 0$ and f satisfies the self-bounding property*

$$||\nabla f||^2 \leq af + b, \tag{3.15}$$

for $a, b > 0$, then the tail of $f(X)$ can be bounded as

$$\mathbb{P}\left\{f(X) > t\right\} \leq \exp\left(\frac{-t^2}{d^2(2b + at)}\right). \tag{3.16}$$

Now to complete the proof of Lemma 3.14, apply Lemma 5.23 to the random variable $Z = h(e)$ to obtain

$$\mathbb{P}\{g_2 - \mathbb{E}g_2 > u\} \leq \exp\left(-\frac{u^2}{8\lambda_{\max}\epsilon^2(2V_2 + u)}\right)$$

for $t > 0$, i.e.,

$$\begin{aligned} \mathbb{P}\{g_2 > u + v_2\} &\leq \exp\left(-\frac{u^2}{8\lambda_{\max}\epsilon^2(2V_2 + t)}\right) \\ = \exp&\left(\frac{-u^2}{8\epsilon^2\lambda(\frac{T}{w+1}F(w) + u)}\right). \end{aligned}$$

3.4 Concluding remarks

Making use of predictions about the future is a crucial, but under-explored, area of online algorithms. In this Chapter, we have introduced a general colored noise model for studying predictions. This model captures a range of important phenomena for prediction errors including, general correlation structures, prediction noise that increases with the prediction horizon, and refinement of predictions as time passes. Further it allows for worst-case analysis of online algorithms in the context of stochastic prediction errors.

To illustrate the insights that can be gained from incorporating a general model of prediction noise into online algorithms, we have focused on online optimization problems with switching costs, specifically, an online LASSO formulation. Our results highlight that a simple online algorithm, AFHC, can simultaneously achieve a constant competitive ratio and a sublinear regret in expectation in nearly any situation where it is feasible for an online algorithm to do so. Further, we show that the cost of AFHC is tightly concentrated around its mean.

We view this chapter as a first step toward understanding the role of predictions in the design of online optimization algorithms and, more generally, the design of online algorithms. In particular, while we have focused on a particular, promising algorithm, AFHC, it is quite interesting to ask if it is possible to design online algorithms that outperform AFHC. We have proven that AFHC uses the asymptotically minimal amount of predictions to achieve constant competitive ratio and sublinear regret; however, the cost of other algorithms may be lower if they can use the predictions more efficiently.

In addition to studying the performance of algorithms other than AFHC, it would also be interesting to generalize the prediction model further, e.g., by considering non-stationary processes or heterogeneous $e(t)$.

3.A Proofs for Section 3.2

Proof of Theorem 3.2

Proof. Let $(x_{ALG,t})_{t=1}^{T}$ be the solution produced by online algorithm ALG. Then

$$\begin{aligned}\text{cost}(ALG) \geq& \frac{1}{2}\sum_{t=1}^{T}||y_t - Kx_{ALG,t}||^2\\ =& \frac{1}{2}\sum_{t=1}^{T}||(I - KK^\dagger)y_t||^2 + ||KK^\dagger y_t - Kx_{ALG,t}||^2,\end{aligned}$$

by the identity $(I-KK^\dagger)K = 0$. Let $\epsilon_t = x_{ALG,t} - K^\dagger y_{t|t-1}$, i.e., $\epsilon_t = x_{ALG,t} - K^\dagger(y_{t|0} - \sum_{s=1}^{t-1} f(t-s)e(s))$. Since all predictions made up until t can be expressed in terms of $y_{\cdot|0}$ and $e(\tau)$ for $\tau < t$, which are independent of $e(t)$, and all other information (internal randomness) available to ALG is independent of $e(t)$ by assumption, ϵ_t is independent of $e(t)$. It follows that

$$\begin{aligned}&\mathbb{E}_e[\text{cost}(ALG)] \geq \mathbb{E}_e[||KK^\dagger y_t - K(K^\dagger y_{t|t-1} + \epsilon_t)||^2]\\ =&\frac{1}{2}\sum_{t=1}^{T}\mathbb{E}_{e\backslash e(t)}\mathbb{E}_{e(t)|e\backslash e(t)}||KK^\dagger e(t)^T - K\epsilon_t||^2 \\ =&\frac{1}{2}\sum_{t=1}^{T}\mathbb{E}_{e\backslash e(t)}(||R_e^{1/2}||^2_{KK^\dagger} + ||(\mathbb{E}_{e(t)}\epsilon_t\epsilon_t^T)^{1/2}||^2_{K^TK})\\ \geq&\frac{T}{2}||R_e^{1/2}||^2_{KK^\dagger},\end{aligned} \tag{3.17}$$

where the first equality uses the identity $(I - KK^\dagger)K = 0$, and the second uses the independence of ϵ_t and $e(t)$. □

Proof of Theorem 3.3

By Theorem 3.2, if $\mathbb{E}[\text{cost}(ALG)] \in o(T)$, there must be some t such that ϵ_t is not independent of $e(t)$. By expanding the square term in (3.17) and noting it is nonnegative:

$$\mathbb{E}[e(t)^T K\epsilon_t] \leq \frac{1}{2}||R_e^{1/2}||^2_{KK^\dagger,F} + \frac{1}{2}||(\mathbb{E}\epsilon_t\epsilon_t^T)^{1/2}||^2_{K^TK,F}.$$

Each nonzero $\mathbb{E}[e(t)^T K\epsilon_t]$ can at most make one term in (3.17) zero, since there are T terms in (3.17) that are each lower bounded by $\frac{1}{2}||R_e^{1/2}||^2$, and by assumption $\mathbb{E}[\text{cost}(ALG)]$ is sublinear. There can be at most a sublinear number of t such that $\mathbb{E}[e(t)^T K\epsilon_t] = 0$.

For every other t, we must have $\mathbb{E}[e(t)^T K\epsilon_t] > 0$. Let $l_t = \mathbb{E}[e(t)^T K\epsilon_t] > 0$, and $a_t = ||(\mathbb{E}\epsilon_t\epsilon_t^T)^{1/2}||^2_{KK^T,F} > 0$.

Therefore, at time t, the algorithm can produce prediction $y'_{t|t-1} = y_{t|t-1} + \frac{1}{w_t}K\epsilon_t$, where the coefficient w_t is chosen later. Then the one step prediction error variance:

$$\begin{aligned}\mathbb{E}||y_t - y'_{t|t-1}||^2 &= \mathbb{E}||e(t) - \frac{1}{w_t}K\epsilon_t||^2\\ &= ||R_e^{1/2}||^2_F - \frac{2}{w_t}l_t + \frac{1}{w_t^2}a_t.\end{aligned}$$

Pick any $w_t > a_t/2l_t$. Then $\mathbb{E}||y_t - y'_{t|t-1}||^2 < ||R_e^{1/2}||^2_F = \mathbb{E}||y_t - y_{t|t-1}||^2$. Thus ALG can produce better one-step prediction for all but sublinearly many t.

Proof of Lemma 3.6

To prove Lemma 3.6, we use the following Lemma.

Lemma 3.17. *The competitive difference of FHC with fixed* $(w+1)$*-lookahead for any realization is given by*

$$\begin{aligned}\text{cost}(FHC^{(k)}) \leq & \text{cost}(OPT) + \sum_{\tau \in \Omega_k} \beta ||(x^*_{\tau-1} - x^{(k)}_{\tau-1})||_1 \\ & + \frac{1}{2} \sum_{\tau \in \Omega_k} \sum_{t=\tau}^{\tau+w} ||KK^\dagger(y_t - y_{t|\tau-1})||^2,\end{aligned}$$

where x^*_t is the action chosen by the dynamic offline optimal.

of lemma 3.6. Note that $\text{cost}(FHC^{(k)})$ is convex. The result then follows with a straightforward application of Jensen's inequality to Lemma 3.17. By the definition of $AFHC$, we have the following inequality:

$$\text{cost}(AFHC) \leq \frac{1}{w+1} \sum_{k=0}^{w} \text{cost}(FHC^{(k)}).$$

By substituting the expression for $\text{cost}(FHC^{(k)})$ into the equation above and simplifying, we get the desired result. □

Before we prove Lemma 3.17, we first introduce a new algorithm we term $OPEN$. This algorithm runs an open loop control over the entire time horizon, T. Specifically, it chooses actions x_t, for $t \in 1, ..., T$, that solve the following optimization problem:

$$\min \frac{1}{2} \sum_{t=1}^{T} (y_{t|0} - Kx_t)^2 + \beta ||(x_t - x_{t-1})||_1.$$

$FHC^{(k)}$ can be seen as starting at $x^k_{FHC,\tau-1}$, using prediction $y_{\cdot|\tau-1}$, and running $OPEN$ from τ to $\tau + w$. Then repeating with updated prediction $y_{\cdot|\tau+\omega}$. We first prove the following Lemma characterizing the performance of $OPEN$.

Lemma 3.18. *Competitive difference of $OPEN$ over a time horizon, T, is given by*

$$\text{cost}(OPEN) - \text{cost}(OPT) \leq \sum_{t=1}^{T} \frac{1}{2} ||\hat{y}_t - y_t||^2_{KK^\dagger}.$$

Proof. Recall that the specific OCO we are studying is

$$\min_x \sum_{t=1}^{T} \frac{1}{2} ||y_t - Kx_t||^2 + \beta ||(x_t - x_{t-1})||_1, \tag{3.18}$$

where $x_t \in \mathbb{R}^n$, $y_t \in \mathbb{R}^m$, $K \in \mathbb{R}^{m \times n}$ and the switching cost, $\beta \in \mathbb{R}_+$.

We first derive the dual of (3.18) by linearizing the l_1 norm, which leads to the following equivalent expression of the objective above:

$$\min_{x,z} \frac{1}{2} \sum_{t=1}^{T} ||y_t - Kx_t||^2 + \beta \mathbb{1}^T z_t$$
$$\text{s.t.} \quad z_t \geq x_t - x_{t-1}, z_t \geq x_{t-1} - x_t, \quad \forall t.$$

The Lagrangian is

$$L(x, z; \bar{\lambda}, \underline{\lambda}) = \frac{1}{2} \sum_{t=1}^{T} ||y_t - Kx_t||^2 + \langle \bar{\lambda}_t - \underline{\lambda}_t, x_t - x_{t-1} \rangle$$
$$+ \langle \beta \mathbb{1} - (\bar{\lambda}_t + \underline{\lambda}_t), z_t \rangle,$$

where we take $\lambda_{T+1} = 0$ and $x_0 = 0$.

Let $\lambda_t = \bar{\lambda}_t - \underline{\lambda}_t$ and $w_t = \bar{\lambda}_t + \underline{\lambda}_t$. Dual feasibility requires $w_t \leq \beta \mathbb{1}, \forall t$, which implies $-\beta \mathbb{1} \leq \lambda_t \leq \beta \mathbb{1}, \forall t$. Dual feasibility also requires $\langle \beta \mathbb{1} - w_t, z_t \rangle = 0, \forall t$.

Now by defining $s_t = \lambda_t - \lambda_{t+1}$ and equating the derivative with respect to x_t to zero, the primal and dual optimal x_t^*, s_t^* must satisfy $K^T K x_t^* = K^T y_t - s_t^*$.

Note by premultiplying the equation above by x_t^{*T}, we have $\langle x_t^*, s_t^* \rangle = \langle Kx_t^*, y_t \rangle - ||Kx_t^*||^2$. If instead we premultiply the same equation by $(K^T)^\dagger$, we have after some simplification that $Kx_t^* = (KK^\dagger)y_t - (K^T)^\dagger s_t^*$. We can now simplify the expression for the optimal value of the objective by using the above two equations:

$$\text{cost}(OPT) = \sum_{t=1}^{T} \frac{1}{2} ||y_t - Kx_t^*||^2 + \langle x_t^*, s_t^* \rangle$$
$$= \sum_{t=1}^{T} \frac{1}{2} ||y_t||^2 - \frac{1}{2} ||KK^\dagger y_t - (K^T)^\dagger s_t^*||^2. \tag{3.19}$$

Observe that (3.19) implies s_t^* minimizes the following expression $\sum_{t=1}^{T} ||KK^\dagger y_t - (K^T)^\dagger s_t^*||^2$ over the constraint set $\mathcal{S} = \{s_t | s_t = \lambda_t - \lambda_{t+1}, -\beta \mathbb{1} \leq \lambda_t \leq \beta \mathbb{1}$ for $1 \leq$

$t \leq T, \lambda_{T+1} = 0\}$.

$$\begin{aligned}
&\text{cost}(OPEN) - \text{cost}(OPT) = p(\hat{x}; y) - p(x; y) \\
=&p(\hat{x}; \hat{y}) - p(x; y) + p(\hat{x}; y) - p(\hat{x}; \hat{y}_t) \\
=&\sum_{t=1}^{T} \frac{1}{2}||\hat{y}_t||^2 - \frac{1}{2}||K\hat{x}_t||^2 - \frac{1}{2}||y_t||^2 + \frac{1}{2}||Kx_t^*||^2 \\
&+ \frac{1}{2}||y_t - K\hat{x}_t||^2 - \frac{1}{2}||\hat{y}_t - K\hat{x}_t||^2.
\end{aligned}$$

Expanding the quadratic terms, using the property of the pseudo-inverse that $K^\dagger K K^\dagger = K^\dagger$, and using the fact that $Kx_t^* = KK^\dagger y_t - (K^T)^\dagger s_t^*$, we have

$$\begin{aligned}
&\text{cost}(OPEN) - \text{cost}(OPT) \\
=&\sum_{t=1}^{T} \frac{1}{2}\left(||KK^\dagger y_t - (K^T)^\dagger s_t^*||^2 - ||(KK^\dagger y_t - (K^T)^\dagger \hat{s}_t)||^2\right) \\
+&\frac{1}{2}\left(||\hat{y}_t - y_t||^2 - ||(I - KK^\dagger)(\hat{y}_t - y_t)||^2\right) \\
\leq&\sum_{t=1}^{T} \frac{1}{2}||\hat{y}_t - y_t||^2 - \frac{1}{2}||(I - KK^\dagger)(\hat{y}_t - y_t)||^2 \\
=&\sum_{t=1}^{T} \frac{1}{2}||KK^\dagger(\hat{y}_t - y_t)||^2,
\end{aligned}$$

where the first inequality is because of the characterization of s_t^* following (3.19). □

Proof of Lemma 3.17. The proof is a straightforward application of Lemma 3.18. Summing the cost of $OPEN$ for all $\tau \in \Omega_k$ and noting that the switching cost term satisfying the triangle inequality gives us the desired result. □

Proof of Theorem 3.4

We first define the sub-optimality of the open loop algorithm over expectation of the noise. $\mathbb{E}[||(y_t - \hat{y}_t)||^2_{KK^\dagger}]$ is the expectation of the projection of the prediction error

$t+1$ time steps away onto the range space of K, given by:

$$\begin{aligned}
&\mathbb{E}[||(y_t - \hat{y}_t)||^2_{KK^\dagger}] = \mathbb{E}||\sum_{s=1}^{t} KK^\dagger(f(t-s)e(s))||^2\\
=&\mathbb{E}[\sum_{s_1=1}^{t}\sum_{s_2=1}^{t} e(s_1)^T f(t-s_1)^T (KK^\dagger)^T (KK^\dagger) f(t-s_2)e(s_2)]\\
=&\mathrm{tr}(\sum_{s_1=1}^{t}\sum_{s_2=1}^{t} f(t-s_1)^T (KK^\dagger)(KK^\dagger) f(t-s_2)\mathbb{E}[e(s_2)e(s_1)^T])\\
=&\sum_{s=0}^{t-1}\mathrm{tr}(f(s)^T(KK^\dagger)f(s)R_e),
\end{aligned}$$

where the last line is because $\mathbb{E}[e(s_1)e(s_2)^T] = 0$ for all $s_1 \neq s_2$, and $KK^\dagger K = K$. Note that this implies $||f_{t-1}||^2 = \sum_{s=0}^{t-1}\mathrm{tr}(f(s)^T f(s)R_e)$. We now write the expected suboptimality of the open loop algorithm as

$$\begin{aligned}
&\mathbb{E}[\mathrm{cost}(OPEN) - \mathrm{cost}(OPT)] \leq \sum_{t=1}^{T}\frac{1}{2}\mathbb{E}[||y_t - \hat{y}_t||^2_{KK^\dagger}]\\
=&\frac{1}{2}\sum_{s=0}^{T-1}\sum_{t=s}^{T-1}\mathrm{tr}(f(s)^T KK^\dagger f(s)R_e)\\
=&\frac{1}{2}\sum_{s=0}^{T-1}(T-s)\mathrm{tr}(f(s)^T KK^\dagger f(s)R_e) = F(T-1),
\end{aligned}$$

where the first equality is by rearranging the summation.

Now we take expectation of the expression we have in Lemma 3.6. Taking expectation of the second penalty term (prediction error term), we have

$$\begin{aligned}
&\frac{1}{w+1}\sum_{k=0}^{w}\sum_{\tau\in\Omega_k}\sum_{t=\tau}^{\tau+w}\mathbb{E}\frac{1}{2}||(KK^\dagger)(y_t - y_{t|\tau-1})||^2\\
=&\frac{1}{2(w+1)}\sum_{k=0}^{w}\sum_{\tau\in\Omega_k}F(w) = \frac{T}{2(w+1)}F(w).
\end{aligned}$$

We now need to bound the first penalty term (switching cost term). By taking the subgradient with respect to x_t and by optimality we have $\forall t = 1,\ldots,T$

$$\begin{aligned}
&0 \in K^T(Kx_t^* - y_t) + \beta\partial||(x_t^* - x_{t-1}^*)||_1 + \beta\partial||(x_{t+1}^* - x_t^*)||_1\\
\Rightarrow& x_t^* \in [(K^TK)^{-1}(K^Ty_t - 2\beta\mathbb{1}), (K^TK)^{-1}(K^Ty_t + 2\beta\mathbb{1})],
\end{aligned}$$

where the implication is because the sub-gradient of a 1-norm function $||\cdot||_1$ is between $-\mathbb{1}$ to $\mathbb{1}$.

Similarly, since $x^{(k)}_{\tau-1}$ is the last action taken over a FHC horizon, we have that for all $\tau \in \Omega_k$,

$$\begin{aligned} x^{(k)}_{\tau-1} \in [&(K^TK)^{-1}(K^Ty_{\tau-1|\tau-w-2} - \beta\mathbb{1}), \\ &(K^TK)^{-1}(K^Ty_{\tau-1|\tau-w-2} + \beta\mathbb{1})]. \end{aligned}$$

Taking expectation of one of the switching cost term and upper bounding with triangle inequality:

$$\begin{aligned} &\mathbb{E}\left\|(x^*_{\tau-1} - x^{(k)}_{\tau-1})\right\|_1 \\ \leq &||K^\dagger||_1\mathbb{E}||y_{\tau-1} - y_{\tau-1|\tau-2-w}||_1 + 3\beta||(K^TK)^{-1}\mathbb{1}||_1 \\ \leq &||K^\dagger||_1||f_w|| + 3\beta||(K^TK)^{-1}\mathbb{1}||_1, \end{aligned} \tag{3.20}$$

where the first inequality is by the definition of induced norm, the second inequality is due to concavity of the square-root function and Jensen's inequality. Summing (3.20) over k and τ, we have the expectation of the switching cost term. Adding the expectation of both penalty terms (loss due to prediction error and loss due to switching cost) together, we get the desired result.

Proof of Lemma 3.7

We first characterize cost(STA):

$$\text{cost}(STA) = \min_x \frac{1}{2}\sum_{t=1}^T ||y_t - Kx||_2^2 + \beta\mathbb{1}^Tx.$$

By first order conditions, we have the optimal static solution $x = K^\dagger\bar{y} - \frac{\beta}{T}(K^TK)^{-1}\mathbb{1}$. Substituting this to cost(STA) and simplifying, we have

$$\begin{aligned} cost(STA) =&\frac{1}{2}\sum_{t=1}^T\left(||(I - KK^\dagger)y_t||_2^2 + ||KK^\dagger(y_t - \bar{y})||_2^2\right) \\ &- \frac{\beta^2\mathbb{1}^T(K^TK)^{-1}\mathbb{1}}{2T} + \beta\mathbb{1}^TK^\dagger\bar{y}. \end{aligned}$$

Let $C = \frac{\beta^2\mathbb{1}^T(K^TK)^{-1}\mathbb{1}}{2T}$. Subtracting cost($OPT$) in (3.19) from the above, we have

$\text{cost}(STA) - \text{cost}(OPT)$ equals:

$$\begin{aligned}
&\frac{1}{2}\sum_{t=1}^{T}(||KK^{\dagger}(y_t - \bar{y})||_2^2 - ||KK^{\dagger}y_t||_2^2 + ||KK^{\dagger}y_t - (K^T)^{\dagger}s_t^*||_2^2) \\
&+ \beta\mathbb{1}^T K^{\dagger}\bar{y} - C \\
=&\frac{1}{2}\sum_{t=1}^{T}\left(||KK^{\dagger}(y_t - \bar{y}) - (K^T)^{\dagger}s_t^*||_2^2\right) + \langle K^{\dagger}\bar{y}, \beta\mathbb{1} - \lambda_1\rangle - C \\
\geq&\frac{1}{2}\sum_{t=1}^{T}\left(||KK^{\dagger}(y_t - \bar{y}) - (K^T)^{\dagger}s_t^*||_2^2\right) - C.
\end{aligned}$$

The first equality is by expanding the square terms and noting $s_t = \lambda_t - \lambda_{t+1}$. The last inequality is because $-\beta\mathbb{1} \leq \lambda_t \leq \beta\mathbb{1}$ and $\beta\mathbb{1}^T K^{\dagger}\bar{y}$ being positive by assumption that the optimal static solution is positive. Now we bound the first term of the inequality above:

$$\begin{aligned}
&\frac{1}{2}\sum_{t=1}^{T}\left(||KK^{\dagger}(y_t - \bar{y}) - (K^T)^{\dagger}s_t^*||_2^2\right) \\
\geq&\frac{1}{2}\sum_{t=1}^{T}\left(||KK^{\dagger}(y_t - \bar{y})||^2\right) - \sum_{t=1}^{T}\langle KK^{\dagger}(y_t - \bar{y}), (K^T)^{\dagger}s_t^*\rangle \\
\geq&\frac{1}{2}\sum_{t=1}^{T}\left(||KK^{\dagger}(y_t - \bar{y})||^2\right) - 2\beta\sum_{t=1}^{T}||KK^{\dagger}(y_t - \bar{y})||\,||(K^T)^{\dagger}\mathbb{1}|| \\
\geq&\frac{1}{2}\sum_{t=1}^{T}\left(||KK^{\dagger}(y_t - \bar{y})||^2\right) - 2B\sqrt{T\sum_{t=1}^{T}||(KK^{\dagger})(y_t - \bar{y})||^2},
\end{aligned}$$

where $B = \beta||(K^T)^{\dagger}\mathbb{1}||_2$.

By subtracting C from the expression above and completing the sqaure, we have the desired result.

Proof of Theorem 3.8

Using the results of Lemma 3.7, taking expectation and applying Jensen's inequality, we have

$$\begin{aligned}
&\mathbb{E}_e\left[\text{cost}(STA) - \text{cost}(OPT)\right] \\
\geq&\mathbb{E}_e\Big[\frac{1}{2}\sum_{t=1}^{T}||KK^{\dagger}(y_t - \bar{y})||^2 - 2B\sqrt{T\sum_{t=1}^{T}||(KK^{\dagger})(y_t - \bar{y})||^2 - C}\Big] \\
\geq&\frac{1}{2}\left(\sqrt{\mathbb{E}_e\sum_{t=1}^{T}\left(||KK^{\dagger}(y_t - \bar{y})||^2\right)} - 2B\sqrt{T}\right)^2 - 2B^2T - C.
\end{aligned}$$

Therefore by Theorem 3.4, the regret of $AFHC$ is

$$\sup_{\hat{y}} \left(\mathbb{E}_e[\text{cost}(AFHC)-\text{cost}(OPT) + \text{cost}(OPT)-\text{cost}(STA)]\right)$$
$$\leq VT + 2B^2T + C - \frac{1}{2}\inf_{\hat{y}}\left(\sqrt{\mathbb{E}_e\sum_{t=1}^{T}||(y_t - \bar{y})||^2_{KK^\dagger}} - 2B\sqrt{T}\right)^2.$$

Let $S(T) = \mathbb{E}_e \sum_{t=1}^{T} ||y_t - \bar{y}||^2_{KK^\dagger}$. By the above, to prove $AFHC$ has sublinear regret, it is sufficient that

$$VT + 2B^2T - \frac{1}{2}\inf_{\hat{y}}(\sqrt{S(T)} - 2B\sqrt{T})^2 < g(T) \tag{3.21}$$

for some sublinear $g(T)$. By the hypothesis of Theorem 3.8, we have $\inf_{\hat{y}} S(T) \geq (8A + 16B^2)T$.

Then, $S(T) \geq (\sqrt{2VT + 4B^2T} + 2B\sqrt{T})^2$, and (3.21) holds since $VT + 2B^2T - \frac{1}{2}\inf_{\hat{y}}(\sqrt{S(T)} - 2B\sqrt{T})^2 \leq VT + 2B^2T - \frac{1}{2}\left(\sqrt{2VT + 4B^2T}\right)^2 = 0$.

3.B Proofs for Section 3.3

Proof of Lemma 3.11

By the triangle inequality, we have

$$\begin{aligned} g_1 =& \frac{1}{w+1}\sum_{k=0}^{w}\sum_{\tau\in\Omega_k}\beta|x^*_{\tau-1} - x^{(k)}_{\tau-1}| \\ \leq& \frac{1}{w+1}\sum_{k=0}^{w}\sum_{\tau\in\Omega_k}\beta\Big(|x^*_{\tau-1} - y_{\tau-1}| + |y_{\tau-1} - y_{\tau-1|\tau-w-2}| \\ &+ |y_{\tau-1|\tau-w-2} - x^{(k)}_{\tau-1}|\Big). \end{aligned}$$

By first order optimality condition, we have $x^*_{\tau-1} \in \{y_{\tau-1} - 2\beta, y_{\tau-1} + 2\beta\}$, and $x^{(k)}_{\tau-1} \in \{y_{\tau-1|\tau-w-2} - \beta, y_{\tau-1|\tau-w-2} + \beta\}$. Thus, by the prediction model,

$$g_1 \leq \frac{3\beta^2T}{w+1} + \frac{\beta}{w+1}\sum_{k=0}^{w}\sum_{\tau\in\Omega_k}\left|\sum_{s=1\vee(\tau-w-2)}^{\tau-1} f(\tau-1-s)e(s)\right|.$$

Proof of Lemma 3.12

Note that by Lemma 3.11, we have

$$\begin{aligned} \mathbb{E}g_1'(e) \leq& \frac{\beta}{w+1}\sum_{k=0}^{w}\sum_{\tau\in\Omega_k}\mathbb{E}\left|\sum_{s=1\vee(\tau-w-2)}^{\tau-1} f(\tau-1-s)e(s)\right| \\ \leq& \frac{\beta}{w+1}\sum_{k=0}^{w}\sum_{\tau\in\Omega_k}\sqrt{\sigma^2\sum_{s=0}^{w} f^2(s)} = \frac{\beta T}{w+1}||f_w||^2. \end{aligned}$$

where the second inequality is by Jensen's inequality and taking expectation.

Proof of Lemma 3.15

By definition of g_2 and unraveling the prediction model, we have

$$\begin{aligned} g_2 &= \frac{1}{w+1}\sum_{k=0}^{w}\sum_{\tau\in\Omega_k}\sum_{t=\tau}^{\tau+w}\frac{1}{2}(y_t - y_{t|\tau-1})^2 \\ &= \frac{1}{w+1}\sum_{k=0}^{w}\sum_{\tau\in\Omega_k}\sum_{t=\tau}^{\tau+w}\frac{1}{2}(\sum_{s=\tau}^{t} f(t-s)e(s))^2. \end{aligned}$$

Writing it in matrix form, it is not hard to see that

$$g_2 = \frac{1}{w+1}\sum_{k=0}^{w}\frac{1}{2}||A_k e||^2,$$

where A_k has the block diagonal structure given by

$$A_k = \mathrm{diag}(A_k^1, A_k^2, \dots, A_k^2, A_k^3),^{4} \tag{3.22}$$

and there are the types of submatrices in A_k given by, for $i = 1, 2, 3$:

$$A_k^i = \begin{pmatrix} f(0) & 0 & \dots & 0 \\ f(1) & f(0) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f(v_i) & f(v_{i-1}) & \dots & f(0) \end{pmatrix},$$

where $v_1 = k - 2$ if $k \geq 2$ and $v_1 = k + w - 1$ otherwise. $v_2 = w$, $v_3 = (T - k + 1) \mod (w + 1)$. Note that in fact, the matrix A_k^2 is the same for all k. Therefore, we have

$$g_2 = \frac{1}{2}e^T(\frac{1}{w+1}\sum_{k=0}^{w}A_k^T A_k)e = \frac{1}{2}||Ae||^2,$$

where we define A to be such that $A^T A = \frac{1}{w+1}\sum_{k=0}^{w} A_k^T A_k$, this can be done because the right-hand side is positive semidefinite, since A_k is lower triangular. The last equality is because all A_l^2 has the same structure. Let λ be the maximum eigenvalue of AA^T, which can be expressed by

$$\begin{aligned} \lambda &= \max_{||x||=1} x^T AA^T x \\ &= \frac{1}{w+1}\max_{||x||=1}\sum_{k=0}^{w} x^T A_k A_k^T x \leq \frac{1}{w+1}\sum_{k=0}^{w}\lambda_k, \end{aligned}$$

[4] The submatrix A_k^2 is repeated $\lfloor\frac{T-k+1}{\omega+1}\rfloor$ times in A_k for $k \geq 2$, and $\lfloor\frac{T-k-\omega}{\omega+1}\rfloor$ times for otherwise.

where λ_k is the maximum eigenvalue of $A_k A_k^T$. Note that A_k has a block diagonal structure, hence $A_k A_k^T$ also has block diagonal structure, and if we divide the vector $x = (x_1, x_2, \ldots, x_m)$ into sub-vectors of appropriate dimension, then by the block diagonal nature of $A_k A_k^T$, we have

$$\begin{aligned} x^T A_k A_k^T x =& x_1^T A_k^1 {A_k^1}^T x_1 + x_2^T A_k^2 {A_k^2}^T x_2 + \ldots \\ &+ x_{m-1}^T A_k^2 {A_k^2}^T x_{m-1} + x_m^T A_k^3 {A_k^3}^T x_m. \end{aligned}$$

As such, if we denote the maximum eigenvalues of λ_k^i as the maximum eigenvalue of the matrix $A_k^i {A_k^i}^T$, then we have

$$\begin{aligned} \lambda_k &= \max_x \frac{x^T A_k A_k^T x}{x^T x} \\ &= \max_{x_1, \ldots, x_m} \frac{x_1^T A_k^1 {A_k^1}^T x_1 + x_2^T A_k^2 {A_k^2}^T x_2 + \ldots + x_m^T A_k^3 {A_k^3}^T x_m}{x_1^T x_1 + \ldots + x_m^T x_m} \\ &\leq \max_{x_1, \ldots, x_m} \frac{\max(\lambda_k^1, \lambda_k^2, \lambda_k^3) \cdot (x_1^T x_1 + \ldots + x_m^T x_m)}{x_1^T x_1 + \ldots + x_m^T x_m} \\ &\leq \max(\lambda_k^1, \lambda_k^2, \lambda_k^3), \end{aligned}$$

where λ_k^i is the maximum eigenvalue of A_k^i for $i \in \{1, 2, 3\}$. As $A_k^i {A_k^i}^T$ are all positive semidefinite, we can bound the maximum eigenvalue by trace, and note that A_k^1 and A_k^3 are submatrix of A_k^2, we have

$$\lambda_k \leq \max(\lambda_k^1, \lambda_k^2, \lambda_k^3) \leq \mathrm{tr}(A_k^2 {A_k^2}^T) = \frac{1}{\sigma^2} F(w).$$

Proof of Lemma 5.23

To prove the lemma, we use the following variant of Log-Sobolev inequality

Lemma 3.19 (Theorem 3.2, [53]). *Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be convex, and random variable X be supported on $[-d/2, d/2]^n$, then*

$$\begin{aligned} &\mathbb{E}[\exp(f(X)) f(X)] - \mathbb{E}[\exp(f(X))] \log \mathbb{E}[\exp(f(X))] \\ \leq& \frac{d^2}{2} \mathbb{E}[\exp(f(X)) ||\nabla f(X)||^2]. \end{aligned}$$

We will use Lemma 5.22 to prove Lemma 5.23. Denote the moment generating function of $f(X)$ by

$$m(\theta) := \mathbb{E} e^{\theta f(X)}, \qquad \theta > 0.$$

The function $\theta f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is convex, and therefore it follows from Lemma 5.22 that

$$\begin{aligned} \mathbb{E}\left[e^{\theta f}\theta f\right] - \mathbb{E}\left[e^{\theta f}\right]\ln\mathbb{E}\left[e^{\theta f}\right] &\leq \frac{d^2}{2}\mathbb{E}\left[e^{\theta f}||\theta\nabla f||^2\right], \\ \theta m'(\theta) - m(\theta)\ln m(\theta) &\leq \frac{1}{2}\theta^2 d^2 \mathbb{E}[e^{\theta f}||\nabla f||^2]. \end{aligned}$$

By to the self-bounding property (5.23),

$$\begin{aligned} \theta m'(\theta) - m(\theta)\ln m(\theta) &\leq \frac{1}{2}\theta^2 d^2 \mathbb{E}[e^{\theta f(X)}(af(X) + b)] \\ &= \frac{1}{2}\theta^2 d^2\left[am'(\theta) + bm(\theta)\right]. \end{aligned}$$

Since $m(\theta) > 0$, dividing by $\theta^2 m(\theta)$ gives

$$\frac{d}{d\theta}\left[\left(\frac{1}{\theta} - \frac{ad^2}{2}\right)\ln m(\theta)\right] \leq \frac{bd^2}{2}. \tag{3.23}$$

Since $m(0) = 1$ and $m'(0) = \mathbb{E}f(X) = 0$, we have

$$\lim_{\theta\to 0^+}\left(\frac{1}{\theta} - \frac{ad^2}{2}\right)\ln m(\theta) = 0,$$

and therefore integrating both sides of (3.23) from 0 to s gives

$$\left(\frac{1}{s} - \frac{ad^2}{2}\right)\ln m(s) \leq \frac{1}{2}bd^2 s, \tag{3.24}$$

for $s \geq 0$. We can bound the tail probability $\mathbb{P}\{f > t\}$ with the control (5.25) over the moment generating function $m(s)$.

In particular,

$$\begin{aligned} \mathbb{P}\{f(X) > t\} &= \mathbb{P}\left\{e^{sf(X)} > e^{st}\right\} \leq e^{-st}\mathbb{E}\left[e^{sf(X)}\right] \\ &= \exp[-st + \ln m(s)] \\ &\leq \exp\left[-st + \frac{bd^2 s^2}{2 - asd^2}\right], \end{aligned}$$

for $s \in [0, 2/(ad^2)]$. Choose $s = t/(bd^2 + ad^2 t/2)$ to get

$$\mathbb{P}\{f(X) > t\} \leq \exp\left(\frac{-t^2}{d^2(2b + at)}\right).$$

Chapter 4

OPTIMIZING THE USE OF PREDICTION

In this chapter, we investigate the problem of designing online algorithms that make the best use of predictions. One of the most popular online algorithms that use predictions to make online decisions is Receding Horizon Control (RHC). RHC has a long history in the control theory literature [11, 18, 33, 50, 63, 83], but was first studied analytically in the context of OCO in [56]. In [56], RHC was proven to have a competitive ratio (the ratio of the cost incurred by RHC to the cost incurred by the offline optimal algorithm) of $1 + O(1/w)$ in the one-dimensional setting, where w is the size of the prediction window. However, the competitive ratio of RHC is $1 + \Omega(1)$ in the general case, and thus does *not* decrease to one as the prediction window grows in the worst case; this is despite the fact that predictions are assumed to have no noise (the perfect lookahead model). To this point there is no analytic work characterizing the performance of RHC with noisy predictions.

The poor worst-case performance of RHC motivated the proposal of Averaging Fixed Horizon Control (AFHC) [56], which provides an interesting contrast. While RHC is entirely "forward looking", AFHC keeps an "eye on the past" by respecting the actions of FHC algorithms in previous timesteps and thus avoids switching costs incurred by moving too quickly between actions. As a result, AFHC achieves a competitive ratio of $1+O(1/w)$ in both single and multi-dimensional action spaces, under the assumption of perfect lookahead, [56]. Further, strong guarantees on the performance of AFHC have been established in the case of noisy predictions [22].

Surprisingly, while the competitive ratio of AFHC is smaller than that of RHC, RHC provides *better* performance than AFHC in many practical cases. Further, RHC is seemingly more resistant to prediction noise in many settings (see Figure 4.1 for an example), though no analytic results are known for this case. Thus, at this point, two promising algorithms have been proposed, but it is unclear in what settings each should be used, and it is unclear if there are other algorithms that dominate these two proposals. In this chapter, we investigate how we can optimize the performance of online algorithm under different forms of prediction errors.

4.1 Problem formulation

We consider online convex optimization (OCO) problems with switching costs and noisy predictions. We first introduce OCO with switching costs (Section 4.1) and then describe the model of prediction noise (Section 4.1). Finally, we discuss the performance metric we consider in this chapter – the competitive difference – and how it relates to common measures such as regret and competitive ratio (Section 4.1).

OCO with switching costs

An OCO problem with switching costs considers a convex, compact decision/action space $F \subset \mathbb{R}^n$ and a sequence of cost functions $\{h_1, h_2, \ldots\}$, where each $h_t : F \longrightarrow \mathbb{R}^+$ is convex, and F is a compact set.

At time t, the following sequence occurs: (i) the online algorithm first chooses an action, which is a vector $x_t \in F \subset \mathbb{R}^n$, (ii) the environment chooses a cost function h_t from a set C, and (iii) the algorithm pays a stage cost $h_t(x_t)$ and a switching cost $\beta \|x_t - x_{t-1}\|$, where $\beta \in \mathbb{R}^+$, and $\|\cdot\|$ can be any norm in $\mathbb{R}^n$, and F is bounded in terms of this norm, i.e., $\|x - y\| \leq D$ for all $x, y \in F$.

Motivated by path planning and image labeling problems [22, 70, 75], we consider a variation of the above that uses a *parameterized* cost function $h_t(x_t) = h(x_t, y_t)$, where the parameter $y_t \in \mathbb{R}^m$ is the focus of prediction. This yields a total cost over T rounds of

$$\min_{x_t \in F} \sum_{t=1}^{T} h(x_t, y_t) + \beta \|x_t - x_{t-1}\|, \tag{4.1}$$

Note that prior work [22] considers only the case where a least-square penalty is paid each round, i.e., an online LASSO formulation with $h(x_t, y_t) = \frac{1}{2} \|y_t - Kx_t\|_2^2$. In this chapter, we consider more general h. We impose that $h(x_t, y_t)$ is separately convex in both x_t and y_t along with the following smoothness criteria.

Definition 6. *A function h is α-Hölder continuous in the second argument for $\alpha \in \mathbb{R}^+$, i.e., for all $x \in F$, there exists $G \in \mathbb{R}^+$, such that*

$$|h(x, y_1) - h(x, y_2)| \leq G \|y_1 - y_2\|_2^{\alpha}, \forall y_1, y_2.$$

G and α control the sensitivity of the cost function to a disturbance in y.

For this Chapter, we focus on $\alpha \leq 1$, since the only α-Hölder continuous function with $\alpha > 1$ is the constant function [6]. When $\alpha = 1$, h is *G-Lipschitz* in the second argument; if h is differentiable in the second argument, this is equivalent to $\left\|\partial_y h(x, y)\right\|_2 \leq G, \quad \forall x, y.$

Modeling prediction noise

Predictions about the future play a crucial role in almost all online decision problems. However, while significant effort has gone into designing predictors, e.g., [46, 73, 74, 89], much less work has gone into *integrating* predictions efficiently into algorithms. This is, in part, due to a lack of tractable, practical models for prediction noise. As a result, most papers that study online decision making problems, such as OCO, use numerical simulations to evaluate the impact of prediction noise, e.g., [2, 8, 59, 66].

The papers that do consider analytic models often use either i.i.d. prediction noise or worst-case bounds on prediction errors for tractability. An exception is the recent work [22, 32] which introduces a model for prediction noise that captures three important features of real predictors: (i) it allows for correlations in prediction errors (both short range and long range); (ii) the quality of predictions decreases the further in the future we try to look ahead; and (iii) predictions about the future are refined as time passes.

Recall that we model prediction error via the following equation:

$$y_t - y_{t|\tau} = \sum_{s=\tau+1}^{t} f(t-s)e(s), \tag{4.2}$$

where $y_{t|\tau}$ is the prediction of y_t made at time $\tau < t$. This model characterizes prediction error as white noise being passed through a causal filter. In particular, the prediction error is a weighted linear combination of per-step noise terms $e(s)$ with weights $f(t-s)$, where f is a deterministic impulse response function. The noise terms $e(s)$ are assumed to be uncorrelated with mean zero and positive definite covariance R_e; let $\sigma^2 = \operatorname{tr}(R_e)$. Further, the impulse response function f is assumed to satisfy $f(0) = I$ and $f(t) = 0$ for $t < 0$.

Note that i.i.d. prediction noise can be recovered by imposing that $f(0) = I$ and $f(t) = 0$ for all $t \neq 0$. Further, the model can represent prediction errors that arise from classical filters such as Wiener filters and Kalman filters (see [22]). In both cases the impulse response function decays as $f(s) \sim \eta^s$ for some $\eta < 1$.

These examples highlight that the form of the impulse response function captures the degree of short-term/long-term correlation in prediction errors. The form of the correlation structure plays a key role in the performance results we prove, and its impact can be captured through the following definition. For any $k > 0$, let $\|f_k\|^2$ be the two norm square of prediction error covariance over k steps of prediction, i.e.,

$$\|f_k\|^2 = \mathrm{tr}(\mathbb{E}[\delta y_k \delta y_k^T]) = \mathrm{tr}(R_e \sum_{s=0}^{k} f(s)^T f(s)), \tag{4.3}$$

where $\delta y_k^T = y_{t+k} - y_{t+k|t} = \sum_{s=t+1}^{t+k} f(t+k-s)e(s)$. Derivation of (4.3) can be found in Appendix 4.B Equation (4.19).

The competitive difference

For any algorithm ALG that comes up with feasible actions $x_{ALG,t} \in F, \forall t$, the cost of the algorithm over the horizon can be written as

$$\mathrm{cost}(ALG) = \sum_{t=1}^{T} h(x_{ALG,t}, y_t) + \beta \left\|x_{ALG,t} - x_{ALG,t-1}\right\|. \tag{4.4}$$

We compare the performance of our online algorithm against the optimal offline algorithm OPT, which makes the optimal decision with full knowledge of the trajectory of y_t.

$$\mathrm{cost}(OPT) = \min_{x_t \in F} \sum_{t=1}^{T} h(x_t, y_t) + \beta \|x_t - x_{t-1}\|. \tag{4.5}$$

The results in this chapter bound the *competitive difference* of algorithms for OCO with switching costs and prediction noise. Informally, the competitive difference is the additive gap between the cost of the online algorithm and the cost of the offline optimal.

To define the competitive difference formally in our setting we need to first consider how to specify the instance. To do this, let us first return to the specification of the prediction model in (4.2) and expand the summation all the way to time zero. This expansion highlights that the process y_t can be viewed as a random deviation around the predictions made at time zero, $y_{t|0} := \hat{y}_t$, which are specified externally to the model:

$$y_t = \hat{y}_t + \sum_{s=1}^{t} f(t-s)e(s). \tag{4.6}$$

Thus, an instance can be specified either via the process y_t or via the initial predictions $\hat{y}_t$, and then the random noise from the model determines the other. The latter is preferable for analysis, and thus we state our definition of competitive difference (and our theorems) using this specification.

Definition 7. *We say an online algorithm ALG has (expected)* ***competitive difference*** *at most $\rho(T)$ if:*

$$\sup_{\hat{y}} \mathbb{E}_e \left[\text{cost}(ALG) - \text{cost}(OPT)\right] \leq \rho(T). \tag{4.7}$$

Note that the expectation in the definition above is with respect to the prediction noise, $(e(t))_{t=1}^T$, and so both terms cost(ALG) and cost(OPT) are random. Unlike ALG, the offline optimal algorithm OPT knows each exact realization of e before making the decision.

Importantly, though we specify our results in terms of the competitive difference, it is straightforward to convert them into results about the competitive ratio and regret, which are more commonly studied in the OCO literature. Recall that the **competitive ratio** bounds the ratio of the algorithm's cost to that of OPT, and the **regret** bounds the difference between the algorithm's cost and the offline *static* optimal.

Converting a result on the competitive difference into a result on the competitive ratio requires lower bounding the offline optimal cost, and such a bound can be found in Theorem 6 of [56]. Similarly, converting a result on the competitive difference into a result on the regret requires lower bounding the offline static optimal cost, and such a bound can be found in Theorem 2 of [22].

4.2 Committed horizon control

There is a large literature studying algorithms for online convex optimization (OCO), both with the goal of designing algorithms with small regret and algorithms with small competitive ratio.

These algorithms use a wide variety of techniques. For example, there are numerous algorithms that maintain sub-linear regret, e.g., *online gradient descent* (OGD) based algorithms [40, 90] and *Online Newton Step and Follow the Approximate Leader* algorithms [40]. (Note that the classical setting does not consider switching costs; however, [9] shows that similar regret bounds can be obtained when switching costs are considered.) By contrast, there only exist algorithms that achieve constant competitive ratio in limited settings, e.g., [58] shows that, when F is a one-dimensional normed space, there exists a deterministic online algorithm that is 3-competitive. This is because, in general, obtaining a constant competitive ratio is impossible in the worst-case: [13] has shown that any deterministic algorithm must be $\Omega(n)$-competitive given metric decision space of size n and [12] has shown that any randomized algorithm must be $\Omega(\sqrt{\log n/\log\log n})$-competitive.

However, all of the algorithms and results described above are in the worst-case setting and do not consider algorithms that have noisy predictions available. Given noisy predictions, the most natural family of algorithms to consider come from the family of Model Predictive Control (MPC) algorithms, which is a powerful, prominent class of algorithms from the control community. In fact, the only analytic results for OCO problems with predictions to this point have come from algorithms inspired by MPC, e.g., [11, 18, 50, 83]. (Note that there is an abundance of literature on such algorithms in control theory, e.g., [33, 63] and the references therein, but the analysis needed for OCO is different than from the stability analysis provided by the control literature.)

To this point, two promising candidate algorithms have emerged in the context of OCO: Receding Horizon Control (RHC) [51] and Averaging Fixed Horizon Control (AFHC) [56]. We discuss these two algorithms in Section 4.2 below and then introduce our novel class of Committed Horizon Control (CHC) algorithms, which includes both RHC and AFHC as special cases, in Section 4.2. The class of CHC algorithms is the focus of this Chapter.

Two promising algorithms

At this point the two most promising algorithms for integrating noisy predictions into solutions to OCO problems are RHC and AFHC.

Receding Horizon Control (RHC): RHC operates by determining, at each timestep t, the optimal actions over the window $(t+1, t+w)$, given the starting state x_t and a prediction window (horizon) of length w.

To state this more formally, let $y_{\cdot|\tau}$ denote the vector $(y_{\tau+1|\tau}, \ldots, y_{\tau+w|\tau})$, the prediction of y in a w timestep prediction window at time τ. Define $X^{\tau+1}(x_\tau, y_{\cdot|\tau})$ as the vector in F^w indexed by $t \in \{\tau+1, \ldots, \tau+w\}$, which is the solution to

$$\begin{aligned} \min_{x_{\tau+1},\ldots,x_{\tau+w}} \quad & \sum_{t=\tau+1}^{\tau+w} h(x_t, y_{t|\tau}) + \sum_{t=\tau+1}^{\tau+w} \beta \left\| x_t - x_{t-1} \right\|, \\ \text{subject to} \quad & x_t \in F. \end{aligned} \tag{4.8}$$

RHC has a long history in the control theory literature, e.g., [11, 18, 33, 63]. However, there are few results known in the OCO literature, and most such results

Algorithm 4.1 Receding Horizon Control

For all $t \leq 0$, set $x_{RHC,t} = 0$. Then, at each timestep $\tau \geq 0$, set

$$x_{RHC,\tau+1} = X_{\tau+1}^{\tau+1}(x_{RHC,\tau}, y_{\cdot|\tau}) \tag{4.9}$$

are negative. In particular, the competitive ratio of RHC with perfect lookahead window w is $1 + O(1/w)$ in the one-dimensional setting. The performance is not so good in the general case. In particular, outside of the one-dimensional case the competitive ration of RHC is $1 + \Omega(1)$, i.e., the competitive ratio does not decrease to 1 as the prediction window w increases in the worst case [58].

Averaging Fixed Horizon Control (AFHC): AFHC provides an interesting contrast to RHC. RHC ignores all history – the decisions and predictions that led it to be in the current state – while AFHC constantly looks both backwards and forwards. Specifically, AFHC averages the choices made by Fixed Horizon Control (FHC) algorithms. In particular, AFHC with prediction window size w averages the actions of w FHC algorithms, each with different predictions available to it. At time t, a FHC algorithm determines the optimal actions $x_{t+1}, \ldots, x_{t+w}$ given a prediction window (horizon) of length w as done in RHC. But, then FHC implements all actions in the trajectory $x_{t+1}, \ldots, x_{t+w}$ instead of just the first action x_t. Fixed Horizon Control algorithms are individually more naive than RHC, but by averaging them AFHC can provide improved worst-case performance compared to RHC. To define the algorithm formally, let

$$\Omega_k = \{i : i \equiv k \mod w\} \cap [-w+1, T] \text{ for } k = 0, \ldots, w-1.$$

Note that, for $k \geq 1$, the algorithm starts from $\tau = k - w$ rather than $\tau = k$ in order

Algorithm 4.2 Fixed Horizon Control, version k

$FHC^{(k)}(w)$, is defined in the following manner. For all $t \leq 0$, set $x_{FHC,t}^{(k)} = 0$. At timeslot $\tau \in \Omega_k$ (i.e., before $y_{\tau+1}$ is revealed), for all $t \in \{\tau+1, \ldots, \tau+w\}$, use (4.8) to set

$$x_{FHC,t}^{(k)} = X_t^{\tau+1}\left(x_{FHC,\tau}^{(k)}, y_{\cdot|\tau}\right). \tag{4.10}$$

to calculate $x_{FHC,t}^{(k)}$ for $t < k$.

While individual FHC can have poor performance, surprisingly, by averaging different versions of FHC we can obtain an algorithm with good performance guarantee. Specifically, $AFHC$ is defined as follows.

Algorithm 4.3 Averaging Fixed Horizon Control

```
For all $k$, at each timeslot $\tau \in \Omega_k$, use $FHC^{(k)}$ to determine $x^{(k)}_{FHC,\tau+1}, \ldots, x^{(k)}_{FHC,\tau+w}$, and for $t = 1, \ldots, T$, set
    $$x_{AFHC,t} = \frac{1}{w}\sum_{k=0}^{w-1} x^{(k)}_{FHC,t}. \quad (4.11)$$
```

In contrast to RHC, AFHC has a competitive ratio of $1 + O(1/w)$ regardless of the dimension of the action space in the perfect lookahead model [56][1]. This improvement of AFHC over RHC is illustrated in Figure 4.1(a), which shows for a specific setting with perfect lookahead, AFHC approaches the offline optimal with increasing prediction window size while RHC is relatively constant. (The setting used for the figure uses a simple model of a data center with a multi-dimensional action space, and is described in Appendix 4.A.)

Comparing RHC and AFHC: Despite the fact that the worst-case performance of AFHC is dramatically better than RHC, RHC provides better performance than AFHC in realistic settings when prediction can be inaccurate in the lookahead window. For example, Figure 4.1(b) highlights that RHC can outperform AFHC by an arbitrary amount if the predictions are noisy. Specifically, if we make predictions accurate for a small window γ and then inaccurate for the remaining $(w - \gamma)$ steps of the lookahead window, AFHC is affected by the inaccurate predictions whereas RHC only acts on the correct ones. The tradeoff between the worst-case bounds and average-case performance across AFHC and RHC is also evident in the results shown in Figure 3 of [56].

The contrast between Figure 4.1a and 4.1b highlights that, at this point, it is unclear when one should use AFHC/RHC. In particular, AFHC is more robust but RHC may be better in many specific settings. Further, the bounds we have described so far say nothing about the impact of noise on the performance (and comparison) of these algorithms.

A general class of algorithms

The contrast between the performance of receding horizon control (RHC) and averaging fixed horizon control (AFHC) in worst-case and practical settings is a

[1] Note that this result assumes that there exists $e_0 > 0$, s.t. $h(x, y) \geq e_0 \cdot x, \forall x, y$, and the switching cost is $\beta \cdot (x_t - x_{t-1})^+$ where $(x)^+ = \max(x, 0)$.

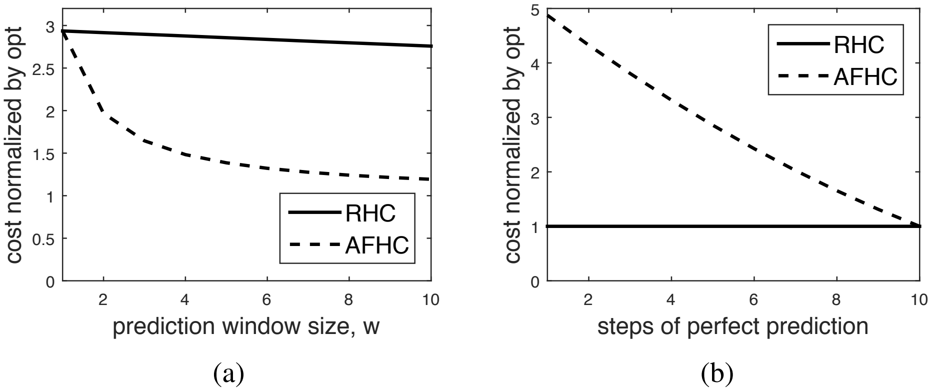


Figure 4.1: Total cost of RHC and AFHC, normalized by the cost of the offline optimal, versus: (a) prediction window size, (b) number of steps of perfect prediction with $w = 10$. Note (a) and (b) were produced under different cost settings, see Appendix 4.A.

consequence of the fact that RHC is entirely "forward looking" while AFHC keeps an "eye on the past". However, both algorithms are extreme cases in that RHC does not consider any information that led it to its current state, while AFHC looks back at w FHC algorithms – every set of predictions that led to the current state.

One way to view this difference between RHC and AFHC is in terms of *commitment*. In particular, AFHC has FHC algorithms that commit to the w decisions at each timestep and then the final choice of the algorithm balances these commitments by averaging across them. In contrast, RHC commits only one step at a time.

Building on this observation, we introduce the class of Committed Horizon Control (CHC) algorithms. The idea behind the class is to allow commitment of a fixed number, say v, of steps. The minimal level of commitment, $v = 1$, corresponds to RHC and the maximal level of commitment, $v = w$, corresponds to AFHC. Thus, the class of CHC algorithms allows variation between these extremes.

Formally, to define the class of CHC algorithms we start by generalizing the class of FHC algorithms to allow limited commitment. An FHC algorithm with commitment level v uses a prediction window of size w but then executes (commits to) only the first $v \in [1, w]$ actions which can be visualized by Figure 4.2. To define this formally, let

$$\Psi_k = \{i : i \equiv k \mod v\} \cap [-v+1, T] \text{ for } k = 0, \dots, v-1.$$

Fixed horizon control with lookahead window w and commitment level v, $FHC^{(k)}(v, w)$, is defined in the following manner. For notational convenience, we write $x^{(k)} \equiv$

$x^{(k)}_{FHC(v,w)}$. Note that, for $k \geq 1$, the algorithm starts from $\tau = k - v$ rather than $\tau = k$

Algorithm 4.4 FHC with Limited Commitment

For all $t \leq 0$, set $x^{(k)}_{FHC,t} = 0$.
At timeslot $\tau \in \Psi_k$ (i.e., before $y_{\tau+1}$ is revealed), for all $t \in \{\tau + 1, \ldots, \tau + v\}$, use (4.8) to set

$$x^{(k)}_t = X^{\tau+1}_t \left(x^{(k)}_\tau, y_{\cdot|\tau} \right). \tag{4.12}$$

in order to calculate $x^{(k)}_t$. We can see that FHC with limited commitment is very similar to FHC as both use (4.8) to plan w timesteps ahead, but here only the first v steps are committed to action.

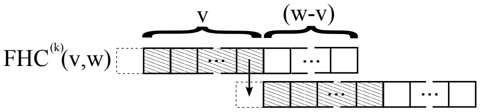


Figure 4.2: Fixed Horizon Control with commitment level v: optimizes once every v timesteps for the next w timesteps and commits to use the first v of them.

$CHC(v, w)$, the CHC algorithm with prediction window w and commitment level v, averages over v FHC algorithms with prediction window w and commitment level v. Figure 4.3 provides an overview of CHC. For conciseness in the rest of the chapter, we will use $x^{(k)}_t$ to denote the action decided by FHC$^{(k)}(v, w)$ at time t.

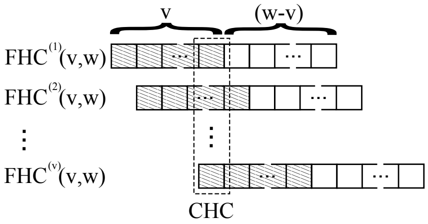


Figure 4.3: Committed Horizon Control: at each timestep, it averages over all v actions defined by the v FHC algorithms with limited commitment.

Algorithm 4.5 Committed Horizon Control

At each timeslot $\tau \in \Psi_k$, use $FHC^{(k)}(v, w)$ to determine $x^{(k)}_{\tau+1}$, ..., $x^{(k)}_{\tau+v}$, and at timeslot $t \in 1, ..., T$, $CHC(v, w)$ sets

$$x_{CHC,t} = \frac{1}{v} \sum_{k=0}^{v-1} x^{(k)}_t. \tag{4.13}$$

RHC and AFHC are the extreme levels of commitment in CHC policies and, as we see in the analysis that follows, it is often beneficial to use intermediate levels of commitment depending on the structure of prediction noise.

4.3 Optimal commitment level

We now present the main technical results of this chapter, which analyze the performance of committed horizon control (CHC) algorithms and address several open challenges relating to the analysis of receding horizon control (RHC) and averaging fixed horizon control (AFHC). In this section we characterize the average case performance of CHC as a function of the commitment level v of the policy and properties of the prediction noise, i.e., the variance of prediction noise $e(s)$ and the form of the correlation structure, $f(s)$. Concentration bounds are discussed in Section 4.4. All proofs are presented in Appendix 4.B.

Our main result establishes *bounds on the competitive difference of CHC under noisy predictions*. Since CHC generalizes RHC and AFHC, our result also provides the *first* analysis of RHC with noisy predictions and further enables a comparison between RHC and AFHC based on the properties of the prediction noise.

Previously, only AFHC has been analyzed in the case of OCO with noisy predictions [22]. Further, the analysis of AFHC in [22] depends delicately on the structure of the algorithm and thus cannot be generalized to other policies, such as RHC. Our results here are made possible by a *novel* analytic technique that transforms the control strategy employed by OPT, one commitment length at a time, to the control strategy employed by $FHC^{(k)}(v, w)$. At each intermediate step, we exploit the optimality of $FHC^{(k)}(v, w)$ within the commitment length to bound the difference in costs; the sum of these costs over the entire transformation gives a bound on the difference in costs between OPT and $FHC^{(k)}(v, w)$. We then exploit Jensen's inequality to extend this bound on competitive difference to CHC.

Theorem 4.1 below presents our main result characterizing the performance of CHC algorithms under noisy predictions for functions that are α-Hölder continuous in the second argument; in particular, $\alpha = 1$ corresponds to the class of function that is Lipschitz continuous in the second argument.

Theorem 4.1. *Assuming that the prediction error follows* (2.6)*, then for h that is α-Hölder continuous in the second argument, we have*

$$\mathbb{E}\text{cost}(CHC) \leq \mathbb{E}\text{cost}(OPT) + \frac{2T\beta D}{v} + \frac{2GT}{v}\sum_{k=0}^{v-1} \|f_k\|^{\alpha}. \tag{4.14}$$

Note that, while Theorem 4.1 is stated in terms of the competitive difference, it can easily be converted into results about the competitive ratio and regret as explained in Section 4.1.

There are two terms in the bound on the competitive difference of CHC: (i) The first term $\frac{2T\beta D}{v}$ can be interpreted as the price of switching costs due to limited commitment; this term decreases as the commitment level v increases. (ii) The second term $\frac{2GT}{v}\sum_{k=0}^{v-1}\|f_k\|^{\alpha}$ represents the impact of prediction noise on the competitive difference and can be characterized by $\|f_k\|$ (defined in (4.3)), which is impacted by both the variance of $e(s)$ and the structural form of the prediction noise correlation, $f(s)$.

Theorem 4.1 allows us to immediately analyze the performance of RHC and AFHC as they are special cases of CHC. We present our results comparing the performance of RHC and AFHC by analyzing how the optimal level of commitment, v, depends on properties of the prediction noise.

In order to make concrete comparisons, it is useful to consider specific forms of prediction noise. Here, we consider four cases: (i) i.i.d. prediction noise, (ii) prediction noise with long range correlation, (iii) prediction noise with short range correlation, and (iv) prediction noise with exponentially decaying correlation. All four cases can be directly translated to assumptions on the correlation structure, $f(\cdot)$. Recall that many common predictors, e.g., Wiener and Kalman filters, yield f that is exponentially decaying.

i.i.d. prediction noise. The assumption of i.i.d. prediction noise is idealistic since it only happens when the forecast for y_t is optimal based on the information prior to time t for all $t = 1, \ldots, T$ [43]. However, analysis of the i.i.d. noise is instructive and provides a baseline for comparison with more realistic models. In this case, Theorem 4.1 can be specialized as follows. Recall that $\mathbb{E}[e(s)e(s)^T] = R_e$, and $\operatorname{tr}(R_e) = \sigma^2$.

Corollary 4.2. *Consider i.i.d. prediction error, i.e.,*

$$f(s) = \begin{cases} I, & s = 0 \\ 0, & \textit{otherwise}. \end{cases}$$

If h satisfies is α-Hölder continuous in the second argument, then the expected competitive difference of CHC is upper bounded by

$$\mathbb{E}\mathrm{cost}(CHC) \leq \mathbb{E}\mathrm{cost}(OPT) + \frac{2T\beta D}{v} + 2GT\sigma^{\alpha},$$

which is minimized when $v^* = w$.

This can be proved by simply applying the form of $f(s)$ to (4.14). Corollary 4.2 highlights that, in the i.i.d. case, the level of commitment that minimizes the competitive difference always coincides with the lookahead window w, independent of all other parameters. This is intuitive since, when prediction noise is i.i.d., increasing commitment level does not increase the cost due to prediction errors. Combined with the fact that increasing the commitment level decreases the costs incurred by switching, we can conclude that *AFHC is optimal* in the i.i.d. setting.

Long range correlation. In contrast to i.i.d. prediction noise, another extreme case is when prediction noise has strong correlation over a long period of time. This is pessimistic and happens when past prediction noise has far-reaching effects on the prediction errors in the future, i.e., the current prediction error is sensitive to errors in the distant past. In this case, prediction only offers limited value since prediction errors accumulate. For long range correlation, we can apply Theorem 4.1 as follows.

Corollary 4.3. *Consider prediction errors with long range correlation such that*

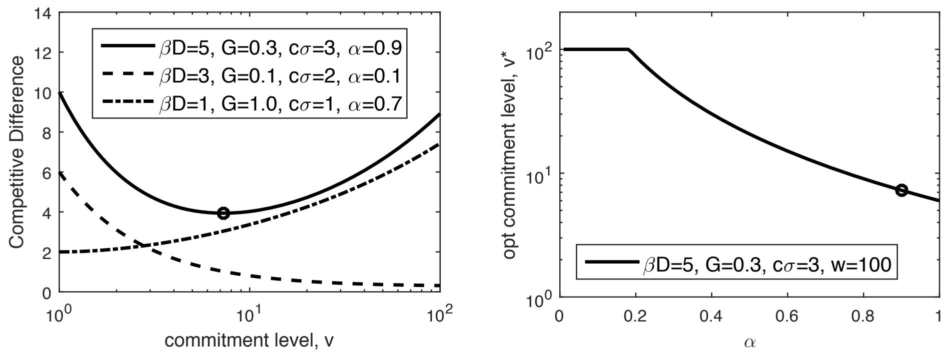
$$\|f(s)\|_F = \begin{cases} c, & s \leq L \\ 0, & s > L, \end{cases}$$

where $L > w$. If h is α-Hölder continuous in the second argument, the expected competitive difference of CHC is upper bounded by

$$\begin{aligned}\mathbb{E}\mathrm{cost}(CHC) - \mathbb{E}\mathrm{cost}(OPT) \leq\ & \frac{2T\beta D(\alpha+2) - 4GTc^{\alpha}\sigma^{\alpha}}{\alpha+2} v^{-1} \\ & + \frac{2^{3+\frac{\alpha}{2}}GTc^{\alpha}\sigma^{\alpha}}{\alpha+2} v^{\alpha/2}.\end{aligned}$$

If $\frac{\beta D}{Gc^{\alpha}\sigma^{\alpha}} > \alpha(2w)^{1+\frac{\alpha}{2}} + 2$, then $v^* = w$; if $\frac{\beta D}{Gc^{\alpha}\sigma^{\alpha}} < \frac{2}{\alpha+2}$, then $v^* = 1$, otherwise v^* is in between 1 and w.

Corollary 4.3 highlights that, in the case of long range correlation, the level of commitment that minimizes the competitive difference depends on the variance σ^2, the switching cost β, the smoothness G, α, and diameter of the action space D.



(a) (b)

Figure 4.4: Illustration of Corollary 4.3, for long range dependencies. (a) shows the time averaged expected competitive difference as a function of the commitment level, and (b) shows the optimal commitment level as a function of α.

The term $\frac{\beta D}{Gc^\alpha \sigma^\alpha}$ can be interpreted as a measure of the relative importance of the switching cost and the prediction loss. If $\frac{\beta D}{Gc^\alpha \sigma^\alpha} = \frac{2}{\alpha+2} \in O(1)$, i.e., the one step loss due to prediction error is on the order of the switching cost, then $v^* = 1$ and RHC optimizes the performance bound; if $\frac{\beta D}{Gc^\alpha \sigma^\alpha} = \alpha(2w)^{1+\frac{\alpha}{2}} + 2 \in \Omega(w)$, then $v^* = w$ and $AFHC$ optimizes the performance bound. Otherwise, $v^* \in (1, w)$.

We illustrate these results in Figure 4.4 which plots the competitive difference as a function of the commitment level for various parameter values. The case for the dashed line satisfies $\frac{\beta D}{Gc^\alpha \sigma^\alpha} > \alpha(2w)^{1+\frac{\alpha}{2}} + 2$ and shows competitive difference decreases with increasing levels of commitment. Here, the window size is 100, and thus AFHC minimizes the competitive difference, validating Corollary 4.3. The dot-dashed line satisfies $\frac{\beta D}{Gc^\alpha \sigma^\alpha} < \frac{2}{\alpha+2}$ and shows the increase in competitive difference with commitment, highlighting that RHC is optimal. The solid line does not satisfy either of these conditions and depicts the minimization of competitive difference at intermediate levels of commitment (marked with a circle). Figure 4.4b illustrates the relationship between α and the optimal commitment level v^* (marked with a circle that corresponds to the same v^* as in Figure 4.4a). As α increases, the prediction loss increases, and thus the optimal commitment level decreases to allow for updated predictions.

Short range correlation. Long range correlation is clearly pessimistic as it assumes that the prediction noise is always correlated within the lookahead window. Here, we study another case where prediction noise can be correlated, but only within a small interval that is less than the lookahead window w. This is representative of scenarios where only limited past prediction noises affect the current prediction. For such short range correlation, Theorem 4.1 gives us:

Corollary 4.4. *Consider prediction errors with short range correlation such that*

$$\|f(s)\|_F = \begin{cases} c, & s \leq L \\ 0, & s > L, \end{cases}$$

where $L \leq w$. If h is α-Hölder continuous in the second argument, the expected competitive difference of CHC is upper bounded by:

if $v > L$

$$\begin{aligned} \mathbb{E}\text{cost}(CHC) - \mathbb{E}\text{cost}(OPT) \leq & \frac{2T\beta D}{v} + 2GT(c\sigma)^\alpha (L+1)^{\alpha/2} \\ & - \frac{2GT}{v}\frac{(c\sigma)^\alpha}{\alpha+2}((L+1)^{\alpha/2}(\alpha L - 2) + 1); \end{aligned}$$

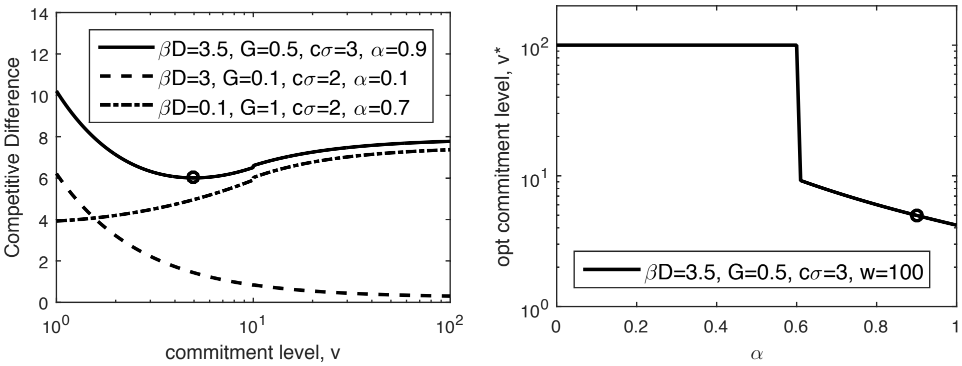
if $v \leq L$

$$\begin{aligned} \mathbb{E}\text{cost}(CHC) - \mathbb{E}\text{cost}(OPT) \leq & \frac{2T\beta D}{v} \\ & + \frac{4GTc^\alpha\sigma^\alpha}{v(\alpha+2)}((v+1)^{(\alpha+2)/2} - 1). \end{aligned}$$

If $\frac{\beta D}{Gc^\alpha\sigma^\alpha} > H(L)$, where $H(L) = \frac{1}{\alpha+2}\big((L+1)^{\alpha/2}(\alpha L - 2) + 1\big)$, then $v^* = w$; if $\frac{\beta D}{Gc^\alpha\sigma^\alpha} < \min(H(L), \frac{2}{\alpha+2})$, then $v^* = 1$, otherwise v^* is in between 1 and w.

Corollary 4.4 shows that the structure of the bound on the competitive difference itself depends on the relative values of v and L. In terms of the optimal commitment level, Corollary 4.4 shows that, similar to Corollary 4.3, the term $\frac{\beta D}{Gc^\alpha\sigma^\alpha}$ comes into play; however, unlike Corollary 4.3 (where $L > w$), the optimal commitment level now also depends on the length of the interval, L, within which prediction errors are correlated. Note that $H(L)$ is increasing in L. If $\frac{\beta D}{Gc^\alpha\sigma^\alpha} > H(L)$, i.e., the prediction loss and L are small compared to the switching cost, then $v^* = w$ and thus $AFHC$ optimizes the performance bound. On the other hand, if the prediction loss and L are large compared to the switching cost, then $v^* = 1$, and thus RHC optimizes the bound; otherwise, v^* lies is between 1 and w, and thus intermediate levels of commitment under CHC perform better than $AFHC$ and RHC.

Note that when prediction noise is i.i.d., we have $L = 0$ and $H(L) < 0$; hence we have $\frac{\beta D}{Gc^\alpha\sigma^\alpha} > H(L)$ and thus $v^* = w$, which corresponds to the conclusion of Corollary 4.2.



(a) (b)

Figure 4.5: Illustration of Corollary 4.4, for short range correlations. (a) shows the time averaged expected competitive difference as a function of the commitment level, and (b) shows the optimal commitment level as a function of α.

We illustrate these results in Figure 4.5a, which plots the competitive difference as a function of the commitment for various parameter values. The dashed line satisfies $\frac{\beta D}{Gc^{\alpha}\sigma^{\alpha}} > H(L)$ and shows the drop in competitive difference with increasing levels of commitment. The competitive difference is lowest when the commitment level is 100, which is also the window size, thus validating the optimality of AFHC as per Corollary 4.4. The dot-dashed line satisfies $\frac{\beta D}{Gc\sigma} < \min(H(L), \frac{2}{\alpha+2})$ and shows the increase in competitive difference with commitment, highlighting that RHC is optimal. The solid line does not satisfy either of these conditions and depicts the minimization of competitive difference at intermediate levels of commitment. Figure 4.5b illustrates the relationship between α and the optimal commitment level v^*. As α increases, loss due to prediction noise increases; as a result, v^* decreases.

Exponentially decaying correlation. Exponentially decaying correlation is perhaps the most commonly observed model in practice and is representative of predictions made via Wiener [84] or Kalman [48] filters. For clarity of illustration we consider the case of $\alpha = 1$ here. In this case, Theorem 4.1 results in the following corollary.

Corollary 4.5. *Consider prediction errors with exponentially decaying correlation, i.e., there exists $a < 1$, such that*

$$\|f(s)\|_F = \begin{cases} ca^s, & s \geq 0 \\ 0, & s < 0. \end{cases}$$

If h is 1-Hölder continuous, then the expected competitive difference of CHC is *upper bounded by*

$$\mathbb{E}\text{cost}(CHC) - \mathbb{E}\text{cost}(OPT) \leq \frac{2T\beta D}{v} + \frac{2GTc\sigma}{1-a^2} - \frac{a^2(1-a^{2v})GTc\sigma}{v(1-a^2)^2}.$$

When $\frac{\beta D}{Gc\sigma} \geq \frac{a^2}{2(1-a^2)}$ the commitment that minimizes the performance bound is $v^* = w$, i.e., AFHC minimizes the performance bound. When $\frac{\beta D}{Gc\sigma} < \frac{a^2}{2(1+a)}$, $v^* = 1$, i.e., RHC minimizes the performance bound.

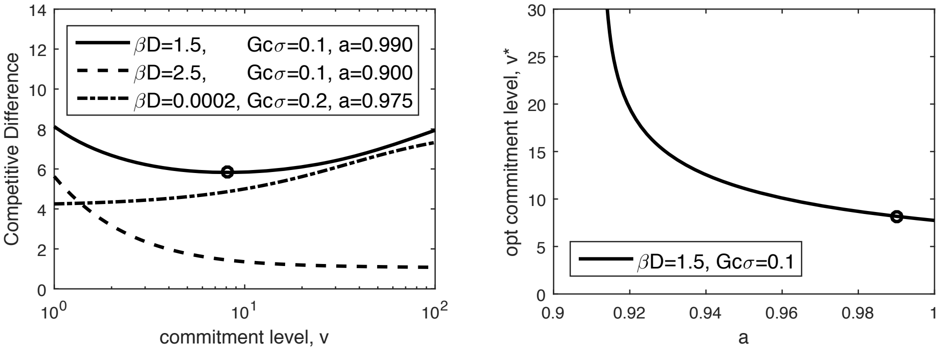
Corollary 4.5 shows that when the prediction noise σ and the correlation decay a are small, the loss due to switching costs is dominant, and thus commitment is valuable; on the other hand, when σ and a are large, then the loss due to inaccurate predictions is dominant, and thus a smaller commitment is preferable to exploit more updated predictions.

We illustrate these results in Figure 4.6a, which plots the competitive difference as a function of the commitment for various parameter values. The dashed line satisfies $\frac{\beta D}{Gc\sigma} > \frac{a^2}{2(1-a^2)}$ and shows the drop in competitive difference with increasing levels of commitment. The competitive difference is lowest when the commitment level is 100, which is also the window size, thus validating the optimality of AFHC as per Corollary 4.5. The dot-dashed line satisfies $\frac{\beta D}{Gc\sigma} > \frac{a^2}{2(1+a)}$ and shows the increase in competitive difference with commitment, highlighting that RHC is optimal. The solid line does not satisfy either of these conditions and depicts the minimization of competitive difference at intermediate levels of commitment. Figure 4.6b illustrates the relationship between a and the optimal commitment level v^*. As a increases, correlation decays more slowly, and thus the loss due to prediction noise becomes dominant; as a result, v^* decreases.

Strong convexity. All of our results to this point depend on the diameter of the action space D. While this is common in OCO problems, e.g., [47, 90], it is not desirable.

Our last result in this section highlights that it is possible to eliminate the dependence on D by making a stronger structural assumption on h – strong convexity. In particular, we say that $h(\cdot)$ is $m-$strongly convex in the first argument w.r.t. the norm of the switching cost $\|\cdot\|$ if $\forall x_1, x_2, y$,

$$h(x_1, y) - h(x_2, y) \geq \langle \partial_x h(x_2, y) \cdot (x_1 - x_2) \rangle + \frac{m}{2} \|x_1 - x_2\|^2 .$$



(a) (b)

Figure 4.6: Illustration of Corollary 4.5, for exponentially decaying correlations. (a) shows the time averaged expected competitive difference as a function of the commitment level, and (b) shows the optimal commitment level as a function of the decay parameter, a.

Strong convexity is used in the online learning literature to obtain performance bounds that are independent of the diameter of action space, see, e.g., [41, 76]. Under the assumption of strong convexity, we obtain the following.

Theorem 4.6. *If h is m-strongly convex in the first argument with respect to $\|\cdot\|$ and α-Hölder continuous in the second argument, we have*

$$\mathbb{E}\text{cost}(CHC) - \mathbb{E}\text{cost}(OPT) \leq \frac{2\beta^2 T}{mv} + 2GT \sum_{k=0}^{v-1} \|f_k\|^\alpha .$$

Theorem 4.6 is useful when the diameter of the feasible set D is large or unbounded; when D is small, we can apply Theorem 4.1 instead. As above, it is straightforward to apply the techniques in Corollaries 4.2 – 4.5 to compute v^* for strongly convex h under different types of prediction noise[2].

4.4 Concentration bounds

Our results to this point have focused on the performance of CHC algorithms *in expectation*. In this section, we establish bounds on the *distribution* of costs under CHC algorithms. In particular, we prove that, under a mild additional assumption, the likelihood of cost exceeding the average case bounds proven in Section 4.3 decays exponentially.

[2] We only need to change βD with β^2/m in the bounds of the corollaries to draw parallel conclusions.

For simplicity of presentation, we state and prove the concentration result for CHC when the online parameter y is one-dimensional. In this case, $R_e = \sigma^2$, and the correlation function $f : \mathbb{N} \to R$ is a scalar valued function. The results can be generalized to the multi-dimensional setting at the expense of considerable notational complexity in the proofs.

Additionally, for simplicity of presentation we assume (for this section only) that $\{e(t)\}_{t=1}^T$ are uniformly bounded, i.e., $\exists \epsilon > 0$, s.t. $\forall t$, $|e(t)| < \epsilon$. Note that, with additional effort, the boundedness assumption can be relaxed to $e(t)$ being subgaussian, i.e., $\mathbb{E}[\exp(e(t)^2/\epsilon^2)] \leq 2$, for some $\epsilon > 0$.[3]

Given $\{\hat{y}_t\}_{t=1}^T$, the competitive difference of CHC is a random variable that is a function of the prediction error $e(t)$. To state our concentration results formally, let $V_1 T$ be the upper bound of the expected competitive difference of CHC in (4.14), i.e., $V_1 T = \frac{2T\beta D}{v} + \frac{2GT}{v} \sum_{k=1}^{v} \|f_k\|^\alpha$.

Theorem 4.7. *Assuming that the prediction error follows* (2.6)*, and h is α-Hölder continuous in the second argument, we have*

$$\begin{aligned}&\mathbb{P}(\text{cost}(CHC) - \text{cost}(OPT) > V_1 T + u)\\ \leq &\exp\left(\frac{-u^2\alpha^2}{2^{1+2\alpha} G^2 \varepsilon^2 T F(v)}\right),\end{aligned}$$

for any $u > 0$, where $F(v) = \left(\frac{1}{v}\sum_{k=0}^{v-1}(v-k)^\alpha |f(k)|^\alpha\right)^2$.

This result shows that the competitive difference has a sub-Gaussian tail, which decays much faster than the normal large deviation bounds obtained by bounding moments, i.e., Markov Inequality, the rate of decay is dependent on the sensitivity of h to disturbance in the second argument (G, α), the size of variation (ε), and the correlation structure $(F(v))$. This is illustrated in Figure 4.7, where we show the distribution of the competitive difference of CHC under different prediction noise correlation assumptions. We can see that, for prediction noise that decays fast (i.i.d. and exponentially decaying noise with small a) in Figure 4.7a, the distribution is tightly concentrated around the mean, whereas for prediction noise that are fully correlated (short range correlation and long range correlation) in Figure 4.7b, the distribution is more spread out.

[3] This involves more computation and worse constants in the concentration bounds. Interested readers are referred to Theorem 12 and the following remark of [14] for a way to generalize the concentration bound.

If we consider the time-averaged competitive difference, or the *regret* against the offline optimal, we can equivalently state Theorem 4.7 as follows.

Corollary 4.8. *Assuming that the prediction error follows* (2.6)*, and h is α-Hölder continuous, then probability that the competitive difference of CHC exceeds V_1 can be bounded by*

$$\mathbb{P}\left(\frac{1}{T}\left[\text{cost}(CHC) - \text{cost}(OPT)\right] > V_1 + u\right)$$
$$\leq \exp\left(\frac{-u^2}{2^{1+2\alpha}G^2\varepsilon^{2\alpha}F(v)/T}\right),$$

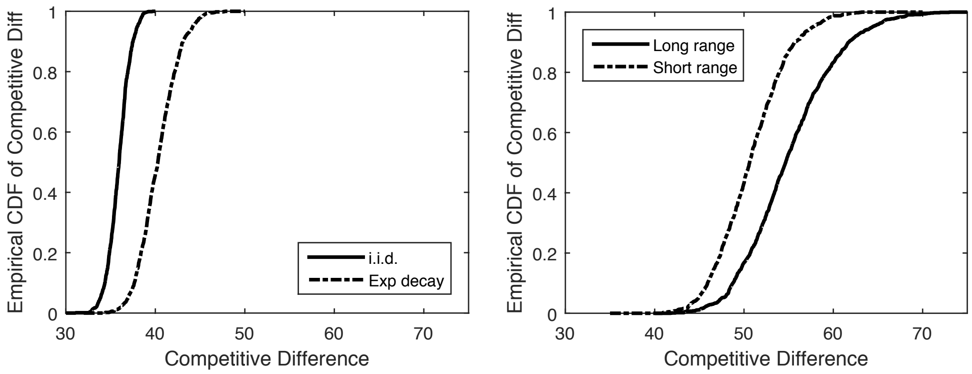
where $F(v) = \left(\frac{1}{v}\sum_{k=0}^{v-1}(v-k)|f(k)|^{\alpha}\right)^2$. Assuming $f(s) \leq C$ for $s = 0, \ldots, v$, then $\lim_{T\to\infty} F(v)/T = 0$ *if either $v \in O(1)$, or $f(s) \leq c\eta^s$ for some $\eta < 1$.*

Corollary 4.8 shows that, when either the commitment level v is constant, or the correlation $f(s)$ is exponentially decaying, the parameter of concentration $F(v)/T$ for the regret of CHC tends to 0. The full proof is given in Appendix 4.B. To prove this result on the concentration of the competitive difference, we make heavy use of the fact that h is α-Hölder continuous in the second argument, which implies that the competitive difference is α-Hölder continuous in e. This allows application of the method of bounded difference, i.e., we bound the difference of $V(e)$ where one component of e is replaced by an identically-distributed copy. More specifically, we use the following lemma, the one-sided version of one due to McDiarmid:

Lemma 4.9 ([64], Lemma 1.2)**.** *Let $X = (X_1, \ldots, X_n)$ be independent random variables and Y be the random variable $f(X_1, \ldots, X_n)$, where function f satisfies $|f(x) - f(x'_k)| \leq c_k$ whenever x and x'_k differ in the kth coordinate. Then for any $t > 0$,* $\mathbb{P}(Y - \mathbb{E}Y > t) \leq \exp\left(\frac{-2t^2}{\sum_{k=1}^{n} c_k^2}\right)$.

4.5 Concluding remarks

Online convex optimization (OCO) problems with switching costs and noisy predictions are widely applicable in networking and distributed systems. Prior efforts in this area have resulted in two promising algorithms – Receding Horizon Control (RHC) and Averaging Fixed Horizon Control (AFHC). Unfortunately, it is not obvious when each algorithm should be used. Further, thus far, only AFHC has been analyzed in the presence of noisy predictions, despite the fact that RHC is seemingly more resistant to prediction noise in many settings.



(a) (b)

Figure 4.7: The cumulative distribution function of average-case bounds under different correlation structures: (a) i.i.d prediction noise; exponentially decaying, $a = 2/3$; (b) long range; short range, $L = 4$. Competitive differences simulated with random realization of standard normal $e(t)$ 1000 times under the following parameter values: $T = 100, v = 10, \beta D = 1, G = 0.1, \alpha = 1, c = 1$.

In this chapter, we provide the first analysis of RHC with noisy predictions. This novel analysis is made possible by the introduction of our new class of online algorithms, Committed Horizon Control (CHC), that allows for arbitrary levels of commitment, thus generalizing RHC and AFHC. Our analysis of CHC provides explicit results characterizing the optimal commitment level as a function of the variance and correlation structure of the prediction noise. In doing so, we characterize when RHC/AFHC is better depending on the properties of the prediction noise, thus addressing an important open challenge in OCO.

Our focus in this Chapter has been on the theoretical analysis of CHC and its implications for RHC and AFHC. The superiority of CHC suggests that it is a promising approach for integrating predictions into the design of systems, especially those that operate in uncertain environments. Going forward, it will be important to evaluate the performance of CHC algorithms in settings where RHC and AFHC have been employed, such as dynamic capacity provisioning, geographical load balancing, and video streaming.

4.A Experimental setup for Fig. 4.1

Setting for Figure 4.1(a): This example corresponds to a simple model of a data center. There are $(w+1)$ types of jobs and $(w+2)$ types of servers available to process these jobs. Each server has a different linear cost $\{a(t), b, c : 0 < a(t) < b < c\}$ (low, medium, high respectively) depending on the job type. The low cost is a monotonically increasing function of time that asymptotically approaches the constant medium cost (i.e. $a(t) = \alpha + (b - \alpha)\frac{t-1}{t}$, where $0 < \alpha < b$). The switching cost β only applies when a server is turned on (shut down costs can be included in the turning on cost) and has a magnitude greater than the difference between the medium and low costs (i.e. $\beta > b - \alpha$). The high cost is constant but greater than the difference between the medium and low costs multiplied by the prediction window size plus the switching cost. (i.e. $c > (b - \alpha)w + \beta$). One special server (server 0) can process all jobs with medium cost. Label all other servers 1 through $(w + 1)$ and all job types 1 through $(w + 1)$. Let server $s \in \{1, ..., w + 1\}$ be able to process job type s with low cost, job type $s - 1$ with high cost, and all other job types with medium cost.

We assume perfect prediction within the prediction window, w. The trace that forms Figure 4.1(a), is one in which the whole work load is only with one job type at each timestep starting with job type 1 and sequentially cycles through all job types every $(w + 1)$ timesteps.

This forces RHC to switch every timestep and FHC to switch every w timesteps to avoid a future high cost but take advantage of a low cost at the current timestep.

The offline optimal puts all of the workload on server 0 that processes all jobs with medium cost and so never incurs a switching cost after the first timestep.

RHC and AFHC try to take advantage of the low cost but the trace tricks them with a high cost one timestep beyond the prediction window. Switching to server 0 is always slightly too expensive by $(b - \alpha)\frac{1}{t}$ within the prediction window. The values used in Figure 4.1(a) are as follows: cycling workload of size 1 for 100 timesteps, $\alpha = 0.9$, $b = 1$, $\beta = 2$, $c = 0.1(w + 1) + 3$.

Setting for Figure 4.1(b): Similar to Figure 4.1(a), the setting in which this example was constructed corresponds to a simple model of a data center. The key difference is that predictions are noisy. There are $(w + 1)$ types of jobs and $(w + 1)$ types of servers available to process these jobs. Each server has a different linear cost $\{a, c : 0 < a < c\}$ (low, high respectively) depending on the job type. The switching cost β only applies when a server is turned on (shut down costs can be included in the turning on cost) and has a magnitude less than the difference between the high and low cost (i.e. $\beta < c - a$). Label all servers 1 through $(w + 1)$ and all job types 1 through $(w + 1)$. Let server $s \in \{1, ..., w + 1\}$ be able to process job type s with low cost, and all other job types with high cost.

We assume perfect prediction within only the first γ timesteps of the prediction window, w. The trace that forms Figure 4.1(b), is one in which the whole work load is only with one job type at each timestep starting with job type 1 and sequentially cycles through all job types every $(w+1)$ timesteps. Error in the last $w-\gamma$ timesteps of the prediction window is produced by making those predictions be equal to the prediction of the last perfect prediction (i.e. the γ-th timestep within the prediction window).

RHC equals the offline optimal solution in this setting which is to switch the whole workload at every timestep to the server with the unique low cost. AFHC on the other hand puts $(w-\gamma)/w$ of the workload on servers with high cost and only γ/w of the workload on the server with the unique low cost. The values used in Figure 4.1(b) are as follows: cycling workload of size 1 for 30 timesteps, $a = 1$, $c = 6$, $\beta = 0.1$.

4.B Proof of analytic results

We first introduce some additional notation used in the proofs. For brevity, for any vector x we write $x_{i..j} = (x_i, \ldots, x_j)$ for any $i \leq j$. Let x^* denote the offline optimal solution to (4.5), and let the cost of an online algorithm during time period $[t_1, t_2]$ with boundary conditions x_S, x_E and with online data $y_{t_1..t_2}$ be

$$g_{t1,t2}(x; x_S; x_E; y) = \sum_{t=t_1}^{t_2} h(x_t, y_t) + \beta \left\|x_S, x_{t_1}\right\| + \sum_{t=t_1+1}^{t_2} \beta \left\|x_{t-1}, x_t\right\| + \beta \left\|x_{t_2}, x_E\right\|.$$

If x_E is omitted, then by convention $x_E = x_{t_2}$ (and thus $\beta \left\|x_{t_2} - x_E\right\| = 0$). If x_S is omitted, then by convention $x_S = x_{t_1 - 1}$. Note that $g_{t_1,t_2}(x)$ depends only on x_i for $t_1 - 1 \leq i \leq t_2$.

Proof of Theorem 4.1

To characterize the suboptimality of CHC in the stochastic case, we first analyze the competitive difference of fixed horizon control with commitment level v, $FHC^k(v)$. Without loss of generality, assume that $k = 0$. Subsequently we omit k and v in FHC for simplicity. Construct a sequence of T-tuples $(\xi^1, \xi^2, \ldots, \xi^{M_1})$, where $M_1 = \#\{t \in [1,T] \mid t \mod v = 1\} \leq \lceil T/v \rceil$, such that $\xi^1 = x^*$ is the offline optimal solution, and $\xi^\tau_t = x_{FHC,t}$ for all $t < \tau v + 1$ hence, $\xi^{M_1} = x_{FHC}$. At stage τ, to calculate $\xi^{\tau+1}$, apply FHC to get $(\tilde{x}_{\tau v+1}, \ldots, \tilde{x}_{\tau v+w}) = X^\tau(\xi^\tau_{\tau v}, y_{\cdot|\tau v})$, and replace

$\xi^{\tau}_{\tau v+1:(\tau+1)v}$ with $\tilde{x}_{\tau v+1:(\tau+1)v}$ to get $\xi^{\tau+1}$, i.e.,

$$\xi^{\tau+1} = (\xi^{\tau}_1, \ldots, \xi^{\tau}_{\tau v}, \tilde{x}_{\tau v+1}, \ldots, \tilde{x}^{\tau}_{(\tau+1)v}, \xi^{\tau}_{(\tau+1)v+1}, \ldots, \xi^{\tau}_T).$$

By examining the terms in ξ^{τ} and $\xi^{\tau+1}$, we have

$$\begin{aligned}
&g_{1,T}(\xi^{\tau+1}; y) - g_{1,T}(\xi^{\tau}; y) \\
=& -\beta \left\| x^*_{(\tau+1)v+1} - x^*_{(\tau+1)v} \right\| + \beta \left\| x^*_{(\tau+1)v+1} - \tilde{x}_{(\tau+1)v} \right\| \\
&- \beta \left\| x^*_{\tau v+1} - \xi^{\tau}_{\tau v} \right\| + \beta \left\| \tilde{x}_{\tau v+1} - \xi^{\tau}_{\tau v} \right\| \\
&- \sum_{t=\tau v+1}^{(\tau+1)v} \left(h(x^*_t, y_t) + \beta \left\| x^*_t - x^*_{t-1} \right\| \right) \\
&+ \sum_{t=\tau v+1}^{(\tau+1)v} \left(h(\tilde{x}_t, y_t) + \beta \left\| \tilde{x}_t - \tilde{x}_{t-1} \right\| \right).
\end{aligned} \tag{4.15}$$

By construction of $(\tilde{x}_{\tau v+1}, \ldots, \tilde{x}_{(\tau+1)v})$, it is the optimal solution for $g_{\tau v+1,(\tau+1)v}(x; \xi^{\tau}_{\tau v}; \tilde{x}_{(\tau+1)v+1}; y_{\cdot|\tau})$, hence

$$\begin{aligned}
&\sum_{t=\tau v+1}^{(\tau+1)v} \left(h(\tilde{x}_t, y_{t|\tau v}) + \beta \left\| \tilde{x}_t - \tilde{x}_{t-1} \right\| \right) \\
&\quad + \beta \left\| \tilde{x}_{\tau v+1} - \xi^{\tau}_{\tau v} \right\| + \beta \left\| \tilde{x}_{(\tau+1)v+1} - \tilde{x}_{(\tau+1)v} \right\| \\
\leq& \sum_{t=\tau v+1}^{(\tau+1)v} \left(h(x^*_t, y_{t|\tau v}) + \beta \left\| x^*_t - x^*_{t-1} \right\| \right) \\
&\quad + \beta \left\| x^*_{\tau v+1} - \xi^{\tau}_{\tau v} \right\| + \beta \left\| \tilde{x}_{(\tau+1)v+1} - x^*_{(\tau+1)v} \right\|.
\end{aligned}$$

Substituting the above inequality into (4.15) and by triangle inequality, we have

$$\begin{aligned}
&g_{1,T}(\xi^{\tau+1}; y) - g_{1,T}(\xi^{\tau}; y) \\
\leq& 2\beta \left\| x^*_{(\tau+1)v+1} - \tilde{x}_{(\tau+1)v+1} \right\| + \sum_{t=\tau v+1}^{(\tau+1)v} \left| h(x^*_t, y_{t|\tau v}) - h(x^*_t, y_t) \right| \\
&+ \sum_{t=\tau v+1}^{(\tau+1)v} \left| h(\tilde{x}_t, y_t) - h(\tilde{x}_t, y_{t|\tau v}) \right| \\
\leq& 2\beta D + 2G \sum_{t=\tau v+1}^{(\tau+1)v} \left\| y_t - y_{t|\tau v} \right\|_2^{\alpha}.
\end{aligned} \tag{4.16}$$

Summing these inequalities from $\tau = 0$ to $\tau = M_1$ and noting that $\xi^{M_1} = x_{FHC^1(v)}$ and $\xi^1 = x^*$, we have

$$\begin{aligned}\text{cost}(FHC^1(v)) \leq &\text{cost}(OPT) + 2M_1\beta D + 2G\sum_{\tau=0}^{M_1}\sum_{t=\tau v+1}^{(\tau+1)v}\left\|y_t - y_{t|\tau v}\right\|_2^\alpha \\ =&\text{cost}(OPT) + 2M_1\beta D + 2G\sum_{t=1}^{T}\left\|y_t - y_{t|t-\phi^1(t)}\right\|_2^\alpha, \qquad (4.17)\end{aligned}$$

where $\phi^k(t) = \arg\min_{u\in\Psi_k, u\leq t}|t-u|$. For $k = 1$, $\phi^1(t) = u$ whenever $u = \tau v$ and $t \in [u, u+v-1]$ for some τ. We only have M_1 terms of the switching cost $\left\|x^*_{(\tau+1)v+1} - \tilde{x}_{(\tau+1)v+1}\right\|$ since $(M_1+1)v + 1 > T$. By the same argument, we have

$$\text{cost}(FHC^k(v)) \leq \text{cost}(OPT) + 2M_k\beta D + 2G\sum_{t=1}^{T}\left\|y_t - y_{t|\phi^k(t)}\right\|_2^\alpha.$$

Recall that $x_{CHC,t} = \frac{1}{v}\sum_{k=1}^{v} x_{FHC,t}^{(k,v)}$, by convexity of the cost function and Jensen's inequality, we have

$$\begin{aligned}&\text{cost}(CHC) \leq \frac{1}{v}\sum_{k=0}^{v-1}\text{cost}(FHC^k(v)) \\ \leq&\text{cost}(OPT) + \frac{2\sum_{k=0}^{v-1}M_k\beta D}{v} + \frac{2G}{v}\sum_{t=1}^{T}\sum_{k=0}^{v-1}\left\|y_t - y_{t|t-\phi^k(t)}\right\|_2^\alpha \\ \leq&\text{cost}(OPT) + \frac{2T\beta D}{v} + \frac{2G}{v}\sum_{t=1}^{T}\sum_{k=0}^{v-1}\left\|y_t - y_{t|t-\phi^k(t)}\right\|_2^\alpha \\ \leq&\text{cost}(OPT) + \frac{2T\beta D}{v} + \frac{2G}{v}\sum_{t=1}^{T}\sum_{k=0}^{v-1}\left\|y_t - y_{t|t-(k+1)}\right\|_2^\alpha, \qquad (4.18)\end{aligned}$$

where the third inequality is because $\sum_{k=0}^{v-1}M_k = T$ since by definition M_k is the number of elements in $[1, T]$ that is congruent to k modulus v; and the fourth inequality is because for all t, $t - \phi^k(t)$ always range from 1 to v when k goes from 0 to $v-1$.

Finally, we show that $\mathbb{E}\left\|y_\tau - y_{\tau|\tau-(k+1)}\right\|_2^\alpha \leq \|f_k\|^\alpha$ to finish the proof. Note that for

$\alpha = 2$, by (2.6), we have

$$\begin{aligned}
&\mathbb{E}\left\|y_\tau - y_{\tau|\tau-(k+1)}\right\|_2^2 = \mathbb{E}\left\|\sum_{s=\tau-k}^{\tau} f(\tau-s)e(s)\right\|_2^2 \\
=&\mathbb{E}\mathrm{tr}\left(\sum_{s_1,s_2=0}^{k} e(\tau-s_1)^T f(s_1)^T f(s_2)e(\tau-s_2)\right) \\
=&\mathrm{tr}\left(\sum_{s_1,s_2=0}^{k} f(s_1)^T f(s_2)\mathbb{E}e(\tau-s_2)e(\tau-s_1)^T\right) \\
=&\mathrm{tr}\left(R_e\sum_{s=0}^{k} f(s)^T f(s)\right) = \|f_k\|^2, \qquad (4.19)
\end{aligned}$$

where the second equality is due to cyclic invariance of trace and linearity of expectation, and the third equality is due to the fact that $e(s)$ are uncorrelated. When $\alpha \le 2$, $F(x) = x^{\alpha/2}$ is a concave function, hence by Jensen's inequality,

$$\begin{aligned}
&\mathbb{E}\left\|y_\tau - y_{\tau|\tau-(k+1)}\right\|_2^\alpha = \mathbb{E}F(\left\|y_\tau - y_{\tau|\tau-(k+1)}\right\|_2^2) \\
\le& F(\mathbb{E}\left\|y_\tau - y_{\tau|\tau-(k+1)}\right\|_2^2) = \|f_k\|^\alpha .
\end{aligned}$$

Proof of Corollary 4.3

Taking expectation over the prediction error and assuming long range correlation, we have for all $k \le v \le w$

$$\begin{aligned}
\|f_k\|^2 &= \sum_{s=0}^{k} \mathrm{tr}(R_e f(s)^T f(s)) = \sum_{s=0}^{k} \langle R_e^{1/2}, f(s)\rangle^2 \\
&\le \sum_{s=0}^{k} (\left\|R_e^{1/2}\right\|_F \|f(s)\|_F)^2 = (k+1)c^2\sigma^2,
\end{aligned}$$

where the inequality is due to Cauchy-Schwarz and $\|f_k\| = \sqrt{k+1}c\sigma$. To compute competitive difference of CHC, note that

$$\begin{aligned}
&\sum_{k=0}^{v-1} \|f_k\|^\alpha = \sum_{k=0}^{v-1} (\sqrt{k+1}c\sigma)^\alpha \le c^\alpha\sigma^\alpha \int_1^{v+1} k^{\alpha/2} dk \\
=&\frac{2c^\alpha\sigma^\alpha}{\alpha+2}((v+1)^{\frac{\alpha+2}{2}} - 1). \qquad (4.20)
\end{aligned}$$

Thus, by Theorem 4.1,

$$
\begin{aligned}
&\mathbb{E}\mathrm{cost}(CHC) - \mathbb{E}\mathrm{cost}(OPT) \leq \frac{2T\beta D}{v} + \frac{2GT}{v}\sum_{k=0}^{v-1}(\sqrt{k+1}c\sigma)^{\alpha} \\
\leq&\frac{2T\beta D}{v} + \frac{4GTc^{\alpha}\sigma^{\alpha}}{v(\alpha+2)}((v+1)^{1+\alpha/2} - 1) \\
=&\frac{2T\beta D(\alpha+2) - 4GTc^{\alpha}\sigma^{\alpha}}{\alpha+2}v^{-1} + \frac{4GTc^{\alpha}\sigma^{\alpha}}{\alpha+2}v^{\alpha/2}\left(\frac{v+1}{v}\right)^{1+\alpha/2} \\
\leq&\frac{2T\beta D(\alpha+2) - 4GTc^{\alpha}\sigma^{\alpha}}{\alpha+2}v^{-1} + \frac{2^{3+\frac{\alpha}{2}}GTc^{\alpha}\sigma^{\alpha}}{\alpha+2}v^{\alpha/2},
\end{aligned}
$$

where the last inequality is because $(v+1)/v \leq 2$ for $v \geq 1$. If $\beta D(\alpha+2) \leq 2Gc^{\alpha}\sigma^{\alpha}$, which implies $\frac{\beta D}{Gc^{\alpha}\sigma^{\alpha}} \leq \frac{2}{\alpha+2}$, then the right hand side is an increasing function of v, hence the commitment level that minimizes the performance guarantee is $v^* = 1$.

On the other hand, if $\beta D(\alpha+2) > 2Gc^{\alpha}\sigma^{\alpha}$, then let $A = \frac{\beta D(\alpha+2)-2Gc^{\alpha}\sigma^{\alpha}}{\alpha+2}$, $B = \frac{2^{2+\alpha/2}Gc^{\alpha}\sigma^{\alpha}}{\alpha+2}$, then the right hand side is $F(v) = 2T(Av^{-1} + Bv^{\alpha/2})$, by examining the gradient $F'(v) = 2T(-Av^{-2} + \frac{\alpha}{2}Bv^{-(1-\alpha/2)})$, since $F'(v) > 0$ iff $v^2F'(v) > 0$ and $v^2F'(v)$ is an increasing function in v, we can see that when $v < (\frac{2A}{\alpha B})^{2/(\alpha+2)}$, $F(v)' < 0$ and $F(v)$ is a decreasing function, when $v > (\frac{2A}{\alpha B})^{2/(\alpha+2)}$, $F(v)' > 0$ and $F(v)$ is an increasing function, hence when $v = (\frac{2A}{\alpha B})^{2/(\alpha+2)}$, $F(v)' = 0$ is the global minimum point of $F(v)$.

Therefore, when $\left(\frac{2A}{\alpha B}\right)^{2/(\alpha+2)} \geq w$, we have $v^* = w$, this happens when $\frac{\beta D}{Gc^{\alpha}\sigma^{\alpha}} > \alpha(2w)^{1+\frac{\alpha}{2}} + 2$. When $\frac{\beta D}{Gc^{\alpha}\sigma^{\alpha}} \in \left(\frac{2}{\alpha+2}, \alpha(2w)^{1+\frac{\alpha}{2}} + 2\right)$, v^* is between 1 to w, in this case, $v^* = \left(\frac{\beta D(\alpha+2)-2Gc^{\alpha}\sigma^{\alpha}}{2^{2+\alpha/2}\alpha Gc^{\alpha}\sigma^{\alpha}}\right)^{2/(\alpha+2)}$.

Proof of Corollary 4.4

Taking expectation over the prediction error, when $k < L$, similar to the proof of Corollary 4.3, $\|f_k\|^2 \leq (k+1)c^2\sigma^2$. When $k \geq L$, $\|f_k\|^2 \leq (L+1)c^2\sigma^2$. Thus if

$v > L$, we have

$$\begin{aligned}\sum_{k=0}^{v-1} \|f_k\|^\alpha &= \sum_{k=0}^{L-1} \|f_k\|^\alpha + \sum_{k=L}^{v-1} \|f_k\|^\alpha \\ \leq& \frac{2(c\sigma)^\alpha}{\alpha+2}((L+1)^{\frac{\alpha+2}{2}} - 1) + (v-L)(c\sigma)^\alpha (L+1)^{\alpha/2} \\ =& v(c\sigma)^\alpha (L+1)^{\alpha/2} \\ &+ \frac{(c\sigma)^\alpha}{\alpha+2}\Big((L+1)^{\alpha/2}(2(L+1) - (\alpha+2)L) - 1\Big) \\ =& v(c\sigma)^\alpha (L+1)^{\alpha/2} \\ &- \frac{(c\sigma)^\alpha}{\alpha+2}\Big((L+1)^{\alpha/2}(\alpha L - 2) + 1\Big),\end{aligned}$$

where the first inequality if by (4.20). Hence, by Theorem 4.1,

$$\begin{aligned}\mathbb{E}\mathrm{cost}(CHC) - \mathbb{E}\mathrm{cost}(OPT) \leq& \frac{2T\beta D}{v} + \frac{2GT}{v}\sum_{k=0}^{v-1} \|f_k\|^\alpha \\ \leq& \frac{2T\beta D}{v} + 2GT(c\sigma)^\alpha (L+1)^{\alpha/2} \\ &- \frac{2GT}{v}\frac{(c\sigma)^\alpha}{\alpha+2}((L+1)^{\alpha/2}(\alpha L - 2) + 1).\end{aligned}$$

The right hand side can be written as $2T(\frac{A-B}{v} + C)$, where $A = \beta D$, $B = G(c\sigma)^\alpha H(L)$, $C = G(c\sigma)^\alpha (L+1)^{\alpha/2}$ and $H(L) = \frac{1}{\alpha+2}\Big((L+1)^{\frac{\alpha}{2}}(\alpha L - 2) + 1\Big)$. When $A > B$, then the right hand side is a decreasing function in v, hence $v^* = w$; this happens when $\frac{\beta D}{Gc^\alpha \sigma^\alpha} > H(L)$. When $A < B$, then the right hand side is an increasing function in v, hence we want v^* to be small, i.e., $v \leq L$, this happens when $\frac{\beta D}{Gc^\alpha \sigma^\alpha} < H(L)$. When $v \leq L$, $\sum_{k=0}^{v-1} \|f_k\|^\alpha \leq \frac{2(c\sigma)^\alpha}{\alpha+2}((v+1)^{1+\frac{\alpha}{2}} - 1)$, hence

$$\begin{aligned}\mathbb{E}\mathrm{cost}(CHC) - \mathbb{E}\mathrm{cost}(OPT) \leq& \frac{2T\beta D}{v} + \frac{2GT}{v}\sum_{k=0}^{v-1} \|f_k\|^\alpha \\ \leq& \frac{2T\beta D(\alpha+2) - 4GTc^\alpha \sigma^\alpha}{v(\alpha+2)} + \frac{4GT(c\sigma)^\alpha}{v(\alpha+2)}((v+1)^{1+\alpha/2}).\end{aligned}$$

If $\frac{\beta D}{Gc^\alpha \sigma^\alpha} \leq \frac{2}{\alpha+2}$, then the right hand side is an increasing function in v, hence $v^* = 1$.

Proof of Corollary 4.5

Taking expectation over the prediction error, assuming that there exists $a < 1$, such that for all s, $\|f(s)\|_F \leq ca^s$, we have

$$\begin{aligned}\|f_k\|^2 &= \sum_{s=0}^{k} \operatorname{tr}(R_e f(s)^T f(s)) = \sum_{s=0}^{k} \langle R_e^{1/2}, f(s)\rangle^2 \\ &\leq \sum_{s=0}^{k} (\left\|R_e^{1/2}\right\|_F \|f(s)\|_F)^2 = \sum_{s=0}^{k} c^2\sigma^2 a^{2s} = c^2\sigma^2 \frac{1-a^{2k}}{1-a^2},\end{aligned}$$

where the inequality is due to Cauchy-Schwarz, hence for h that is G-Lipschitz in the second argument, we have $\|f_k\| \leq \frac{c\sigma(1-a^{2(k+1)}/2)}{1-a^2}$, where the inequality is because $\sqrt{1-a^2} \geq (1-a^2)$, and $1-a^{2(k+1)} \leq 1-a^{2(k+1)} + a^{4(k+1)}/4 = (1-a^{2(k+1)}/2)^2$, hence,

$$\begin{aligned}&\mathbb{E}\mathrm{cost}(CHC) - \mathbb{E}\mathrm{cost}(OPT) \\ \leq&\frac{2T\beta D}{v} + \frac{2GT}{v}\sum_{k=0}^{v-1}\frac{c\sigma(1-a^{2(k+1)}/2)}{1-a^2} \\ \leq&\frac{2T\beta D}{v} + \frac{2GTc\sigma}{1-a^2} - \frac{GTc\sigma}{v(1-a^2)}\sum_{k=1}^{v} a^{2k} \\ \leq&\frac{2T\beta D}{v} + \frac{2GTc\sigma}{1-a^2} - \frac{GTc\sigma}{v(1-a^2)}\frac{a^2(1-a^{2v})}{1-a^2}.\end{aligned}$$

Let $A = 2T\beta D$, $B = \frac{a^2 GTc\sigma}{1-a^2}$ and $C = \frac{2GTc\sigma}{1-a^2}$, then

$$\begin{aligned}\mathbb{E}\mathrm{cost}(CHC) - \mathbb{E}\mathrm{cost}(OPT) &\leq \frac{A}{v} - \frac{B(1-a^{2v})}{v} + C \\ =&\frac{(A-B)+Ba^{2v}}{v} + C.\end{aligned}$$

Therefore when $A \geq B$, $(A-B)+Ba^{2v} > 0$, and the RHS is a decreasing function in v, hence $v^* = w$, and this happens when $2T\beta D \geq \frac{a^2 GTc\sigma}{1-a^2}$ which implies $\frac{\beta D}{Gc\sigma} \geq \frac{a^2}{2(1-a^2)}$.

On the other hand, if $A < B - Ba$, then $(A-B) + Ba^{2v} \leq A - B + Ba < 0$, and the right hand side is an increasing function in v, hence $v^* = 1$, and this happens when $2T\beta D < \frac{a^2 GTc\sigma}{1-a^2}(1-a)$, which implies $\frac{2\beta D}{Gc\sigma} \leq \frac{a^2}{2(1+a)}$.

Proof of Theorem 4.6

The proof follows in the same fashion as that of Theorem 4.1. Recall that we have

$$\begin{aligned}
&g_{1,T}(\xi^{\tau+1}; y) - g_{1,T}(\xi^{\tau}; y) \\
\leq& 2\beta \left\| x^*_{(\tau+1)v+1} - \tilde{x}_{(\tau+1)v+1} \right\| + \sum_{t=\tau v+1}^{(\tau+1)v} |h(x^*_t, y_{t|\tau v}) - h(x^*_t, y_t)| \\
&+ \sum_{t=\tau v+1}^{(\tau+1)v} |h(\tilde{x}_t, y_t) - h(\tilde{x}_t, y_{t|\tau v})| \\
\leq& 2\beta \left\| x^*_{(\tau+1)v+1} - \tilde{x}_{(\tau+1)v+1} \right\| + 2G \sum_{t=\tau v+1}^{(\tau+1)v} \left\| y_t - y_{t|\tau v} \right\|_2^{\alpha}.
\end{aligned} \tag{4.21}$$

Since h is m-strongly convex, $g_{\tau v+1,(\tau+1)v}$ is also m-strongly convex, hence

$$\begin{aligned}
&g_{\tau v+1,(\tau+1)v}(\tilde{x}_{\tau v+1:(\tau+1)v}; \xi^{\tau}_{\tau v}; \tilde{x}_{(\tau+1)v+1}; y_{\cdot|\tau v}) \\
\leq& g_{\tau v+1,(\tau+1)v}(x^*_{\tau v+1:(\tau+1)v+1}; \xi^{\tau}_{\tau v}; \tilde{x}_{(\tau+1)v+1}; y_{\cdot|\tau v}) \\
&- \partial g_{\tau v+1,(\tau+1)v}(\tilde{x}_{\tau v+1:(\tau+1)v}) \cdot (x^*_{\tau v:(\tau+1)v} - \tilde{x}_{\tau v:(\tau+1)v}) \\
&- \frac{m}{2} \sum_{t=\tau v+1}^{(\tau+1)v} \left\| x^*_t - \tilde{x}_t \right\|^2.
\end{aligned}$$

By the optimality of $\tilde{x}_{\tau v+1:(\tau+1)v}$ minimizes the cost function $g_{\tau v+1,(\tau+1)v}(x; \xi^{\tau}_{\tau v}; \tilde{x}_{(\tau+1)v+1}; y_{\cdot|\tau v})$, we have the first order condition

$$\partial g_{\tau v+1,(\tau+1)v}(\tilde{x}_{\tau v+1:(\tau+1)v}) \cdot (x^*_{\tau v+1:(\tau+1)v} - \tilde{x}_{\tau v+1:(\tau+1)v}) \geq 0,$$

hence

$$\begin{aligned}
&\sum_{t=\tau v+1}^{(\tau+1)v} \left(h(\tilde{x}_t, y_{t|\tau v}) + \beta \left\| \tilde{x}_t - \tilde{x}_{t-1} \right\| \right) \\
&+ \beta \left\| \tilde{x}_{\tau v+1} - \xi^{\tau}_{\tau v} \right\| + \beta \left\| \tilde{x}_{(\tau+1)v+1} - \tilde{x}_{(\tau+1)v} \right\| \\
\leq& \sum_{t=\tau v+1}^{(\tau+1)v} \left(h(x^*_t, y_{t|\tau v}) + \beta \left\| x^*_t - x^*_{t-1} \right\| - \frac{m}{2} \left\| \tilde{x}_t - x^*_t \right\|^2 \right) \\
&+ \beta \left\| x^*_{\tau v+1} - \xi^{\tau}_{\tau v} \right\| + \beta \left\| \tilde{x}_{(\tau+1)v+1} - x^*_{(\tau+1)v} \right\|.
\end{aligned}$$

Substituting the above inequality into (4.21) and summing over τ, we have

$$\begin{aligned}
&\operatorname{cost}(FHC^1(v)) - \operatorname{cost}(OPT) = \sum_{\tau=0}^{M} g_{1,T}(\xi^{\tau+1}; y) - g_{1,T}(\xi^{\tau}; y) \\
\leq & 2\beta \sum_{\tau=0}^{M-1} \left\| x^*_{(\tau+1)v+1} - \tilde{x}_{(\tau+1)v+1} \right\| - \sum_{\tau=0}^{M} \sum_{t=\tau v+1}^{(\tau+1)v} \frac{m}{2} \left\| \tilde{x}_t - x^*_t \right\|^2 \\
& + 2G \sum_{\tau=0}^{M} \sum_{t=\tau v+1}^{(\tau+1)v} \left\| y_t - y_{t|\tau v} \right\|_2^{\alpha} \\
\leq & \sum_{\tau=1}^{M} \left(2\beta \left\| x^*_{\tau v+1} - \tilde{x}_{\tau v+1} \right\| - \frac{m}{2} \left\| x^*_{\tau v+1} - \tilde{x}_{\tau v+1} \right\|^2 \right) \\
& + 2G \sum_{\tau=0}^{M} \sum_{t=\tau v+1}^{(\tau+1)v} \left\| y_t - y_{t|\tau v} \right\|_2^{\alpha} \\
\overset{(a)}{\leq} & \frac{2\beta^2 M}{m} + 2G \sum_{t=1}^{T} \left\| y_t - y_{t|t-\phi^1(t)} \right\|^{\alpha},
\end{aligned}$$

where (a) is because for any t,

$$\begin{aligned}
& 2\beta \left\| x^*_t - \tilde{x}_t \right\| - \frac{m}{2} \left\| x^*_t - \tilde{x}_t \right\|^2 \\
= & -\frac{m}{2}(\left\| x^*_t - \tilde{x}_t \right\| - \frac{2\beta}{m})^2 + \frac{2\beta^2}{m} \leq \frac{2\beta^2}{m}.
\end{aligned}$$

Summing over k from 0 to $v - 1$ as in (4.18) and taking expectation on both sides as in (4.19) finishes the proof.

Proof of Theorem 4.7

By the proof of Theorem 4.1 and assuming one dimensional setting, we have

$$\begin{aligned}
\operatorname{cost}(CHC) - \operatorname{cost}(OPT) &\leq \frac{2\beta TD}{v} + \frac{2G}{v} \sum_{\tau=1}^{T} \sum_{k=0}^{v-1} \left| y_\tau - y_{\tau|\tau-k} \right|^{\alpha} \\
= \frac{2\beta TD}{v} + \frac{2G}{v} \sum_{\tau=1}^{T} \sum_{k=1}^{v} & \left| \sum_{s=\tau-k+1}^{\tau} f(\tau - s) e(s) \right|^{\alpha}, \qquad (4.22)
\end{aligned}$$

which is a function of the randomness of prediction noise $e = (e(1), \ldots, e(T))$, let

$$l(e) := \frac{2\beta TD}{v} + \frac{2G}{v} \sum_{\tau=1}^{T} \sum_{k=1}^{v} \left| \sum_{s=\tau-k+1}^{\tau} f(\tau - s) e(s) \right|^{\alpha}$$

be the upper bound of the competitive difference of CHC in terms of the random variables $e(1), \ldots, e(T)$. For every i, let $e(i)'$ be an independent and identical copy of $e(i)$, and let $e_i' = (e(1), \ldots, e(i-1), e(i)', e(i+1), \ldots, e(T))$ be the vector that differ from e by the replacing the ith coordinate with an identical copy of $e(i)$, then let δ_{is} be the kronecker delta, we have

$$\begin{aligned}
|l(e) - l(e_i')| \leq & \frac{2G}{v} \sum_{\tau=1}^{T} \sum_{k=1}^{v} \left| \sum_{s=\tau-k+1}^{\tau} \delta_{is} f(\tau - s) e(s) \right|^{\alpha} \\
& - \frac{2G}{v} \sum_{\tau=1}^{T} \sum_{k=1}^{v} \left| \sum_{s=\tau-k+1}^{\tau} \delta_{is} f(\tau - s) e_i'(s) \right|^{\alpha} \\
& \leq \frac{2G}{v} \sum_{\tau=1}^{T} \sum_{k=1}^{v} \left| \sum_{s=\tau-k+1}^{\tau} \delta_{is} f(\tau - s)(e(s) - e_i'(s)) \right|^{\alpha},
\end{aligned}$$

where the last inequality is due to the fact that $F(x) = x^{\alpha}$ is a α-Hölder continuous with constant 1, hence $\forall x, y, |x^{\alpha} - y^{\alpha}| \leq |x - y|^{\alpha}$. Therefore,

$$\begin{aligned}
|l(e) - l(e_i')| \leq & \frac{2G}{v} \sum_{\tau=1}^{T} \sum_{k=1}^{v} \left| \sum_{s=\tau-k+1}^{\tau} \delta_{is} f(\tau - s)(e(s) - e_i'(s)) \right|^{\alpha} \\
& \overset{(a)}{\leq} \frac{2G}{v} \sum_{k=1}^{v} \sum_{\tau=1}^{T} \sum_{s=\tau-k+1}^{\tau} \delta_{is} |f(\tau - s)|^{\alpha} |e(s) - e_i'(s)|^{\alpha} \\
& \overset{(b)}{=} \frac{2G}{v} |e(i) - e_i'(i)|^{\alpha} \sum_{k=1}^{v} \sum_{s=0}^{k-1} |f(s)|^{\alpha} \\
& \leq \frac{2^{1+\alpha} G \varepsilon^{\alpha}}{v} \sum_{s=0}^{v-1} (v - s) |f(s)|^{\alpha}, \qquad (4.23)
\end{aligned}$$

where (a) is because for all $\alpha \leq 1$, $(\sum_{i=1}^{k} |a_i|)^{\alpha} \leq \sum_{i=1}^{k} |a_i|^{\alpha}$, to prove this, note that we only need to show that

$$\forall x, y \geq 0, (x + y)^{\alpha} \leq x^{\alpha} + y^{\alpha} \qquad (4.24)$$

and iterate this inequality k times. To prove the above, note that when $x = 0$ or $y = 0$ or $\alpha = 1$, the inequality is trivially true. Otherwise, wlog assume $x \geq y$, and let $t = y/x \leq 1$. (4.24) is equivalent to $(1 + t)^{\alpha} \leq 1 + t^{\alpha}$, this is true by examining the function $F(t) = 1 + t^{\alpha} - (1 + t)^{\alpha}$, note that $F(0) = 0$, and for $\alpha < 1$, $F(t)' = \alpha(t^{\alpha-1} - (1 + t)^{\alpha-1}) > 0$, since a^x hence $F(t) \geq 0$ for all t, which proves (a).

(b) is because let $s' = \tau - s$, then

$$\begin{aligned}
&\sum_{k=1}^{v}\sum_{\tau=1}^{T}\sum_{s=\tau-k+1}^{\tau}\delta_{is}|f(\tau-s)|^\alpha|e(s)-e_i'(s)|^\alpha\\
=&\sum_{k=1}^{v}\sum_{\tau=1}^{T}\sum_{s'=0}^{k-1}\delta_{i(\tau-s')}|f(s')|^\alpha|e(\tau-s')-e_i'(\tau-s')|^\alpha\\
=&\sum_{k=1}^{v}\sum_{s'=0}^{k-1}|f(s')|^\alpha\left(\sum_{\tau=1}^{T}\delta_{i(\tau-s')}|e(\tau-s')-e_i'(\tau-s')|^\alpha\right)\\
=&|e(i)-e_i'(i)|^\alpha\sum_{k=1}^{v}\sum_{s'=0}^{k-1}|f(s')|^\alpha.
\end{aligned}$$

Let $c_i = 2^{1+\alpha}G\varepsilon^\alpha\sum_{k=0}^{v-1}\frac{(v-k)|f(k)|^\alpha}{v}$, then by (4.23), $|l(e)-l(e_i')|^2 \le c_i^2$, by Lemma 4.9, we have

$$\begin{aligned}
&\mathbb{P}(\text{cost}(CHC)-\text{cost}(OPT)\ge V_1T+u)\\
\le&\exp\left(\frac{-2u^2}{\sum_{i=1}^{T}c_i^2}\right)=\exp\left(\frac{-u^2}{2^{1+2\alpha}G^2\varepsilon^{2\alpha}TF(v)}\right)
\end{aligned}$$

where $F(v) = \left(\frac{1}{v}\sum_{k=0}^{v-1}(v-k)|f(k)|^\alpha\right)^2$.

Proof of Corollary 4.8

By Theorem 4.7,

$$\begin{aligned}
&\mathbb{P}\left(\frac{1}{T}[\text{cost}(CHC)-\text{cost}(OPT)]>V_1+u\right)\\
=&\mathbb{P}\left(\text{cost}(CHC)-\text{cost}(OPT)>V_1T+uT\right)\\
\le&\exp\left(\frac{-2u^2}{\sum_{i=1}^{T}c_i^2}\right)=\exp\left(\frac{-u^2}{2^{1+2\alpha}G^2\varepsilon^{2\alpha}F(v)/T}\right).
\end{aligned}$$

If $v \in O(1)$, then $F(v)$ is bounded since $f(s)$ is bounded, hence $F(v)/T \to 0$ as $T \to \infty$.

Otherwise, if $f(s) \le c\eta^s$ for $\eta < 1$, denote $a = \eta^\alpha$, then

$$\begin{aligned}
&\sum_{k=0}^{v-1}(v-k)|f(k)|^\alpha \le c^\alpha\left(v\sum_{k=0}^{v-1}a^k - a\sum_{k=0}^{v-1}\frac{d}{da}(a^k)\right)\\
=&c^\alpha\left(v\frac{1-a^v}{1-a}-a\frac{d}{da}\left(\frac{1-a^v}{1-a}\right)\right)=c^\alpha\frac{v-a(v+1)+a^{v+1}}{(1-a)^2}
\end{aligned}$$

then

$$F(v) = \left(\frac{1}{v}\sum_{k=0}^{v-1}(v-k)|f(k)|^{\alpha}\right)^2 \leq \left(\frac{c^{\alpha}(v - \eta^{\alpha}(v+1) + \eta^{\alpha(v+1)})}{(1-a)^2 v}\right)^2$$
$$\leq \left(\frac{c^{\alpha}}{(1-\eta^{\alpha})^2} + \frac{c^{\alpha}\eta^{\alpha}}{(1-\eta^{\alpha})^2 v}\right)^2 \in O(1).$$

Thus in this case $F(v)/T \to 0$ as $T \to \infty$.

Chapter 5

AN APPLICATION

5.1 Smart energy system

The electricity grid is at the brink of change. On the generation side, the penetration of wind and solar in the energy portfolio is on the rise due to environmental concerns. And, on the demand side, many smart appliances and devices with adjustable power consumption levels are entering the market. The combination of these two changes makes generation less controllable and load less predictable, which makes the traditional "generation follows load" model of control much more difficult.

Fortunately, while smart devices make demand forecasting more challenging, they also provide an opportunity to mitigate the intermittency of wind and solar generation from the load side by allowing for demand response. There are two major categories of demand response: direct load control (DLC) and price-based demand response. See [3] for a discussion of the contrasts between these approaches.

In this chapter we focus on *direct load control* with the goal of using demand response to *reduce variations of the aggregate load*. This objective has been studied frequently in the literature, e.g., [30, 77], because reducing the variations of the aggregate load corresponds to minimizing the generation cost of the utilities. In particular, large generators with the smallest marginal costs, e.g., nuclear generators and hydro generators, have limited ramp rates, i.e., their power output cannot be adjusted too quickly. So, if load varies frequently, then it must be balanced by more expensive generators (i.e., "peakers") that have fast ramp rate. Thus, if the load variation is reduced, then the utility can use the least expensive sources of power generation to satisfy the electricity demand.

Model predictive deferrable load control

There is a growing body of work on direct load control algorithms, which includes both simulation-based evaluations [1, 42, 65] and theoretical performance guarantees [27, 62]. The most commonly proposed framework for algorithm design from this literature is, perhaps, model predictive control.

Model predictive control (MPC) is a classical control algorithm, e.g., see [68] for a survey. MPC can be applied to settings where unknown disturbances to the system are present through the robust control paradigm or the certainty equivalence principle, e.g., see [11, 18, 52]. In the context of direct load control, many variations have been proposed. Scalability and performance in the presence of uncertainty are essential to MPC algorithms for direct load control. At this point, there exist model predictive deferrable load control algorithms that can be fully distributed with guaranteed convergence to optimal deferrable load schedules, e.g., [30].

However, to this point, the evaluation of model predictive deferrable load control has focused primarily on average-case analysis, e.g., [24, 71], or worst-case analysis, e.g., [23, 55]. While such analysis provides important insights, there is still much to learn about the performance of model predictive deferrable load control.

For example, it is likely that an algorithm has good average performance but bad worst case performance, and vice versa. What is really needed is a distributional analysis that tells us about the "typical" performance, which can say, e.g., that the load variation will be less than the desired level 95 percent of the time. But, to this point, no results on the distribution of the load variation under model predictive deferrable load control exist.

5.2 Real-time deferrable load control

This chapter studies the design and analysis of real-time control algorithms for scheduling deferrable loads to compensate for the random fluctuations in renewable generation. In the following we present a model of this scenario that serves as the basis for our algorithm design and performance evaluation. The model includes renewable generation, non-deferrable loads, and deferrable loads, which are described in turn. The key differentiation of this model from that of [27] is the inclusion of uncertainties (prediction errors) on future renewable generation and loads.

Throughout, we consider a discrete-time model over a finite time horizon. The time horizon is divided into T time slots of equal length and labeled $1, \ldots, T$. In practice, the time horizon could be one day and the length of a time slot could be 10 minutes.

Renewable generation like wind is stochastic and difficult to predict. Similarly, non-deferrable load including lights are hard to predict at a low aggregation level.

Since the focus is on scheduling deferrable loads, we aggregate renewable generation and non-deferrable load into one process termed the *base load*, $b = \{b(\tau)\}_{\tau=1}^{T}$, which is defined as the difference between non-deferrable load and renewable generation, and is a stochastic process.

To model the uncertainty of base load, we use a causal filter based model described as follows, and illustrated in Figure 5.1. In particular, the base load at time τ is modeled as a random deviation $\delta b = \{\delta b(\tau)\}_{\tau=1}^{T}$ around its expectation $\bar{b} = \{\bar{b}(\tau)\}_{\tau=1}^{T}$. The process $\bar{b}$ is specified externally to the model, e.g., from historical data and weather report, and the process $\delta b(\tau)$ is further modeled as an uncorrelated sequence of identically distributed random variables $e = \{e(\tau)\}_{\tau=1}^{T}$ with mean 0 and variance σ^2, passing through a causal filter. Specifically, let $f = \{f(\tau)\}_{\tau=-\infty}^{\infty}$ denote the impulse response of this causal filter and assume that $f(0) = 1$, then $f(\tau) = 0$ for $\tau < 0$ and

$$\delta b(\tau) = \sum_{s=1}^{T} e(s) f(\tau - s), \quad \tau = 1, \ldots, T.$$

At time $t = 1, \ldots, T$, a prediction algorithm can observe the sequence $e(s)$ for $s = 1, \ldots, t$, and predicts b as[1]

$$b_t(\tau) = \bar{b}(\tau) + \sum_{s=1}^{t} e(s) f(\tau - s), \quad \tau = 1, \ldots, T. \tag{5.1}$$

Note that $b_t(\tau) = b(\tau)$ for $\tau = 1, \ldots, t$ since f is causal.

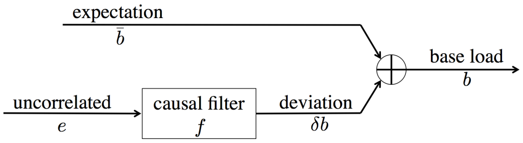


Figure 5.1: Diagram of the notation and structure of the model for base load, i.e., non-deferrable load minus renewable generation.

This model allows for non-stationary base load through the specification of $\bar{b}$ and a broad class of models for uncertainty via f and e. In particular, two specific filters f that we consider in detail later in the chapter are:

1. A filter with finite but flat impulse response, i.e., there exists $\Delta > 0$ such that

$$f(t) = \begin{cases} 1 & \text{if } 0 \le t < \Delta \\ 0 & \text{otherwise;} \end{cases}$$

2. A filter with an infinite and exponentially decaying impulse response, i.e., there exists $a \in (0, 1)$ such that

$$f(t) = \begin{cases} a^t & \text{if } t \ge 0 \\ 0 & \text{otherwise.} \end{cases}$$

[1] This prediction algorithm is a Wiener filter [17].

These two filters provide simple but informative examples for our discussion in Section 5.4.

To model deferrable loads we consider a setting where N deferrable loads arrive over the time horizon, each requiring a certain amount of electricity by a given deadline. Further, a real-time algorithm has imperfect information about the arrival times and sizes of these deferrable loads.

More specifically, we assume a total of N deferrable loads and label them in increasing order of their arrival times by $1, \ldots, N$, i.e., load n arrives no later than load $n+1$ for $n = 1, \ldots, N-1$. Further, we define $N(t)$ as the number of loads that arrive before (or at) time t for $t = 1, \ldots, T$ and fix $N(0) \overset{\text{def}}{=} 0$. Thus, load $1, \ldots, N(t)$ arrive before or at time t for $t = 1, \ldots, T$ and $N(T) = N$.

For each deferrable load, its arrival time and deadline, as well as other constraints on its power consumption, are captured via upper and lower bounds on its possible power consumption during each time. Specifically, the power consumption of deferrable load n at time t, $p_n(t)$, must be between given lower and upper bounds $\underline{p}_n(t)$ and $\overline{p}_n(t)$, i.e.,

$$\underline{p}_n(t) \le p_n(t) \le \overline{p}_n(t), \quad n = 1, \ldots, N, \ t = 1, \ldots, T. \tag{5.2}$$

These are specified externally to the model. For example, if an electric vehicle plugs in with Level II charging, then its power consumption must be within $[0, 3.3]$kW. However, if it is not plugged in (has either not arrived yet or has already departed) then its power consumption is 0kW, i.e., within $[0, 0]$kW. Further, we assume that a deferrable load n must withdraw a fixed amount of energy P_n by its deadline, i.e.,

$$\sum_{t=1}^{T} p_n(t) = P_n, \quad n = 1, \ldots, N. \tag{5.3}$$

Finally, the N deferrable loads arrive randomly throughout the time horizon. Define

$$a(t) \overset{\text{def}}{=} \sum_{n=N(t-1)+1}^{N(t)} P_n \tag{5.4}$$

as the total energy request of all deferrable loads that arrive at time t for $t = 1, \ldots, T$. We assume that $\{a(t)\}_{t=1}^{T}$ is a sequence of independent identically distributed random variables with mean λ and variance s^2. Further, define

$$A(t) \overset{\text{def}}{=} \sum_{\tau=t+1}^{T} a(\tau) \tag{5.5}$$

as the total energy requested after time t for $t = 1, \ldots, T$.

In summary, at time $t = 1, \ldots, T$, a real-time algorithm has full information about the deferrable loads that have arrived, i.e., $\underline{p}_n$, $\overline{p}_n$, and P_n for $n = 1, \ldots, N(t)$, and knows the expectation of future deferrable load total energy request $\mathbb{E}(A(t))$. However, a real-time algorithm has no other knowledge about deferrable loads that arrive after time t.

We can now formally state the deferrable load control problem that is the focus of this thesis. Recall that the objective of real-time deferrable load control is to compensate for the random fluctuations in renewable generation and non-deferrable load in order to "flatten" the *aggregate load* $d = \{d(t)\}_{t=1}^{T}$, which is defined as

$$d(t) = b(t) + \sum_{n=1}^{N} p_n(t), \quad t = 1, \ldots, T. \tag{5.6}$$

In this thesis, we focus on minimizing the *sample path variance* of the aggregate load d, $V(d)$, as a measure of "flatness" that is defined as

$$V(d) = \frac{1}{T} \sum_{t=1}^{T} \left(d(t) - \frac{1}{T} \sum_{\tau=1}^{T} d(\tau) \right)^2 . \tag{5.7}$$

We can now formally specify the optimal deferrable load control (ODLC) problem that we seek to solve:

$$\begin{aligned} \textbf{O}\text{DLC: } \min \quad & \frac{1}{T} \sum_{t=1}^{T} \left(d(t) - \frac{1}{T} \sum_{\tau=1}^{T} d(\tau) \right)^2 \\ \text{over} \quad & p_n(t), d(t), \quad \forall n, t \\ \text{s.t.} \quad & d(t) = b(t) + \sum_{n=1}^{N} p_n(t), \quad \forall t; \\ & \underline{p}_n(t) \le p_n(t) \le \overline{p}_n(t), \quad \forall n, t; \\ & \sum_{t=1}^{T} p_n(t) = P_n, \quad \forall n. \end{aligned} \tag{5.8}$$

In the above ODLC, the objective is simply the sample path variance of the aggregate load, $V(d)$, and the constraints correspond to equations (5.6), (5.2), and (5.3), respectively. We chose $V(d)$ as the objective for ODLC because of its significance for microgrid operators [39]. However, additionally, [27] has proven that the optimal solution does not change if the objective function $V(d)$ is replaced by $f(d) = \sum_{t=1}^{T} U(d(t))$ where $U : \mathbb{R} \to \mathbb{R}$ is strictly convex. Therefore, we can use $V(d)$ without loss of generality.

5.3 Model predictive algorithm

Given the optimal deferrable load control (ODLC) problem defined in (5.8), the first contribution of this thesis is to design an algorithm that solves ODLC in real-time, given uncertain predictions of base and deferrable loads.

There are two key challenges for the algorithm design. First, the algorithm has access only to uncertain predictions at any given time, i.e., at time t the algorithm only knows deferrable loads 1 to $N(t)$ rather than 1 to N, and only knows the prediction b_t instead of b itself. Second, even if there was no uncertainty in predictions, solving the ODLC problem requires significant computational effort when there are a large number of deferrable loads.

Motivated by these challenges, in this section we design a decentralized algorithm with strong performance guarantees even when there is uncertainty in the predictions. The algorithm builds on the work of [27], which provides a decentralized algorithm for the case without uncertainty in predictions. We present the details of the algorithm from [27] in Section 5.3 and then present a modification of the algorithm to handle uncertain predictions in Section 5.3.

Load control without uncertainty

We start with the case where the algorithm has complete knowledge (no uncertainty) about base load and deferrable loads. In this context, the key algorithmic challenge is to solve the ODLC problem in (5.8) via a decentralized algorithm. Such a decentralized algorithm was proposed in [27], and we summarize the algorithm and its analysis here.

Algorithm definition: The algorithm from [27] is given in Algorithm 5.1. It is iterative and the superscripts in brackets denote the round of iteration. In each iteration $k \geq 0$, there are two key steps: Step (ii) and (iii). In Step (ii), the utility calculates the average load $g^{(k)}$ and broadcasts it to all deferrable loads. Note that the utility only needs to know the reported schedule $p_n^{(k)}$, the base load b, and the number of deferrable loads N. It does not need to know the constraints of the deferable loads. In Step (iii), each deferrable load n updates $p_n^{(k+1)}$ by solving a convex optimization. The objective function has two terms. The first term can be interpreted as the electricity bill if the electricity price was set to $g^{(k)}$. The second term vanishes as iterations continue.

Algorithm convergence results: Importantly, though Algorithm 5.1 is iterative, it converges very fast. In fact, the simulations in [27] stop the iterations after 15

```
Algorithm 5.1 Deferrable load control without uncertainty
Require: The utility knows the base load $b$ and the number $N$ of deferrable loads.
    Each load $n \in \{1, \dots, N\}$ knows its energy request $P_n$ and power consumption
    bounds $\overline{p}_n$ and $\underline{p}_n$. The utility sets $K$, the number of iterations.
Ensure: Deferrable load schedule $p = (p_1, \dots, p_N)$.
  (i) Set $k \leftarrow 0$ and intitialize the schedule $p^{(k)}$ as
          $p_n^{(k)}(t) \leftarrow 0, \quad t = 1, \dots, T, \; n = 1, \dots, N.$
  (ii) The utility calculates the average load $g^{(k)} = d^{(k)}/N$,
          $g^{(k)}(t) \leftarrow \frac{1}{N}\left(b(t) + \sum_{n=1}^{N} p_n^{(k)}(t)\right), \quad t = 1, \dots, T,$
      and broadcasts $g^{(k)}$ to all deferrable loads.
  (iii) Each load $n$ updates a new schedule $p_n^{(k+1)}$ by solving
          min   $\sum_{\tau=1}^{T} g^{(k)}(\tau) p_n(\tau) + \frac{1}{2}\left(p_n(\tau) - p_n^{(k)}(\tau)\right)^2$
          over  $p_n(1), \dots, p_n(T)$
          s.t.  $\underline{p}_n(\tau) \le p_n(\tau) \le \overline{p}_n(\tau), \quad \forall \tau;$
                $\sum_{\tau=1}^{T} p_n(\tau) = P_n,$
      and reports $p_n^{(k+1)}$ to the utility.
  (iv) Set $k \leftarrow k + 1$. If $k < K$, go to Step (ii).
```

rounds (i.e., K=15) in all cases because convergence is already achieved. Further, Algorithm 5.1 provably solves the ODLC problem given in (5.8) when there is no uncertainty, i.e., when $N(t) = N$ and $b_t = b$ for $t = 1, \dots, T$ [27]. More precisely, let $\mathcal{O}$ denote the set of optimal solutions to (5.8), and define $d(p, \mathcal{O}) := \min_{\hat{p} \in \mathcal{O}} \|p - \hat{p}\|$ as the distance from a deferrable load schedule p to optimal deferrable load schedules $\mathcal{O}$.

Proposition 5.1 ([27]). *When there is no uncertainty, i.e., $N(t) = N$ and $b_t = b$ for $t = 1, \dots, T$, the deferrable load schedules $p^{(k)}$ obtained by Algorithm 5.1 converge to optimal schedules to ODLC, i.e., $d(p^{(k)}, \mathcal{O}) \to 0$ as $k \to \infty$.*

A particular class of optimal solutions will be of interest to us later in the chapter, so we define them here. Specifically, we call a feasible deferrable load schedule $p = (p_1, \ldots, p_N)$ *v*alley-filling, if there exists some constant $C \in \mathbb{R}$ such that $\sum_{n=1}^{N} p_n(t) = (C - b(t))^+$ for $t = 1, \ldots, T$.

Proposition 5.2 ([27])**.** *If a valley-filling deferrable load schedule exists, then it solves ODLC. Further, in such cases, all optimal schedules to ODLC have the same aggregate load.*

Note that valley-filling schedules tend to exist if there is a large number of deferrable loads, in such settings optimal solutions to ODLC are valley-filling.

Load control with uncertainty

Algorithm 5.1 provides a decentralized approach for solving the ODLC problem; however it assumes exact knowledge (certainty) about base load and deferrable loads. In this subsection, we adapt Algorithm 5.1 to the setting where there is uncertainty in base load and deferrable load predictions, while maintaining strong performance guarantees. In particular, in this subsection we assume that at time t, only the prediction b_t is known, not b itself, and only information about deferrable loads 1 to $N(t)$ and the expectation of future energy requests $\mathbb{E}(A(t))$ are known.

Algorithm definition: To adapt Algorithm 5.1 to deal with uncertainty, the first step is straightforward. In particular, it is natural to replace the base load b by its prediction b_t in Algorithm 5.1 to deal with the unavailability of b.

However, dealing with unavailable future deferrable load information is trickier. To do this we use a pseudo deferrable load, which is simulated at the utility, to represent future deferrable loads. More specifically, let $q = \{q(\tau)\}_{\tau=t}^{T}$ with $q(t) = 0$ denote the power consumption of the pseudo load, and assume that it requests $\mathbb{E}(A(t))$ amount of energy, i.e.,

$$\sum_{\tau=t}^{T} q(\tau) = \mathbb{E}(A(t)). \tag{5.9}$$

We also assume that q is point-wise upper and lower bounded by some upper and lower bounds $\overline{q}$ and $\underline{q}$, i.e.,

$$\underline{q}(\tau) \leq q(\tau) \leq \overline{q}(\tau), \quad \tau = t, \ldots, T. \tag{5.10}$$

Note that $\underline{q}(t) = \overline{q}(t) = 0$. The bounds $\underline{q}$ and $\overline{q}$ should be set according to historical data. Here, for simplicity, we consider them to be $\underline{q}(\tau) = 0$ and $\overline{q}(\tau) = \infty$ for $\tau = t + 1, \ldots, T$.

Given the above setup, the utility solves the following problem at every time slot $t = 1, \ldots, T$, to accommodate the availability of only partial information.

$$\begin{aligned}
\textbf{O}\text{DLC-t: } \min \quad & \sum_{\tau=t}^{T} \left(d(\tau) - \frac{1}{T-t+1} \sum_{s=t}^{T} d(s) \right)^2 && (5.11)\\
\text{over} \quad & p_n(\tau), q(\tau), d(\tau), \quad n \leq N(t), \tau \geq t \\
\text{s.t.} \quad & d(\tau) = b_t(\tau) + \sum_{n=1}^{N(t)} p_n(\tau) + q(\tau), \quad \tau \geq t; \\
& \underline{p}_n(\tau) \leq p_n(\tau) \leq \overline{p}_n(\tau), \quad n \leq N(t),\ \tau \geq t; \\
& \sum_{\tau=t}^{T} p_n(\tau) = P_n(t), \quad n \leq N(t); \\
& \underline{q}(\tau) \leq q(\tau) \leq \overline{q}(\tau), \quad \tau \geq t; \\
& \sum_{\tau=t}^{T} q(\tau) = \mathbb{E}(A(t)),
\end{aligned}$$

where $P_n(t) = P_n - \sum_{\tau=1}^{t-1} p_n(\tau)$ is the energy to be consumed at or after time t, for $n = 1, \ldots, N(t)$ and $t = 1, \ldots, T$.

Now, adjusting Algorithm 5.1 to solve ODLC-t gives Algorithm 5.2, which is real-time and shrinking-horizon. Note that if base load prediction is exact (i.e., $b_t = b$ for $t = 1, \ldots, T$) and all deferrable loads arrive at the beginning of the time horizon (i.e., $N(t) = N$ for $t = 1, \ldots, T$), then ODLC-1 reduces to ODLC and Algorithm 5.2 reduces to Algorithm 5.1.

Algorithm 5.2 Deferrable load control with uncertainty

Require: Prediction of base load b_t and the number $N(t)$ of deferrable loads. Future energy request $P_n(t)$ and power consumption bounds $\overline{p}_n$ and $\underline{p}_n$.

Ensure: At time t, output the power consumption $p_n(t)$ for deferrable loads $1, \ldots, N(t)$. At time slot $t = 1, \ldots, T$:

(i) Set $k \leftarrow 0$. Each deferrable load $n \in \{1, \ldots, N(t)\}$ initializes its schedule $\{p_n^{(0)}(\tau)\}_{\tau=t}^{T}$ as

$$p_n^{(0)}(\tau) \leftarrow \begin{cases} p_n^{(K)}(\tau) & \text{if } n \leq N(t-1) \\ 0 & \text{if } n > N(t-1) \end{cases}, \quad \tau = t, \ldots, T,$$

where $p_n^{(K)}$ is the schedule of load n in iteration K at time $t-1$.

(ii) The utility solves

$$\begin{array}{rl} \min & \displaystyle\sum_{\tau=t}^{T} \left(b_t(\tau) + \sum_{n=1}^{N(t)} p_n^{(k)}(\tau) + q(\tau) \right)^2 \\ \text{over} & q(t), \ldots, q(T) \\ \text{s.t.} & \underline{q}(\tau) \leq q(\tau) \leq \overline{q}(\tau), \quad \tau \geq t; \\ & \displaystyle\sum_{\tau=t}^{T} q(\tau) = \mathbb{E}(A(t)) \end{array}$$

to obtain a pseudo schedule $\{q^{(k)}(\tau)\}_{\tau=t}^{T}$. The utility then calculates the average aggregate load per deferrable load $g^{(k)}$ as

$$g^{(k)}(\tau) \leftarrow \frac{1}{N(t)} \left(b_t(\tau) + \sum_{n=1}^{N(t)} p_n^{(k)}(\tau) + q^{(k)}(\tau) \right)$$

for $\tau = t, \ldots, T$, and broadcasts $\{g^{(k)}(\tau)\}_{\tau=t}^{T}$ to all deferrable loads.

(iii) Each deferrable load $n = 1, \ldots, N(t)$ solves

$$\begin{array}{rl} \min & \displaystyle\sum_{\tau=t}^{T} g^{(k)}(\tau) p_n(\tau) + \frac{1}{2} \left(p_n(\tau) - p_n^{(k)}(\tau) \right)^2 \\ \text{over} & p_n(t), \ldots, p_n(T) \\ \text{s.t.} & \underline{p}_n(\tau) \leq p_n(\tau) \leq \overline{p}_n(\tau), \quad \tau \geq t; \\ & \displaystyle\sum_{\tau=t}^{T} p_n(\tau) = P_n(t), \end{array}$$

to obtain a new schedule $\{p_n^{(k+1)}(\tau)\}_{\tau=t}^{T}$, and reports to the utility.

(iv) Set $k \leftarrow k + 1$. If $k < K$, go to Step (ii).

(v) Set $p_n(t) \leftarrow p_n^{K}(t)$ and $P_n(t+1) \leftarrow P_n(t) - p_n(t)$.

Algorithm convergence results: We provide analytic guarantees on the convergence and optimality of Algorithm 5.2. In particular, we prove that Algorithm 5.2 solves ODLC-t at every time slot t. Specifically, let $\mathcal{O}(t)$ denote the set of optimal schedules to ODLC-t, and define $d(p, \mathcal{O}(t)) := \min_{(\hat{p},\hat{q}) \in \mathcal{O}(t)} \|p - \hat{p}\|$ as the distance from a schedule p to optimal schedules $\mathcal{O}(t)$ at time t, for $t = 1, \ldots, T$.

Theorem 5.3. *At time $t = 1, \ldots, T$, the deferrable load schedules $p^{(k)}$ obtained by Algorithm 5.2 converge to optimal schedules to ODLC-t, i.e., $d(p^{(k)}, \mathcal{O}(t)) \to 0$ as $k \to \infty$.*

The theorem is proved in Appendix 5.A. Though iterative, Algorithm 5.2 converges fast, similar to Algorithm 5.1. In the simulations, setting $K = 15$ is enough for all test cases.

Similar to Proposition 5.2, “t-valley-filling” provides a simple characterization of the solutions to ODLC-t. Specifically, at time $t = 1, \ldots, T$, a feasible schedule (p, q) is called t-valley-filling, if there exists some constant $C(t) \in \mathbb{R}$ such that

$$q(\tau) + \sum_{n=1}^{N(t)} p_n(\tau) = (C(t) - b_t(\tau))^+, \quad \tau = t, \ldots, T. \tag{5.12}$$

Given this definition of t-valley-filling, the following corollary follows immediately from Proposition 5.2.

Corollary 5.4. *At time $t = 1, \ldots, T$, a t-valley-filling deferrable load schedule, if exists, solves ODLC-t. Furthermore, in such cases, all optimal schedules to ODLC-t have the same aggregate load.*

This corollary serves as the basis for the performance analysis we perform in Section 5.4. Remember that t-valley-filling schedules tend to exist in cases where there are a large numbers of deferrable loads.

5.4 Performance analysis

Average-case Analysis

To this point, we have shown that Algorithm 5.2 makes “optimal” decisions with the information available at every time slot, i.e., it solves ODLC-t at time t for $t = 1, \ldots, T$. However, these decisions are still suboptimal compared to what could be achieved if exact information was available. In this section, our goal is to understand the impact of uncertainty on the performance. In particular, we study two questions:

(i) How do the uncertainties about base load and deferrable loads impact the expected sample path load variance obtained by Algorithm 5.2?

(ii) What is the improvement of using the real-time control provided by Algorithm 5.2 over using the optimal static control?

Our answers to these questions are below. Throughout, we focus on the special, but practically relevant, case when a t-valley-filling schedule exists at every time $t = 1, \ldots, T$. As we have mentioned previously, when the number of deferrable loads is large this is a natural assumption that holds for practical load profiles. The reason for making this assumption is that it allows us to use the characterization of optimal schedules given in (5.12). In fact, without loss of generality, we further assume $C(t) \geq b_t(\tau)$ for $\tau = t, \ldots, T$, under which (5.12) implies

$$d(t) = C(t) = \frac{1}{T-t+1}\left(\sum_{\tau=t}^{T} b_t(\tau) + \mathbb{E}(A(t)) + \sum_{n=1}^{N(t)} P_n(t)\right) \tag{5.13}$$

for $t = 1, \ldots, T$. Thus, equation (5.13) defines the model we use for the performance analysis of Algorithm 5.2.

The expected load variance of Algorithm 5.2

We start by calculating the expected load variance, $\mathbb{E}(V)$, of Algorithm 5.2. The goal is to understand how uncertainty about base load and deferrable loads impacts the load variance. Note that, if there are no base load prediction errors and deferrable loads arrive at the beginning of the time horizon, then Algorithm 5.2 obtains optimal schedules that have zero load variance. In contrast, when there are base load prediction errors and stochastic deferrable load arrivals, the expected load variance is given by the following theorem.

To state the result, recall that $\{f(t)\}_{t=-\infty}^{\infty}$ is the causal filter modeling the correlation of base load and define $F(t) \overset{\text{def}}{=} \sum_{s=0}^{t} f(s)$ for $t = 0, \ldots, T$.

Theorem 5.5. *The expected load variance $\mathbb{E}(V)$ obtained by Algorithm 5.2 is*

$$\mathbb{E}(V) = \frac{s^2}{T}\sum_{t=2}^{T}\frac{1}{t} + \frac{\sigma^2}{T^2}\sum_{t=0}^{T-1} F^2(t)\frac{T-t-1}{t+1}. \tag{5.14}$$

The theorem is proved in Appendix 5.A.

Theorem 5.5 explicitly states the interaction of the variability of base load prediction (σ) and deferrable load prediction (s) with the horizon length T. Besides, it highlights the correlation of base load prediction error through F. More specifically, the expected load variance $\mathbb{E}(V)$ tends to 0 as the uncertainties in base load and deferrable loads vanish, i.e., $\sigma \to 0$ and $s \to 0$.

Corollary 5.6. *The expected load variance $\mathbb{E}(V) \to 0$ as $\sigma \to 0$ and $s \to 0$.*

Another remark about Theorem 5.5 is that the two terms in (5.14) correspond to the impact of the uncertainties in deferrable loads and base load respectively. In particular, Theorem 5.5 is proved in Section 5.A by analyzing these two cases separately and then combining the results. Specifically, the following two lemmas are the key pieces in the proof of Theorem 5.5, but are also of interest in their own right.

Lemma 5.7. *If there is no base load prediction error, i.e., $b_t = b$ for $t = 1, \ldots, T$, then the expected load variance obtained by Algorithm 5.2 is*

$$\mathbb{E}(V) = s^2 \frac{\sum_{t=2}^{T} \frac{1}{t}}{T} \approx s^2 \frac{\ln T}{T}.$$

The lemma is proved in Appendix 5.A.

Lemma 5.8. *If there are no deferrable load arrivals after time 1, i.e., $N(t) = N$ for $t = 1, \ldots, T$, then the expected load variance obtained by Algorithm 5.2 is*

$$\mathbb{E}(V) = \frac{\sigma^2}{T^2} \sum_{t=0}^{T-1} F^2(t) \frac{T-t-1}{t+1}.$$

The lemma is proved in Appendix 5.A.

Lemma 5.7 highlights that the more uncertainty in deferrable load arrival, i.e., the larger s, the larger the expected load variance $\mathbb{E}(V)$. On the other hand, the longer the time horizon T, the smaller the expected load variance $\mathbb{E}(V)$.

Similarly, Lemma 5.8 highlights that a larger base load prediction error, i.e., a larger σ, results in a larger expected load variance $\mathbb{E}(V)$. However, if the impulse response $\{f(t)\}_{t=-\infty}^{\infty}$ of the modeling filter of the base load decays fast enough with t, then the following corollary highlights that the expected load variance actually tends to 0 as time horizon T increases despite the uncertainty of base load predictions.

Corollary 5.9. *If there are no deferrable load arrivals after time 1, i.e., $N(t) = N$ for $t = 1, \ldots, T$, and $|f(t)| \sim O(t^{-1/2-\alpha})$ for some $\alpha > 0$, then the expected load variance obtained by Algorithm 5.2 satisfies $\mathbb{E}(V) \to 0$ as $T \to \infty$.*

The corollary is proved in Appendix 5.A.

Improvement over static control

The goal of this section is to quantify the improvement of real-time control via Algorithm 5.2 over the optimal static (open-loop) control. To be more specific, we compare the expected load variance $\mathbb{E}(V)$ obtained by the real-time control Algorithm 5.2, with the expected load variance $\mathbb{E}(V')$ obtained by the optimal static control, which only uses base load prediction at the beginning of the time horizon (i.e., $\bar{b}$) to compute deferrable load schedules. We assume $N(t) = N$ for $t = 1, \ldots, T$ in this section since otherwise any static control cannot obtain a schedule for all deferrable loads. Thus, the interpretation of the results that follow is as a quantification of the value of incorporating updated base load predictions into the deferrable load controller.

To begin the analysis, note that $\mathbb{E}(V)$ for this setting is given in Lemma 5.8. Further, it can be verified that the optimal static control is to solve ODLC with b replaced by $\bar{b}$, and the corresponding expected load variance $\mathbb{E}(V')$ is given by the following lemma.

Lemma 5.10. *If there is no stochastic load arrival, i.e., $N(t) = N$ for $t = 1, \ldots, T$, then the expected load variance $\mathbb{E}(V')$ obtained by the optimal static control is*

$$\mathbb{E}(V') = \frac{\sigma^2}{T^2} \sum_{t=0}^{T-1} \left(T(T-t) f^2(t) - F^2(t) \right).$$

The lemma is proved in Appendix 5.A.

Next, comparing $\mathbb{E}(V)$ and $\mathbb{E}(V')$ given in Lemma 5.8 and 5.10 shows that Algorithm 5.2 always obtains a smaller expected load variance than the optimal static control. Specifically,

Corollary 5.11. *If there is no deferrable load arrival after time* 1*, i.e., $N(t) = N$ for $t = 1, \ldots, T$, then*

$$\mathbb{E}(V') - \mathbb{E}(V) = \frac{\sigma^2}{T} \sum_{t=1}^{T} \frac{1}{2t} \sum_{m=0}^{t-1} \sum_{n=0}^{t-1} (f(m) - f(n))^2 \geq 0.$$

The corollary is proved in the extended version [31].

Corollary 5.11 highlights that Algorithm 5.2 is guaranteed to obtain a smaller expected load variance than the optimal static control. The next step is to quantify how much smaller $\mathbb{E}(V)$ is in comparison with $\mathbb{E}(V')$.

To do this we compute the ratio $\mathbb{E}(V')/\mathbb{E}(V)$. Unfortunately, the general expression for the ratio is too complex to provide insight, so we consider two representative cases for the impulse response $f(t)$ of the causal filter in order to obtain insights. Specifically, we consider examples (i) and (ii) from Section 5.2. Briefly, in (i) $f(t)$ is finite and in (ii) $f(t)$ is infinite but decays exponentially in t. For these two cases, the ratio $\mathbb{E}(V')/\mathbb{E}(V)$ is summarized in the following corollaries.

Corollary 5.12. *If there is no deferrable load arrival after time* 1*, i.e.,* $N(t) = N$ *for* $t = 1, \ldots, T$*, and there exists* $\Delta > 0$ *such that*

$$f(t) = \begin{cases} 1 & \textit{if } 0 \le t < \Delta \\ 0 & \textit{otherwise,} \end{cases}$$

then

$$\frac{\mathbb{E}(V')}{\mathbb{E}(V)} = \frac{T/\Delta}{\ln(T/\Delta)} \left(1 + O\left(\frac{1}{\ln(T/\Delta)}\right)\right).$$

The corollary is proved in the extended version [31].

Corollary 5.13. *If there is no deferrable load arrival after time* 1*, i.e.,* $N(t) = N$ *for* $t = 1, \ldots, T$*, and there exists* $a \in (0, 1)$ *such that*

$$f(t) = \begin{cases} a^t & \textit{if } t \ge 0 \\ 0 & \textit{otherwise,} \end{cases}$$

then

$$\frac{\mathbb{E}(V')}{\mathbb{E}(V)} = \frac{1-a}{1+a} \frac{T}{\ln T} \left(1 + O\left(\frac{\ln \ln T}{\ln T}\right)\right).$$

The corollary is proved in the extended version [31].

Corollary 5.12 highlights that, in the case where f is finite, if we define $\lambda = T/\Delta$ as the ratio of time horizon to filter length, then the load reduction roughly scales as $\lambda/\ln(\lambda)$. Thus, the longer the time horizon is in comparison to the filter length, the larger expected load variance reduction we obtain from using Algorithm 5.2 as compared with the optimal static control.

Similarly, Corollary 5.13 highlights that, in the case where f is infinite and exponentially decaying, the expected load variance reduction scales with T as $T/\ln T$ with coefficient $(1-a)/(1+a)$. Thus, the smaller a is, which means the faster f dies out, the more load variance reduction we obtain by using real-time control. This is similar to having a smaller Δ in the previous case.

Worst-case analysis

The results surveyed above highlight that Algorithm 5.2 performs well on average; however, it is often important to guarantee more than good average case performance. For that reason, many results in the literature focus on worst case analysis, e.g., [11, 54, 57]. While no existing results apply directly to the setting of this thesis, it is straightforward to see that the worst-case performance of Algorithm 5.2 is quite bad.

To see this, let us consider a setting where the prediction error for generation, e, and deferrable load, a, have bounded deviations from their means (0 and λ respectively).

Definition 8. *We say that* ***prediction errors are bounded*** *if there exist ϵ_1 and ϵ_2 such that, at any time $t = 1, \dots, T$,*

$$|a(t) - \lambda| \le \epsilon_1, \; |e(t)| \le \epsilon_2. \tag{5.15}$$

In this situation, it is straightforward to see that the worst case performance of Algorithm 5.2 can potentially be quite bad. For two real numbers $a, b \in \mathbb{R}$, define $a \vee b := \max\{a, b\}$.

Proposition 5.14. *If a t-valley-filling solution exists for $t = 1, 2, \dots, T$, and prediction errors are bounded by ϵ_1 and ϵ_2 as in* (5.15)*, then the worst-case load variance $\sup_{a,e} V$ achieved by Algorithm 5.2 is*

$$\begin{aligned}\sup_{a,e} V \;=\; & \epsilon_1^2 \left(1 - \frac{1}{T} \sum_{k=1}^{T} \frac{1}{k}\right) \\ & + \frac{\epsilon_2^2}{T^2} \sum_{\tau=0}^{T-1} \sum_{s=0}^{T-1} \left(\frac{T}{\tau \vee s + 1} - 1\right) |F(\tau) F(s)|.\end{aligned}$$

The worst-case performance is achieved when all prediction errors on the load arrivals are equal to ϵ_1 while all prediction errors on the generation are equal to ϵ_2 in magnitude with the appropriate signs—the case where $a(t) = \lambda + \epsilon_1$ and $e(t) = \epsilon_2 \cdot \mathrm{sgn}(F(T - t))$ for all t.

Corollary 5.15. *If a t-valley-filling solution exists for $t = 1, 2, \dots, T$, and prediction errors are bounded by ϵ_1 and ϵ_2 as in* (5.15)*, then the worst-case load variance $\sup_{a,e} V$ achieved by Algorithm 5.2 is lower bounded as*

$$\sup_{a,e} V \;\ge\; \epsilon_1^2 \left(1 - \frac{1}{T} \sum_{k=1}^{T} \frac{1}{k}\right) \;\approx\; \epsilon_1^2 \left(1 - \frac{\ln T}{T}\right).$$

Interestingly, the form of Corollary 5.15 implies that, in the worst-case, Algorithm 5.2 can be as bad as having no control at all: the time averaged load variance behaves like the worst one step load variance. Meanwhile, recall from Theorem 5.5 that the average performance $\mathbb{E}(V) \to 0$ as $T \to \infty$. Therefore, while the the load variance V has a small mean $\mathbb{E}(V)$, it can be quite large in the worst case.

Distributional analysis

The contrast between the worst-case analysis (Proposition 5.14) and average-case analysis (Theorem 5.5) motivates the next main goal of this Chapter – to understand how often the "bad cases," where V takes large values, happen. That is, we want to understand what typical variations of V under Algorithm 5.2 look like.

Concentration bounds

We start with analyzing the tail probability of V. Concretely, our focus is on

$$V_\eta := \min\{c \in \mathbb{R} \mid V \leq c \text{ with probability } \eta\},$$

which denotes the minimum value c such that $V \leq c$ with probability η for $\eta \in [0, 1]$. Our main result provides upper bounds on V_η, for large values of η, for arbitrary of prediction error distributions.

More specifically, we prove that *with high probability*, the load variance of Algorithm 5.2 does not deviate much from its average-case performance, i.e., we prove a concentration result for model predictive deferable load control.

Theorem 5.16. *Suppose a t-valley filling solution exists for $t = 1, 2, \ldots, T$, and prediction errors bounded by ϵ_1 and ϵ_2 as in* (5.15). *Then the distribution of the load variance V obtained by Algorithm 5.2 satisfies a Bernstein type concentration, i.e.,*

$$\mathbb{P}(V - \mathbb{E}V > t) \leq \exp\left(\frac{-t^2}{16\epsilon^2\lambda_1(2\mathbb{E}V + t)}\right), \tag{5.16}$$

where $\epsilon = \max(\epsilon_1, \epsilon_2)$ *and*

$$\lambda_1 = \frac{\ln T}{T} + \frac{1}{T^2}\sum_{t=0}^{T-1} F^2(t)\frac{T - t + 1}{t + 1}.$$

The theorem is proven in Appendix 5.B.

To build intuition, the tail probability bound of V in (5.16) can be simplified for two different regimes of t as

$$\mathbb{P}(V - \mathbb{E}V > t) \leq \begin{cases} \exp\left(\frac{-t^2}{48\epsilon^2\lambda_1\mathbb{E}V}\right), & t < \mathbb{E}V \\ \exp\left(\frac{-t}{48\epsilon^2\lambda_1}\right), & t \geq \mathbb{E}V. \end{cases} \tag{5.17}$$

Though looser than that in (5.16), the tail bound in (5.17) highlights that V has a Gaussian tail probability bound when $t < \mathbb{E}V$ and an Exponential tail probability bound when $t \geq \mathbb{E}V$.

Theorem 5.16 relates the tail behavior of V with the maximum prediction error ϵ and the error correlation F over time. It implies that the actual performance of Algorithm 5.2 does not deviate much from its mean. To illustrate this, consider the following example where the prediction on baseload is precise, since the parameter λ_1 has a simple expression in this scenario.

Example 5.17. *Suppose that the baseload prediction is precise, i.e.,* $\epsilon_2 = 0$*. Then the average load variance is*

$$\mathbb{E}[V] = \frac{s^2}{T}\sum_{t=2}^{T}\frac{1}{t} \approx s^2 \ln T/T$$

and the tail bound in Theorem 5.16 can be simplified as

$$\mathbb{P}(V - \mathbb{E}V > c\mathbb{E}V) \leq \exp\left(-\frac{c^2}{2+c}\frac{s^2}{16\epsilon^2}\right).$$

Recall that constant s *is the variance of* a *and constant* ϵ *is the maximum deviation of* a *from its mean. The above expression shows that, with high probability,* V *is at most a constant* $c + 1$ *times of its mean* $\mathbb{E}V$*.*

More generally, the quantity λ_1 controls the decaying speed of the tail bound in (5.16): the smaller λ_1, the faster the tail bound $\mathbb{P}(V - \mathbb{E}V > t)$ decays in t, and the load variance V achieved by Algorithm 5.2 concentrates sharper around its mean $\mathbb{E}V$. The following corollary highlights that λ_1 tends to 0 as T increases, provided that the error correlation $f(t)$ decays fast enough in t. Note that the condition on f is the same for Corollary 5.18 and Theorem 5.5.

Corollary 5.18. *Under the assumptions of Theorem 5.16, if the error correlation* $f \sim O(t^{-\frac{1}{2}-\alpha})$ *for some* $\alpha > 0$*, then* $\lambda_1 \to 0$ *as* $T \to \infty$*.*

A detailed proof of Theorem 5.16 is included in the Appendix; however it is useful to provide some informal intuition for the argument used.

In general, tail probability bounds can be obtained by controlling the moments of a random variable. For example, the Markov inequality gives inverse linear tail probability bound using the first moment, and the Chebyshev inequality provides inverse quadratic tail probability bound using the second moment. However, the bound we obtained in Theorem 5.16 approaches 0 much faster for large t than the aforementioned Markov and Chebyshev bounds. This is done by controlling the moment generating function of V using the convex Log-Sobolev inequality.

A challenge in controlling the moment generating function of V is that, the most commonly used approach—the Martingale bounded difference approach [64]—only obtains very loose tail probability bounds in our case. This is because V can change dramatically when one of the sources $a(t)$ or $e(t)$ of the randomness changes. Instead, we exploit the fact that the gradient of V is bounded by a linear function of itself (similar but slightly different from the "self-bounding" property defined in [15]). Using this property together with Log-Sobolev inequality in the product measure gives us a nice way to bound the entropy of V. After this we apply the Herbst's argument [53] to compute a good estimate on the concentration of V.

Bounds on the variance

To further understand the scale of typical load variance V under Algorithm 5.2, it is useful to also study the variance. In addition, the form of the variance highlights the impact of the tight concentration shown in Theorem 5.16.

Theorem 5.19. *Suppose a t-valley-filling solution exists for $t = 1, 2, \dots, T$, and prediction errors are bounded by ϵ_1 and ϵ_2 as in* (5.15). *Then the variance* $\mathrm{var}(V)$ *of V obtained by Algorithm 5.2 is bounded above by*

$$\mathrm{var}(V) \le \left(\frac{4\epsilon_1 s \ln T}{T}\right)^2 + \left(\frac{4\epsilon_2 \sigma}{T^2}\sum_{t=0}^{T-1} F^2(t)\frac{T-t+1}{t+1}\right)^2. \tag{5.18}$$

To interpret this result, let $\overline{\mathrm{var}(V)}$ denote the upper bound on $\mathrm{var}(V)$ provided in (5.18). Theorem 5.19 implies that $\mathbb{E}V$ and $\sqrt{\overline{\mathrm{var}(V)}}$ scale similarly with T. In particular, the first term $\frac{s^2}{T}\sum_{t=2}^{T}\frac{1}{t}$ in $\mathbb{E}V$ scales with T as $\Omega(\ln T/T)$ while the first term $(4\epsilon_1 s \ln T/T)^2$ in $\overline{\mathrm{var}(V)}$ scales with T as $\Omega\left((\ln T/T)^2\right)$, and the second terms in $\mathbb{E}V$ and $\overline{\text{var}(V)}$ have the same relationship. Therefore, the standard deviation $\sqrt{\text{var}(V)}$, which is upper bounded by $\sqrt{\overline{\text{var}(V)}}$, is at most on the same scale as $\mathbb{E}V$ as T expands. It immediately follows from the Chebyshev inequality that V can only deviate significantly from $\mathbb{E}(V)$ with a small probability.

Corollary 5.20. *Under the assumptions in Theorem 5.19, for $t > 0$,*

$$\begin{aligned}&\mathbb{P}(|V - \mathbb{E}V| > t)\\ &\le \frac{1}{t^2}\left[\left(\frac{4\epsilon_1 s \ln T}{T}\right)^2 + \left(\frac{4\epsilon_2\sigma}{T^2}\sum_{\tau=0}^{T-1} F^2(\tau)\frac{T-\tau+1}{\tau+1}\right)^2\right].\end{aligned} \tag{5.19}$$

While the tail bound (5.16) in Theorem 5.16 scales at least exponentially in t, the Chebyshev inequality only provides a tail bound (5.19) that scales inverse quadratically in t. Hence for large t, (5.16) provides a much tighter tail bound. However for small values of t, the tail bound (5.19) is usually tighter since the variance var(V) is well estimated in (5.18).

Furthermore, the variance var(V) vanishes as T expands, provided that $f(t)$ decays sufficiently fast as t grows, as formally stated in the following corollary.

Corollary 5.21. *Under the assumptions of Theorem 5.19, if the error correlation $f \sim O(t^{-\frac{1}{2}-\alpha})$ for some $\alpha > 0$, then* $\text{var}(V) \to 0$ *as $T \to \infty$.*

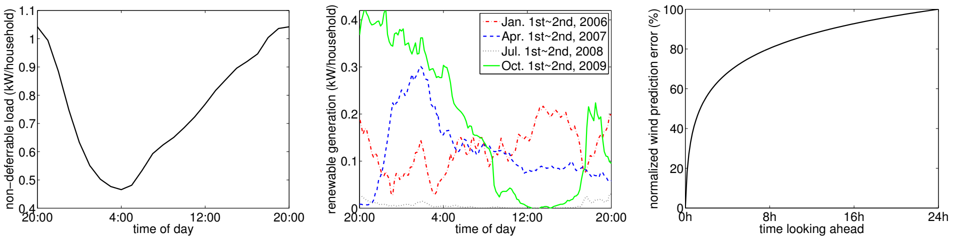
Note that the condition on f parallels that in Theorem 5.5.

5.5 Simulation

In this chapter we use trace-based experiments to explore the generality of the analytic results in the previous section. In particular, the results in the previous section characterize the expected load variance obtained by Algorithm 5.2 as a function of prediction uncertainties, and quantify the improvement of Algorithm 5.2 over the optimal static (open-loop) controller. However, the analytic results make simplifying assumptions on the form of uncertainties and solution schedules (equation (5.13)). Therefore, it is important to assess the performance of the algorithm using real-world data.

Experimental setup

The numerical experiments we perform use a time horizon of 24 hours, from 20:00 to 20:00 on the following day. The time slot length is 10 minutes, which is the granularity of the data we have obtained about renewable generation.



(a) non-deferrable load

(b) wind generation

(c) prediction error over time

Figure 5.2: Illustration of the traces used in the experiments. (a) shows the average residential load in the service area of Southern California Edison in 2012. (b) shows the total wind power generation of the Alberta Electric System Operator scaled to represent 20% penetration. (c) shows the normalized root-mean-square wind prediction error as a function of the time looking ahead for the model used in the experiments.

Base load Recall that base load is a combination of non-deferrable load and renewable generation. The non-deferrable load traces used in the experiments come from the average residential load in the service area of Southern California Edison in 2012 [78]. In the simulations, we assume that non-deferrable load is precisely known so that uncertainties in the base load only come from renewable generation. In particular, non-deferrable load over the time horizon of a day is taken to be the average over the 366 days in 2012 as in Figure 5.2a, and assumed to be known to the utility at the beginning of the time horizon. In practice, non-deferrable load at the substation feeder level can be predicted within 1–3% root-mean-square error looking 24 hours ahead [25].

The renewable generation traces we use come from the 10-minute historical data for total wind power generation of the Alberta Electric System Operator from 2004 to 2009 [4]. In the simulations, we scale the wind power generation so that its average over the 6 years corresponds to a number of penetration levels in the range between 5% and 30%, and pick the wind power generation of a randomly chosen day as the renewable generation during each run. Figure 5.2b shows the wind power generation for four representative days, one for each season, after scaling to 20% penetration.

We assume that the renewable generation is not precisely known until it is realized, but that a prediction of the generation, which improves over time, is available to the utility. The modeling of prediction evolution over time is according to a martingale forecasting process [37, 38], which is a standard model for an unbiased prediction process that improves over time.

Specifically, the prediction model is as follows: for wind generation $w(\tau)$ at time τ, the prediction error $w_t(\tau) - w(\tau)$ at time $t < \tau$ is the sum of a sequence of independent random variables $n_s(\tau)$ as

$$w_t(\tau) = w(\tau) + \sum_{s=t+1}^{\tau} n_s(\tau), \quad 0 \le t < \tau \le T.$$

Here $w_0(\tau)$ is the wind prediction without any observation, i.e., the expected wind generation $\bar{w}(\tau)$ at the beginning of the time horizon (used by static control).

The random variables $n_s(\tau)$ are assumed to be Gaussian with mean 0. Their variances are chosen as

$$\mathbb{E}(n_s^2(\tau)) = \frac{\sigma^2}{\tau - s + 1}, \quad 1 \le s \le \tau \le T,$$

where $\sigma > 0$ is such that the root-mean-square prediction error $\sqrt{\mathbb{E}(w_0(T) - w(T))^2}$ looking T time slots (i.e., 24 hours) ahead is 0%–22.5% of the nameplate wind generation capacity.[2] According to this choice of the variances of $n_s(\tau)$, root-mean-square prediction error only depends on how far ahead the prediction is, in particular as in Figure 5.2c. This choice is motivated by [34].

Deferrable loads For simplicity, we consider the hypothetical case where all deferrable loads are electric vehicles. Since historical data for electric vehicle usage is not available, we are forced to use synthetic traces for this component of the experiments. Specifically, in the simulations the electric vehicles are considered to be identical, each requests 10kWh electricity by a deadline 8 hours after it arrives, and each must consume power at a rate within $[0, 3.3]$kW after it arrives and before its deadline.

In the simulations, the arrival process starts at 20:00 and ends at 12:00 the next day so that the deadlines of all electric vehicles lie within the time horizon of 24 hours. In each time slot during the arrival process, we assume that the number of arriving electric vehicles is uniformly distributed in $[0.8\lambda, 1.2\lambda]$, where λ is chosen so that electric vehicles (on average) account for 5%–30% of the non-deferrable loads. While this synthetic workload is simplistic, the results we report are representative of more complex setups as well.

Uncertainty about deferrable load arrivals is captured as follows. The prediction $\mathbb{E}(A(t))$ of future deferrable load total energy request is simply the arrival rate λ

[2] Average wind generation is 15% of the nameplate capacity, so the root-mean-square prediction error looking T time slots ahead is 0%–150% the average wind generation.

times the length of the rest of the arrival process $T' - t$ where T' is the end of the arrival process (12:00), i.e.,

$$\mathbb{E}(A(t)) = \lambda(T' - t), \quad t = 1, \ldots, T'.$$

If $t > T'$, i.e., the deferrable load arrival process has ended, then $\mathbb{E}(A(t)) = 0$.

Baselines for comparison Our goal in the simulations is to contrast the performance of Algorithm 5.2 with a number of common benchmarks to tease apart the impact of real-time control and the impact of different forms of uncertainty. To this end, we consider four controllers in our experiments:

(i) *Offline optimal control:* The controller has full knowledge about the base load and deferrable loads, and solves the ODLC problem offline. It is not realistic in practice, but serves as a benchmark for the other controllers since offline optimal control obtains the smallest possible load variance.

(ii) *Static control with exact deferrable load arrival information*: The controller has full knowledge about deferrable loads (including those that have not arrived), but uses only the prediction of base load that is available at the beginning of the time horizon to compute a deferrable load schedule that minimizes the expected load variance. This static control is still unrealistic since a deferrable load is known only after it arrives. But, this controller corresponds to what is considered in prior works, e.g., [27, 28, 62].

(iii) *Real-time control with exact deferrable load arrival information.* The controller has full knowledge about deferrable loads (including those that have not arrived), and uses the prediction of base load that is available at the current time slot to update the deferrable load schedule by minimizing the expected load variance to go, i.e., Algorithm 5.2 with $N(t) = N$ for $t = 1, \ldots, T$. The control is unrealistic since a deferrable load is known only after it arrives; however it provides the natural comparison for case (ii) above.

(iv) *Real-time control without exact deferrable load arrival information, i.e., Algorithm 5.2.* This corresponds to the realistic scenario where only predictions are available about future deferrable loads and base load. The comparison with case (iii) highlights the impact of deferrable load arrival uncertainties.

The performance measure that we show in all plots is the “suboptimality” of the controllers, which we define as

$$\eta \overset{\text{def}}{=} \frac{V - V^{\text{opt}}}{V^{\text{opt}}},$$

where V is the load variance obtained by the controller and V^{opt} is the load variance obtained by the offline optimal, i.e., case (i) above. Thus, the lines in the figures correspond to cases (ii)-(iv).

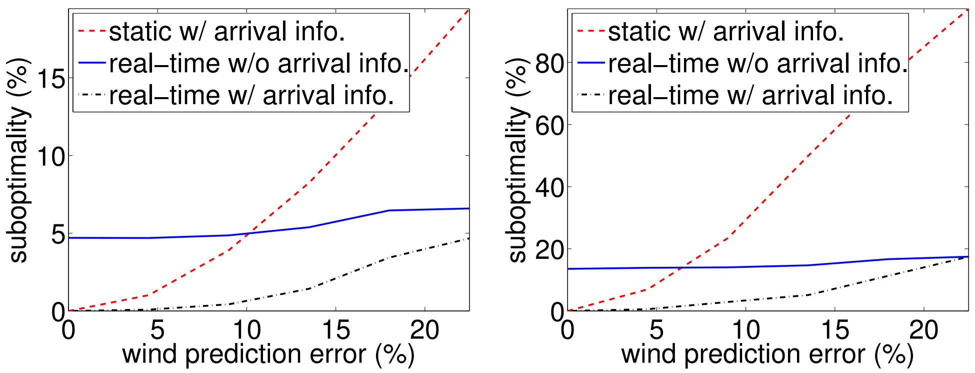
Experimental results

Our experimental results focus on two main goals: (i) understanding the impact of prediction accuracy on the expected load variance obtained by deferrable load control algorithms, and (ii) contrasting the real-time (closed-loop) control of Algorithm 5.2 with the optimal static (open-loop) controller. We focus on the impact of three key factors: wind prediction error, the penetration of deferrable load, and the penetration of renewable energy.

The impact of prediction error To study the impact of prediction error, we fix the penetration of both renewable generation (wind) and deferrable loads at 10% of non-deferrable load, and simulate the load variance obtained under different levels of root-mean-square wind prediction errors (0%–22.5% of the nameplate capacity looking 24 hours ahead). The results are summarized in Figure 5.3a. It is not surprising that suboptimality of both the static and the real-time controllers that have exact information about deferrable load arrivals is zero when the wind prediction error is 0, since there is no uncertainty for these controllers in this case.

As prediction error increases, the suboptimality of both the static and the real-time control increases. However, notably, the suboptimality of real-time control grows much more slowly than that of static control, and remains small (<4.7%) if deferrable load arrivals are known, over the whole range 0%–22.5% of wind prediction error. At 22.5% prediction error, the suboptimality of static control is 4.2 times that of real-time control. This highlights that real-time control mitigates the influence of imprecise base load prediction over time.

Moving to the scenario where deferrable load arrivals are not known precisely, we see that the impact of this inexact information is less than 6.6% of the optimal variance. However, real-time control yields a load variance that is surprisingly resilient to the growth of wind prediction error, and eventually beats the optimal



(a) Wind and deferrable load penetration are both 10%.

(b) Wind and deferrable load penetration are both 20%.

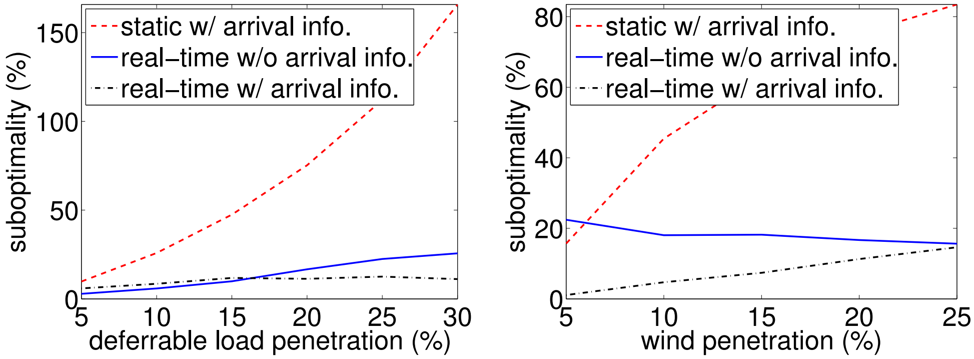
Figure 5.3: Illustration of the impact of wind prediction error on suboptimality of load variance.

static control at around 10% wind prediction error, even though the optimal static control has exact knowledge of deferrable loads and the adaptive control does not.

As prediction error increases, the suboptimality of the real-time control with or without deferrable load arrival information gets close, i.e., the benefit of knowing additional information on future deferrable load arrivals vanishes as base load uncertainty increases. This is because the additional information is used to overfit the base load prediction error.

The same comparison is shown in Figure 5.3b for the case where renewable and deferrable load penetration are both 20%. Qualitatively the conclusions are the same, however at this higher penetration the contrast between the resilience of adaptive control and static control is magnified, while the benefit of knowing deferrable load arrival information is minified. In particular, real-time control without arrival information beats static control with arrival information, at a lower (around 7%) wind prediction error, and knowing deferrable load arrival information does not reduce suboptimality of real-time control with 22.5% wind prediction error.

The impact of deferrable load penetration Next, we look at the impact of deferrable load penetration on the performance of the various controllers. To do this, we fix the wind penetration level to be 20% and wind prediction error looking 24 hours ahead to be 18%, and simulate the load variance obtained under different deferrable load penetration levels (5%–30%). The results are summarized in Figure 5.4a.



(a) Impact of deferrable load penetration (b) Impact of wind penetration

Figure 5.4: Suboptimality of load variance as a function of (a) deferrable load penetration and (b) wind penetration. In (a) the wind penetration is 20% and in (b) the deferrable load penetration is 20%. In both, the wind prediction error looking 24 hours ahead is 18%.

Not surprisingly, if future deferrable loads are known and uncertainty only comes from base load prediction error, then the suboptimality of real-time control is very small (<11.2%) over the whole range 5%–30% of deferrable load penetration, while the suboptimality of static control increases with deferrable load penetration, up to as high as 166% (14.9 times that of real-time control) at 30% deferrable load penetration.

However, without knowing future deferrable loads, the suboptimality of real-time control increases with the deferrable load penetration. This is because a larger amount of deferrable loads introduces larger uncertainties in deferrable load arrivals. But the suboptimality remains smaller than that of static control over the whole range 5%–30% of deferrable load penetration. The highest suboptimality 25.7% occurs at 30% deferrable load penetration, and is less than 1/6 of the suboptimality of static control, which assumes exact deferrable load arrival information.

The impact of renewable penetration Finally, we study the impact of renewable penetration. To do this we fix the deferrable load penetration level to be 20% and the wind prediction error looking 24 hours ahead to be 18%, and simulate the load variance obtained by the 4 test cases under different wind penetration levels (5%–25%). The results are summarized in Figure 5.4b.

A key observation is that if future deferrable loads are known and uncertainty only comes from base load prediction error, then the suboptimality of real-time control grows much slower than that of static control, as wind penetration level increases.

As explained before, this highlights that real-time control mitigates the impact of base load prediction error over time. In fact, the suboptimality of real-time control is small (<15%) over the whole range 5%–25% of wind penetration levels. Of course, without knowledge of future deferrable loads, the suboptimality of real-time control becomes bigger. However, it still eventually outperforms the optimal static controller at around 6% wind penetration, despite the fact that the optimal static controller is using exact information about deferrable loads.

A case study

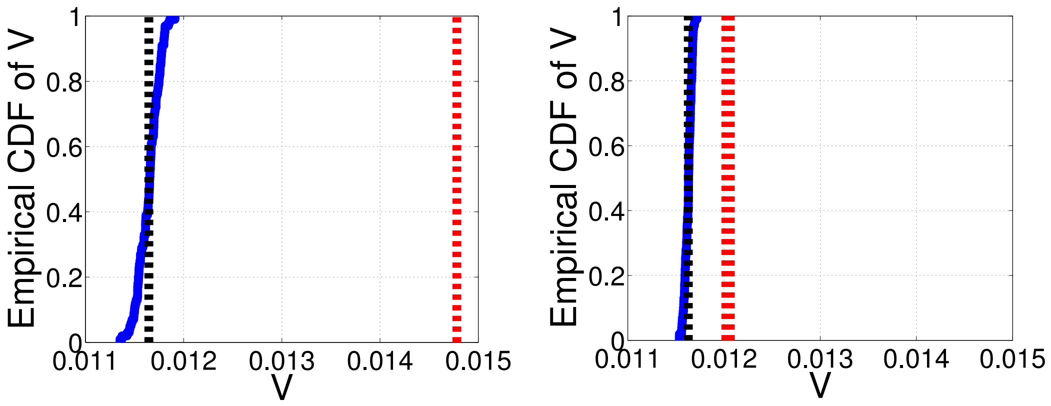
Theorems 5.16 and 5.19 provide theoretical guarantees that the load variance V obtained by Algorithm 5.2 concentrates around its mean, if prediction errors are bounded as in (5.15) and error correlation decays sufficiently fast (c.f. Corollary 2). Thus, they give the intuition that the expected performance of Algorithm 5.2 is a useful metric to focus on, and does indeed give an indication of the "typical" performance of the algorithm.

However, our analysis is based on the assumption that a t-valley-filling solution exists, which relies on the penetration of deferrable load being high enough. This is a necessary technical assumption for our analysis, and has been used by the previous analysis of Algorithm 5.2 as well, e.g., [30].

Given this assumption in the analytic results, it is important to understand the robustness of the results to this assumption. To that end, here we provide a case study to demonstrate that this intuition is robust to the t-valley-filling assumption.

In our case study, we mimic the setting of [30], where an average-case analysis of Algorithm 5.2 is performed. In particular, we use 24 hour residential load trace in the Southern California Edison (SCE) service area averaged over the year 2012 and 2013 [79] as the non-deferrable load, and wind power generation data from the Alberta Electric System Operator from 2004 to 2012 [5]. The wind power generation data is scaled so that its average over 9 years corresponds to 30% penetration level, and pick the wind generation of a random day as renewable during each run. We generate random prediction error in baseload and arrival of deferrable load similar to [30].

Given this setting, we simulate 100 instances in each scenario and compare the results with the Theorems 5.16. The results are shown in Fig. 5.5 where we plot the cumulative distribution (CDF) of the load variance produced by Algorithm 5.2 under two different scenarios. Specifically, in Fig. 5.5a, we assume the prediction



(a) 30% prediction error

(b) 10% prediction error

Figure 5.5: The empirical cumulative distribution function of the load variance under Algorithm 5.2 over 24 hour control horizon using real data. The red line represents the analytic bound on the 90% confidence interval computed from Theorem 5.16, and the black line shows the empirical mean.

error in wind power generation is 30%, and in Fig. 5.5b, we assume the prediction error is 10%. We plot the CDF on the same scale in both plots and additionally show an analytic bound on the 90% confidence interval computed from Theorem 5.16. For both cases, the results highlight a strong concentration around the mean, and the analytic bound from Theorem 5.16 is valid despite the fact that the t-valley-filling assumption is not satisfied. Further, note that the analytic bound is much tighter when prediction error is small, which coincides the statement of Theorem 5.16.

5.6 Concluding remarks

We have proposed a model predictive algorithm for decentralized deferrable load control that can schedule a large number of deferrable loads to compensate for the random fluctuations in renewable generation. At any time, the algorithm incorporates updated predictions about deferrable loads and renewable generation to minimize the expected load variance to go. Further, by modeling the base load prediction updates as a Wiener filtering process, we have conducted performance analysis to our algorithm in average case analysis and distributional analysis. We derived an explicit expression for the aggregate load variance obtained by the average case performance of the algorithm, which quantitatively showed the improvement of model predictive control over static control. Interestingly, the sub-optimality of static control is $O(T/\ln T)$ times that of real-time control in two representative cases of base load prediction updates. Besides average case analysis, we have provided a distributional analysis of the algorithm and shown that the load variance is tightly concentrated around its mean. Thus, our results highlight that the typical performance one should expect to see under model predictive deferrable load control is not-too-different from the average-case analysis. Importantly, the proof technique we develop may be useful for the analysis of model predictive control in more general settings as well. The qualitative insights from the analytic results were validated using trace-based simulations, which confirm that the algorithm has significantly smaller sub-optimality than the optimal static control.

The main limitation in our analysis (which is also true for the prior stochastic analysis of model predictive deferrable load control) is the assumption that a t-valley-filling solution exists. Practically, one can expect this to be satisfied if the penetration of deferrable loads is high; however, relaxing the need for this technical assumption remains an interesting and important challenge. Interestingly, the numerical results we report here highlight that one should also expect a tight concentration in the case where a t-valley-filling solution does not exist.

There remain many open questions on deferrable load control. For example, is it possible to reduce the communication and computation requirements of the proposed algorithm by assuming achievability of t-valley-filling? How to extend the algorithm to a receding horizon implementation? Additionally, how to apply the technique used here to incorporate prediction evolution for other demand response settings.

5.A Proof of average case results

In this section, we only include proofs of the main results due to space restrictions. The remainder of the proofs can be found in the extended version [31].

Proof of Theorem 5.3

For brevity and without loss of generality, we prove Theorem 5.3 for $t = 1$ only. Thus, we can abbreviate b_t and $N(t)$ by b and N respectively without introducing confusion.

For feasible p, q to ODLC-t and $p = (p_1, \dots, p_N)$, define

$$L(p,q) = \sum_{\tau=1}^{T} \left(b(\tau) + \sum_{n=1}^{N} p_n(\tau) + q(\tau) \right)^2 .$$

Since the sum of the aggregate load $\sum_{\tau=1}^{T} d(\tau)$ is a constant, minimizing the ℓ_2 norm of the aggregate load is equivalent to minimizing its variance. Therefore, if subject to the same constraints, the minimizer of L is also the solution to ODLC-t. According to the proof of Proposition 5.1 in [27], we have

$$L(p^{(k+1)}, q^{(k)}) \leq L(p^{(k)}, q^{(k)})$$

for $k \geq 0$, and the equality is attained if and only if $p^{(k+1)} = p^{(k)}$ and $p^{(k)}$ minimizes $L(p, q^{(k)})$ over all feasible p, i.e., (the first order optimality condition)

$$\left\langle b + \sum_{n=1}^{N} p_n^{(k)} + q^{(k)},\ p_n' - p_n^{(k)} \right\rangle \geq 0$$

for $n = 1, \ldots, N$ and all feasible p_n'. According to Step (ii) of Algorithm 5.2, it is straightforward that

$$L(p^{(k+1)}, q^{(k+1)}) \leq L(p^{(k+1)}, q^{(k)})$$

for $k \geq 0$, and the equality is attained if and only if $q^{(k+1)} = q^{(k)}$ and $q^{(k)}$ minimizes $L(p^{(k+1)}, q)$ over all feasible q, i.e., (the first order optimality condition)

$$\left\langle b + \sum_{n=1}^{N} p_n^{(k+1)} + q^{(k)},\ q' - q^{(k)} \right\rangle \geq 0$$

for all feasible q'. It then follows that

$$L(p^{(k+1)}, q^{(k+1)}) \leq L(p^{(k)}, q^{(k)})$$

and the equality if attained if and only if $(p^{(k+1)}, q^{(k+1)}) = (p^{(k)}, q^{(k)})$, and

$$\begin{aligned} \left\langle b + \sum_{n=1}^{N} p_n^{(k)} + q^{(k)},\ p_n' - p_n^{(k)} \right\rangle &\geq 0, \\ \left\langle b + \sum_{n=1}^{N} p_n^{(k)} + q^{(k)},\ q' - q^{(k)} \right\rangle &\geq 0 \end{aligned}$$

for all feasible p and q, i.e., $(p^{(k)}, q^{(k)})$ minimizes $L(p, q)$. Then by Lasalle's Theorem [85], we have $d(p^{(k)}, \mathcal{O}(t)) \to 0$ as $k \to \infty$. ■

Proof of Lemma 5.7

When $b_t = b$ and $\mathbb{E}(a(t)) = \lambda$ for $t = 1, \ldots, T$, the model (5.13) for Algorithm 5.2 reduces to

$$d(t) = \frac{1}{T - t + 1} \left(\sum_{\tau=t}^{T} b(\tau) + \lambda(T - t) + \sum_{n=1}^{N(t)} P_n(t) \right) \tag{5.20}$$

for $t = 1, \ldots, T$. Then

$$(T - t + 1)d(t) = \sum_{\tau=t}^{T} b(\tau) + \lambda(T - t) + \sum_{n=1}^{N(t)} P_n(t)$$

$$(T - t + 2)d(t - 1) = \sum_{\tau=t-1}^{T} b(\tau) + \lambda(T - t + 1) + \sum_{n=1}^{N(t-1)} P_n(t - 1)$$

for $t = 2, \ldots, T$. Subtract the two equations and simplify using the fact that $b(t - 1) + \sum_{n=1}^{N(t-1)} (P_n(t - 1) - P_n(t)) = b(t - 1) + \sum_{n=1}^{N(t-1)} p_n(t - 1) = d(t - 1)$ and the definition of $a(t)$ to obtain

$$d(t) - d(t - 1) \quad = \quad \frac{1}{T - t + 1} (a(t) - \lambda)$$

for $t = 2, \ldots, T$. Substituting $t = 1$ into (5.20), it can be verified that $d(1) = \lambda + \sum_{\tau=1}^{T} b(\tau)/T + (a(1) - \lambda)/T$, therefore

$$d(t) = \lambda + \frac{1}{T} \sum_{\tau=1}^{T} b(\tau) + \sum_{\tau=1}^{t} \frac{1}{T - \tau + 1} (a(\tau) - \lambda)$$

for $t = 1, \ldots, T$. The average aggregate load is

$$u = \frac{1}{T} \sum_{t=1}^{T} d(t) = \lambda + \frac{1}{T} \left(\sum_{\tau=1}^{T} b(\tau) + \sum_{\tau=1}^{T} (a(\tau) - \lambda) \right).$$

Therefore,

$$\begin{aligned}
& \mathbb{E}(d(t) - u)^2 \\
= \quad & \mathbb{E} \left(\sum_{\tau=1}^{t} \frac{1}{T - \tau + 1} (a(\tau) - \lambda) - \frac{1}{T} \sum_{\tau=1}^{T} (a(\tau) - \lambda) \right)^2 \\
= \quad & \mathbb{E} \left(\sum_{\tau=1}^{t} \frac{\tau - 1}{T(T - \tau + 1)} (a(\tau) - \lambda) - \frac{1}{T} \sum_{\tau=t+1}^{T} (a(\tau) - \lambda) \right)^2 \\
= \quad & \frac{s^2}{T^2} \left(\sum_{\tau=1}^{t} \frac{(\tau - 1)^2}{(T - \tau + 1)^2} + T - t \right)
\end{aligned}$$

for $t = 1, \ldots, T$. The last equality holds because $(a(\tau) - \lambda)$ are independent for all τ and each of them have mean zero and variance s^2. It follows that

$$\begin{aligned}
\mathbb{E}(V) &= \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}(d(t)-u)^2 \\
&= \frac{s^2}{T^3}\left(\sum_{t=1}^{T}\sum_{\tau=1}^{t}\frac{(\tau-1)^2}{(T-\tau+1)^2} + \sum_{t=1}^{T}(T-t)\right) \\
&= \frac{s^2}{T^3}\left(\sum_{\tau=1}^{T}\frac{(\tau-1)^2}{T-\tau+1} + \sum_{t=1}^{T}(T-t)\right) \\
&= \frac{s^2}{T^3}\left(\sum_{t=1}^{T}\frac{(T-t)^2}{t} + \sum_{t=1}^{T}\frac{(T-t)t}{t}\right) \\
&= s^2\frac{\sum_{t=2}^{T}\frac{1}{t}}{T} \sim s^2\frac{\ln T}{T}. \qquad \blacksquare
\end{aligned}$$

Proof of Lemma 5.8

In the case where no deferrable arrival after $t = 1$, i.e., $N(t) = N$ for $t = 1, \ldots, T$, the model (5.13) for Algorithm 5.2 reduces to

$$(T-t+1)d(t) = \sum_{\tau=t}^{T} b_t(\tau) + \sum_{n=1}^{N} P_n(t) \tag{5.21}$$

for $t = 1, \ldots, T$. Substitute t by $t-1$ to obtain

$$(T-t+2)d(t-1) = \sum_{\tau=t-1}^{T} b_{t-1}(\tau) + \sum_{n=1}^{N} P_n(t-1)$$

for $t = 2, \ldots, T$. Subtract the two equations to obtain

$$\begin{aligned}
&(T-t+1)d(t) - (T-t+2)d(t-1) \\
=\ & \sum_{\tau=t}^{T} e(t)f(\tau-t) - b(t-1) - \sum_{n=1}^{N} p_n(t-1) \\
=\ & e(t)F(T-t) - d(t-1),
\end{aligned}$$

which implies

$$d(t) - d(t-1) = \frac{1}{T-t+1}e(t)F(T-t)$$

for $t = 2, \ldots, T$. Substituting $t = 1$ into (5.21) and recalling the definition of b_t in (5.1), it can be verified that

$$d(1) = \frac{1}{T}\left(\sum_{n=1}^{N} P_n + \sum_{\tau=1}^{T}\bar{b}(\tau)\right) + \frac{1}{T}e(1)F(T-1).$$

Therefore,

$$d(t) = \frac{1}{T}\left(\sum_{n=1}^{N} P_n + \sum_{\tau=1}^{T} \bar{b}(\tau)\right) + \sum_{\tau=1}^{t} \frac{1}{T-\tau+1} e(\tau)F(T-\tau)$$

for $t = 1, \ldots, T$. The average aggregate load is

$$u \;=\; \frac{1}{T}\left(\sum_{n=1}^{N} P_n + \sum_{t=1}^{T} \bar{b}(t)\right) + \frac{1}{T}\sum_{\tau=1}^{T} e(\tau)F(T-\tau).$$

Therefore,

$$\begin{aligned}
& \mathbb{E}(d(t)-u)^2 \\
=\;& \mathbb{E}\left(\sum_{\tau=1}^{t} \frac{1}{T-\tau+1} e(\tau)F(T-\tau) - \sum_{\tau=1}^{T} \frac{1}{T} e(\tau)F(T-\tau)\right)^2 \\
=\;& \mathbb{E}\left(\sum_{\tau=1}^{t} \frac{\tau-1}{T(T-\tau+1)} e(\tau)F(T-\tau)\right. \\
& \qquad \left. - \sum_{\tau=t+1}^{T} \frac{1}{T} e(\tau)F(T-\tau)\right)^2 \\
=\;& \frac{\sigma^2}{T^2}\left(\sum_{\tau=1}^{t} \frac{(\tau-1)^2}{(T-\tau+1)^2} F^2(T-\tau) + \sum_{\tau=t+1}^{T} F^2(T-\tau)\right)
\end{aligned}$$

for $t = 1, \ldots, T$. The last equality holds because $e(\tau)$ are uncorrelated random variables with mean zero and variance σ^2. It follows that

$$\begin{aligned}
& \mathbb{E}(V) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}(d(t)-u)^2 \\
=\;& \frac{\sigma^2}{T^3}\sum_{t=1}^{T}\left(\sum_{\tau=1}^{t} \frac{(\tau-1)^2}{(T-\tau+1)^2} F^2(T-\tau) + \sum_{\tau=t+1}^{T} F^2(T-\tau)\right) \\
=\;& \frac{\sigma^2}{T^3}\sum_{\tau=1}^{T} F^2(T-\tau)\frac{(\tau-1)^2}{T-\tau+1} + \frac{\sigma^2}{T^3}\sum_{\tau=2}^{T} (\tau-1)F^2(T-\tau) \\
=\;& \frac{\sigma^2}{T^2}\sum_{\tau=1}^{T} F^2(T-\tau)\frac{\tau-1}{T-\tau+1} = \frac{\sigma^2}{T^2}\sum_{t=0}^{T-1} F^2(t)\frac{T-t-1}{t+1}. \blacksquare
\end{aligned}$$

Proof of Theorem 5.5

Similar to the proof of Lemma 5.7 and 5.8, use the model (5.13) to obtain

$$d(t) = \lambda + \frac{1}{T}\sum_{\tau=1}^{T} \bar{b}(\tau) + \sum_{\tau=1}^{t} \frac{1}{T-\tau+1}\left(e(\tau)F(T-\tau) + a(\tau) - \lambda\right)$$

for $t = 1, \ldots, T$ and

$$u = \lambda + \frac{1}{T}\sum_{\tau=1}^{T}\bar{b}(\tau) + \sum_{\tau=1}^{T}\frac{1}{T}\left(e(\tau)F(T-\tau) + a(\tau) - \lambda\right).$$

Therefore,

$$\begin{aligned} & \mathbb{E}(d(t) - u)^2 \\ = \ & \mathbb{E}\left(\sum_{\tau=1}^{t}\frac{1}{T-\tau+1}e(\tau)F(T-\tau) - \sum_{\tau=1}^{T}\frac{1}{T}e(\tau)F(T-\tau)\right)^2 \\ & +\mathbb{E}\left(\sum_{\tau=1}^{t}\frac{1}{T-\tau+1}\left(a(\tau)-\lambda\right) - \sum_{\tau=1}^{T}\frac{1}{T}\left(a(\tau)-\lambda\right)\right)^2. \end{aligned}$$

The first term is exactly that in Lemma 5.8, and the second term is exactly that in Lemma 5.7. Thus, the expected load variance is

$$\mathbb{E}(V) = \frac{\sigma^2}{T^2}\sum_{t=0}^{T-1}F^2(t)\frac{T-t-1}{t+1} + \frac{s^2}{T}\sum_{t=2}^{T}\frac{1}{t}. \qquad \blacksquare$$

Proof of Corollary 5.9

If $|f(t)| \sim O(t^{-1/2-\alpha})$ for some $\alpha > 0$, then $|f(t)| \leq Ct^{-1/2-\alpha}$ for some $C > 0$ and all $t \geq 1$. Without loss of generality, assume that $0 < \alpha < 1/2$ and $C \geq (1-2\alpha)/(1+2\alpha)$. Then $F(0) = 1$ and

$$|F(t)| = \left|\sum_{\tau=0}^{t}f(\tau)\right| \leq 1 + \sum_{\tau=1}^{t}C\tau^{-1/2-\alpha} \leq \frac{2C}{1-2\alpha}t^{1/2-\alpha}$$

for $t = 1, \ldots, T$. The last inequality holds because $C \geq (1-2\alpha)/(1+2\alpha)$. Therefore it follows from Lemma 5.8 that

$$\begin{aligned} \mathbb{E}(V) & \leq \frac{\sigma^2}{T}\sum_{s=0}^{T-1}F^2(s)\frac{1}{s+1} \\ & \leq \frac{\sigma^2}{T} + \frac{\sigma^2}{T}\sum_{s=1}^{T-1}\frac{4C^2}{(1-2\alpha)^2}s^{1-2\alpha}\frac{1}{s+1} \\ & \leq \frac{\sigma^2}{T} + \frac{\sigma^2}{T}\frac{4C^2}{(1-2\alpha)^2}\sum_{s=1}^{T-1}\frac{1}{s^{2\alpha}} \\ & \leq \frac{\sigma^2}{T} + \frac{4\sigma^2C^2}{(1-2\alpha)^2T} + \frac{4\sigma^2C^2}{(1-2\alpha)^3T^{2\alpha}}. \end{aligned}$$

Therefore, $\mathbb{E}(V) \to 0$ as $T \to \infty$. $\blacksquare$

Proof of Lemma 5.10

The aggregate load d obtained by the optimal static algorithm is

$$\begin{aligned} d(t) &= \frac{1}{T}\left(\sum_{n=1}^{N} P_n + \sum_{\tau=1}^{T} \bar{b}(\tau)\right) - \bar{b}(t) + b(t) \\ &= \frac{1}{T}\left(\sum_{n=1}^{N} P_n + \sum_{\tau=1}^{T} \bar{b}(\tau)\right) + \sum_{\tau=1}^{T} e(\tau) f(t-\tau) \end{aligned}$$

for $t = 1, \ldots, T$. Thus,

$$\begin{aligned} &\mathbb{E}(d(t)-u)^2 \\ =\; &\mathbb{E}\left(\sum_{\tau=1}^{T} e(\tau)\left(f(t-\tau) - \frac{1}{T}F(T-\tau)\right)\right)^2 \\ =\; &\frac{\sigma^2}{T^2}\sum_{\tau=1}^{T} T^2 f^2(t-\tau) - 2Tf(t-\tau)F(T-\tau) + F^2(T-\tau) \end{aligned}$$

for $t = 1, \ldots, T$. It follows that

$$\begin{aligned} &\mathbb{E}(V') = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}(d(t)-u)^2 \\ =\; &\frac{\sigma^2}{T}\sum_{t=1}^{T}\sum_{\tau=1}^{T} f^2(t-\tau) - \frac{2\sigma^2}{T^2}\sum_{\tau=1}^{T} F(T-\tau)\sum_{t=1}^{T} f(t-\tau) \\ &+\frac{\sigma^2}{T^2}\sum_{\tau=1}^{T} F^2(T-\tau) \\ =\; &\frac{\sigma^2}{T}\sum_{t=1}^{T}\sum_{\tau=0}^{t-1} f^2(\tau) - \frac{\sigma^2}{T^2}\sum_{\tau=1}^{T} F^2(T-\tau) \\ =\; &\frac{\sigma^2}{T}\sum_{\tau=0}^{T-1}(T-\tau)f^2(\tau) - \frac{\sigma^2}{T^2}\sum_{\tau=0}^{T-1} F^2(\tau) \\ =\; &\frac{\sigma^2}{T^2}\sum_{t=0}^{T-1}\left(T(T-t)f^2(t) - F^2(t)\right). \end{aligned}$$

■

5.B Proofs of distributional results

Proof of Proposition 5.14

It has been computed in [30] that the load variance V obtained by Algorithm 5.2 is composed of two parts:

$$V = V_1 + V_2,$$

where

$$\begin{aligned} V_1 \;:=\; & \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{\tau=1}^{t}\frac{\tau-1}{T(T-\tau+1)}(a(\tau)-\lambda) \right. \\ & \left. -\sum_{\tau=t+1}^{T}\frac{1}{T}(a(\tau)-\lambda)\right]^2 \end{aligned}$$

is the variance due to the prediction error on deferrable load and

$$\begin{aligned} V_2 \;:=\; & \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{\tau=1}^{t}\frac{\tau-1}{T(T-\tau+1)}e(\tau)F(T-\tau) \right. \\ & \left. -\sum_{\tau=t+1}^{T}\frac{1}{T}e(\tau)F(T-\tau)\right]^2 \end{aligned}$$

is the variance due to the prediction error on baseload. Now we compute the worst-case V_1 and V_2 under the bounded prediction error assumption (5.15).

We start with computing the worst-case V_1. Let $x(\tau) := a(\tau) - \lambda$ for $\tau = 1, 2, \ldots, T$, then

$$
\begin{aligned}
V_1 &= \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{\tau=1}^{t}\frac{\tau-1}{T(T-\tau+1)}x(\tau)-\sum_{\tau=t+1}^{T}\frac{1}{T}x(\tau)\right]^2 \\
&= \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{\tau=1}^{t}\frac{1}{T-\tau+1}x(\tau)-\sum_{\tau=1}^{T}\frac{1}{T}x(\tau)\right]^2 \\
&= \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{\tau=1}^{t}\frac{1}{T-\tau+1}x(\tau)\right]^2+\frac{1}{T}\sum_{t=1}^{T}\left[\sum_{\tau=1}^{T}\frac{1}{T}x(\tau)\right]^2 \\
&\qquad -\frac{2}{T}\sum_{t=1}^{T}\sum_{\tau=1}^{t}\frac{1}{T-\tau+1}x(\tau)\sum_{s=1}^{T}\frac{1}{T}x(s) \\
&= \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{\tau=1}^{t}\frac{1}{T-\tau+1}x(\tau)\right]^2+\left[\sum_{\tau=1}^{T}\frac{1}{T}x(\tau)\right]^2 \\
&\qquad -\frac{2}{T^2}\sum_{s=1}^{T}x(s)\sum_{\tau=1}^{T}\sum_{t=\tau}^{T}\frac{1}{T-\tau+1}x(\tau) \\
&= \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{\tau=1}^{t}\frac{1}{T-\tau+1}x(\tau)\right]^2+\frac{1}{T^2}\left[\sum_{\tau=1}^{T}x(\tau)\right]^2 \\
&\qquad -\frac{2}{T^2}\sum_{s=1}^{T}x(s)\sum_{\tau=1}^{T}x(\tau) \\
&= \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{\tau=1}^{t}\frac{1}{T-\tau+1}x(\tau)\right]^2-\frac{1}{T^2}\left[\sum_{\tau=1}^{T}x(\tau)\right]^2 .
\end{aligned}
$$

The first term

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\left[\sum_{\tau=1}^{t}\frac{1}{T-\tau+1}x(\tau)\right]^2\\
&= \frac{1}{T}\sum_{t=1}^{T}\sum_{\tau=1}^{t}\left[\frac{1}{T-\tau+1}x(\tau)\right]^2\\
&\quad + \frac{2}{T}\sum_{t=1}^{T}\sum_{\tau=1}^{t}\frac{1}{T-\tau+1}x(\tau)\sum_{s=\tau+1}^{t}\frac{1}{T-s+1}x(s)\\
&= \frac{1}{T}\sum_{\tau=1}^{T}\sum_{t=\tau}^{T}\frac{1}{(T-\tau+1)^2}x^2(\tau)\\
&\quad + \frac{2}{T}\sum_{\tau=1}^{T}\sum_{s=\tau+1}^{T}\sum_{t=s}^{T}\frac{1}{T-\tau+1}\frac{1}{T-s+1}x(\tau)x(s)\\
&= \frac{1}{T}\sum_{\tau=1}^{T}\frac{1}{T-\tau+1}x^2(\tau)\\
&\quad + \frac{2}{T}\sum_{\tau=1}^{T}\sum_{s=\tau+1}^{T}\frac{1}{T-\tau+1}x(\tau)x(s)\\
&= \frac{1}{T}\sum_{\tau=1}^{T}\sum_{s=1}^{T}\frac{1}{T-\tau\wedge s+1}x(\tau)x(s),
\end{aligned}$$

where $a \wedge b := \min\{a, b\}$ for $a, b \in \mathbb{R}$. Let the matrix $A \in \mathbb{R}^{T\times T}$ be given by

$$A_{\tau s} := \frac{T}{T-\tau\wedge s+1}$$

for $\tau, s = 1, 2, \ldots, T$, i.e.,

$$A = \begin{bmatrix}
\frac{T}{T} & \frac{T}{T} & \frac{T}{T} & \cdots & \frac{T}{T}\\
\frac{T}{T} & \frac{T}{T-1} & \frac{T}{T-1} & \cdots & \frac{T}{T-1}\\
\frac{T}{T} & \frac{T}{T-1} & \frac{T}{T-2} & \cdots & \frac{T}{T-2}\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
\frac{T}{T} & \frac{T}{T-1} & \frac{T}{T-2} & \cdots & \frac{T}{1},
\end{bmatrix}$$

then

$$V_1 = \frac{1}{T^2}x^T\left(A - \mathbf{1}\mathbf{1}^T\right)x,$$

where the vector $x := (x(1), x(2), \dots, x(T))^T$. When prediction error is bounded as in (5.15), one has $|x(t)| \le \epsilon_1$ for all t, and therefore

$$\begin{aligned} V_1 &= \frac{1}{T^2}\sum_{\tau=1}^{T}\sum_{s=1}^{T}(A_{\tau s}-1)\,x(\tau)x(s) \\ &\le \frac{1}{T^2}\sum_{\tau=1}^{T}\sum_{s=1}^{T}\frac{\tau\wedge s-1}{T-\tau\wedge s+1}\epsilon_1^2 \end{aligned}$$

and the equality is attained if and only if $x(t) = \epsilon_1$ for all t, or $x(t) = -\epsilon_1$ for all t. Finally, we simplify the worst-case expression of V_1 as follows:

$$\begin{aligned} \sup_a V_1 &= \frac{1}{T^2}\sum_{\tau=1}^{T}\sum_{s=1}^{T}\frac{\tau\wedge s-1}{T-\tau\wedge s+1}\epsilon_1^2 \\ &= \frac{\epsilon_1^2}{T^2}\sum_{k=1}^{T}\frac{k-1}{T-k+1}(2T+1-2k) \\ &= \epsilon_1^2\left(1-\frac{1}{T}\sum_{k=1}^{T}\frac{1}{k}\right) \approx \epsilon_1^2\left(1-\frac{\ln T}{T}\right). \end{aligned}$$

We proceed to compute the worst-case V_2. Using the same derivation, it can be computed that

$$V_2 = \frac{1}{T^2}y^T\left(A-\mathbf{1}\mathbf{1}^T\right)y,$$

where

$$\begin{aligned} y &:= (y(1), y(2), \dots, y(T))^T, \\ y(t) &:= e(t)F(T-t), \quad t = 1, 2, \dots, T. \end{aligned}$$

It follows that

$$\begin{aligned} V_2 &= \frac{1}{T^2}\sum_{\tau=1}^{T}\sum_{s=1}^{T}(A_{\tau s}-1)\,y(\tau)y(s) \\ &\le \frac{1}{T^2}\sum_{\tau=1}^{T}\sum_{s=1}^{T}\frac{\tau\wedge s-1}{T-\tau\wedge s+1}\epsilon_2^2|F(T-\tau)F(T-s)| \end{aligned}$$

and that the equality is attained if and only if $e(t) = \epsilon_2\cdot\text{sgn}(F(T-t))$ for all t, or $e(t) = -\epsilon_2\cdot\text{sgn}(F(T-t))$ for all t. Finally, we simplify the worst-case expression of V_2 as follows:

$$\begin{aligned} \sup_e V_2 &= \frac{1}{T^2}\sum_{\tau=1}^{T}\sum_{s=1}^{T}\frac{\tau\wedge s-1}{T-\tau\wedge s+1}\epsilon_2^2|F(T-\tau)F(T-s)| \\ &= \frac{\epsilon_2^2}{T^2}\sum_{\tau=0}^{T-1}\sum_{s=0}^{T-1}\left(\frac{T}{\tau\vee s+1}-1\right)|F(\tau)F(s)|. \end{aligned}$$

To summarize, the worst-case load variance V obtained by Algorithm 5.2 is

$$\begin{aligned}\sup_{a,e} V &= \epsilon_1^2 \left(1 - \frac{1}{T}\sum_{k=1}^{T}\frac{1}{k}\right) \\ &\quad + \frac{\epsilon_2^2}{T^2}\sum_{\tau=0}^{T-1}\sum_{s=0}^{T-1}\left(\frac{T}{\tau \vee s + 1} - 1\right)|F(\tau)F(s)|.\end{aligned}$$

The lower bound in the lemma can be obtained from the case where all prediction errors of the load arrival is equal to $d_1/2$, then

$$\begin{aligned}\sup_a V &\geq \frac{d_1^2}{4T}\sum_{t=1}^{T}\left(\sum_{\tau=1}^{t}\frac{\tau-1}{T(T-\tau+1)} - \sum_{\tau=t+1}^{T}\frac{1}{T}\right)^2 \\ &= \frac{d_1^2}{4T^3}\sum_{t=1}^{T}\left(\sum_{\tau=1}^{t}\frac{T}{T-\tau+1} - T\right)^2 \\ &= \frac{d_1^2}{4T}\sum_{t=1}^{T}\left(\sum_{\tau=1}^{t}\frac{1}{T-\tau+1} - 1\right)^2 \\ &= \frac{d_1^2}{4T}\left(\sum_{t=1}^{T}(\sum_{\tau=T-t+1}^{T}\frac{1}{\tau})^2 - T\right) \\ &\geq \frac{d_1^2}{4T}\left(\sum_{t=1}^{T}(\int_{T-t+1}^{T}\frac{1}{u}du)^2 - T\right) \\ &= \frac{d_1^2}{4T}\left(\sum_{k=1}^{T}(\ln(\frac{T}{k}))^2 - T\right).\end{aligned}$$

Proof of Theorem 5.16

The theorem relies on a variant of the Log-Sobolev inequality provided in the following lemma.

Lemma 5.22 (Theorem 3.2, [53]). *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be convex and X be supported on $[-d/2, d/2]^n$, then*

$$\begin{aligned}&\mathbb{E}[\exp(f(X))f(X)] - \mathbb{E}[\exp(f(X))]\log\mathbb{E}[\exp(f(X))] \\ &\leq \frac{d^2}{2}\mathbb{E}[\exp(f(X))||\nabla f(X)||^2]. \qquad (5.22)\end{aligned}$$

If f is further "self-bounded", then its tail probability can be bounded as in the following lemma.

Lemma 5.23. *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be convex and X be supported on $[-d/2, d/2]^n$. If $\mathbb{E}[f(X)] = 0$ and f satisfies the following self-bounding property*

$$||\nabla f||^2 \leq af + b, \tag{5.23}$$

then the tail probability of $f(X)$ can be bound as

$$\mathbb{P}\{f(X) > t\} \leq \exp\left(\frac{-t^2}{2b + at}\right). \tag{5.24}$$

Proof. Denote the moment generating function of $f(X)$ by

$$m(\theta) := \mathbb{E}e^{\theta f(X)}, \qquad \theta > 0.$$

The function $\theta f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex, and therefore it follows from Lemma 5.22 that

$$\begin{aligned} \mathbb{E}\left[e^{\theta f}\theta f\right] - \mathbb{E}\left[e^{\theta f}\right]\ln\mathbb{E}\left[e^{\theta f}\right] &\leq \frac{d^2}{2}\mathbb{E}\left[e^{\theta f}||\theta\nabla f||^2\right], \\ \theta m'(\theta) - m(\theta)\ln m(\theta) &\leq \frac{1}{2}\theta^2 d^2\mathbb{E}[e^{\theta f}||\nabla f||^2]. \end{aligned}$$

According to the self-bounding property (5.23), one has

$$\begin{aligned} \theta m'(\theta) - m(\theta)\ln m(\theta) &\leq \frac{1}{2}\theta^2 d^2\mathbb{E}[e^{\theta f}(af + b)] \\ &= \frac{1}{2}\theta^2 d^2\left[am'(\theta) + bm(\theta)\right]. \end{aligned}$$

Divide both sides by $\theta^2 m(\theta)$ to get

$$\frac{d}{d\theta}\left[\left(\frac{1}{\theta} - \frac{ad^2}{2}\right)\ln m(\theta)\right] \leq \frac{bd^2}{2}.$$

Integrate both sides from 0 to s to get

$$\left(\frac{1}{\theta} - \frac{ad^2}{2}\right)\ln m(\theta)\Bigg|_{\theta=0}^{s} \leq \frac{1}{2}bd^2 s$$

for $s \geq 0$. Noting that $m(0) = 1$ and $m'(0) = \mathbb{E}f = 0$, one has

$$\lim_{\theta\to 0^+}\left(\frac{1}{\theta} - \frac{ad^2}{2}\right)\ln m(\theta) = 0,$$

and therefore

$$\left(\frac{1}{s} - \frac{ad^2}{2}\right)\ln m(s) \leq \frac{1}{2}bd^2 s \tag{5.25}$$

for $s \geq 0$. We can bound the tail probability $\mathbb{P}\{f > t\}$ with the control (5.25) over the moment generating function $m(s)$.

In particular, one has

$$\begin{aligned}
\mathbb{P}\{f > t\} &= \mathbb{P}\left\{e^{sf} > e^{st}\right\} \leq e^{-st}\mathbb{E}\left[e^{sf}\right] \\
&= \exp[-st + \ln m(s)] \\
&\leq \exp\left[-st + \frac{bd^2s^2}{2 - asd^2}\right]
\end{aligned}$$

for $s \geq 0$. Choose $s = t/(bd^2 + ad^2t/2)$ to get

$$\mathbb{P}\{f > t\} \leq \exp\left(\frac{-t^2}{d^2(2b + at)}\right).$$

□

Proof of Theorem 5.16. It has been computed in [30] that the load variance V obtained by Algorithm 5.2 is composed of two parts:

$$V = V_1 + V_2,$$

where

$$\begin{aligned}
V_1 &:= \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{\tau=1}^{t}\frac{\tau - 1}{T(T - \tau + 1)}(a(\tau) - \lambda)\right. \\
&\qquad \left. - \sum_{\tau=t+1}^{T}\frac{1}{T}(a(\tau) - \lambda)\right]^2
\end{aligned}$$

is the variance due to the prediction error on deferrable load and

$$\begin{aligned}
V_2 &:= \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{\tau=1}^{t}\frac{\tau - 1}{T(T - \tau + 1)}e(\tau)F(T - \tau)\right. \\
&\qquad \left. - \sum_{\tau=t+1}^{T}\frac{1}{T}e(\tau)F(T - \tau)\right]^2
\end{aligned}$$

is the variance due to the prediction error on baseload.

Let $x(\tau) := a(\tau) - \lambda$ for $\tau = 1, 2, \ldots, T$, then

$$\begin{aligned}
V_1 &= \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{\tau=1}^{t}\frac{\tau - 1}{T(T - \tau + 1)}x(\tau) - \sum_{\tau=t+1}^{T}\frac{1}{T}x(\tau)\right]^2 \\
&= \frac{1}{T}||Bx||_2^2,
\end{aligned}$$

where the $T \times T$ matrix B is given by

$$B_{t\tau} := \begin{cases} \frac{\tau-1}{T(T-\tau+1)} & \tau \leq t \\ -\frac{1}{T} & \tau > t \end{cases}, \qquad 1 \leq t, \tau \leq T.$$

Similarly, the variance V_2 due to the prediction error on baseload can be written as

$$V_2 = g(e) = \frac{1}{T}||Ce||_2^2,$$

where the $T \times T$ matrix C is given by

$$C_{t\tau} := \begin{cases} \frac{\tau-1}{T(T-\tau+1)}F(T-\tau), & \tau \leq t \\ -\frac{1}{T}F(T-\tau), & \tau > t \end{cases}$$

for $1 \leq t, \tau \leq T$. Therefore, the load variance

$$V = V_1 + V_2 = \frac{1}{T}||Ay||_2^2,$$

where

$$A = \begin{bmatrix} B & 0 \\ 0 & C \end{bmatrix}, \quad y = \begin{bmatrix} x \\ e \end{bmatrix}.$$

Define a centered random variable

$$Z := h(y) := V - \mathbb{E}V = \frac{1}{T}||Ay||^2 - \mathbb{E}V$$

and note that the function h is convex. Let $\lambda_{\max}$ be the maximum eigenvalue of AA^T/T, then

$$\begin{aligned} ||\nabla h(y)||^2 &= \frac{4}{T^2}||A^TAy||^2 = \frac{4}{T}(Ay)^T\left(\frac{AA^T}{T}\right)(Ay) \\ &\leq \frac{4\lambda_{\max}}{T}(Ay)^T(Ay) = 4\lambda_{\max}[h(y) + \mathbb{E}V]. \end{aligned}$$

According to the bounded prediction error assumption (5.15), one has $|y| \leq \epsilon$ componentwise. Then, apply Lemma 5.23 to the random variable Z to obtain

$$\mathbb{P}\{Z > t\} \leq \exp\left(-\frac{t^2}{16\lambda_{\max}\epsilon^2(2\mathbb{E}V + t)}\right)$$

for $t > 0$, i.e.,

$$\mathbb{P}\{V - \mathbb{E}V > t\} \leq \exp\left(-\frac{t^2}{16\lambda_{\max}\epsilon^2(2\mathbb{E}V + t)}\right)$$

for $t > 0$. Finally, the largest eigenvalue $\lambda_{\max}$ of AA^T/T can be bounded above as

$$\begin{aligned}\lambda_{\max} &\leq \operatorname{tr}\left(\frac{AA^T}{T}\right) = \operatorname{tr}\left(\frac{BB^T}{T}\right) + \operatorname{tr}\left(\frac{CC^T}{T}\right) \\ &= \frac{1}{T}\left(\sum_{t=2}^{T}\frac{1}{t}\right) + \frac{1}{T^2}\sum_{t=0}^{T-1}F^2(t)\frac{T-t-1}{t+1} \\ &\leq \frac{\ln T}{T} + \frac{1}{T^2}\sum_{t=0}^{T-1}F^2(t)\frac{T-t-1}{t+1} \quad =: \lambda_1,\end{aligned}$$

which completes the proof of Theorem 5.16.

□

Proof of Theorem 5.19

The derivation of the theorem is based on the following two lemma, which separates the cases when there is only one type of prediction error.

Lemma 5.24. *If there is no prediction error in the base load, then the variance of the performance of Algorithm 5.2 is bounded by*

$$\operatorname{Var}(V) \leq 4d_1^2 s^2 \left(\frac{\ln T}{T}\right)^2. \tag{5.26}$$

Lemma 5.25. *If there is no prediction error in the deferrable load, then the variance of the performance of Algorithm 5.2 is bounded by*

$$\operatorname{Var}(V) \leq 4d_2^2\sigma^2 \left(\frac{1}{T^2}\sum_{t=0}^{T-1}F^2(t)\frac{T-t+1}{t+1}\right)^2. \tag{5.27}$$

Firstly we will prove Lemma 5.24, where we only consider prediction error in deferrable load.

Proof of Lemma 5.24. Let $x(\tau) = a(\tau) - \lambda$, then $x(\tau)$ is centered, with variance s^2. Let $x = (x(1), \ldots, x(T))$. From the results in [30] Lemma 1, we have

$$V = \frac{1}{T}\sum_{t=1}^{T}\left(\sum_{\tau=1}^{t}\frac{\tau-1}{T(T-\tau+1)}x(\tau) - \sum_{\tau=t+1}^{T}\frac{1}{T}x(\tau)\right)^2.$$

Define an auxilary matrix B such that

$$B_{t\tau} = \begin{cases} \frac{\tau-1}{T(T-\tau+1)} & \tau \le t \\ -\frac{1}{T} & \tau > t. \end{cases}$$

Then we have

$$V_1 = f(x(1), x(2), \ldots, x(T)) = \frac{1}{T}||Bx||_2^2.$$

Therefore $V_1 = f(x)$ is a convex function, by convex Poincaré inequality, we have

$$\mathrm{Var}(V) \le d_1^2 \mathbb{E}[||\nabla f(x)||^2]. \tag{5.28}$$

Whereas

$$\begin{aligned} \mathbb{E}\left[||\nabla f(x)||^2\right] &= \frac{4}{T^2}\mathbb{E}\left[||B^T Bx||^2\right] \\ &\le \frac{4}{T^2}\lambda_{\max}(B^T B)\mathbb{E}\left[||Bx||^2\right] \\ &\le 4\mathrm{tr}\left(\frac{1}{T}B^T B\right)\mathbb{E}\left[\frac{1}{T}||Bx||^2\right] \\ &= 4s^2\left[\mathrm{tr}\left(\frac{1}{T}B^T B\right)\right]^2 \\ &\le 4s^2\left(\frac{\ln T}{T}\right)^2. \end{aligned}$$

The last inequality is because

$$\begin{aligned} \mathrm{tr}\left(B^T B\right) &= \frac{1}{T}\sum_{i=1}^{T}(B^T B)_{ii} \\ &= \sum_{i=1}^{T}\sum_{k=1}^{T}(B_{ki})^2 \\ &= \frac{1}{T^2}\sum_{i=1}^{T}\left(\sum_{k=1}^{i}\frac{(k-1)^2}{(T-k+1)^2} + (T-i)\right) \\ &= \frac{1}{T^2}\sum_{k=1}^{T}(\frac{(k-1)^2}{(T-k+1)} + \sum_{i=1}^{T}(T-i) \\ &= \frac{1}{T^2}\sum_{k=1}^{T}\frac{(T-k)^2}{k} + \sum_{k=1}^{T}\frac{(T-k)k}{k} \\ &= \sum_{k=2}^{T}\frac{1}{k} \le \ln T. \qquad \square \end{aligned}$$

Next we proof lemma 5.25 the case where we only consider the prediction error in the base load.

Proof of Lemma 5.25. Let $e = (e(1), \dots, e(T))$, when there is no prediction error in the deferrable load arrival, we have

$$\begin{aligned} V =& \frac{1}{T}\sum_{t=1}^{T}(\sum_{\tau=1}^{t}\frac{\tau-1}{T(T-\tau+1)}F(T-\tau)e(\tau) \\ & - \sum_{\tau=t+1}^{T}\frac{1}{T}F(T-\tau)e(\tau))^2. \end{aligned}$$

If we define an auxilary matrix C such that

$$C_{t\tau} = \begin{cases} \frac{\tau-1}{T(T-\tau+1)}F(T-\tau), & \tau \le t \\ -\frac{1}{T}F(T-\tau), & \tau > t. \end{cases}$$

Then we have

$$V = g(e(1), e(2), \dots, e(T)) = \frac{1}{T}||Ce||_2^2.$$

Therefore $V = g(e)$ is a convex function in e. By similar argument as Lemma 5.24

$$\mathrm{Var}(V) \le d_2^2\mathbb{E}[||\nabla g(e)||^2]. \tag{5.29}$$

Whereas

$$\begin{aligned} \mathbb{E}\left[||\nabla g(e)||^2\right] &= \frac{4}{T^2}\mathbb{E}\left[||C^TCe||^2\right] \\ &\le \frac{4}{T^2}\lambda_{\max}(C^TC)\mathbb{E}\left[||Ce||^2\right] \\ &\le 4\mathrm{tr}\left(\frac{1}{T}C^TC\right)\mathbb{E}\left[\frac{1}{T}||Ce||^2\right] \\ &= 4\sigma^2\left[\mathrm{tr}\left(\frac{1}{T}C^TC\right)\right]^2 \\ &= 4\sigma^2\left(\frac{1}{T^2}\sum_{t=0}^{T-1}F^2(t)\frac{T-t+1}{t+1}\right)^2. \end{aligned}$$

The last equality is because

$$
\begin{aligned}
&\operatorname{tr}(C^T C) \\
=&\sum_{i=1}^{T}\left(\sum_{k=1}^{T} C_{ki}^2\right) \\
=&\frac{1}{T^2}\sum_{i=1}^{T}\left(\sum_{k=1}^{i}\frac{(k-1)^2}{(T-k+1)^2}F^2(T-k)+\sum_{k=i+1}^{T}F^2(T-k)\right) \\
=&\frac{1}{T^2}\left(\sum_{k=2}^{T}\frac{(k-1)^2}{T-k+1}F^2(T-k)+\sum_{k=2}^{T}(k-1)F^2(T-k)\right) \\
=&\frac{1}{T}\sum_{k=2}^{T}F^2(T-k)\frac{k-1}{T-k+1}. \qquad \square
\end{aligned}
$$

Next, we bring the two results together to get a proof of Theorem 5.19.

Proof of Theorem 2. Let V_1 be the load variance without prediction error in base load and V_2 be the load variance without prediction error in the deferrable load.

$$V = V_1 + V_2.$$

By independence of x and e, the variance of V is bounded by

$$
\begin{aligned}
&\operatorname{Var}(V) = \operatorname{Var}(V_1) + \operatorname{Var}(V_2) \\
\le&\left(\frac{2d_1 s \ln T}{T}\right)^2 + \left(\frac{2d_2\sigma}{T^2}\sum_{t=0}^{T-1}F^2(t)\frac{T-t+1}{t+1}\right)^2. \qquad \square
\end{aligned}
$$

Chapter 6

CONCLUSION

Systems that constantly adapt to the state of environment or the need of customers are prevalent and have huge impact in our lives, the Internet and power systems are just two of the prominent examples. Online algorithms lie in the heart of these systems that are increasingly being automated. Thus, it is important to understand how to design good online algorithms. However, there is a divide between the pessimistic theoretical analysis about the performance of online algorithms and their success in practice, which suggests that the worst case analysis of prediction error may be missing the bigger picture. *The goal of this thesis has been to give a better understanding of online algorithm through a general modeling of practical prediction errors.*

6.1 Theoretical contributions

To this end, we have proposed a general model for prediction errors (Chapter 2). Using this model, this thesis has addressed the following theoretical challenge:

Simultaneous sublinear regret and constant competitive ratio. While previously shown that no online algorithm can achieve good performance for both regret and competitive ratio under the adversarial prediction model [58]. We show that AFHC can achieve simultaneous sublinear regret and constant competitive ratio, under the practical assumption of prediction errors (Chapter 3).

Performance bound of MPC. While MPC enjoyed success in many applications, the performance bound of MPC as a function of prediction error has been elusive. In Chapter 4, we proved the performance bound of MPC as a special case of the more general Committed Horizon Control algorithm (CHC).

6.2 Practical insights

This thesis also provides the following practical insights for the design of online algorithms:

The optimal use of prediction. Theorem 4.1 shows that the performance of online algorithm depends on the following three factors: the correlation structure of prediction errors, the size of per-step prediction noise, and the sensitivity of the online cost function. The Corollaries following Theorem 4.1 discusses optimal use of predictions under the various scenarios.

Bridging between average and worst case analysis. In Chapter 5, we showed that the there is a sharp contrast between the performance guarantee of average case and worst case analysis for the direct load control problem. Our distributional analysis offered a way to understand the typical performance of online algorithm, and showed that when prediction errors are not strongly correlated, the typical performance is close to the average performance. We expect this result to hold true more generally for online algorithms under different applications.

6.3 Open problems

There remain many challenges in the area of designing online algorithm, below we briefly outline a few of them:

1. **More general prediction models.** While the general prediction model introduced in Chapter 2 (2.6) can model any prediction error that is *stationary* in nature, it would be interesting to further generalize the prediction error model to *nonstationary* case, this can model that fact that for some prediction tasks, the accuracy of prediction not only depends on how far into future we are looking, but also when we are looking.

2. **Incorporating learning into online algorithms.** While this thesis is the first to consider the problem of designing prediction-aware online algorithms that make the best use of the given predictions, we are assuming that predictions are given externally. However, if the system has additional capability to learn from data and come up with predictions, then it gives new opportunities to jointly design learning, prediction, and decision making in online algorithms.

BIBLIOGRAPHY

[1] Salvador Acha, Tim C Green, and Nilay Shah. Effects of optimised plug-in hybrid vehicle charging strategies on electric distribution network losses. In *Transmission and Distribution Conference and Exposition, 2010 IEEE PES*, pages 1–6. IEEE, 2010.

[2] Muhammad Abdullah Adnan, Ryo Sugihara, and Rajesh K Gupta. Energy efficient geographical load balancing via dynamic deferral of workload. In *IEEE Int. Conf. Cloud Computing (CLOUD)*, pages 188–195, 2012.

[3] M. H. Albadi and E.F. El-Saadany. Demand response in electricity markets: An overview. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1–5, June 2007. doi: 10.1109/PES.2007.385728.

[4] Alberta Electric System Operator. Wind power / ail data, 2009. `http://www.aeso.ca/gridoperations/20544.html`.

[5] Alberta Electric System Operator. Alberta electric system operator. wind power and alberta internal load data. `http://www.aeso.ca/gridoperations/20544.html`, 2012.

[6] AlphaHolderWiki. Hölder condition. `{https://en.wikipedia.org/wiki/Holder_condition}`, 2016.

[7] Hrishikesh Amur, James Cipar, Varun Gupta, Gregory R Ganger, Michael A Kozuch, and Karsten Schwan. Robust and flexible power-proportional storage. In *Proc. ACM Symp. Cloud computing*, pages 217–228, 2010.

[8] Ganesh Ananthanarayanan, Ali Ghodsi, Scott Shenker, and Ion Stoica. Effective straggler mitigation: Attack of the clones. In *Proc. NSDI*, volume 13, pages 185–198, 2013.

[9] Lachlan Andrew, Siddharth Barman, Katrina Ligett, Minghong Lin, Adam Meyerson, Alan Roytman, and Adam Wierman. A tale of two metrics: Simultaneous bounds on competitiveness and regret. In *Conf. on Learning Theory (COLT)*, pages 741–763, 2013.

[10] Martin F. Arlitt and Carey L. Williamson. Web server workload characterization: The search for invariants. In *Proc. ACM SIGMETRICS*, pages 126–137, 1996. doi: 10.1145/233013.233034.

[11] Alberto Bemporad and Manfred Morari. Robust model predictive control: A survey. In *Robustness in identification and control*, pages 207–226. Springer, 1999.

[12] A. Blum, H. Karloff, Y. Rabani, and M. Saks. A decomposition theorem and bounds for randomized server problems. In *Proc. Symp. Found. Comp. Sci. (FOCS)*, pages 197–207, Oct 1992. doi: 10.1109/SFCS.1992.267772.

[13] Allan Borodin, Nathan Linial, and Michael E Saks. An optimal on-line algorithm for metrical task system. *J. ACM*, 39(4):745–763, 1992.

[14] Stéphane Boucheron, Gábor Lugosi, and Olivier Bousquet. Concentration inequalities. In *Advanced Lectures on Machine Learning*, pages 208–240. Springer, 2004.

[15] Stéphane Boucheron, Gábor Lugosi, Pacal Massart, et al. On concentration of self-bounding functions. *Electronic Journal of Probability*, 14(64):1884–1899, 2009.

[16] Stephen Boyd, Mark Mueller, Brendan O'Donoghue, and Yang Wang. Performance bounds and suboptimal policies for multi-period investment. *Foundations and Trends in Optimization*, 1(1):1–69, 2012.

[17] Robert Grover Brown, Patrick YC Hwang, et al. *Introduction to random signals and applied Kalman filtering*, volume 3. John Wiley & Sons New York, 1992.

[18] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer, 2013.

[19] Jose Camacho, Ying Zhang, Minghua Chen, and D Chiu. Balance your bids before your bits: The economics of geographic load-balancing. *Proc. of ACM e-Energy*, 2014.

[20] Emmanuel J Candès, Yaniv Plan, et al. Near-ideal model selection by $\ell 1$ minimization. *Ann. Stat.*, 37(5A):2145–2177, 2009.

[21] Venkat Chandrasekaran, Benjamin Recht, Pablo A Parrilo, and Alan S Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[22] Niangjun Chen, Anish Agarwal, Adam Wierman, Siddharth Barman, and Lachlan L. H. Andrew. Online convex optimization using predictions. In *Proc. ACM SIGMETRICS*, pages 191–204. ACM, 2015.

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

[24] Antonio J Conejo, Juan M Morales, and Luis Baringo. Real-time demand response model. *Smart Grid, IEEE Transactions on*, 1(3):236–242, 2010.

[25] Eugene A. Feinberg and Dora Genethliou. Load forecasting. In *Applied Mathematics for Restructured Electric Power Systems*, Power Electronics and Power Systems, pages 269–285. Springer US, 2005.

[26] Mário AT Figueiredo, Robert D Nowak, and Stephen J Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE J. Sel. Topics Signal Processing*, 1(4):586–597, 2007.

[27] Lingwen Gan, Ufuk Topcu, and Steven H. Low. Optimal decentralized protocol for electric vehicle charging. In *IEEE CDC*, pages 5798–5804, 2011.

[28] Lingwen Gan, Ufuk Topcu, and Steven H. Low. Stochastic distributed protocol for electric vehicle charging with discrete charging rate. In *IEEE PES General Meeting*, pages 1–8, 2012.

[29] Lingwen Gan, Ufuk Topcu, and S Low. Optimal decentralized protocol for electric vehicle charging. *IEEE Transactions on Power Systems*, 28(2):940–951, 2013.

[30] Lingwen Gan, Adam Wierman, Ufuk Topcu, Niangjun Chen, and Steven H Low. Real-time deferrable load control: handling the uncertainties of renewable generation. In *Proceedings of the fourth international conference on Future energy systems*, pages 113–124. ACM, 2013.

[31] Lingwen Gan, Adam Wierman, Ufuk Topcu, Niangjun Chen, and Steven H. Low. Real-time deferrable load control: handling the uncertainties of renewable generation, 2013. Technical report, available at http://www.its.caltech.edu/~lgan/index.html.

[32] Lingwen Gan, Adam Wierman, Ufuk Topcu, Niangjun Chen, and Steven H. Low. Real-time deferrable load control: Handling the uncertainties of renewable generation. *SIGMETRICS Perform. Eval. Rev.*, 41(3):77–79, January 2014. ISSN 0163-5999. doi: 10.1145/2567529.2567553. URL http://doi.acm.org/10.1145/2567529.2567553.

[33] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[34] Gregor Giebel, Richard Brownsword, George Kariniotakis, Michael Denhard, and Caroline Draxl. *The State-Of-The-Art in Short-Term Prediction of Wind Power*. ANEMOS.plus, 2011.

[35] Daniel Gmach, Jerry Rolia, Ludmila Cherkasova, and Alfons Kemper. Workload analysis and demand prediction of enterprise data center applications. In *Proc. IEEE Int. Symp. Workload Characterization*, IISWC '07, pages 171–180. IEEE Computer Society, 2007. ISBN 978-1-4244-1561-8. doi: 10.1109/IISWC.2007.4362193. URL http://dx.doi.org/10.1109/IISWC.2007.4362193.

[36] Daniel Gmach, Jerry Rolia, Cullen Bash, Yuan Chen, Tom Christian, Amip Shah, Ratnesh Sharma, and Zhikui Wang. Capacity planning and power management to exploit sustainable energy. In *Proc. IEEE Int. Conf. Network and Service Management (CNSM)*, pages 96–103, 2010.

[37] S. C. Graves, H. C. Meal, S. Dasu, and Y. Qiu. Two-stage production planning in a dynamic environment, 1986. `http://web.mit.edu/sgraves/www/papers/GravesMealDasuQiu.pdf`.

[38] S. C. Graves, D. B. Kletter, and W. B. Hetzel. A dynamic model for requirements planning with application to supply chain optimization. *Manufacturing & Service Operation Management*, 1(1):50–61, 1998.

[39] N. Hatziargyriou, H. Asano, R. Iravani, and C. Marnay. Microgrids. *IEEE Power and Energy Magazine*, 5(4):78–94, 2007.

[40] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[41] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[42] Marija Ilic, Jason W Black, and Jill L Watz. Potential benefits of implementing load control. In *Power Engineering Society Winter Meeting, 2002. IEEE*, volume 1, pages 177–182. IEEE, 2002.

[43] InnovationProcess. Innovation process. `https://en.wikipedia.org/wiki/Innovation_(signal_processing)`, 2016.

[44] Vinay Joseph and Gustavo de Veciana. Variability aware network utility maximization. *arXiv preprint arXiv:1111.3728*, 2011.

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

[46] Thomas Kailath, Ali H Sayed, and Babak Hassibi. *Linear estimation*. Prentice-Hall, Inc., 2000.

[47] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[48] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *J. Fluids Engineering*, 82(1):35–45, 1960.

[49] Seung-Jun Kim and Geogios B Giannakis. Real-time electricity pricing for demand response using online convex optimization. In *IEEE Innovative Smart Grid Tech. Conf. (ISGT)*, pages 1–5, 2014.

[50] Dara Kusic, Jeffrey O Kephart, James E Hanson, Nagarajan Kandasamy, and Guofei Jiang. Power and performance management of virtualized computing environments via lookahead control. *Cluster computing*, 12(1):1–15, 2009.

[51] W.H. Kwon and A.E. Pearson. A modified quadratic cost problem and feedback stabilization of a linear system. *IEEE Transactions on Automatic Control*, 22 (5):838–842, 1977.

[52] Wook Hyun Kwon and Soo Hee Han. *Receding horizon control: model predictive control for state models*. Springer Science & Business Media, 2006.

[53] Michel Ledoux. Concentration of measure and logarithmic Sobolev inequalities. In *Seminaire de probabilites XXXIII*, pages 120–216. Springer, 1999.

[54] JH a Lee and Zhenghong Yu. Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica*, 33(5):763–781, 1997.

[55] Qiao Li, Tao Cui, Rohit Negi, Franz Franchetti, and Marija D Ilic. On-line decentralized charging of plug-in electric vehicles in power systems. *arXiv preprint arXiv:1106.5063*, 2011.

[56] Minghong Lin, Zhenhua Liu, Adam Wierman, and Lachlan LH Andrew. Online algorithms for geographical load balancing. In *Int. Green Computing Conference (IGCC)*, pages 1–10. IEEE, 2012.

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

[58] Minghong Lin, A. Wierman, L.L.H. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. *IEEE/ACM Trans. Networking*, 21(5):1378–1391, Oct 2013. ISSN 1063-6692. doi: 10.1109/TNET.2012.2226216.

[59] Zhenhua Liu, Adam Wierman, Yuan Chen, Benjamin Razon, and Niangjun Chen. Data center demand response: Avoiding the coincident peak via workload shifting and local generation. *Performance Evaluation*, 70(10):770–791, 2013.

[60] Zhenhua Liu, Iris Liu, Steven Low, and Adam Wierman. Pricing data center demand response. In *Proc. ACM Sigmetrics*, 2014.

[61] Lian Lu, Jinlong Tu, Chi-Kin Chau, Minghua Chen, and Xiaojun Lin. Online energy generation scheduling for microgrids with intermittent energy sources and co-generation. In *Proc. ACM SIGMETRICS*, pages 53–66. ACM, 2013.

[62] Zhongjing Ma, Duncan Callaway, and Ian Hiskens. Decentralized charging control for large populations of plug-in electric vehicles. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 206–212. IEEE, 2010.

[63] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[64] Colin McDiarmid. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.

[65] Kevin Mets, Tom Verschueren, Wouter Haerick, Chris Develder, and Filip De Turck. Optimizing smart energy control strategies for plug-in hybrid electric vehicle charging. In *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*, pages 293–299. IEEE, 2010.

[66] Balakrishnan Narayanaswamy, Vikas K Garg, and TS Jayram. Online optimization for the smart (micro) grid. In *Proc. ACM Int. Conf. on Future Energy Systems*, page 19, 2012.

[67] OptimalPowerFlow. History of optimal power flow and formulations. `https://www.ferc.gov/industries/electric/indus-act/market-planning/opf-papers/acopf-1-history-formulation-testing.pdf`, 2012.

[68] S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

[69] Asfandyar Qureshi, Rick Weber, Hari Balakrishnan, John Guttag, and Bruce Maggs. Cutting the electric bill for internet-scale systems. *ACM SIGCOMM Computer Communication Review*, 39(4):123–134, 2009.

[70] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast Cost-volume Filtering for Visual Correspondence and Beyond. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 3017–3024, 2011.

[71] JG Roos and IE Lane. Industrial power demand response analysis for one-part real-time pricing. *Power Systems, IEEE Transactions on*, 13(1):159–164, 1998.

[72] Mark Rudelson and Roman Vershynin. Hanson-Wright inequality and sub-Gaussian concentration. *Electron. Commun. Probab.*, 18(82):1–9, 2013. ISSN 1083-589X. doi: 10.1214/ECP.v18-2865. URL `http://ecp.ejpecp.org/article/view/2865`.

[73] Andrew P Sage and James L Melsa. Estimation theory with applications to communications and control. Technical report, DTIC Document, 1971.

[74] Shankar Sastry and Marc Bodson. *Adaptive control: stability, convergence and robustness*. Courier Dover Publications, 2011.

[75] John Schulman, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, 33(9):1251–1270, 2014.

[76] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

[77] E. Sortomme, M.M. Hindi, S.D.J. MacPherson, and S.S. Venkata. Coordinated charging of plug-in hybrid electric vehicles to minimize distribution system losses. *Smart Grid, IEEE Transactions on*, 2(1):198–205, March 2011. ISSN 1949-3053. doi: 10.1109/TSG.2010.2090913.

[78] Southern California Edison. 2012 static load profiles, 2012. `http://www.sce.com/005_regul_info/eca/DOMSM12.DLP`.

[79] Southern California Edison. Southern california edison dynamic load profiles. `https://www.sce.com/wps/portal/home/regulatory/load-profiles`, 2013.

[80] Eno Thereska, Austin Donnelly, and Dushyanth Narayanan. Sierra: a power-proportional, distributed storage system. *Microsoft Research, Cambridge, UK, Tech. Rep. MSR-TR-2009-153*, 2009.

[81] Robert Tibshirani. Regression shrinkage and selection via the LASSO. *J. Royal Stat. Soc. Ser. B*, pages 267–288, 1996.

[82] Hao Wang, Jianwei Huang, Xiaojun Lin, and Hamed Mohsenian-Rad. Exploring smart grid and data center interactions for electric power load balancing. *ACM SIGMETRICS Performance Evaluation Review*, 41(3):89–94, 2014.

[83] Xiaorui Wang and Ming Chen. Cluster-level feedback power control for performance optimization. In *High Performance Computer Architecture*, pages 101–110. IEEE, 2008.

[84] Norbert Wiener. *Extrapolation, interpolation, and smoothing of stationary time series*, volume 2. MIT press, Cambridge, MA, 1949.

[85] Wikipedia. Krasovskii-lasalle principle, 2016. `http://en.wikipedia.org/wiki/Krasovskii-LaSalle_principle`.

[86] Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *J. Machine Learning Research*, 11:2543–2596, 2010.

[87] Francesco Zanini, David Atienza, Luca Benini, and Giovanni De Micheli. Multicore thermal management with model predictive control. In *Proc. IEEE. European Conf. Circuit Theory and Design (ECCTD)*, pages 711–714, 2009.

[88] Francesco Zanini, David Atienza, Giovanni De Micheli, and Stephen P Boyd. Online convex optimization-based algorithm for thermal management of MP-SoCs. In *Proc. ACM Great Lakes Symp. VLSI*, pages 203–208, 2010.

[89] Xun Yu Zhou and Duan Li. Continuous-time mean-variance portfolio selection: A stochastic LQ framework. *Applied Mathematics & Optimization*, 42 (1):19–33, 2000.

[90] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proc. Int. Conf. Machine Learning (ICML)*, pages 928–936. AAAI Press, 2003.