Online Optimization for Demand Response

by

Antoine Lesage-Landry

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy
Graduate Department of The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
University of Toronto

© Copyright 2019 by Antoine Lesage-Landry
Abstract

Online Optimization for Demand Response

Antoine Lesage-Landry
Doctor of Philosophy

Graduate Department of The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
University of Toronto
2019

Electric power systems are shifting away from conventional fuel-burning generation and moving towards renewable energy generation. Flexibility from energy storage and demand response is needed to accommodate the variability of renewable power sources. A key challenge to demand response is the uncertainty of loads. In this thesis, we design online optimization-based algorithms for demand response. Using the performance-guaranteed frameworks we develop, we can utilize flexible loads for demand response in real-time while accounting for uncertainty.

We first design online convex optimization algorithms for power setpoint tracking with flexible loads. We deal with different types of feedback from the loads: full and limited feedback and two intermediary feedback levels. This is done to account for different plausible communication scenarios. We apply our approaches to thermostatically controlled loads and charging electric vehicles. We then formulate a two-level method for the energy management of multi-energy buildings. A scheduling level and a tracking level are utilized to leverage the building’s several sources of flexibility for providing ancillary services while satisfying its different energy requirements. Next, we establish a predictive online convex optimization framework in which estimates about future rounds are used to improve the performance of online convex optimization algorithms. We apply this framework for frequency regulation and curtailment, and observe performance improvement over standard algorithms. Thereafter, we extend the multi-armed bandit framework to an instance where the number of arms to play evolves according to a wide-sense stochastic process. We use this extension to curtail flexible loads’ power consumption in response to random power imbalance due to renewables. Our approaches rely only on algebraic operations and projections onto a compact and convex sets. They thus are computationally efficient and can be scaled up to very large numbers of loads for real-time decision making.
Acknowledgements

This research and thesis were only made possible through the help and support of many individuals. I would like to especially thank:

Professor Josh Taylor, my supervisor. Thank you for welcoming me in your group, for answering my endless questions, and for offering guidance and mentorship so generously over these past years.

Dr. Iman Shames and Professor Pierluigi Mancarella for hosting me at the University of Melbourne. Special thank you to Iman for hosting me twice and for all the input you provided on both research and personal life topics.

I would also like to extend my thanks to all the colleagues that worked near me, and that I now call friends: Andrea, Richard, Alok, Mohammad, Amr, Peter, Zhongbin, Rafael, Mireille and Grant. You have very much contributed to making this experience a great one.

I finally thank the Natural Sciences and Engineering Research Council (NSERC), the Fonds de recherche du Québec – Nature et technologies (FRQNT), and The Edward S. Rogers Sr. Department of Electrical & Computer Engineering from the University of Toronto for supporting me financially throughout my graduate studies.

Voici le moment de changer de langue et de retourner pour un bref moment à mon français un peu rouillé. J’aimerais remercier de tout mon cœur mes parents France et Serge, mon frère Guillaume et ma sœur Léa, pour leurs encouragements soutenus tout au long de mes études. C’est sans nul doute grâce à mes parents qui ont permis aux études de jouer un rôle central dans mon développement que j’ai entrepris et mené avec succès mon doctorat.

Je tiens aussi à remercier tous mes complices qui ont su m’accompagner malgré la distance pendant ces dernières années. Merci Vincent, Jean-Philippe, Tommy, Julien et Antoine pour votre présence à chaque étape et votre amitié qui ont su me motiver à persévérer. Un merci tout spécial à mes compagnons de voyage, Olivier et Nazim, qui m’ont généreusement épaulé et ont fait de ce périple une belle aventure.

Finalement, je voudrais conclure cette section en remerciant Audrey du plus profond de mon cœur pour sa patience, sa confiance en moi et son incommensurable soutien durant ces dernières années. Merci ô Audrey.
# Contents

1 Introduction 1
   1.1 Demand response 1
      1.1.1 Indirect and direct load control 2
      1.1.2 Uncertainty 2
   1.2 Online learning 3
      1.2.1 Online convex optimization 3
      1.2.2 Multi-armed bandit 4
   1.3 Thesis objective & overview 4
   1.4 Relation between chapters 5

2 Setpoint Tracking with Partially Observed Loads 7
   2.1 Introduction 7
      2.1.1 Related work 7
      2.1.2 Contributions 8
   2.2 Background 9
      2.2.1 Online convex optimization 9
      2.2.2 Composite objective mirror descent 9
   2.3 Setpoint tracking with online convex optimization 10
   2.4 Limited feedback 12
      2.4.1 Bandit feedback 12
      2.4.2 Partial bandit feedback 14
      2.4.3 Bernoulli feedback setting 15
   2.5 Examples 17
      2.5.1 Setpoint tracking using TCLs 17
      2.5.2 Setpoint tracking using EVs storage unit 21
   2.6 Conclusion 23
   2.7 Proofs 24
      2.7.1 Proof of Lemma 2.1 24
      2.7.2 Proof of Proposition 2.1 25
      2.7.3 Proof of Lemma 2.2 25
      2.7.4 Proof of Theorem 2.1 26
      2.7.5 Proof of Theorem 2.2 27
      2.7.6 Proof of Theorem 2.3 28
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Total setpoint tracking improvement $T$ comparison (averaged over 100 simulations)</td>
<td>19</td>
</tr>
<tr>
<td>2.2</td>
<td>Per round average regularizer improvement (averaged over 100 simulations)</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>Breakdown of $x_t$</td>
<td>33</td>
</tr>
<tr>
<td>3.2</td>
<td>Breakdown of $s_t$</td>
<td>33</td>
</tr>
<tr>
<td>3.3</td>
<td>Building parameters</td>
<td>45</td>
</tr>
<tr>
<td>3.4</td>
<td>Ratio of tracking rounds in which time-varyings constraint are satisfied over the total number of tracking rounds</td>
<td>52</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparison between OCO, POCO and OMD (averaged over 1000 trials)</td>
<td>74</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Cumulative setpoint tracking loss comparison through different feedback type</td>
<td>20</td>
</tr>
<tr>
<td>2.2</td>
<td>Setpoint tracking examples under different type of feedback</td>
<td>21</td>
</tr>
<tr>
<td>2.3</td>
<td>Regularization performance of the CGGD for setpoint tracking in the full information setting</td>
<td>22</td>
</tr>
<tr>
<td>2.4</td>
<td>Temperature inside a subset of TCLs in the full information setting (left) and without regularization (right)</td>
<td>22</td>
</tr>
<tr>
<td>2.5</td>
<td>Charging $\mu_c$ and discharging $\mu_d$ signal comparison with regularization (top) and without regularization (bottom)</td>
<td>24</td>
</tr>
<tr>
<td>3.1</td>
<td>Scheduling rounds &amp; decisions (top) and tracking rounds &amp; decisions (bottom)</td>
<td>32</td>
</tr>
<tr>
<td>3.2</td>
<td>Scheduled energy resources distribution for a day</td>
<td>46</td>
</tr>
<tr>
<td>3.3</td>
<td>Scheduled ancillary services</td>
<td>47</td>
</tr>
<tr>
<td>3.4</td>
<td>Real-time energy dispatch for a whole day</td>
<td>48</td>
</tr>
<tr>
<td>3.5</td>
<td>Real-time ancillary services dispatch</td>
<td>49</td>
</tr>
<tr>
<td>3.6</td>
<td>State of the building’s different energy storage</td>
<td>50</td>
</tr>
<tr>
<td>3.7</td>
<td>Energy usage and constraints between scheduling round 11 and 14</td>
<td>51</td>
</tr>
<tr>
<td>3.8</td>
<td>Building’s net revenue for different day</td>
<td>52</td>
</tr>
<tr>
<td>3.9</td>
<td>Real-time energy resources usage using MPC as tracking algorithm</td>
<td>53</td>
</tr>
<tr>
<td>4.1</td>
<td>State of charge objective</td>
<td>71</td>
</tr>
<tr>
<td>4.2</td>
<td>Experimental comparison between the POCO with fixed step size and OCO</td>
<td>71</td>
</tr>
<tr>
<td>4.3</td>
<td>Experimental comparison between the POCO with backtracking and OCO</td>
<td>73</td>
</tr>
<tr>
<td>4.4</td>
<td>Experimental regret comparison (averaged over 1000 trials)</td>
<td>74</td>
</tr>
<tr>
<td>5.1</td>
<td>Regret when $K^t \sim \text{Poisson}_{\lambda}(\lambda)$ i.i.d..</td>
<td>83</td>
</tr>
<tr>
<td>5.2</td>
<td>Markov Chain for $K$ loads to deploys in regulation model</td>
<td>83</td>
</tr>
<tr>
<td>5.3</td>
<td>Regret when $K^t$ is a Markov chain</td>
<td>84</td>
</tr>
<tr>
<td>5.4</td>
<td>Regret when $K^t$ is a function of the moving sample mean</td>
<td>85</td>
</tr>
</tbody>
</table>
## List of Algorithms

<table>
<thead>
<tr>
<th>Section</th>
<th>Algorithm Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>COGD for setpoint tracking algorithm</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>BCOGD for setpoint tracking with limited feedback algorithm</td>
<td>13</td>
</tr>
<tr>
<td>2.3</td>
<td>PBCOGB for setpoint tracking with partial feedback algorithm</td>
<td>15</td>
</tr>
<tr>
<td>2.4</td>
<td>BerCOGD for setpoint tracking with Bernoulli feedback algorithm</td>
<td>17</td>
</tr>
<tr>
<td>3.1</td>
<td>Two-level MES building real-time optimization algorithm</td>
<td>44</td>
</tr>
<tr>
<td>4.1</td>
<td>Backtracking algorithm for predictive gradient projection</td>
<td>64</td>
</tr>
<tr>
<td>5.1</td>
<td>Upper Confidence Bound with Stochastic Plays (UCB-SP) algorithm</td>
<td>80</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Renewable generators like solar photovoltaics and wind turbines rely on natural phenomena to produce electric power. The resources they exploit are virtually infinite, but also intermittent and random. This intermittent character, from sudden to seasonal changes in renewable generation, puts additional stress on the power systems [1–5]. Consequently, sustained balancing between the power generation and the network’s demand is required to overcome the intermittency and to ensure the stability of the grid [3–4].

Increases in energy storage capacity [4, 6] and the development of a more flexible grid [3, 7–9] have been put forward as effective ways to manage the intermittent behavior of renewables. In this thesis, we focus on the latter. Grid flexibility comes on the one hand from the generation side, where sets of generators can respond to variation in demand and renewable generation [3]. On the other hand, the demand, and thus the loads’ consumption can be adjusted momentarily to cope with the sudden change in generation [10,11]. The latter strategy is referred to as demand-side management or demand response. In our work, we work with the demand side of the electric network and we design novel online learning algorithms to provide demand response in power systems.

In the following sections, we introduce demand response in detail and discuss the main challenge it faces. Next, we introduce the mathematical framework on which our approaches are based. Lastly, we provide an overview of the thesis and highlight the connections between the different chapters.

1.1 Demand response

Demand response (DR) refers to modifying the normal electric power consumption of flexible loads to improve the efficiency, stability, and resiliency of the power grid [10,14]. This is achieved by controlling and coordinating the flexible loads’ power consumption. Examples of demand response applications are:

- *peak-shaving* to reduce the maximum power demand and thus reduce the required generation capacity;

- *load-shifting*, i.e. moving the power consumption to earlier and/or later time, e.g. for valley-filling, to improve the efficiency of the system and;

- *ancillary services* like frequency regulation made by balancing demand and generation on a short-time scale to ensure the stability of the grid [15].

In each chapter, a detailed discussion of the relevant literature is presented.
Chapter 1. Introduction

1.1.1 Indirect and direct load control

Demand response control approaches are divided in two types: indirect and direct load control \cite{10, 11}. In the former case, loads are encouraged to participate in DR in response to a signal, for example, a change in the electricity rate. The load decides whether to modify its power consumption. An example of this type of demand response is real-time pricing where the electricity price changes according to the operator’s needs, e.g., high prices when generation is low and demand is high \cite{15, 16}. Time-of-use pricing is a similar strategy where a few different rates are in place \cite{15, 16}. In Ontario, for example, three rates are employed \cite{17}. This is done to encourage loads to shift their power consumption to nighttime when the demand is at its lowest. The second type is referred to as direct load control, wherein the entity in charge of providing demand response, the load aggregator, directly modifies the consumption of the loads \cite{11, 15}. For example, the load aggregator can curtail non-essential power consumption by shutting off air conditioning when a load is already at an acceptable temperature or momentarily interrupting charging an electric vehicle. In this case, the load receives compensation for their participation, for example, discounted electricity rates or a financial reward \cite{15}. In this thesis, we focus on direct load control demand response because it offers faster and more reliable outcomes.

1.1.2 Uncertainty

Uncertainty is a core challenge in demand response \cite{14, 18}. This uncertainty comes from:

- limitations in the current load models;
- the limited amount of measurements or feedback from the loads available to the aggregator and;
- exogenous factors such as weather and human behavior.

Even when precise models are known, the large number of loads makes it impractical to have accurate parameters for every single load in an aggregation. This means that flexible loads participating in DR should most of the time be considered uncertain.

The limited measurements or feedback from the loads available to the aggregator also play a substantial role \cite{14}. For example, only substation power measurements might be available instead of individual load power measurements, in which case individual responses to DR must be inferred. Moreover, measurements are in many cases power readings. Information about the state of the loads, e.g., the temperature for a thermostatically controlled load, are not often available to the load aggregator without more refined metering.

This last issue combined with unknown load parameters or model means that a load aggregator is faced with the trade-off between exploration and exploitation \cite{19, 20}. New information can only be acquired about a load when it is dispatched for DR. Then, if a load performs well according to the load aggregator’s metrics, for example, curtailing a large amount of its power consumption, these loads might be re-used on several occasions (exploitation) to meet the aggregator’s objective. If a load is not dispatched, then no information about this load can be collected. The aggregator must thus balance between learning enough about the loads to identify the best candidates (exploration) and deploy them to maximize performance.
1.2 Online learning

We now introduce two types of online learning algorithms: online convex optimization, and the multi-armed bandit. Detailed literature surveys on each framework are given in the subsequent chapters.

Online learning is a subfield of machine learning in which training and prediction are done concurrently. In online learning, a forecaster (player or decision maker) must sequentially make predictions or take actions to minimize their losses \[21, 22\]. These losses are unknown when the prediction is made. When the round ends, the loss function is revealed to the forecaster. This information is then used to improve future predictions. The performance metric or algorithm design tool is the regret. The regret is defined as the cumulative difference between the loss suffered by the forecaster and the loss associated with a sequence of comparators, for example, the best-fixed decision in hindsight for the static regret \[20, 21, 23, 24\] or the round optima for the dynamic regret \[23, 25, 26\]. The regret definition is formally stated in Chapters \[2, 3\] for each of the online algorithms we use. An online learning algorithm with a regret that grows sublinearly with the number of rounds is referred as a low-regret algorithm and is qualified as performing well \[24\]. If the regret grows sublinearly, it implies that the average regret goes to zero as the time horizon increases. Consequently, the algorithm will play, on average, at least as well as the comparator used in the regret definition \[20, 21, 24, 27, 28\]. In other words, it will play, on average, at least as well as the best-fixed action in hindsight in the case of the static regret or as well as the round optima for the dynamic regret. This hence gives a guarantee on the performance of the online learning algorithm.

1.2.1 Online convex optimization

In online convex optimization (OCO) \[23, 24, 27\], we consider a convex and compact decision set \( \mathcal{X} \subset \mathbb{R}^N \), \( N \in \mathbb{N} \), and a convex loss function \( f_t : \mathcal{X} \mapsto \mathbb{R} \) for all round \( t = 1, 2, \ldots, T \). Other technical assumptions are also added and will be discussed in later chapters. Let \( x_t \in \mathcal{X} \) be the prediction made at round \( t \). At this time, the forecaster wishes to sequentially solve the following problem:

\[
\min_{x_t \in \mathcal{X}} f_t(x_t)
\]

where \( \mathcal{X} \) is known but \( f_t \) is only revealed after the round. The loss function \( f_t \) is assumed to have bounded variation, i.e., limited change from round to round. This makes information gathered in prior rounds useful in future rounds. In OCO, the prediction is computed using an update rule which is a single step of a standard convex optimization algorithm, e.g., a gradient descent step \[23\], a mirrored descent step \[29, 30\] or a Newton step \[31\]. This update rule uses only algebraic operations and a projection onto a compact, convex set and is thus very efficiently computable.

The computational complexity of the update is a function of the projection step. If an analytical solution exists for the projection, e.g. for the examples given in \[32\] Section 12.3.3], then the computational complexity of the update is also polynomial. In this case, a precise \( O(\cdot) \) bound can be obtained depending on the solution’s closed form. If a projection algorithm is required, the complexity does not suffer badly as efficient projection algorithms with polynomial complexity can also be used. For example, \[33, 34\] offers \( O(n^2) \), \( O(n \log n) \) and \( O(n) \) algorithms for boxed-constrained Euclidean norm minimization problem. Therefore, the OCO update has worst-case polynomial complexity, and can usually be scaled to large problems.
Online convex optimization-based approaches allow a load aggregator to dispatch demand response in real-time. Real-time decision making is possible because OCO can operate, due to the low computational complexity of its update, when the rounds’ duration is significantly reduced. Hence, by reducing the duration to the seconds level, for example, it is possible to adjust a load’s power consumption following an event (e.g., system power imbalance or load’s performance change) almost instantaneously. A quantitative study is presented Chapter 3 where timing results for OCO and model predictive control are presented and compared.

There are no assumptions on the distribution of the loss functions, i.e., OCO can handle stochastic, deterministic or adversarial sequences [21]. This makes OCO promising for managing uncertainty in DR. We use OCO for demand response in Chapters 2, 3 and 4.

1.2.2 Multi-armed bandit

The multi-armed bandit (MAB) framework [19,20] is similar to OCO except that the actions are discrete and are represented by arms, i.e., the arms of a slot machine, an one-armed bandit. When a round ends, the forecaster observes only the reward associated with the arms that were played. The MAB framework is therefore used in settings with a discrete set of decisions, e.g., which loads to curtail. In MAB, it is common to use the term gain instead of loss [20]. The process is then similar to OCO: the decision maker must sequentially select an arm or a subset of arms to maximize its cumulative gain. In our work, we more specifically investigate settings in which multiple arms must be played in each round. Three main families of MAB problems are well studied in the literature. They all differ in the arms’ reward process. First, in the stochastic MAB, the reward of an arm is modeled as independent and identically distributed (i.i.d.) random variables [35,36]. Second, in the adversarial MAB, the reward of the arms is fixed by an adversary that can be non-oblivious [37]. Finally, in the Markovian bandit, the arm’s reward is determined by a Markov process [38,39].

The multi-armed bandit problem is a canonical case of the exploration vs. exploitation trade-off as one is searching for better arms than the best current known, or continues to exploit that arm. MAB problems often admit index policies for choosing which arms to play in each round. Index policies are desirable because indices can be computed for each arm individually rather than a high-dimensional problem over all arms. In Chapter 5 we extend a standard MAB algorithm, the upper confidence bound−1 (UCB1) [36], to the case of stochastically varying number of arms to be played in each round. We also extend the new algorithm to curtailment via DR.

1.3 Thesis objective & overview

The objective of this thesis is to design demand response algorithms that accommodate uncertainty, detailed modeling, and are computationally efficient. We use the OCO and the MAB frameworks. We propose performance-guaranteed algorithms under mild modeling assumptions that achieve this objective. Our online optimization methodology allows us to learn from past information and adapt to the current situation while dispatching loads in real time. We now present a short overview of the contributions of this thesis.

We first use online convex optimization to track an aggregate power setpoint with uncertain, flexible loads in Chapter 2. We formulate a full feedback model and then generalize an existing limited feedback extension for our application. We propose two new intermediary types of feedback: the partial
bandit and Bernoulli feedback. Both reduce the need for communication. We apply our online approaches to thermostatically controlled loads and electric vehicles in demand response. This chapter was previously presented in part at *X Bulk Power Systems Dynamics and Control Symposium, IREP’2017 Symposium* [40] and then published in *IEEE Transactions on Power Systems* [41].

In Chapter 3, we model multi-energy buildings providing ancillary services to the grid. These buildings can utilize several sources of flexibility while having to meet multiple energy requirements. We formulate a two-level algorithm for the optimal energy management under uncertainty of multi-energy buildings. First, a planning level sets hourly objectives. Then, an online convex optimization-based level tracks the planned objectives under uncertainty. We present the performance of our approach in a case study where we consider a building located in Melbourne, Australia. This work is presently under review in *IEEE Transactions on Control Systems Technology* [42].

For the next two chapters, we focus on the algorithm theory and, motivated by demand response applications, we design new online learning algorithms. We extend the online convex optimization framework to integrate future information in the form of inexact loss function gradients in Chapter 4. We show that under certain conditions, our framework strictly improves the performance over standard online convex optimization algorithms. We conclude this chapter by presenting two demand response examples of our predictive framework. This work is currently under review in *Automatica* [43].

In Chapter 5, we propose a new extension to the stochastic multi-armed bandit framework in which the number of arms to play at each round is distributed according to a wide-sense stochastic process. This allows us to model the dispatch of a varying number of unknown, uncertain loads for curtailment to mitigate random power imbalance in power systems. This chapter was published in *IEEE Transactions on Automatic Control* [44].

We conclude in Chapter 6 and discuss future directions for the work done in this thesis.

### 1.4 Relation between chapters

In each chapter, demand response of flexible loads subject to uncertainty is considered under different sets of modeling assumptions. These modeling assumptions represent scenarios in which demand response could be used. In each case, we propose an online learning algorithm specifically tailored for this set of assumptions.

In Chapter 2, the focus is on the different feedback types the load aggregator has access to. This translates to the different available communication configurations for aggregations of loads: so-called smart metering allows for full feedback, substation reading allows only for limited feedback and privacy considerations might force the load aggregator to opt for partial bandit, Bernoulli or limited feedback. Finally, these three last types of feedback can also be used to decrease communication needs.

In Chapter 3, we focus on load modeling for demand response. This ultimately requires a new OCO extension. We model the different energy systems that can provide flexibility for a building: heating/cooling systems, battery energy storage, water heaters, and solar photovoltaics. To meet the requirements like the temperature, the electric baseload or the domestic hot water, which are uncertain and time-varying, while providing ancillary services, we design an OCO with time-varying constraints algorithm. This new algorithm can then take into account the time-varying aspects of the multi-energy building.

In Chapters 4 and 5, we refocus on algorithms. In Chapter 4, we develop a new, predictive OCO
algorithm. The algorithms use prior observations, as in conventional OCO, and predictions of the future via an inexact gradient of the next round’s unknown loss function. In the context of demand response, an estimated gradient may be available using weather or baseload predictions, and thus this extension can be used to improve the DR performance, if available.

In Chapter 5, we use the multi-armed bandit framework to choose which subset of loads to curtail in each time period. Existing MAB frameworks do not allow a time-varying number of arms to be played in each round. We extend the MAB with multiple plays to stochastic number of arms played in each round and obtain a sublinear bound on the regret.
Chapter 2

Setpoint Tracking with Partially Observed Loads

2.1 Introduction

Demand response (DR) is an important source of flexibility for the electric power grid [5, 13, 15]. In this work, we use online convex optimization (OCO) to design algorithms for tracking setpoints with uncertain flexible loads in DR programs. Setpoint tracking has several potential applications in power systems such as frequency regulation and load following [45, 46]. The services are essential to absorbing renewable intermittency and ensuring the stability of the power grid [45, 47].

Uncertainty is a key challenge in DR [14]. This uncertainty arises from weather, human behavior and unknown load models. In addition, loads are time-varying, e.g., a heater may not have much flexibility during cold nights. The uncertain, time-varying nature of loads means that load aggregators must deploy loads for DR to observe their capabilities.

We base our setpoint tracking model on OCO [21, 24]. This allows us to dispatch loads for DR without precise knowledge of their responses by relying only on information from previous rounds. We invoke theoretical bounds to guarantee the performance of the approach. In this work, we do not propose a general setpoint tracking method. We rather propose a specialized algorithm for setpoint tracking with flexible loads.

We focus on controllable loads that can both increase and decrease their power consumption when instructed. In each round, the load aggregator sends adjustment signals to loads so that the aggregated power consumption tracks a setpoint. We use a sparsity regularizer and a mean regularizer to reduce the number of dispatched loads and the impact on the loads, respectively. To minimize the setpoint tracking objective and regularizers, Composite Objective MIrror Descent (COMID) [30], an OCO algorithm, is used to compute the adjustment signal in each round.

2.1.1 Related work

Online learning [20, 21] and online convex optimization [23, 24, 27] have already seen a number of applications in DR. Several variants of the multi-armed bandit framework have been used to curtail flexible loads in [44, 48, 51]. Reference [52] used adversarial bandits to shift load while learning load parameters.
Online learning has also been used in models of price-based DR. Reference [53] used a continuum-armed bandit to do real-time pricing of price responsive dynamic loads. Reference [54] used OCO and conditional random fields to predict the price sensitivity of Electric Vehicles (EVs). This model was then used as input to compute real-time prices. Using OCO, [28] developed real-time pricing algorithms to flatten the aggregate load consumption. They later applied their algorithm to EVs charging. OCO was also used in [55] to flatten the aggregated power consumption using EVs charging scheduling. Ledva et al. [56] used OCO to identify the controllable portion of demand in real-time from aggregate measurements.

Setpoint tracking using direct load control was previously studied in [46, 57, 58] under different assumptions. These works are based on model predictive control, which requires precise load modeling and observations or estimates of the current state in each time step. In our case, the load aggregator does not need to precisely model the loads or to obtain their current states. Only the mean regularizer, which is optional, needs modeling to balance the impact of positive and negative power adjustments. References [59–61] used Lyapunov optimization to derive real-time tracking algorithms using flexible loads and storages under uncertainty. Their setting, however, differs from ours as the information at round \( t \) is first observed, and then the real-time decisions are computed and dispatched. In the online learning setting, we do not observe the system’s parameters until the decisions are sent and the round is over. Moreover, similarly to the model predictive control-based approach, [59–61] require accurate load parameters. References [60, 61] do, however, offer a distributed version of their algorithm. In our problem formulation, the setpoint may also be unknown when decisions are computed, which is not the case in [46, 57–61]. To the best of our knowledge, no existing approaches accommodate our minimal level of assumptions or the partial feedback settings.

2.1.2 Contributions

Reference [28] is the most closely related to ours. In [28], COMID is used to set prices with the objective of load flattening. We differ in that we use direct load control [11] and our goal is to track a setpoint. Reference [28] also provides a bandit extension to COMID. We first generalize their bandit formulation for setpoint tracking. In this work, we also give two novel limited feedback extensions to COMID. The limited feedback extensions allows us to apply our approach when communication capacity is limited. This can reduce the need to invest in new communication infrastructure when it is not already available. Our main contribution is that we address uncertainty in direct load control demand response in an original and efficient way using online convex optimization. Our specific contributions are:

- We formulate an OCO-based setpoint tracking model for flexible loads (Section 2.3);
- We introduce a mean regularizer to minimize the impact of DR on loads;
- We generalize the bandit-COMID algorithm, obtain a regret bound for the algorithm and apply it to setpoint tracking (Section 2.4.1);
- We introduce a partial bandit feedback extension in which only a subset of loads are observed, and provide a sublinear regret bound. (Section 2.4.2);
- We introduce a Bernoulli feedback extension in which the aggregator receives full feedback in some rounds and bandit feedback in others, and prove that the algorithm achieves sublinear regret (Section 2.4.3);
• We numerically demonstrate the performance of our algorithms for setpoint tracking with Thermostatatically Controlled Loads (TCLs) and Electric Vehicles (EVs) (Section 2.5).

2.2 Background

In this section, we introduce OCO and the COMID algorithm.

2.2.1 Online convex optimization

In OCO, a player chooses a decision from a convex set and suffers a loss, which is unknown at the time of the decision [24]. At the end of each round, the player observes that round’s loss. We index the rounds by $t$, denote the time horizon $T$ and the decision variable $\mu_t \in K$ for all $t$ where $K \subseteq \mathbb{R}^N$ is the convex and compact decision set. The performance of an online learning algorithm is characterized by its regret, defined as

$$\text{Regret}_T = \sup_{\{F_1, F_2, \ldots, F_T\} \subset \mathcal{L}} \left\{ \sum_{t=1}^{T} F_t(\mu_t) - \min_{\mu \in K} \sum_{t=1}^{T} F_t(\mu) \right\},$$

where $F_t$ is the loss function at time $t$ and $\mathcal{L}$ is the set of loss functions. The regret compares the cumulative loss suffered in each round by the player to the cumulative loss of the best fixed decision in hindsight. An algorithm that achieves a regret that is sublinear in the number of rounds eventually performs at least as well as the best fixed decision in hindsight [28]. Note that this definition of regret is based on a fixed decision, while our OCO algorithms produce continually changing decisions for the time-varying examples. As a result, the algorithms we develop often significantly outperform the theoretical regret bound. Therefore, the regret bounds are better viewed as design tools rather than performance metrics.

2.2.2 Composite objective mirror descent

Online Gradient Descent (OGD) was first proposed in [23] and then generalized to Online Mirror Descent (OMD) [29]. In this work, we use the Composite Objective Mirror Descent (COMID) [30]. This generalization of OMD handles loss functions of the form $F_t(\mu_t) = f_t(\mu_t) + r(\mu_t)$, where $f_t$ is a round-dependent loss function and $r$ is a round-independent regularizer.

Define $\mathcal{R} : K \rightarrow \mathbb{R}$ as a regularization function (cf. references [21, 24] for more detail about regularization functions). We only consider $\alpha$-strongly convex regularization functions $\mathcal{R}$. Let $x, z \in K$ be two arbitrary vectors. The Bregman divergence with respect to $\mathcal{R}$ of $x$ and $z$ is defined as

$$B_{\mathcal{R}}(x, z) = \mathcal{R}(x) + \mathcal{R}(z) - \nabla \mathcal{R}(z)^\top (x - z).$$

Also define the dual norm of $z$ as

$$\|z\|_* = \sup \left\{ z^\top x \mid \|x\| \leq 1 \right\}.$$

Note that the dual of the $\ell_2$-norm is the $\ell_2$-norm.
The COMID update is given by
\[
\mu_{t+1} = \arg \min_{\mu \in \mathcal{K}} \eta \nabla f_t(\mu_t)^T \mu + B_{\mathcal{R}}(\mu, \mu_t) + \eta r(\mu),
\]
(2.1)

where \(\eta\) is a numerical parameter. We now present a specialized version of the original COMID regret bound of [30]. In this version, \(\eta\) is tuned to avoid too small step sizes when applied to setpoint tracking.

**Lemma 2.1** (Regret bound for COMID). Let \(f_t\) be a \(L\)-Lipschitz function and \(\|\nabla f_t(\mu_t)\|_* \leq G_*\) for all \(t\), \(r(\mu_1) = 0\) and define the tuning parameter \(\chi \geq 1\). Then, using
\[
\eta = \chi \sqrt{\frac{2 \alpha B_{\mathcal{R}}(\mu^*, \mu_1)}{G_*^2 T}},
\]
(2.2)

the regret of COMID is bounded above by,
\[
\text{Regret}_T(\text{COMID}) \leq \sqrt{\frac{2TB_{\mathcal{R}}(\mu^*, \mu_1)G_*^2 \chi^2}{\alpha}}.
\]
(2.3)

The proof of Lemma 2.1 is given in Section 2.7.1 and relies on [30, Corollary 4].

In this work, we set \(\mathcal{R}(\cdot) = \frac{1}{2} \| \cdot \|_2^2\), a 1-strongly convex function. The Bregman divergence \(B_{\mathcal{R}}\) then simplifies to \(B_{\mathcal{R}}(x, z) = \frac{1}{2} \| x - z \|_2^2\). Under this regularization function, the OMD algorithm simplifies to OGD. Accordingly, we refer to COMID with \(\mathcal{R}(\cdot) = \frac{1}{2} \| \cdot \|_2^2\) as the Composite Objective Gradient Descent (COGD). For the rest of this work, we will be using the COGD algorithm. Finally, we let \(D\) be the diameter of the compact set \(\mathcal{K}\),
\[
D = \text{diam } \mathcal{K} = \sup \{ \| x - z \|_2 \mid x, z \in \mathcal{K} \},
\]
and we note that \(B_{\mathcal{R}}(x, z) \leq \frac{1}{4} D^2\) for \(x, z \in \mathcal{K}\).

### 2.3 Setpoint tracking with online convex optimization

We consider a load aggregator providing ancillary services to the power grid. The task is to convert a regulation signal, e.g., from a portion of the system operator’s Area Control Error (ACE) [62], into commands to the individual loads. The objective of the load aggregator is to send adjustment signals to the loads so that the total adjustment tracks a power setpoint, \(s_t \in \mathbb{R}\). We consider \(N\) flexible loads. We let the decision variable \(\mu_t \in [-1, 1]^N\) denote the vector of adjustment signals sent to the loads at time \(t\). \(\mu_t\) represents instructions to scale down \((\mu < 0)\) or scale up \((\mu > 0)\) the load power consumption to match the setpoint. For example, in the case of Thermostatically Controlled Loads (TCLs), scaling down its power consumption represents lowering the air conditioning intensity or simply shutting it down, and scaling up represents increasing intensity to stock thermal energy. This signal is scaled by the response of the loads, \(c_t\), and the sum over all the loads gives the aggregate power adjustment, \(c_t^T \mu_t\), at time \(t\).

Depending on the feedback structure, the aggregate adjustment and part of \(c_t\) are revealed immediately after the decision in the current round. The uncertainty in \(c_t\) represents, for instance, erratic consumer behavior or the loads’ dependence on weather.
We define the setpoint tracking loss as the square of the tracking error:

$$\ell_t(\mu_t) = (s_t - c_i^T \mu_t)^2.$$  

(2.4)

This loss function is chosen to penalize large deviations from the setpoint. Note that $s_t$ can be known or unknown when $\mu_t$ is chosen.

We also use a mean-regularizer to penalize deviations of the mean adjustment over time from zero. The purpose of this is to minimize the impact of demand response on the loads, e.g., to reduce rebound effects \[11\]. Let $\langle \cdot \rangle_t$ denote the mean of its argument over rounds 1 to $t$ and let the mean of a vector be the vector of the means of its elements. We define the mean-regularizer as

$$\|\langle \mu \rangle_t\|_2^2 = 1_t \sum_{s=1}^{t} \mu_s \|s_t - c^T \mu_t\|_2^2 + \rho \|\langle \mu \rangle_t\|_2^2,$$

The mean regularizer can also use a weighted average to balance the adjustment if the loads’ responses are asymmetrical.

We use an $\ell_1$-norm sparsity regularizer to minimize the number of loads dispatched in each round and to avoid dispatching loads with small contributions.

The total objective function is therefore given by

$$F_t(\mu_t) = (s_t - c_i^T \mu_t)^2 + \rho \|\langle \mu \rangle_t\|_2^2 + \lambda \|\mu_t\|_1,$$

(2.5)

where $\rho$ and $\lambda$ are numerical parameters.

We now apply COGD to setpoint tracking. This allows us to compute adjustment signals to be sent to the load while the current round information is unavailable. In this case, the unavailable information are the loads responses $c_t$ and signal $s_t$. These predictions are based on past round information. We define the loss function and regularizer to be

$$f_t(\mu_t) = (s_t - c_i^T \mu_t)^2 + \rho \|\langle \mu \rangle_t\|_2^2,$$

(2.6)

$$r(\mu_t) = \lambda \|\mu_t\|_1.$$

(2.7)

Note that the mean-regularizer is included in (2.6) and not in (2.7) because COGD, like the more general COMID, only handles round-independent regularizers. The adjustment signal update is obtained applying (2.5) to (2.1), which yields

$$\mu_{t+1} = \arg\min_{\mu \in [-1,1]^N} \left\{ \frac{1}{2} \|\mu_t - \mu\|_2^2 + \eta \lambda \|\mu\|_1 + \eta \left[ -2c_t(s_t - c_i^T \mu_t) + \frac{2\rho}{t} \frac{(t-1)\langle \mu \rangle_{t-1} + \mu_t}{t} \right]^T \mu \right\}. \quad (2.8)$$

The COGD algorithm for setpoint tracking is given in Algorithm 2.1.

We conclude this section by presenting Proposition 2.1. This result follows from Lemma 2.1 and establishes that our COGD algorithm for setpoint tracking achieves sublinear regret.

**Proposition 2.1.** Let $\mu_1 = 0$, $\chi \geq 1$, $f_t(\mu_t) \leq B$ for all $t$ and set

$$\eta = \chi \sqrt{\frac{4N}{G^2T}}.$$  

(2.9)
Algorithm 2.1 COGD for setpoint tracking algorithm

1: **Parameters:** Given $T$, $\lambda$, $\rho$ and $\chi$.
2: **Initialization:** Set $\mu_1 = 0$ and set $\eta$ as in (2.9).
3: for $t = 1, 2, \ldots, T$ do
4:  Deploy adjustment according to $\mu_t$.
5:  Compute the gradient,
6:     $$\nabla f_t(\mu_t) = -2c_t(s_t - c_t^\top \mu_t) + \frac{2\rho}{t} \frac{(t-1)\langle \mu \rangle_{t-1} + \mu_t}{t}.$$  
7:  Update load dispatch,
8:     $$\mu_{t+1} = \arg \min_{\mu \in [-1,1]^N} \eta \nabla f_t(\mu)^\top \mu + \frac{1}{2} \|\mu_t - \mu\|_2^2 + \eta \lambda \|\mu\|_1.$$  
9: end for

Then, the COGD for setpoint tracking algorithm has a regret bound given by

$$\text{Regret}_T \leq 4\chi \sqrt{TKB}, \quad (2.10)$$

where $K = \max_{t=1,2,\ldots,T} \{\rho, \|c_t\|_2^2 \}.$

The reader is referred to Section 2.7.2 for the proof of Proposition 2.1. We subsequently refer to the assumptions of this section as the full information setting.

## 2.4 Limited feedback

In this section, we present three limited-feedback extensions to the OCO model introduced in the last section. We give bounded-regret algorithms for each extension. The mean regularizer can be used in all limited feedback settings with a weighted mean only if the weights are not a function of previous rounds.

### 2.4.1 Bandit feedback

In the bandit feedback setting, the aggregator only observes aggregate effect of its decision in each round. Specifically, it sees $c_t^\top \mu_t$ and $s_t$. Hence, no gradient can be computed for the COGD update (2.8). The motivation for this setting is, for example, the lack of proper metering and communication infrastructure in a distribution network. Using only the total loss function, a point-wise gradient estimator $g_t$ can be computed at each round. This point-wise gradient is computed using the approach proposed by [24, 63] which gives $E[g_t] \approx \nabla f_t$. The following lemma shows that for quadratic functions, $E[g_t] = \nabla f_t$.

**Lemma 2.2** (Point-wise gradient estimator of quadratic functions). Let $h(\mu) = \mu^\top Q \mu + p^\top \mu + r$, where $\mu, p \in \mathbb{R}^N$, $Q \in \mathbb{R}^{N \times N}$ and $r \in \mathbb{R}$. Let $v$ be a random variable uniformly sampled from

$$S_1^N = \left\{ v \in \mathbb{R}^N \mid \|v\|_2 = 1 \right\}.$$
Define \( g \) to be the point-wise gradient estimator of \( h \) given by,

\[
g = \frac{N}{\delta} h(\mu + \delta v)v,\]

where \( \delta > 0 \). Then,

\[
E_{v \sim S_1^N}[g] = \nabla h(\mu).
\]

The proof of Lemma 2.2 can be found in Section 2.7.3. We define the point-wise gradient estimator for setpoint tracking as

\[
g_t = \frac{N}{\delta} f_t(\mu_t + \delta v_t)v_t, \quad (2.11)
\]

where \( \delta > 0 \) is a numerical parameter, \( v_t \) is sampled uniformly from \( S_1^N \) and \( f_t \) is given in (2.6). The expectation of the point-wise gradient (2.11) over \( v_t \) is equal to the gradient of \( f_t \).

We must also modify the decision set so that the estimated gradient does not lead to an infeasible step. Define the convex set \( K^\delta \subset K \) as

\[
K^\delta = \left\{ \mu \mid \frac{\mu}{1 - \delta} \in K \right\} = [\delta - 1, 1 - \delta]^N. \quad (2.12)
\]

The update for the bandit-COGD (BCOGD) algorithm is then given by

\[
\mu_{t+1} = \arg \min_{\mu \in K^\delta} \eta g_t^\top \mu + \frac{1}{2} \| \mu_t - \mu \|_2^2 + \eta r(\mu). \quad (2.13)
\]

The full algorithm is presented in Algorithm 2.2.

**Algorithm 2.2** BCOGD for setpoint tracking with limited feedback algorithm

1. **Parameters:** Given \( T, \rho, \lambda \) and \( \chi \).
2. **Initialization:** Set \( \mu_1 = 0 \) and set \( \eta \) and \( \delta \) according to (2.14).
3. **for** \( t = 1, 2, \ldots, T \) **do**
4. Sample \( v_t \sim S_1 \).
5. Deploy adjustment according to \( \mu_t + \delta v_t \).
6. Suffer loss \( l_t(\mu_t + \delta v_t) \).
7. Compute the point-wise gradient,

\[
g_t = \frac{N}{\delta} f_t(\mu_t + \delta v_t)v_t.\]

8. Update load dispatch,

\[
\mu_{t+1} = \arg \min_{\mu \in [\delta - 1, 1 - \delta]^N} \left\{ \eta g_t^\top \mu + \frac{1}{2} \| \mu_t - \mu \|_2^2 + \eta \lambda \| \mu \|_1 \right\}.
\]

9. **end for**

**Theorem 2.1** (Regret bound for BCOGD). Let \( F_t(\mu_t) \) be \( L \)-Lipschitz and \( B \)-bounded for all \( t \) and let \( r(\mu_1) = 0 \). Then, using the point-wise gradient estimator \( g_t \) and setting

\[
\eta = \frac{D \chi}{B NT^2}, \quad \delta = \frac{1}{T^2}, \quad (2.14)
\]
where $D = \text{diam } \mathcal{K}$ and $\chi \geq 1$, the BCOGD regret is upper bounded by

$$E[\text{Regret}_T(\text{BCOGD})] \leq (DBN\chi + 2DL + 2L)T^{3/2}. \quad (2.15)$$

The proof of Theorem 2.1 is given in Section 2.7.4. The $O(T^{3/4})$ bound is similar to the regret of other bandit OCO algorithms [24,28,63]. Observe that there is an increase in the round-dependence of the bound from 1/2 to 3/4 due to less information being available to the algorithm.

This result is a generalization of the bandit-OCGD proposed in [28], which is based on the bandit-OGD of [63]. The algorithm in [28] imposes a lower bound on the time horizon $T$. In [28], the bandit algorithm is limited to cases where the time horizon satisfies

$$T \geq \frac{1}{\rho_{\text{in}}} \left( \frac{\rho_{\text{out}} B'N}{L + 2\nu} \right)^2,$$

where $|F_t(\mu_t)| \leq B'$ for all $\mu_t \in \mathcal{K}$ and $t = 1, 2, \ldots, T$, $\rho_{\text{in}}$ and $\rho_{\text{out}}$ are respectively the radius of a ball centered at the origin contained in and containing $\mathcal{K}$. Not meeting this condition is equivalent to setting $\delta$ in [2.12] greater than one, which leads to an empty feasible set when updating the prediction. This time horizon condition also makes $\eta$, the gradient descent step size, potentially very small, leading to insufficient change between each round.

Our generalization solves the issue of [28] requiring a very large time horizon and small $\eta$, the update step size, in the case of setpoint tracking. Furthermore, the tuning parameter, $\chi$, allows more control over the step size.

When applied to setpoint tracking, the regret of BCOGD reduces to

$$E[\text{Regret}_T(\text{BCOGD})] \leq \left( 2N^{3/2}B\chi + 4\sqrt{NL} \right) T^{3/2}.$$

2.4.2 Partial bandit feedback

We now present the Partial Bandit COGD (PBCOGD) algorithm for when the aggregator only receives full feedback from a subset of the loads and has access to the aggregate effect of its decision, $c_i^T \mu_t$. This could represent a total power measurement, e.g., at a substation. This extension applies, for example, when individual monitoring has only been implemented for a subset of the loads, when some loads have opted out due to privacy issues, or when the aggregator is subject to bandwidth constraints.

Let $n \in \{1, 2, \ldots, N-1\}$ be the number of loads for which full information is available in each round. We denote $\mu_t^F \in \mathcal{K}_n$ and $\mu_t^B \in \mathcal{K}_{N-n}$ as the decisions variables and sets for the full information and the bandit-like load subsets, where $\mathcal{K}_n \subseteq \mathbb{R}^n$ and $\mathcal{K}_{N-n} \subseteq \mathbb{R}^{N-n}$ are compact and convex sets. We can write $\mu_t$ as $\mu_t = \mu_t^B + \mu_t^F$ where $\mu_t^B = (\mu_t^B, 0)^T \in \mathcal{K}_{N-n}$ and $\mu_t^F = (0, \mu_t^F)^T \in \mathcal{K}_n$.

Define $\beta_t = c_i^T \mu_t - c_i^F^T \mu_t^F$, where $c_i^F$ is the subset of responses corresponding to the full information loads. The update for the PBCOGD is,

$$\mu_{t+1} = \arg \min_{\mu_t^F \in \mathcal{K}_n, \mu_t^B \in \mathcal{K}_{N-n}} \left\{ \frac{1}{2} \| \mu_t - \mu^* \|_2^2 + \eta_1 r(\mu_t^B) + \eta_2 r(\mu_t^F) + \eta_3 g_i^T \mu_t^B + \eta_4 \nabla \mu_t^B f_i^F(\beta_t, \mu_t^F)^T \mu_t^F \right\} \quad (2.16)$$

where $f_i^F(\beta_t, \mu_t^F)$ is a function of $\beta_t$ and the full information decision variables and is equal to $f_i(\mu_t)$. The definition of $g_i^p$, the point-wise gradient for PBCOGD, is given with the full algorithm in Figure 2.3.
Algorithm 2.3 PBCOGB for setpoint tracking with partial feedback algorithm

1: **Parameters:** Given $N$, $n$, $T$, $\rho$, $\lambda$ and $\chi$.
2: **Initialization:** Set $\mu_1 = 0$ and set $\eta_1$ and $\delta$ according to (2.14) and $\eta_2$ according to (2.2).
3: for $t = 1, 2, \ldots, T$ do
4: Sample $v_p^t \sim S_{n \mid n}$.
5: Dispatch adjustment according to $\mu_t = \begin{pmatrix} \mu^B_t \\ \mu^F_t \end{pmatrix} + \begin{pmatrix} \delta v_p^t \\ 0 \end{pmatrix}$.
6: Suffer loss $\ell_t(\mu_t)$ and compute $\beta_t = c_t^\top \mu_t - c_t^F \mu^F_t$.
7: Compute the gradients,
   
   $g_t^p = \frac{(N - n)}{\delta} f_t \begin{pmatrix} \mu_t + \delta \begin{pmatrix} v_p^t \\ 0 \end{pmatrix} \end{pmatrix} v_p^t$,

   $\nabla_{\mu^F} \tilde{f}_t^F(\beta_t, \mu^F_t) = -2c_t^F (s_t - \beta_t - c_t^F \mu^F_t) + \frac{2\rho}{t} (t - 1) (\mu^F)_{t - 1} + \mu^F_t$.
8: Update adjustment dispatch,

   $\mu_{t+1} = \begin{pmatrix} \mu^B \\ \mu^F \end{pmatrix}$

   $\arg \min_{\mu^F \in [-1, 1]^N} \{ \eta_1 g_t^p \mu^B_t + \eta_2 \nabla_{\mu^F} \tilde{f}_t^F(\mu^F_t) \mu^F_t + \eta_1 \lambda \| \mu^B \|_1$

   $+ \eta_2 \lambda \| \mu^F \|_1 + \frac{1}{2} \| \begin{pmatrix} \mu^B_t \\ \mu^F_t \end{pmatrix} - \begin{pmatrix} \mu^B \\ \mu^F \end{pmatrix} \|_2^2 \}$
9: end for

We then have the following regret bound.

**Theorem 2.2** (Regret bound for PBCOGB). Assume that $r(\mu_t)$, the regularizer, satisfies $r(\mu_t) = r(\mu^B_t) + r(\mu^F_t)$ and $\beta_t$ is available for all $t$. Let $\eta_1$ and $\delta$ be defined as in (2.14) and set $\eta_2$ as in (2.2). Then, under the assumptions of Lemma 2.1 and Theorem 2.1, PBCOGB achieves $O \left( T^3 \right)$ regret bound.

The proof of Theorem 2.2 is given in Section 2.7.5.

We now apply the PBCOGB to setpoint tracking. We have $r(\mu) = \lambda \| \mu \|_1$ which respects the regularizer condition and

$\tilde{f}_t^F(\beta_t, \mu^F_t) = \left( s_t - \beta_t - c_t^F \begin{pmatrix} 0 \\ \mu^F_t \end{pmatrix} \right)^2$.

The PBCOGB for setpoint tracking with partial feedback algorithm is presented in Figure 2.3. By Theorem 2.2, the regret of PBCOGB for setpoint tracking is sublinear.

### 2.4.3 Bernoulli feedback setting

We now consider an extension in which the decision maker receives full feedback in some rounds and bandit feedback in others, depending on the outcome of a Bernoulli random variable. The motivation
Chapter 2. Setpoint Tracking with Partially Observed Loads

of the Bernoulli feedback setting is to reduce the communication requirement between loads and the load aggregator without degrading setpoint tracking performance. Decreasing communication needs has several advantages such as lower infrastructure costs and enhanced privacy [64,65]. Reduced communication also improves performance under unreliable communication due to, for example, data transmission jamming [65,66] or when using existing channels opportunistically, i.e., only when spare communications capacity is available [67,68].

During the initialization of the algorithm, the player randomly determines the feedback types of all rounds. We present here the Bernoulli-COGD (BerCOGD) algorithm to deal with this new type of feedback.

If the probability of bandit feedback in a given round is below a threshold, we can obtain a regret upper-bound of $O\left(\frac{T_1}{T^2}\right)$ for the BerCOGD algorithm, similarly to the COGD algorithm for full feedback.

We denote the feedback type in each round with the random variable $I_t \sim \text{Bernoulli}(p)$ for $t = 1, 2, \ldots, T$, with $I_t = 1$ referring to a bandit feedback and $I_t = 0$ to full feedback.

The BerCOGD is as follows: if $I_t = 0$ at round $t$, then the COGD update is used and if $I_t = 1$, then the BCODG update is used. The only difference is that in the bandit feedback update, the projection of $\mu_t$ onto $K^\delta$ is done in two steps. When the decision $\mu_{t+1}$ using the bandit update is computed, it is projected onto $K$ rather than $K^\delta$ as shown on line 15 of the algorithm in Figure 2.4. The projection onto $K^\delta$ is made at the beginning of a bandit round (see line 10 of the algorithm). This is necessary to ensure that a bandit round can follow a full information round.

**Theorem 2.3 (Regret bound for BerCOGD).** Let the feedback type in round $t$ be determined by the random variable $I_t \sim \text{Bernoulli}(p)$ where $p \leq \frac{1}{T^2}$ and $a \in \mathbb{R}$. Then, under the assumptions of Lemma 2.1 and Theorem 2.1 and setting

$$
\eta_F = \frac{D_{\chi_F}}{G(T - T_B + 1)^{\frac{2}{3}}}, \quad \eta_B = \frac{D_{\chi_B}}{BN(T_B + 1)^{\frac{2}{3}}},
$$

$$
\delta = \frac{1}{(T_B + 1)^{\frac{2}{3}}},
$$

the BerCOGD algorithm’s expected regret is bounded by,

$$
\mathbb{E}[\text{Regret}_T(\text{BerCOGD})] \leq \left( A_1 + A_2 a^\frac{2}{3} \right) T^\frac{1}{2} + A_1 + A_2,
$$

where $A_1 = DG_{\chi_F}$, $A_2 = DBN_{\chi_B} + 2DL + 2L$.

The proof of Theorem 2.3 is given in Section 2.7.6. We note that as $T$ gets larger, we can accommodate less and less bandit rounds. However, the constant $a$ of the probability $p$ can be tuned to increase the ratio of bandit rounds. Another approach is to choose a smaller $T$ and periodically reinitialize the algorithm.

We now apply BerCOGD to the setpoint tracking problem. The algorithm is shown in Figure 2.4. The gradient $\nabla f_t(\mu_t)$ and point-wise gradient $g_t$ are defined as in Section 2.3 and Section 2.4.1. Applying Theorem 2.3 to the BerCOGD for setpoint tracking algorithm, the regret bound (2.18) becomes

$$
\mathbb{E}[\text{Regret}_T] \leq \left( \sqrt{16KB_{\chi_F}^2 + DBN_{\chi_B a^\frac{2}{3}} + 2DLa^\frac{2}{3}} \right) T^\frac{1}{2} + \sqrt{16KB_{\chi_F}^2} + (DBN_{\chi_B} + 2DL).
$$

Thus, (2.19) shows that the regret upper bound is $O(\sqrt{T})$ as in the full information setting even if on average $T_p$ rounds only have access to bandit feedback.
Algorithm 2.4 BerCOGD for setpoint tracking with Bernoulli feedback algorithm

1: Parameters: Given $N$, $a$, $T$, $\rho$, $\lambda$, $\chi_F$, $\chi_B$.

Initialization:
3: Set $p = a/T^{1/3}$ and $\mu_1 = 0$.
4: Sample $I_t \sim \text{Bernoulli}(p)$ for $t = 1, 2, \ldots, T$.
5: Set $T_B$ and set $\eta_F, \eta_B$ and $\delta$ according to (2.17).

6: Run one full information round.
8: Run one bandit feedback round.
9: for $t = 1, 2, \ldots, T$ do
10: if $I_t = 1$ then
11: Bandit feedback round:
12: Project $\mu_t$ in $\mathcal{K}$,
13: $\mu_t^\delta = \arg\min_{\mu \in [\delta-1,1]^N} \|\mu_t - \mu\|_2^2$.
13: Sample $v_t \sim \mathcal{S}_1^N$.
14: Deploy adjustment according to $\mu_t^\delta + \delta v_t$.
15: Suffer loss $f_t(\mu_t^\delta + \delta v_t)$.
16: Compute the point-wise gradient $g_t$.
17: Update load dispatch,
18: $\mu_{t+1} = \arg\min_{\mu \in [-1,1]^N} \left\{ \eta_B g_t^\top \mu + \frac{1}{2} \|\mu_t^\delta - \mu\|_2^2 + \eta \lambda \|\mu\|_1 \right\}$.

19: else if $I_t = 0$ then
20: Full information round:
21: Deploy adjustment according to $\mu_t$.
22: Observe the signal $s_t$ and adjustment performance $c_t$.
23: Compute the gradient $\nabla f_t(\mu_t)$.
24: Update load dispatch,
25: $\mu_{t+1} = \arg\min_{\mu \in [-1,1]^N} \left\{ \eta_F \nabla f_t(\mu_t)^\top \mu + \frac{1}{2} \|\mu_t - \mu\|_2^2 + \eta_F \lambda \|\mu\|_1 \right\}$.

24: end if
25: end for

2.5 Examples

We now apply our OCO setpoint tracking algorithms to two types of flexible loads: TCLs and EVs.

2.5.1 Setpoint tracking using TCLs

2.5.1.1 Modeling

We use the model describing the temperature dynamics of a TCL developed by [69,70] and commonly used in the DR literature, for example by [47,71]. The temperature of a TCL, $\theta_t$, for a time step of length $h$ evolves as

$$\theta_{t+1} = b \theta_t + (1-b) (\theta_{a,t} - m R P_R),$$  \quad (2.20)
where $\theta_{a,t}$ is the ambient temperature at time $t$, $b = e^{-h/RC}$, $m \in \{0,1\}$ is the cooling unit control and $R, C, P_R$ are thermal parameters. Note that this model also applies to heating by instead letting $m$ be negative.

To make the model amenable to OCO, we relax the binary control to $m \in [0,1]$, a common simplification [72,73]. We assume that the unit tries to stay at its desired temperature $\theta_d$ by setting its cooling unit control to $m = m$. Hence, given $\theta_{a,t}$, we define $m_t$ for $t = 1,2,\ldots,T$ as

$$
m_t = \begin{cases} 1, & \text{if } \theta_{a,t} - \theta_d \geq P_R R \\ \frac{\theta_{a,t} - \theta_d}{P_R R}, & \text{if } 0 < \theta_{a,t} - \theta_d < P_R R \\ 0, & \text{otherwise}. \end{cases}
$$

Any deviation at time $t$ from $m_t$ will represent an increase or decrease in power consumption and can hence be used as an adjustment for setpoint tracking. The power consumption of the TCLs can be set proportionally to any value in the interval $[-m_t, 1-m_t]$ for this round. We constrain the adjustment interval to be symmetric so the maximum increase or decrease in power consumption is equal in absolute value. For all loads $i$, we define the average adjustment response as

$$
c_t(i) = p(i) \min\{m_t(i), 1-m_t(i)\}
$$

where $p(i) = P_R(i)/COP(i)$. The cooling unit control is

$$
m_t(i) = \mu_t(i) \min\{m_t(i), 1-m_t(i)\}.
$$

In the definition of $c_t(i)$, $P_R$ is the rated power and $COP$ is the coefficient of performance of the cooling unit.

We rewrite the setpoint tracking loss function (2.4) as

$$
\ell_t(\mu_t) = (s_t - p^\top m_t - c_t^\top \mu_t)^2.
$$

The mean and sparsity regularizers are as in (2.5), and in this context penalize discomfort and using too many TCLs, respectively.

In our numerical simulations, we consider $N = 100$ loads and let $n = 10$ in the partial bandit feedback simulations. The time interval we consider for our simulations represents the middle of a warm day, in which thermostatic loads can provide greater flexibility to the grid. We set the ambient temperature to $\theta_{a,t} = [30 + 0.25 \sin(t/120)]^\circ C$ and the time step to $h = 1$ minute. We set $s_t = 15 \sin(0.1t) + 155$. We define the loads’ response at round $t$ to be $c_t(i) = c_t(i) + w_t$, where $w_t \sim N[-1,1](0,0.5)$, a truncated Gaussian variable, is used to model the uncertainty of the loads’ response for all load $i$ and $t = 1,2,\ldots,T$. This uncertainty can be due to temperature model limitations, e.g., radiant house heating by the sun or open windows.

The TCLs parameters are sampled uniformly from thermal parameters in [58] except for the desired temperature $\theta_d$, which is sampled uniformly between 20$^\circ$C and 25$^\circ$C for all loads for the purpose of increasing the available flexibility.
Table 2.1: Total setpoint tracking improvement \( I \) comparison (averaged over 100 simulations)

<table>
<thead>
<tr>
<th>Feedback type</th>
<th>with regularization</th>
<th>without regularization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full information</td>
<td>96.44% ( (\rho = 125 &amp; \lambda = 5) )</td>
<td>98.29%</td>
</tr>
<tr>
<td>Bandit</td>
<td>36.68% ( (\rho = 0.5 &amp; \lambda = 130) )</td>
<td>41.79%</td>
</tr>
<tr>
<td>Partial Bandit ( (n = 10) )</td>
<td>46.95% ( (\rho = 0.75 &amp; \lambda = 25) )</td>
<td>62.42%</td>
</tr>
<tr>
<td>Bernoulli ( (p = 0.9) )</td>
<td>54.32% ( (\rho = 10 &amp; \lambda = 50) )</td>
<td>64.13%</td>
</tr>
</tbody>
</table>

2.5.1.2 Numerical results

We now present the simulation results for the setpoint tracking algorithms using TCLs. The optimization problem in each update is solved numerically using CVX \[74, 75\] and Gurobi \[76\]. For each case except full information, the average over 100 trials is shown. The parameter \( \chi \) is set to 200 when COGD updates are used, to \( 5.5 \times 10^4 \) when BCOGD updates are used, and to 150 and \( 3 \times 10^4 \) when respectively COGD and BCOGD updates are used in the BerCOGD algorithm. The probability parameter \( a \) is fixed to 7.6 for the BerCOGD algorithm. The regularization parameters \( \rho \) and \( \lambda \) are given in Tables 2.1 and 2.2 for each algorithm.

Figure 2.1 presents the cumulative setpoint tracking loss for each algorithm with and without regularization. Table 2.1 compares the percentage improvement of the total setpoint tracking loss function compared to when no DR is performed, defined as:

\[
I = \frac{\sum_{t=1}^{T} (\ell_t(\mu_t) - \ell_t(0))}{\sum_{t=1}^{T} \ell_t(0)}.
\]

Table 2.2 shows the impact of the regularizers on each algorithm. To quantify the effects of the mean and sparsity regularizers, we compare the Euclidean norm of the mean of \( \mu_t \) up to the current round and the 1-norm of \( \mu_t \) with and without regularization. We average the result over all rounds. The performance indicators for the mean and sparsity regularization are:

\[
M = \frac{1}{T} \sum_{t=1}^{T} \left( \frac{\|\langle \mu_t \rangle_t\|^2}{\|\langle \mu_{no\ reg} \rangle_t\|^2} \right),
\]

\[
S = \frac{1}{T} \sum_{t=1}^{T} \left( \frac{\|\mu_t\|_1}{\|\mu_{t, no\ reg}\|_1} \right),
\]

where the subscript \( no\ reg \) refers to simulation ran without regularization. Table 2.2 presents the results and the regularization parameters \( \rho \) and \( \lambda \) used in the simulations. We note that the sparsity regularizer also promotes a zero mean and that the mean regularizer has a reduced impact on the total setpoint tracking loss.

Figure 2.1 and Table 2.1 show how feedback can improve the setpoint tracking loss. The full information algorithm performs significantly better than the bandit and partial feedback algorithms. The Bernoulli feedback extension, however, offers the second best performance while receiving on average full feedback in 10% of the rounds and bandit in the rest. Lastly, we compare the bandit and the partial bandit algorithms. Figure 2.1 shows the improvement of the cumulative loss by the partial bandit when only 10% of full information variables are available to the load aggregator.

An example of setpoint tracking for all four different types of feedback is presented in Figure 2.2.
Figure 2.1: Cumulative setpoint tracking loss comparison through different feedback type (averaged over 100 simulations)

Table 2.2: Per round average regularizer improvement (averaged over 100 simulations)

<table>
<thead>
<tr>
<th>Feedback type</th>
<th>Mean $\mathcal{M}$</th>
<th>Sparsity $S$</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full information</td>
<td>27.32%</td>
<td>23.96%</td>
<td>$\rho = 125$ &amp; $\lambda = 5$</td>
</tr>
<tr>
<td>Bandit</td>
<td>37.68%</td>
<td>14.55%</td>
<td>$\rho = 0.5$ &amp; $\lambda = 130$</td>
</tr>
<tr>
<td>Partial Bandit (n=10)</td>
<td>31.76%</td>
<td>4.67%</td>
<td>$\rho = 0.75$ &amp; $\lambda = 25$</td>
</tr>
<tr>
<td>Bernoulli ($p = 0.9$)</td>
<td>47.78%</td>
<td>14.34%</td>
<td>$\rho = 10$ &amp; $\lambda = 50$</td>
</tr>
</tbody>
</table>

Figure 2.2 shows the setpoint tracking ability of the proposed approaches under different intensity of feedback when $c_t$, the load response, is uncertain. The random variable in the point-wise gradient estimator of the bandit feedback rounds can add noise to the setpoint tracking curve, as seen in Figures 2.2b, 2.2c and 2.2d. In some rounds, this leads to a high discrepancy between the total adjustment and the setpoint. For these occasions, the load aggregator may have to rely on alternate sources of flexibility.

Figure 2.3 shows the effects of the regularizers in the case of the full information. Note, that these figures show the combined effect of both regularizers. Figure 2.3a shows which loads received a non-zero signal in each round. Figure 2.3b shows that regularizers prevent loads from being dispatched unevenly over time.

Lastly, Figure 2.4 presents the temperature of a subset of TCLs and shows the combined effect of both regularizers in the full information setting. We observe that for some loads the temperature is almost constant reflecting the sparsity of the adjustment signal while for the others, the temperature oscillates in the vicinity of their desired temperature given by the temperature at $t = 1$. Without regularization, the temperature drifts away from the desired temperature.
2.5.2 Setpoint tracking using EVs storage unit

2.5.2.1 Modeling

We now apply OCO to setpoint tracking with EVs. To adapt the framework to EVs, some assumptions must be modified. We assume that EVs can either provide or store energy. We also assume that the initial state-of-charge, $S_1$, is at 75% for all EVs. We split the adjustment signal $\mu$ into a charging signal $\mu_c \in [0, 1]^N$ where the EV stores energy and hence increases its power consumption, and a corresponding discharging signal $\mu_d \in [-1, 0]^N$. We make this assumption to avoid a non-convex loss function. We note that this formulation implies that charging and discharging can occur at the same time. As it will be shown later, this unwanted consequence of the convex relaxation is avoided with the sparsity regularizer.

We let $c_{c,t}$ and $c_{d,t}$ be the maximum charging and discharging rate at round $t$. The setpoint tracking loss function is written as

$$\ell_t(\mu_t) = (s_t - c_{c,t}^T \mu_{c,t} - c_{d,t}^T \mu_{d,t})^2.$$  

We modify the mean regularizer for the present non-symmetric formulation because charging and dis-
charging have different effects on the state-of-charge. The state-of-charge at time $t+1$ for the EVs $i$ is,

$$S_{t+1}(i) = S_t(i) + \frac{h}{B} \left[ \eta_{s}^{\text{inj}}(i)c_{c,t}(i)\mu_{c,t}(i) + \frac{1}{\eta_{s}^{\text{ext}}(i)}c_{d,t}(i)\mu_{d,t}(i) \right],$$  \hspace{2em} (2.21)

where $\eta_{s}$ is the extraction/injection coefficient, $B$ is the battery energy capacity and $h$ is the length of the time step. We neglect battery leakage in (2.21) because it is likely to be very small on the faster
timescales we are considering. Defining $\langle \mu^w \rangle_t$ as the weighted average of charging and discharging signals for EV, we have

$$\langle \mu^w (i) \rangle_t = \frac{1}{t} \sum_{n=1}^{t} \left( \eta^i_{n} \mu_{c,n}(i) + \frac{1}{\eta^i_{s,n}} \mu_{d,n}(i) \right),$$

for $i = 1, \ldots, N$. The mean regularizer is the squared $\ell_2$-norm of the vector $\langle \mu^w \rangle_t$, and prevents the EVs from being dispatched far from their initial state of charge.

For this application example, we only use the full information setting. We do not apply limited feedback because the randomly perturbed update does not ensure that only one of decision variables, $\mu_c$ or $\mu_d$, be non-zero in each time period.

We set $N = 100$ and the time step $h = 1$ minute. The load response is $c_{c,t}(i) = T_{\text{charging}} + w_t$ and $c_{d,t}(i) = T_{\text{discharging}} + w_t$ where $w_t \sim N[-1.5,1.5](0,0.1)$, a truncated Gaussian noise, models the uncertainty. The setpoint to track is $s_t = 25 \sin(0.1t)$. For the storage unit, we use $\eta^i_{inj}(i) = \eta^i_{ext}(i) = 0.85$ for all $i$, a battery capacity of $B = 10$ kWh and charging and discharging rates given by $T_{\text{charging}} = 3$ kW and $T_{\text{discharging}} = 1.5$ kW. These parameters are chosen according to [77–79]. Finally, we set $\chi = 35$, $\lambda = 46$ and $\rho = 100$.

2.5.2.2 Numerical Results

We now present numerical simulation results for the setpoint tracking algorithm using EVs. Using our proposed approach, we observe a decrease of 67.08% in the total setpoint tracking loss. Using the regularizers, in average per round, the sparsity is improved by 84.41% and the impact on comfort decreases by 59.71%. This results means that the drift in state-of-charge level is significantly reduced.

Finally, we observe that almost no simultaneous charging and discharging occurs when regularizers are used in the full information setting. Figure 2.5 presents $\max_i |\mu_{c,t}(i)|$ and $\max_i |\mu_{d,t}(i)|$, the maximum signal with and without regularization as a function of $t$. Note that this does not represent the energy in and out of the storage units but only the instruction sent to them. In Figure 2.5 bottom subfigure, as expected when no regularization is used, charging and discharging are dispatched at the same time. Figure 2.5 top subfigure, shows that with the regularizers, simultaneous charging and discharging almost never occurs.

2.6 Conclusion

In this work, we propose several online convex optimization algorithms for setpoint tracking using demand response when the load’s response is uncertain. We consider three limited feedback extensions: bandit feedback, partial bandit feedback, and Bernoulli feedback. We add two regularizers to our formulation to minimize the number of loads dispatched in each round and to minimize the impact on the loads. We show that each extension’s regret has a sublinear upper bound. We apply our model to thermostatically controlled loads and electric vehicles. In our numerical tests, we showed that the limited feedback implementations can still offer good performance while requiring less communication. A relevant extension to this work would be to investigate pricing mechanisms based on our proposed
Figure 2.5: Charging $\mu_c$ and discharging $\mu_d$ signal comparison with regularization (top) and without regularization (bottom).

framework, for example using dual variables of online convex optimization algorithms. Finally, another interesting approach would be to study the performance of our proposed algorithms from a dynamic regret [23] perspective. Dynamic regret may provide bounds that more closely match the performance of the algorithms, and thus serve as better design tools. This is a topic for future work.

2.7 Proofs

Here we give proofs of all theoretical claims.

2.7.1 Proof of Lemma 2.1

Let $\eta = \chi \eta_C$ where $\eta_C$ is defined as in [30 Corollary 4]. Using [30 Corollary 3], the COMID algorithm’s regret is bounded by

$$\text{Regret}_T(\text{COMID}) \leq \frac{B_R(\mu^*, \mu_1)}{\eta} + \frac{T\eta G^2}{2\alpha} + r(\mu_1),$$

where $r(\mu_1) = 0$ by assumption. Substituting (2.2) in (2.23) and given that $\chi \geq 1$, we obtain,

$$\text{Regret}_T(\text{COMID}) \leq \frac{\sqrt{B_R(\mu^*, \mu_1)G^2T}}{\sqrt{2\alpha \chi}} + \chi\frac{\sqrt{B_R(\mu^*, \mu_1)G^2T}}{\sqrt{2\alpha}},$$

$$\leq \frac{\sqrt{2TB_R(\mu^*, \mu_1)G^2} \chi^2}{\alpha}.$$

Note that setting $\chi = 1$, we retrieve the bound of [30].
2.7.2 Proof of Proposition 2.1

Proposition 2.1 follows from Lemma 2.1. We have $\alpha = 1$ since the regularization function, $R(\cdot) = \frac{1}{2} \| \cdot \|_2^2$, is 1-strongly convex and recall the bound on $B_R$ given in Section 2.2.2.

The norm of the gradient of $f_t$ is upper-bounded by,

$$\|\nabla f_t(\mu)\|^2 = \|\nabla \ell(\mu) + \nabla \left( \rho \|\langle \mu \rangle_t \|\right)\|^2,$$

$$\leq \|\nabla \ell(\mu)\|^2 + \|\nabla \left( \rho \|\langle \mu \rangle_t \|\right)\|^2,$$

$$\leq 4 \|c_i\|^2 \ell_t(\mu) + 4 \left( \frac{\rho}{t} \right)^2 \|\langle \mu \rangle_t\|^2,$$

$$\leq 4 \max_{t=1,2,...,T} \left\{ \|c_i\|^2, \rho \right\} \left( \ell_t(\mu) + \rho \|\langle \mu \rangle_t\|\right),$$

$$= 4 \max_{t=1,2,...,T} \left\{ \|c_i\|^2, \rho \right\} f_t(\mu).$$

and we obtain the bound given in (2.10).

2.7.3 Proof of Lemma 2.2

We start by expanding the expectation of the point-wise gradient of a quadratic function.

$$E_{\nu \sim S_N^c} [g] = \frac{N}{\delta} E \left[ \left( (\mu + \delta v)^\top Q(\mu + \delta v) + p^\top (\mu + \delta v) + r \right) v \right]$$

$$= \frac{N}{\delta} E \left[ \left( \mu^\top Q \mu + \delta v^\top Q \mu + \delta v^\top Q v + \delta^2 v^\top Q v + p^\top \mu + \delta p^\top v + r \right) v \right]$$

$$= \frac{N}{\delta} E \left[ \left( \mu^\top Q \mu + \mu^\top p + r \right) v \right] + \frac{N}{\delta} E \left[ \delta^2 v^\top Q v \right] + \frac{N}{\delta} E \left[ \left( \delta^2 v^\top Q \mu + \mu^\top Q v + \mu^\top p \right) v \right]$$

$$= \frac{N}{\delta} \left( \mu^\top Q \mu + \mu^\top p + r \right) E[v] + N\delta E \left[ v^\top Q v \right] + N E \left[ v^\top v \right] \left( Q \mu + Q^\top \mu + p\right) \quad (2.24)$$

Then, we compute the expected values with respect to $v \sim S_N^c$. By symmetry of the uniform distribution over $S_N^c$, $E_\nu[v]$ and $E_\nu[Q_2 Q v v]$ are equal to 0. To compute the expectation of the quadratic term $E_\nu[v vv^\top]$, we define

$$v_a = (v_1, v_2, \ldots, v_i, \ldots, v_N)^\top$$

$$v_b = (v_1, v_2, \ldots, -v_i, \ldots, v_N)^\top$$

where $v_i$ is the $i^{th}$ component of $v$. It follows that $v$, $v_a$ and $v_b$ are uniformly distributed on $S_N^c$. Therefore, $v$, $v_a$ and $v_b$ have the same correlation, that is $E[v vv^\top] = E[v_a v_a^\top] = E[v_b v_b^\top]$. Consequently, for $i \neq j$,

$$E[v_i v_j] = E[(v_i) v_j],$$

$$= -E[v_i v_j],$$

$$= 0,$$
and for $i = j$,

$$\mathbb{E}[v_i v_i] = \mathbb{E}[(v_i)^2].$$

This means that $\mathbb{E}[vv^\top]$ is a diagonal matrix. We use [80, Theorem 2.1 (3)] to compute

$$|v_i|^q \sim \text{Beta}\left(\frac{1}{q}, \frac{N-1}{q}\right)$$

Setting $q = 2$,

$$\mathbb{E}[(v_i)^2] = \frac{1}{N}$$

for $i = 1, 2, \ldots, N$. The expected value of the quadratic term is,

$$\mathbb{E}[vv^\top] = \frac{1}{N}I,$$

where $I$ is the identity matrix. Finally, (2.24) reduces to,

$$\mathbb{E}_{v \sim \mathcal{S}_N^1}[g] = (Q + Q^\top)\mu + p,$$

$$= \nabla h(\mu).$$

### 2.7.4 Proof of Theorem 2.1

We use the proof technique of [24, Theorem 6.6] and apply it to the COMID instead of the online gradient descent. We start by deriving three relations to bound the regret.

First, let $\mu^*$ be the best fixed action in hindsight and let $\hat{\mu}^*$ be the projection of $\mu^* \in \mathcal{K}$ onto $\mathcal{K}^\delta$. $\mathcal{K}^\delta \subseteq \mathcal{K}$ where all directions of the original set $\mathcal{K}$ are reduced by a factor of $\delta$. Consequently, the reduction along any axis of $\mathcal{K}^\delta$ is less or equal to $\delta D$, with equality in the direction that leads to the diameter. By the definition of the projection we have that $\hat{\mu}^*$ is the closest point in $\mathcal{K}^\delta$ to $\mu^*$ and $\hat{\mu}^*$ lies on the boundary of $\mathcal{K}^\delta$. Hence,

$$||\hat{\mu}^* - \mu^*||_2 \leq \delta D$$

This relation upper-bounds the difference between the best fixed action in hindsight and the closest action in the feasible set one can expect obtaining using BCOGD. By the Lipschitz assumption,

$$|F_t(\hat{\mu}^*) - F_t(\mu^*)| \leq \delta LD$$

Second, using the Lipschitz assumption again, we have

$$|F_t(\mu_t + \delta v_t) - F_t(\mu_t)| \leq \delta LD,$$

for any $\mu_t \in \mathcal{K}^\delta$ and $v_t \sim \mathcal{S}_N^1$. The previous relation bounds the difference between the BCOGD update in (2.13) and the actual randomly perturbed decision played by the algorithm.

Third, the definition of the point-wise gradient $g_t$ is valid for any $\delta$-smoothed [24, Lemma 6.4]. The reader is referred to [24] for the definition of $\delta$-smoothed functions. Let $\tilde{F}_t$ be the $\delta$-smoothed version of $F_t$. Then, by the Lipschitz propriety of $F_t$ and Lemma 2.6 of [24], we have,

$$\left|\tilde{F}_t(\mu_t) - F_t(\mu_t)\right| \leq \delta L$$
We now use these three relations to bound the regret of the BCGD algorithm.

\[
E[\text{Regret}_T(\text{BCGD})] = E \left[ \sum_{t=1}^{T} F_t(\mu_t + \delta v_t) - F_t(\mu^*) \right] \\
= \sum_{t=1}^{T} E \left[ F_t(\mu_t + \delta v_t) \right] - F_t(\mu^*) \\
\leq \sum_{t=1}^{T} E \left[ F_t(\mu_t) \right] - F_t(\mu^*) + \delta DL \\
\leq \sum_{t=1}^{T} E \left[ \hat{F}_t(\mu_t) \right] - \hat{F}_t(\hat{\mu}^*) + 2\delta DL + 2\delta L \\
= E \left[ \text{Regret}_T(\text{COGD}; g_1, g_2, \ldots, g_T) \right] + \sum_{t=1}^{T} 2\delta DL + 2\delta L \\
\leq \frac{B_R(\hat{\mu}^*, \mu_1) - \eta}{\eta} + \frac{T \eta G^2}{2\alpha} + r(\mu_1) + 2\delta DL T + 2\delta LT, \tag{2.28}
\]

where previously stated relations were used to get (2.25)-(2.27). Lastly, we used Lemma 2.1 to obtain (2.28) where \(E[g_t] = \nabla f_t(\mu_t)\) and \(\|g_t\| \leq G\) for all \(t\). By assumption, \(r(\mu_1) = 0\) and recalling that \(B_R(\mu^*, \mu_1) \leq D^2/2\) and \(\alpha = 1\), we have

\[
E[\text{Regret}_T(\text{BCGD})] \leq \frac{D^2}{2\eta} + \frac{T \eta G^2}{2} + 2\delta DL T + 2\delta LT.
\]

Using the definition of the point-wise gradient we have \(\|g_t\| = \frac{N}{\tau} f_t(\mu_t) + \frac{\delta v_t}{\tau} \leq \frac{N}{\tau} B = G\). Hence,

\[
E[\text{Regret}_T(\text{BCGD})] \leq \frac{D^2}{2\eta} + \frac{N^2 T \eta B^2}{2\delta^2} + 2\delta DL T + 2\delta LT.
\]

Setting \(\eta\) and \(\delta\) according to (2.14), we obtain,

\[
E[\text{Regret}_T(\text{BCGD})] \leq \frac{D^2 B N T \frac{3}{4}}{2D\chi} + \frac{D \chi N B T \frac{4}{4}}{2} + 2DL T \frac{3}{2} + 2L T \frac{4}{2} \\
\leq (DBN \chi + 2DL + 2L) T \frac{4}{4} \tag{2.27}
\]

2.7.5 Proof of Theorem 2.2

Assume that the aggregate response of the unobserved loads, \(\beta_t\), is available in all rounds \(t\). Note that all decision variables are known under limited feedback and hence do not change the regularizer, \(r(\mu_t)\).

Similar to \(\beta_t\), we define \(i_t\) as the effect due to the full information decision variable at round \(t\). Then, the loss function \(f_t(\mu_t)\) can be rewritten as a function of bandit or full information feedback decision variable only,

\[
f_t(\mu_t) = \tilde{f}^F_t(\beta_t, \mu^F_t) = \tilde{f}^B_t(i_t, \mu^B_t). \tag{2.29}
\]

where both \(\tilde{f}^F_t\) and \(\tilde{f}^B_t\) can be computed at round \(t\) and each of them are a function of only one type of
feedback. Hence, the loss function can be rewritten as,
\[ f_t(\mu_t) = \alpha \tilde{f}^F(\beta_t, \mu_t^F) + (1 - \alpha) \tilde{f}^B(i_t, \mu_t^B), \]
for \(0 < \alpha < 1\). Note that \(\alpha\) cannot equal zero or one because then the expression would not contain the full \(\mu\) vector, and hence would not contain the full regularizer. We do not allow \(\alpha\) to depend on \(T\). For convenience, we set \(\alpha = 1/2\). The regret of the partial bandit feedback algorithm, PBCOGD, becomes
\[
\text{Regret}_T(\text{PBCOGD}) = \sum_{t=1}^{T} F_t(\mu_t) - F_t(\mu^*)
\]
\[
= \sum_{t=1}^{T} \left( f_t(\mu_t) + r(\mu_t^B) + r(\mu_t^F) \right) - \left( f_t(\mu^*) + r(\mu^B) + r(\mu^F) \right)
\]
\[
= \sum_{t=1}^{T} \left( \frac{1}{2} \left( \tilde{f}^F(\beta_t, \mu_t^F) + \tilde{f}^B(i_t, \mu_t^B) \right) - \frac{1}{2} \left( \tilde{f}^F(\beta^*, \mu^F) + \tilde{f}^B(i_t, \mu^B) \right) \right)
\]
\[
+ r(\mu_t^B) + r(\mu^F) - r(\mu^B) - r(\mu^*)
\]
\[
= \frac{1}{2} \sum_{t=1}^{T} \left[ \tilde{f}^F(\beta_t, \mu_t^F) + 2r(\mu_t^F) - \tilde{f}^F(\beta^*, \mu^F) - 2r(\mu^F) \right]
\]
\[
+ \frac{1}{2} \sum_{t=1}^{T} \left[ \tilde{f}^B(i_t, \mu_t^B) + 2r(\mu_t^B) - \tilde{f}^B(i_t, \mu^B) - 2r(\mu^B) \right]
\]
(2.30)

The PBCOGD algorithm uses the COGD update on \(\tilde{f}^F(\beta_t, \mu_t^F)\) to compute \(\mu_t^{F+1}\) and the BCOGD update on \(\tilde{f}^B(i_t, \mu_t^B)\) to compute \(\mu_t^{B+1}\). Note that, the COGD and BCOGD algorithms are applied to the loss functions in (2.29). \(\beta_t\) and \(i_t\) are now seen as new sources of uncertainty by the algorithms. For example, in the setpoint tracking loss case, the full information algorithm tracks a signal given by \(s_{t+B}\), rather than \(s_t\).

For completeness, the point-wise gradient \(g_t^p\) used by the BCOGD update is,
\[
g_t^p = \frac{(N - n)}{\delta} \tilde{f}^B(i_t, \mu_t^B + \delta v_t^p) v_t^p
\]
\[
= \frac{(N - n)}{\delta} f_t \left( \mu_t + \delta \begin{pmatrix} v_t^p \\ 0 \end{pmatrix} \right) v_t^p,
\]
as given on line 7 of the algorithm in Figure 2.3, where \(v_t^p \sim \mathcal{N}^{N-n}\) for all \(t\).

Finally, substituting in the regret bounds for COGD and BCOGD, we can rewrite (2.30) as
\[
\text{Regret}_T(\text{PBCOGD}) = \frac{1}{2} R_T(\text{COGD}; \tilde{f}^F_t) + \frac{1}{2} R_T(\text{BCOGD}; \tilde{f}^B_t)
\]
\[
\propto O \left( T^{\frac{7}{2}} \right). \quad \square
\]

2.7.6 Proof of Theorem 2.3

Let \(T_B\) be number of times the player receives bandit feedback over \(T\) trials. Then, \(T_B \sim \text{Binomial}(T, p)\) and the number of times the player receives full feedback is \(T - T_B\). Using the approach of the proof of
Lemma \textit{2.1} \cite{30,40} with $\mathcal{R}(\cdot) = \frac{1}{2} ||\cdot||^2_2$ and of the proof of Theorem \textit{2.1} \cite{40}, we observe that

\[
\mathbb{E}[\text{Regret}_T(\text{BerCOGD})] \leq \mathbb{E} \left[ r(\mu_1) + \frac{D^2}{2} \left( \frac{1}{\eta_F} + \frac{1}{\eta_B} \right) + \frac{1}{2} \sum_{t=1}^T w_t + \sum_{t=1}^T z_t \right]
\]

where,

\[
w_t = \begin{cases} 
\eta_F G^2, & \text{if } I_t = 0 \\
\frac{\eta_B N^2 B^2}{\eta^2}, & \text{if } I_t = 1 
\end{cases}
\]

\[
z_t = \begin{cases} 
0, & \text{if } I_t = 0 \\
2\delta DL + 2\delta L, & \text{if } I_t = 1 
\end{cases}
\]

We remind the reader that $I_t = 1$ in the case of bandit feedback and $I_t = 0$ in the case of full feedback. Therefore, by linearity of $R_T$ and taking into consideration the initialization steps, we re-express the conditional expected regret of the algorithm as

\[
\mathbb{E}[R_T(\text{BerCOGD})] = \mathbb{E} \mathbb{E} [R_T|T_B + 1]]
\]

\[
= \mathbb{E} [\mathbb{E} [R_T(\text{BCOGD})]|T_B + 1]] + \mathbb{E} [\mathbb{E} [R_T(\text{COGD})]|(T - T_B + 1)]
\]

since $r(\mu_1) = 0$. Then, using Lemma \textit{2.1} and Theorem \textit{2.1} for the inner conditional expectations, we have,

\[
\mathbb{E}[\text{Regret}_T(\text{BerCOGD})] \leq \mathbb{E} \left[ A_1 (T - T_B + 1)^{\frac{3}{2}} + A_2 (T_B + 1)^{\frac{3}{2}} \right]
\]

with $A_1 = DG\chi_1$ and $A_2 = DBN\chi_2 + 2DL + 2L$ and where we used

\[
\eta_F = \frac{D\chi_F}{G(T - T_B + 1)^2}
\]

\[
\eta_B = \frac{D\chi_B}{BN(T_B + 1)^3}
\]

\[
\delta_B = \frac{1}{(T_B + 1)^4}
\]

to bound the regret in the inner expected value.

$(T_B + 1)^{\frac{3}{2}}$ and $(T - T_B + 1)^{\frac{3}{2}}$ are concave functions for $0 \leq T_B \leq T$. Hence, by Jensen’s inequality, we have

\[
\mathbb{E}[\text{Regret}_T(\text{BerCOGD})] \leq A_1 \mathbb{E}[(T - T_B + 1)^{\frac{3}{2}}] + A_2 \mathbb{E}[T_B + 1]^{\frac{3}{2}}
\]

\[
\leq A_1 (T - Tp + 1)^{\frac{3}{2}} + A_2 (Tp + 1)^{\frac{3}{2}}
\]

\[
\leq A_1 T^{\frac{5}{2}} + A_2 Tp^{\frac{5}{2}} + A_1 + A_2
\]

since $T_B$ is a binomial random variable and $0 \leq p \leq 1$. Setting $p \leq a/T^{\frac{5}{2}}$ leads to the given expected regret upper bound. Lastly, $\eta_F$, $\eta_B$ and $\delta_B$ are set according to the number of feedback rounds $T_B$ sampled in the initialization step. Note that the initialization rounds are necessary to ensure that $\eta_F$, $\eta_B$ and $\delta_B$ are defined.
Chapter 3

Online Convex Optimization of Multi-energy Building-to-grid Ancillary Services

3.1 Introduction

Multi-energy systems (MESs) couple multiple energy sectors, e.g., electricity, heating, cooling, and transportation [81, 82]. Buildings are a significant class of MES, and can provide useful services to the power grid while increasing their own revenues [5, 47]. In this work, we focus on buildings as MESs. This is a challenging optimization problem due to its high dimensionality and its multiple sources of uncertainty.

We propose a real-time decision algorithm to optimize the energy consumption of a multi-energy system under uncertainty. Several sources of flexibility are available to the MES: the building’s thermal inertia, battery energy storage, and thermal energy storage of water. The MES is also subject to several uncertain and time-varying requirements: indoor temperature regulation, domestic hot water demand, and electric baseload demand. The MES can import and export electricity from and to the grid and generate power from its own photovoltaic (PV) panels.

By leveraging its flexibility, a multi-energy building can contribute to different ancillary services, e.g., contingency services, benefiting the network and generating additional revenue for the building. For example, building-to-grid flexibility could be used by the system operator for fast frequency response, regulation or power balancing services and thus improve the grid’s stability and resiliency of renewable-powered grids [47, 83]. A building must have some communication and control capabilities to provide these services, e.g., measurements of indoor temperature and the ability to send instructions to its cooling system. In this paper we propose a real-time algorithm for one or multiple buildings to provide ancillary services.

We propose a two-level algorithm consisting of a level for lookahead planning under limited information using a mixed-integer linear program (MILP) and a real-time decision level based on online convex optimization (OCO). We design a novel OCO algorithm for time-varying feasible sets to decide the building’s energy usage in real-time. This allows the MES to deal with time-varying and uncertain
constraints like intermittent PV generation while optimizing its energy usage. The approach is flexible due to its real-time decision process and requires relatively little information: only the building’s parameters and approximate information about its environment and users are needed. Our two-level algorithm is not computationally demanding and can be used to aggregate several buildings together to provide network-level ancillary services. We test our approach on a building located in Melbourne, Australia.

**Related work.** Several authors have designed static or offline computation-based approaches for providing flexibility with buildings [73,84–88]. Uncertainty was considered using stochastic optimization in [86] but not in real-time. In [89], the authors aggregated several MES at the community level and computed its financial value using an offline MILP without uncertainty. We use their aggregated MES model, but only assume access to the building’s parameters and a rough estimate of unknowns like mean temperature and solar irradiance.

Model predictive control (MPC) [90,91] is a related approach that has been used for real-time MES optimization under uncertainty [92–94]. In this context, MPC-based approaches require accurate information about the loads, consumer, environment and significant time for computation. This motivates our use of OCO [24, 95], which requires minimal information and computation time. OCO has been previously used for demand response in power systems [28,41]. To the best of the authors’ knowledge, OCO has not to date been used to optimize MES.

Our specific contributions are:

- We propose a two-level algorithm to manage an MES in real-time and under uncertainty (Section 3.4).
- We test our two-level algorithm in a case study based on a building in Melbourne, Australia (Section 3.5) and compare its performance to a model predictive control-based approach (Section 3.5.3).
- We propose a new OCO algorithm for time-varying constraints and provide a sublinear regret bound for the algorithm (Section 3.4.3).

We first detail the models used for different buildings’ systems in Section 3.2. The scheduling level of our approach is presented in Section 3.3 followed by the tracking level in Section 3.4. We present an analysis of the new OCO algorithm in Section 3.4.3. A detailed case study is presented in Section 3.5 and the performance of the two-level approach is shown. We conclude by highlighting the key points of our approach in Section 3.6.

### 3.2 MES modeling

We present our two-level algorithm for managing building energy consumption under uncertainty. In this work, uncertainty takes the form of inexact information, e.g., about natural phenomena such as weather forecast, and is modeled using random variables.

The first level is a scheduling program. The goal of the scheduling program is to use predictions, e.g. electric market average data, to plan an energy consumption trajectory that maximizes the building benefits while satisfying its constraints. This is formulated as a mixed-integer linear program. The output of the scheduling MILP is a sequence of energy consumption objectives for the building to follow in the second level of the algorithm. The second level of the algorithm is an OCO-based real-time tracking algorithm. Each scheduling round is then split into tracking rounds of the order of less than
A minute. The tracking algorithm manages the energy resources on a short time-scale to the follow objective set by the scheduling algorithm. Between tracking rounds, the algorithm observes the building parameters and adapts its next decision accordingly. This allows the tracking algorithm to deal with the different sources of uncertainty like weather, scheduling prediction error and unpredictable human behavior.

We define two separate timescales. Let $T = 1, 2, \ldots, \tau$ be the scheduling round index and $\tau$ be the scheduling horizon. The scheduling rounds represent time intervals $H$ of an hour in our case study. Let $t = 1, 2, \ldots, F$ be the tracking round index, where $F$ is the number of tracking round in each scheduling round. The tracking rounds represent time intervals $h$ of thirty seconds. The scheduling and tracking timescales are summarized in Figure 3.1.

Let $R$ be the number of energy flows into, out of, and utilized by the building and $S$ be the number of services the building can offer. Let $x_T \in \mathbb{R}^R$ be the energy resource decision variable during scheduling round $T$ such that

$$x_T = \left( x_{T,\text{import}} \ x_{T,\text{export}} \ s_T^+ \ x_{T,\text{HVAC}} \ x_{T,\text{baseload}} \ x_{T,\text{EB}} \ x_{T,\text{BES}}^+ \ x_{T,\text{TES}} \ p_T \right)^\top,$$

where $s_T \in \mathbb{R}^S$ and $s_T = (s_{T,\text{reg}}^+ s_{T,\text{reg}}^- s_{T,\text{fast}}^+ s_{T,\text{fast}}^- s_{T,\text{slow}}^+ s_{T,\text{slow}}^- s_{T,\text{delay}}^+ s_{T,\text{delay}}^-)^\top$, the service vector. The components of $x_T$ and $s_T$ are detailed in Tables 3.1 and 3.2.

The service vector represents the energy dispatched by the building to ancillary service. Frequency control ancillary services (FCAS) are used to maintain the system frequency within limits by maintaining the balance between demand and generation. More specifically, raise FCAS increases the frequency by increasing generation or reducing demand, while lower FCAS decreases frequency by decreasing the generation or rising demand.

In the Australian National Electricity Market (NEM), there are two types of FCAS: regulation FCAS and contingency FCAS. Regulation FCAS is used to continuously correct minor frequency deviations, while contingency FCAS is used for larger deviations [96]. The contingency FCAS is further grouped into fast (respond within 6 seconds and maintain the service for 60 seconds), slow (respond within 60 seconds and maintain the service for 5 minutes), delayed (respond within 5 minutes and maintain the service no longer than 10 minutes) contingency FCAS. The FCAS providers are paid for their availability, regardless of whether they are called.

![Figure 3.1: Scheduling rounds & decisions (top) and tracking rounds & decisions (bottom)]
Table 3.1: Breakdown of $x_t$

<table>
<thead>
<tr>
<th>Component</th>
<th>Detail (kWh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{T,\text{import}}$</td>
<td>Energy purchased by the building</td>
</tr>
<tr>
<td>$x_{T,\text{export}}$</td>
<td>Energy sold by the building</td>
</tr>
<tr>
<td>$s_T$</td>
<td>Ancillary services provided by the building</td>
</tr>
<tr>
<td>$x_{T,\text{HVAC}}$</td>
<td>Energy used to regulate the building temperature by the heat, ventilation and air conditioning (HVAC) system</td>
</tr>
<tr>
<td>$x_{T,\text{baseload}}$</td>
<td>Energy used to satisfy the building electric baseload demand</td>
</tr>
<tr>
<td>$x_{T,\text{EB}}$</td>
<td>Energy dispatched to the electric water boiler (EB)</td>
</tr>
<tr>
<td>$x_{T,\text{BES}}^d$</td>
<td>Energy taken out of the battery energy storage (BES)</td>
</tr>
<tr>
<td>$x_{T,\text{BES}}^c$</td>
<td>Energy use to charge the battery energy storage</td>
</tr>
<tr>
<td>$x_{T,\text{TES}}$</td>
<td>Energy (heat) taken out of the thermal energy storage (TES)</td>
</tr>
<tr>
<td>$p_T$</td>
<td>Power generated by the PV</td>
</tr>
</tbody>
</table>

Table 3.2: Breakdown of $s_t$

<table>
<thead>
<tr>
<th>Component</th>
<th>Detail: energy consumption (kWh)</th>
<th>Service: FCAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_{T,\text{reg}}^+$</td>
<td>curtailed for regulation</td>
<td>regulation raise</td>
</tr>
<tr>
<td>$s_{T,\text{reg}}^-$</td>
<td>increased for regulation</td>
<td>regulation lower</td>
</tr>
<tr>
<td>$s_{T,\text{fast}}^+$</td>
<td>curtailed for contingency on fast time-scale</td>
<td>contingency fast raise</td>
</tr>
<tr>
<td>$s_{T,\text{fast}}^-$</td>
<td>increased for contingency on fast time-scale</td>
<td>contingency fast lower</td>
</tr>
<tr>
<td>$s_{T,\text{slow}}^+$</td>
<td>curtailed for contingency on slow time-scale</td>
<td>contingency slow raise</td>
</tr>
<tr>
<td>$s_{T,\text{slow}}^-$</td>
<td>increased for contingency on slow time-scale</td>
<td>contingency slow lower</td>
</tr>
<tr>
<td>$s_{T,\text{delay}}^+$</td>
<td>delayed &amp; curtailed for contingency</td>
<td>contingency delay raise</td>
</tr>
<tr>
<td>$s_{T,\text{delay}}^-$</td>
<td>delayed &amp; increased for contingency</td>
<td>contingency delay lower</td>
</tr>
</tbody>
</table>

3.2.1 Building requirements

The building requirements are as follows. The temperature $\theta_T$ at each scheduling round $T$ must stay inside a pre-defined interval set by the building users:

$$\underline{\theta} \leq \theta_T \leq \overline{\theta}, \quad (3.1)$$

where $\underline{\theta}$ and $\overline{\theta}$ are the maximum and minimum temperatures. The maximum and minimum temperature constraints can be a function of the occupancy of the building. When the building’s occupants are absent, a wider temperature interval can be used to increase the flexibility of the load. In this work, we set the temperature constraints to be constant throughout the day. The thermal energy stored at time $T$, $X_T$, must not exceed the thermal limits of the storage unit:

$$\underline{X} \leq X_T \leq \overline{X}, \quad (3.2)$$

where $\underline{X} = (X' - \overline{\theta})$ and $\overline{X} = (X' - \underline{\theta})$ with $X'$ and $X'$ being the maximum or minimum temperature of the thermal energy storage unit. We have reformulated [88]'s constraints $\underline{X} = (X' - \theta_t)$ and $\overline{X} = (X' - \theta_t)$.
as the more-conservative but time-invariant constraint (3.2). The electric energy stored in the battery energy system $B_T$ at time $T$ must not exceed the battery storage unit limits:

$$B \leq B_T \leq \overline{B}, \quad (3.3)$$

where $\overline{B}$ and $\underline{B}$ are the maximum and minimum energy stored in the battery. We add the following constraint on the battery energy level to ensure continuity between days:

$$B_T \geq B_0. \quad (3.4)$$

The electric baseload demand $D_T^{\text{baseload}}$ must be satisfied in each round:

$$x_{T, \text{baseload}} \geq D_T^{\text{baseload}}, \quad (3.5)$$

where we have relaxed the equality constraints into an inequality constraint. This improve the stability of the OCO algorithm. We note that for the optimum at a scheduling or tracking round, the constraints will be active and thus $x_{T, \text{baseload}}$ will be equal to the baseload demand. The domestic hot water (DHW) demand $D_T^{\text{DHW}}$ must be satisfied at each round:

$$x_{T, \text{EB}} \rho^{\text{EB}} + x_{T, \text{TES}} \geq D_T^{\text{DHW}}, \quad (3.6)$$

where $\rho^{\text{EB}}$ is the electric boiler efficiency coefficient. The power generated by the building’s photovoltaic panel can take any positive value below the maximum power generated, $\overline{p_T}$. The constraint is:

$$0 \leq p_T \leq \overline{p_T}. \quad (3.7)$$

### 3.2.2 Energy balance

The energy consumption of the building must satisfy the following balance equations:

$$U_T = x_{T, \text{export}} + x_{T, \text{HVAC}} + x_{T, \text{baseload}} + x_{T, \text{EB}} + x_{T, \text{BES}} + \overline{\gamma}_T^{+} s_{T, \text{reg}}^{+} + \overline{\gamma}_T^{+} s_{T, \text{slow}}^{+} + \overline{\gamma}_T^{+} s_{T, \text{fast}}^{+} + \overline{\gamma}_T^{+} s_{T, \text{delay}}^{+}$$

$$G_T = x_{T, \text{import}} + H p_T + x_{T, \text{BES}}^{d} + \overline{\gamma}_T^{+} s_{T, \text{reg}}^{-} + \overline{\gamma}_T^{+} s_{T, \text{fast}}^{-} + \overline{\gamma}_T^{+} s_{T, \text{slow}}^{-} + \overline{\gamma}_T^{+} s_{T, \text{delay}}^{-}$$

$$U_T = G_T \quad (3.8)$$

where $H p_T$ is the energy generated during round $T$ by the PV. The parameters $\overline{\gamma}$ represent the average ratio of rounds where services are being called by the operator.

### 3.2.3 Building dynamics

The third set of constraints comes from the dynamics of the building. The thermal energy loss $X_T^{\text{loss}}$ of the thermal storage unit at time $T$ is:

$$X_T^{\text{loss}} = \left( \frac{X_T}{R_{\text{TES}}} - \theta_T \right) H \quad (3.9)$$
where $C_{TES}$ and $R_{TES}$ are respectively the thermal capacitance and resistance of the thermal energy storage unit. The temperature $\theta_T$ inside the building changes according to:

$$\theta_{T+1} = \theta_T + \frac{1}{C^T} \left( -x_{T,\text{HVAC}}\rho_{\text{HVAC}}^{\text{HVAC}} + (1 - \pi_{\text{int}}^T)\text{Int}_T + (1 - \zeta_{\text{int}}^T)\text{Sol}_T - (\theta_T - \theta_{\text{out}}^T) \frac{h}{R^T} \right) + X_{T}^{\text{loss}},$$

(3.10)

where $\text{Int}_T$ and $\text{Sol}_T$ are the internal and solar heat gains, $\pi_{\text{int}}^T$ and $\zeta_{\text{int}}^T$ are respectively their degree of ventilation, $C^T$ is the thermal capacitance, $R^T$ the thermal resistance and $\theta_{\text{out}}^T$ is the ambient temperature. We fix $\pi_{\text{int}}^T$, $\zeta_{\text{int}}^T$, $\text{Int}_T$ and $\text{Sol}_T$ to be constant parameters. This represents a summer setting when cooling is required. We note that the minus sign from the term $-x_{T,\text{HVAC}}\rho_{\text{HVAC}}^{\text{HVAC}}$ is removed if a winter setting with heating is modeled. The thermal energy inside the TES $X_T$ evolves according to:

$$X_{T+1} = X_T + x_{T,\text{EB}}\rho_{\text{EB}} - X_T^{\text{loss}} - D_{\text{DHW}}^T, \quad (3.11)$$

The energy level $B_T$ inside the battery evolves according to:

$$B_{T+1} = B_T + x_{T,BES}^c - \frac{x_{T,BES}^d}{\Phi}, \quad (3.12)$$

where $\Phi$ is the round trip efficiency of the battery energy storage.

### 3.2.4 Operational constraints

The last set of constraints represents the operational constraints. The operating limits of the electric boiler are:

$$\frac{H_{\text{EB}}\rho_{\text{EB}}}{\rho_{\text{EB}}} \leq x_{T,\text{EB}} \leq \frac{H_{\text{EB}}}{\rho_{\text{EB}}}, \quad (3.13)$$

where $\rho_{\text{EB}}$ is the coefficient of performance of the electric boiler and $P_{\text{EB}}^{\text{max}}$ and $P_{\text{EB}}^{\text{min}}$ are its maximum and minimum rated power transfer. The maximum and minimum heat exchange by the TES is defined similarly:

$$\frac{H_{\text{TES}}}{\rho_{\text{TES}}} \leq x_{T,\text{TES}} \leq \frac{H_{\text{TES}}}{\rho_{\text{TES}}}, \quad (3.14)$$

The HVAC system must operate under its limits:

$$\frac{H_{\text{HVAC}}}{\rho_{\text{HVAC}}} \leq x_{T,\text{HVAC}} \leq \frac{H_{\text{HVAC}}}{\rho_{\text{HVAC}}}, \quad (3.15)$$

where $\rho_{\text{HVAC}}$ is the coefficient of performance of the HVAC system and $P_{\text{HVAC}}^{\text{max}}$ and $P_{\text{HVAC}}^{\text{min}}$ the maximum and minimum power transfer of the cooling system. The battery energy storage is also subject to operational limits and cannot be charged and discharged at the same time:

$$0 \leq x_{T,BES}^c \leq H_{\text{BES}}\rho_{\text{BES}}^{\text{max}}$$

$$0 \leq x_{T,BES}^d \leq H_{\text{BES}}\rho_{\text{BES}}^{\text{min}} (1 - i_T),$$

(3.16)

(3.17)
where $P_{\text{BES}}^i$ and $P_{\text{BES}}^c$ are the BES maximum and minimum power transfer and $i_T \in \{0,1\}$ is the discharging or charging indicator. All energy resources must be greater than or equal to zero:

$$x_T(n) \geq 0 \text{ for } n = 1, 2, \ldots, R.$$  \hfill (3.18)

Energy import and export are limited by the grid. This leads to the following constraints:

$$x_{T,\text{import}} \leq I(1 - e_T)$$
$$x_{T,\text{export}} \leq E e_T,$$  \hfill (3.19)

where $e_T \in \{0,1\}$ for all $T = 1, 2, \ldots, \tau$ and $I$ and $E$ are respectively the network maximum imported and exported energy during a scheduling round. Finally, the building cannot curtail and increase its energy consumption at the same time. Also, if the energy demand is to be reduced, export is not allowed. Lastly, the total provided services cannot exceed the maximum curtailment or the maximum consumption increase the building can support. This translates to the following set of constraints:

$$s_{T,\text{reg}}^+ \leq M d_T$$  \hfill (3.20)
$$s_{T,\text{reg}}^- \leq M (1 - d_T)$$  \hfill (3.21)
$$s_{T,\text{fast}}^+ \leq M d_T$$  \hfill (3.22)
$$s_{T,\text{fast}}^- \leq M (1 - d_T)$$  \hfill (3.23)
$$s_{T,\text{slow}}^+ \leq M d_T$$  \hfill (3.24)
$$s_{T,\text{slow}}^- \leq M (1 - d_T)$$  \hfill (3.25)
$$s_{t,\text{delay}}^+ \leq M d_T$$  \hfill (3.26)
$$s_{t,\text{delay}}^- \leq M (1 - d_T)$$  \hfill (3.27)
$$s_{t,\text{export}} \leq E d_T$$  \hfill (3.28)
$$s_{T,\text{reg}}^+ + s_{T,\text{fast}}^+ + s_{T,\text{slow}}^+ + s_{T,\text{delay}}^+$$  \hfill (3.29)
$$s_{T,\text{reg}}^- + s_{T,\text{fast}}^- + s_{T,\text{slow}}^- + s_{T,\text{delay}}^-$$  \hfill (3.30)

where $M > 0$ is a large scalar and $d_T \in \{0,1\}$ for all $T = 1, 2, \ldots, \tau$. The upper bounds on the services are given by:

$$\bar{s}_T = \left( \min \left\{ \frac{H P_{\text{HVAC}}^i}{\rho_{\text{HVAC}}}, D_{\text{HVAC}}^i \right\} - x_{T,\text{HVAC}} \right) + \left( H P_{\text{BES}}^i i_T - x_{t,\text{BES}}^c \right)$$
$$+ \left( \min \left\{ \frac{H P_{\text{EB}}^i}{\rho_{\text{EB}}}, D_{\text{EB}}^i \right\} - x_{T,\text{EB}} \right),$$

$$\bar{s}_T^+ = \left( x_{t,\text{HVAC}} - D_{\text{HVAC}}^i \right) + x_{T,\text{BES}}^c + \left( x_{T,\text{EB}} - D_{\text{EB}}^i \right),$$
where

\[ D_{HVAC}^T = \frac{1}{\rho_{HVAC}} \left( -C^b \theta + C^b \theta_T + (1 - \pi_{int}^T) Int_T + (1 - \zeta_{pol}^T) Sol_T - \left( \theta_T - \theta_{out}^T \right) \frac{H}{Re} + C^b X_{loss}^T \right), \]

\[ D_{HVAC}^T = \frac{1}{\rho_{HVAC}} \left( -C^b \theta + C^b \theta_T + (1 - \pi_{int}^T) Int_T + (1 - \zeta_{pol}^T) Sol_T - \left( \theta_T - \theta_{out}^T \right) \frac{H}{Re} + C^b X_{loss}^T \right), \]

\[ D_{EB}^T = \frac{1}{\rho_{EB}} \left( X - X_T + D_{DHW}^T + X_{loss}^T \right), \]

\[ D_{EB}^T = \frac{1}{\rho_{EB}} \left( X - X_T + D_{DHW}^T + X_{loss}^T \right). \]

Each of these quantities is computed at the end of each round.

### 3.3 Scheduling level

In this section, we present the first-level mixed-integer linear program to schedule the building’s energy consumption. Let \( w_T \in \mathbb{R}^T \) be a weight vector. The price \( \lambda_{T,i} \) of electricity on different markets averaged over a scheduling round \( T \) is used as weight. These prices are exogenous parameters and are assumed to be available prior to the corresponding time period, e.g., computed from historical data. The weight vector \( w_T \) is given by:

\[ w_T = \begin{pmatrix} -\lambda_{T,\text{import}} & \lambda_{T,\text{export}} & \lambda_{T,\text{reg}}^+ & \lambda_{T,\text{reg}}^- & \lambda_{T,\text{fast}}^+ & \lambda_{T,\text{fast}}^- & \lambda_{T,\text{slow}}^+ & \lambda_{T,\text{slow}}^- & \lambda_{T,\text{delay}}^+ & \lambda_{T,\text{delay}}^- & 0 \end{pmatrix}^T, \]

where \( 0 \in \mathbb{R}^{R-S-2} \) is a vector of zeros. The weight vector is used to favor or disfavor energy export and ancillary services or energy import, respectively. When markets are favorable, the focus can be put on, for example, exporting energy at high prices or importing it at low costs. The entries of the weight vector corresponding to the building requirements’ or sources of flexibility’s energy variables are all zero, because requirements must always be met and are modeled as constraints. The entries of \( w_T \) corresponding to sources of flexibility like charging or discharging the battery energy storage are then an intermediary between both aforementioned types of variables. These energy consumption variables must be coordinated as part of the total energy consumption to minimize the cost of operating the building throughout the day.

The scheduling MILP to be solved at the beginning of the desired horizon is:

\[
\min_{x_T, d_T, c_T, e_T} \sum_{T=1}^{\tau} w_T^T x_T \]

subject to

\[
(3.1) - (3.30) \tag{3.31}
\]

\[
i_T \in \{0, 1\}
\]

\[
c_T \in \{0, 1\}
\]

\[
d_T \in \{0, 1\}
\]

While (3.31) is a mixed-integer program, there are only 3\( \tau \) binary variables. For example, if \( \tau = 24 \), there are 72 binary variables, which is within the tractable range of industrial solvers.
3.4 Tracking level

In this section, we first introduce the OCO with time-varying constraints framework. Based on this framework, we formulate the tracking level of our two-level algorithm. The technical analysis of our proposed algorithm for OCO with time-varying constraints is then presented. We conclude this section by stating the full two-level algorithm for MES building optimization.

3.4.1 OCO

We here describe the OCO with time-varying constraints framework. We note that this setting differs from the one introduced in Chapter 2 because some constrains are now also online parameters. Let \(i\) denote the rounds’ index and \(I\) be the time horizon. Let \(z_i \in \mathbb{Z}\) be the decision variable at round \(i\) and \(\mathbb{Z} \subseteq \mathbb{R}^n\) be set of constraints that do not vary in time. An OCO with time-varying constraints takes the form

\[
\begin{align*}
\min_{z_i \in \mathbb{Z}} & \quad f_i(z_i) \\
\text{subject to} & \quad g_{i,j}(z_i) \leq 0 \text{ for } j = 1, 2, \ldots, J \\
& \quad h_{i,k}(z_i) = 0 \text{ for } k = 1, 2, \ldots, K,
\end{align*}
\]

(3.32)

where \(f_i : \mathbb{R}^n \rightarrow \mathbb{R}\) is the differentiable convex loss function, \(g_{i,j} : \mathbb{R}^n \rightarrow \mathbb{R}\) are convex functions for all \(j = 1, 2, \ldots, J\) and \(h_{j,k} : \mathbb{R}^n \rightarrow \mathbb{R}\) are affine functions for all \(k = 1, 2, \ldots, K\), for all \(i = 1, 2, \ldots, I\).

The goal of the forecaster is to minimize their cumulative loss up to the time horizon \(I\). The information about the current round is only available to the forecaster when the round ends. Hence, at each round \(i\), the forecaster must make a decision \(z_i\) to minimize an unknown loss function while trying to satisfy unknown constraints.

3.4.1.1 Prior work on OCO with time-varying constraints

Several authors have integrated time-varying constraints into OCO by allowing constraints to be instantaneously violated, but then showing that they will be met on average [97–101]. Neely and Hao [97] proposed an OCO algorithm based on virtual queues and gave a static regret bound. The authors of [98] used a primal descent and dual ascent algorithm, the Modified Online Saddle-Point (MOSP), and showed that it achieves a sublinear dynamic regret bound. They also provided a sublinear bound on the dynamic fit which ensures that the constraints will be satisfied, at least on average, under certain necessary conditions. Reference [100] provided an alternate virtual queue-based algorithm in which the total violation is shown to be sublinear under different necessary conditions. In [102], a penalty term is added to the loss function to simplify the projection step and improve the computation time. However, only time-invariant constraints are covered in [102].

Our approach is novel in its use of a penalty in the objective, and in that it only requires standard OCO assumptions on variation of round optima.

3.4.1.2 Regret

The design tool of OCO algorithms is the regret. In this work, we consider the dynamic regret

\[
\text{Regret}^d_i = \sum_{i=1}^{I} f_i(z_i) - f_i(z^*_i),
\]

(3.33)
where
\[
z^*_t = \arg \min_{z \in \mathbb{Z}} \{ f_i(z) \text{ s.t. } g_{i,j}(z) \leq 0 \forall j, h_{i,k}(z) = 0 \forall k\},
\]
the round optimal decision in hindsight. Alternatively, in Chapter 2 and other works the static regret is used as the design tool. The static regret uses the best-fixed decision in hindsight as the comparator in the second term of the sum in (4.3). A bounded dynamic regret implies a bounded static regret and for this reason, we only use the former. When designing an OCO algorithm, the goal is to obtain, in the worst-case, a sublinear regret in the number of rounds. A sublinear dynamic regret implies that, on average, the algorithm plays the round optimal decision in hindsight at each round 24,26.

### 3.4.2 OCO-based tracking level

We now introduce the second level of our building management algorithm. We present our OCO-based algorithm for tracking the energy trajectory set by the scheduling MILP of Section 3.3.

Let \( z_t \in \mathbb{R}^{R+S} \) for \( t = 1, 2, \ldots, F \) at all scheduling round \( T \) be the tracking decision variable. The variable \( z_t \) has the same structure as \( x_T \) for the \( R \) first entries. The variable \( u_t \) represents the services that are accepted and uncalled by the operator. It is analogous to \( s_T \) for the tracking rounds. Similarly, we use \( q_t \) as the tracking round analog of \( p_T \) for the PV generation. Let \( y_t \in \mathbb{R}^S \) be the services dispatched after being called at round \( t \) for \( t = 1, 2, \ldots, F \). The component of \( y_t \) is organized as in the \( u_t \) vector. The vector \( y_t \) makes the last \( S \) entries of \( z_t \). Finally, we define \( \bar{x}_T \) as,

\[
\bar{x}_T = \left( x_T \ s_{T,\text{reg}}^+ \ s_{T,\text{reg}}^- \ s_{T,\text{fast}}^+ \ s_{T,\text{fast}}^- \ s_{T,\text{slow}}^+ \ s_{T,\text{slow}}^- \ s_{T,\text{delay}}^+ \ s_{T,\text{delay}}^- \right)^T
\]

for \( T = 1, 2, \ldots, \tau \).

The objective function \( o_t(\bar{x}_T, z_t) \) of our tracking algorithm is:

\[
o_t(\bar{x}_T, z_t) = \sum_{r=1}^{R+S} a_{r,t} \left( p_{r,t}(\bar{x}_T(r)) - z_t(r) \right)^2,
\]

for all \( t = 1, 2, \ldots, F \) and for \( T = 1, 2, \ldots, \tau \). In (3.34), \( a_{r,t} \) is the tracking weight and \( p_{r,t} : \mathbb{R} \rightarrow \mathbb{R} \) is a scaling function for the requirement or service \( r \) at time \( t \). For \( r = R+1 \) to \( R+S \), \( p_{r,t} \) also indicates if the service is called or not. The function \( p_{r,t} \) is used to scale the scheduling objective to the tracking time scale. In our case study, we use

\[
p_{r,t}(\bar{x}_T(r)) = \begin{cases} \frac{\tau_t(r)}{f} & \text{for } r = 1, 2, \ldots, R \\ \gamma_{r,t}^+ \tau_t(r) & \text{for } r = R+1, R+2, \ldots, R+S, \end{cases}
\]

where \( \gamma_{t,r}^+ \in \{0,1\} \) indicates if a service is called (\( \gamma = 1 \)) or not (\( \gamma = 0 \)). We use a weighted squared tracking error to penalize large deviations from the scheduling objective. Equivalently, we express the objective function as,

\[
o_t(\bar{x}_T, z_t) = \| p_t(\bar{x}_T) - z_t \|_{A_t}^2,
\]

where \( A_t = \text{diag}(a_{r,t} \text{ for } r = 1, 2, \ldots, R + S) \) and \( \| \cdot \|_{A_t}^2 = v^T A_t v \) is the norm induced by the matrix \( A_t \). Note that \( o_t \) is strongly convex if \( A_t \) is positive definite. In the case study, we set the tracking
weight matrix $A_t$ to

$$A_t = \text{blockdiag}[I + \text{diag}(w_T) - \text{diag}(	ilde{w}_t), \alpha I + \text{diag}(w_T[2 : 8]) - \text{diag}(	ilde{w}_t[2 : 8])]$$  \hspace{1cm} (3.36)

for $t = 1, 2, \ldots, F$, at each $T$ and where $\alpha$ is a tuning parameter. It can be used to decrease the response time to a call signal by setting it larger than 1. When $\alpha$ is set to values higher than one, the weight $a_{r,t}$ in the loss function (3.34) for the called services $y_t$ will dominate the other weights and hence will lead quicker response to called services because the absolute value of gradient will take larger values.

In (3.36), $w_T$ represents the average price used in the scheduling MILP and $\tilde{w}_t$ represents the current prices. The parameter $w_T$ is known prior the the tracking rounds while $\tilde{w}_t$ is observed after the tracking round. We assume that $w_T(r) - \tilde{w}_t(r) > -1$ for all $r$, and thus $A_t$ is positive definite for all $t$. This matrix is used to prioritize or deprioritize services that respectively have a higher or lower value at time $t$ than the average used for scheduling.

Let $\mathcal{Z}_T$ be the set defined by the union of (3.1) – (3.3) and (3.13) – (3.28), where $x_T$ and $H$ have been replaced by $z_t$ and $h$, respectively. Note that the binary variables $i_T$, $e_T$ and $d_T$ have already been fixed by the first-level algorithm and thus the parameters of $\mathcal{Z}_T$ do not change over a scheduling round $T$. The tracking problem is also subject to time-dependent constraints, which we adapt from Section 3.2:

$$D_{t,\text{HVAC}} \leq z_{t,\text{HVAC}} \leq \overline{D}_t, \hspace{1cm} (3.37)$$

$$z_{t,\text{baseload}} \geq D_{t,\text{baseload}}, \hspace{1cm} (3.38)$$

$$z_{t,EB} + z_{t,\text{TES}} \geq \overline{D}_t, \hspace{1cm} (3.39)$$

$$q_t \leq PV_t, \hspace{1cm} (3.40)$$

$$u_{t,\text{reg}} + u_{t,\text{fast}} + u_{t,\text{slow}} + u_{t,\text{delay}} \leq \overline{u}_t, \hspace{1cm} (3.41)$$

$$u_{t,\text{reg}} - u_{t,\text{fast}} - u_{t,\text{slow}} - u_{t,\text{delay}} \leq \overline{u}_t, \hspace{1cm} (3.42)$$

$$y_{t,\text{reg}} + y_{t,\text{fast}} + y_{t,\text{slow}} + y_{t,\text{delay}} \leq \overline{y}_t, \hspace{1cm} (3.43)$$

$$y_{t,\text{reg}} - y_{t,\text{fast}} - y_{t,\text{slow}} - y_{t,\text{delay}} \leq \overline{y}_t, \hspace{1cm} (3.44)$$

where $\overline{u}_t$ and $\overline{y}_t$ are defined similarly to $\overline{\pi}_T$ and $\overline{\pi}_T^r$. Finally, $D_{t,\text{baseload}}$, $D_{t,\text{DHW}}$ and $PV_t$ are fixed values observed at the end of a round. The parameter $PV_t$ is the tracking round analog of $\overline{p}_t$. 
The tracking problem during the scheduling round $T$ is

$$\min_{z_t \in \mathbb{Z}_T} \sum_{t=1}^{F} \left\| p_t(x_T) - z_t \right\|_A^2,$$

subject to \([3.37] - [3.48]\) for all $t = 1, 2, \ldots, F$.

$$U_t = z_{t,\text{export}} + z_{t,\text{HVAC}} + z_{T,\text{baseload}} + z_{t,\text{EB}}$$

$$+ z_{t,\text{BES}}{y_{t,\text{reg}}} + y_{t,\text{slow}} + y_{t,\text{fast}} + y_{t,\text{delay}}$$

$$G_t = z_{t,\text{import}} + q_t + z_{t,\text{BES}} + y_{t,\text{reg}} + y_{t,\text{fast}}$$

$$+ y_{t,\text{slow}} + y_{t,\text{delay}}$$

$$U_t = G_t$$

for all $t = 1, 2, \ldots, F$.

where the information for round $t$ is only observed when the current round ends. We solve this problem using the OGD with exact penalty presented in Corollary 3.1 of Section 3.4.3. The exact penalty function for the tracking algorithm is provided in \([3.49]\) in which $[b]^+ = \max\{0, b\}$ and the indicator function $\mathbb{I}_b = 1$ if $b$ is true and $\mathbb{I}_b = 0$ otherwise.

$$P_t(z_t) = \left[D_t^{\text{HVAC}} - z_{t,\text{HVAC}} \right]^+ + \left[z_{t,\text{HVAC}} - \overline{D}_t^{\text{HVAC}} \right]^+ + \left[D_t^{\text{baseload}} - z_{t,\text{baseload}} \right]^+ + \left[q_t - P_V \right]^+$$

$$+ \left[D_t^{\text{DHW}} - z_{t,\text{DHW}} \right]^+ + \left[z_{t,\text{DHW}} - \overline{D}_t^{\text{DHW}} \right]^+$$

$$+ \left[u_{t,\text{reg}} + u_{t,\text{fast}} + u_{t,\text{slow}} + u_{t,\text{delay}} - \overline{u}_t \right]^+$$

$$+ \left[y_{t,\text{reg}} + y_{t,\text{fast}} + y_{t,\text{slow}} + y_{t,\text{delay}} - \overline{y}_t \right]^+$$

$$+ \mathbb{I}_{X_{t+1}} | z_{t,\text{EB}} | + \mathbb{I}_{X_{t+1}} | z_{t,\text{BES}} |.$$  \(\text{(3.49)}\)

The OCO with exact penalty is given by

$$\min_{z_t \in \mathbb{Z}_T} \left\| p_t(x_T) - z_t \right\|_A^2 + cP_t(z_t),$$  \(\text{(3.50)}\)

where $c > 0$ is set according to Theorem 3.1. We denote the objective function of \((3.50)\) as $f_{T,t}(x_T, z_t)$. Note that because the first term of $f_{T,t}(x_T, z_t)$ is strongly convex, so is $f_{T,t}(x_T, z_t)$.

For the OGD with exact penalty update, we use a subgradient of the loss function. A subgradient of the penalty function, $P_t'$, is given by:

$$P_t' = \sum_{j=1}^{J} \nabla g_j(z_t) \left\| g_j(z_t) > 0 \right\| + \sum_{k=1}^{K} \text{sign}(h_k(z_t)) \nabla h_k(z_t).$$

The following is then a subgradient of the loss function:

$$\nabla f_{T,t}(x_T, z_t) = 2A_t(z_t - p_t(x_T)) + cP_t'.$$

### 3.4.3 Theoretical analysis of OCO with exact penalty

The OCO with exact penalty framework is a minor extension of existing frameworks for OCO with time-varying constraints. In this subsection, we prove that our algorithm for the OCO with exact penalty
Chapter 3. OCO of Multi-energy Building-to-grid Ancillary Services

framework achieves sublinear regret.

We first introduce some notation and the assumptions we will be using to in this work. We define the cumulative variation term $V_I$ and cumulative constraint variation term $V^g_I$ as:

$$V_I = \sum_{i=2}^{I} \|z^*_i - z^*_{i-1}\|,$$

$$V^g_I = \sum_{i=2}^{I} \max_{z \in \mathcal{Z}} \|g_i(z) - g_{i-1}(z)\|,$$

where $[y]^+$ returns the vector $v \in \mathbb{R}^n$ were $v(m) = \max\{0, y(m)\}$ for $m = 1, 2, \ldots, n$. These terms represent the variation in the optimal solutions and the changes of the feasible set size. We don’t make use of the constraint variation term in our analysis. We only define it for future comparison with the literature.

We now list our assumptions. These assumption are standard in the OCO literature [24,95].

**Assumption 3.1.** The decision variable $z \in \mathcal{Z}$, where $\mathcal{Z}$ is a convex and compact set.

**Assumption 3.2.** The loss function is $B$-bounded: $\|f_i(z)\| \leq B$ for $i = 1, 2, \ldots, I$.

**Assumption 3.3.** The gradient of the loss function is $G$-bounded: $\|\nabla f_i(z)\| \leq G$ for $i = 1, 2, \ldots, I$.

As a consequence of Assumption 3.1, the decision variable is also $X$-bounded: $\|z_i\| \leq Z$ for $i = 1, 2, \ldots, I$.

We show that using a $\ell_1$-exact penalty, we can solve an OCO problem with a time-varying feasible set without the need to characterize its variation.

**Definition 3.1 ($\ell_1$-exact penalty function).** Let $P : \mathbb{R}^n \mapsto \mathbb{R}$ be the $\ell_1$-exact penalty function defined as:

$$P_i(z_i) = \sum_{j=1}^{J} |g_{i,j}(z_i)|^+ + \sum_{k=1}^{K} |h_{i,k}(z_i)|,$$

We first shown the following lemma and then use it in the main technical result of Theorem 3.1.

**Lemma 3.1 (Penalized problem).** Suppose that the constrained program (3.32) is primal and dual feasible at time $i$ and let $z^*_{i,\text{constrained}}$ be its the optimum. Define the penalized optimization problem as

$$\min_{z_i \in \mathcal{Z}} f_i(z_i) + cP_i(z_i),$$

and denote its optimum by $z^*_{i,\text{penalized}}$. If $f_i$ is strongly convex and $c > \left\|\left(\lambda^*_i \nu^*_i\right)^\top\right\|_\infty$ where $\nu^*_i$ and $\lambda^*_i$ are the optimal dual variables of the constrained program (3.32), then $z^*_{i,\text{penalized}} = z^*_{i,\text{constrained}}$.

**Proof.** Because $f_i$ is strongly convex, the objective function of (3.33) is also strongly convex and thus strictly convex. Therefore, the set of minimizers of (3.33) is a singleton. It then follows from Theorem 17.3 that $z^*_{i,\text{penalized}} = z^*_{i,\text{constrained}}$. \qed

**Theorem 3.1 (OCO with $\ell_1$-exact penalty).** Suppose that the original constrained program (3.32) is primal and dual feasible. Consider the penalized optimization problem (3.1) at round $i$ where $f_i$ is strongly convex for all $i$. Suppose that $c > \left\|\left(\lambda^*_i \nu^*_i\right)^\top\right\|_\infty$ for all $i$. Then, a sublinearly bounded regret
OCO algorithm for time-invariant feasible set applied on (3.53) achieves a regret bound of the same order. Thus, the algorithm plays, on average, the round optimal decision and plays, at least on average, a feasible decision with respect to the original problem, (3.32).

Proof. Let \( A_Z \) be an OCO algorithm for time-invariant feasible set \( Z \) such that \( \text{Regret}_I(A_Z) < O(I) \). First, observe that the penalized optimization problem (3.53) has a time-invariant feasible set. We can therefore apply \( A_Z \) to sequentially solve (3.53). Denote by \( A^p_Z \) the algorithm \( A_Z \) applied to the penalized problem (3.53).

Let \( z^*_i, \text{penalized} \) be the optimum of (3.53) and \( z^*_i, \text{constrained} \) be the solution of (3.32) at time \( i \). The regret of \( A^p_Z \) is given by,

\[
\text{Regret}_d(I) = \sum_{i=1}^{I} f_i(z_i) - f_i(z^*_i, \text{penalized}).
\]

By Lemma 3.1 we have \( z^*_i, \text{penalized} = z^*_i, \text{constrained} \) for all \( i \). We can re-express (3.54) as

\[
\text{Regret}_d(I) = \sum_{i=1}^{I} f_i(z_i) - f_i(z^*_i, \text{constrained}) < O(I),
\]

where the last inequality follows from the assumption \( \text{Regret}_I(A_Z) < O(I) \). Consequently, since \( \text{Regret}_I(A_Z) \to 0 \) as \( I \) increases, the forecaster plays, at least on average, the optimum of the time-varying constrained problem (3.32). By definition, the optimum of (3.32) is feasible and it follows that the decision played by \( A^p_Z \) will be, at least on average, a feasible point of the time-varying constraints.

Corollary 3.1 (OGD with exact penalty). Suppose we apply the online gradient descent algorithm (OGD) \[23\] to (3.53) and that \( f_i \) and \( c \) satisfy the assumptions of Theorem 3.1. Then

\[
\text{Regret}_d(I) \leq \left( \frac{7Z^2}{4\chi} + \frac{\chi C^2}{2} + \frac{ZV_I}{\chi} \right) \sqrt{I},
\]

where \( \chi \) is a positive scalar. Equivalently we have \( \text{Regret}_d(I) \leq O \left( \sqrt{I} (1 + V_I) \right) \).

Proof. The result follows from applying OGD with \( \eta = \chi I^{-1/2} \) to (3.53).

Corollary 3.1 implies a stricter constraint on \( V_I \) when compared to the MOSP algorithm [98]. We note that any OCO algorithm can be used in pair with the exact penalty of Theorem 3.1. Thus using a different algorithm may lead to a looser constraint on \( V_I \) and an improved bound.

Our approach has two main advantages. First, it does not constrain the set variation \( V_I^R \), unlike [98]’s and [100]’s algorithms. With our approach, \( V_I \) suffices to capture the variation of the time-varying feasible set and round optimum. The second gain is that the algorithm is not constrained by the dynamic fit for the constraint satisfaction [97, 98, 101]. As showed in Theorem 3.1 the penalty-based update directly ensures that the constraints are satisfied when the regret is sublinearly bounded above.

### 3.4.4 Two-level algorithm

The full two-level algorithm is given in Algorithm 3.1.
Algorithm 3.1 Two-level MES building real-time optimization algorithm

1: **Parameters:** Historical data/predictions for $w_T$ and (3.1) – (3.30), $\chi > 0, \eta = \chi/\sqrt{T}$ and $c$.

**Level 1:**
2: Solve Scheduling MILP:

$$\{x_T, i_T, e_T, d_T\}_{T=1}^{\tau} = \arg\min_{x_T, i_T, e_T, d_T, \quad T=1,2,\ldots,\tau} w_T^T x_T$$

subject to (3.1) – (3.30),

\begin{align*}
i_T &\in \{0,1\} \\
e_T &\in \{0,1\} \\
d_T &\in \{0,1\}
\end{align*}

**Level 2:**
3: for $T = 1,2,\ldots,\tau$ do

#Track the energy objectives $x_T$ using OCO with exact penalty:

4: Initialization: $z_0 = 0$

5: for $t = 1,2,\ldots,F$ do

6: Schedule the building energy usage and services according to $z_t$.

7: Observe outcome of the round.

8: Calculate the gradient $\nabla f_{T,t}(x_T, z_t)$.

9: Compute $z_{t+1} = \text{proj}_{Z_T}(z_t - \eta \nabla f_{T,t}(x_T, z_t))$.

10: Dispatch $z_{t+1}$.

11: end for

12: end for

3.5 Case study

We apply the two-level algorithm to a case-study based on a terraced building in Melbourne, Australia. We use CVXPY \[104\] with the Gurobi solver \[76\] for the scheduling MILP, and with the ECOS \[105\] solver for projection steps of the OCO algorithm.

3.5.1 Setting

The building’s parameters are listed in Table 3.3. The electric baseload demand and domestic hot water demand are based on United Kingdom data \[88,106,107\]. The ambient temperature and solar irradiance are taken from \[111\] for a Summer day for the region of Melbourne, Victoria, Australia. The electricity retail import and export prices of the state of Victoria are from \[112\]. The FCAS prices are taken form \[113\]. Let $H = 1$ hour, $\tau = 24$, $h = 0.5$ minute and $F = 120$. We set $\theta_0 = 21^\circ$C, $X_0 = 15$ kWh and $B_0 = 6.75$ kWh. We set the numerical parameter $\chi$ to 0.1 and $\alpha = 5$. We compute the penalty constant using Monte Carlo simulations and set $c = 3.0$. We linearly interpolate all demand, environment and price data to compute their respective values for each 0.5 minute rounds.

All time-varying data are subject to Gaussian noise to model uncertainty. Truncated Gaussian variables with standard deviations computed from historical data are used for all parameters except for prices where the standard deviation is set to 0.01, i.e., the equivalent of $\$0.01$. The decision $\gamma_{t,s}$ of an operator to call a service or not is modeled by independent and identically distributed Bernoulli random
variables with probability $\beta$ for each service and each tracking rounds. Note that $\gamma_{t,s} = 1$ to indicate if a service is called and 0 otherwise. The value of the decision is kept to 1 for the total duration of the service, e.g., for 10 rounds in the case of contingency slow services, after which it is re-sampled. In the case of delay contingency, it is kept to 1 for 10 minutes and then re-sample. We set $\beta = 0.2$ and $\gamma_{T,s} = 0.4$ for all services. To emphasize on the adaptability of the real-time level, $\gamma_{T,s}$ is set to differ from the actual average ratio of rounds the service are called.

Because the tracking rounds in our case study are 0.5 minute, we only consider slow contingency services. For fast contingency services, the tracking round size must be decreased below ten seconds as the response time of the loads must be within six seconds [89].

### 3.5.2 Numerical results

#### 3.5.2.1 Scheduling level

First, we give an example of the output of the first level of the algorithm for a day. The plan produced by the scheduling level has a net revenue of $4.60, including any expenses due to electricity import. The
Chapter 3. OCO of Multi-energy Building-to-grid Ancillary Services

Figure 3.2: Scheduled energy resources distribution for a day

energy used for requirements and services are presented in Figures 3.2 and 3.3 respectively. A discussion of the outcome is provided in the following section.
3.5.2.2 Two-level algorithm example

We now discuss the output of the two-level algorithm. The real-time energy dispatch is presented on Figure 3.4. The real-time called and uncalled services are presented on Figure 3.5. We note the similarities of the real-time energy dispatch with the scheduled energy consumption of Figures 3.2 and 3.3, which highlights the performance of our two-level algorithm. The net revenue of the building under uncertainty is $2.13. The discrepancy between the planned and observed net revenue can be explained by the major difference between the mean data used for the scheduling level and the real-time level.

In Figures 3.4 and 3.5 and all the following figures, the curves are not smooth. There are two reasons for this. First, the algorithm has to deal with services being intermittently called and stopped. Each time a service is called or stopped, the algorithm has to re-balance the building’s generation and consumption. For example, this can be due increase consumption contingency service being called for 5 minutes. The algorithm can then increase the energy sent to the electric boiler or BES charging while ensuring all constraints are met, leading to jumps in $z_{t,EB}$ and $z_{t,BES}$. This is also observed when an MPC tracking algorithm is implemented (see Figure 3.9 of Section 3.5.3). The second reason for the non-smoothness of the curves is the penalty function algorithm. Each time the prediction violates a constraint, the prediction is pushed back to the interior of the feasible set leading to another local peak. This again requires re-balancing of the buildings generation and consumption.

In Figure 3.6, the temperature, BES energy level and TES energy level are shown. We remark that these constraints are satisfied at almost all time. Figure 3.6a shows that the two-level algorithm maintains low temperature and pre-cools the unit before the high ambient temperature. At this time, it is at the mid-point of its temperature dead-band and can therefore bid on increased consumption contingency services. This has a two-fold benefit: (i) it increases the income of the unit and (ii) when called, the algorithm services energy to the HVAC unit. This process is also repeated later during the day. This can be seen by the high values of the HVAC curve in Figures 3.4a and 3.5 for the called and
uncalled ancillary services. The algorithm exploits the lower import rate at the beginning of the day to balance its energy consumption when no ancillary services are offered.
The algorithm uses the same contingency service to charge its BES and TES (via the electric boiler) shown on Figures 3.6b and 3.6c. Charging the BES is the main source of flexibility at the end of the day because the TES and temperature are close to their upper bounds. The BES can be discharged at most times because it does not rely on exogenous phenomena like hot water demand. For this reason, the BES cycles through its full capacity. When it reaches a high state of charge, the algorithm exports the energy in the BES as seen in Figure 3.4d. When solar energy is available to the building, we observe large energy export. Around midday, the export is higher than the discharge as the solar energy can satisfy the building’s requirements.

Lastly, we show how the real-time algorithm is able to meet the MES requirements. Figure 3.7 shows the upper limits and dispatch of the PV, domestic hot water demand, and baseload demand for scheduling rounds 11 to 14. Figure 3.7a shows that the solar power generated by the PV is almost always below the maximum generation. Figure 3.7b shows that the energy used by the electric boiler added to the energy taken out of the TES nearly always satisfy the domestic hot water requirement. Finally, Figure 3.7c shows that the energy dispatched to the baseload also almost always exceeds the baseload requirements.

Finally, we run the simulation over a 45-day data set and compute the ratio of rounds in which the time-varying constraints are satisfied and the net revenue after each day. Table 3.4 presents the ratio of tracking rounds in which the specified time-varying constraint is satisfied over the total number of tracking rounds. The overall average time-varying constraint satisfaction rate is 97.56%. These results show the high performance of OCO in uncertain, dynamic settings. The tracking rounds have a very small duration (30 seconds) and hence the 2.44% of tracking rounds with constraint violations is a relatively short time period. Because the OCO algorithm penalizes strongly decision variables that lead to constraint violations, the round with violated constraints should be distributed over time. This is observed in Figure 3.7, where constraint violation for very few consecutive rounds, which are evenly spaced throughout the tracking rounds.

Figure 3.8 shows the net revenue of the building for each different day. The average net revenue is $216. This is a significant improvement as the MES building net revenue would be on average −$1.62 if no energy management was done. This shows that under unknown and noisy exogenous factors like the
natural phenomena and domestic requirements, our proposed two-level algorithm can efficiently manage the energy resources of building-based MES.

3.5.3 Discussion: model predictive control

MPC shares some common features with OCO and could also be used in the real-time tracking level. The MPC framework assumes knowledge of future parameters, which it uses in the receding horizon computation. On the other hand, the OCO relies only on past and present information. In our two-level algorithm, only approximate information about the future is used in the scheduling level. This makes our approach amenable to a wider range of settings, e.g., outside of major metropolitan regions where extensive data may not be available.

MPC typically outperforms OCO because an optimization problem is fully solved in each round. Note that regret bounds of OCO algorithms ensure that the OCO predictions are round optimal at least on average. The trade-off is that MPC is more computationally demanding because it must solve a multi-period optimization problem in each round. In comparison, OCO only requires an algebraic
Figure 3.7: Energy usage and constraints between scheduling round 11 and 14

For comparison purposes, we now discuss how MPC could be applied to our problem. Let $\mu \in \mathbb{N}$ be the length of the receding horizon. At each tracking round $t$, we solve the receding horizon quadratic program:

$$
\min_{z_{t+k}} \sum_{k=1}^{\mu} \|p_{t+k}(x_T) - z_{t+k}\|_2^2
$$

subject to (3.1) – (3.30) at $t + k$ for $k = 1, 2, \ldots, \mu$

$\theta_t, B_t, X_t, X_t^{loss}$ are the observed state at $t$

+ models for $\theta_t^{out}, D_t^{\text{base load}}, D_t^{DHW}, PV_t, \lambda_t$,

and implement the decision $z_{t+1}$.

Simulations run using MPC lead to slightly stronger performance than our two-level algorithm. A
Table 3.4: Ratio of tracking rounds in which time-varyings constraint are satisfied over the total number of tracking rounds (tolerance $10^{-8}$)

<table>
<thead>
<tr>
<th>Constraints</th>
<th>Satisfaction rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td>99.53%</td>
</tr>
<tr>
<td>BES</td>
<td>96.55%</td>
</tr>
<tr>
<td>TES</td>
<td>99.74%</td>
</tr>
<tr>
<td>PV</td>
<td>92.61%</td>
</tr>
<tr>
<td>DHW</td>
<td>99.39%</td>
</tr>
<tr>
<td>Baseload</td>
<td>97.53%</td>
</tr>
</tbody>
</table>

Figure 3.8: Building’s net revenue for different day

The comparison between MPC with $\mu = 150$ rounds and OCO is shown in Figure 3.9 for several energy resource variables. Note that the two-level algorithm was re-run in the case of OCO and thus different time-varying data (e.g. temperature) were observed because of the uncertainty. Because MPC solves an optimization problem at each time step, it meets the scheduled objective more quickly. MPC also makes use of more future information than OCO, and hence has better foresight. The main advantage of OCO over MPC is scalability. For example, running the two-level algorithm with OCO takes on average 2.70 minutes compared to 20.43 hours when MPC with $\mu = 150$ is used. In this case, MPC requires a factor of 454 more computation time than OCO. Hence, on average, the MPC update requires 25.5 seconds to compute the energy dispatch at each tracking round as opposed to 0.05 seconds for the OCO algorithm. The average MPC computation time is almost as long as the duration of the round. In 13.36% of cases the MPC computation took more time than the length of the round. Thus, in these cases the operator would not have an implementable decision. For this reason, OCO is more suitable for real-time energy dispatch of buildings.
3.6 Conclusion

We have proposed a two-level algorithm to manage MES under uncertainty. We model a building as an MES with multiple sources of flexibility, energy requirements and access to several electricity markets. We use MILP to schedule hourly energy consumption using mean data about markets and natural phenomena for each hour. We then use OCO to track this consumption objective while facing uncertainty. The performance of our two-level algorithm is assessed in a case study based on a terraced house in Melbourne, Australia. In simulations over a 45-day data set, our approach attained an average net revenue of $2.16 each day improving over a balance of $1.62 when no energy management is done.

We have shown that a building-based MES can use its flexibility to provide ancillary services. This benefits both the building by increasing its revenue and the grid by contributing to its reliability and grid efficiency. In future work, we will extend our approach to aggregations of multiple buildings providing ancillary services in a network.
Chapter 4

Predictive Online Convex Optimization

4.1 Introduction

Online convex optimization (OCO) has found applications in fields like network resource allocation and demand response in power systems. It is used for sequential decision-making when contextual information or feedback is only revealed to the decision maker at the end of the current round. Theoretical results showing that OCO algorithms have bounded regret guarantee the performance of these algorithms under mild assumptions.

In many applications, the decision maker has access to both revealed past information and estimates about the future rounds. For example, in power systems, weather forecasts or historical load patterns can be used to estimate the future regulation needs. In this work, we present the predictive online convex optimization (POCO) framework. POCO works under the assumption that an estimate of the gradient of the loss function for the next round is available to the decision maker. We introduce explicit criteria for determining if the quality of the estimated gradient is high enough to guarantee an improvement in which case a predictive update is applied after the OCO step. A regret bound is obtained for all our algorithms. We conclude this work by presenting numerical examples where a POCO algorithm is used to improve on the performance of demand response with standard OCO. This example is motivated by the fact that a load manager often has access to an estimate of the power imbalance they have to counteract for regulation purposes.

Literature review. Recent work in online convex optimization has focused on including prior or side information. Reference builds on , assumes that the problem’s unknown and uncertain parameters follow a predictable process plus some noise for their OCO algorithm. As in our setting, a second update with an estimated gradient-like term follows a mirror descent update. This second update is used by the algorithm in every step regardless of the quality of the estimated gradient. For this reason, the algorithm is referred to optimistic. Optimistic algorithms were also studied in. No conditions are provided about the estimated gradient in this case except that it comes from past observations and/or side information via an oracle. The authors of show that the optimistic mirror descent can lead to a tighter bound than a standard online mirror descent algorithm if the process is indeed predictable. If this is not the case, in the worst case, the bound on the static regret achieved by
the optimistic mirror descent is the same as the online mirror descent. In [120], the authors provide a
dynamic regret bound for the optimistic mirror descent. There is, however, no guarantee that in a given
round the optimistic update does not do worse than the standard OCO update. An algorithm similar
to [114] is given in [121]. In their work, they make the stronger assumption in which the exact gradient
of the next round loss function is available and then provide a static regret bound for their setting. This
differs from our setting in that we provide dynamic regret-bounded algorithms and use an estimated
gradient which entails less restrictive assumptions. In [122], OCO with receding horizon algorithms are
presented in a dynamic regret setting. The exact gradient of multiple future rounds is required in [122]
and thus this work differs form ours. Several other authors have studied different ways to incorporate
side information in OCO like using state information [123] or the direction of the loss function’s gradient
in an online linear optimization setting [124].

Because we characterize conditions under which the predictive step improves performance, we guaran-
tee improvement over conventional OCO and require no predictability assumptions. In sum, in this
work we make the following contributions:

- We introduce a novel predictive online convex optimization framework and provide conditions for
  when to use side information.
- We propose two types of predictive updates with a predetermined step size for loss functions that
  have a Lipschitz gradient. We show that this update leads to a strict improvement over an OCO
  update when used (Section 4.4).
- We give a predictive update with backtracking line search that applies to a broader family of
  problems. We show that it leads to strict improvement over an OCO update (Section 4.5).
- We obtain sublinear regret bounds in the number of rounds for all algorithms.
- We apply our framework to demand response in power systems and find that it outperforms a
  standard OCO algorithm (Section 4.6).

4.2 Background

Let \( t \) denote the current round index and \( T \) be the time horizon. Let \( \mathcal{X} \) be the decision set and let
\( x_t \in \mathcal{X} \) be the decision variable at time \( t \). The decision set \( \mathcal{X} \) represents all constraints on \( x_t \). In this
version of OCO, only time-independent constraints are considered. We denote the differentiable convex
loss function by \( f_t(x_t) \) for \( t = 1, 2, \ldots, T \). We denote the projection operator onto the set \( \mathcal{Y} \) as \( \text{proj}_\mathcal{Y}(\cdot) \).

We recall from previous chapters that the goal of the decision maker is to sequentially solve the
following sequence of problems:

\[
\min_{x_t \in \mathcal{X}} f_t(x_t) \tag{4.1}
\]

for \( t = 1, 2, \ldots, T \). The decision maker observes the loss function \( f_t \) after choosing \( x_t \). For this reason,
even if the loss function has a simple form, an analytical solution to the round optimization problem (4.1)
is not obtainable.

The decision \( x_t \) is computed using an update rule. For example, using the online gradient descent
(OGD) [23] algorithm, the decision at round \( t + 1 \), \( x_{t+1} \), is given by the update:

\[
x_{t+1} = \text{proj}_{\mathcal{X}}(x_t - \eta \nabla f_t(x_t)) , \tag{4.2}
\]
where $\eta \propto T^{-1/2}$.

Throughout this work, we assume that Assumptions 3.1–3.3 hold. We now restate these assumptions using this chapter’s notation.

**Assumption 4.1.** The decision set $X$ is convex and compact.

**Assumption 4.2.** The loss function is $B$-bounded: $|f_t(x)| \leq B$ for $t = 1, 2, \ldots, T$ and $B > 0$.

**Assumption 4.3.** The gradient of the loss function is $G$-bounded: $\|\nabla f_t(x)\| \leq G$ for $t = 1, 2, \ldots, T$ and $G > 0$.

We recall that Assumption 4.1 implies that the decision variable is also $X$-bounded: $\|x\| \leq X$ for $t = 1, 2, \ldots, T$ and recall that the diameter of the compact set $X$ is defined as

$$\text{diam } X = \sup \left\{ \|x - y\| \mid x, y \in X \right\}$$

and let $D = \text{diam } X$, a positive scalar. The remainder of the assumptions will be stated when a specific technical result requires it.

The design tool of OCO algorithms is the regret [24, 95]. Similarly to Chapter 3, we use the dynamic regret [23, 26, 98, 120] which we now recall:

$$\text{Regret}_T^d = \sum_{t=1}^{T} f_t(x_t) - f_t(x_t^*),$$

(4.3)

where $x_t^* = \arg\min_{x \in X} f_t(x)$. We design performance-guaranteed POCO algorithms by upper bounding their regret.

We conclude by recalling from the previous chapter the variation term:

$$V_T = \sum_{t=2}^{T} \|x_t^* - x_{t-1}^*\|.$$  

The term $V_T$ quantifies the variation of the optimal predictions through all rounds.

### 4.3 Predictive OCO

We now introduce our POCO framework. We let $x_{t+1} \in X$ be the decision computed by an OCO algorithm in round $t$. This OCO algorithm can be, for example, the aforementioned OGD. The decision $x_{t+1}$ is then given by the update (4.2). In our predictive setting, we consider an $\epsilon$-forecaster as introduced in Definition 4.1. Let $g_t(x_{t+1}) \in \mathbb{R}^N$ be the estimated gradient of the loss function $f_{t+1}$ at $x_{t+1}$.

**Definition 4.1** ($\epsilon$-forecaster). An $\epsilon$-forecaster has access to an estimated gradient, $g_t(x_{t+1})$, such that $\|g_t(x_{t+1}) - \nabla f_{t+1}(x_{t+1})\| \leq \epsilon$ where $\epsilon$ is a positive scalar, for $x_{t+1} \in X$ and all time $t$.

In other words, we consider a forecaster that has access to inexact information about the next round, the estimated gradient $g_t(x_{t+1})$, in addition to the standard OCO assumptions. For conciseness, we denote the estimated gradient by $g_t$. We omit its dependency on $x_{t+1}$ because it is always evaluated at the OCO update output, $x_{t+1}$, and no other points. The decision maker can meet Definition 4.1 by relying on an exogenous model to estimate the gradient $\nabla f_{t+1}(x_{t+1})$. In the context of demand
response, historical data of the load’s consumption and generator output’s patterns, weather history and
the historical values of the gradient, for example, can be used to build a statistical model to estimate
the value of the $\nabla f_{t+1}$ at the decision given by OCO update. The parameter $\epsilon$ can then be set according
to, for example, a high confidence interval or a worst-case performance parameter. The forecaster would
then provide $g_t$ using this model.

Then, if certain conditions are met, the following update rule for our proposed POCO algorithm is
used.

**Definition 4.2 (Predictive update).** Let $\beta_t > 0$ be an appropriately chosen step size. The predictive
update is

$$x_{t+1} = \text{proj}_X (x_{t+1} - \beta_t g_t).$$

(4.4)

The predictive update is to be used directly after the OCO update and will lead to a strict improve-
ment over the OCO update under certain conditions. The aforementioned conditions will be discussed
in the next sections and depend on the properties of the loss function. If the conditions are not met,
$x_{t+1}$ is directly used. We define the counter $c_t$:

$$c_{t+1} = \begin{cases} 
  c_t + 1 & x_{t+1} \neq x_{t+1} \\
  c_t & \text{otherwise}
\end{cases}
$$

with $c_0 = 0$. The variable $c_t$ represents the number of time the predictive update of Definition 4.2 is
used. Let $\nu = c_T / T$, the ratio of rounds using the predictive update to the total number of rounds.

Depending on the loss function, any regret-bounded OCO update can be used in the POCO frame-
work. Back to the the OGD example, the predictive online gradient descent uses the update (4.2) and if
certain conditions are met,

$$x_{t+1} = \text{proj}_X (x_{t+1} - \beta_t g_t),$$

and if not, $x_{t+1} = x_{t+1}$.

We write $x_{t+1}(\beta_t) = \text{proj}_X (x_{t+1} - \beta_t g_t)$ as a function of the step size $\beta_t > 0$ and let $d_{t+1} =
(x_{t+1}(\beta_t) - x_{t+1}$ be the descent direction.

Next, we provide sufficient conditions for the estimated gradient $g_t$ to be a feasible descent direction.
Later, we consider the step size selection problem. Particularly, two cases are considered where (i) the
step sizes are chosen via *a priori* determined rules based on the properties of the loss functions, or (ii)
the step sizes are selected through the application of a backtracking line search that enforces a modified
version of the Armijo condition [125].

The following lemma introduces a sufficient condition for the estimated gradient $g_t$ to be a descent direction of the OCO problem (4.1).

**Lemma 4.1 (Estimated descent direction).** The estimated gradient $g_t$ provided by the $\epsilon$-forecaster is a
descent direction for $f_{t+1}(x_{t+1})$ if $\|g_t\| > \epsilon$.

*Proof.* Define $e_t \in \mathbb{R}^n$ as $e_t = g_t - \nabla f_{t+1}(x_{t+1})$ where $\|e_t\| \leq \epsilon$ by the definition of the $\epsilon$-forecaster. $g_t$
is a descent direction if $\mathbf{g}_t^T \nabla f_{t+1}(\mathbf{x}_{t+1}) > 0$. From this, we have

$$
0 < \mathbf{g}_t^T \nabla f_{t+1}(\mathbf{x}_{t+1}) \\
= \mathbf{g}_t^T (\mathbf{g}_t - \mathbf{e}_t) \\
= \mathbf{g}_t^T \mathbf{g}_t - \mathbf{g}_t^T \mathbf{e}_t.
$$

Equivalently, we have

$$
\mathbf{g}_t^T \mathbf{g}_t > \mathbf{g}_t^T \mathbf{e}_t,
$$

and, taking the norm of both sides,

$$
\|\mathbf{g}_t\|^2 > \|\mathbf{g}_t\| \|\mathbf{e}_t\| \cos \theta_{\mathbf{g}_t, \mathbf{e}_t},
$$

where $\theta_{\mathbf{g}_t, \mathbf{e}_t}$ is the angle between $\mathbf{g}_t$ and $\mathbf{e}_t$. Dividing both side by the norm of $\mathbf{g}_t$ gives

$$
\|\mathbf{g}_t\| > \|\mathbf{e}_t\| \cos \theta_{\mathbf{g}_t, \mathbf{e}_t}.
$$

By assumption, $\|\mathbf{g}_t\| > \epsilon$ and $\|\mathbf{e}_t\| \leq \epsilon$. Therefore (4.5) always holds and we have proved the lemma.

**The following lemma ensures that the predictive step follows a feasible descent direction.**

**Lemma 4.2** (Feasible estimated descent direction). *For all $\beta_t > 0$ and $\mathbf{x}_{t+1} \in \mathcal{X}$, if $\|\mathbf{g}_t\| > \epsilon$ and $\mathbf{x}_{t+1}(\beta_t) \neq \mathbf{x}_{t+1}$, then $\mathbf{x}_{t+1}(\beta_t) - \mathbf{x}_{t+1}$ is a feasible descent direction at $\mathbf{x}_{t+1}$ and

$$
\mathbf{g}_t^T (\mathbf{x}_{t+1}(\beta_t) - \mathbf{x}_{t+1}) \leq -\frac{1}{\beta_t} \|\mathbf{x}_{t+1}(\beta_t) - \mathbf{x}_{t+1}\|^2.
$$

**Proof.** We adapt the proof of [125, Proposition 6.1.1] for the estimated gradient $\mathbf{g}_t$. By the properties of the projection [126, Proposition 2.2.1], we have

$$
0 \geq (\mathbf{x}_{t+1} - \beta_t \mathbf{g}_t - \mathbf{x}_{t+1}(\beta_t))^T (\mathbf{x} - \mathbf{x}_{t+1}(\beta_t))
$$

for all $\mathbf{x} \in \mathcal{X}$. We let $\mathbf{x} = \mathbf{x}_{t+1} \in \mathcal{X}$ and obtain,

$$
0 \geq (\mathbf{x}_{t+1} - \beta_t \mathbf{g}_t - \mathbf{x}_{t+1}(\beta_t))^T (\mathbf{x}_{t+1} - \mathbf{x}_{t+1}(\beta_t)) \\
= (\mathbf{x}_{t+1} - \mathbf{x}_{t+1}(\beta_t))^T (\mathbf{x}_{t+1} - \mathbf{x}_{t+1}(\beta_t)) - \beta_t \mathbf{g}_t^T (\mathbf{x}_{t+1} - \mathbf{x}_{t+1}(\beta_t))
$$

Rearranging the terms leads to,

$$
\mathbf{g}_t^T (\mathbf{x}_{t+1}(\beta_t) - \mathbf{x}_{t+1}) \leq -\frac{1}{\beta_t} \|\mathbf{x}_{t+1}(\beta_t) - \mathbf{x}_{t+1}\|^2.
$$

It then follows from Lemma 4.1 that $\mathbf{g}_t$ is a descent direction at $\mathbf{x}_{t+1}$. Thus, $\mathbf{d}_{t+1} = \mathbf{x}_{t+1}(\beta_t) - \mathbf{x}_{t+1}$ is a feasible descent direction because $\mathbf{d}_{t+1} \in \mathcal{X}$ and $\mathbf{g}_t^T \mathbf{d}_{t+1} < 0$ for all $t$ and $\mathbf{x}_{t+1} \in \mathcal{X}$. \qed
4.4 POCO with predetermined step size updates

We now present predictive updates where step sizes $\beta_t$ are selected based on \emph{a priori} determined rules. Specifically, two cases are considered: step size selection in the presence of a minimum change in the decision variable (Section 4.4.1) and a fixed step size (Section 4.4.2). We conclude this section by providing regret-bounded algorithms using these updates.

4.4.1 Predictive update with minimum change in the decision variable

In this subsection, we make the following additional assumptions:

\textbf{Assumption 4.4.} Let $L > 0$, the loss function $f_t(x)$ has an $L$-Lipschitz gradient:

$$\|\nabla f_t(x) - \nabla f_t(y)\| \leq L\|x - y\|$$

for all $t = 1, 2, \ldots, T$ and $x, y \in X$.

\textbf{Assumption 4.5.} If there exists a $\delta > 0$ such that $f_{t+1}(x_{t+1}) - f_{t+1}(x_{t+1}(\beta_t)) \geq \delta$, then there exists at time $t$ a positive scalar $\lambda_t$ such that $\lambda_t \leq \|x_{t+1}(\beta_t) - x_{t+1}\|$, or, equivalently, $\lambda_t \leq \|d_{t+1}\|$.

Under these assumptions, we obtain the following lemma for the predictive update with minimum change step size. This update is referred as minimum change because it assumes the existence of $\lambda_t$, a lower bound on the difference between $x_{t+1}$ and $x_{t+1}$, for this round (cf. Assumption 4.5).

\textbf{Lemma 4.3.} (Predictive update with minimum change step size) Suppose Assumptions 4.4 and 4.5 are met and $\|g_t\| > \epsilon$. Then, if $\beta_t \leq \frac{2\lambda_t^2}{L^2(\delta + \epsilon D)} + \frac{L\lambda_t}{2}$, the predictive update \eqref{eq:predictive-update} used by the $\epsilon$-forecaster strictly improves on the OCO update and the improvement is bounded below by $\delta > 0$.

\textbf{Proof.} $f_{t+1}$ has an $L$-Lipschitz gradient, and therefore, using \cite[Theorem 2.1.5]{[127]}, the following inequality holds:

$$f_{t+1}(y) \leq f_{t+1}(x) + \nabla f_{t+1}(x)^\top (y - x) + \frac{L}{2}\|x - y\|^2$$

for all $x, y \in X$. We substitute $y = x_{t+1}(\beta_t)$ and $x = x_{t+1}$ into \eqref{eq:predictive-update} and obtain:

$$f_{t+1}(x_{t+1}(\beta_t)) \leq \nabla f_{t+1}(x)^\top (x_{t+1}(\beta_t) - x_{t+1}) + f_{t+1}(x_{t+1}) + \frac{L}{2}\|x_{t+1}(\beta_t) - x_{t+1}\|^2.$$ 

For the reminder of the proof, we use $d_{t+1} = x_{t+1}(\beta_t) - x_{t+1}$ to simplify the notation. Rewriting the gradient in terms of the estimated gradient leads to

$$f_{t+1}(x_{t+1}(\beta_t)) \leq f_{t+1}(x_{t+1}) + (g_t - e_t)^\top d_{t+1} + \frac{L}{2}\|d_{t+1}\|^2$$

$$= f_{t+1}(x_{t+1}) + g_t^\top d_{t+1} - e_t^\top d_{t+1} + \frac{L}{2}\|d_{t+1}\|^2.$$ 

By assumption, $x_{t+1}(\beta_t) \neq x_{t+1}$, and hence $d_{t+1} \neq 0$. We use Lemma 4.2 to upper bound the second
term of the right-hand side of (4.7). Thus,

\[ f_{t+1}(x_{t+1}(\beta_t)) \leq f_{t+1}(\mathbf{x}_{t+1}) - \frac{1}{\beta} \|d_{t+1}\|^2 - e^\top t d_{t+1} + \frac{L}{2} \|d_{t+1}\|^2 \]

\[ = f_{t+1}(\mathbf{x}_{t+1}) - \left( \frac{1}{\beta_t} - \frac{L}{2} \right) \|d_{t+1}\|^2 - e^\top t d_{t+1} \]

\[ \leq f_{t+1}(\mathbf{x}_{t+1}) - \left( \frac{1}{\beta_t} - \frac{L}{2} \right) \|d_{t+1}\|^2 + \|e_t\|d_{t+1} \]

\[ \leq f_{t+1}(\mathbf{x}_{t+1}) - \left( \frac{1}{\beta_t} - \frac{L}{2} \right) \|d_{t+1}\|^2 + \epsilon D \]

where the last inequality is due to \( \|d_{t+1}\| \leq D = \text{diam} \mathcal{X} \). The predictive update will therefore improve the OCO update at least by \( \delta > 0 \) if

\[- \left( \frac{1}{\beta_t} - \frac{L}{2} \right) \|d_{t+1}\|^2 + \epsilon D \leq -\delta.\]

Assuming \( \beta_t < \frac{2}{L} \), and because the norm of the feasible descent direction \( d_{t+1} \) is bounded below by \( \lambda_t \) according to Assumption 4.5 at time \( t \), we have

\[ \left( \frac{1}{\beta_t} - \frac{L}{2} \right) \lambda_t^2 - \epsilon D \geq \delta. \]

Rearranging, we obtain

\[ \beta_t \leq \frac{2\lambda_t^2}{2(\delta + \epsilon D) + L\lambda_t^2} \]

that satisfies the assumption \( \beta_t < \frac{2}{L} \). Therefore, by setting \( \beta_t \leq \frac{2\lambda_t^2}{2(\delta + \epsilon D) + L\lambda_t^2} \), we guarantee that

\[ f_{t+1}(x_{t+1}(\beta_t)) \leq f_{t+1}(\mathbf{x}_{t+1}) - \delta \]

where \( \delta > 0 \). This proves that the predictive update strictly improves over the OCO update.

4.4.2 Predictive update with fixed step size

Next, we remove Assumption 4.5 and propose a predictive update with fixed step size.

Lemma 4.4. (Predictive update with fixed step size) Suppose that Assumption 4.4 holds and \( \|g_t\| > \epsilon \). If \( \beta \leq \frac{1}{L} \) and \( \|d_{t+1}\| = \|\text{proj}_X (\mathbf{x}_{t+1} - \beta_t g_t) - \mathbf{x}_{t+1}\| \geq \frac{\epsilon}{L} + \sqrt{\frac{\epsilon^2}{L^2} + \frac{\delta^2}{L}} \), then the predictive update (4.4) used by the \( \epsilon \)-forecaster strictly improves on the OCO update and the improvement is bounded below by \( \delta > 0 \).

Proof. We reuse the same first steps as in the proof of Lemma 4.3. By Assumption 4.4, \( f_{t+1} \) has an \( L \)-Lipschitz gradient. We use the following inequality from [127, Theorem 2.1.5]

\[ f_{t+1}(y) \leq f_{t+1}(x) + \nabla f_{t+1}(x)^\top (y - x) + \frac{L}{2} \|y - x\|^2 \]

(4.9)
for all \( x, y \in \mathcal{X} \). We substitute \( y = x_{t+1}(\beta) \) and \( x = x_{t+1} \) into (4.9) to obtain

\[
f_{t+1}(x_{t+1}(\beta)) \leq f_{t+1}(x_{t+1}) + \nabla f_{t+1}(x_{t+1})^\top (x_{t+1}(\beta) - x_{t+1}) + \frac{L}{2} \|x_{t+1}(\beta) - x_{t+1}\|^2.
\]

For the reminder of the proof, we use \( d_{t+1} = x_{t+1}(\beta) - x_{t+1} \) to simplify the notation. We rewrite the gradient in terms of the estimated gradient and, which yields

\[
f_{t+1}(x_{t+1}(\beta)) \leq f_{t+1}(x_{t+1}) + (g_t - e_t)^\top d_{t+1} + \frac{L}{2} \|d_{t+1}\|^2
\]

\[
= f_{t+1}(x_{t+1}) + g_t^\top d_{t+1} - e_t^\top d_{t+1} + \frac{L}{2} \|d_{t+1}\|^2.
\]

(4.10)

By assumption, \( x_{t+1}(\beta) \neq x_{t+1} \), which ensures that Lemma 4.2 holds. We use Lemma 4.2 to upper bound the second term of the right-hand side of (4.10). We then have

\[
f_{t+1}(x_{t+1}(\beta)) \leq f_{t+1}(x_{t+1}) - \frac{1}{\beta} \|d_{t+1}\|^2 - e_t^\top d_{t+1} + \frac{L}{2} \|d_{t+1}\|^2
\]

\[
= f_{t+1}(x_{t+1}) - \left( \frac{1}{\beta} - \frac{L}{2} \right) \|d_{t+1}\|^2 - e_t^\top d_{t+1}
\]

\[
\leq f_{t+1}(x_{t+1}) - \left( \frac{1}{\beta} - \frac{L}{2} \right) \|d_{t+1}\|^2 + \|e_t\| \|d_{t+1}\|
\]

\[
\leq f_{t+1}(x_{t+1}) - \left( \frac{1}{\beta} - \frac{L}{2} \right) \|d_{t+1}\|^2 + \epsilon \|d_{t+1}\|.
\]

Therefore, the predictive update with fixed step size will improve on the OCO update by a minimum of \( \delta > 0 \) if the following condition is satisfied:

\[-\frac{1}{\beta} \|d_{t+1}\|^2 + \frac{L}{2} \|d_{t+1}\|^2 + \epsilon \|d_{t+1}\| \leq -\delta,
\]

or equivalently,

\[
\frac{1}{\beta} \|d_{t+1}\|^2 - \frac{L}{2} \|d_{t+1}\|^2 - \epsilon \|d_{t+1}\| \geq \delta.
\]

(4.11)

Assuming \( 0 < \beta \leq \frac{1}{L} \), then \( \frac{1}{\beta} \geq L \), and if

\[
\frac{L}{2} \|d_{t+1}\|^2 - \epsilon \|d_{t+1}\| \geq \delta,
\]

then (4.11) also holds for any \( \beta \in ]0, \frac{1}{L}]. \) Solving for the norm of the feasible descent direction \( \|d_{t+1}\| \), we have

\[
\|d_{t+1}\| = \|x_{t+1}(\beta) - x_{t+1}\| \geq \frac{\epsilon}{L} + \sqrt{\frac{\epsilon^2}{L^2} + \frac{2\delta}{L}}.
\]

(4.12)

Thus, by setting \( 0 < \beta \leq \frac{1}{L} \) and satisfying (4.12), we obtain

\[
f_{t+1}(x_{t+1}(\beta)) \leq f_{t+1}(x_{t+1}) - \delta,
\]

where \( \delta > 0 \). This implies that the predictive update strictly improves over the OCO update when the feasible descent direction satisfies the condition (4.12). The improvement is bounded below by \( \delta \). □

Lemmas 4.3 and 4.4 differ in their required assumptions and the conditions needed to achieve an
improvement \( \delta \) over the OCO update. In Lemma 4.3, a minimum change in the decision variable, \( \lambda_t \), must be assumed beforehand and the minimum improvement over the loss function \( \delta \) must be set accordingly. If used in a POCO algorithm, \( \beta_t \) can change from one round to another depending on the desired minimum improvement \( \delta \) because Assumption 4.5 concerns only the loss function in the current step. This update can be potentially more adaptive to loss function with large variation between rounds. In Lemma 4.4, Assumption 4.5 is not required. In other words, there is no need to know a priori the minimum descent norm with respect to some minimum improvement. As a consequence, the step size \( \beta_t \) is the same for all rounds but a minimum on the norm of the feasible descent direction must be checked at each rounds to ensure that the feasible estimated gradient step leads to an improvement. This condition is given by (4.12). In the next section, we present a backtracking algorithm for predictive gradient projection. This approach allows round-varying step size without requiring Assumptions 4.4 and 4.5. Thus, this approach can be applied to a broader family of functions.

4.4.3 Regret bound for POCO algorithm with predetermined step size

We conclude this section by presenting regret bounds for the POCO algorithm. This algorithm uses the predictive update to improve the performance of OCO algorithms.

**Theorem 4.1** (POCO regret bound). Consider an OCO algorithm with a sublinear regret upper bound. Suppose that the forecaster uses the predictive update (4.4) only at rounds \( t \) when the estimated gradient \( g_t \) and feasible descent direction \( d_{t+1} = \text{proj}_X(x_{t+1} - \beta_t g_t) - x_{t+1} \) satisfy the assumptions of Lemma 4.3 or Lemma 4.4. If the ratio of rounds satisfying these assumptions is greater than \( \nu \), then the regret of the POCO algorithm is bounded above by

\[
\text{Regret}_T(\text{POCO}) \leq \text{Regret}_T(\text{OCO}) - T\nu\delta.
\]

**Proof.** Let \( x_t \) denote the decision variable with \( \beta_t = 0 \) for all \( t \). In other words, \( x_t \) represents the decision variable computed without the predictive algorithm. Denote the set of assumptions of Lemma 4.3 or 4.4 at round \( t \) by \( A_t \). Let \( I_{A_t} \) be the indicator function where \( I_{A_t} = 1 \) if the assumptions are satisfied and 0 otherwise.

Observe that the improvement, \( i_t \), is given by

\[
i_t I_{A_t} = f_t(x_t) - f_t(x_t^*),
\]

where \( i_t \) is the improvement when \( I_{A_t} = 1 \). The regret of the POCO algorithm is

\[
\text{Regret}_T(\text{POCO}) = \sum_{t=1}^T f_t(x_t) - f_t(x_t^*).
\]
Using (4.13), we re-express $f_t(x_t)$ in (4.14):

$$
\text{Regret}^d_T(POCO) = \sum_{t=1}^{T} f_t(\hat{x}_t) - i_t I_{A_t} - f_t(x_t^*)
$$

$$
= \sum_{t=1}^{T} f_t(\hat{x}_t) - f_t(x_t^*) - \sum_{t=1}^{T} i_t I_{A_t}
$$

$$
= \text{Regret}^d_T(OCO) - \sum_{t=1}^{T} i_t I_{A_t}. \tag{4.15}
$$

By Lemma 4.3 and 4.4, the improvement $i_t$ is bounded below by $\delta$. We rewrite (4.15) as

$$
\text{Regret}^d_T(POCO) \leq \text{Regret}^d_T(OCO) - \sum_{t=1}^{T} \delta I_{A_t}.
$$

A minimum of $Tv$ rounds satisfy $A_t$ and hence

$$
\text{Regret}^d_T(POCO) \leq \text{Regret}^d_T(OCO) - T\nu \delta. \tag{4.16}
$$

Applying this theorem gives the following corollary for when the predictive update is used with the OGD algorithm (POGD).

**Corollary 4.1** ($O(\sqrt{T})$ regret bound for POGD). Suppose that the ratio $\nu$ of rounds that respects the assumptions of Lemma 4.3 or Lemma 4.4 is $\nu > \frac{1}{\sqrt{T}}$. Then the predictive OGD algorithm’s regret is bounded above by

$$
\text{Regret}^d_T(POGD) \leq \text{Regret}^d_T(OGD) - \delta \sqrt{T},
$$

$$
= \left( \frac{7X^2}{4} + \frac{G^2}{2} + XV_T - \delta \right) \sqrt{T},
$$

which is sublinear and tighter than the OGD regret bound.

**Proof.** The dynamic regret bound for the OGD algorithm, $\text{Regret}^d_T(OGD)$, is given in [23]. The results then follows from substituting $\text{Regret}^d_T(OGD)$ and $\nu > \frac{1}{\sqrt{T}}$ in Theorem 4.1. 

We now provide an extra corollary for cases where a random gradient estimator is used.

**Corollary 4.2** (Random gradient estimator). Suppose the forecaster uses the predictive update that meets the assumptions of Lemma 4.3 or Lemma 4.4 with probability $p$ at each round. Then the expected regret is bounded above by

$$
\mathbb{E} \left[ \text{Regret}^d_T(POCO) \right] \leq \text{Regret}^d_T(OCO) - \delta Tp.
$$

**Proof.** This results follows from Theorem 4.1 and the proof of Corollary 4.1.
4.5 POCO with backtracking line search

In this section, we do not require Assumption 4.4 nor Assumption 4.5 to hold. We however make the following additional assumption:

**Assumption 4.6.** The loss function $f_t(x)$ is $\Delta$-time-Lipschitz with $\Delta > 0$, that is:

$$|f_t(x) - f_{t+\tau}(x)| \leq \Delta |\tau|$$

for all $\tau \in \{-t, -t+1, \ldots, -1, 1, \ldots, T-t-1, T-t\}$, $x \in X$.

In the case of the POCO with backtracking (POCOb), we re-express the update (4.4) given in Definition 4.3.

**Definition 4.3** (POCOb update). Let $\zeta$ be a positive scalar and $\beta^m$ be determined by a backtracking line search algorithm. The predictive update with backtracking line search is

$$x_{t+1} = x_{t+1} + \beta^m \left( \text{proj}_X (x_{t+1} - \zeta g_t) - x_{t+1} \right)$$  \hspace{1cm} (4.17)

The backtracking line search for predictive update is given in Algorithm 4.1.

**Algorithm 4.1** Backtracking algorithm for predictive gradient projection

1: **Parameters:** Given $\beta \in (0, 1)$ and $M \in \mathbb{N}$.
2: **Initialization:** Set $\zeta > 0$.
3: $d_{t+1} = \text{proj}_X (x_{t+1} + \zeta g_t) - x_{t+1}$
4: $m = 0$.
5: while $f_t(x_{t+1} + \beta^m d_{t+1}) > f_t(x_{t+1}) + \beta^m (g_t^T d_{t+1} - \epsilon \|d_{t+1}\|) - 2\Delta$ and $m \leq M$ do
6: $m = m + 1$.
7: end while
8: if $m > M$ then
9: $\beta = 0$.
10: end if

**Lemma 4.5** (Sufficient decrease for POCOb). Suppose that Assumption 4.6 holds and $\|g_t\| > \epsilon$. If Algorithm 4.1 terminates to a step size $\beta^m > 0$, then the predictive update with backtracking line search (4.17) used by the $\epsilon$-forecaster satisfies the modified Armijo condition,

$$f_{t+1}(x_{t+1} + \beta^m d_{t+1}) \leq f_{t+1}(x_{t+1}) + \beta^m \nabla f_{t+1}(x_{t+1})^T d_{t+1},$$  \hspace{1cm} (4.18)

and will thus lead to a sufficient decrease in the loss function, outperforming the OCO update.

**Proof.** We show that for some step size $\beta^m$, the estimated gradient descent projection leads to a sufficient decrease thus outperforming the OCO update. Our claim relies on the modified Armijo condition for gradient projection. This condition ensures a sufficient decrease in the objective when using an estimated gradient projection descent direction [103]. We adapt this condition to the estimated gradient and online setting. The modified Armijo condition for gradient projection [125] on $f_{t+1}$ and feasible descent direction...
\[ d_{t+1} = \text{proj}_X (x_{t+1} + \zeta g_t) - x_{t+1} \] for some \( \zeta > 0 \) with step size \( \beta^m \) is given by (4.18). We show that if \( \beta^m \) satisfies

\[ f_t (x_{t+1} + \beta^m d_{t+1}) \leq f_t (x_{t+1}) + \beta^m g_t^\top d_{t+1} - \beta^m \epsilon \| d_{t+1} \| - 2\Delta \] (4.19)

then it also satisfies satisfies (4.18), ensuring a sufficient decrease in the objective function. Note that (4.19) is the condition under which the backtracking algorithm, Algorithm 4.1, is used. We can see from the left-hand side of the condition (4.19) that the update improves over the OCO update because the three last terms are bounded above by 0, i.e., \( g_t^\top d_{t+1} \leq -\xi \| d_{t+1} \|^2 \) by Lemma 4.2. Thus all three terms are less or equal to zero. By assumption, \( \beta > 0 \), and these terms are also bounded away from zero since \( x_{t+1} \neq x_{t+1} \).

We start from (4.19) and shows it implies (4.18). By assumption, \( \| e_t \| \leq \epsilon \) for all \( t \) and hence (4.19) implies,

\[ f_t (x_{t+1} + \beta^m d_{t+1}) \leq f_t (x_{t+1}) + \beta^m (g_t - e_t)^\top d_{t+1} - 2\Delta. \]

Rearranging the terms, we have

\[ f_t (x_{t+1} + \beta^m d_{t+1}) + \Delta \leq f_t (x_{t+1}) - \Delta + \beta^m \nabla f_{t+1} (x_{t+1})^\top d_{t+1}. \] (4.20)

By assumption, \( f_{t+1} \) is time-Lipschitz with constant \( \Delta > 0 \) for all \( x \in X \) and all \( t \), and therefore the following inequalities hold:

\[ f_{t+1} (x) \leq f_t (x) + \Delta \] (4.21)

\[ f_{t+1} (x) \geq f_t (x) - \Delta \] (4.22)

We use (4.21) to lower bound the left-hand side of (4.20) and (4.22) to upper bound the its right-hand side. This leads to

\[ f_{t+1} (x_{t+1} + \beta^m d_{t+1}) \leq f_{t+1} (x_{t+1}) + \beta^m \nabla f_{t+1} (x_{t+1})^\top d_{t+1}, \]

the modified Armijo condition (4.18).

**Remark 4.1.** Every element of (4.19) is available at time \( t \), which is not the case in (4.18). This allows us to use a backtracking line search algorithm to determine \( \beta_t \) in an OCO setting. Algorithm 4.1 also ensures that the step size is not too small (cf. [103, Section 3.1]).

Note that there is an additional \( \epsilon \| d_{t+1} \| \) term in the modified Armijo condition for estimated gradient projection. This is a consequence of not having access to the exact gradient of \( f_t \). Hence, to ensure that the update is valid, the modified Armijo condition is augmented by a term proportional to the error of the estimated gradient. The second additional term, \( 2\Delta \), is due to the time-varying setting of OCO.

**Proposition 4.1** (Relaxation of Assumption 4.6). The Assumption 4.6 in Lemma 4.5 can be relaxed to milder time-Lipschitz assumptions at point \( x \in X \) and \( t \in \{1, 2, \ldots, T\} \) without consequences on our results.
Consider \( x \in X \) at round \( t \in \{1, 2, \ldots, T\} \), the loss function \( f_t(x) \) is \( \Delta_t(x) \)-time-Lipschitz at \( x \) and \( t \) with \( \Delta_t(x) > 0 \) if it satisfies
\[
|f_t(x) - f_{t+\tau}(x)| \leq \Delta_t(x)|\tau|
\]
(4.23)
for all \( \tau \in \{-t, -t+1, \ldots, -1, 1, \ldots, T - t - 1, T - t\} \).

We can relax Assumption 4.6 using (4.23) and the fact that we only need a bound for \( \tau = 1 \) in our results, thus we can use the following bound at \( x \) for \( t \).

**Assumption 4.7.** For \( x \in X \), the loss function \( f_t(x) \) is \( \Delta_{t,1}(x) \)-bounded at \( x \) and \( t \) with \( \Delta_{t,1}(x) > 0 \) if
\[
|f_t(x) - f_{t+1}(x)| \leq \Delta_{t,1}(x)
\]
holds.

Using Assumption 4.7, we can re-express the modified Armijo condition (4.18). For example, under Assumption 4.7, we set \( 2\Delta = \Delta_{t,1}(x_{t+1} + \beta g_t) + \Delta_{t,1}(x_t) \) in (4.19) and on Line 5 of Algorithm 4.1. We then prove Lemma 4.5 by using Assumption 4.7 instead of Assumption 4.6 to get (4.21) and (4.22).

We now discuss the existence of step sizes \( \beta \) that satisfy (4.19). Before stating the main result, for a given \( x_{t+1} \) and \( g_t \), define the set of step sizes that comply with line 5 in the line search algorithm, which is the modified Armijo condition for online settings:
\[
S = \{ \beta > 0 | f_t(x_{t+1} + \beta d_{t+1}) \leq f_t(x_{t+1}) + \beta \nabla f_t(x_{t+1}) d_{t+1} - \beta \epsilon \|d_{t+1}\| - 2\Delta \}.
\]

**Theorem 4.2.** Suppose \( d_{t+1} = \text{proj}_X(x_{t+1} - \zeta g_t) - x_{t+1} \neq 0 \) is a feasible descent direction and \( f_t \) is bounded below for all \( t \). Then there exists \( x \in X \) such that \( f_t(x_{t+1}) - f_t(x) > 2\Delta \) if and only if \( S \neq \emptyset \).

**Proof.** Assume \( f_t(x_{t+1}) - f_t(x) > 2\Delta \). Recalling inequalities (4.21) and (4.22), this assumption implies that
\[
f_{t+1}(x_{t+1}) - f_{t+1}(x) > 0.
\]
Thus, \( x_{t+1} \) is not the minimum point of \( f_{t+1} \). It follows that \( \nabla f_{t+1}(x_{t+1}) \neq 0 \). By assumption, \( d_{t+1} \neq 0 \) is a feasible descent direction and we have
\[
\nabla f_{t+1}(x_{t+1})^T d_{t+1} < 0.
\]
(4.24)

Let \( a \in (0, 1) \). Subtracting \( \nabla f_t(x_{t+1} + a\beta d_{t+1})^T d_{t+1} \) on both side of (4.24) we obtain,
\[
(\nabla f_{t+1}(x_{t+1}) - \nabla f_t(x_{t+1} + a\beta d_{t+1}))^T d_{t+1} < -\nabla f_t(x_{t+1} + a\beta d_{t+1})^T d_{t+1}.
\]
(4.25)
If the following condition holds, then (4.25) also holds:
\[
\|\nabla f_{t+1}(x_{t+1}) - \nabla f_t(x_{t+1} + a\beta d_{t+1})\|d_{t+1}\| < -\nabla f_t(x_{t+1} + a\beta d_{t+1})^T d_{t+1}.
\]
(4.26)
We observe that under Assumption [L.3] for all $x, z \in \mathcal{X}$ we have

$$\|\nabla f_{t+1}(x) - \nabla f_t(z)\| \leq \|\nabla f_{t+1}(x)\| + \|\nabla f_t(z)\| \leq 2G$$

and by Assumption [4.1] we have $\|d_{t+1}\| \leq D$. Then, if

$$2GD < -\nabla f_t(x_{t+1} + \beta d_{t+1})^\top d_{t+1}$$

holds, then so does [4.26]. We rewrite [4.27] as

$$\nabla f_t(x_{t+1} + \beta d_{t+1})^\top d_{t+1} < -2GD$$

Recalling Taylor’s Theorem [103] Theorem 2.1:

$$f_t(y + p) = f_t(y) + \nabla f_t(y + a p)^\top p.$$  

where $y, x \in \mathcal{X}$, $p \in \mathbb{R}^n$ and for some $a \in (0, 1)$. We let $y = x_{t+1}$ and $p = \beta d_{t+1}$. We have,

$$f_t(x_{t+1} + \beta d_{t+1}) = f_t(x_{t+1}) + \beta \nabla f_t(x_{t+1} + a \beta d_{t+1})^\top d_{t+1}.$$  

We bound above the last term of [4.29] using [4.28] and obtain

$$f_t(x_{t+1} + \beta d_{t+1}) < f_t(x_{t+1}) - 2\beta GD$$

By setting $\beta \leq \frac{\Delta}{GD}$ in [4.30], we then have

$$f_t(x_{t+1} + \beta d_{t+1}) < f_t(x_{t+1}) - 2\Delta$$

This shows that there always exists at least one point which satisfies the assumption on the existence of $x \in \mathcal{X}$ such that $f_t(x_{t+1}) - f_t(x) > 2\Delta$ that is along the feasible descent direction $d_{t+1}$ from $x_{t+1}$.

We now adapt the proof of [103] to our setting. Let $\phi(\beta) = f_t(x_{t+1} + \beta d_{t+1})$ and $\ell(\beta) = f_t(x_{t+1}) - \beta (g_t^\top d_{t+1} + \epsilon \|d_{t+1}\|) - 2\Delta$. Using these definition, we can rewrite [4.19] as $\phi(\beta m) \leq \ell(\beta)$. Observe that $\ell(\beta)$ is a line with a negative slope and an intercept given by $f_t(x_{t+1}) - 2\Delta$. The line $\ell(\beta)$ is strictly decreasing as a function $\beta$ because the slope is bounded away from zero (cf. proof of Lemma [4.1]). $f_t$ is assumed to be bounded below and hence $\phi(\beta)$ is also bounded below.

Setting $\beta_1 = \beta \leq \frac{\Delta}{GD}$ implies that $\phi(\beta_1) < f_t(x_{t+1}) - 2\Delta = \ell(0)$. Since the line $\ell(\beta m)$ is strictly decreasing and $\phi(\beta m)$ has at least one points below $\ell(0)$, there exists $\overline{\beta} \in \mathcal{S}$ for which both functions intersect. Expanding $\phi(\overline{\beta}) \leq \ell(\overline{\beta})$, we have

$$f_t(x_{t+1} + \overline{\beta} d_{t+1}) \leq f_t(x_{t+1}) + \overline{\beta} g_t^\top d_{t+1} - \overline{\beta} \epsilon \|d_{t+1}\| - 2\Delta.$$  

The set $\mathcal{S}$ is therefore non-empty if there exists $x \in \mathcal{X}$ such that $f_t(x_{t+1}) - f_t(x) > 2\Delta$. 

We now show the converse. Assuming $\mathcal{S} \neq \emptyset$, then there exists $\beta_2 \in \mathcal{S}$ such that

$$\phi(\beta_2) \leq \ell(\beta_2) = f_t(\mathbf{x}_{t+1}) + \beta_2 (\mathbf{g}_t^\top \mathbf{d}_{t+1} - \epsilon \| \mathbf{d}_{t+1} \|) - 2\Delta$$

$$< f_t(\mathbf{x}_{t+1}) - 2\Delta \tag{4.31}$$

since $\mathbf{g}_t^\top \mathbf{d}_{t+1} - \epsilon \| \mathbf{d}_{t+1} \| < 0$ by Lemma 4.2. Let $\mathbf{x} = \mathbf{x}_{t+1} + \beta_2 \mathbf{d}_{t+1}$. Rearranging terms gives $\phi(\beta_2) = f_t(\mathbf{x}_{t+1} + \beta_2 \mathbf{d}_{t+1}) = f_t(\mathbf{x})$. Then, (4.31) implies that,

$$\phi(\beta_2) = f_t(\mathbf{x}) < f_t(\mathbf{x}_{t+1}) - 2\Delta$$

and thus,

$$2\Delta < f_t(\mathbf{x}_{t+1}) - f_t(\mathbf{x}) \tag{4.32}$$

Hence, (4.32) implies that there exists $\mathbf{x} \in \mathcal{X}$ such that $f_t(\mathbf{x}_{t+1}) - f_t(\mathbf{x}) > 2\Delta$ and one of such point is $\mathbf{x} = \mathbf{x}_{t+1} + \beta_2 \mathbf{d}_{t+1}$. This completes the proof.

We note that Theorem 4.2 does not guarantee that the backtracking algorithm, Algorithm 4.1, will find a non-zero step size. Other techniques like exact line searches, might be required to identify an adequate step size in some problem instances. Using Theorem 4.2, we can provide a lower bound on the improvement of the predictive update with backtracking line search.

**Corollary 4.3 (POCOb minimum improvement).** Suppose that the assumptions of Lemma 4.5 hold and $\beta_m > 0$, then the predictive update with backtracking line search improves on the OCO update by a minimum of $2\Delta$.

**Proof.** Since $\beta > 0$, then $\mathcal{S} \neq \emptyset$. By the converse of Theorem 4.2, we have

$$f_t(\mathbf{x}_{t+1}) - f_t(\mathbf{x}) > 2\Delta \tag{4.33}$$

where $\mathbf{x} = \mathbf{x}_{t+1} + \beta \mathbf{d}_{t+1}$, the decision played by the predictive update (4.17). The predictive update hence improves on the OCO update by at least $2\Delta$.

We now state a regret bound for the POCO with backtracking (POCOb) algorithm.

**Theorem 4.3 (POCOb regret bound).** Consider an OCO algorithm with bounded regret. Suppose Assumption 4.6 and the assumptions of Lemma 4.5 are met. If the ratio of rounds with $\beta > 0$ and satisfying these assumptions to $T$ is greater than $\nu$, then the regret of the POCO algorithm with backtracking used by the $\epsilon$-forecaster is bounded above by

$$\text{Regret}_T^{\text{d}}(\text{POCOb}) \leq \text{Regret}_T^{\text{d}}(\text{OCO}) - 2T\nu\Delta \tag{4.34}$$

and thus outperforms the OCO algorithm.

**Proof.** Let $I_{\mathcal{A}_t}$ be the indicator function where $I_{\mathcal{A}_t} = 1$ if the $\beta_t > 0$ and $\| \mathbf{g}_t \| > \epsilon$ or 0 otherwise. Using the same approach as in Theorem 4.1’s proof with Corollary 4.3, we obtain the regret bound. The last term of (4.34) is strictly positive and thus the POCOb regret is always bounded above by the OCO algorithm regret.
Remark 4.2. Note that if Proposition 4.1 is used, then $\Delta$ in (4.34) must be substituted by $\min_{x \in X} \Delta_{t,1}(x)$.

We conclude our technical result sections with the following proposition concerning the forecaster’s assumptions.

Proposition 4.2 ($\epsilon, \theta_t$-forecaster). Let $\theta_t$ be the smallest angle between the $g_t$ and $\nabla f_t(x_{t+1})$. If we assume that the $\epsilon$-forecaster presented in Definition 4.1 also provides $g_t$ such $\theta_t < \frac{\pi}{2}$, then $g_t$ is a descent direction and we do not need to ensure that $\|g_t\| \geq \epsilon$ in Lemmas 4.3, 4.4 and 4.5 to obtain a strict improvement when using the predictive update.

Recall that the $\epsilon$-forecaster provides $g_t$ such that $\|g_t - \nabla f_{t+1}(x_{t+1})\| < \epsilon$. When checking if the inexact gradient is a descent direction using $\|g_t\| \geq \epsilon$, we are effectively restricting ourself to the case $\theta_t \leq \frac{\pi}{2}$. The $\epsilon, \theta_t$-forecaster can thus lead to an increased number of rounds meeting the sufficient condition for a strict improvement when the predictive update is used, e.g., when very close to the optimum and the norm of the gradient is very small. However, it requires a stronger assumption on the forecaster because to evaluate the angle $\theta_t$, one needs $\nabla f_{t+1}(x_{t+1})$. Thus the angle condition cannot be tested at each round like the norm of the inexact gradient.

4.6 Example

In this section, we apply POCO algorithms to demand response in power systems [11, 13], specifically regulation and curtailment. At each time step, a demand response (DR) aggregator sends instructions to their loads to follow a regulation signal, e.g., a power imbalance due to a sudden change in renewable power generation [5, 47]. A second example of a regulation signal is the area control error (ACE) [62]. Each load responds to the signal by adjusting its power consumption. The power consumption is constrained by a storage capacity, which could represent physical storage like a battery or the load’s limits, e.g., thermal constraints. The regulation signal is unknown at the time the DR instructions are sent. This can be due, for example, to a drop in renewable power generation which is only assessed after the generator has committed to some amount of power. The objective of the DR aggregator is, therefore, to predict the DR dispatch at each time instance. This problem can be formulated as POCO, in which an estimate of the regulation signal is available to the load aggregator.

4.6.1 Setting

We consider $N$ loads. We denote $x_t \in \mathbb{R}^N$ as the decision variable at round $t$. The variable $x$ represents the instructions sent to the loads. Let $r_t \in \mathbb{R}$ be the regulation at time $t$. Let $\mathbf{r}, \mathbf{x} \in \mathbb{R}^N$ be the maximum and minimum power that can be consumed or delivered for all loads. Define the decision set $\mathcal{X} = \{x \in \mathbb{R}^N \mid \underline{x} \leq x \leq \overline{x}\}$, a convex and compact set. We denote $s_t \in \mathbb{R}^N$ as the state of the loads at time $t$ and $\mathbf{c} \in \mathbb{R}^N$ as the vector vector of load energy capacities. The state of charge of a load $i$ is $s_t(i) = s_0(i) + \sum_{n=1}^{t} x_n(i)$. In the current case, we assume that there is no leakage nor energy losses.

The OCO problem takes the following form:

$$\min_{x_t \in \mathcal{X}} (r_t - 1^T x_t)^2 + \sigma \|s_{t-1} + x_t - \frac{\mathbf{c}}{2}\|^2$$  (4.35)
The loss function has two terms: (i) a regulation term where the aggregated loads are dispatched to follow a regulation signal \( r_t \) and (ii) a state of charge objective added to keep the loads near half their energy capacity. The loss function given in (4.35) is \( \sigma \)-strongly convex. For this reason we use the OGD for strongly convex functions proposed in [26], which offers tighter regret bound than the standard OGD. The following corollary gives an upper bound on the regret of predictive OGD for strongly convex function (\( \sigma \)POGD).

**Corollary 4.4 (POGD for strongly convex functions).** Suppose \( f_t \) is \( \sigma \)-strongly convex and satisfies Assumption 4.4 for all \( t \). Consider the OGD for strongly convex functions update

\[
x_{t+1} = x_t + \eta \left( \text{proj}_X \left( x_t - \frac{1}{\gamma} \nabla f_t(x_t) \right) - x_t \right)
\]

where \( \eta \in (0, 1] \) and \( 0 < \gamma \leq L \). Then, the \( \sigma \)POGD with fixed step size, given that the assumptions of Lemma 4.4 hold for a ratio of the total rounds greater than \( \nu \), has a regret bounded above by

\[
\text{Regret}^d_T(\sigma \text{POGD}) \leq \text{Regret}^d_T(\sigma \text{OGD}) - T\nu\delta \\
\leq O(V_T + 1) - T\nu\delta.
\]

**Proof.** For Corollary 4.4 we follow the proof of Theorem 4.1 and obtain

\[
\text{Regret}^d_T(\sigma \text{POGD}) \leq \text{Regret}^d_T(\sigma \text{OGD}) - T\nu\delta,
\]

where we have substituted the \( \sigma \)OGD algorithm in (4.16). From [26], we have

\[
\text{Regret}^d_T(\sigma \text{OGD}) \leq O(V_T + 1).
\]

Combining (4.36) and (4.37) leads to our result.

We now present simulation results. All optimizations are solved using CVXPY [104] and the ECOS [105] solver.

### 4.6.2 Fixed step size numerical examples

We set the duration of a round to be \( h = 30 \) seconds and \( T = 200 \). The values of \( \bar{x} \) and \( \underline{x} \) are uniformly sampled between 1 and 3 kW and then converted into energy limits by multiplying by \( h \). The storage capacity values are uniformly sampled between 10 and 15 kWh. The initial state of charge of each load is set to half its capacity. The regulation signal is \( r_t = 0.2 \sin \left( \frac{2\pi}{T} t \right) + w_t \). The parameter \( w_t \sim N(0, 0.01) \) is a Gaussian noise used to model sudden changes. We assume that the aggregator has access to gradient estimates such that \( \epsilon = 0.01 \). This value represents an error of at least 14% of the actual gradient norm. We set \( \delta = 10^{-6} \). The parameter \( \sigma \) is fixed to 0.005 and will be discussed shortly. The algorithm’s numerical parameters are \( \eta = 1 \), \( \gamma = L \) and \( \beta = \frac{1}{L} \).

The parameter \( \sigma \) is set to achieve adequate regulation performance without deviating too much from each load’s desired state of charge. Figure 4.1 shows the cumulative value of (4.35)’s second term as a function of \( \sigma \). Note that \( \sigma \) must be greater to zero for (4.35) to be strongly convex and for the regret
bound to hold. The case of $\sigma = 0$ is only shown for comparison purposes. We fix $\sigma = 0.005$ as it leads to a good decrease in the state of charge objective without significantly impacting the regulation objective.

We now present the performance of our POCO algorithm with a fixed step size. An instance of the experimental regret is given for the POCO and OCO algorithm together with their respective regret bounds in Figure 4.2a. The predictive update has been used 173 out of 200 times during the simulation. Figure 4.2a shows how the POCO outperforms both its bound and the OCO algorithm. The predictive algorithm leads to an improvement of 97% of the final regret value of the OCO algorithm.

Lastly, Figure 4.2b presents the regulation services provided by the DR aggregator. In this figure, the tracking done by the POCO and the OCO algorithm are shown in blue and orange, respectively. The
POCO algorithm accurately follows the regulation signal and consequently is almost always superimposed on \( r_t \) in Figure 4.2b. The high performance of the POCO algorithm can be observed in the zoomed subplot of Figure 4.2b.

### 4.6.3 Backtracking line search step size numerical examples

We now present an example of POCO with backtracking. We consider a curtailment scenario. We let \( p_t \) be the total power to be curtailed by the loads at time \( t \) for \( t = 1, 2, \ldots, T \). When a contingency occurs in the network, flexible loads are called to curtail their power consumption, e.g., by using their battery energy storage or temporarily shutting down their HVAC system. Contrary to the regulation case, the loads are not contracted to follow a setpoint and no penalties are assessed on loads curtailing more than asked. Similar to the regulation setting, the curtailment signal is unknown until immediately after the current round. This setting can be modeled as POCO where an inexact version of the curtailment signal is available to the aggregator at each round.

We use the same notation as the previous examples. Let \( [\cdot]^+ = \max\{0, \cdot\} \). This curtailment scenario is modeled by loss function given below:

\[
f_t(x_t) = \left( [p_t - 1^T x_t]^+ \right)^2 + \sigma \left\| \alpha s_{t-1} + x_t - c/2 \right\|^2
\]

where we have added a recovery coefficient to the state of charge objective term used previously. This coefficient models the usual evolution of the load (e.g., ambient temperature heating for a thermostatic load). We let \( \alpha = 1.001 \). This is equivalent to a recovery coefficient of 1.13 per hour. The function \( f_t \) given in (4.38) is not gradient Lipschitz and Assumption 4.4 does not hold. We model the curtailment signal to be quickly increasing at first and then slowly plateauing to represent new level of available generation. This event is assumed to be limited in time, after which the network goes back to its normal state and no curtailment is then required. We let \( p_t = 0.04 + w_t \) where \( w_t \sim N(0, 0.01) \) for \( t = 1, 2, \ldots, T/4 \) and then \( p_t = 0.04 \left( T/4 \right)^{0.3} + w'_t \) where \( w'_t \sim N(0, 0.001) \) for \( t = T/4, T/4 + 1, \ldots, T \). The noise variance is equivalent to approximately 10% of curtailment signal’s value at first and then approximately 1%.

We use Proposition 4.1 and suppose that the forecaster has access to the bounds as given in Assumption 4.7. We re-sample the loads’ parameters and we set \( \epsilon = 0.01 \), \( \eta = \frac{1}{10 \sqrt{T}} \) and \( \sigma = 5 \times 10^{-5} \). The backtracking algorithm parameters are set to \( \zeta = 0.5 \), \( M = 100 \) and \( \beta = 0.9 \). Figure 4.3 shows the performance of our algorithm. The predictive update is used 2 times out of 200 rounds and the final regret is improved by 30% when compared to the standard OCO algorithm. The POCO experimental regret shown in Figure 4.3a is sublinear in the numbers of round and outperforms the OCO’s regret. While the performance is not as high in the fixed step size, this algorithm can be applied to a broader family of functions since it does not require the loss function to be gradient Lipschitz continuous. We note that POCOb performs better when large variation of \( p_t \) are registered. Similarly to the fixed size case, the POCO allows better curtailment than its OCO counterpart in the context of DR as presented on Figure 4.3b.
4.6.4 Comparison with the Optimistic Mirror Descent

In this section, we compare our POCO setting with the low-regret Optimistic Mirror Descent (OMD) algorithm. The OMD approach assumes that the decision maker has access to an arbitrary predictable sequence \( \{M_t\}_{t=1}^T \) where \( M_t \in \mathbb{R}^N \) for all \( t \). For our comparison, we consider the Bregman divergence with respect to the \( \ell_2 \) norm for the mirror update. This reduces the OMD update to a gradient descent update similar to the POCO setting. The OMD update is the following:

\[
\eta_{t+1} = \frac{a}{\sqrt{D_t} + \sqrt{D_{t-1}}}, \tag{4.39}
\]

\[
y_{t+1} = \arg \min_{y \in \mathcal{X}} \left\{ \eta_t \nabla f_t(x_t)^\top y + \frac{1}{2} \|y - y_t\| \right\}, \tag{4.40}
\]

\[
x_{t+1} = \arg \min_{x \in \mathcal{X}} \left\{ \eta_{t+1} M_{t+1}^\top x + \frac{1}{2} \|x - y_{t+1}\| \right\}, \tag{4.41}
\]

where

\[
D_t = \sum_{i=1}^t \|\nabla f_t(x_i) - M_i\|^2 \tag{4.42}
\]

and \( a > 0 \) is a tuning parameter. The OMD framework does not enforce any conditions on the predictive sequence, but characterizes its performance with respect to \( D_T \). There are no guarantees that the predictive sequence-based update (4.41) will improve over the standard OGD update (4.40). Depending on the closeness of \( M_t \) to \( \nabla f_{t+1}(x_{t+1}) \), the update could decrease the performance of the standard update. In the POCO framework, because we evaluate the accuracy of \( g_t \) at each round, the added predictive update can only be used to improve the performance over the OCO-based algorithm.

We now give a numerical comparison of both algorithms based on the example of Section 4.6.2. Recall that the \( \sigma \text{OOGD} \) is used as the standard OCO algorithm in this POCO example. We set

\[
g_t = M_{t+1} = \nabla \left( (r_{t+1} - 1)^\top x + n_{t+1} \right)^2 + \sigma \left\| s_t + x - \frac{c}{2} \right\|^2.
\]
evaluated at $x_t = x_t$ and where $n_{t+1} = 0.999\epsilon (-1)^{\text{Bernoulli}}(\frac{1}{2})$ for all $t$. We set $\epsilon = 0.05$. All the other parameters are set according to Section 4.6.2 and let $a = \frac{10}{T}$ in the OMD. We run the OCO, POCO and OMD algorithms with $T = 200$ for 1000 times and re-sample the signal and load parameters in each time. The regret comparison between the three algorithms is presented in Figure 4.4. Figure 4.4 shows that the POCO setting outperforms both the OCO and OMD algorithms. We also present several performance indicator in Table 4.1.

Table 4.1: Comparison between OCO, POCO and OMD (averaged over 1000 trials)

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of rounds with predictive update</td>
<td>102 out of 200</td>
</tr>
<tr>
<td>Rounds where loss(POCO) &lt; loss(OMD)</td>
<td>152 out of 200</td>
</tr>
<tr>
<td>Rounds where loss(OMD) &lt; loss(POCO)</td>
<td>46 out of 200</td>
</tr>
<tr>
<td>Rounds where loss(POCO) ≤ loss(OCO)</td>
<td>200 out of 200</td>
</tr>
<tr>
<td>Rounds where loss(OMD) ≤ loss(OCO)</td>
<td>46 out of 200</td>
</tr>
<tr>
<td>Regret reduction POCO vs. OCO</td>
<td>82%</td>
</tr>
<tr>
<td>Regret reduction OMD vs. OCO</td>
<td>27%</td>
</tr>
<tr>
<td>Regret reduction POCO vs. OMD</td>
<td>75%</td>
</tr>
</tbody>
</table>

We now discuss the results gathered in Table 4.1. The predictive update was used in about half the number of rounds on average. This highlights the fact that future information may be too inaccurate to lead to an improvement. On average, in 154 out of the 200 rounds, the OMD update decreased the performance of the standard OCO update. This scenario is avoided in the POCO setting as the update is optional. Moreover, in around 76% of the rounds, the POCO round performance improved over the OMD algorithm.

Lastly, POCO and OMD lead respectively to an 82% and 27% regret reduction on average when compared to OCO update. POCO achieves a regret reduction of 75% on average over the OMD.
We conclude this section by discussing the algorithmic advantages of the POCO framework over the OMD:

1. As previously mentioned, POCO is designed such that it cannot decrease the performance of the standard algorithm OCO algorithm it used, whereas no such guarantee exists for the OMD.

2. The step size given in (4.39) of the OMD update (4.40) and (4.41) is monotonically decreasing. This is ill-adapted to dynamic settings because the step size vanishes as the time horizon increases and the update becomes ineffective.

3. The dynamic regret bound order of the OMD is given by [120]:

\[
\text{Regret}^d_f(\text{OMD}) \leq O\left(\sqrt{T} (1 + V_T)\right),
\]

where we substituted \(D_T = \sum_{t=1}^{T} \|\nabla f_t(x_t) - M_t\|^2 \leq \epsilon^2 T\) for our setting. This result is equivalent to OGD bound. Corollaries 4.1 and 4.4 shows that the POCO setting achieves a tighter regret bound and thus outperforms the OMD under the assumptions of this work.

4.7 Conclusion

In this work, we have presented the predictive online convex optimization framework. In POCO, a second update is used after the OCO update to improve performance using an estimated gradient. We have presented three versions of the predictive update that can be used under different assumptions. We have shown a regret upper bound for all of our POCO algorithms. We have applied POCO to demand response in electric power systems and found that they outperform conventional OCO using commonly available forecast information. In the case of fixed step size update, we have observed an improvement of 97% in the final regret and of 30% in the backtracking case.
Chapter 5

The Multi-armed Bandit with Stochastic Plays

5.1 Introduction

The multi-armed bandit (MAB) is a decision-making framework in which current rewards must be balanced with information gathering. It has found a wide range of applications in resource allocation problems such as ad placement [20] and more recently demand response in power systems [48-52]. Motivated by such applications, the original stochastic, adversarial and Markovian settings have all been extended to allow multiple plays, i.e., when multiple resources are deployed in each time period. This setup is appropriate for ad placement, in which the number of slots on a website is more than one and does not change over time.

Demand response refers to modifying the power consumption of electric loads to assist power system operations [11, 12]. For example, non-critical loads like air conditioners may be turned off or curtailed during times of insufficient generation to avoid instability. Because the power system’s operating condition changes through time due to varying base load, random failures, and renewables, the number of loads needed in each time period for demand response also changes in time. This motivates our extension of the stochastic MAB to the case where the number of arms to play changes in each round.

We define a new upper confidence bound-based algorithm, the UCB-SP, for the case in which the number of arms to play in each round varies according to a wide-sense stationary stochastic process. We prove that the algorithm achieves sublinear regret in the number of rounds, and test its performance on several examples.

5.1.1 Related work

5.1.1.1 Multi-armed bandits

The stochastic multi-armed bandit problem was first addressed by Lai and Robbins [35]. The well-known UCB1 algorithm was proposed by Auer et al. [36] and is now widely used on stochastic bandit problems. Several extensions to the stochastic setting have been proposed to deal with multiple plays. In [128, 129], the authors first extended the framework to allow multiple plays and derived an asymptotic bound for both stochastic and Markovian bandits. Agrawal et al. [130] incorporated switching costs into
the multiple play framework. Gai et al. [131] further extended multiple plays to the case of linear combinations of the individual rewards. Chen et al. [132] proposed an algorithm to deal with more general functions of the rewards of stochastic bandits, and [133] tightened their regret bound. Combes et al. [134] provided the tightest known bound for the stochastic bandit with multiple plays and derived UCB1-like and KL-UCB-like algorithms.

In this work, we focus on the stochastic multi-armed bandit. We note however that multiple play extensions exist for adversarial bandits [135, 136] and for the Markovian bandit [39, 129], the latter of which is known as the restless bandit.

### 5.1.1.2 Application to demand response

Loads are good candidates for bandit algorithms because they are numerous, uncertain, and admit little opportunity for measurement [14]. Curtailment of thermostatically controlled loads (TCLs) was studied by Taylor et al. [48] and later by Wang et al. [49]. These works used the Markovian bandit framework to derive policies for dispatching TCLs for curtailment. Kalathil et al. [50] developed an extension to the stochastic bandit that uses a discount factor to model overuse of arms. Bandyopadhyay et al. [51] used the CUCB algorithm [132] and a knapsack oracle to optimally curtail solar power producers when supply exceeds demand. In [52], the adversarial bandit was applied to load-shifting to learn loads’ parameters and improve performance.

### 5.1.2 Contributions

In this paper, we give a UCB-based algorithm for the MAB in which the number of arms to play in each time period varies according to a wide-sense stationary stochastic process. We obtain a sublinear upper bound for its pseudo-regret that depends on the statistics of the stochastic process. Our contribution is to extend the regret analysis of a UCB-based algorithm to a new problem setting where the number of arms to play comes from an exogenous stochastic process. We obtain the first sublinearly bounded pseudo-regret for this setting. Note that this work can be classified as semi-bandit because the player receives feedback from all played arms instead of one aggregated arm.

The most closely related works to ours are [131], [132] and [133]. Our work differs because our algorithm selects a time-varying, random number of arms at each round, which depends on an exogenous stochastic process. This contrasts [131], [132] and [133], in which the pseudo-regret is based on a fixed set of optimal arms rather than a set of arms that varies according to an exogenous stochastic process. Indeed, the number of arms to play may vary between rounds in [131], [132] and [133], but only as a result of deviations of the sample mean from the mean. They do, however, develop more general classes of policies than UCB1 algorithms. Lastly, we remark that the examples we give in Section 5.4 do not fall in the frameworks of [131], [132] or [133].

Our algorithm and analysis enables us to apply low-regret learning strategies to demand response scenarios in which the power need is random and varies in time, for example, curtailment and regulation.

In summary, we make the following contributions:

- We propose a new extension of the multi-armed bandit, the MAB with variable number of plays and define its associated pseudo-regret (Definition 5.1);
- We introduce the UCB-Stochastic Plays and derive a regret bound for the case where the number of arms to play at each round is a wide-sense stationary stochastic process (Theorem 5.1).
• We consider the special case in which the number of arms in each round is a function of a moving average of the rewards and a stationary stochastic process. (Corollary 5.1);

• We apply the new framework to several demand response scenarios and numerically compare the theoretical and experimental regrets of our algorithm and the regret of a naive algorithm (Section 5.4).

5.2 Background

In this section, we introduce our notation and definitions. We index the rounds by $t$ and denote the time horizon by $\tau$. We consider the stochastic multi-armed bandit with $N$ arms where the player has to pick $1 \leq K^t \leq N$ arms, a round dependent integer. The reward obtained from playing arm $i$ is the i.i.d. bounded random variable $X_i$ for all $t$. Let $T_{i,t}$ be the number of times arm $i$ has been played after $t$ rounds. Let $\mu_i = E[X_i]$ and let $\hat{\mu}_{i,T}$ be the sample mean after the arm has been played $T_{i,t}$ times.

Let $\mathbb{K}$ be the set of all possible combinations of arms. Let $I_{K,t}$ denote the set of arms played at round $t$, i.e., the output of the algorithm at each round. Let $i$ be any set of arms with arbitrary cardinality and let $i_K$ be any set with cardinality equal to $K$. A superscripted star ($*$) denotes an optimal set of arms. At round $t$, the reward $X_i^t$ is observed for each arm $i \in I_t$. We denote the difference between the optimal expected reward and expected reward when the set of arm $i \in \mathbb{K}$ is played by $\Delta_i$, given by

$$\Delta_i = \sum_{i \in \text{i}^*} \mu_i - \sum_{i \in i} \mu_i,$$

where $i^*$ has the same cardinality as $i$.

We quantify the performance of our learning algorithm by its pseudo-regret [20] which differs from the regret definition previously used in the other chapters. We must extend the definition of pseudo-regret to allow for a variable number of arms to play in each time period. We begin with the standard definition of expected cumulative regret from [20]:

$$E[\text{Regret}_\tau] = \max_{t=1, \ldots, \tau} \left( \sum_{i \in I_{K,t}} X_i^t \right) - \left( \sum_{i \in I_{K,t}} X_i^t \right).$$

The pseudo-regret is obtained by swapping the expectation and the maximum:

$$\text{pRegret}_\tau = \max_{t=1, \ldots, \tau} \left( \sum_{i \in I_{K,t}} X_i^t \right) - \left( \sum_{i \in I_{K,t}} X_i^t \right).$$
Let \( A \) be the event \( \{K^1, K^2, \ldots, K^\tau, I^1_{K^1}, I^2_{K^2}, \ldots, I^\tau_{K^\tau}\} \). Equivalently, we have

\[
p\text{Regret}_\tau = \max_{i^t \in K^t_{K^t}} \mathbb{E}_A \left[ \mathbb{E}_X \left[ \sum_{t=1}^\tau \sum_{i \in I^t} X^t_i - \sum_{t=1}^\tau \sum_{i \in I^t_{K^t}} X^t_i \right] \mid A \right]
\]

The maximum is attained when \( i^t = i^*_K \), the optimal arm set, which gives our definition of pseudo-regret.

**Definition 5.1 (Pseudo-regret for MAB with variable multiple plays).**

\[
p\text{Regret}_\tau = \mathbb{E} \left[ \sum_{t=1}^\tau \sum_{i \in I^t_{K^t}} \mu_i - \sum_{t=1}^\tau \sum_{i \in I^t_{K^t}} \mu_i \right]
\]

Note that \( p\text{Regret}_\tau \) reduces to the pseudo-regret in [20] if \( K^t = 1 \) for all \( t \).

We seek a sub-linear upper bound for the pseudo-regret achieved by our learning algorithm. This ensures that the algorithm improves, i.e., learns, in each period, and therefore that it asymptotically approaches optimality.

### 5.3 Main Results

In this section, we present a UCB-like algorithm for MAB with Stochastic Plays (MAB-SP). We assume that \( K^t \) is sampled from a wide-sense stationary stochastic process taking values in \( \{1, 2, \ldots, N\} \) with mean \( \kappa \) and variance \( \sigma^2 \) for all \( t \). We then specialize to functions of a moving average of the reward.

Our algorithm, the UCB-Stochastic Plays (UCB-SP), is given in Figure 5.1. Unlike the UCB1 algorithm, the UCB-SP is allowed to play a different number of arms, \( K^t \) instead of having to play a single arm at each round. The number of arms to play is sampled by the player at the begin of each round and the player chooses the \( K^t \) arms with the largest indices \( \lambda_i \). As in UCB1, the indices \( \lambda_i \) are composed of the sample mean and an upper confidence bound. The sample mean encourages exploitation while the upper confidence bound encourages exploration.

**Theorem 5.1 (Regret bound of UCB-SP algorithm).** The pseudo-regret of UCB-SP is bounded above by

\[
p\text{Regret}_\tau \leq \left( \frac{6 \ln \tau}{\Delta^2_{\min}} (\sigma^2 + \kappa^2) + \frac{\kappa \pi^2}{3} + 1 \right) N \Delta_{\max}.
\]

**Proof.** Let \( i^*_K \) be an optimal set of arms in round \( t \). Note that the order in which the arms are played in a given round does not matter. We condition the pseudo-regret over the whole realization of the
Algorithm 5.1 Upper Confidence Bound with Stochastic Plays (UCB-SP) algorithm

1: Initialization: Play each arm once and update accordingly $\hat{\mu}_i$ and $T_i$
2: for $t = 1, 2, \ldots$ do
3: Receive number of arms to play this round, $K^t$.
4: Set
   \[
   \lambda_i = \hat{\mu}_{i,T_i} + \sqrt{\frac{3 \ln t}{2T_i}}
   \]
   for $i = 1, 2, \ldots, N$.
5: Let $I_{t,K^t}$ be the $K^t$ largest $\lambda_i$.
6: Play all arms $i \in I_{t,K^t}$.
7: Observe $X_i$ for each $i \in I_{t,K^t}$.
8: Update $\hat{\mu}_{i,T_i}$ and $T_i$ for each $i \in I_{t,K^t}$
9: end for

The stochastic process $K^t$. Denote the event $B = \{K^1, K^2, \ldots, K^\tau\}$. We rewrite the pseudo-regret (5.4) as

\[
\text{pRegret}_\tau = \mathbb{E} \left[ \mathbb{E} \left[ \sum_{t=1}^{\tau} \sum_{i \in I_{t,K^t}} \mu_i - \sum_{t=1}^{\tau} \sum_{i \in I_{t,K^t}} \mu_i \mid B \right] \right].
\]

Let $\tilde{T}_i^t$ be the number of times the set of arms $i$ has been selected at time $t$. Then,

\[
\text{pRegret}_\tau = \mathbb{E} \left[ \mathbb{E} \left[ \sum_{i \in \mathbb{K}} \mu_i - \sum_{i \in \mathbb{K}} \tilde{T}_i^\tau \left( \sum_{i \in \mathbb{K}} \mu_i \right) \mid B \right] \right],
\]

where $\mathbb{K} = \{i \in \mathbb{K} \mid \Delta_i > 0\}$. We re-write the previous relation as,

\[
\text{pRegret}_\tau \leq \mathbb{E} \left[ \mathbb{E} \left[ \sum_{i \in \mathbb{K}} \Delta_i \tilde{T}_i^\tau \mid B \right] \right],
\]

\[
\text{pRegret}_\tau \leq \Delta_{\max} \mathbb{E} \left[ \mathbb{E} \left[ \sum_{i \in \mathbb{K}} \tilde{T}_i^\tau \mid B \right] \right], \quad (5.6)
\]

where \(\Delta_{\max} = \max_{i \in \mathbb{K}} \Delta_i\).

Next, similarly to \[131\] and \[132\], define $T_i^t$ as follows. If $I_{t,K^t} \in \mathbb{K}$, then $T_i^t = T_i^{t-1} + 1$ where $i$ is the minimum index in the set $I_{t,K^t}$, and otherwise, $T_i^t = T_i^{t-1}$. In other words, if the set of arms picked at round $t$ is not optimal, then the counter $T_i^t$ where $i$ is the lowest index in the set of played arms is
incremented by one. Note that for all $t$, we have,

$$\sum_{i \in \mathbb{E}} \tilde{T}_i^t = \sum_{i=1}^{N} T_i^t. \quad (5.7)$$

Substituting (5.7) in (5.6), we have,

$$p\text{Regret}_\tau \leq \Delta_{\text{max}} E\left[ E\left[ \sum_{i=1}^{N} T_i^\tau \bigg| B \right] \right],
= \Delta_{\text{max}} \sum_{i=1}^{N} E\left[ T_i^\tau \bigg| B \right]. \quad (5.8)$$

Using Lemma 5.1 given in Section 5.6 to upper bound (5.8), we obtain the result:

$$p\text{Regret}_\tau \leq \Delta_{\text{max}} \sum_{i=1}^{N} \left( \frac{6 \ln \tau}{\Delta_{\text{min}}^2} \left( \sigma_K^2 + \kappa^2 \right) + \frac{\kappa \pi}{3} + 1 \right),
= \left( \frac{6 \ln \tau}{\Delta_{\text{min}}^2} \left( \sigma_K^2 + \kappa^2 \right) + \frac{\kappa \pi^2}{3} + 1 \right) N \Delta_{\text{max}}. \quad (5.9)$$

**Remark 5.1** (Bound comparison). If we set $K = 1$ for all $t$, then within a constant (6 instead of 8 in the first term) our bound reduces to the original regret for the UCB1 algorithm obtained in [36].

If $K^t$ is constant through time our regret bound reduces to $O(N^3 \ln \tau)$, which is smaller than the $O(N^4 \ln \tau)$ of [131] and larger than the $O(N^2 \ln \tau)$ of [132,133], and the $O(N^2 \ln \tau)$ of [134].

### 5.3.1 Function of moving sample mean

We now assume that $K^t$ is a function of an independent stationary stochastic process and of the moving sample average of the rewards. Our motivation for this case is as follows. A load aggregator is more likely to receive in each time period a power imbalance, a real number, than an integer number of arms. By making the number of arms, $K^t$, depend on the sample mean of the rewards, we can map a real-valued power imbalance to an integer number of arms and still invoke the regret bound.

**Corollary 5.1 (UCB-SP bound for function of moving sample mean).** Let $\hat{\mu}_i^d$ be the $d$-moving average of the $i$th arm, $C_i^t$ a non-negative independent stationary stochastic process and $f \in L^2 : \mathbb{R}^{N+1} \to \{1,2,\ldots,N\}$, a square-integrable function. If $K = f(\hat{\mu}_1^d, \hat{\mu}_2^d, \ldots, \hat{\mu}_N^d, C_t)$, and $d$ initialization rounds are used, the regret is bounded by

$$p\text{Regret}_\tau \leq \left( \frac{6 \ln \tau}{\Delta_{\text{min}}^2} E[f^2] + \frac{\pi^2}{3} E[f] + 1 \right) N \Delta_{\text{max}}. \quad (5.9)$$

**Proof.** Since $\hat{\mu}_i^d$ is a $d$-moving average and $X_i$ an i.i.d. process, then $d$ initialization rounds ensures that $\hat{\mu}_i^d$ is a stationary process for all $i$. Hence, the joint probability distribution function of independent variables $\hat{\mu}_i^d$ and $C_i$ is stationary and thus wide-sense stationary. Because $f \in L^2$, its first two moments exist. It then follows from Theorem 5.1 that the regret is bounded by (5.9). \qed
5.4 Application to Demand Response

We now demonstrate the MAB-SP framework on several demand response examples. On each example we compare the regret of UCB-SP to its theoretical upper bound and show that our regret analysis hold. For comparison, we also show the regret of a naive policy that in each rounds plays the $K^t$ arms with the highest sample mean. Note that no other algorithm has been extended to the case when the number of arms to play depends on an exogenous process.

5.4.1 Curtailment model

We first formulate the model of load curtailment that will be used in next subsections. The goal in each of the following examples is two-fold. Firstly, the demand response model must learn the curtailment performance of the load and, secondly, send curtailment signal to the best performing loads. This is motivated, for example, by regulation or peak-shaving [11, 13]. In both cases, the aggregator wants to mobilize a number of loads whose total change in power matches some target in each time period. The uncertain loads and limited feedback call for a multi-armed bandit based algorithm. Indeed, the player or aggregator can only obtain current information about the loads it deploys.

Let $N$ be the number of loads. We model the uncertainty of the load as in [50] and use the following definition of curtailment at round $t$ for load $i$,

$$c_{i,t} = \frac{\mu_i + w_{i,t}}{A}, \quad (5.10)$$

where $w_{i,t}$ is a zero-mean bounded random variable and $A$ is a normalization constant to ensure that $c_i \in [0,1]$ for all $t$ and $i$.

The mean load curtailments, $\mu_i$, is 0.45 for all loads except for five that are sampled from $U[0.5,0.55]$ for $i = 1, 2, \ldots, N$. Note that by making the sample means similar, we make it more difficult to learn the optimal policy. Each time a load is curtailed, the associated noise is sampled from $w_{i,t} \sim U[-0.4,0.4]$. We let $A = 1$ since no normalization is required in this case. We set $N = 20$ and $\tau = 10^5$ which corresponds to one minute rounds over an approximately 70 day period.

5.4.2 Up-regulation using UCB-SP

In this example, the objective is to lower power demand when it exceeds the scheduled power generation. This is referred to some system operators as up-regulation, and helps maintain grid frequency [137]. In practice, the curtailment signal could be a portion of the Area Control Error (ACE) when it is greater than zero.

The number of loads to curtail in each time period is an i.i.d. zero-truncated Poisson random variable $K^t \sim \text{Poisson}_{>0}(\lambda)$. In this case the regret of the UCB-SP algorithm is bounded by

$$p\text{Regret}_\tau \leq \frac{6 \ln \tau}{\Delta_{\min}} \left( \frac{\lambda + \lambda^2}{1 - e^{-\lambda}} \right) + \frac{\pi^2}{3} \left( \frac{\lambda}{1 - e^{-\lambda}} \right) + 1 \right) N\Delta_{\max}.$$ 

Figure 5.1 shows the upper bound and experimental regret for UCB-SP when $\lambda = 0.5$. We see that the experimental regret increases sublinearly and outperforms the naive policy. The experimental regret also outperforms the bound by a factor of 40.
Chapter 5. The Multi-armed Bandit with Stochastic Plays

5.4.3 Markov Chain-based regulation using UCB-SP

We now assume that $K^t$ evolves according to a Markov chain with transition probability $P$. The transition probabilities in $P$ are chosen to reflect the ratio from one time instant to another of the number of loads needed to curtail the power equivalent of the historical ACE data. Note that autocorrelation for ACE decreases significantly across more than two time periods.

The Markov Chain state $k_m$ takes on values 1, 2, 3, 4 and 5 respectively for $m = 1, \ldots, 5$, respectively, and represents the number of loads to deploy. It is graphically represented in Figure 5.2. We use the load model presented in 5.4.1. Lastly, we present numerical results using the transition probability matrix $P$ given in (5.11) and an initial value sampled from $\pi$. 
5.4.4 Curtailment using UCB-SP for functions of a moving sample mean

In this last example, we let the number of curtailed loads be a function of the sample mean and an independent stationary stochastic process. The i.i.d. stochastic process $C^t$ is the power curtailment required by the aggregator at round $t$. In each time period $C^t$ is drawn from a normal distribution with mean $\mu_{ACE}$ and variance $\sigma_{ACE}^2$. Negative values are resampled. These values are the mean and the variance of historical one-minute ACE data [138] scaled down by a factor of 1000.

The aggregator uses the curtailment signal and the moving sample mean to compute the number of
loads to curtail. Specifically, it chooses $K^t$ such that the sum of the $K^t$ largest running averages of the rewards is greater than the curtailment signal:

$$K^t = \arg\min_{i \subseteq \{0, 1, \ldots, N\}} \text{card}(i)$$

$$\text{subject to } \sum_{i \in i} \hat{\mu}_d \geq C^t,$$

where $\text{card}(\cdot)$ denotes the cardinality of the argument, $\hat{\mu}_d$ is the $d$-moving sample mean of the $i^{th}$ load at round $t$. We write (5.12) more concisely as $K^t = f(\hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_N, C^t)$. Since $f$ only returns values in $\{1, 2, \ldots, N\}$, its mean and its variance are finite and $f \in L^2$. Hence, in this example, Corollary 5.1 provides an upper bound on the regret of UCB-SP. Note, however, that the sample means appearing in the sum in (5.12) may not correspond to the loads that are actually dispatched by the UCB-SP algorithm. In this regard, (5.12) is an approximate mapping from the size of the imbalance, $C^t$, to the number of loads to curtail.

We simulate UCB-SP in this example with $d = 10$ and $\mu_{\text{ACE}}$ and $\sigma^2_{\text{ACE}}$ computed from PJM’s ACE signal [138]. The mean and the variance of $f$ were computed using Monte Carlo simulations. The sublinear experimental pseudo-regret, its performance against the naive policy and its bound are shown in Figure 5.4. Again UCB-SP outperforms the bound and the naive policy.

## 5.5 Conclusion

We have extended the stochastic multi-armed bandit with multiple plays to the case where the number of plays can change in time according to a wide-sense stationary stochastic process. This extension is
motivated by the uncertainty of loads providing demand response and the time-varying needs of power systems. We obtained a sublinear regret bound for the algorithm that simplifies to existing bounds in the literature when the number of arms is constant. We provided several examples that demonstrate the utility of our results.

5.6 Proof of Lemma 5.1

Lemma 5.1. Let $K_t \in \{1, 2, \ldots, N\}$ for all $t$ be a wide-sense stationary stochastic process with mean $\kappa$ and variance $\sigma^2_{K_t}$. Then, for the UCB-SP algorithm, we have

$$\mathbb{E} \left[ \mathbb{E} \left[ T_i \mid B \right] \right] \leq \frac{6 \ln \tau}{\Delta_{\min}^2} (\sigma^2_{K_t} + \kappa^2) + \frac{\kappa \pi}{3} + 1.$$

Proof. We first bound the inner expected value, $\mathbb{E} \left[ T_i \mid B \right]$. We adapt the proof of Theorem 2.1 in Bubeck & Cesa-Bianchi [20] which is itself based on the original proof of [36] to the case where $K_t \geq 1$ and is a wide-sense stationary stochastic process. We also use proof techniques from [131] and [132]. The proof in [20] is based on three inequalities. Let $I_{K_t} = i_{K_t} \neq i^*_{K_t}$ be the arms chosen by the algorithm at round $t$. In other words, consider a round where the optimal set of arms is not picked. Then, possible reasons for this outcome are

$$\sum_{i \in I_{K_t}} \left( \hat{\mu}_{i,T_i} - 1 + \sqrt{\frac{z \ln t}{T_i^{t-1}}} \right) \leq \sum_{i \in I_{K_t}} \mu_i,$$  \hspace{1cm} (5.13)

or

$$\sum_{i \in I_{K_t}} \hat{\mu}_{i,T_i} > \sum_{i \in I_{K_t}} \left( \mu_i + \sqrt{\frac{z \ln t}{T_i^{t-1}}} \right).$$  \hspace{1cm} (5.14)

Also, because $I_{K_t} \neq I^*_{K_t}$, we have,

$$\sum_{i \in I^*_{K_t}} \left( \hat{\mu}_{i,T_i} - 1 + \sqrt{\frac{z \ln t}{T_i^{t-1}}} \right) \leq \sum_{i \in I^*_{K_t}} \left( \hat{\mu}_{i,T_i} - 1 + \sqrt{\frac{z \ln t}{T_i^{t-1}}} \right).$$

If we assume that (5.13) and (5.14) are false, then

$$\sum_{i \in I_{K_t}} \mu_i < \sum_{i \in I_{K_t}} \hat{\mu}_{i,T_i} + \sum_{i \in I_{K_t}} \sqrt{\frac{z \ln t}{T_i^{t-1}}}$$

$$< \sum_{i \in I_{K_t}} \left( \mu_i + \sqrt{\frac{z \ln t}{T_i^{t-1}}} \right) + \sum_{i \in I_{K_t}} \sqrt{\frac{z \ln t}{T_i^{t-1}}}.$$

Therefore,

$$\sum_{i \in I_{K_t}} \mu_i - \sum_{i \in I_{K_t}} \mu_i = \Delta_{I_{K_t}} < 2 \sum_{i \in I_{K_t}} \sqrt{\frac{z \ln t}{T_i^{t-1}}}.$$
Defining $\Delta_{\text{min}}$ as,

$$\Delta_{\text{min}} = \min_{i \in K} \Delta_i,$$

we have,

$$\Delta_{\text{min}} < 2 \sum_{i \in 1_{K^t}} \sqrt{\frac{z \ln t}{T_i^{t-1}}}.$$ 

Note that $T_i^t \leq T_i^t$ for all $t$. Then

$$\Delta_{\text{min}} < 2K^t \sqrt{\frac{z \ln t}{T_i^{t-1}}}$$  \hspace{1cm} (5.15)

for all $i \in 1_{K^t}$. Rearranging (5.15), we have

$$T_i^{t-1} < \frac{4 (K^t)^2 z \ln t}{\Delta_{\text{min}}}.$$  \hspace{1cm} (5.16)

Since $T_i^{t-1}$ must be integer, we write

$$T_i^{t-1} < b(K^t),$$  \hspace{1cm} (5.17)

our third inequality, where,

$$b(K^t) = \left[ \frac{4 (K^t)^2 z \ln \tau}{\Delta_{\text{min}}^2} \right].$$  \hspace{1cm} (5.18)

$b(K^t)$ is a random variable that only depends on $K^t$.

The inequality (5.17) only holds if (5.13) and (5.14) are both false. However, the three inequalities cannot be simultaneously false. If we assume that (5.13) to be false, we have

$$\sum_{i \in 1_{K^t}} \left( \hat{\mu}_{i,T_i^{t-1}} + \sqrt{\frac{z \ln t}{T_i^{t-1}}} \right) > \sum_{i \in 1_{K^t}} \mu_i = \sum_{i \in 1_{K^t}} \mu_i + \Delta_i^{K^t}$$  \hspace{1cm} (5.19)

Assuming that (5.17) is false implies that

$$\Delta_{\text{min}} \geq 2K^t \sqrt{\frac{z \ln t}{T_i^{t-1}}}.$$
Since \( i = \arg \min_j j \), we have

\[
\Delta_{k_t} \geq 2K_t \sqrt{\frac{z \ln t}{T_{i_t}^{T-1}}}
\]

\[
= 2 \sum_{i \in i_{k_t}} \sqrt{\frac{z \ln t}{T_{i_t}^{T-1}}}
\]

\[
\geq 2 \sum_{i \in i_{k_t}} \sqrt{\frac{z \ln t}{T_{i_t}^{T-1}}}
\]

\[\text{(5.20)}\]

because \( T_{i_t} \leq T_{i_t} \) for all \( i \). Substituting (5.20) in (5.19) gives

\[
\sum_{i \in i_{k_t}} \left( \hat{\mu}_{i,T_{i_t}^{t-1}} + \sqrt{\frac{z \ln t}{T_{i_t}^{T-1}}} \right) > \sum_{i \in i_{k_t}} \mu_i + 2 \sum_{i \in i_{k_t}} \sqrt{\frac{z \ln t}{T_{i_t}^{T-1}}}
\]

\[
\sum_{i \in i_{k_t}} \left( \hat{\mu}_{i,T_{i_t}^{t-1}} + \sqrt{\frac{z \ln t}{T_{i_t}^{T-1}}} \right) > \sum_{i \in i_{k_t}} \left( \mu_i + \sqrt{\frac{z \ln t}{T_{i_t}^{T-1}}} \right) + \sum_{i \in i_{k_t}} \sqrt{\frac{z \ln t}{T_{i_t}^{T-1}}}
\]

Then assuming that (5.14) is false, we have

\[
\sum_{i \in i_{k_t}} \left( \hat{\mu}_{i,T_{i_t}^{t-1}} + \sqrt{\frac{z \ln t}{T_{i_t}^{T-1}}} \right) > \sum_{i \in i_{k_t}} \left( \hat{\mu}_{i,T_{i_t}^{t-1}} + \sqrt{\frac{z \ln t}{T_{i_t}^{T-1}}} \right).
\]

\[\text{(5.21)}\]

However, since \( i_{k_t} \neq i_{k_t} \), (5.21) cannot hold and thus (5.13), (5.14), and (5.17) cannot be all false at the same time.

The probability that (5.13) or (5.14) holds can be computed using approaches developed in [20], [131] and [132]. We now compute \( \Pr \{ \text{(5.14)} | K_t \} \). The same approach can be used for (5.13) and thus only the result is given. To determine \( i_{k_t} \), the full history \( \{ K^1, K^2, \ldots, K_t \} \) is required. We have

\[
\Pr \{ \text{(5.14)} | B \} = \Pr \left[ \sum_{i \in i_{k_t}} \hat{\mu}_{i,T_{i_t}^{t-1}} - \sum_{i \in i_{k_t}} \mu_i \geq \sum_{i \in i_{k_t}} \sqrt{\frac{z \ln t}{T_{i_t}^{T-1}}} \middle| B \right]
\]

\[
\leq \sum_{i \in i_{k_t}} \Pr \left[ \hat{\mu}_{i,T_{i_t}^{t-1}} - \mu_i \geq \sqrt{\frac{z \ln t}{T_{i_t}^{T-1}}} \middle| B \right]
\]

This is then upper-bounded [132] by

\[
\Pr \{ \text{(5.14)} | B \} \leq \sum_{i \in i_{k_t}} \sum_{j=1}^{t-1} \Pr \left[ T_{i_t}^{t-1} = j \middle| B \right] \Pr \left[ \hat{\mu}_{i,T_{i_t}^{t-1}} - \mu_i \geq \sqrt{\frac{z \ln t}{T_{i_t}^{T-1}}} \middle| T_{i_t}^{t-1} = j, B \right]
\]

\[
\leq \sum_{i \in i_{k_t}} \sum_{j=1}^{t} \Pr \left[ \hat{\mu}_{i,T_{i_t}^{t-1}} - \mu_i \geq \sqrt{\frac{z \ln t}{T_{i_t}^{T-1}}} \middle| T_{i_t}^{t-1} = j \right]
\]
Hoeffding’s inequality yields

\[
\Pr \left[ 5.14 \mid B \right] \leq \sum_{i \in I_{K^t}} \sum_{j=1}^{t} e^{-2T_{ij} - 1 \left( \frac{\ln T}{T_{ij}} \right)^2} \leq \sum_{i \in I_{K^t}} \sum_{j=1}^{t} \frac{1}{T_{ij}} = K^t \sum_{i \in I_{K^t}} \sum_{j=1}^{t} \frac{1}{T_{ij}}
\]

Observe that the upper bound is independent of all prior values of \( K^t \) given \( K^t \). We denote this upper-bound by \( \Pr_{ub} \left[ 5.14 \mid K^t \right] \). Similarly, we obtain an upper bound, \( \Pr_{ub} \left[ 5.13 \mid K^t \right] \), for \( \Pr \left[ 5.13 \mid B \right] \):

\[
\Pr \left[ 5.13 \mid B \right] \leq \frac{K^t}{T_{ij}}.
\]

We can now bound the inner expected value. Let \( \mathbb{I}_x \) be one if condition \( x \) is true and zero otherwise. Then,

\[
\mathbb{E} \left[ T_{ij} \mid B \right] = \mathbb{E} \left[ \sum_{t=1}^{T} \mathbb{I}_{i = \arg \min_{j \in I^t} j} \mid B \right] = \mathbb{E} \left[ \sum_{t=b(K^s) + 1}^{T} \mathbb{I}_{i = \arg \min_{j \in I^t} j \text{ & } 5.17 \text{ is false}} \mid B \right] \leq \mathbb{E} \left[ b(K^s) + \sum_{t=b(K^s) + 1}^{T} \mathbb{I}_{i = \arg \min_{j \in I^t} j \text{ & } 5.17 \text{ is false}} \mid B \right] = b(K^s) + \mathbb{E} \left[ \sum_{t=b(K^s) + 1}^{T} \mathbb{I}_{\{5.13 \text{ or } 5.14 \text{ holds}} \mid B \right] \leq b(K^s) + \sum_{t=b(K^s) + 1}^{T} \left( \Pr \left[ \{5.13 \} \mid B \right] + \Pr \left[ \{5.14 \} \mid B \right] \right) \leq b(K^s) + \sum_{t=1}^{\infty} \left( \Pr_{ub} \left[ 5.13 \mid K^t \right] + \Pr_{ub} \left[ 5.14 \mid K^t \right] \right).
\]

We use \( 5.24 \) to upper-bound the outer expectation. \( K^t \) is a wide-sense stationary stochastic process and hence its mean, \( \kappa \), is constant and its auto-covariance is only a function of the time difference between, \( c \). Hence, for \( c = 0 \), the variance, \( \sigma^2 \), of the process is constant for all \( t \). We thus have

\[
\mathbb{E} \left[ \mathbb{E} \left[ T_{ij} \mid B \right] \right] \leq \mathbb{E} \left[ b(K^s) + \sum_{t=1}^{\infty} \left( \Pr_{ub} \left[ 5.13 \mid K^t \right] + \Pr_{ub} \left[ 5.14 \mid K^t \right] \right) \right].
\]
We set $z = 3/2$ in $\Pr_{ub}^t[K^t]$ and $\Pr_{ub}^t[K^t]$. Then

\[
E\left[ E\left[ T_i^\tau \mid B \right] \right] \leq E\left[ \frac{6(K^\tau)^2 \ln \tau}{\Delta_{\min}^2} + 1 + \sum_{t=1}^{\infty} \frac{2K^t}{t^2} \right],
\]

\[
= \frac{6E[(K^\tau)^2 \ln \tau]}{\Delta_{\min}^2} + 1 + \sum_{t=1}^{\infty} \frac{2E[K^t]}{t^2},
\]

\[
= 6 \ln \tau \frac{(\sigma^2 + \kappa^2)}{\Delta_{\min}^2} + 1 + \kappa \sum_{t=1}^{\infty} \frac{2}{t^2},
\]

\[
= 6 \ln \tau \frac{(\sigma^2 + \kappa^2)}{\Delta_{\min}^2} + 1 + \frac{\kappa \pi^2}{3}.
\]
Chapter 6

Conclusion

In this thesis, we designed performance-guaranteed online optimization-based approaches for demand response. Our methods enabled load aggregators to better manage uncertainty in demand response.

We first formulated a power setpoint tracking demand response model using online convex optimization. We considered different types of feedback from the loads. We applied our methods to two types of flexible loads: thermostatically controlled loads and electric vehicles. We then formulated a two-level algorithm for multi-energy buildings to meet their energy requirements while providing ancillary services. We presented a case-study in which we applied our two-level approach to a building located in Melbourne, Australia. Next, we added side information by means of an inexact gradient of the current round’s loss function to the standard online convex optimization setting. The predictive online convex optimization yields strict improvement in performance when compared to standard OCO when certain conditions are met. An application of the predictive OCO was presented in the context of demand response and showed it outperforms standard OCO. Lastly, we extended the multi-armed bandit framework to deal with a number of arms to play determined by a stochastic process. We utilized this extension for curtailment when the power deficit was changing in time.

We conclude this thesis by discussing some potential avenues for future work. In this thesis, we looked at how electric flexible loads can assist the power system operations. It would be of interest to implement our approaches on real loads. Second, we assumed that enrolled loads were fully available for demand response. In future work, we would like to investigate the compensation that attracts these loads to DR programs for future real-world implementation and to investigate pricing or reward mechanisms for loads participating in direct control demand response.

In Chapter 3, we discussed briefly online convex optimization with time-varying constraints. This is an interesting extension to the standard OCO framework since it has the potential to include a wider range of problems and permits fully dynamic modeling. To improve current algorithms for OCO with time-varying constraints and to reduce their requirements, e.g., on the cumulative variation and cumulative constraints variation terms (see Chapter 3, (3.51) and (3.52)), an interior-point method could be developed for OCO. Second-order updates have already been shown to improve regret bounds in the static case [31] and could thus potentially reduce the requirements on the variation terms in the dynamic case. Moreover, Newton step-like and barrier functions can directly handle equality and inequality constraints [139,140] instead of necessitating an extra penalty term as we used in Chapter 3.

An OCO with time-varying constraints algorithm as described previously would have further ap-
plication in power systems with high penetration of renewables. Specifically, this approach could be used in the future to deal with the optimal power flow problem \cite{141,142} under uncertainty. In a grid with a high penetration of renewable, real-time optimal power flow takes further importance as it could replace the unit commitment problem \cite{5}. This is because renewables have very low inertia and could be dispatch in real-time with respect to the power systems timescale \cite{143,144}. Such a change reduces the complexity of the system's operations as unit commitment is a combinatorial scheduling problem \cite{5}. Using an interior-point method-based OCO algorithm, the system operator could manage, in real-time, conventional generation and uncertain, time-varying demand constraints (equality constraints) and uncertain, time-varying renewable generation and its curtailment (inequality constraints) and minimize the generation costs.

Online optimization is a powerful tool to handle uncertainty since its performance is guaranteed by a low-regret analysis. Uncertainty is present in all power system problems. In this thesis, we only discussed demand response applications. Several other applications could also be found within the power system field, for example, the aforementioned optimal power flow. Further applications could lead to additional improvements in system operations or planning and reliability of the grid.
Bibliography


[75] ——, “Graph implementations for nonsmooth convex programs,” in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, 2008, pp. 95–110.


Bibliography


Bibliography


