Real-Time Data-Driven Optimization Algorithms for Modern Power Systems

by

Ana Maria Ospina Sierra

B.S. Electronic Engineering, Universidad de Los Andes, 2011

B.S. Electrical Engineering, Universidad de Los Andes, 2011

M.S. Electrical Engineering, University of Colorado Boulder, 2021

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Electrical, Computer, and Energy Engineering

2023

Committee Members:

Professor Emiliano Dall'Anese, Chair

Professor Stephen Becker

Professor Xudong Chen

Professor Lucy Pao

Professor Andrea Simonetto

Ospina Sierra, Ana Maria (Ph.D. Systems and Controls)

Real-Time Data-Driven Optimization Algorithms for Modern Power Systems

Thesis directed by Professor Emiliano Dall'Anese

Power systems have experienced significant transformations in recent years driven by the integration of new technologies, such as renewable generation and distributed energy resources (DERs). However, the variability of renewable generation and uncontrollable loads, along with the increasing grid-to-user interactions, pose severe challenges to the grid operation. Therefore, ensuring a reliable, safe, and optimal real-time operation of *modern power systems* requires new scalable optimization approaches that can cope with uncertainty and learn models in real-time. Towards this end, this dissertation presents contributions to the field of time-varying optimization, focusing on applications where renewable generation and controllable DERs interact with the grid.

The dissertation is divided into two parts. Part I (Chapter 2 and 3) investigates time-varying optimization problems associated with a physical system and human-in-the-loop interactions. In this part, we study two settings. First, we consider a cost comprising a partially known time-varying function capturing performance metrics of the system, as well as unknown functions associated with users interacting with the physical system. Second, we feature an optimization problem where the objective is to minimize cost functions related to the individuals' preferences, subject to time-varying constraints that capture the physical or operational limits of the network. Based on these time-varying optimization problems, we develop learning-based distributed online optimization algorithms. Our focus lies in the synthesis of first-order online methods, where feedback from the user is leveraged to learn the unknown functions concurrently with the execution of the online algorithm, and measurements of the output of the system are used to estimate the gradient and evaluate the partially known engineering functions.

Part II (Chapter 4 - 7) focuses on optimization and control techniques for different applications where power systems interact with controllable DERs. First, we present an application of the

theoretical framework of Part I to the real-time management of DERs. Second, we present a demand response application on commercial buildings. In this case, we introduce a predictive controller for a grid-interactive multi-zone building where the temperature dynamics are learned via a supervised learning technique. This controller uses the learned dynamics to solve a multi-objective problem to guarantee occupants' comfort and energy efficiency during normal conditions and demand response events. Third, we study the estimation of sensitivity matrices in power grids with applications in transmission and distribution systems. By leveraging a low-rank approximation of certain classes of sensitivity matrices, we propose an online proximal-gradient algorithm based on a robust nuclear norm minimization problem to estimate linear sensitivities from measurements. Finally, we address a problem related to the charging schedule of electric vehicles (EVs). We consider a scenario where a ride-service provider utilizes a 100%-EV fleet to serve customers, while a power utility company aims to maximize the utilization of renewable generation at specific charging stations. To achieve this, we propose a novel mechanism that encourages EV charging during hours of high renewable generation while minimizing the impact on the quality of service for the ride-service provider.

Dedication

To my present and future family.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to Prof. Emiliano Dall'Anese. Thank you for your exceptional patience, guidance, and mentorship. I am grateful for the invaluable opportunity to be a part of your research group and work alongside you. This professional and personal journey has been enlightening and is one of my life's most remarkable experiences.

I am grateful to my committee. I would like to extend my sincere appreciation to Prof. Andrea Simonetto. Your insightful comments and perspectives have played a vital role in shaping the outcome of this research. I would also like to thank Prof. Xudong Chen, for your patience during class. Furthermore, I express my thanks to Prof. Stephen Becker and Prof. Lucy Pao for your constructive feedback and suggestions. My gratitude extends to my mentors at NREL and Sandia, Andrey Bernstein, Yue Chen, Manuel Garcia, and Felipe Wilches, for providing me with an outstanding career opportunity. To all the exceptional collaborators and friends I have made along this journey — Dr. Tamara Becejac, Daniel, Nicola, Lily, Elisabetta, Marija, Seung, Gianluca, Yiting, and Killian — I am immensely thankful for your support, motivation, enjoyable moments, and countless cups of coffee.

Lastly, and most importantly, I want to thank my family for your unconditional support. To Santi and Yelas, and now Lore and Omar, I am grateful for the loving home you create, no matter where we find ourselves. To Cata, Simon, Sophie, and Martin, thank you for being our support during our time in Colorado. To my parents, I would not be where I am today without the innumerable opportunities you have provided me. And to Jorge, I am forever grateful for always believing in me and working with me to make this life our reality.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

1.1 Motivation

Ensuring minimal environmental impact from our power system is crucial and is one of the most significant challenges in combating climate change and global warming [58]. To achieve this goal, our power system is actively shifting towards renewable energy sources, such as wind turbines and photovoltaic solar panels, and adopting decentralized energy production and consumption through distributed energy resources (DERs) in distribution and transmission grids. In addition, we are promoting a more flexible consumption pattern through demand response (DR) programs [79]. As a result of these initiatives, the operation of power systems is undergoing a profound transformation, driven by the introduction of new controllable elements such as power converters in DERs and flexible loads in virtual power plants. Pervasive sensing technologies such as smart meters, phasor measurement units, supervisory control, data acquisition, and advanced communication networks enable fast sampling and access to rich, high-dimensional data. Additionally, the collection of DERs, which can be controlled directly by users, is imposing new challenges related to the real-time interaction between users (via human preferences) and the power consumption of devices, that may affect the grid operation.

In general, the integration of these new technologies brings numerous benefits, such as the reduction in operational costs while simultaneously enhancing the efficiency, reliability, and sustainability of existing infrastructures. However, the abundance of high-dimensional data and the unpredictability in modern power systems poses severe challenges to the grid operation [29]. There-

fore, it is crucial to develop new optimization and learning strategies to fully unlock the potential of these technologies in the grid. In this context, strategies for managing DERs, such as DR programs, hold promise in increasing the flexibility and efficiency of power systems. They enable controllable devices to provide services across various time-scales, ranging from real-time frequency and voltage support to a slower time-scale, such as peak-shifting service [96, 124, 95, 11]. Typically, real-time management of DERs involves the balance between system-level operational objectives and (dis)satisfaction of the device's owner [67, 97]; e.g., deviations from a preferred indoor temperature or charging profile of the electric vehicle. This aspect makes the actual implementation of strategies for real-time management of DERs challenging.

Traditional optimization approaches, commonly known as *batch* optimization, can be employed to solve problems related to the management of DERs in power grids. These approaches are typically categorized as *offline* optimization and often involve simplifying complex problems. However, when it comes to real-time control and optimization of modern power systems, solving an optimization problem *offline* (assuming complete knowledge of the problem data and structure) and implementing the optimization output in a *feedforward* approach does not adequately address the current challenges in power systems. In contrast, *feedback*-based *static* optimization approaches, such as those studied in [27, 40, 49, 74], have been explored in the context of constrained nonlinear optimization to steer a system towards an optimal operational state. These feedback-based approaches offer several advantages, including increased robustness, reduced reliance on model information, minimized computational effort, and elimination of the need for exogenous data.

One prominent approach in the context of *feedback*-based optimization is the integration of time-varying settings. These types of problems are centered on the idea that optimization problems vary or evolve over time. In recent years, this topic has garnered interest due to its relevance to various applications in the real-time operation and control of power systems. The primary objective is to develop online algorithms that can effectively *track* the solution for time-varying optimization problems while ensuring robust performance. In this dissertation, our focus lies on time-varying convex optimization problems from a control theory perspective in the sense of

[146, 52, 130, 148, 61, 132]. These works describe time-varying optimization as a tracking problem, where an optimization algorithm is seen as a time-varying solution map. Various metrics have been proposed to evaluate the performance of such algorithms, as seen in works such as [80, 113, 43], where tracking error bounds or regret analysis (static or dynamic) are primarily utilized. When incorporating system measurements into the time-varying solution map, we adopt a *feedback online* approach, as studied in previous works like [89, 19, 53, 155], which forms the core of the work presented here.

This dissertation focuses on developing feedback-based online strategies that strike a balance between power systems operation and the power consumption of DERs, taking into account the preferences of DERs' owners. The proposed strategies aim to overcome the following challenges:

C1) Users' function uncertainty: The functions that model the users' preferences regarding the power consumption (or operating point) of DERs may be unknown, and the available models may be inaccurate.

C2) Pervasive metering: Solving and implementing control and optimization strategies may require real-time measurements of the powers of non-controllable devices or loads at multiple locations, which is a challenging task in power systems.

C3) Network model uncertainty: Obtaining perfect knowledge of power systems models is highly complex. Therefore, estimating parameters or interactions of the grid with different DERs is a crucial task.

C4) Intermittent feedback: When the performance of optimization or control strategies depends on system measurements, such as feedback settings, they must also consider scenarios where these measurements can be noisy and intermittent during the execution of the algorithms.

The work presented in this dissertation is motivated by the challenges associated with integrating and optimally operating renewable resources and DERs in power systems, as described above. The research questions addressed in the context of modern power systems can be summarized as follows:

- How can we incorporate human preferences regarding the specific power consumption of

DERs while simultaneously achieving power system objectives? Additionally, how can we implement feedback-based strategies that effectively handle intermittent measurements?

• Is it possible to reach an agreement among multiple users who share controllable DERs in network systems?

• How can we learn certain parameters of power systems from corrupted data in a real-time setting? Can we utilize controllable DERs to provide services to the grid?

1.2 Contributions and Dissertation Outline

The contributions of this dissertation can be classified into three groups:

Data-Driven Optimization and Learning Algorithms: We proposed an online data-driven feedback-based projected gradient descent method that incorporates intermittent updates and noisy gradients. We provided new bounds for the expected error incurred in each iteration of the algorithmic steps. Specifically, we modeled missing measurements as Bernoulli random variables. Additionally, we provided new bounds with high probability, derived by utilizing a sub-Weibull distribution to characterize the error that affects the gradient [119].

We also developed an online algorithm based on a primal-dual method to solve a consensus-based time-varying network optimization problem. The algorithm is implemented online and modified to accommodate measurements and learned functions. We proposed a consensus-based formulation that leads to a distributed online algorithm in which users minimize the sum of their learned functions without revealing their preferences or feedback. We demonstrated how our framework models concurrent learning approaches, where functional evaluations are utilized to learn the unknown cost. For the performance analysis, we treated the algorithm as an inexact online primal-dual method, taking into account errors arising from the estimation of the unknown function and errors when utilizing measurements from the system in a feedback configuration [122].

Optimization and Control for Distributed Energy Resources: We have a range of applications related to the real-time management of controllable distributed energy resources.

• *Demand response:* We tested the performance of our online data-driven optimization algorithms

in cases where (part of) the cost is estimated via Gaussian Processes (GPs), shaped-constrained GPs, and feedforward neural networks.

• *Building temperature control:* We formulated a GP-based predictive control scheme that balances operational objectives, demand response objectives, and users' thermal discomfort costs. To solve this optimization problem, we implemented a primal-dual projected-gradient method. We demonstrated how to periodically train the GP model to adaptively incorporate changes in the environment, such as occupants' behaviors or weather conditions. Specifically, for the GP characterization, we employed a composite covariance function that combines a squared exponential kernel with a locally periodic kernel, capturing specific properties of the building dynamics behavior [120].

• *Electric vehicle charging schedule:* We proposed a new mechanism to facilitate interactions between power utility and ride-sharing companies. This mechanism enables the power utility company to issue demand response signals, representing financial incentives tied to specific renewable generation profiles at charging locations, aiming to promote the use of available renewable energy [128].

Learning for Transmission and Distribution Power Systems: In this area, we have proposed a method for estimating linear sensitivities in power grids by utilizing a nuclear norm minimization approach and sparsity-promoting regularization functions. Our approach is motivated by the observation that certain classes of sensitivity matrices can be effectively approximated with low rank. This methodology is applicable to the estimation of various sensitivities at both the transmission and distribution levels. The method is capable of identifying outliers caused by faulty sensors and is not hindered by missing measurements. We have developed an online proximal-gradient algorithm to estimate sensitivity matrices in real-time, allowing operators to maintain up-to-date information on sensitivities under dynamic operating conditions [121].

This dissertation is organized into two parts:

Part I: Theoretical Framework

• **Chapter 2:** This chapter introduces an online projected gradient descent (PGD) algorithm, and a theoretical contribution based on a variation of the online PGD algorithm with intermittent

updates. The work discussed in this chapter corresponds to the publications titled "Personalized Demand Response via Shape-Constrained Online Learning" by Ana M. Ospina, Andrea Simonetto and Emiliano Dall'Anese [123], and "Feedback-Based Optimization with Sub-Weibull Gradient Errors and Intermittent Updates" by Ana M. Ospina, Nicola Bastianello and Emiliano Dall'Anese [119].

• **Chapter 3:** This chapter presents a theoretical contribution based on a consensus-based online primal-dual algorithm. The work in this chapter corresponds to the publication titled "Time-Varying Optimization of Networked Systems with Human Preferences" by Ana M. Ospina, Andrea Simonetto and Emiliano Dall'Anese [122].

Part II: Optimization and Control for Distributed Energy Resources

• **Chapter 4:** This chapter presents applications of the theoretical framework presented in Chapter 2 and 3 in the real-time management of distributed energy resources, where, the first-order methods shown above are used to incorporate learning, infrequent feedback, and consensus-based strategies in demand response scenarios [123, 119, 122].

• **Chapter 5:** An application of building temperature control for demand response is shown in this chapter. The work in this chapter corresponds to the publication titled "Learning-Based Demand Response in Grid-Interactive Buildings via Gaussian Processes" by Ana M. Ospina, Yue Chen, Andrey Bernstein and Emiliano Dall'Anese [120].

• **Chapter 6:** An application of online estimation of linear sensitivity factors in transmission and distribution power systems is presented in this chapter. The work discussed here corresponds to the preprint titled "Data-Driven and Online Estimation of Linear Sensitivity Distribution Factors: A Low-rank Approach" by Ana M. Ospina and Emiliano Dall'Anese [121].

• **Chapter 7:** A renewable-based electric vehicle charging schedule strategy is studied in the last chapter. The work presented here is part of the preprint titled "Towards the Decarbonization of the Mobility Sector: Promoting Renewable-Based Charging in Green Ride-Sharing" by Elisabetta Perotti, Ana M. Ospina, Gianluca Bianchin, Andrea Simonetto and Emiliano Dall'Anese [128].

1.3 Notation

- The set of (whole) natural numbers is denoted as $(\mathbb{N} \cup \{0\})$ $\mathbb{N}$.
- The set of (nonnegative) real numbers is denoted as $(\mathbb{R}_{\geq 0})$ $\mathbb{R}$.
- Upper-case (lower-case) boldface letters will be used for matrices (column vectors).
- Transposition is denoted by $(\cdot)^\top$.
- For a given column vector $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\| := \sqrt{\mathbf{x}^\top \mathbf{x}}$.
- For a given column vector $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|_1 := \sum_{i=1}^n |x_i|$.
- A vector of zeros is represented by $\mathbf{0}$, with the corresponding dimensions.
- A vector of ones by $\mathbf{1}$ is represented by, with the corresponding dimensions.
- $\mathcal{O}(\cdot)$ refers to the big O notation, whereas $o(\cdot)$ refers to the little-o notation.
- For a random variable $X \in \mathbb{R}$, $\mathbb{E}[X]$ denotes the expected value of X, and $\mathbb{P}[X \leq \epsilon]$ denotes the probability of X taking values smaller than or equal to ϵ.
- For a continuously-differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, the gradient is denoted by $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$.
- Given a convex set $\mathcal{X} \subseteq \mathbb{R}^n$, $\text{proj}_{\mathcal{X}}(\mathbf{z})$ denotes the projection of $\mathbf{z} \in \mathbb{R}^n$ onto $\mathcal{X}$.
- $\mathcal{I}$ is the identity mapping.
- The Euler's number is denoted by e.
- The max operator is defined as $[x]^+ := \max\{0, x\}$.
- The diameter of $\mathcal{X}_t$ is defined as $\text{diam}(\mathcal{X}_t) = \max\{\|\mathbf{x} - \mathbf{x}'\| : \mathbf{x}, \mathbf{x}' \in \mathcal{X}_t\}$.
- For a given random variable $\xi \in \mathbb{R}$, $\mathbb{E}[\xi]$ denotes the expected value of ξ.
- Given a matrix $\mathbf{X} \in \mathbb{R}^{n \times l}$, $\text{vec}(\mathbf{X}) \in \mathbb{R}^p$ denotes the column vectorized $\mathbf{X}$ with its columns stacked in order on top of one another and $p := nl$.
- For a matrix $\mathbf{X} \in \mathbb{R}^{n \times l}$, $\|\mathbf{X}\|_* := \sum_{i=1}^r \sigma_i(\mathbf{X})$ where r is the rank of $\mathbf{X}$, and σ_i represented the singular values of $\mathbf{X}$.

Chapter 2

Online Data-Driven Optimization

This chapter is divided in two main sections. In Section 2.1, we present an optimization problem featuring a known time-varying engineering cost and an unknown (dis)comfort function. Based on this model, we develop a feedback-based projected gradient method to solve the problem in an online fashion, where: i) feedback from the user is leveraged to learn the (dis)comfort function concurrently with the execution of the algorithm; and, ii) measurements are used to estimate the gradient of the known engineering cost. To learn the unknown function, a shape-constrained Gaussian Process (GP) is leveraged; this approach allows one to obtain an estimated function that is strongly convex and smooth. The performance of the online algorithm is analyzed by using metrics such as the tracking error and the dynamic regret.

In Section 2.2, we also consider a feedback-based projected gradient method for optimizing systems modeled as algebraic maps. The focus is on a setup where the gradient is corrupted by random errors that follow a sub-Weibull distribution, and where the measurements of the output – which replace the input-output map of the system in the algorithmic updates – may not be available at each iteration. The sub-Weibull error model is particularly well-suited in frameworks where the cost of the problem is learned via GP regression (from functional evaluations) concurrently with the execution of the algorithm; however, it also naturally models setups where nonparametric methods and neural networks are utilized to estimate the cost. Using the sub-Weibull model, and with Bernoulli random variables modeling missing measurements of the system output, we show that the online algorithm generates points that are within a bounded error from the optimal solutions.

2.1 Online Data-Driven Gradient Algorithm with Learning

In this section, we introduce an optimization problem featuring a *known* time-varying engineering cost and an *unknown* (dis)comfort function. The engineering cost can be related to operational efficiency and may capture objectives such as aggregate set-point tracking when devices aggregate in a virtual power-plant fashion; it is time-varying [52] in a sense that it captures time-varying objectives (e.g., tracking of a power set-point that evolves over time), dynamic pricing, or real-time measurements. In lieu of synthetic mathematical models for the user's functions (based on e.g., statistics or averaged models), this section leverages Gaussian Processes (GPs) [133, 147] to learn the function from data (e.g., users' feedback). Approximating a function with a GP often leads to a nonconvex smooth cost; to favor computational tractability, and since user's preferences are often well approximated by convex functions (see, e.g.,[85] and references therein), we leverage a shape-constrained GPs approach where the discomfort function is approximated with a function that is strongly convex, differentiable, and with a Lipschitz gradient [168]. We then develop a feedback-based projected gradient method to solve the problem in an online fashion. The proposed strategy allows to overcome the following challenges:

• **Challenge 1:** (*Discomfort function uncertainty*) The functions that model the users' discomfort may not be known and models may be inaccurate. Feedback from the user is leveraged to learn the (dis)comfort function concurrently with the execution of the algorithm using a shape-constrained GPs.

• **Challenge 2:** (*Pervasive metering*) To solve the optimization problem, one may require the measurements of the powers of non-controllable loads at all locations in real time. In the proposed strategy, measurements of electrical quantities are used to estimate the gradient of the known engineering cost, and information about the non-controllable loads is not necessary.

This section is organized as follows. Section 2.1.1 presents the problem statement, Section 2.1.2 and 2.1.3 shows a brief introduction to GPs regression and shaped-constrained GPs, and finally Section 2.1.4 presents the proposed online algorithm.

2.1.1 Problem Statement

We consider a network with M controllable devices. Time is discretized as $t \in \mathcal{T} := \{k\Delta, k \in \mathbb{N}\}$, where Δ is a given time interval (e.g., one second or a few seconds [53, 95]). Commands are dispatched to the devices at each time t, and the commanded set-point for the mth device is denoted as $x_{m,t} \in \mathcal{X}_{m,t}$, where $\mathcal{X}_{m,t} \subseteq \mathbb{R}$ is a convex and compact set modeling hardware or operational constraints (e.g., real power commands or temperature set-points). If a device (e.g., a load) can be controlled at the slower rate (e.g., at the minute-level), the respective set-point is kept constant over a number of time steps (i.e., $\mathcal{X}_{m,t}$ is a singleton set). To simplify the notation, the set-points at time t are aggregated in the column vector $\mathbf{x}_t = [x_{1,t}, x_{2,t}, \dots, x_{M,t}]^\top \in \mathcal{X}_t \subseteq \mathbb{R}^M$, where $\mathcal{X}_t$ is a convex and compact set and is defined as $\mathcal{X}_t := \mathcal{X}_{1,t} \times \mathcal{X}_{2,t} \times \cdots \times \mathcal{X}_{M,t}$.

The set-points $\mathbf{x}$ are mapped to pertinent electrical states $\mathbf{y} \in \mathbb{R}^S$ through a mapping $\mathbf{y} = \mathcal{M}(\mathbf{x}, \mathbf{w}_t)$, where $\mathcal{M} : \mathbb{R}^M \times \mathbb{R}^W \mapsto \mathbb{R}^S$ models the power network effects and $\mathbf{w}_t \in \mathbb{R}^W$ is a (possibly high-dimensional) vector of powers consumed by W non-controllable devices. In particular, in this section we focus on a model of the form $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{w}_t$, where $\mathbf{A} \in \mathbb{R}^{S \times N}$ and $\mathbf{B} \in \mathbb{R}^{S \times W}$ are known (and possibly time-varying) network matrices.

The objective is to formulate a problem [67, 97] that allows real-time control of end-user devices by minimizing a cost that accounts for both network performance metrics and user satisfaction. Accordingly, let $u_m : \mathcal{X}_m \mapsto \mathbb{R}$ be a "discomfort function" for the the mth user or device. The function u_m is assumed to be time-invariant for simplicity; however, the proposed approach can be naturally extended to cases where some of the functions $u_{m,t}$ are time-varying functions to model a dynamic user behavior. Many exiting works presume that the function u_m is *known* and it is convex; however, here u_m will be *learned* from data.

Consider the following *time-varying* problem [52]:

$$\min_{\{\mathbf{y}_t \in \mathbb{R}^S,\, \mathbf{x} \in \mathcal{X}_t\}_{i=1}^T} \quad \sum_{m=1}^{M} u_m(x_m) + C_t(\mathbf{y}_t) \tag{2.1a}$$

$$\text{subject to:} \quad \mathbf{y}_t = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{w}_t \tag{2.1b}$$

for $t \in \mathcal{T}$, where $C_t : \mathbb{R}^S \to \mathbb{R}$ is a time-varying smooth and convex function associated with the vector of states $\mathbf{y}_t$. Let $\mathbf{x}_{t,*}$ be an optimal of (2.1); the objective then is to identify an optimal *trajectory* $\{\mathbf{x}_{t,*}, t \in \mathcal{T}\}$. Before proceeding, a couple of examples of applications are provided.

Example 1: Feeder-level problem. For a feeder, $\mathbf{y}_t$ can collect voltages at some selected nodes [53] and the net powers measured at the point of connection of the feeder with the rest of the grid. One may want to drive the state $\mathbf{y}_t$ towards a time-varying reference point $\mathbf{y}_{\text{ref},t}$ using the function $C_t(\mathbf{x}_t) = \frac{\beta}{2}\|\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{w}_t - \mathbf{y}_{\text{ref},t}\|^2$, with $\beta > 0$. In this case, $\mathbf{A}$ can be constructed based on the Jacobian of the power flow equations, linear approximations of the power flow equations, or by estimating the sensitivities of the network. As shown shortly, the proposed algorithmic framework does not need knowledge of the matrix $\mathbf{B}$.

Example 2: Neighborhood-level problem. For an aggregation of devices in a neighborhood or community, y_t represents the total active power at the point of interconnection of the rest of the grid. In this case, $\mathbf{A}$ boils down to a row-vector with all ones and $x_{m,t}$ represents the active power set-points of the devices. In a "virtual power plant" setting, $y_{\text{ref},t}$ can be a time-varying reference signal for the active power at the point of interconnection to provide, for example, primal or secondary grid services.

However, solving problem (2.1) at each time step t might be not viable because of the main challenges 1 and 2; more specifically, one may not be able to collect measurements of the non-controllable powers $\mathbf{w}_t$ because of sensing limitations, and because the function u_m may be unknown or largely different from synthetic models. Here, we propose a feedback-based online algorithm where: *i)* measurements of $\mathbf{y}_t$ are utilized to estimate the gradient of the function $C_t\left(\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{w}_t\right)$; and, *ii)* feedback from the users are utilized to estimate the functions $\{u_m\}_{m=1}^M$ concurrently with the execution of the online algorithm. In this chapter, the function u_m is estimated using feedback information from the user via GPs. Specifically, a shape-constrained GPs approach [168] is pursued to approximate the discomfort function with a strongly convex and smooth function. Accordingly, let $\hat{u}_m(x_{m,t})$ be the estimate of $u_m(x_{m,t})$ available at time t. In lieu of (2.1), the goal then is to

identify solutions of the following optimization problem in an online fashion:

$$\mathbf{x}_{t,*} = \underset{\{x_{m,t}\in\mathcal{X}_{m,t}\}_{i=1}^{M}}{\operatorname{argmin}} \sum_{m=1}^{M} \hat{u}_m(x_{m,t}) + C_t\left(\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{w}_t\right). \tag{2.2}$$

How to construct $\hat{u}_m(x_{m,t})$ is explained next.

2.1.2 Gaussian Processes Regression

In this section, we introduce the main concepts underpinning GPs [133]. This is a supervised learning technique that offer a non-parametric model that is convenient for the learning setting of this work because of the simplicity of the online updates and the ability to handle asynchronous and noisy data. In this section, the subscripts m and t are removed under the understanding that the technical arguments apply to each of the discomfort functions for all times. Then, x and x' are the inputs of the $\mathcal{GP}$, and are related to the control inputs $x_{m,t}$ and $x'_{m,t}$. Also, the set $\mathcal{X}$ makes reference to $\mathcal{X}_{m,t}$ and u represented u_m.

A GP is a stochastic process and it is specified by its mean function $\mu(x)$ and its covariance function $k(x,x')$; i.e., for any $x, x' \in \mathcal{X} \subseteq \mathbb{R}$, $\mu(x) = \mathbb{E}[u(x)]$ and $k(x,x') = \mathbb{E}[(u(x)-\mu(x))(u(x')-\mu(x'))]$ [133]. Let $\mathbf{x}_p = [x_1 \in \mathcal{X}, \ldots, x_p \in \mathcal{X}]^\top$ be the set of p sample points; let $z_i = u(x_i) + \epsilon_i$, with $\epsilon_i \overset{iid}{\sim} \mathcal{N}(0,\sigma^2)$ Gaussian noise, be the noisy measurements at the sample points $x_i \,\forall\, i = 1,\ldots,p$; and, define $\mathbf{z}_p = [z_1,\ldots,z_p]^\top$. Then, the posterior distribution of $(u(x)|\mathbf{x}_p,\mathbf{z}_p)$ is a GP with mean $\mu_p(x)$, covariance $k_p(x,x')$, and variance $\varsigma_p^2(x)$ given by:

$$\mu_p(x) = \mathbf{k}_p(x)^\top(\mathbf{K}_p + \sigma^2\mathbf{I}_p)^{-1}\mathbf{z}_p, \tag{2.3a}$$

$$k_p(x,x') = k(x,x') - \mathbf{k}_p(x)^\top(\mathbf{K}_p + \sigma^2\mathbf{I}_p)^{-1}\mathbf{k}_p(x'), \tag{2.3b}$$

$$\varsigma_p^2(x) = k_p(x,x), \tag{2.3c}$$

where $\mathbf{k}_p(x) = [k(x_1,x),\ldots,k(x_p,x)]^\top$, $\mathbf{K}_p$ is the positive definite kernel matrix $[k(x,x')]$, and the subscript p indicates the number of data points in $\mathbf{x}_p$. Thus, an estimate of the (unknown) function $u(x)$ can be written as $u(x) \sim \mathcal{GP}(\mu_p(x), k_p(x,x'))$. The covariance function specifies the covariance $\operatorname{cov}(u(x),u(x'))$ between pairs of random variables; using squared exponential (SE) kernel as an

example, it is defined as

$$k(x, x') = \sigma_f^2 \, e^{-\frac{1}{2l^2}(x-x')^2}$$

for the univariante input case, where the hyperparameters are the variance σ_f^2 and the characteristic length-scale l. For the multivariante case, the characteristic length-scale l_i for each dimension, is $\{l_i\}_{i=1}^H$ where $H \in \mathbb{R}$ is the dimension. In that case, the squared length-scales are collected into a $H \times H$, diagonal matrix P.

The Derivative Processes of GP: It is convenient to consider the SE covariance function because the resulting process has derivatives of all orders (see, e.g., [2, Theorem 2.2.2]). Since differentiation is a linear operator, derivatives of the GP remains a GP [133]. To obtain a strongly convex function, we will use the second derivative process of the GP. In particular, the corresponding mean and covariance function (jointly with the original process and the second-order derivative process) are [168]:

$$\mathbb{E}\left[\frac{\partial^2 u(x)}{\partial x^2}\right] = \frac{\partial^2 \mu(x)}{\partial x^2} = 0, \tag{2.4a}$$

$$\begin{aligned} k^{22}(x, x') &:= \operatorname{cov}\left[\frac{\partial^2 u(x)}{\partial x^2}, \frac{\partial^2 u(x')}{\partial x'^2}\right] \\ &= \sigma_f^2 e^{-\frac{1}{2l^2}(x-x')^2} \times \frac{1}{l^4}\left(\frac{1}{l^4}(x-x')^4 - \frac{1}{l^2}6(x-x')^2 + 3\right), \end{aligned} \tag{2.4b}$$

$$k^{02}(x, x') := \operatorname{cov}\left[\frac{\partial^2 u(x)}{\partial x^2}, u(x')\right] = \sigma_f^2 e^{-\frac{1}{2l^2}(x-x')^2}\left(\frac{1}{l^4}(x-x')^2 - \frac{1}{l^2}\right). \tag{2.4c}$$

2.1.3 Shape-constrained Gaussian Processes Regression

In this section, we introduce the main concepts related to shape-constrained GPs [168].

Suppose that one acquires noisy observations $\mathbf{z}_p$ of the GP at p points $\mathbf{x}_p$ (based on, e.g., the user's feedback), but no observations over the derivative process are available. However, we will impose derivative constraints at q points $\mathbf{d} := [d_1, \ldots, d_q]^\top$ [168]; that is, constraints on the shape of the function are imposed even at points where there is not observation of the actual process.

Let $\mathbf{u}(\mathbf{x}) = [u(x_1), \ldots, u(x_p)]^\top$ and $\mathbf{u}''(\mathbf{d}) = [u''(d_1), \ldots, u''(d_q)]^\top$; then, the joint distribution of the GP and its second-order derivative is:

$$\begin{bmatrix} \mathbf{u}(\mathbf{x}) \\ \mathbf{u}''(\mathbf{d}) \end{bmatrix} \sim \mathcal{N}\Bigg(\begin{bmatrix} \mu \mathbf{1}_p \\ \mathbf{0}_q \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{x},\mathbf{x}) & \mathbf{K^{02}}(\mathbf{x},\mathbf{d}) \\ \mathbf{K^{20}}(\mathbf{d},\mathbf{x}) & \mathbf{K^{22}}(\mathbf{d},\mathbf{d}) \end{bmatrix} \Bigg),$$

where $\mathbf{K}(\mathbf{x},\mathbf{x}) = \mathbf{K}_p$, $\mathbf{K^{02}}(\mathbf{x},\mathbf{d}) = [k^{02}(x,d)]$, $\mathbf{K^{20}}(\mathbf{d},\mathbf{x}) = \mathbf{K^{02}}(\mathbf{x},\mathbf{d})^\top$ and $\mathbf{K^{22}}(\mathbf{d},\mathbf{d}) = [k^{22}(d,d')]$.

In the following, we will impose constraints via indicator functions. Assign to $u(\cdot)$ a GP prior, and consider obtaining an estimated function that is L_u-smooth and γ_u-strongly convex, for a given $L_u > 0$ and $\gamma_u > 0$. We adapt the results presented in [168] for the marginal constrained prior distribution.

Following [168, Lemma 3.1], the joint conditional posterior distribution of $(\mathbf{u}(x^\circ)|\mathbf{u}''(\mathbf{d}), \mathbf{x}_p, \mathbf{z}_p)$, for a point x° of a new set of p° points, given the current observations $\mathbf{z}_p$, is a GP with mean, covariance, and standard deviation given by:

$$\begin{aligned} \bar{\mu}_{p^\circ}(x^\circ) &= \mu \mathbf{1}_{p^\circ} + B_3(\mathbf{x}, x^\circ, \mathbf{d}) B_1(\mathbf{x},\mathbf{d})^{-1}(\mathbf{z}_p - \mu \mathbf{1}_p) \\ &\quad + (A_2(x^\circ,\mathbf{d}) - B_3(\mathbf{x}, x^\circ, \mathbf{d}) B_1(\mathbf{x},\mathbf{d})^{-1} A_1(\mathbf{x},\mathbf{d}))\mathbf{u}''(\mathbf{d}), \qquad (2.5a) \\ \bar{k}_{p^\circ}(x^\circ, x^{\circ\prime}) &= A(\mathbf{x}, x^\circ, \mathbf{d}), \qquad (2.5b) \\ \bar{\varsigma}_{p^\circ}(x^\circ) &= \sqrt{A(\mathbf{x}, x^\circ, \mathbf{d})}, \qquad (2.5c) \end{aligned}$$

and the posterior distribution of $(\mathbf{u}''(\mathbf{d})|\mathbf{x}_p, \mathbf{z}_p)$ is given by:

$$(\mathbf{u}''(\mathbf{d})|\mathbf{x}_p, \mathbf{z}_p) \propto \mathcal{N}(\boldsymbol{\mu}(\mathbf{d}), \mathbf{D}(\mathbf{d},\mathbf{d})) \mathbf{1}_{\{\gamma_u \leq u''(d_i) \leq L_u,\, i=1,\ldots,q\}}$$

where $(\mathbf{u}''(\mathbf{d})|\mathbf{x}_p, \mathbf{z}_p)$ is a truncated normal distribution and,

$$\begin{aligned} \boldsymbol{\mu}(\mathbf{d}) &= \mathbf{K^{20}}(\mathbf{d},\mathbf{x})(\sigma^2 \mathbf{I} + \mathbf{K}(\mathbf{x},\mathbf{x}))^{-1}(\mathbf{z}_p - \mu \mathbf{1}_p), \\ \mathbf{D}(\mathbf{d},\mathbf{d}) &= \mathbf{K^{22}}(\mathbf{d},\mathbf{d}) - \mathbf{K^{20}}(\mathbf{d},\mathbf{x})(\sigma^2 \mathbf{I} + \mathbf{K}(\mathbf{x},\mathbf{x}))^{-1}\mathbf{K^{02}}(\mathbf{x},\mathbf{d}), \\ A_1(\mathbf{x},\mathbf{d}) &= \mathbf{K^{02}}(\mathbf{x},\mathbf{d})\mathbf{K^{22}}(\mathbf{d},\mathbf{d})^{-1}, \\ A_2(x^\circ,\mathbf{d}) &= \mathbf{K^{02}}(x^\circ,\mathbf{d})\mathbf{K^{22}}(\mathbf{d},\mathbf{d})^{-1}, \\ B_1(\mathbf{x},\mathbf{d}) &= \sigma^2 \mathbf{I} + \mathbf{K}(\mathbf{x},\mathbf{x}) - \mathbf{K^{02}}(\mathbf{x},\mathbf{d})\mathbf{K^{22}}(\mathbf{d},\mathbf{d})^{-1}\mathbf{K^{20}}(\mathbf{d},\mathbf{x}), \end{aligned}$$

$$B_2(x^\circ, \mathbf{d}) = \mathbf{K}(x^\circ, x^\circ) - \mathbf{K^{02}}(x^\circ, \mathbf{d})\mathbf{K^{22}}(\mathbf{d}, \mathbf{d})^{-1}\mathbf{K^{20}}(\mathbf{d}, x^\circ),$$

$$B_3(\mathbf{x}, x^\circ, \mathbf{d}) = \mathbf{K}(x^\circ, \mathbf{x}) - \mathbf{K^{02}}(x^\circ, \mathbf{d})\mathbf{K^{22}}(\mathbf{d}, \mathbf{d})^{-1}\mathbf{K^{20}}(\mathbf{d}, \mathbf{x}),$$

$$A(\mathbf{x}, x^\circ, \mathbf{d})) = B_2(x^\circ, \mathbf{d}) - B_3(\mathbf{x}, x^\circ, \mathbf{d})B_1(\mathbf{x}, \mathbf{d})^{-1}B_3(\mathbf{x}, x^\circ, \mathbf{d})^\top,$$

with μ and σ^2 given parameters of the prior. The parameters l and σ_f^2 of the GP can be estimated, for example, by using the maximum likelihood estimator [133]. The locations of the virtual derivative points are defined beforehand. By imposing the smooth and strong convexity constraints on points $\mathbf{d}$ that are dense enough, shape-constrained GPs ensure that the posterior mean function $\bar{\mu}_{p^\circ}(x^\circ)$ is "practically" (i.e., indistinguishable for all practical purposes) smooth and strongly convex [168]. The choice of shape-constrained GPs versus exact methods, such as smooth strong convex regression [145] (which would ensure shape properties exactly and everywhere) is motivated by the fact that the latter is more computationally intensive and its learning rate can be significantly slower.

2.1.4 Online Gaussian Processes-based Gradient Descend Algorithm

When the functions $\{u_m\}_{m=1}^M$ are known and the non-controllable powers can be measured at each time instant t, then the time-varying problem (2.2) can be solved in an online fashion using the following online projected gradient algorithm:

$$\mathbf{x}_t = \text{proj}_{\mathcal{X}_t}\{\mathbf{x}_{t-1} - \alpha\left(\nabla_{\mathbf{x}} U(\mathbf{x}_{t-1}) + \nabla_{\mathbf{x}} C_t(\mathbf{x}_{t-1})\right)\} \tag{2.6}$$

where $U(\mathbf{x}) := \sum_{m=1}^M u_m(x_m)$ for brevity, $\text{proj}_{\mathcal{X}}\{\mathbf{y}\} := \arg\min_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x} - \mathbf{y}\|^2$ is the projection operator, and $\alpha > 0$ is the step size. To address the challenges 1 and 2, the online algorithm (2.6) is modified as explained next.

2.1.4.1 Online Algorithm

Recall that $t \in \mathcal{T}$ is the time index. We now introduce an additional index $p_m(t)$ (one per device or user), used as a counter for the number of data points $\mathbf{z}_{m,t} := [z_{m,1}, \ldots, z_{m,p_m(t)}]^\top$ received from the mth user up to time t; we recall that $z_{m,p_m(t)} = u_m(x_{m,t}) + \epsilon_{m,t}$ (t being the time when

the p_mth user feedback is received). The counter $p_m(t)$ does not generally coincide with t, since a user may provide feedback sporadically or at a slower time scale (whereas the algorithm is run on a fast time scale). Hereafter, we omit the dependence of p_m on t for notation simplicity.

With p_m data points available (i.e., received from the mth user), we define the estimate $\hat{u}_{m,p_m}(x_{m,t})$ of $u_m(x_{m,t})$ as:

$$\hat{u}_{m,p_m}(x_{m,t}) := \bar{\mu}_{m,p_m}(x_{m,t}) \tag{2.7}$$

where $\bar{\mu}_{m,p_m}(x_{m,t})$ is given by (2.5a) based on p_m data points (where we remind that is "practically" smooth and strongly convex). In other words, $\hat{u}_{m,p_m}(x_{m,t})$ is obtained via the mean of the shape-constrained GPs when feedback from the user is received p_m times. Further, at a given point $x_{m,t}$, the derivative of $\hat{u}_{m,p_m}(x_{m,t})$ is estimated via finite-difference as [63]:

$$v_{m,p_m}(x_{m,t}) := \frac{\hat{U}_{m,p_m}(x_{m,t}+\delta) - \hat{U}_{m,p_m}(x_{m,t})}{\delta} \tag{2.8}$$

with δ a pre-selected parameter. For future developments, let $\mathbf{v}(\mathbf{x}_t) := [v_{1,p_1}(x_{1,t}), ..., v_{M,p_M}(x_{M,t})]^\top$.

The evaluation of the gradient of $C(\mathbf{y}_t)$ requires measurements of the non-controllable devices $\mathbf{w}_t$ at each time step t. Similar to, e.g., [27, 53], measurements $\hat{\mathbf{y}}_t$ can be utilized in the computation of the gradient of $C(\mathbf{y}_t)$ instead of the map $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{w}_t$. For example, if the function $C_t(\mathbf{x}_t)$ is $C_t(\mathbf{x}_t) = \frac{\beta}{2}\|\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{w}_t - \mathbf{y}_{\text{ref},t}\|^2$, its gradient reads $\nabla C_t(\mathbf{x}_t) = \beta\mathbf{A}^\top(\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{w}_t - \mathbf{y}_{\text{ref},t})$; on the other hand, an estimate of the gradient using the measurement $\hat{\mathbf{y}}_t$ amounts to $\mathbf{s}_t := \beta\mathbf{A}^\top(\hat{\mathbf{y}}_t - \mathbf{y}_{\text{ref},t})$. Indeed, $\mathbf{s}_t$ can be interpreted as a noisy version of $\nabla_{\mathbf{x}} C(\mathbf{y}_t)$ [52].

Overall, the proposed shape-constrained GP-based online projected gradient descent (SGP-OPGD) algorithm involves the sequential execution of the following step:

$$\mathbf{x}_t = \text{proj}_{\mathcal{X}_t}\{\mathbf{x}_{t-1} - \alpha\,(\mathbf{v}(\mathbf{x}_{t-1}) + \mathbf{s}_t)\} \tag{2.9}$$

where we recall that $\mathbf{v}(\mathbf{x}_t)$ is an estimate of the gradient of $\hat{U}(\mathbf{x}_t)$, where $\hat{U}(\mathbf{x}_t) := \sum_{m=1}^{M} \hat{u}_{m,p_m}(x_{m,t})$, $\mathbf{s}_t$ is a noisy version of $\nabla_{\mathbf{x}} C(\mathbf{y}_t)$, t represents the time index, and p_m is the data counter for the user's feedback per device.

The steps of the SGP-OPGD are detailed in Algorithm 1. Notice that the update of $\mathbf{x}_t$ decouples into M parallel steps (one per device); this enables a distributed setting with a so-called

"gather-and-broadcast" architecture where measurements of $\hat{\mathbf{y}}_t$ are collected at a central location, $\mathbf{s}_t$ is broadcasted to the devices, and $x_{m,t}$ is computed locally at each device. Further, the function $\hat{u}_{m,p_m}(x_{m,t})$ is computed locally.

Algorithm 1: SGP-OPGD method

Initialize: $\mathbf{x}_0$, $\alpha = \frac{2}{\gamma+L}$; prior on $\{\hat{u}_m\}_{m=1}^M$ if available.
for $t = 1, 2, \ldots, T$ **do**
  Collect measurement $\hat{\mathbf{y}}_t$
  Compute the estimate gradient $\mathbf{s}_t$
  for $m = 1, 2, \ldots M$ **do**
    if Feedback is given:
      $p_m \to p_m + 1$
      Collect z_{m,p_m} and add it to $\mathbf{z}_{m,t}$
      Update $\hat{U}_{m,p_m}(x_{m,t})$ and compute $v_{m,p_m}(x_{m,t})$
    else Keep $\hat{U}_{m,p_m}(x_{m,t-1})$ and $v_{m,p_m}(x_{m,t-1})$
    Update setpoint as

$$x_{m,t} = \text{proj}_{\mathcal{X}_m}\{x_{m,t-1} - \alpha(v_{m,p_m}(x_{m,t-1}) + s_{m,t})\}$$

  end for
end for

2.1.4.2 Regret Analysis

The convergence of the online algorithm is compared against the optimal trajectory $\{\mathbf{x}_{t,*}\}_{t\in\mathcal{T}}$ and the optimal value function of (2.2). Hereafter, we define $f_t(\mathbf{x}) := \hat{U}(\mathbf{x}) + C_t(\mathbf{x})$ for brevity. We begin with the following standard assumptions.

AS1: The function f_t is L-smooth on $\mathcal{X}$; i.e., $\|\nabla f_t(\mathbf{x}) - \nabla f_t(\mathbf{x}')\| \leq L\|\mathbf{x} - \mathbf{x}'\|$ for all $t \in \mathcal{T}$ and $\mathbf{x}$, $\mathbf{x}' \in \mathcal{X}$.

AS2: The function f_t is γ-strongly convex.

AS3: The inexact gradient $\tilde{\nabla} f_t(\mathbf{x}) = \mathbf{v}_p(\mathbf{x}) + \mathbf{s}_t$ is defined as $\tilde{\nabla} f_t(\mathbf{x}_t) := \nabla f_t(\mathbf{x}_t) + \mathbf{e}_{1,t} + \mathbf{e}_{2,t}$, where $\mathbf{e}_{1,t}$ is the error in the gradient of $\{\hat{u}_{m,p_m}\}_{m=1}^M$ and $\mathbf{e}_{2,t}$ is the error in the estimated gradient $\mathbf{s}_t$. The sequence $\{\mathbf{e}_t := \mathbf{e}_{1,t} + \mathbf{e}_{2,t} \in \mathbb{R}^M, t \in \mathcal{T}\}$ is bounded; i.e., $\|\mathbf{e}_t\| < \infty$.

Regarding *AS1*, L is given by $L = L_U + L_C$, with L_U and L_C the Lipschitz constants of the gradients of $\hat{U}$ and C_t, respectively; notice that the Lipschitz constant of the gradient of each

individual function $\hat{u}_{m,p_m}$ is set a priori as in (2.5). If C_t is convex but not strongly convex, only the strong convexity coefficient of $\hat{U}$ plays a role in *AS2* [cf. (2.5)].

The variation between any two consecutive optimal points is defined as $\phi_t := \|\mathbf{x}_{t-1,*} - \mathbf{x}_{t,*}\|$. Now, define the path length and the cumulative gradient error as [52, 80]

$$\omega_T := \sum_{t=1}^{T} \phi_t, \qquad E_T = \sum_{t=1}^{T} \|\mathbf{e}_t\|. \tag{2.10}$$

These metrics will be utilized in the following results.

Proposition 1 Assume that $\alpha \in (0, 2/L)$. Under Assumptions *AS1-AS3*, the SGP-OPGD algorithm constructs a sequence $\{\mathbf{x}_t\}_{t\in\mathcal{T}}$ such that

$$\|\mathbf{x}_t - \mathbf{x}_{t,*}\| \leq \rho \|\mathbf{x}_{t-1} - \mathbf{x}_{t-1,*}\| + \rho\phi_t + \alpha\|\mathbf{e}_t\|, \tag{2.11}$$

where $\rho := \max\{|1 - \alpha\gamma|, |1 - \alpha L|\} < 1$.

Corollary 1 Under assumption *AS1-AS3*, with $\alpha \in (0, 2/L)$, the cumulative tracking error of the SGP-OPGD algorithm can be bounded as:

$$\sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}_{t,*}\| \leq \frac{1}{1-\rho} \left[\rho\|\mathbf{x}_0 - \mathbf{x}_{0,*}\| + \rho\omega_T + \alpha E_T\right]. \tag{2.12}$$

Proposition 1 establishes Q-linear convergence to a bounded error of the SGP-OPDG algorithm [52]; that is, each step of the algorithm is contractive up an error $\alpha\|\mathbf{e}_t\| + \phi_t$ given by the temporal variability of the problem and the errors in the gradient computation. On the other hand, Corollary 1 asserts that the tracking error of the algorithm is bounded if E_T and ω_T grow as $\mathcal{O}(T)$, and it goes to zero asymptotically if E_T and ω_T grow sublinearly in T; that is, if they grow as $o(T)$.

Finally, we provide a bound on the dynamic regret next.

Proposition 2 Suppose that Assumptions *AS1-AS3* hold, and let $\alpha \in (0, 2/L)$. Then, the dynamic regret of the SGP-OPGD algorithm can be bounded as:

$$\frac{1}{T}\sum_{t=1}^{T} [f_t(\mathbf{x}_t) - f_t(\mathbf{x}_{t,*})] = \mathcal{O}\left(T^{-1}\omega_T + T^{-1}E_T\right). \tag{2.13}$$

In par with Corollary 1, the dynamic regret is sublinear if ω_T and E_T are both sublinear. If ω_T and E_T grow linearly, then the dynamic regret behaves as $\mathcal{O}(1)$. Moreover, if the gradient is obtained without error (i.e., $\mathbf{e}_t = \mathbf{0}$), the dynamic regret exhibits a behavior similar to [55].

The proofs follow steps similar to [4, 55]; Proposition 2 uses the fact that, from the continuity of the gradient and the compactness of $\mathcal{X}_t$, the norm of the gradient is bounded. For completeness, the proofs are presented as follows.

Proof Proposition 1: Proposition 1 follows the results presented in [55]. First, by using the definition of the projected gradient descent iteration of $\mathbf{x}_t$, we can write the difference between $\mathbf{x}_t$ and the optimal point $\mathbf{x}_{t,*}$ as,

$$\|\mathbf{x}_t - \mathbf{x}_{t,*}\| = \|\text{proj}_{\mathcal{X}}[\mathbf{x}_{t-1} - \alpha\tilde{\nabla} f_t(\mathbf{x}_{t-1})] - \text{proj}_{\mathcal{X}}[\mathbf{x}_{t,*} - \alpha\nabla f_t(\mathbf{x}_{t,*})]\| \tag{2.14}$$

using the nonexpansiveness property of the projection on (2.14),

$$\begin{aligned}\|\mathbf{x}_t - \mathbf{x}_{t,*}\| &\leq \|\mathbf{x}_{t-1} - \alpha\tilde{\nabla} f_t(\mathbf{x}_{t-1}) - \mathbf{x}_{t,*} + \alpha\nabla f_t(\mathbf{x}_{t,*})\| \\ &= \|[\mathbf{x}_{t-1} - \alpha\nabla f_t(\mathbf{x}_{t-1})] - [\mathbf{x}_{t,*} - \alpha\nabla f_t(\mathbf{x}_{t,*})] - \alpha\mathbf{e}_t\| \\ &\leq \|[\mathbf{x}_{t-1} - \alpha\nabla f_t(\mathbf{x}_{t-1})] - [\mathbf{x}_{t,*} - \alpha\nabla f_t(\mathbf{x}_{t,*})]\| + \alpha\|\mathbf{e}_t\|, \end{aligned} \tag{2.15}$$

after applying triangle inequality. By using the fact that the map $\mathcal{I} - \alpha\nabla f_t$ is Lipschitz continuous with parameter $\rho = \max\{|1 - \alpha\gamma|, |1 - \alpha L|\}$, then (2.15) can be bounded as

$$\|\mathbf{x}_t - \mathbf{x}_{t,*}\| \leq \rho\|\mathbf{x}_{t-1} - \mathbf{x}_{t,*}\| + \alpha\|\mathbf{e}_t\|. \tag{2.16}$$

Now, by adding and subtracting $\mathbf{x}_{t-1,*}$ in the first term of the right hand side of (2.16) we have,

$$\|\mathbf{x}_t - \mathbf{x}_{t,*}\| \leq \rho\|\mathbf{x}_{t-1} - \mathbf{x}_{t-1,*} + \mathbf{x}_{t-1,*} - \mathbf{x}_{t,*}\| + \alpha\|\mathbf{e}_t\| \tag{2.17}$$

$$\leq \rho\|\mathbf{x}_{t-1} - \mathbf{x}_{t-1,*}\| + \rho\|\mathbf{x}_{t-1,*} - \mathbf{x}_{t,*}\| + \alpha\|\mathbf{e}_t\|. \tag{2.18}$$

where (2.18) follows after applying triangular inequality. Now, we can use (2.10) as

$$\|\mathbf{x}_t - \mathbf{x}_{t,*}\| \leq \rho\|\mathbf{x}_{t-1} - \mathbf{x}_{t-1,*}\| + \rho\phi_t + \alpha\|\mathbf{e}_t\|. \qquad \square$$

Proof Corollary 1: Corollary 1 follows the results presented in [55, 4]. First, we write the cumulative sum for $\mathbf{x}_t$ and its optimal value $\mathbf{x}_{t,*}$, and using the results from Proposition 1, we have,

$$\sum_{t=1}^{T}\|\mathbf{x}_t - \mathbf{x}_{t,*}\| \leq \sum_{t=1}^{T}[\rho\|\mathbf{x}_{t-1} - \mathbf{x}_{t-1,*}\| + \rho\phi_t + \alpha\|\mathbf{e}_t\|] \tag{2.19}$$

$$\leq \rho\sum_{t=1}^{T}\|\mathbf{x}_{t-1} - \mathbf{x}_{t-1,*}\| + \rho\sum_{t=1}^{T}\phi_t + \alpha\sum_{t=1}^{T}\|\mathbf{e}_t\|], \tag{2.20}$$

$$\leq \rho\sum_{t=1}^{T}\|\mathbf{x}_{t-1} - \mathbf{x}_{t-1,*}\| + \rho\omega_T + \alpha E_T, \tag{2.21}$$

where (2.21) holds after using the definition in (2.10).

The inequality in (2.21) still holds if we add an additional non-negative term $\|\mathbf{x}_0 - \mathbf{x}_{0,*}\|$. So, by reordering terms in (2.21), we have,

$$\sum_{t=1}^{T}\|\mathbf{x}_t - \mathbf{x}_{t,*}\| \leq \rho\|\mathbf{x}_0 - \mathbf{x}_{0,*}\| + \rho\sum_{t=1}^{T}\|\mathbf{x}_t - \mathbf{x}_{t,*}\| + \rho\omega_T + \alpha E_T$$

$$(1-\rho)\sum_{t=1}^{T}\|\mathbf{x}_t - \mathbf{x}_{t,*}\| \leq \rho\|\mathbf{x}_0 - \mathbf{x}_{0,*}\| + \rho\omega_T + \alpha E_T. \tag{2.22}$$

Then, (2.22) becomes,

$$\sum_{t=1}^{T}\|\mathbf{x}_t - \mathbf{x}_{t,*}\| \leq \frac{\rho}{1-\rho}\|\mathbf{x}_0 - \mathbf{x}_{0,*}\| + \frac{\rho}{1-\rho}\omega_T + \frac{\alpha}{1-\rho}E_T, \tag{2.23}$$

where we have a bound for the cumulative optimality gap. □

Proof Proposition 2: By the definition of convexity, we can write the cumulative sum of the difference of $f_t(\mathbf{x}_t)$ and the optimal value function $f_t(\mathbf{x}_{t,*})$ as,

$$\sum_{t=1}^{T}[f_t(\mathbf{x}_t) - f_t(\mathbf{x}_{t,*})] \leq \sum_{t=1}^{T}\langle\nabla f_t(\mathbf{x}_{t,*}), \mathbf{x}_t - \mathbf{x}_{t,*}\rangle \leq \sum_{t=1}^{T}\|\nabla f_t(\mathbf{x}_{t,*})\|\|\mathbf{x}_t - \mathbf{x}_{t,*}\|, \tag{2.24}$$

where (2.24) is obtained after applying the Cauchy-Schwartz inequality. Now, by assumption the continuity of the gradients and the compactness of $\mathcal{X}_t$ (i.e. $\|\nabla f_t(\mathbf{x})\| \leq G$ for any $\mathbf{x} \in \mathcal{X}$) and Corollary 1, (2.24) becomes,

$$\sum_{t=1}^{T}[f_t(\mathbf{x}_t) - f_t(\mathbf{x}_{t,*})] \leq \frac{\rho G}{1-\rho}\|\mathbf{x}_0 - \mathbf{x}_0^*\| + \frac{\rho G}{1-\rho}\omega_T + \frac{\alpha G}{1-\rho}E_T. \tag{2.25}$$

The inequality in (2.25) holds the result for Proposition 2. □

2.2 Online Data-Driven Gradient Algorithm with Infrequent Feedback

Similar to Section 2.1, in this section we consider optimization problems associated with systems that feature M controllable inputs $\mathbf{x} \in \mathbb{R}^M$ and unknown exogenous inputs $\mathbf{w} \in \mathbb{R}^w$. Modeling the system as an algebraic map $\mathbf{y} = \mathcal{M}(\mathbf{x}, \mathbf{w})$, where $\mathcal{M} : \mathbb{R}^M \times \mathbb{R}^w \to \mathbb{R}^y$ is well-defined [74, 27, 19], the objective is to steer the system to optimal solutions of the following time-varying problem

$$\mathbf{x}_{t,*} \in \min_{\mathbf{x} \in \mathcal{X}_t} f_t(\mathbf{x}) := U_t(\mathbf{x}) + C_t(\mathcal{M}(\mathbf{x}, \mathbf{w}_t)), \tag{2.26}$$

where $t \in \mathbb{N}$ is the time index, $\mathcal{X}_t \subseteq \mathbb{R}^M$ is a time-varying constraint set for the inputs, $\mathbf{x} \mapsto U_t(\mathbf{x})$ is a cost associated with the inputs, and $\mathbf{y} \mapsto C_t(\mathbf{y})$ is a cost associated with the outputs. We consider the case where $\mathbf{w}_t$ is unknown or it cannot be directly measured; with $\mathbf{w}_t$ *not* known, (2.26) can be solved using online algorithms of the following form (see, e.g.,[27, 19]):

$$\mathbf{x}_t = \text{proj}_{\mathcal{X}_t} \left[\mathbf{x}_{t-1} - \alpha(\mathbf{J}_t^\top \nabla C_t(\mathbf{y}_{t-1}) + \nabla U_t(\mathbf{x}_{t-1})) \right], \tag{2.27}$$

where $\mathbf{J}_t := [\frac{\partial \mathcal{M}}{\partial x_m}(\mathbf{x}_{t-1})]$ is the Jacobian of $\mathcal{M}$ (x_m here denotes the mth entry of $\mathbf{x}$). The algorithm is "feedback-based" since the output measurement $\mathbf{y}_{t-1}$ replaces the system model $\mathcal{M}(\mathbf{x}_{t-1}, \mathbf{w}_{t-1})$ in the computation of the gradient.

We consider an online algorithm similar to (2.27), but in a setting where: *(i)* the gradient is corrupted by random errors that follow a sub-Weibull distribution [166]; and, *(ii)* measurements $\{\mathbf{y}_t\}_{t \in \mathbb{N} \cup \{0\}}$ are noisy and may not be available at each time t. The latter models processing and communication bottlenecks in the sensing layers of the system (for example, in power grid metering systems and transportation systems). On the other hand, the sub-Weibull model allows us to consider concurrent learning and optimization frameworks where the costs $\mathbf{x} \mapsto U_t(\mathbf{x})$ and $\mathbf{y} \mapsto C_t(\mathbf{y})$ are learned via Gaussian Process (GP) regression [133], parametric methods [117], non-parametric methods [73], and neural networks [109] from both a set of recorded data and (an infrequent set of) functional evaluations acquired during the execution of the algorithm. Learning the cost concurrently with the execution of the algorithm finds ample applications in cyber-physical

systems with human-in-the-loop [114], where $U_t(\mathbf{x})$ models users' preferences, as well as data-enabled and perception-based optimization [100] where $C_t(\mathbf{y})$ is learned from data. The ability of our framework to model various learning settings is grounded on the fact that the sub-Weibull distribution includes sub-Gaussian and sub-exponential errors as sub-cases, as well as random errors whose distribution has a finite support [166, 165].

Prior works in the context of feedback-based optimization considered time-invariant costs [27, 40, 76] and time-varying costs [19, 49, 155]; the convergence of algorithms were investigated when the cost is known, measurements are noiseless, and measurements are received at each iteration (see also the survey [74] for a comprehensive list of references).

Online optimization methods with concurrent learning of the cost with GPs were considered in [147] for functions satisfying the Polyak-Łojasiewicz inequality; the algorithm employed the upper confidence bound and the regret was investigated. Quadratic functions were considered in [117], and they were estimated via recursive least squares. However, in [147, 117], algorithms were not implemented with a system in the loop and no missing measurements were considered. Online algorithms with concurrent learning via shape-constrained GPs were considered in [122]; however, only bounds in expectation for the regret were provided. High probability convergence results were provided in [59] for non-monotone games, where the pseudo-gradient is learned from data. Although it is not the main focus of this section, we also acknowledge works on zeroth-order methods (see, e.g., [104, 157]) where gradient errors emerge from single- or multi-point gradient estimation; our framework based on a sub-Weibull model can be applied to derive convergence bounds when the gradient in (2.27) is estimated via single- or multi-point estimation. Finally, we mention that several convergence results have been derived for classical online algorithms [142, 113], including asynchronous implementations [16]; our results provide extensions to cases with sub-Weibull gradient errors.

The main contributions of this section are as follows. *i)* We consider an online feedback-based projected gradient descent method with intermittent updates and with noisy gradients. We provide new bounds for the error $\|\mathbf{x}_t - \mathbf{x}_{t,*}\|$ in *expectation* that hold iteration-wise, where $\{\mathbf{x}_t\}_{t\in\mathbb{N}}$

is the sequence generated by the algorithm; where, missing measurements are modeled as Bernoulli random variables. *ii)* We provide new bounds on $\|\mathbf{x}_t - \mathbf{x}_{t,*}\|$ in *high probability*; the bounds are derived by adopting a sub-Weibull distribution for the error affecting the gradient. *iii)* We show how our framework models concurrent learning approaches, where one leverages functional evaluations to learn the unknown cost via GPs during the execution of the algorithm. *iv)* We test the performance of our algorithm in cases where the cost is estimated via GPs and feedforward neural networks.

The remainder of this section is organized as follows. Section 2.2.1 introduces preliminary definitions. Section 2.2.2 presents the proposed algorithm and Section 2.2.3 provides the corresponding analysis. Finally, Section 2.2.4 presents the concurrent learning approach.

2.2.1 Preliminaries

In this section, we introduce the class of sub-Weibull random variables and provide relevant properties. For a random variable (rv) X, when the k-th moment of X exists for some $k \geq 1$, we define $\|X\|_k := (\mathbb{E}[|X|^k])^{1/k}$.

Definition 1 (Sub-Weibull rv [166]) A random variable $X \in \mathbb{R}$ is sub-Weibull if $\exists\, \theta > 0$ such that (s.t.) one of the following conditions is satisfied:

(i) $\exists\ \nu_1 > 0$ s.t. $\mathbb{P}[|X| \geq \epsilon] \leq 2e^{-(\epsilon/\nu_1)^{1/\theta}}$, $\forall\, \epsilon\ >\ 0$.

(ii) $\exists\ \nu_2 > 0$ s.t. $\|X\|_k \leq \nu_2 k^\theta$, $\forall\, k \geq 1$. □

The parameters ν_1, ν_2 differ by a constant that depends on θ; in particular, if property (ii) holds with parameter ν_2, then property (i) holds with $\nu_1 = (2e/\theta)^\theta\, \nu_2$. Hereafter, we use the shorthand notation $X \sim \mathrm{subW}(\theta, \nu)$ to indicate that X is a sub-Weibull rv according to Definition 1(ii) (i.e., $\|X\|_k \leq \nu k^\theta$, $\forall\, k \geq 1$). We note that the sub-Weibull class includes sub-Gaussian and sub-exponential rvs as sub-cases; in particular, if $\theta = 1/2$ and $\theta = 1$ we have sub-Gaussian and sub-exponential rvs, respectively. Furthermore, if a rv has a distribution with finite support, it

belongs to the sub-Gaussian class (by Hoeffding's inequality [165, Theorem 2.2.6]) and, thus, to the sub-Weibull class.

Proposition 3 (*Inclusion* [166]) Let $X \sim \text{subW}(\theta, \nu)$ and let θ', ν' s.t. $\theta' \geq \theta$, $\nu' \geq \nu$. Then, $X \sim \text{subW}(\theta', \nu')$. □

Proposition 4 (*Closure of sub-Weibull class [14]*) Let $X_i \sim \text{subW}(\theta_i, \nu_i)$, $i = 1, 2$, based on Definition 1(ii).

(a) *Product by scalar:* Let $a \in \mathbb{R}$, then $aX_i \sim \text{subW}(\theta_i, |a|\nu_i)$.

(b) *Sum by scalar:* Let $a \in \mathbb{R}$, then $a + X_i \sim \text{subW}(\theta_i, |a| + \nu_i)$.

(c) *Sum:* Let $\{X_i, i = 1, 2\}$ be possibly dependent; then, $X_1 + X_2 \sim \text{subW}(\max\{\theta_1, \theta_2\}, \nu_1 + \nu_2)$.

(d) *Product:* Let $\{X_i, i = 1, 2\}$ be independent; then, $X_1 X_2 \sim \text{subW}(\theta_1 + \theta_2, \nu_1 \nu_2)$. □

Proposition 5 (*High probability bound [166]*) Let $X \sim \text{subW}(\theta, \nu)$ according to Definition 1(ii), for some $\theta > 0$ and $\nu > 0$. Then, for any $\delta \in (0, 1)$, the bound:

$$|X| \leq \nu \log^{\theta}\left(\frac{2}{\delta}\right)\left(\frac{2e}{\theta}\right)^{\theta} \tag{2.28}$$

holds with probability $1 - \delta$. □

2.2.2 Online Data-Driven Gradient Descent Algorithm

In this section, we consider an online feedback-based algorithm of the form (2.27) to solve (2.26) where: *(i)* we utilize noisy measurements of the output $\hat{\mathbf{y}}_t = \mathcal{M}(\mathbf{x}_t, \mathbf{w}_t) + \mathbf{n}_t$, where $\mathbf{n}_t \in \mathbb{R}^y$ is a measurement noise, instead of requiring full knowledge of the map $\mathcal{M}$ and of the vector $\mathbf{w}_t$; *(ii)* we rely on *inexact* gradient information; and, *(iii)* the measurements of $\mathbf{y}_t$ may not be received at each iteration. Accordingly, the online algorithm is as follows (where we recall that $t \in \mathbb{N}$ is the time index):

$$\mathbf{x}_t = \begin{cases} \text{proj}_{\mathcal{X}_t}\left[\mathbf{x}_{t-1} - \alpha(p_t(\hat{\mathbf{y}}_{t-1}) + s_t(\mathbf{x}_{t-1}))\right] & \text{if } \hat{\mathbf{y}}_{t-1} \text{ is received} \\ \text{proj}_{\mathcal{X}_t}\left[\mathbf{x}_{t-1}\right] & \text{if } \hat{\mathbf{y}}_{t-1} \text{ is not received,} \end{cases} \tag{2.29}$$

where $s_t(\mathbf{x}) := \nabla U_t(\mathbf{x}) + \boldsymbol{\varepsilon}_t$ and $p_t(\hat{\mathbf{y}}) := \mathbf{J}_t^\top \nabla C_t(\mathbf{y}) + \boldsymbol{\xi}_t$ are approximations of the gradients $\nabla U_t(\mathbf{x})$ and $\mathbf{J}_t^\top \nabla C_t(\mathcal{M}(\mathbf{x}, \mathbf{w}_t))$, respectively, with $\boldsymbol{\varepsilon}_t \in \mathbb{R}^M$ and $\boldsymbol{\xi}_t \in \mathbb{R}^M$ random vectors that model the gradient errors. As discussed in Section 2.2.1, errors in the gradients emerge when a concurrent learning approach is utilized to estimate the functions $\mathbf{x} \mapsto U_t(\mathbf{x})$ and $\mathbf{y} \mapsto C_t(\mathbf{y})$ from samples using, e.g., GPs [147], parametric methods [117], non-parametric methods [73], or neural networks [109]. Additional sources of errors include the noise affecting the measurement of $\mathbf{y}_t$. Hereafter, we denote the overall gradient error as $\mathbf{e}_t := \boldsymbol{\varepsilon}_t + \boldsymbol{\xi}_t$.

We rewrite (2.29) in the following manner:

$$\mathbf{x}_t = \text{proj}_{\mathcal{X}_t} \left[\mathbf{x}_{t-1} - v_{t-1}\, \alpha \left(p_t(\hat{\mathbf{y}}_{t-1}) + s_t(\mathbf{x}_{t-1}) \right) \right], \tag{2.30}$$

where v_t is a rv taking values in the set $\{0, 1\}$, and is used to indicate whether the measurement of the system output is received or not. When a measurement is not received, we still utilize a projection onto the time-varying set $\mathcal{X}_t$. In the following, we introduce the main assumptions and we analyze the performance of the online algorithm (2.30).

2.2.3 Convergence Analysis

Here, we outline the main assumptions used to analyze the converges of the online algorithm (2.30)

AS1: The set $\mathcal{X}_t \subseteq \mathbb{R}^M$ is non-empty, convex and compact for all t.

AS2: For any $\mathbf{w} \in \mathbb{R}^w$, the function $\mathbf{x} \mapsto C_t(\mathcal{M}(\mathbf{x}, \mathbf{w}))$ is convex over $\mathbb{R}^M$, $\forall\, t$. Moreover, the composite function $\mathbf{x} \mapsto f_t(\mathbf{x})$ is μ_t-strongly convex and L_t-smooth over $\mathbb{R}^M$, for some $0 < \mu_t \leq L_t < \infty$ and $\forall\, t$.

We recall that $f_t(\mathbf{x})$ is L_t-smooth over $\mathbb{R}^M$, for some $L_t \geq 0$, if it is differentiable and $\|\nabla f_t(\mathbf{x}) - \nabla f_t(\mathbf{y})\| \leq L_t \|\mathbf{x} - \mathbf{y}\|, \forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^M$. The previous assumptions imply that the norm of the gradient of f_t is bounded over the compact set $\mathcal{X}_t$. Assumption *AS2* also implies that there is a unique optimizer $\mathbf{x}_{t,*}$ for each t. Furthermore, the map $\mathcal{I} - \alpha \nabla f_t : \mathbb{R}^M \to \mathbb{R}^M$ is ζ_t-Lipschitz

with $\zeta_t = \max\{|1-\alpha\mu_t|, |1-\alpha L_t|\}$ (see [138, Section 5.1]); if $\alpha \in \left(0, \frac{2}{L}\right)$, $L = \sup_{1\leq i\leq t}\{L_i\}$, then $\zeta_t < 1$; in other words, the map $\mathcal{I} - \alpha\nabla f_t$ is contractive.

AS3: The rv v_t is Bernoulli distributed with parameter $p := \mathbb{P}[v_t = 1] > 0$. The rvs $\{v_t\}_{t\in\mathbb{N}\cup\{0\}}$ are i.i.d..

AS4: For all $t \in \mathbb{N}\cup\{0\}$, $\exists\ \theta_\varepsilon > 0, \nu_{\varepsilon,t} > 0$ s.t. each entry of the vector $\boldsymbol{\varepsilon}_t$ is subW$(\theta_\varepsilon, \nu_{\varepsilon,t})$ according to Definition 1(ii). Moreover, $\boldsymbol{\varepsilon}_t$ is independent of $v_t\ \forall\ t$.

AS5: For all $t \in \mathbb{N}\cup\{0\}$, $\exists\ \theta_\xi > 0, \nu_{\xi,t} > 0$ s.t. each entry of the vector $\boldsymbol{\xi}_t$ is subW$(\theta_\xi, \nu_{\xi,t})$, according to Definition 1(ii). Moreover, $\boldsymbol{\xi}_t$ is independent of $v_t\ \forall\ t$.

The following lemma is then presented.

Lemma 1 Suppose that Assumptions *AS4-AS5* hold. Then, $\|\mathbf{e}_t\|$ is a sub-Weibull rv and, in particular, $\|\mathbf{e}_t\| \sim \text{subW}(\max\{\theta_\varepsilon, \theta_\xi\}, (2^{\theta_\varepsilon}\sqrt{M}\nu_{\varepsilon,t}) + (2^{\theta_\xi}\sqrt{M}\nu_{\xi,t}))$, $\forall\ t$. □

This lemma can be proved by using [14, Lemma 3.4] and part (c) of Proposition 4; the proof is omitted.

Our analysis seeks bounds on the error $\|\mathbf{x}_t - \mathbf{x}_{t,*}\|$, $t \in \mathbb{N}$, where we recall that $\mathbf{x}_{t,*}$ is the unique optimal solution of (2.26) at time t. To this end, we introduce the well-known definition of path length $\phi_t := \|\mathbf{x}_{t,*} - \mathbf{x}_{t+1,*}\|$, which measures the temporal variability of the optimal solution of (2.26). Finally, we let $E_t := \mathbb{E}[\|\mathbf{e}_t\|]$ to make the notation lighter. The main result of the section is stated in the following.

Theorem 1 Let Assumptions *AS1-AS5* hold, and let $\{\mathbf{x}_t\}_{t\in\mathbb{N}}$ be a sequence generated by (2.30) for a given initial point $\mathbf{x}_0 \in \mathcal{X}_0$. Recall that $\alpha \in (0, \frac{2}{L})$ and $\zeta_t = \max\{|1-\alpha\mu_t|, |1-\alpha L_t|\}$. Then, the following holds for all $t \in \mathbb{N}$:

1(i) The mean $\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{x}_{*,t}\|\right]$ is bounded as:

$$\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{x}_{*,t}\|\right] \leq \beta_t \mathbb{E}[\|\mathbf{x}_0 - \mathbf{x}_{*,0}\|] + \sum_{i=1}^{t}[\kappa_i\phi_{i-1} + \alpha p\omega_i E_i],$$

where, defining $\rho_t(\alpha) := 1 - p + p\zeta_t$, we have

$$\beta_t := \prod_{i=1}^{t} \rho_i(\alpha), \quad \kappa_i = \begin{cases} \rho_i(\alpha) & \text{if } i = t \\ \prod_{k=i}^{t} \rho_k(\alpha) & \text{if } i \neq t \end{cases}, \quad \omega_i = \begin{cases} 1 & \text{if } i = t \\ \prod_{k=i+1}^{t} \rho_k(\alpha) & \text{if } i \neq t \end{cases}.$$

1(ii) For any $\delta \in (0,1)$, the following bound holds with probability $1-\delta$:

$$\|\mathbf{x}_t - \mathbf{x}_{t,*}\| \leq \log^{\theta_x}\left(\frac{2}{\delta}\right)\left(\frac{2e}{\theta_x}\right)^{\theta_x}\left(\eta(t)\|\mathbf{x}_0 - \mathbf{x}_{*,0}\| + \frac{1-\zeta^t}{1-\zeta}\sup_{0\leq i\leq t}\left\{\alpha\nu_{\mathrm{e},i} + p^{-1}\phi_i\right\}\right), \tag{2.31}$$

where $\zeta := \sup_{1\leq i\leq t}\zeta_i$, $\theta_x := \max\{1, \theta_\varepsilon, \theta_\xi\}$, $\nu_{\mathrm{e},t} := (2^{\theta_\varepsilon}\sqrt{M}\nu_{\varepsilon,t}) + (2^{\theta_\xi}\sqrt{M}\nu_{\xi,t})$, and where the function $t \mapsto \eta(t)$ is defined as $\eta(t) := \max_k\left\{\frac{(1-p+\zeta^k p)^{\frac{t}{k}}}{\sqrt{k}}\right\}$.

Proof of Theorem 1. The proof of the theorem utilizes the definition of sub-Weibull rv in Definition 1(ii). To derive the main result, it is first necessary to characterize the rv ζ^{Ω_t}, where $\Omega_t := \sum_{i=0}^{t-1} v_i$ and $\zeta := \sup_{1\leq i\leq t}\zeta_i$, $\zeta \in (0,1)$. We will find the parameters $\theta > 0$ and $\eta(t) > 0$ s.t. ζ^{Ω_t} can be modeled as $\zeta^{\Omega_t} \sim \mathrm{subW}(\theta, \eta(t))$. By definition, Ω_t is a binomial rv, i.e., $\Omega_t \sim \mathcal{B}(p,t)$ since it is the sum of t Bernoulli trials. Further, $\zeta^{\Omega_t} \in [\zeta^t, 1)$, which implies that ζ^{Ω_t} is a bounded rv. Hence, we can model ζ^{Ω_t} as a sub-Gaussian rv [165] and, thus, a sub-Weibull with $\theta = 1/2$.

Regarding $\eta(t)$, by the definition of the k-th moment of a rv, we have that the k-th moment of the bounded rv ζ^{Ω_t} is

$$\begin{aligned} \left\|\zeta^{\Omega_t}\right\|_k^k = \mathbb{E}\left[\left(\zeta^{\Omega_t}\right)^k\right] &= \mathbb{E}\left[\left(\zeta^k\right)^{\Omega_t}\right] \\ &\overset{(a)}{=} \sum_{h=0}^{t}(\zeta^k)^h\binom{t}{h}p^h(1-p)^{t-h} \\ &\overset{(b)}{=} \sum_{h=0}^{t}\binom{t}{h}(\zeta^k p)^h(1-p)^{t-h} = (1-p+\zeta^k p)^t, \end{aligned}$$

where (a) follows by the definition of expected value and probability mass function of the binomial rv; and, (b) uses the binomial identity. Thus,

$$\left\|\zeta^{\Omega_t}\right\|_k = (1-p+\zeta^k p)^{\frac{t}{k}}. \tag{2.32}$$

We can see from (2.32) that the k-th moment of ζ^{Ω_t}, for a fixed k, decays to zero as $t \to \infty$. On the other hand, if we fix a finite $t \in \mathbb{N}$, we have that $\left\|\zeta^{\Omega_t}\right\|_k \to 1$ as $k \to \infty$. By Definition 1(ii) of

the sub-Weibull rv, we have that

$$\eta(t) \geq \frac{\|\zeta^{\Omega_t}\|_k}{\sqrt{k}} = \frac{(1-p+\zeta^k p)^{\frac{t}{k}}}{\sqrt{k}}, \qquad \forall k \geq 1.$$

Therefore, $\eta(t)$ is a decreasing function of t, and it takes values in the set $(0,1)$; also, $\eta(t) \to 0^+$ for $k \to \infty$. Thus, for any given t and any finite k, we can choose $\eta(t) = \max_k \left\{ \frac{(1-p+\zeta^k p)^{\frac{t}{k}}}{\sqrt{k}} \right\} \in (0,1)$.

With this characterization in place, we now derive a bound for $d_t := \|\mathbf{x}_t - \mathbf{x}_{*,t}\|$. Notice that $\mathbf{x}_{*,t}$ satisfies the fixed-point equation $\mathbf{x}_{t,*} := \text{proj}_{\mathcal{X}_t}[\mathbf{x}_{t,*} - \alpha \nabla f_t(\mathbf{x}_{t,*})]$. Then,

$$\begin{aligned}
d_{t+1} &\overset{(a)}{=} \|v_t \,\text{proj}_{\mathcal{X}_{t+1}}[\mathbf{x}_t - \alpha(\nabla f_{t+1}(\mathbf{x}_t) + \mathbf{e}_t)] - \mathbf{x}_{*,t+1} + (1-v_t)\,\text{proj}_{\mathcal{X}_{t+1}}[\mathbf{x}_t]\,\| \\
&\overset{(b)}{=} \|\text{proj}_{\mathcal{X}_{t+1}}[\mathbf{x}_t] + v_t \,\text{proj}_{\mathcal{X}_{t+1}}[\mathbf{x}_t - \alpha(\nabla f_{t+1}(\mathbf{x}_t) + \mathbf{e}_t)] - v_t \text{proj}_{\mathcal{X}_{t+1}}[\mathbf{x}_t] - \mathbf{x}_{*,t+1} + v_t \mathbf{x}_{*,t+1} - v_t \mathbf{x}_{*,t+1}\| \\
&\overset{(c)}{\leq} \|\text{proj}_{\mathcal{X}_{t+1}}[\mathbf{x}_t] - \mathbf{x}_{*,t+1} + v_t \mathbf{x}_{*,t+1} - v_t \text{proj}_{\mathcal{X}_{t+1}}[\mathbf{x}_t]\,\| + v_t \|[\mathbf{x}_t - \alpha(\nabla f_{t+1}(\mathbf{x}_t) + \mathbf{e}_t)] - \mathbf{x}_{*,t+1}\| \\
&\overset{(d)}{\leq} (1-v_t)\|\mathbf{x}_t - \mathbf{x}_{*,t+1}\| + v_t \zeta_{t+1} \|\mathbf{x}_t - \mathbf{x}_{*,t+1}\| + v_t \alpha \|\mathbf{e}_t\| \\
&\overset{(e)}{\leq} \zeta_{t+1}^{v_t} d_t + \zeta_{t+1}^{v_t} \phi_t + v_t \alpha \|\mathbf{e}_t\|.
\end{aligned} \tag{2.33}$$

where (a) holds by (2.30); (b) by adding and subtracting $\mathbf{x}_{t+1,*}$; (c) by reorganizing terms, using the triangle inequality, and the non-expansiveness property of the projection operator into the second term; (d) by using the triangle inequality on the second term, using the fact that $\mathcal{I} - \alpha \nabla f_{t+1}$ is ζ_{t+1}-Lipschitz, and the non-expansiveness property of the projection; and, (e) holds by adding and subtracting $\mathbf{x}_{t,*}$, using the triangle inequality, by the definition of ϕ_t, and $\zeta_{t+1}^{v_t} := 1 - v_t + v_t \zeta_{t+1}$.

To show *1(i)*, take the expectation of (2.33) to obtain

$$\begin{aligned}
\mathbb{E}[d_{t+1}] &\leq \mathbb{E}[\zeta_{t+1}^{v_t} d_t + \zeta_{t+1}^{v_t} \phi_t + v_t \alpha \|\mathbf{e}_t\|] \\
&\overset{(a)}{=} \mathbb{E}[\zeta_{t+1}^{v_t} d_t] + \mathbb{E}[\zeta_{t+1}^{v_t} \phi_t] + \mathbb{E}[v_t \alpha \|\mathbf{e}_t\|] \\
&\overset{(b)}{=} \mathbb{E}[\zeta_{t+1}^{v_t}]\mathbb{E}[d_t] + \mathbb{E}[\zeta_{t+1}^{v_t}]\phi_t + \alpha \mathbb{E}[v_t]\mathbb{E}[\|\mathbf{e}_t\|] \\
&\overset{(c)}{\leq} \rho_{t+1}(\alpha)\mathbb{E}[d_t] + \rho_{t+1}(\alpha)\phi_t + \alpha p E_t,
\end{aligned} \tag{2.34}$$

where (a) holds by the linearity of the expected value; (b) by the independence of the rvs v_t and $\|\mathbf{e}_t\|$; and, (c) holds by the expected value of the Bernoulli rv v_t, where we also used $\rho_t(\alpha) \geq \zeta_t$.

Recursively applying (2.34), we have,

$$\mathbb{E}\left[d_t\right] \leq \beta_t \mathbb{E}[d_0] + \sum_{i=1}^{t} \kappa_i \phi_{i-1} + \alpha p \sum_{i=1}^{t} \omega_i E_i.$$

To show *1(ii)*, let $\zeta := \sup_{1\leq i\leq t}\{\zeta_i\} \in (0,1)$; then, $\zeta_t \leq \zeta \, \forall t$. Then

$$d_{t+1} \leq \zeta^{v_t} d_t + \zeta^{v_t}\phi_t + v_t \alpha \|\mathbf{e}_t\|, \tag{2.35}$$

almost surely, and iterating (2.35) we have

$$\begin{aligned} d_t &\leq \prod_{i=1}^{t} \zeta^{v_i} d_0 + \sum_{i=1}^{t}\prod_{k=i}^{t} \zeta^{v_k}\phi_{i-1} + \alpha \sum_{i=1}^{t}\prod_{k=i+1}^{t} \zeta^{v_k} v_i \|\mathbf{e}_i\| \\ &= \zeta^{\Omega_t} d_0 + \sum_{i=1}^{t} \zeta^{\left(\sum_{k=i}^{t} v_k\right)}\phi_{i-1} + \alpha \sum_{i=1}^{t} \zeta^{\left(\sum_{k=i+1}^{t} v_k\right)} v_i \|\mathbf{e}_i\|. \end{aligned} \tag{2.36}$$

Define the sub-sequence $\{i_j\}_{j=1}^{\Omega_t}$ with $v_{i_j} = 1$ for $j = 1, \ldots, \Omega_t$; i.e., $\{j\}$ are the indices of the iterations where an update is performed. Then,

$$\sum_{i=1}^{t} \zeta^{\left(\sum_{k=i+1}^{t} v_k\right)} v_i \|\mathbf{e}_i\| = \sum_{j=1}^{\Omega_t} \zeta^{\left(\sum_{k=i_j+1}^{t} v_k\right)} \|\mathbf{e}_{i_j}\|.$$

By definition of $\{i_j\}_{j=1}^{\Omega_t}$, between the times $i_j + 1$ and $t-1$ the total number of updates is $\Omega_t - j$; then, we rewrite (2.36) as follows:

$$\begin{aligned} d_t &\leq \zeta^{\Omega_t} d_0 + \alpha \sum_{j=1}^{\Omega_t} \zeta^{\Omega_t - j} \|\mathbf{e}_{i_j}\| + \sum_{i=1}^{t} \zeta^{\left(\sum_{k=i}^{t} v_k\right)}\phi_{i-1} \\ &\leq \zeta^{\Omega_t} d_0 + \alpha \sum_{j=1}^{t} \zeta^{j} \|\mathbf{e}_{t-i_j}\| + \sum_{i=1}^{t} \zeta^{\left(\sum_{k=i}^{t} v_k\right)}\phi_{i-1}, \end{aligned} \tag{2.37}$$

where $t - \Omega_t$ terms are added to the sum to remove the dependence on Ω_t. Recall that $\zeta^{\Omega_t} \sim \mathrm{subW}(1/2, \eta(t))$, where $t \mapsto \eta(t)$ is monotonically decreasing. By Assumptions *AS4-AS5* and Lemma 1, we have that $\|\mathbf{e}_t\| \sim \mathrm{subW}(\theta_{\mathrm{e}}, \nu_{\mathrm{e},t}) \, \forall t$, where $\theta_{\mathrm{e}} = \max\{\theta_\varepsilon, \theta_\xi\}$ and $\nu_{\mathrm{e},t} = (2^{\theta_\varepsilon}\sqrt{M}\nu_{\varepsilon,t}) + (2^{\theta_\xi}\sqrt{M}\nu_{\xi,t})$. Then, using Proposition 4, we get:

$$\zeta^{\Omega_t} d_0 + \alpha \sum_{j=1}^{t} \zeta^{j} \|\mathbf{e}_{t-i_j}\| \sim \mathrm{subW}(\theta', \nu'),$$

where $\theta' = \max\{1/2, \max\{\theta_\varepsilon, \theta_\xi\}\}$,

$\nu' = \eta(t)d_0 + \alpha\frac{1-\zeta^t}{1-\zeta}\sup_{0\le i\le t}\{(2^{\theta_\varepsilon}\sqrt{M}\nu_{\varepsilon,i}) + (2^{\theta_\xi}\sqrt{M}\nu_{\xi,i})\}$,

and where we used the closure of the sub-Weibull rv with respect to sum and product (with a scalar), and then the inclusion property.

To characterize the last term of (2.37) we have that

$$\sum_{i=1}^{t}\zeta^{\left(\sum_{k=i}^{t}v_k\right)}\phi_{i-1} = \sum_{j=1}^{\Omega_t-1}\zeta^j\sum_{i=i_j}^{i_{j+1}}\phi_i \le \sum_{j=1}^{t-1}\zeta^j\sum_{i=i_j}^{i_{j+1}}\phi_i.$$

Note that $\sum_{i=i_j}^{i_{j+1}}\phi_i$ is the sum of a given number of the deterministic path lengths ϕ_t; then we get

$$\sum_{i=i_j}^{i_{j+1}}\phi_i \le \sup_{0\le i\le t}\{\phi_i\}\sum_{i=i_j}^{i_{j+1}}1 = \sup_{0\le i\le t}\{\phi_i\}(i_{j+1}-i_j). \tag{2.38}$$

The geometric rv $i_{j+1}-i_j$ can be characterized as a sub-Weibull rv. Since the exponential distribution is the continuous analogue of geometric distribution, and by logarithmic properties, we have $(i_{j+1}-i_j) \le Z \sim \text{subW}(1, 1/p)$; by Proposition 4, we have that (2.38) $\sim$ subW$(1, \sup_{0\le i\le t}\{\phi_i\}\, p^{-1})$. Therefore, (2.37) follows a sub-Weibull distribution with parameters $\max\{1, \theta_e\}$ and $\eta(t)d_0 + \frac{1-\zeta^t}{1-\zeta}\sup_{0\le i\le t}\left\{\alpha\nu_{e,i} + p^{-1}\phi_i\right\}$. Using the high probability bound (2.28) in Proposition 5 the result follows. □

We note that, if $\alpha \in \left(0, \frac{2}{L}\right)$, then $\rho_t(\alpha) < 1$ for all t; in this case, letting $\rho(\alpha) := \sup_{1\le i\le t}\{\rho_i(\alpha)\}$, the claim of Theorem 1(i) implies that

$$\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{x}_{*,t}\|\right] \le \rho(\alpha)^t\mathbb{E}[\|\mathbf{x}_0 - \mathbf{x}_{*,0}\|] + \frac{1}{1-\rho(\alpha)}\sup_{0\le i\le t}\{\phi_i\} + \frac{\alpha p}{1-\rho(\alpha)}\sup_{0\le i\le t}\{E_i\} \tag{2.39}$$

where the term $\rho(\alpha)^t\mathbb{E}[\|\mathbf{x}_0 - \mathbf{x}_{*,0}\|] \to 0$ as $t \to \infty$. From (2.39), it can be seen that the error $\|\mathbf{x}_t - \mathbf{x}_{*,t}\|$ is asymptotically bounded in expectation, with an error bound that depends on the variability of the optimal solution, the mean of the norm of the gradient error, and the Bernoulli parameter p. We also draw a link with stability of stochastic discrete-time systems [160, 84] by noting that (2.39) establishes that the stochastic algorithm (2.30) renders the set $\{0\}$ exponentially input-to-state stable (E-ISS) in expectation.

Theorem 1(ii) asserts that $\|\mathbf{x}_t - \mathbf{x}_{*,t}\|$ is bounded in high probability. Since $t \mapsto \eta(t)$ is monotonically decreasing, $\eta(t) \to 0$ as $t \to \infty$; thus, Theorem 1(ii) provides an asymptotic error

bound that holds in high probability. We also note that the first term on the right-hand-side of (2.31) is a $\mathcal{KL}$ function [84]; upper-bounding the second term as $\frac{1}{1-\zeta}\sup_{0\leq i\leq t}\left\{\alpha\nu_{\mathrm{e},i}+p^{-1}\phi_i\right\}$, we obtain a result in terms of ISS in high-probability under a sub-Weibull error model.

2.2.4 Optimization with Concurrent Learning

In this section, we provide an example of a framework where the cost function is estimated using GPs during the execution of the online algorithm (2.30). We will show that the results of Theorem 1 directly apply to this framework.

To streamline exposition, suppose that the function $\mathbf{y}\mapsto C_t(\mathbf{y})$ is known, and the function $\mathbf{x}\mapsto U(\mathbf{x})$ is static but unknown (a similar approach can be used for time-varying functions). Furthermore, suppose that $U(\mathbf{x})=\sum_{m=1}^{M}u_m(x_m)$ where $u_m:\mathbb{R}\to\mathbb{R}$ is a cost associated with the m-th input. We then consider a concurrent learning approach where we estimate each function $x_m\mapsto u_m(x_m)$ via GP, based on noisy functional evaluations [147].

For completeness of this section, below we recalled the GPs definitions. As shown in Section 2.1.2, a GP is a stochastic process and is specified by its mean function and its covariance function [133]. Accordingly, let $u_m(x)$ be characterized by a GP, i.e., for any $x,x'\in\mathcal{X}\subseteq\mathbb{R}$, $\mu_m(x)=\mathbb{E}[u_m(x)]$ and $k_m(x,x')=\mathbb{E}[(u_m(x)-\mu_m(x))(u_m(x')-\mu_m(x'))]$. Let $\boldsymbol{\chi}_{m,t}=[x_{m,t_1}\in\mathcal{X},\ldots,x_{m,t_q}\in\mathcal{X}]^\top$ be the set of q sampling points at times $\{t_i\}_{i=1}^{q}\subset\{0,\ldots,t\}$; let $z_{m,t_i}=u(x_{m,t_i})+\varpi_{t_i}$, with $\varpi_{t_i}\overset{\text{iid}}{\sim}\mathcal{N}(0,\sigma^2)$ Gaussian noise, be the noisy functional evaluation at x_{m,t_i}; finally, define $\mathbf{z}_{m,t}=[z_{m,t_1},\ldots,z_{m,t_q}]^\top$. Then, the posterior distribution of $(u_m(x)|\boldsymbol{\chi}_{m,t},\mathbf{z}_{m,t})$ is a GP with mean $\mu_{m,t}(x)$, covariance $k_{m,t}(x,x')$, and variance $\varsigma^2_{m,t}(x)$ given by [133]:

$$\mu_{m,t}(x)=\mathbf{k}_{m,t}(x)^\top(\mathbf{K}_{m,t}+\sigma^2\mathbf{I})^{-1}\mathbf{z}_{m,t}, \tag{2.40a}$$

$$k_{m,t}(x,x')=k_m(x,x')-\mathbf{k}_{m,t}(x)^\top(\mathbf{K}_{m,t}+\sigma^2\mathbf{I})^{-1}\mathbf{k}_{m,t}(x'),$$

$$\varsigma^2_{m,t}(x)=k_m(x,x), \tag{2.40b}$$

where $\mathbf{k}_{m,t}(x)=[k_m(x_{m,t_1},x),\ldots,k_m(x_{m,t_q},x)]^\top$, and $\mathbf{K}_{m,t}$ is the positive definite kernel matrix $[k_m(x,x')]$. For example, using a squared exponential kernel, $k_m(x,x')$ is given by $k_m(x,x')=$

$\sigma_f^2 e^{-\frac{1}{2\ell^2}(x-x')^2}$, where the hyperparameters are the variance σ_f^2 and the characteristic length-scale ℓ [133].

The idea is then to utilize the posterior mean $\mu_{m,t}(x)$, computed via (2.40a) based on the samples collected up to the current time t, as an estimate of the function $u_m(x)$. Accordingly, the function $U(\mathbf{x})$ can be approximated at time t as $\mu_t(\mathbf{x}) = \sum_{m=1}^{M} \mu_{m,t}(x_m)$ and $s_t(\mathbf{x})$ in (2.30) can be set to $s_t(\mathbf{x}) = \nabla\mu_t(\mathbf{x})$. The resulting GP-based learning framework would involve the sequential execution of the online algorithm (2.30), with $s_t(\mathbf{x}) = \nabla\mu_t(\mathbf{x})$, and where the estimates $\{\mu_{m,t}(x)\}$ are updated via (2.40a) whenever a new functional evaluation becomes available. In this case, the m-th entry of the gradient error vector $\boldsymbol{\varepsilon}_t$ can be expressed as $\varepsilon_{m,t}(x_m) = \frac{du_m}{dx_m}(x_m) - \frac{d\mu_{m,t}}{dx_m}(x_m)$.

Since the function $u_m(x_m)$ is modeled as a GP, its derivative is also a GP [133]. For a given $x_m \in \mathbb{R}$, it follows that the error $\varepsilon_{m,t}(x_m)$ is a Gaussian random variable [133] (and, hence, sub-Gaussian [165]). Since the class of sub-Weibull rvs includes sub-Gaussian distributions by simply setting $\theta = 1/2$ [166], it follows that $\varepsilon_{m,t}(x_m) \sim \text{subW}(1/2, \nu_{\varepsilon,t})$, for some $\nu_{\varepsilon,t} > 0$.

Summarizing, when GPs are utilized to estimate the function $\mathbf{x} \mapsto U(\mathbf{x})$, Assumption *AS4* is satisfied (similar arguments hold if we utilize GPs to estimate the function $\mathbf{y} \mapsto C_t(\mathbf{y})$). In particular, it holds that $\|\boldsymbol{\varepsilon}_t\| \sim \text{subW}(\theta_\varepsilon, 2^{\theta_{\varepsilon,t}}\sqrt{M}\nu_{\varepsilon,t})$, with $\theta_\varepsilon = 1/2$.

2.3 Conclusions

In this chapter, we considered a feedback-based projected gradient descent algorithms to solve a time-varying optimization problem associated with a system modeled with an algebraic map. The algorithms relies on inaccurate gradient information and exhibits random updates. We derived bounds for the error between the iterate of the algorithm and the optimal solution of the optimization problem in expectation and in high probability, by modeling gradient errors as sub-Weibull rvs and missing measurements as Bernoulli rvs. We established a connection with results in the context of ISS in expectation and in high probability for discrete-time stochastic dynamical systems.

Chapter 3

Consensus-Based Data-Driven Online Optimization

We consider optimization problems associated with a network of systems, where each of the systems is shared by a number of individuals. Typically, such a multiuser problem includes a cost given by a sum of user-specific functions, and a set of constraints that capture physical or operational limits of the network (and, thus, that couple the users' decisions). For example, multiuser problems were considered in [90], and solved via primal-dual method based on a regularization of the Lagrangian function; and, a distributed resource allocation problem was investigated in [163] (where the communications between users' are susceptible to adversarial attacks). Similar formulations arise in a time-varying setting, where the cost and/or constraints evolve over time to reflect changes in the constraints or problem inputs [130, 148, 61, 146]. Specifically, unconstrained problems with a known time-varying cost are analyzed in [130] via a gradient descent method. A time-varying multiuser problem is presented in [148], where a double regularization both in the primal and in the dual space is employed to increase the convergence rate of the algorithm. In [61], convex optimization problems with known time-varying objective functions are solved using a real-time self-triggered control method. We refer the reader to the survey in [146], for a complete list of references on time-varying convex optimization.

Generally, one prerequisite for solving these problems is that the user-specific functions are known or properly crafted based on synthetic or average models. Instead, we consider a case where synthetic costs are not representative of the preferences of individual users, or fail to capture the diversity in their perception of comfort, safety, or dissatisfaction [12, 92, 94]. In this context, a

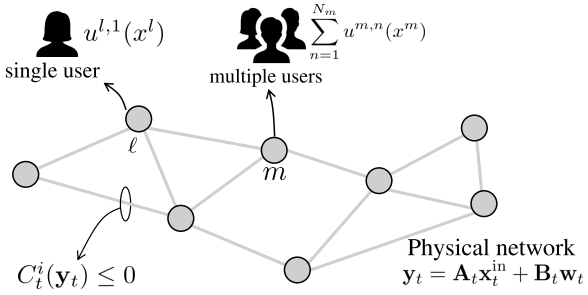


Figure 3.1: We consider a network of systems, coupled through physical or operational dependencies, where each system is shared by (and affects) a number of users.

learning method that leverages human feedback is explored in [12], where a robot observes data in the form of state-action and the discrete dynamics are learned during the execution of the algorithm. An example of field research on occupant satisfaction to the indoor temperature via recording of smiley-face polling station is presented in [92]. Moreover, [94] presents an overview of opportunities and risks of data-driven decision-making methods.

We tackle the problem of solving a network optimization problem when the users' cost functions are *unknown*, and develop an online algorithm where the cost functions are learned concurrently with the execution of the algorithmic steps. In particular, the learning procedure leverages "users' feedback", and utilizes learning tools from shape-constrained Gaussian Processes (GPs) [168]. The algorithm is implemented in a *distributed* fashion; this allows users sharing a system to agree on a solution that minimizes the sum of their (learned) functions, without revealing their preferences or their feedback. In this work, we consider the case where the users report their preferences trustworthy (and we do not consider adversarial behaviors). In addition to the feedback from the users, the algorithm leverages the ideas from, e.g., [27, 19], and utilizes measurements of network outputs instead of the network model.

The endeavors are motivated by a number of problems arising in power systems [167, 95, 27], charging of electric vehicles [42], and human-aware robot systems [106], just to name a few. While Section 3.1 will explain some examples, we stress here that an accurate model of the comfort/satisfaction level of users is typically unknown, and it may vary not only across individuals, but also for the same individual [12, 92]. For example, in an office building, the system of interest

might correspond to the heating, ventilation, and air-conditioning (HVAC) system per floor, and the perception of comfort may be different across occupants. In electric vehicle charging, the comfort function can model a preferred charging schedule that depends on the current energy price. In this context, shape-constrained GP is a powerful tool for nonparametric function estimation when the underlying function is presumed to be convex or monotonic [168]; this implies that each user has a preference for a set of solutions.

Overall, the main contributions of the chapter are as follows. *i)* We show how to estimate the users' functions via GPs and in particular, how to handle asynchronous and noisy data easily. *ii)* We develop an online algorithm based on a primal-dual method to solve the formulated time-varying network optimization problem; the algorithm is implemented online, and it is modified to accommodate measurements and learned functions. *iii)* We propose a consensus-based formulation that leads to a distributed online algorithm where users minimize the sum of their (learned) functions, without revealing their preferences or their feedback. *iv)* For the performance analysis, we view the algorithm as an inexact online primal-dual method where errors are due to the shape-constrained GP approximation and errors in the computation of the gradient of the GP. We derive bounds for a dynamic extension of the network regret [89, 93], and for the average constraint violation (see also, e.g., [43, 72, 143, 176] and references therein for other notions of dynamic regret). And, *v)* we showcase the proposed methodology in an example related to control of distributed energy resources in power grids [95].

In the context of bandit optimization, a static regret analysis is performed in [43], where the objective is to minimize a sequence of unknown convex cost functions and one has access to functional evaluations. Time-varying cost and constraints are considered in the context of a bandit setting in [63] and online saddle-point algorithms are utilized, where the gradient information is acquired via one-point or multi-point estimates. In [98], a gradient-free approach is presented to solve a multi-agent distributed constrained optimization problem where the goal is to cooperatively minimize the sum of time-changing local cost functions subject to time-varying coupled constraints. As an example of works in the context of zeroth-order methods, a multi-agent optimization problem

is analyzed in [156] where the objective is to minimize the average of the nonconvex local costs that depends on the joint actions of the agents. See also [103] for an overview of zeroth-order algorithms. Our approach is different from zeroth-order or bandit methods [63, 43, 103, 156, 98]; these would require multiple functional evaluations at each step to estimate the gradient of the users' costs – something not feasible for our problem. Our contribution is to use shape-constrained GPs to process feedback that is provided parsimoniously and at irregular intervals. The considered setting is different from [152], where GPs are utilized to maximize an unknown function.

A similar problem setup was studied in [117]; however, no constraints are considered (in fact, a gradient tracking scheme is utilized) and the user's cost is assumed to have a known quadratic structure a priori. We significantly extend our previous works [147, 123] by considering a distributed setting and constrained problems. Relative to the works [89, 93] on distributed online primal-dual algorithms, we consider problems where the optimal solution set changes in time (and it is not fixed in time), and with both consensus constraints and time-varying constraints that capture operational limits of the network. We also analyze the network regret when the online algorithm is *inexact*. Indeed, we can recover an asymptotic bound of $\mathcal{O}(T^{\frac{1}{2}})$ as in [89, 93] from our regret bound when the cost functions are known, the gradients are computed exactly, and the optimal solutions are not time-varying (with T the number of time steps). Our analysis can also be utilized to extend [156] to time-varying settings. Finally, we extend the findings of [19, 27] by considering consensus constraints (and accompanying distributed implementations), and by including shape-constrained GPs in the overall algorithmic framework.

The remainder of this chapter is organized as follows. Section 3.1 presents the time-varying problem, and Section 3.2 outlines the GP-based learning approach and the proposed online algorithm. Section 3.3 presents the performance analysis. Finally, Section 3.4 presents conclusions and further direction of this work.

3.1 Problem Statement

We consider a network of M systems or devices, with each system interacting with (or shared by) a number of users as illustrated in Figure 3.1. In particular, we denote by $\mathcal{N}_m = \{1, \ldots, N_m\}$ the set of $N_m \geq 1$ users interacting with the mth system. The input-output relationship of the network is modeled through the time-varying map

$$\mathbf{y}_t = \mathbf{A}_t \mathbf{x}_t^{\text{in}} + \mathbf{B}_t \mathbf{w}_t \tag{3.1}$$

where t is the time index; $\mathbf{x}_t^{\text{in}} := [x_t^1, \ldots, x_t^M]^\top$, with $x_t^m \in \mathbb{R}$ the controllable input of the mth system; $\mathbf{y}_t \in \mathbb{R}^Y$ is a vector of observables or outputs of the network; $\mathbf{w}_t \in \mathbb{R}^W$ is a vector of *unknown* exogenous inputs; and, $\mathbf{A}_t$ and $\mathbf{B}_t$ are possibly time-varying matrices (with $\mathbf{A}_t$ known and $\mathbf{A}_t \neq \mathbf{0}$) of appropriate dimensions. We assume that $t \in \mathcal{T} := \{k\Delta, k \in \mathbb{N}\}$, where $\Delta > 0$ is a given time interval.

Let $u^{m,n} : \mathbb{R} \to \mathbb{R}$ be a convex function representing the cost of a user $n \in \mathcal{N}_m$ associate with the mth device; this function can capture a sense of dissatisfaction, discomfort, or simply preferences depending on the specific application (a few examples will be provided shortly). The goal is then to compute the sequence of inputs that solves the following time-varying optimization problem [130, 148, 61] associated with the network:

$$\min_{\{x^m \in \mathcal{X}_t^m\}_{m=1}^M} \quad \sum_{m=1}^{M} \sum_{n=1}^{N_m} u^{m,n}\left(x^m\right) \tag{3.2a}$$

$$\text{s. to: } C_t^i\left(\mathbf{A}_t \mathbf{x}^{\text{in}} + \mathbf{B}_t \mathbf{w}_t\right) \leq 0, \; i = 1, \ldots, N_C \tag{3.2b}$$

where $\{\mathcal{X}_t^m\}_{m=1}^M$ are time-varying convex and compact sets for the inputs, and $C_t^i : \mathbb{R}^M \to \mathbb{R}$ is a time-varying function parameterized by $\mathbf{w}_t$. In particular, (3.2b) models operational constraints associated with the network output $\mathbf{y}_t$. The main goal of (3.2) is to generate a sequence of inputs that minimizes the cost of the users, while respecting pertinent network constraints. To this end, this work focuses on addressing the following four main challenges related to (3.2):

• **Challenge 1:** The variable x_m is *coupled* across the users $\mathcal{N}_m$. Our goal is to solve (3.2) in

a distributed fashion, where users are not required to share information about their costs $\{u_{m,n}\}$ (and their functional evaluations) with the device or the network operator.

• **Challenge 2:** The costs $\{u^{m,n}\}$ are *unknown*. One may utilize synthetic cost functions for users' preferences, comfort, or satisfaction based on statistical models; however, these synthetic costs may fail to capture the diversity in the users' perception, preferences, and goals.

• **Challenge 3:** The exogenous inputs $\mathbf{w}_t$ may be *unknown* or partially known.

• **Challenge 4:** The constraints C_t^i and the sets $\{\mathcal{X}_t^m\}$ may change at each time t; the interval Δ may not be sufficient to solve each instance of the time-varying problem (3.2) to convergence.

To address the challenges 1-4 above, this chapter will develop a *consensus-based online* algorithm where the cost functions of the users are *learned* concurrently with the execution of the algorithm, and *measurements* of $\mathbf{y}_t$ are utilized in the algorithmic updates.

Before proceeding, we list representative examples of applications in networked systems in which the problem (3.2) and the accompanying online optimization framework considered in this work are particularly well-suited.

Example 1 (Power grids): In the context of power grids, (3.2) may capture a demand response task [167, 95] or a real-time optimal power flow problem [27]. In this case, $\mathbf{x}^{\text{in}}$ are the power setpoints of distributed energy resources, $\mathbf{w}_t$ is a vector of powers consumed by uncontrollable loads, and y_t may represent the total power as $y_t = \mathbf{1}^\top \mathbf{x}^{\text{in}} + \mathbf{1}^\top \mathbf{w}_t$ and/or voltage magnitudes (one can also compute the matrix $\mathbf{A}_t$ based on a linearized AC model). Constraints (3.2b) may ensure that the net power follows a given automatic gain control or demand response signal; e.g., $C_t^i(y) = (y - y_{\text{ref},t})^2 - \zeta$, where $y_{\text{ref},t}$ is a given reference signal and $\zeta > 0$ is an error tolerance. One may also have voltage constraints. The costs $\{u^{m,n}\}$ model discomfort, e.g., the indoor temperature. The number of users depends on the particular setting, for example, one may expect a large number of users sharing a conference room in an office building. The users that share a device m agree on the input x_m; for example, people sharing a room agree on a temperature and, thus, on the use of an HVAC system. □

Example 2 (Electric vehicle (EV) charging): The formulation (3.2) may represent the problem

of charging a fleet of vehicles at a charging station; the variable x_m represents the charging rate (or the expected time for full charging) of a vehicle, constraint (3.2b) captures limits on the total power consumed by the charging station, while $\{u^{m,n}\}$ capture the dissatisfaction for a given expected time of charging completion [42]. If an EV is shared by multiple users, x^m represents the agreed expected charging time x^m. □

Additional examples in transportation systems, communication systems, and human-aware robot systems are not included for space limitations.

We now proceed with the reformulation of (3.2) into a consensus-based problem. To this end, let $x^{m,n}$ be an auxiliary variable representing the input for the mth system preferred by user $n \in \mathcal{N}_m$; with the auxiliary variables in place, stack the optimization variables in the column vector

$$\mathbf{x}_t = \left[x_t^1, x_t^{1,1}, \ldots, x_t^{1,N_1}, \ldots, x_t^M, x_t^{M,1}, \ldots, x_t^{M,N_M}\right]^\top.$$

Accordingly, (3.1) can be rewritten as $\mathbf{y}_t = \bar{\mathbf{A}}_t \mathbf{x}_t + \mathbf{B}_t \mathbf{w}_t$, where $\bar{\mathbf{A}}_t$ is an augmented version of $\mathbf{A}_t$ with appropriate zero entries.

Then, (3.2) can be equivalently reformulated as:

$$\min_{\{x^m,\, x^{m,n} \in \mathcal{X}_t^m\}} \quad \sum_{m=1}^{M} \sum_{n=1}^{N_m} u^{m,n}\left(x^{m,n}\right) \tag{3.3a}$$

$$\text{s. to: } C_t^i\left(\bar{\mathbf{A}}_t \mathbf{x} + \mathbf{B}_t \mathbf{w}_t\right) \leq 0,\ i = 1, \ldots, N_C \tag{3.3b}$$

$$x^m = x^{m,n},\ \forall\, n \in \mathcal{N}_m,\ \forall\, m = 1, \ldots, M \tag{3.3c}$$

where (3.3c) defines consensus constraints for the mth system and its users. We note that the choice of posing the consensus constraints as in (3.3c) is not generic; the particular structure of (3.3c), which can be modeled as a star graph (with the system as the central node) will facilitate the development of a closed-loop algorithm where $\{x_m\}$ will be physical inputs (as in (3.1)) and $\{x^{m,n}\}$ will be auxiliary control variables.

For future developments, define $f(\mathbf{x}) := \sum_{m=1}^{M} \sum_{n=1}^{N_m} u^{m,n}\left(x^{m,n}\right)$ and $\mathcal{X}_t := \mathcal{X}_t^1 \times \cdots \times \mathcal{X}_t^M \subseteq \mathbb{R}^M$. For each device m, we define the incidence matrix $\mathbf{D}^m \in \mathbb{R}^{N_m \times (1+N_m)}$, which represents the relation between the device and the users sharing that device. For example, if the device 1 is shared

by 2 users we have that

$$\mathbf{D}^1 = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}.$$

By organizing these incidence matrices in a block diagonal matrix, we construct the augmented incidence matrix $\mathbf{D} = \text{diag}(\mathbf{D}^1, \mathbf{D}^2, \ldots, \mathbf{D}^M)$. Then, (3.3c) can be compactly written as $\mathbf{Dx} = \mathbf{0}$, capturing consensus constraints over M systems.

The next section outlines the proposed algorithm to solve the time-varying network optimization problem (3.2). We note that in what follows we consider the case where $N_C = 1$ to simplify exposition and notation; however, the technical arguments of this work straightforwardly extend to problems with multiple constraints on $\mathbf{y}_t$.

3.2 Distributed Online Algorithm

In this section, we present our *consensus-based online* algorithm for solving (3.3). We first explain how to derive a distributed online algorithm based on a primal-dual method; then, we elaborate on how to estimate the cost functions based on feedback received from the users at infrequent times.

3.2.1 Online Primal–Dual Algorithm

We start by defining the time-varying Lagrangian function associated with (3.3) as follows:

$$\mathcal{L}_t(\mathbf{x}_t, \nu_t, \boldsymbol{\lambda}_t) := f(\mathbf{x}_t) + \nu_t \, C_t \left(\bar{\mathbf{A}}_t \mathbf{x}_t + \mathbf{B}_t \mathbf{w}_t \right) + (\mathbf{D}\mathbf{x}_t)^\top \boldsymbol{\lambda}_t$$

where $\nu_t \in \mathbb{R}_+$, $\boldsymbol{\lambda}_t \in \mathbb{R}^N$ are the dual variables associated with the constraint (3.3b), and the M consensus constraints (3.3c), respectively.

Recall that $t \in \mathcal{T} := \{k\Delta, k \in \mathbb{N}\}$ (and take $\Delta = 1$ for simplicity of exposition). For a given step size $\alpha > 0$, a *model-based* online projected primal-dual algorithm involves the sequential

execution of the following steps:

$$\mathbf{x}_{t+1} = \text{proj}_{\mathcal{X}_t}\left\{\mathbf{x}_t - \alpha\left(\nabla_{\mathbf{x}} f(\mathbf{x}_t) + \nu_t(\bar{\mathbf{A}}_t)^\top \nabla_{\mathbf{x}} C_t(\bar{\mathbf{A}}_t\mathbf{x}_t + \mathbf{B}_t\mathbf{w}_t) + \mathbf{D}^\top \boldsymbol{\lambda}_t\right)\right\}, \tag{3.4a}$$

$$\nu_{t+1} = \text{proj}_{\Psi_t}\left\{\nu_t + \alpha C_t\left(\bar{\mathbf{A}}_t\mathbf{x}_t + \mathbf{B}_t\mathbf{w}_t\right)\right\}, \tag{3.4b}$$

$$\boldsymbol{\lambda}_{t+1} = \text{proj}_{\Lambda_t}\left\{\boldsymbol{\lambda}_t + \alpha\mathbf{D}\mathbf{x}_t\right\}, \tag{3.4c}$$

where Ψ_t is a convex and compact set, and $\Lambda_t = \Lambda_t^{1,1} \times \cdots \times \Lambda_t^{1,N_1} \times \cdots \times \Lambda_t^{M,N_M}$, with $\Lambda_t^{m,n}$ also a convex and compact set constructed as explained shortly in Section 3.3 (and in, e.g., [90] and [89]).

We note that the steps (3.4a) and (3.4b) require one to know $\mathbf{w}_t$ and $\mathbf{B}_t$ (as explained in Challenges 3 and 4). To bypass this hurdle, we adopt the strategy proposed in, e.g., [27, 19]. To this end, assume that $\mathbf{x}_t^{\text{in}}$ are implemented as inputs to the systems at time t, and let $\hat{\mathbf{y}}_t$ represent a *measurement* of $\mathbf{y}_t$ collected at time t. Then, ν_{t+1} can be computed as $\nu_{t+1} = \text{proj}_{\Psi}\left\{\nu_t + \alpha C_t\left(\hat{\mathbf{y}}_t\right)\right\}$; it is clear that information about $\mathbf{w}_t$ and $\mathbf{B}_t$ is not required in this case. We also note that the step (3.4a) would require the gradient of the functions $\{u^{m,n}\}$, which are not known (see Challenge 2). Let $\hat{u}_t^{m,n}(x_t^m)$ represent an estimate of the function $u^{m,n}(x_t^m)$ available at x_t^m, and let $g_t^{m,n}$ denote an *estimate* of the gradient (or derivative, in this case) of $u^{m,n}(x_t^m)$ at time t. With these definitions, the steps in (3.4) can be suitably modified to accommodate measurements of $\mathbf{y}_t$ and estimates of the gradient of the cost function:

$$x_{t+1}^m = \text{proj}_{\mathcal{X}_t^m}\left\{x_t^m - \alpha\left(\nu_t(\mathbf{a}_t^m)^\top \nabla C_t(\hat{\mathbf{y}}_t) + \sum_{n=1}^{N_m} \lambda_t^{m,n}\right)\right\} \tag{3.5a}$$

$$x_{t+1}^{m,n} = \text{proj}_{\mathcal{X}_t^m}\left\{x_t^{m,n} - \alpha\left(g_t^{m,n} - \lambda_t^{m,n}\right)\right\}\ \forall\, n \in \mathcal{N}_m,\, \forall\, m, \tag{3.5b}$$

$$\nu_{t+1} = \text{proj}_{\Psi_t}\left\{\nu_t + \alpha C_t\left(\hat{\mathbf{y}}_t\right)\right\}, \tag{3.5c}$$

$$\lambda_{t+1}^{m,n} = \text{proj}_{\Lambda_t^{m,n}}\left\{\lambda_t^{m,n} + \alpha\left(x_t^m - x_t^{m,n}\right)\right\}, \forall\, n \in \mathcal{N}_m,\, \forall\, m, \tag{3.5d}$$

where $\mathbf{a}_t^m$ is the mth column of the matrix $\mathbf{A}_t$, and we note that step (3.5a) is performed in parallel at each system. While measurements of $\mathbf{y}_t$ are utilized in the primal step (3.5a) and the dual step (3.5c), an estimate of the gradient $g_t^{m,n}$ will be obtained by leveraging feedback from the users as explained in the following subsection. Subsequently, Section 3.2.3 will overview the steps of the proposed algorithm.

3.2.2 Online Learning via Shape-Constrained Gaussian Processes

We assume that the feedback of the user comes in the form of a (possibly noisy) functional evaluation; that is, for a given point x, one may receive from a user n interacting with the device m a rating z given by $z = u^{m,n}(x) + \epsilon$. With this model, a popular way to obtain the gradient of $u^{m,n}$ at a point x is via zeroth-order methods (see, e.g., [63, 103] and references therein). However, zeroth-order methods are not well-suited for our algorithm, because they would require feedback from each user at *each iteration* of the algorithm (either a single functional evaluation or more, depending on the particular method utilized). Instead, we consider a more realistic case where users may provide feedback more parsimoniously and at irregular intervals.

With this setting in mind, we consider utilizing the noisy functional evaluations provided by a user to estimate the function $u^{m,n}$ via GPs [133]. In particular, we apply a GP with specific constraints on the shape of the function. As explained in [168], shape-constrained GP offers a powerful tool for nonparametric function estimation when the underlying function is presumed to be convex or monotonic. In our case, convexity or monotonicity implies that each user has a preference for a particular solution (or a convex set of solutions), and the dissatisfaction or discomfort increases as the system deviates from such a preferred point. In the following, we first briefly introduce GPs, and then we provide the main equations to estimate $u^{m,n}$ via a shape-constrained GPs. Since we utilize a GP per function $u^{m,n}$, we drop the subscripts m, n for simplicity of exposition. We also note that the extension to the multi-dimensional shape-constrained GP is possible (see, for example, [168]); however, it is left as future work.

In the next section we use the definition of GP presented in Section 2.1.2 and the shaped-constrained GP definition in Section 2.1.3. Similar as presented in the previous sections, the hyperparameters ℓ and σ_f^2 of the GP can be estimated, for example, by using the maximum likelihood estimator [133]. On the other hand, L_u and γ_u can be estimated via cross-validation. The locations of the virtual derivative points are defined beforehand. Note that the shape-constrained GP is a GP whose second-order derivative is constrained on a set of points. When we consider a GP

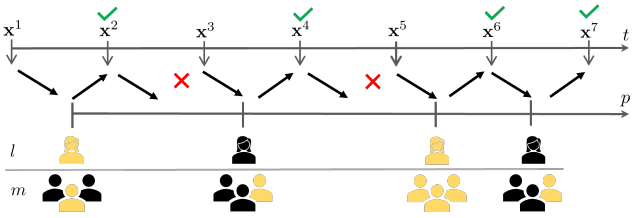


Figure 3.2: Feedback (color-coded in yellow) from users may come sporadically, and each user may provide feedback at different times (green ticks indicate that a functional evaluation is received, red crosses indicate that a functional evaluation is not received). The functions $\{u_{m,n}\}$ are estimated concurrently with the execution of the algorithm.

prior with a differentiable kernel function, we have the advantage that their derivative processes are also GPs and are jointly Gaussian with the original processes [168].

In this context, shape-constrained GPs guarantee that the posterior mean function $\bar{\mu}(x^\circ)$ is practically smooth and strongly convex [168]. With a limited number of enforcing points $\mathbf{s}$, the mean has the functional properties we require up to a very (almost negligible) small error, which is smaller and smaller increasing the number of data points $\mathbf{x}_p$. The error will be incorporated into the gradient estimation inaccuracies. We will now incorporate the estimates of the users' function in the algorithm as explained next.

Remark 1 One may want to select a dense set of points $\mathbf{s}$ that ensures a uniform covering of the domain. However, it is important to note that the number of points q influences the dimensions of the matrices $\mathbf{K^{20}}$, $\mathbf{K^{02}}$, and $\mathbf{K^{22}}$; therefore, q should be selected to ensure a sufficiently dense covering, but based on computational complexity considerations. Imposing the constraint uniformly over the domain would lead to infinite-dimensional matrices $\mathbf{K^{20}}$, $\mathbf{K^{02}}$, and $\mathbf{K^{22}}$, thus rendering the implementation of the shape-constrained GP regression infeasible.

Remark 2 In this work, we use the SE kernel. In this case, two hyperparameters are required for $u_{m,n}$: ℓ and σ_f^2. These two hyperparameters are learned by maximizing the likelihood [133], i.e., $\min_{\ell,\sigma_f} \Pr(\mathbf{z}|\mathbf{x}_p;\ell,\sigma_f)$. Note that the ability to impose shape constraints on the posterior GP is not affected by the kernel function or the hyperparameters; see [168].

Remark 3 It is important to note that we can only constrain the derivative at a discrete set of input point, in this case $\mathbf{s}$. The reason is that if we constrain the derivative in an infinite set of point on the domain, we will have infinite matrix on the evaluation of (2.5), specifically on the evaluation of the covariance matrices $\mathbf{K}$, then we need to enforce discretization on the input set. Therefore, the constraints can not be applied at all possible inputs, then the shape constrained GP realizations will not strictly follow the constraint. However, we would like to emphasise that the resulting posterior of the constraint GP can be consider uniform on the domain for all practical purposes if a dense enough set of constrained inputs $\mathbf{s}$ is chosen [168].

3.2.3 Distributed Gaussian Processes-Based Online Algorithm

Using the shape-constrained GPs machinery described in the previous section, the idea is to utilize the posterior mean (2.5a) as a surrogate for $u^{m,n}$. In particular, for a given user $n \in \mathcal{N}_m$, the posterior mean (2.5a) is computed based on the $p_t^{m,n}$ functional evaluations $\{u^{m,n}(x_{t_i}^m) + \epsilon_i\}_{i=1}^{p_t^{m,n}}$, $t_i \in \{1, \ldots, t\}$, provided by the user up to time t; that is, we set $\hat{u}_t^{m,n}(x) = \bar{\mu}^{p_t^{m,n}}(x)$. As pictorially illustrated in Figure 3.2, functional evaluations are provided sporadically by each user, and each user may provide feedback at different times. Furthermore, feedback is not required at each step of the online algorithm as in zeroth-order methods.

Once the posterior mean is available, an estimate of the gradient can be obtained via, e.g., multivariable zeroth-order methods when $\mathbf{x}_t^m \in \mathbb{R}^n$, or via finite difference when $x_t^m \in \mathbb{R}$. As an example of the latter, one way to obtain $g_t^{m,n}$ is:

$$g_t^{m,n} = \frac{\hat{u}_t^{m,n}(x_t^m + \delta/2) - \hat{u}_t^{m,n}(x_t^m - \delta/2)}{\delta} \tag{3.6}$$

with δ a preselected parameter. Note that even though $\hat{u}_t^{m,n}$ is available, and therefore we can obtain the derivative $g_t^{m,n}$ with arbitrarily high accuracy, we need to be also considerate of computational overhead. Typically a central derivative as in (3.6) represents a good trade-off.

The overall algorithm with feedback from both, users and the network, is tabulated as Algorithm 2. In terms of operation, the updates (3.5a) and (3.5c) are implemented at each system

m, with $\{x_t^m\}$ being physical inputs for the system; on the other hand, the update (3.5b) is implemented at the user's side. The updates (3.5d) are implemented by both users and systems, and users and systems are required to exchange the local variables x_t^m and $x_t^{m,n}$.

3.3 Performance Analysis

In this section, we investigate the convergence of the online algorithm presented in Section 3.2. To this end, the following standard assumptions are presumed.

AS1: The set $\mathcal{X}_t$ is convex and compact for all t.

AS2: The functions $\mathbf{x} \mapsto f(\mathbf{x})$ and $\mathbf{x} \mapsto C_t(\bar{\mathbf{A}}_t\mathbf{x}+\mathbf{B}_t\mathbf{w}_t)$ are convex and continuously differentiable for all $\{x^m, x^{m,n} \in \mathcal{X}_t^m\}$.

The previous assumptions imply that f and C_t are Lipschitz continuous, their gradients are also bounded by their Lipschitz constant, and they are uniformly bounded for all $\mathbf{x}^t \in \mathcal{X}^t$ and for all t. Let L be the Lipschitz constant of f. The next assumption pertains to each user-system star network.

AS3: Each system m is connected with its users via a star network (induced by (3.3c)). The largest singular value of the incidence matrix $\mathbf{D}$ is Ω.

Regarding Assumption *AS3*, each system-user network has a diameter $h^m \leq 2$.

AS4: Slater's constraint qualification holds $\forall\ t$.

AS5: ([89]) The convex set Λ_t is included in a 2-norm ball of radius B_λ, where $B_\lambda \geq N_{\max}h_{\max}LM+1$, $h_{\max} = \max\limits_{i=1,\ldots,M}\{h_i\}$, and $N_{\max} = \max\limits_{i=1,\ldots,M}\{N_i\}$ for all t.

AS6: ([90]) The set Ψ_t for the dual variable ν is convex and compact for all t.

We note that, under the Slater's constraint qualification, the dual variables ν_t are bounded as shown in [90, Sec. 3.1]. Additional comments on Assumptions *AS5*-*AS6* will be provided in the next subsection.

We tackle the analysis by viewing our algorithm as an online primal-dual with *inexact* gradient information [21]. In particular, the approximate gradient (obtained based on a finite difference method based on the estimated functions $\{\hat{u}_t^{m,n}\}$) can be expressed as $\mathbf{g}_t = \nabla f(\mathbf{x}_t) + \mathbf{e}_t$, where $\nabla f(\mathbf{x}_t)$ is the true gradient and $\mathbf{e}_t$ is a stochastic vector. Before stating appropriate assumptions, define the filtration $\mathcal{F}_t = \{\mathbf{e}_1, \ldots, \mathbf{e}_{t-1}\}$. The following assumption is made.

AS7: $\exists\ \bar{E}_t < \infty$ such that $\mathbb{E}[\|\mathbf{e}_t\|^2 | \mathcal{F}_t] \leq \bar{E}_t$, $\forall\ t$.

Notice that the error $\mathbf{e}_t$ is not assumed to have zero mean, since the GP may introduce a bias in the estimated gradient. Assumption *AS7* presumes that the expected value of $\|\mathbf{e}_t\|^2$ is bounded; this is consistent with standard assumptions in the context of inexact gradient methods and stochastic gradient methods (see, e.g., [21, 70]). We also note that $\bar{E}_t$ can be taken to be the Bayes errors for regression with GPs; in fact, [168] showed that the error of the GP regression is higher than the error incurred by the shape-constrained GP regression. We point out that Assumption *AS7* is also supported by empirical evidence; an example for the shape-constrained GP is provided in Figure 3.3, which illustrates the estimated function and the estimated derivative for different numbers of noisy functional evaluations. With these standard assumptions in place, the next section will provide performance bounds for the proposed algorithm.

3.3.1 Dynamic Regret

We characterize the performance of Algorithm 2 using the so-called dynamic regret (see, e.g., [143, 176, 89, 93] and references therein). In particular, given the distributed nature of our algorithm, we consider an extension of the network regret [89, 93] to a time-varying setting. To this end, define the *dynamic regret* per user $j \in \mathcal{N}_m$ interacting with the system m, with respect to an optimal solution of (2.14) for the system m at time t, i.e., $x_{t,*}^m$, as

$$\mathrm{Reg}_T^{m,j} := \sum_{t=1}^{T} \sum_{i=1}^{N_m} u^{m,i}(x_t^{m,j}) - \sum_{t=1}^{T} \sum_{i=1}^{N_m} u^{m,i}(x_{t,*}^m). \tag{3.7}$$

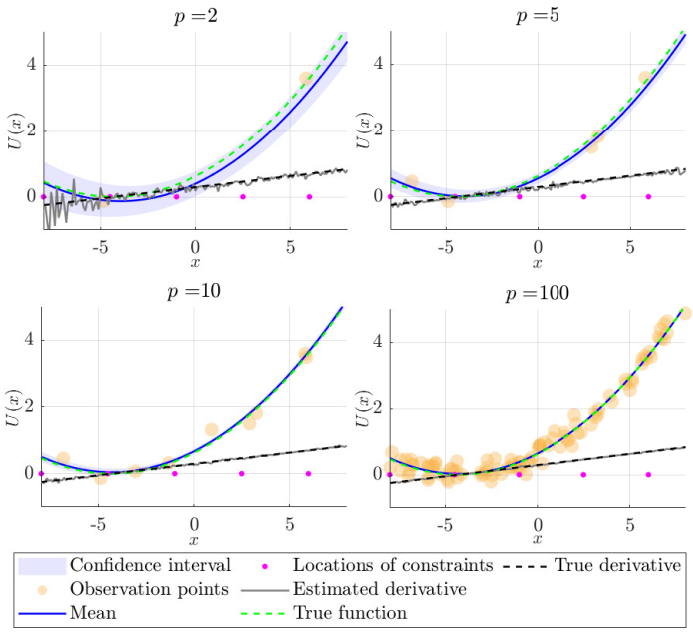


Figure 3.3: Example of the derivative estimation using shape-constrained GPs with hyperparameters $\sigma_f = 1$ and $\ell = 10$. As expected, the estimated derivative error decreases with the increase of the number of observations p. In this case, shape-constrained GP outperforms GP for low numbers of samples.

Then, based on (3.7), we define the *dynamic network regret* for the system m as [89, 93]

$$\mathrm{Reg}_T^m = \frac{1}{N_m}\sum_{j=1}^{N_m}\mathrm{Reg}_T^{m,j} = \frac{1}{N_m}\sum_{t=1}^{T}\sum_{i,j=1}^{N_m} u^{m,i}(x_t^{m,j}) - \sum_{t=1}^{T}\sum_{i=1}^{N_m} u^{m,i}(x_{t,*}^m). \tag{3.8}$$

By using (3.8), the dynamic *global network regret* after T iterations can be defined as

$$\mathrm{Reg}_T := \sum_{m=1}^{M}\mathrm{Reg}_T^m. \tag{3.9}$$

To analyze the global network regret, let $\mathbf{x}_{t,*}$ be an optimal solution of (3.3) at time t, and consider the following standard definitions of the path length, which captures the drift of the optimal solutions over T time steps

$$\Phi_T := \sum_{t=1}^{T}\|\mathbf{x}_{t,*} - \mathbf{x}_{t+1,*}\|, \qquad \Upsilon^T := \sum_{t=1}^{T}\|\mathbf{x}_{t,*} - \mathbf{x}_{t+1,*}\|^2. \tag{3.10}$$

Further, consider the following definitions for the cumulative expected error in the gradient:

$$\xi_T := \sum_{t=1}^{T}\mathbb{E}[\|\mathbf{e}_t\||\mathcal{F}_t], \qquad \Xi_T := \sum_{t=1}^{T}\bar{E}_t. \tag{3.11}$$

Algorithm 2: GP-based online primal-dual method

Initialize: $\mathbf{x}_1$, $\boldsymbol{\lambda}_1 = \mathbf{0}$, $\nu_1 = 0$, set the constant step size α; prior on $\{\hat{u}_1^{m,n}\}$ if available.
for $t = 1, 2, \ldots$ **do**
  Collect measurement of $\mathbf{y}_t$
  for each system $m = 1, 2, \ldots M$ **do**
    Update the input x_t^m via (3.5a)
    Send x_t^m to users $n \in \mathcal{N}_m$
    Update ν_t via (3.5c)
    for each user $n \in \mathcal{N}_m$ **do**
      if Feedback is given by user n
        Update $\hat{u}_t^{m,n}$ via (2.5a)
      else
        Keep $\hat{u}_t^{m,n} = \hat{u}_{t-1}^{m,n}$
      end if
      Estimate $g_t^{m,n}$ via (3.6)
      Update $x_t^{m,n}$ via (3.5b)
      Update $\lambda_t^{m,n}$ via (3.5d)
      Send $x_t^{m,n}$ to system m
    end for (users)
  System m updates $\lambda_t^{m,n}$ via (3.5d)
  end for (systems)
end for (time)

With these metrics in place, we start by bounding the norm of the true gradient of the time-varying Lagrangian with respect to the primal variable for any $\mathbf{x} \in \mathcal{X}_t$, $\nu \in \Psi_t$ and $\boldsymbol{\lambda} \in \Lambda_t$. Let $J := \sup_{t\in\mathcal{T}} \max_{\mathbf{x}\in\mathcal{X}_t} \|\nabla C_t(\mathbf{y}_t(\mathbf{x}))\|$ by Assumptions *AS1-AS2*, $B_\lambda := \sup_{t\in\mathcal{T}} \max_{\lambda\in\Lambda_t} \|\boldsymbol{\lambda}\|$ and $B_\nu := \sup_{t\in\mathcal{T}} \max_{\nu\in\Psi_t} |\nu|$ under Assumptions *AS5-AS6*, then we have

$$\begin{aligned} \|\nabla_{\mathbf{x}} \mathcal{L}_t(\mathbf{x}, \nu, \boldsymbol{\lambda})\| &\leq \|\nabla f(\mathbf{x})\| + \nu \|\nabla C_t(\mathbf{y}_t(\mathbf{x}))\| + \|\mathbf{D}^\top \boldsymbol{\lambda}\| \\ &\leq L + B_\nu J + \Omega B_\lambda := \Gamma_x, \end{aligned} \tag{3.12}$$

where, the last inequality holds by using the triangle and Cauchy-Schwarz inequality, and Assumption *AS3*. The term $\Gamma_x = L + B_\nu J + \Omega B_\lambda$ is an upper bound for the gradient of the Lagrangian with respect to $\mathbf{x}$.

Similarly, we bound the norm of the gradient of the Lagrangian with respect to the dual variables $\boldsymbol{\lambda}$ and ν for any $\mathbf{x} \in \mathcal{X}_t$, $\nu \in \Psi_t$ and $\boldsymbol{\lambda} \in \Lambda_t$. Let $\boldsymbol{\kappa} = [\nu, \boldsymbol{\lambda}^\top]^\top$, $|C_t(\mathbf{y}_t(\mathbf{x}))| \leq H < \infty$ for all t by Assumption *AS1* and Assumption *AS2*, and $B_x := \sup_{t\in\mathcal{T}} \max_{\mathbf{x}\in\mathcal{X}^t} \|\mathbf{x}\|$ under Assumption *AS1*.

Then, $\|\nabla_\kappa \mathcal{L}_t(\mathbf{x}, \boldsymbol{\kappa})\|^2$ can be bounded by

$$\|\nabla_\kappa \mathcal{L}_t(\mathbf{x}, \boldsymbol{\kappa})\|^2 = \left\| \begin{matrix} \nabla_\lambda \mathcal{L}_t(\mathbf{x}, \nu, \boldsymbol{\lambda}) \\ \nabla_\nu \mathcal{L}_t(\mathbf{x}, \nu, \boldsymbol{\lambda}) \end{matrix} \right\|^2 \leq \Omega^2 B_x^2 + H^2 := \Gamma_\kappa, \tag{3.13}$$

where we used the Cauchy-Schwarz inequality, Assumption *AS1*, and Assumption *AS3*. Further, let $\operatorname{diam}(\mathcal{X}_t) \leq D_{t,x}$ where $D_{t,x} \leq D_x < \infty$ by Assumption *AS1*. The term $\Gamma_\kappa = \Omega^2 B_x^2 + H^2$ is an upper bound for the gradient of the Lagrangian with respect to $\boldsymbol{\kappa}$.

With these notations in place, we are now ready to state the main result for the dynamic regret.

Theorem 2 Let Assumptions *AS1*-*AS7* hold. Set $\boldsymbol{\lambda}_1 = \mathbf{0}$, $\nu_1 = 0$. Then, the dynamic network regret after T iterations is upper bounded by

$$\begin{aligned} \mathbb{E}[\mathrm{Reg}_T] \leq \frac{1}{2\alpha}(\|\mathbf{x}_1 - \mathbf{x}_{1,*}\|^2 + B_\lambda^2 + B_\nu^2) + \frac{\alpha}{2} T(\Gamma_x^2 + \Gamma_\kappa) + \frac{\alpha}{2}\Xi_T \\ + \xi_T\,(2B_x + \alpha\Gamma_x) + \frac{1}{2\alpha}\Upsilon_T + \frac{1}{\alpha} D_x \Phi_T. \end{aligned} \tag{3.14}$$

The proof of this theorem will be provided in Section 3.3.3. Theorem 2 asserts that the bound on the expected dynamic network regret $\mathbb{E}[\mathrm{Reg}_T]$ after T time steps depends on the temporal variability of the solutions (through the terms $\frac{1}{2\alpha}\Upsilon_T$ and $\frac{1}{\alpha} D_x \Phi_T$) and the errors associated with the estimation of the gradient (through the terms $\frac{\alpha}{2}\Xi_T$ and $\xi_T(2B_x + \alpha\Gamma_x)$). The remaining terms are in line with standard regret results (e.g., [89]). In particular, we can observe the following:

• *Persistent variations and gradient errors:* Suppose that Υ_T and Φ_T grow as $\mathcal{O}(T)$, modeling a persistent temporal variation of the constraints and/or the vector $\mathbf{w}_t$; suppose further that Ξ_T and ξ_T grow as $\mathcal{O}(T)$, modeling a persistent error in the gradient $\mathbf{g}_t$. Then, $\mathbb{E}[\mathrm{Reg}_T]/T$ behaves as $\mathcal{O}(1)$.

• *Vanishing gradient errors:* If Ξ_T and ξ_T grows sublinearly, i.e., as $o(T)$, then the asymptotic average regret is in general $\frac{1}{T}\mathbb{E}[\mathrm{Reg}_T] = \mathcal{O}(1 + T^{-1}\Upsilon^T + T^{-1}\Phi^T)$.

Based on numerical evidence, this is indeed the case for the shape-constrained GPs when the true function is strongly convex, as qualitatively shown in Figure 3.3. Notice that if the algorithm is executed over a finite interval of T steps and the step-size is chosen as $\alpha = \frac{1}{\sqrt{T}}$, then $\mathbb{E}[\mathrm{Reg}_T]/T$ behaves as $\frac{1}{T}\mathbb{E}[\mathrm{Reg}_T] = \mathcal{O}(T^{-\frac{1}{2}} + T^{-\frac{1}{2}}\Upsilon^T + T^{-\frac{1}{2}}\Phi^T)$.

• *Vanishing variations and errors:* If Υ_T, Φ_T, Ξ_T, and ξ_T all grow as $o(T)$, then we recover the asymptotic result of [89].

Remark 4 Regarding the errors in the gradient, if the function $u^{m,n}$ is strongly convex, then the estimation error yielded by a shape-constrained GPs will decrease with the increase of the number of functional evaluations; see the illustrative example in Figure 3.3, and the discussion in [168]. However, zeroth-order and finite-different methods will still generate an error in the estimation of the gradient, and thus Ξ_T and ξ_T would generally increase as $\mathcal{O}(T)$. To obtain a trend as $o(T)$, one has to increase the accuracy in the gradient estimation to make $\mathbf{e}^t$ arbitrarily small. □

Remark 5 Assumption *AS4* guarantees boundedness of the dual variable ν_t, at each t, as shown in [90, Sec. 3.1]. On the other hand, we note that Assumption *AS5* does not provide an analytical expression for an upper bound on the radius of Λ_t. This is one limitation of the technical findings of this work. However, we note that in our algorithm we project the dual variables $\boldsymbol{\lambda}_t$ onto Λ_t; if Λ_t is not "large enough" to contain the optimal multipliers $\boldsymbol{\lambda}_{t,*}$, then one would incur a consensus error due to clipping of the multipliers. This is in the spirit of a relaxation of the constraints with dual smoothing; see, for example, [98], where a regularization term is added to the Lagrange function to avoid the growth of the dual multiplier. In a practical implementation of the algorithm, clipping leads to an increased consensus error; in case of persistent and large consensus errors, the radius of Λ^t can be increased. A detailed analysis of the boundedness of this set is a future work direction.

3.3.2 Average Constraint Violation

We now consider a bound on the violation of the constraint on the network output $\mathbf{y}_t$. To this end, we consider the so-called average constraint violation (ACV), which we define here as

$$\mathrm{ACV}_T := \sum_{t=1}^{T} \left[C_t(\mathbf{y}_t(\mathbf{x}_t))\right]^+ . \tag{3.15}$$

We note that some prior works in context (e.g., [43, 177]) consider the modified definition

$$\left[\sum_{t=1}^{T} C^t(\mathbf{y}^t(\mathbf{x}^t))\right]^+ ,$$

which may lead to looser bounds. The expected ACV_T after T iterations can be bounded as follows.

Theorem 3 Let Assumptions *AS1-AS7* hold. Then, the average constraint violation incurred by the algorithm (3.5) can be bounded as

$$\mathbb{E}\left[\text{ACV}_T\right] \leq B_\nu^{-1} T \left(D_x L + B_\lambda \Omega B_x\right) + B_\nu^{-1} \left(\frac{\alpha}{2}\Xi^T + \xi^T(2B_x + \alpha\Gamma_x) + \frac{1}{2\alpha}\Upsilon^T + \frac{1}{\alpha} D_x \Phi^T\right) + B_\nu^{-1} T \left(\frac{1}{\alpha}(4B_x^2 + B_\nu^2) + \frac{\alpha}{2}(\Gamma_x^2 + H^2)\right). \tag{3.16}$$

The second and third terms of (3.16) exhibit the same asymptotic behavior of the dynamic regret under persistent variations and gradient errors, and with vanishing variations and errors. In case of vanishing variations and gradient errors, the bottlenecks are the two terms TD_xL and $TB_\lambda \Omega B_x$, which make $\mathbb{E}[\text{ACV}_T]/T$ behave as $\mathcal{O}(1)$.

3.3.3 Proofs of the Results

This section will provide the proofs of the previous theorems. To streamline exposition, define $\phi_t := \|\mathbf{x}_{t,*} - \mathbf{x}_{t+1,*}\|$ and $\bar{\text{e}}_t := \mathbb{E}[\|\mathbf{e}_t\| | \mathcal{F}_t]$. To derive our main results, we will utilize the following lemma.

Lemma 2 Let Assumptions *AS1-AS7* hold, and set $\boldsymbol{\lambda}_1 = \mathbf{0}$, $\nu_1 = 0$. Then, for any $\boldsymbol{\kappa} \in \Psi_t \times \Lambda_t$ the following holds:

$$\mathbb{E}\left[\sum_{t=1}^{T} \left(\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\kappa}) - \mathcal{L}_t(\mathbf{x}_{t,*}, \boldsymbol{\kappa}_t)\right)\right] \leq \frac{1}{2\alpha}(\|\mathbf{x}_1 - \mathbf{x}_{1,*}\|^2 + \|\boldsymbol{\kappa}\|^2) + \frac{\alpha}{2} T(\Gamma_x^2 + \Gamma_\kappa) + \frac{\alpha}{2}\Xi_T + \xi_T\left(2B_x + \alpha\Gamma_x\right) + \frac{1}{2\alpha}\Upsilon_T + \frac{1}{\alpha} D_x \Phi_T. \tag{3.17}$$

Proof. Notice first that

$$\|\tilde{\nabla}_{\mathbf{x}} \mathcal{L}_t(\mathbf{x}_t, \nu, \boldsymbol{\lambda}) - \nabla_{\mathbf{x}} \mathcal{L}_t(\mathbf{x}_t, \nu, \boldsymbol{\lambda})\| = \|\mathbf{e}_t\|, \tag{3.18}$$

where $\tilde{\nabla}_{\mathbf{x}} \mathcal{L}_t(\mathbf{x}_t, \nu, \boldsymbol{\lambda})$ is the inexact gradient of the Lagrangian function (where we utilize $\mathbf{g}_t$ instead of $\nabla f(\mathbf{x}_t)$). Recall that $\mathbf{x}_{t,*}$ denotes an optimal solution at time t. Using the primal update, adding

and subtracting $\mathbf{x}_{t,*}$, we get:

$$\begin{aligned}
\|\mathbf{x}_{t+1} - \mathbf{x}_{t+1,*}\|^2 &= \|\mathbf{x}_{t+1} - \mathbf{x}_{t,*} + \mathbf{x}_{t,*} - \mathbf{x}_{t+1,*}\|^2 \\
&= \|\mathbf{x}_{t+1} - \mathbf{x}_{t,*}\|^2 + 2(\mathbf{x}_{t+1} - \mathbf{x}_{t,*})^\top(\mathbf{x}_{t,*} - \mathbf{x}_{t+1,*}) + \|\mathbf{x}_{t,*} - \mathbf{x}_{t+1,*}\|^2, \\
&= \left\|\text{proj}_{\mathcal{X}_t}\left\{\mathbf{x}_t - \alpha\tilde{\nabla}_{\mathbf{x}}\mathcal{L}_t(\mathbf{x}_t, \nu_t, \boldsymbol{\lambda}_t)\right\} - \mathbf{x}_{t,*}\right\|^2 + (\phi_t)^2 + 2(\mathbf{x}_{t+1} - \mathbf{x}_{(t,*)})^\top(\mathbf{x}_{t,*} - \mathbf{x}_{t+1,*}), \\
&\leq \|\mathbf{x}_t - \alpha\tilde{\nabla}_{\mathbf{x}}\mathcal{L}_t(\mathbf{x}_t, \nu_t, \boldsymbol{\lambda}_t) - \mathbf{x}_{t,*}\|^2 + (\phi_t)^2 + 2\|\mathbf{x}_{t+1} - \mathbf{x}_{t,*}\|\|\mathbf{x}_{t,*} - \mathbf{x}_{t+1,*}\|, \\
&\leq \|\mathbf{x}_t - \mathbf{x}_{t,*} - \alpha\tilde{\nabla}_{\mathbf{x}}\mathcal{L}_t(\mathbf{x}_t, \nu_t, \boldsymbol{\lambda}_t)\|^2 + (\phi_t)^2 + 2D_x\phi_t, \qquad (3.19)
\end{aligned}$$

where (3.19) holds by the non-expansiveness property of the projection operator, the Cauchy-Schwarz inequality, and the definition of ϕ_t.

By adding and subtracting $\alpha\nabla_{\mathbf{x}}\mathcal{L}_t(\mathbf{x}_t, \nu_t, \boldsymbol{\lambda}_t)$ (to simplify notation we use $\nabla_{\mathbf{x}}\mathcal{L}_t$ to represent $\nabla_{\mathbf{x}}\mathcal{L}_t = \nabla_{\mathbf{x}}\mathcal{L}_t(\mathbf{x}_t, \nu_t, \boldsymbol{\lambda}_t)$), and expanding the first term of the right-hand side (RHS) of (3.19), we get

$$\begin{aligned}
\|\mathbf{x}_t - \mathbf{x}_{t,*} - \alpha\tilde{\nabla}_{\mathbf{x}}\mathcal{L}_t\|^2 &= \|\mathbf{x}_t - \mathbf{x}_{t,*} - \alpha\nabla_{\mathbf{x}}\mathcal{L}_t + \alpha(\nabla_{\mathbf{x}}\mathcal{L}_t - \tilde{\nabla}_{\mathbf{x}}\mathcal{L}_t)\|^2, \\
&= \|\mathbf{x}_t - \mathbf{x}_{t,*} - \alpha\nabla_{\mathbf{x}}\mathcal{L}_t\|^2 + \alpha^2\|\nabla_{\mathbf{x}}\mathcal{L}_t - \tilde{\nabla}_{\mathbf{x}}\mathcal{L}_t\|^2 + 2\alpha(\mathbf{x}_t - \mathbf{x}_{t,*} - \alpha\nabla_{\mathbf{x}}\mathcal{L}_t)^\top(\nabla_{\mathbf{x}}\mathcal{L}_t - \tilde{\nabla}_{\mathbf{x}}\mathcal{L}_t), \\
&\leq \|\mathbf{x}_t - \mathbf{x}_{t,*} - \alpha\nabla_{\mathbf{x}}\mathcal{L}_t\|^2 + \alpha^2\|\nabla_{\mathbf{x}}\mathcal{L}_t - \tilde{\nabla}_{\mathbf{x}}\mathcal{L}_t\|^2 + 2\alpha\|\mathbf{x}_t - \mathbf{x}^{t*} - \alpha\nabla_{\mathbf{x}}\mathcal{L}^t\|\|\nabla_{\mathbf{x}}\mathcal{L}^t - \tilde{\nabla}_{\mathbf{x}}\mathcal{L}^t\|.
\end{aligned} \qquad (3.20)$$

By using (3.18), we have that (3.20) can be bounded as

$$\begin{aligned}
\|\mathbf{x}_t - \mathbf{x}_{t,*} - \alpha\tilde{\nabla}_{\mathbf{x}}\mathcal{L}_t\|^2 &\leq \|\mathbf{x}_t - \mathbf{x}_{t,*} - \alpha\nabla_{\mathbf{x}}\mathcal{L}_t\|^2 + \alpha^2\|\mathbf{e}_t\|^2 + 2\alpha\|\mathbf{e}_t\|\,\|\mathbf{x}_t - \mathbf{x}_{t,*} - \alpha\nabla_{\mathbf{x}}\mathcal{L}_t\|, \\
&\leq \|\mathbf{x}_t - \mathbf{x}_{t,*} - \alpha\nabla_{\mathbf{x}}\mathcal{L}_t\|^2 + \alpha^2\|\mathbf{e}_t\|^2 + 2\alpha\|\mathbf{e}_t\|\,(\|\mathbf{x}_t - \mathbf{x}_{t,*}\| + \alpha\|\nabla_{\mathbf{x}}\mathcal{L}^t\|), \\
&\leq \|\mathbf{x}_t - \mathbf{x}_{t,*} - \alpha\nabla_{\mathbf{x}}\mathcal{L}_t\|^2 + \alpha^2\|\mathbf{e}_t\|^2 + 2\alpha\|\mathbf{e}_t\|\,(2B_x + \alpha\Gamma_x), \qquad (3.21)
\end{aligned}$$

where the last inequality is derived using (3.12) and the triangle inequality. By expanding the first term on the RHS of (3.21), we get

$$\begin{aligned}
\|\mathbf{x}_t - \mathbf{x}_{t,*} - \alpha\nabla_{\mathbf{x}}\mathcal{L}^t\|^2 &= \|\mathbf{x}_t - \mathbf{x}_{t,*}\|^2 + \alpha^2\|\nabla_{\mathbf{x}}\mathcal{L}^t\|^2 - 2\alpha(\mathbf{x}_t - \mathbf{x}_{t,*})^\top\nabla_{\mathbf{x}}\mathcal{L}_t, \\
&\leq \|\mathbf{x}_t - \mathbf{x}_{t,*}\|^2 + \alpha^2\Gamma_x^2 - 2\alpha[\mathcal{L}_t(\mathbf{x}_t, \nu_t, \boldsymbol{\lambda}_t) - \mathcal{L}_t(\mathbf{x}_{t,*}, \nu_t, \boldsymbol{\lambda}_t)]. \qquad (3.22)
\end{aligned}$$

Since the function $f(\mathbf{x})$ is convex, the time-varying Lagrangian is convex in $\mathbf{x}$. Thus, from the first-order characterization of convexity (3.22) holds.

Using (3.22) the bound (3.21), one has that

$$\|\mathbf{x}_t - \mathbf{x}_{t,*} - \alpha\tilde{\nabla}_{\mathbf{x}}\mathcal{L}_t\|^2 \leq \|\mathbf{x}_t - \mathbf{x}_{t,*}\|^2 + \alpha^2\Gamma_x^2 + \alpha^2\|\mathbf{e}_t\|^2 + 2\alpha\|\mathbf{e}_t\|\,(2B_x + \alpha\Gamma_x) \\ - 2\alpha[\mathcal{L}_t(\mathbf{x}_t, \nu_t, \boldsymbol{\lambda}_t) - \mathcal{L}_t(\mathbf{x}_{t,*}, \nu_t, \boldsymbol{\lambda}_t)]. \tag{3.23}$$

By using (3.23) in (3.19), we have

$$\|\mathbf{x}_{t+1} - \mathbf{x}_{t+1*}\|^2 \leq \|\mathbf{x}_t - \mathbf{x}_{t,*}\|^2 + \alpha^2\Gamma_x^2 + \alpha^2\|\mathbf{e}_t\|^2 + 2\alpha\|\mathbf{e}_t\|\,(2B_x + \alpha\Gamma_x) \\ - 2\alpha[\mathcal{L}_t(\mathbf{x}_t, \nu_t, \boldsymbol{\lambda}_t) - \mathcal{L}_t(\mathbf{x}_{t,*}, \nu_t, \boldsymbol{\lambda}_t)] + (\phi_t)^2 + 2D_x\phi_t. \tag{3.24}$$

It thus follows that:

$$\mathcal{L}_t(\mathbf{x}_t, \nu_t, \boldsymbol{\lambda}_t) - \mathcal{L}_t(\mathbf{x}_{t,*}, \nu_t, \boldsymbol{\lambda}_t) \leq \frac{1}{2\alpha}(\|\mathbf{x}_t - \mathbf{x}_{t,*}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}_{t+1,*}\|^2) + \frac{\alpha}{2}\Gamma_x^2 \\ + \frac{\alpha}{2}\|\mathbf{e}_t\|^2 + \|\mathbf{e}_t\|\,(2B_x + \alpha\Gamma_x) + \frac{1}{2\alpha}(\phi_t)^2 + \frac{1}{\alpha}D_x\phi_t. \tag{3.25}$$

Now, we proceed with a similar analysis for the distance between the updates of the dual variables $\boldsymbol{\lambda}_{t+1}, \nu_{t+1}$ and an arbitrary dual variable, i.e., for any $\boldsymbol{\lambda} \in \Lambda_t$ and any $\nu \in \Psi_t$. We start with the following:

$$\begin{aligned}\|\boldsymbol{\kappa}_{t+1} - \boldsymbol{\kappa}\|^2 &\leq \|\boldsymbol{\kappa}_t + \alpha\nabla_\kappa\mathcal{L}_t - \boldsymbol{\kappa}\|^2, \\ &\leq \|\boldsymbol{\kappa}_t - \boldsymbol{\kappa}\|^2 + \alpha^2\|\nabla_\kappa\mathcal{L}_t\|^2 + 2\alpha\nabla_\kappa\mathcal{L}_t^\top(\boldsymbol{\kappa}_t - \boldsymbol{\kappa}), \\ &\leq \|\boldsymbol{\kappa}_t - \boldsymbol{\kappa}\|^2 + \alpha^2\Gamma_\kappa + 2\alpha\nabla_\kappa\mathcal{L}_t^\top(\boldsymbol{\kappa}_t - \boldsymbol{\kappa})\end{aligned} \tag{3.26}$$

where $\nabla_\kappa\mathcal{L}_t = \nabla_\kappa\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\kappa}_t)$ for brevity.

The time-varying Lagrangian is concave in $\boldsymbol{\lambda}$ and ν; i.e., any $\boldsymbol{\kappa}$, which implies that the time-varying Lagrangian difference for a fixed $\mathbf{x}_t \in \mathcal{X}_t$ satisfies

$$\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\kappa}_t) - \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\kappa}) \geq (\boldsymbol{\kappa}_t - \boldsymbol{\kappa})^\top\nabla_\kappa\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\kappa}). \tag{3.27}$$

Substituting (3.27) in (3.26) for any $\boldsymbol{\kappa} \in \Psi_t \times \Lambda_t$, we get

$$\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\kappa}_t) - \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\kappa}) \geq \frac{1}{2\alpha}(\|\boldsymbol{\kappa}_{t+1} - \boldsymbol{\kappa}\|^2 - \|\boldsymbol{\kappa}_t - \boldsymbol{\kappa}\|^2) - \frac{\alpha}{2}\Gamma_\kappa. \tag{3.28}$$

By subtracting the results in (3.25) and (3.28) we have

$$\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\kappa}) - \mathcal{L}_t(\mathbf{x}_{t,*}, \boldsymbol{\kappa}_t) \leq \frac{1}{2\alpha}(\|\mathbf{x}_t - \mathbf{x}_{t,*}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}_{t+1,*}\|^2) + \frac{1}{2\alpha}(\|\boldsymbol{\kappa}^t - \boldsymbol{\kappa}\|^2 - \|\boldsymbol{\kappa}^{t+1} - \boldsymbol{\kappa}\|^2) + \|\mathbf{e}^t\|\,(2B_x + \alpha\Gamma_x) + \frac{\alpha}{2}\|\mathbf{e}^t\|^2 + \frac{1}{2\alpha}(\phi^t)^2 + \frac{1}{\alpha}D_x\phi^t + \frac{\alpha}{2}(\Gamma_x^2 + \Gamma_\kappa). \tag{3.29}$$

Assume that $\boldsymbol{\kappa}_1 = \mathbf{0}$. Then, summing (3.29) over time, using the telescopic property for the first two terms on the RHS, and by taking the expectation of both sides, the result follows. Note that $\mathbb{E}\left[\sum_{t=1}^{T}\|\mathbf{e}_t\|\right] = \sum_{t=1}^{T}\mathbb{E}\left[\|\mathbf{e}_t\|\right]$ by the independence of $\mathbf{e}_t$ for all t. ■

Based on Lemma 2, in the following, we provide the proof for Theorem 2.

3.3.3.1 Proof for Theorem 2

First, we can rewrite $\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\kappa}) - \mathcal{L}_t(\mathbf{x}_{t,*}, \boldsymbol{\kappa}_t)$ as:

$$\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\kappa}) - \mathcal{L}_t(\mathbf{x}_{t,*}, \boldsymbol{\kappa}_t) = f(\mathbf{x}_t) - f(\mathbf{x}_{t,*}) + \underbrace{\boldsymbol{\kappa}^\top \begin{bmatrix} C_t\left(\mathbf{y}_t(\mathbf{x}_t)\right) \\ \mathbf{D}\mathbf{x}_t \end{bmatrix} - \boldsymbol{\kappa}_t{}^\top \begin{bmatrix} C_t\left(\mathbf{y}_t(\mathbf{x}_{t,*})\right) \\ \mathbf{D}\mathbf{x}_{t,*} \end{bmatrix}}_{:=\beta_t} \tag{3.30}$$

where we recall that $\mathbf{D}\mathbf{x}_{t,*} = \mathbf{0}$ and $C_t\left(\mathbf{y}_t(\mathbf{x}_{t,*})\right) \leq 0$.

By using the definition of the cost function $f(\mathbf{x})$ and summing (3.30) over time, we have

$$\sum_{t=1}^{T}\left(\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\kappa}) - \mathcal{L}_t(\mathbf{x}_{t,*}, \boldsymbol{\kappa}_t)\right) = \sum_{t=1}^{T}\beta_t + \sum_{t=1}^{T}\left(\sum_{m=1}^{M}\sum_{n=1}^{N_m}u^{m,n}(x_t^{m,n}) - \sum_{m=1}^{M}\sum_{n=1}^{N_m}u^{m,n}(x_{t,*}^{m})\right). \tag{3.31}$$

Taking the expected value and using Lemma 2 in (3.31), we have

$$\mathbb{E}\left[\sum_{t=1}^{T}\left(\sum_{m=1}^{M}\sum_{n=1}^{N_m}u^{m,n}(x_t^{m,n}) - \sum_{m=1}^{M}\sum_{n=1}^{N_m}u^{m,n}(x_{t,*}^{m})\right) + \sum_{t=1}^{T}\beta_t\right] \leq \frac{1}{2\alpha}(\|\mathbf{x}_1 - \mathbf{x}_{1,*}\|^2 + \|\boldsymbol{\kappa}\|^2) + \frac{\alpha}{2}T(\Gamma_x^2 + \Gamma_\kappa) + \frac{\alpha}{2}\Xi_T + \xi_T\,(2B_x + \alpha\Gamma_x) + \frac{1}{2\alpha}\Upsilon_T + \frac{1}{\alpha}D_x\Phi_T. \tag{3.32}$$

In order to bound the global network regret, we add and subtract the RHS of (3.31) in (3.9),

to obtain

$$\begin{aligned}\mathrm{Reg}_T = &\sum_{t=1}^{T}\left[\sum_{m=1}^{M}\left(\frac{1}{N_m}\sum_{i,j=1}^{N_m}u^{m,i}(x_t^{m,j}) - \sum_{n=1}^{N_m}u^{m,n}(x_t^{m,n})\right) - \beta_t\right] \\ &+ \sum_{t=1}^{T}\left[\left(\sum_{m=1}^{M}\sum_{n=1}^{N_m}u^{m,n}(x_t^{m,n}) - \sum_{m=1}^{M}\sum_{n=1}^{N_m}u^{m,n}(x_{t,*}^{m})\right) + \beta_t\right]. \end{aligned} \tag{3.33}$$

The expected value of the last term of (3.33) is bounded by (3.32). Next, focus on the first term in (3.33). Assumption *AS2* implies that

$$\sum_{i,j=1}^{N_m}u^{m,i}(x_t^{m,j}) - \sum_{n=1}^{N_m}u^{m,n}(x_t^{m,n}) \leq \sum_{i,j=1}^{N_m}L_{m,i}|x_t^{m,j} - x_t^{m,i}|, \tag{3.34}$$

where $L_{m,i} \leq L_m \leq L$ per each network composed of a system and the users interacting with such a system. Maximizing over the RHS of (3.34) we can get an expression for the worst-case disagreement per system m as

$$\sum_{i,j=1}^{N_m}L_{m,i}|x_t^{m,j} - x_t^{m,i}| \leq N_m^2 L_m h_m \|\mathbf{D}_m \mathbf{x}_t^m\|, \tag{3.35}$$

where:

$$\max_{i,j}|x_t^{m,j} - x_t^{m,i}|$$

can by upper bounded using Assumption *AS3*, and $\mathbf{x}^m := [x^m, x^{m,1}, \ldots, x^{m,N_m}] \,\forall\, m$.

Back to the first term in (3.33), we have that

$$\begin{aligned}&\sum_{t=1}^{T}\left[\sum_{m=1}^{M}\left(\frac{1}{N_m}\sum_{i,j=1}^{N_m}u^{m,i}(x_t^{m,j}) - \sum_{n=1}^{N_m}u^{m,n}(x_t^{m,n})\right) - \beta_t\right] \\ &\quad= \sum_{t=1}^{T}\left[\sum_{m=1}^{M}\frac{1}{N_m}\sum_{i,j=1}^{N_m}\left(u^{m,i}(x_t^{m,j}) - U^{m,n}(x_t^{m,n})\right) - \beta_t\right] \\ &\quad\leq \sum_{t=1}^{T}\sum_{m=1}^{M}\frac{1}{N_m}(N_m^2 L_m h_m \|\mathbf{D}_m \mathbf{x}_t^m\|) - \sum_{t=1}^{T}\beta_t \\ &\quad\leq \sum_{t=1}^{T} N_{\max} h_{\max} L\, M \|\mathbf{D}\mathbf{x}_t\| - \sum_{t=1}^{T}\beta_t\,. \end{aligned} \tag{3.36}$$

By the fact that at optimality $\boldsymbol{\lambda}_t^\top \mathbf{D}\mathbf{x}_{t,*} = 0$ and $\nu_t C_t(\mathbf{y}_t(\mathbf{x}_{t,*})) \leq 0$, we can rewrite the RHS of

(3.36) as

$$\sum_{t=1}^{T} N_{\max} h_{\max} LM \|\mathbf{D}\mathbf{x}_t\| - \sum_{t=1}^{T} \boldsymbol{\kappa}^\top \begin{bmatrix} C_t\left(\mathbf{y}_t(\mathbf{x}_t)\right) \\ \mathbf{D}\mathbf{x}_t \end{bmatrix}$$
$$= \sum_{t=1}^{T} \left(N_{\max} h_{\max} LM \frac{\mathbf{D}\mathbf{x}_t}{\|\mathbf{D}\mathbf{x}_t\|} - \boldsymbol{\lambda} \right)^\top \mathbf{D}\mathbf{x}_t - \sum_{t=1}^{T} \nu\, C_t\left(\mathbf{y}_t(\mathbf{x}_t)\right). \quad (3.37)$$

By construction, we can choose a feasible dual variable $\bar{\boldsymbol{\lambda}}$ using the compactness of $\mathcal{X}_t$ and the boundedness of $\|\mathbf{D}\mathbf{x}_t\|$, as in [89], so that the first term in (3.37) equals 0. Similarly, we can set $\bar{\nu} = 0$. With this choice for $\bar{\boldsymbol{\kappa}} = [\bar{\nu}, \bar{\boldsymbol{\lambda}}^\top]^\top$, the expression on (3.37) is equal to 0. Then, the expected value of the regret defined in (3.9) is bounded as shown in (3.14). ■

3.3.3.2 Proof for Theorem 3

We start by bounding the distance between the update of the dual variable ν_{t+1} and an arbitrary dual variable, i.e., for any $\nu \in \Psi_t$

$$|\nu_{t+1} - \nu|^2 \leq |\nu_t + \alpha \nabla_\nu \mathcal{L}_t - \nu|^2,$$
$$\leq |\nu_t - \nu|^2 + \alpha^2 H^2 + 2\alpha \nabla_\nu \mathcal{L}_t (\nu_t - \nu), \quad (3.38)$$

where $\nabla_\nu \mathcal{L}_t = \nabla_\nu \mathcal{L}_t(\mathbf{x}_t, \nu, \boldsymbol{\lambda}) = C_t(\mathbf{y}_t(\mathbf{x}_t))$ for any feasible $\boldsymbol{\lambda}$. Reorganizing (3.38) for any $\nu \in \Psi_t$, we get

$$(\nu_t - \nu)\nabla_\nu \mathcal{L}_t \geq \frac{1}{2\alpha}(|\nu_{t+1} - \nu|^2 - |\nu_t - \nu|^2) - \frac{\alpha}{2} H^2. \quad (3.39)$$

The time-varying Lagrangian differences for a fixed $\mathbf{x}_t \in \mathcal{X}_t$ and $\boldsymbol{\lambda}_t \in \Lambda_t$ satisfies

$$\mathcal{L}_t(\mathbf{x}_t, \nu_t, \boldsymbol{\lambda}_t) - \mathcal{L}_t(\mathbf{x}_t, \nu, \boldsymbol{\lambda}_t) = (\nu_t - \nu)\nabla_\nu \mathcal{L}_t(\mathbf{x}_t, \nu, \boldsymbol{\lambda}). \quad (3.40)$$

Replacing (3.40) in (3.39), and subtracting the results in (3.25), for any $\nu \in \Psi_t$, we get

$$\mathcal{L}_t(\mathbf{x}_t, \nu, \boldsymbol{\lambda}_t) - \mathcal{L}_t(\mathbf{x}_{t,*}, \nu_t, \boldsymbol{\lambda}_t) \leq \frac{1}{2\alpha}(\|\mathbf{x}_t - \mathbf{x}_{t,*}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}_{t+1,*}\|^2)$$
$$+ \frac{1}{2\alpha}(|\nu_t - \nu|^2 - |\nu_{t+1} - \nu|^2) + \|\mathbf{e}_t\| \left(2B_x + \alpha \Gamma_x\right)$$
$$+ \frac{\alpha}{2}\|\mathbf{e}_t\|^2 + \frac{1}{2\alpha}(\phi_t)^2 + \frac{1}{\alpha} D_x \phi_t + \frac{\alpha}{2}(\Gamma_x^2 + H^2). \quad (3.41)$$

Since $\boldsymbol{\lambda}_t^\top \mathbf{D}\mathbf{x}_{t,*} = 0$, we can also express $\mathcal{L}_t(\mathbf{x}_t, \nu, \boldsymbol{\lambda}_t) - \mathcal{L}_t(\mathbf{x}_{t,*}, \nu_t, \boldsymbol{\lambda}_t)$ as

$$\mathcal{L}_t(\mathbf{x}_t, \nu, \boldsymbol{\lambda}_t) - \mathcal{L}_t(\mathbf{x}_{t,*}, \nu_t, \boldsymbol{\lambda}_t) = (f(\mathbf{x}_t) - f(\mathbf{x}_{t,*})) + \nu\, C_t(\mathbf{y}_t(\mathbf{x}_t)) + \boldsymbol{\lambda}_t^\top \mathbf{D}\mathbf{x}_t - \nu_t C_t(\mathbf{y}_t(\mathbf{x}_{t,*})). \quad (3.42)$$

Reorganizing terms in (3.42), and by the fact that $\nu_t C_t(\mathbf{y}_t(\mathbf{x}_{t,*})) \leq 0$, we have

$$\nu\, C_t(\mathbf{y}_t(\mathbf{x}_t)) \leq (\mathcal{L}_t(\mathbf{x}_t, \nu, \boldsymbol{\lambda}_t) - \mathcal{L}_t(\mathbf{x}_{t,*}, \nu_t, \boldsymbol{\lambda}_t)) - (f(\mathbf{x}_t) - f(\mathbf{x}_{t,*})) - \boldsymbol{\lambda}_t^\top \mathbf{D}\mathbf{x}_t. \quad (3.43)$$

On the other hand, since the cost function is convex, we have

$$f(\mathbf{x}_t) - f(\mathbf{x}_{t,*}) \leq \|\mathbf{x}_t - \mathbf{x}_{t,*}\| \|\nabla f(\mathbf{x}^{t,*})\| \leq D_x L, \quad (3.44)$$

and because $\mathcal{X}_t$ and Λ_t are compact sets uniformly in time, we further get

$$|\boldsymbol{\lambda}^\top \mathbf{D}\mathbf{x}_t| \leq \|\boldsymbol{\lambda}_t\| \|\mathbf{D}\| \|\mathbf{x}_t\| \leq B_\lambda \Omega B_x. \quad (3.45)$$

By using (3.41), (3.44) and (3.45) in (3.43), and reorganizing terms,

$$\begin{aligned} C_t(\mathbf{y}_t(\mathbf{x}_t)) \leq \nu^{-1} \Bigg(& \frac{1}{\alpha}(4B_x^2 + B_\nu^2) + D_x L + B_\lambda \Omega B_x \\ & + \frac{\alpha}{2}(\Gamma_x^2 + H^2) + \|\mathbf{e}^t\| \left(2B_x + \alpha \Gamma_x\right) + \frac{\alpha}{2}\|\mathbf{e}^t\|^2 + \frac{1}{2\alpha}(\phi^t)^2 + \frac{1}{\alpha} D_x \phi^t \Bigg). \end{aligned} \quad (3.46)$$

We now take the max operator, sum over time, and take the expectation of both sides of (3.46), then

$$\begin{aligned} \mathbb{E}\left[\sum_{t=1}^{T} [C_t(\mathbf{y}_t(\mathbf{x}_t))]^+\right] \leq & T\nu^{-1}\left(\frac{1}{\alpha}(4B_x^2 + B_\nu^2) + D_x L + B_\lambda \Omega B_x + \frac{\alpha}{2}(\Gamma_x^2 + H^2)\right) \\ & + \nu^{-1}\left(\xi^T(2B_x + \alpha\Gamma_x)\right) + \nu^{-1}\left(\frac{\alpha}{2}\Xi^T + \frac{1}{2\alpha}\Upsilon^T + \frac{1}{\alpha} D_x \Phi^T\right) \forall\, \nu \in (0, B_\nu]. \end{aligned}$$

The previous inequality implies that the tightest result happens at $\nu = B_\nu$, then, (3.16) holds. ■

Finally, we present an additional result for the dynamic fit. Let $\text{Fit}_T := \left[\sum_{t=1}^{T} C_t(\mathbf{y}_t(\mathbf{x}_t))\right]^+$. By using (3.41), (3.44) and (3.45) in (3.43), summing over time, and using the telescopic property for the first two terms on the RHS, we get that for any $\nu \in \Psi_t$

$$\begin{aligned} \nu \sum_{t=1}^{T} C_t(\mathbf{y}_t(\mathbf{x}_t)) \leq & \frac{1}{2\alpha}(\|\mathbf{x}_1 - \mathbf{x}_{1,*}\|^2 + |\nu|^2) + \frac{\alpha}{2} T(\Gamma_x^2 + H^2) \\ & + T D_x L + T B_\lambda \Omega B_x + (2B_x + \alpha\Gamma_x) \sum_{t=1}^{T} \|\mathbf{e}_t\| + \frac{\alpha}{2} \sum_{t=1}^{T} \|\mathbf{e}_t\|^2 + \frac{1}{2\alpha}\Upsilon_T + \frac{1}{\alpha} D_x \Phi_T. \end{aligned} \quad (3.47)$$

By taking the max operator and the expectation of both sides on (3.47), we obtain for any $\nu \in (0, B_\nu]$

$$\begin{aligned}\mathbb{E}\left[\left[\sum_{t=1}^{T} C^t(\mathbf{y}^t(\mathbf{x}^t))\right]^+\right] &\leq \nu^{-1}(TD_xL + TB_\lambda\Omega B_x) \\ &+ \nu^{-1}\left(\frac{1}{2\alpha}(4B_x^2 + B_\nu^2) + \frac{\alpha}{2}T(\Gamma_x^2 + H^2)\right) \\ &+ \nu^{-1}\left(\frac{\alpha}{2}\Xi^T + \xi^T(2B_x + \alpha\Gamma_x) + \frac{1}{2\alpha}\Upsilon^T + \frac{1}{\alpha}D_x\Phi^T\right).\end{aligned}$$

Similar as the ACV, set $\nu = B_\nu$ in order to get the tightest bound for the expected dynamic fit, i.e., $\mathbb{E}[\text{Fit}_T]$. ■

3.4 Conclusions

In this chapter, we presented an online consensus-based algorithm to solve a time-varying optimization problem associated with a network of systems shared by multiple users. The cost functions that represent the individuals' preferences were learned concurrently with the execution of the algorithm via shape-constrained GPs using users' feedback. We developed an online algorithm based on a primal-dual method, properly modified to accommodate feedback from the users and measurements from the network. We analyzed the performance via dynamic network regret and average constraint violation, and we showed that the bounds of these metrics depend on the temporal variability of the optimal solution set, and the errors associated with the estimation of the gradients. Numerical results were presented in the context of real-time management of distributed energy resources.

The setting presented in this chapter assumed that the function $\mathbf{x} \mapsto U(\mathbf{x})$ is static but unknown, studying of time-varying functions is a further extension of this work. In this area, our main idea is to leverage the tools presented in [25], where two extensions of the classical GP upper confidence bound (GP-UCB) algorithm were proposed to concurrently learn and maximize a time-varying function. Although, this chapter do not provide an analysis for the case where the time-varying GP-UCB of [25] is utilized in lieu of (2.5a) in our framework, we would like to provide some preliminary numerical simulations to illustrate the idea.

The objective in [25] is to sequentially find a sequence of points to optimize an unknown time-varying function from noisy functional evaluations. In order to handle the time variations, in [25] the function is modeled as a GP that varies according to a Markov process as follows:

$$g_1(x) = v_1(x)$$
$$g_{t+1}(x) = \sqrt{1-\epsilon}\, g_t(x) + \sqrt{\epsilon}\, v_{t+1}(x) \quad \forall t \geq 2,$$

where g_t is the unknown function defined over a convex and compact set $\mathcal{X}$, v_i are independent random functions on $\mathcal{X}$, with $v_i \sim \mathcal{GP}(0, k)$, and $\epsilon \in (0, 1)$. Figure 3.4 presents an illustration of this function model.

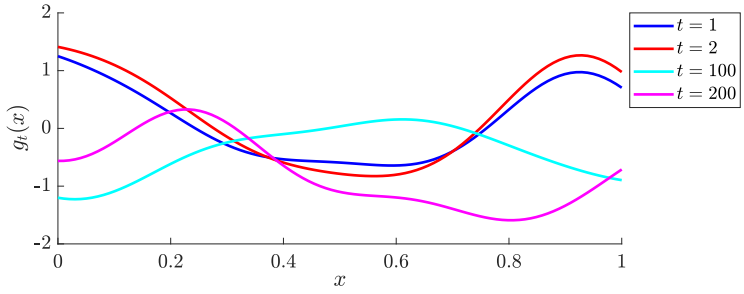


Figure 3.4: Example of the reward function $g_t(x)$ when $\epsilon = 0.03$ using a square exponential kernel ($\ell = 0.2$, $\sigma_f = 1$).

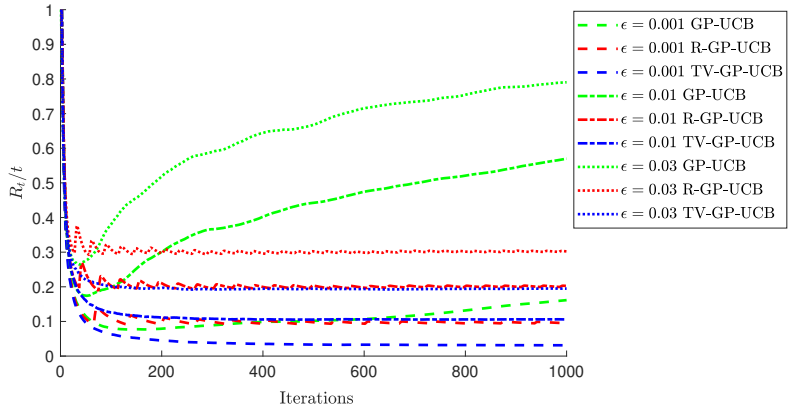


Figure 3.5: Regret over time for different values of ϵ using the square exponential kernel.

In [25], two time-varying GP-UCB are presented: R-GP-UCB and TV-GP-UCB. The first algorithm is R-GP-UCB where the traditional GP-UCB algorithm is executed within block of a fixed

size, and a resetting strategy is applied at the start of each block (see [25, Algorithm 1]). The second algorithm, TV-GP-UCB, instead forgets smoothly the old data points by using a posterior update rule obtained through the time-varying model of g_i. The previous algorithm can be considered as a special case of GP-UCB with a spatio-temporal kernel. The difference between these two time-varying methods and the traditional GP-UCB is that the later treats all of the previous samples as being equally important, while on the contrary in TV-GP-UCB and R-GP-UCB the samples become nonrelevant or not enough informative with time (see [25, Algorithm 2]). Figure 3.5 shows the results from some numerical simulations we performed, and reports the trajectories of the average regret $\frac{R_t}{t}$ of GP-UCB, TV-GP-UCB, and R-GP-UCB for $\epsilon \in \{0.001, 0.01, 0.03\}$. In this simulation, we average the performance over 100 independent trials. We observe that the GP-UCB, is not an appropriate algorithm for a time-varying setting. On the other hand, TV-GP-UCB and R-GP-UCB can handle the time-varying cost where the TV-GP-UCB outperform R-GP-UCB. As expected, the curves for TV-GP-UCB are smoother than the R-GP-UCB where it tends to incur more regret just after a reset is performed.

Chapter 4

Real-Time Management of Distribute Energy Resources

In this chapter, we evaluate the performance of the three algorithms proposed in Chapter 2 and 3 numerically in the context of real-time management of distributed energy resources (DERs).

Net-load and demand response (DR) strategies hold promise to increase the flexibility and efficiency of power systems by allowing controllable devices to provided services at various time-scales – from real-time frequency and voltage support to a slower time-scale peak-shifting service [96, 124, 95, 11]. Typical DR formulations involve a composite cost function to strike a balance between system-level operational objectives and (dis)satisfaction of the device's owner [67, 97]; e.g., deviations from a preferred indoor temperature or charging profile of the electric vehicle. Then, this aspect makes the actual implementation of DR programs challenging: users' preferences, satisfaction and responsiveness to pricing[173] are not easy to model; synthetic cost functions adopted in existing demand response and net-load management frameworks favor computational tractability, but may not capture the users' goals truthfully.

Examples of related works on real-time DR include the online convex optimization strategy applied to DR problems in [95]; however, the function associated with heating, ventilation, and air-conditioning (HVAC) systems of commercial buildings is known, and no measurements are utilized in the algorithm. An online learning approach for computing users' optimal scheduling policy were investigated in [11], for a given householder's cost function. Also, an online learning approach was considered in [167], based on a multi-armed restless bandit problem with controlled bandits. Price responsiveness of the end users that participate in DR programs was studied in, e.g., [173]

by using a dynamical model that captures the temporal behavior of the users. Community-level energy management systems that control consumers were investigated in [144]. For completeness, we point out that users' perception was incorporated in the decision making process with GPs in other application, e.g., [10, 102].

Here, we target a real-time management of distributed energy resources (DERs) at the power distribution level. The experimental setup can be summarized as follows. For an aggregations of devices in a neighborhood or community, $\mathbf{y}_t$ represents the total active power at the point of interconnection of the rest of the grid. If $\mathbf{y}_t \in \mathbb{R}$, then $\mathbf{A}$ boils down to a row-vector with all ones and $x_{m,t}$ represents the active power set-points of the devices. In the spirit of a "virtual power plant", $y_{\text{ref},t}$ can be a time-varying reference signal for the active power at the point of interconnection (to provide, for example, primal or secondary grid services).

In this chapter, first we present the performance of the proposed shape-constrained GP-based online projected gradient descent (SGP-OPGD) in Section 4.1. Second, the numerical results for the case of the inexact online projected gradient descent with infrequent measurements is shown in Section 4.2. Finally, in Section 4.3, the results for the consensus-strategy for the inexact online data-driven primal-dual approach are presented.

4.1 Gradient-based Approach with learning

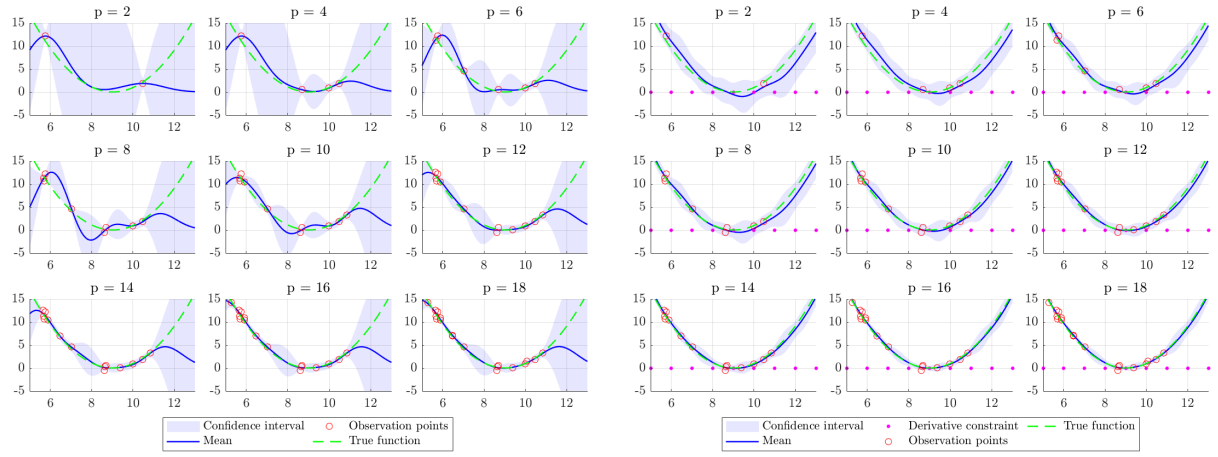
The objective in this section is to formulate a demand-side management problem [67, 97] that allows real-time scheduling of end-user devices by minimizing a cost that accounts for both network performance metrics and user satisfaction. Accordingly, let $u_m : \mathcal{X}_m \mapsto \mathbb{R}$ be a "discomfort function" for the the mth user or device. For example, for a thermostatically controllable load, this function may model the discomfort of the user for deviations from a preferred setpoint; for an electric vehicle, u_m may model the dissatisfaction of the user for deviations relative to a preferred charging profile. The function u_m is assumed to be time-invariant for simplicity; however, the proposed approach can be extended to cases where some of the functions $u_{m,t}$ are time-varying functions to model a dynamic user behavior, as explained in Section 3.4. Many exiting works

presume that the function u_m is *known* and it is convex; as explained shortly, here u_m will be *learned* from data.

We consider a neighborhood-level problem as in *Example 2* in Chapter 2 Section 2.1.1. In this example, we control 15 batteries, 10 HVAC units (equipped with variable speed drives), and 5 electric vehicles (EVs). The objective is to maintain the aggregate active power $\sum_m x_{m,t}$ close to a reference point $y_{\text{ref},t}$ while minimizing the discomfort/dissatisfaction for each user. The operational sets for the devices are: *(i)* batteries constraints $\mathcal{X}_m = [-8, 8]$ kW $\forall\, m = 1, \ldots, 15$; *(ii)* HVAC constraints $\mathcal{X}_m = [5, 15]$ kW $\forall\, m = 16, \ldots, 25$; and *(iii)* EV constraints $\mathcal{X}_m = [7, 50]$ kW $\forall\, m = 26, \ldots, 30$. To concretely assess the performance of the shape-constrained GP, the discomfort functions $\{u_m\}_{m=1}^M$ are assumed to be quadratic; the minimum of each of the functions is inside the set constraints $\mathcal{X}_m$, and it corresponds to a preferred setting of the user. For example, for EVs they represent a preferred charging rate; for HVAC systems, they represent a preferred temperature setpoint (converted into a preferred power setpoint) [95]. The function $C_t(\mathbf{x})$ is $C_t(\mathbf{x}_t) = \frac{\beta}{2}(\sum_m x_{m,t} + \mathbf{1}^\top \mathbf{w}_t - y_{\text{ref},t})^2$, where the non-controllable loads are taken from the Anatolia dataset (National Renewable Energy Laboratory, Tech. Rep. NREL/TP-5500-56610) and have a granularity of 1 second.

As an example of estimation of the discomfort functions using the shape-constrained GP, Figure 4.1 illustrates the estimated function for a device for a different number of observations p; in particular, the estimated functions using a standard GP regression and the shape-constrained GP are illustrated.

We run the online algorithm for a period of 12 hours staring at 12:00 am; each step of Algorithm 1 is performed every 5 seconds (expect for HVAC, which are updated at a slower rate). A prior $\{\hat{u}_{m,p_m}\}_{m=1}^M$ is determined from some noisy measurements ($\sigma = 5$) and $\hat{u}_{m,p_m}(x_{m,t})$ is updated through user's feedback every 30 min. The results in Figure 4.2 for the the SGP-OPGD algorithm are compared with two trajectories: *(i)* trajectory for the optimal solution $\mathbf{x}_t^*$ for a known synthetic discomfort functions $\{u_m\}_{m=1}^M$, where the problem is solved to convergence; *(ii)* trajectory for the learned optimal solution $\hat{\mathbf{x}}_t^*$ when $\{\hat{u}_{m,p_m}\}_{m=1}^M$ is estimated as in (2.7), where also



(a) (b)

Figure 4.1: Example of the estimation of the discomfort function $\hat{U}_{m,p_m}$ with hyperparameters $\sigma_f = 1$ and $l = 10$. (a) Standard GP regression and (b) shape-constrained GP regression. The observations points are a mixed of prior points and a sub-sequence produced by the online algorithm. One can see that the function estimated with shape-constrained GPs is "practically" smooth and strongly convex, as desired, after only a few feedback points p_m.

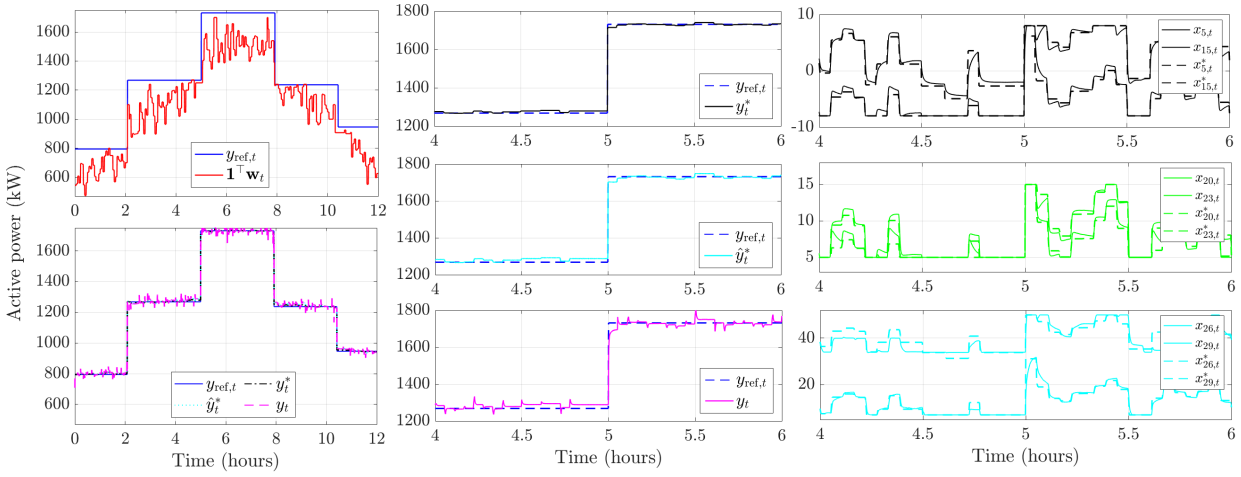


Figure 4.2: Solution of the SGP-OPGD algorithm. *Left top*: references setpoints $y_{\text{ref},t}$ and aggregate non-controllable loads $\mathbf{1}^\top \mathbf{w}_t$. *Left bottom*: optimal trajectories y_t^* for a known synthetic functions $\{u_m\}_{m=1}^M$, learned trajectory $\hat{y}_t^*$ where $\hat{u}_{m,p_m}$ is used and the problem is solved to convergence at each time, and trajectory y_t for the SGP-OPGD algorithm. *Center*: Zoomed view for the trajectories y_t^*, $\hat{y}_t^*$ and y_t on the time period 4:00 pm - 6:00 pm. *Right*: example of active power setpoints for the SGP-OPGD method of 6 representative devices on the time period 4:00 pm - 6:00 pm.

the problem is solved to convergence. In this case, the estimate of the gradients for $\{\hat{u}_{m,p_m}\}_{m=1}^M$

are calculated using a finite difference method; 21 noisy observations ($\sigma = 0.5$) for each $\{u_m\}_{m=1}^{M}$ are used. For the online algorithm, the step-size is $\alpha = 0.002$, that corresponds to the optimal step-size for the online gradient descent algorithm considered here.

Figure 4.3 shows the behavior of the performance metric for the SGP-OPGD algorithm, i.e., the dynamic regret $\frac{1}{T}\sum_{T=1}^{T} |f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)|$. It can be seen that the dynamic regret exhibits a $\mathcal{O}(1)$ asymptotic behavior; the jumps in the dynamic regret corresponds to instants where the reference $y_{\text{ref},t}$ changes abruptly.

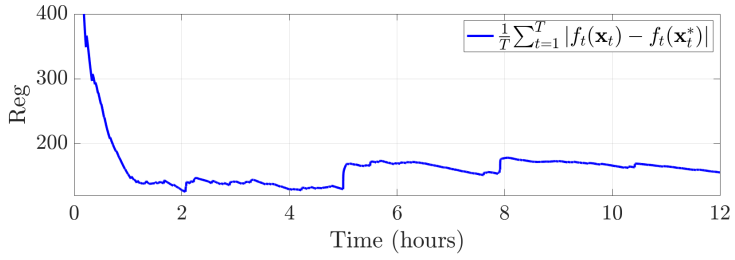


Figure 4.3: Dynamic regret of the SGP-OPGD algorithm.

4.2 Gradient-based Approach with Infrequent Feedback

In this section, we consider an application in the context of demand response in power distribution systems [95], for the case where we have infrequent updates form the distribution power network [119]. Here, $\mathbf{x}$ is the vector of active power setpoints from controllable distributed energy resources (DERs), $\mathbf{w}_t$ is the vector of powers consumed by non-controllable loads, $\mathbf{y}_t$ represents the net real power exchanged at some points of common coupling (PCC), and the map $\mathcal{M}(\mathbf{x}, \mathbf{w}_t) = \mathbf{G}\mathbf{x} + \mathbf{H}\mathbf{w}_t$ is built based on a linearization of the power flow equations [27]. We assume that the powers consumed by non-controllable loads cannot be individually measured; rather, measurements of $\mathbf{y}_t$ are available from meters and sensing units. The function $U_t(\mathbf{x})$ represents the dissatisfaction of the users (e.g., relative to indoor temperature if the DER is an AC unit, or charging rate of an electric vehicle); finally, we consider the function $C_t(\mathbf{y}_t) = \frac{\beta}{2}\|\mathbf{y}_t - \mathbf{y}_{t,\text{ref}}\|^2$, with $\mathbf{y}_{t,\text{ref}}$ a time-varying demand response setpoint for the points of common coupling (PCCs), and $\beta > 0$ a given parameter.

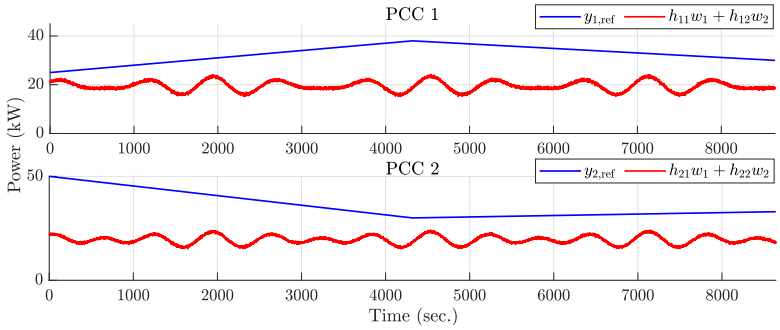
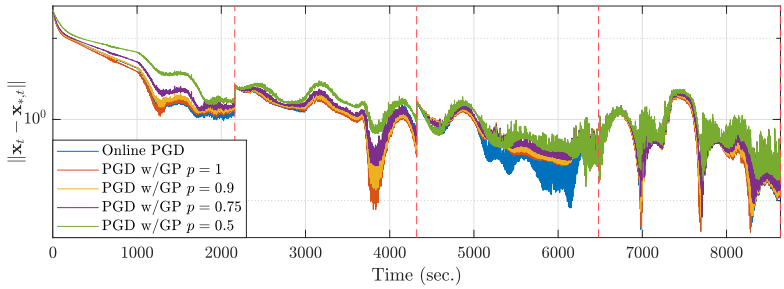
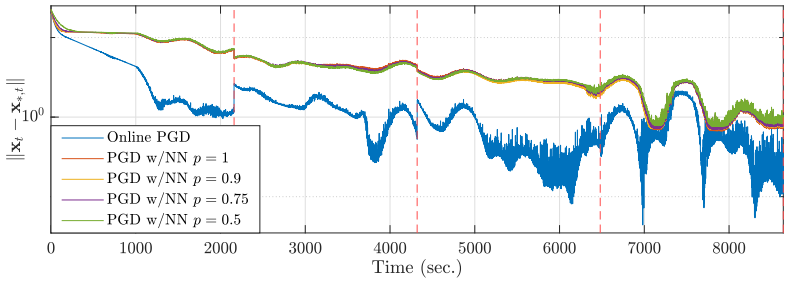


Figure 4.4: Reference signal $\mathbf{y}_{\text{ref}}$ and overall contribution of the non-controllable loads at the point of common coupling.



(a)



(b)

Figure 4.5: Error $\|\mathbf{x}_t - \mathbf{x}_{*,t}\|$ for different values of p. (a) Learning with GPs. (b) Learning with feedforward NNs. In both cases, the blue line corresponds to the online algorithm with exact knowledge of $\{u_m\}$.

We consider $M = 6$ controllable DERs (accordingly, $x_m \in \mathbb{R}$ is the active power produced or consumed by the DER m), and 2 PCCs. The power limits for the DERs are time-variant and change within the following bounds: $\mathcal{X}_{1,t} = [\{-10, -6\}; \{6, 10\}]$kW, $\mathcal{X}_{2,t} = [\{3, 7\}; \{13, 17\}]$kW, and $\mathcal{X}_{3,t} = [\{0, 3\}; \{28, 32\}]$kW. The aggregate power of the non-controllable loads are shown in red in Figure 4.4, and the two demand response setpoints are color-coded in blue. The function $U_t(\mathbf{x})$ is unknown; it is assumed that $U_t(\mathbf{x})$ switches between two quadratic functions (with different coefficients), to

reflect changes in the preferences of the DER-owners (switching times are represented by vertical red lines in Figure 4.5); for example, a DER owner can change preferences for the indoor temperature. We test two learning methods: *(i)* $U_t(\mathbf{x})$ is estimated via GPs as explained in Section 2.1.2, and *(ii)* we use feedforward neural networks (NNs) to estimate the functions associated with the DERs. We evaluate the performance of the algorithm in (2.30) over a period equivalent to 12 hours, with each step of the algorithm performed every 5 seconds. We start to learn each of the users' function with 5 noisy observations, and we collect functional evaluations from the DER owners every 30 minutes during the execution of the algorithm.

Figure 4.5(a) illustrates the performance of the online method when GPs are utilized (we use the labels "PGD w/GP"), averaged over 10 experiments. For each experiment, we use a fixed step size $\alpha = 0.5$ and a random initial point $\mathbf{x}_0$. We show the mean error for four values of the Bernoulli parameter p; for comparison purposes, we run the online proximal gradient method with exact knowledge of $U_t(\mathbf{x})$ (labeled as "online PGD"). The tracking error decreases linearly and then settles to a range of values that depend on p as expected. It can also be seen that, when $p = 1$, the red trajectory and the blue one are very close after 6000 steps, indicating that the GP approximates well the function $U_t(\mathbf{x})$. Figure 4.5(b) presents the case when $U_t(\mathbf{x})$ is learned via a feedforward NNs with one hidden layer of size 10 (labeled as "PGD w/NN"). The online algorithm exhibits a similar behavior; however, the tracking error is in general higher compared to the case where we use GPs. One reason for this behavior is the higher error in the gradient estimation; while GPs offer a closed-form expression for the gradient of the posterior mean, in the case of the NNs we estimated the gradient via centered difference.

We considered a feedback-based projected gradient method to solve a time-varying optimization problem associated with a system modeled with an algebraic map. The algorithm relies on inaccurate gradient information and exhibits random updates. We derived bounds for the error between the iterate of the algorithm and the optimal solution of the optimization problem in expectation and in high probability, by modeling gradient errors as sub-Weibull rvs and missing measurements as Bernoulli rvs. We established a connection with results in the context of ISS in

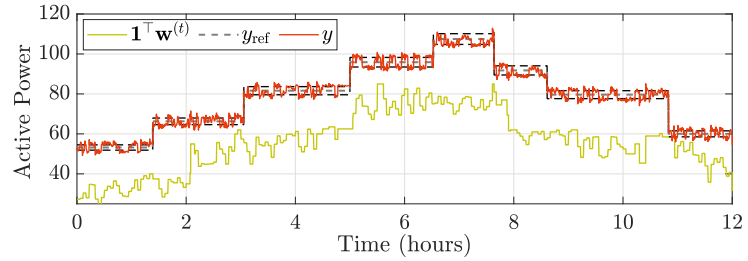
expectation and in high probability for discrete-time stochastic dynamical systems.

4.3 Consensus-Based Strategy

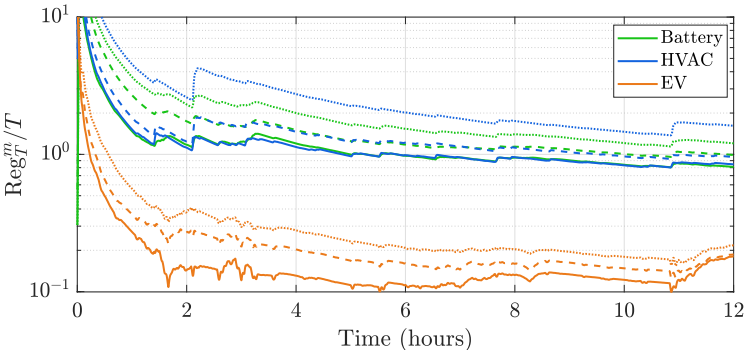
In this section, the proposed online data-driven primal-dual method with learning algorithm is numerically evaluated on a problem related to real-time management of distributed energy resources (DERs) at the power distribution level. In particular, we consider an aggregation of controllable DERs as well as uncontrollable loads connected to a distribution transformer. DERs include HVAC systems, electric vehicles, and energy storage systems. We consider $M = 3$ DERs shared among $N = 6$ users; specifically, $N_1 = 2$, $N_2 = 3$, $N_3 = 1$. The goal is to minimize the discomfort or dissatisfaction for each user while enabling the aggregation of DERs to actively emulate a virtual power plant where the total power $y_t = \mathbf{1}^\top \mathbf{x}_t^{\text{in}} + \mathbf{1}^\top \mathbf{w}_t$ follows a given automatic gain control or demand response reference signal $y_{\text{ref},t}$. The operational sets for the devices are: battery $\mathcal{X}_1 = [-8, 8]$ kW; HVAC $\mathcal{X}_2 = [0, 10]$ kW; and electric vehicle $\mathcal{X}_3 = [2, 30]$ kW. We note that, in the context of inverter-interfaced DERs, the set of feasible active power setpoints is typically convex.

The discomfort functions $\{u^{m,n}\}$ are assumed to be quadratic; the minimum of each of the functions is inside the set constraints $\mathcal{X}^m$, and it corresponds to a preferred setting of the user. For example, for EVs they represent a preferred charging rate; for HVAC systems, they represent a preferred temperature setpoint (converted into a preferred power setpoint) [95].

The time-varying constraint is $C_t(\mathbf{x}_t^{\text{in}}) = \frac{\beta}{2}(\mathbf{1}^\top \mathbf{x}_t^{\text{in}} + \mathbf{1}^\top \mathbf{w}_t - y_{\text{ref},t})^2 - \zeta_t$; where ζ_t is a given tolerance, which is set to 5% of the value of $y_{\text{ref},t}$. The power of the uncontrollable loads is taken from the Anatolia dataset (National Renewable Energy Laboratory, Tech. Rep. NREL/TP-5500-56610), and have a granularity of 1 second (see Figure 4.6). In this case, the evaluation of the gradient of C_t requires measurements of the total power at each step t. The proposed algorithm has a "gather-and-broadcast" architecture where the updates of x_{t+1}^m are computed at M parallel steps, and the updates for $x_{t+1}^{m,n}$ at N_m parallel steps. The measurements of the non-controllable devices are collected at a central location, the gradient of C_t is broadcasted to the devices, and $x_t^{m,n}$ are computed locally at each user.



(a)



(b)

Figure 4.6: (a) Tracking of y_{ref} by the GP-based online primal-dual method and (b) Local network regret (solid line corresponds to true $\{u_{m,n}\}$, dashed line to estimate $\{\hat{u}_{m,n}\}$ via shape-constrained GPs, and dotted line to estimate $\{\hat{u}_{m,n}\}$ via GPs).

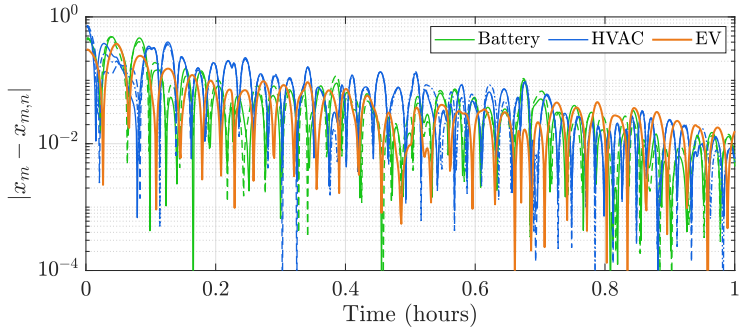


Figure 4.7: Disagreement between system and users (zoom for the first hour of simulation where each user is represented by a different line style). The primal variables x_m and $x_{m,n}$ are normalized such that 10^{-2} corresponds to 1% error.

We evaluate the performance of the online algorithm at a period of time equivalent to 12 hours; each step of Algorithm 2 is performed every 5 seconds (except for the HVAC systems, which are updated at a slower rate). The priors $\{\hat{U}_{m,n}\}$ are determined from some noisy measurements ($\sigma = 1.5$) and the discomfort function is updated through the user's feedback every 30 min. In this

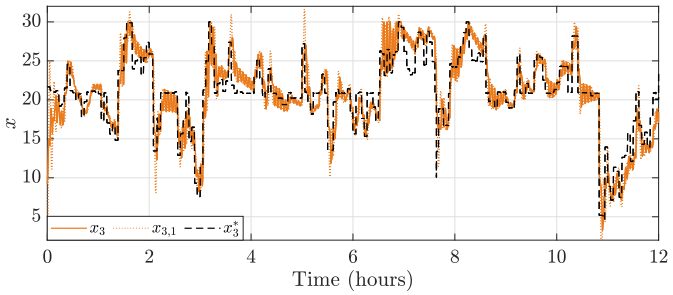


Figure 4.8: Behavior of the setpoint x^3 and the desirable setpoint for the user $x_{3,1}$, compared with the optimal solution x_*^3

case the parameter L_u and γ_u are defined beforehand; however, these values can be estimated via cross-validation. In Figure 4.6(a) it can be seen that the trajectory y is within 5% of the reference setpoint y_{ref} most of the time, in spite of the variation of the uncontrollable loads.

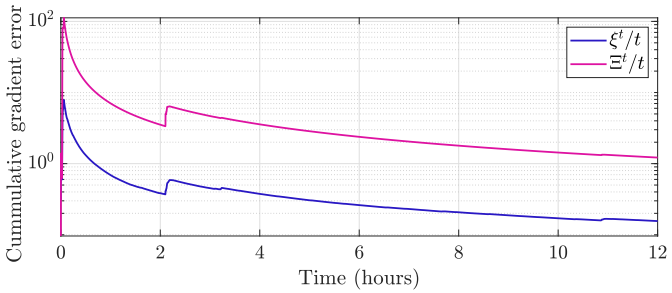


Figure 4.9: Behavior of ξ^t and Ξ^t over time.

The local network regret is presented in Figure 4.6(b). It can be seen that the jumps in the dynamic regret corresponds to instants where the reference y_{ref} changes abruptly. The regret of the proposed algorithm is, as one would expect, higher than the clairvoyant case where the functions $\{u^{m,n}\}$ are known. However, the difference diminishes over time, as the estimation error decreases. Further, Figure 4.7 shows the behavior of the users' disagreement during the first hour of the simulation, while Figure 4.8 illustrates the trajectory of the individual preferences of the user connected to the EV (system 3). Additionally, Figure 4.9 shows how the metrics in (3.11), related to the accuracy of the gradient estimates of the cost function, decrease over time.

Finally, we test the performance of Algorithm 2 in terms of the global network regret under three cases. First, Figure 4.10 illustrates the sensitivity of the proposed method to the choice

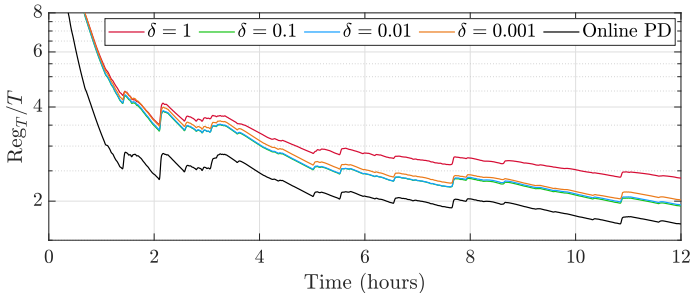


Figure 4.10: Global network regret for different values of δ.

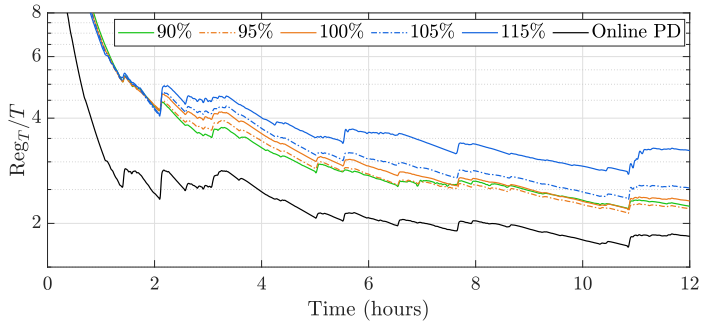


Figure 4.11: Global network regret over time under different load profiles.

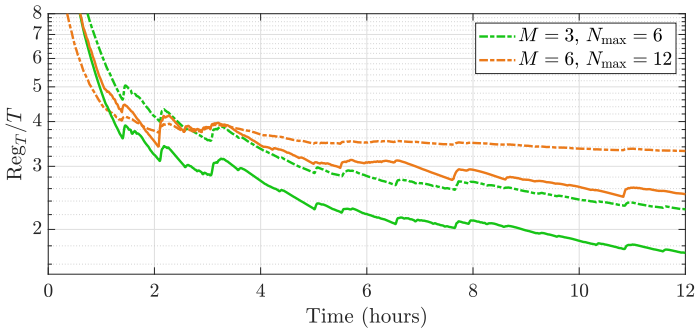


Figure 4.12: Global network regret for different values of M and $N_{\max}$ (solid line corresponds to true $\{u^{m,n}\}$, dashed line to estimate $\{\hat{u}^{m,n}\}$ via shape-constrained GPs).

of the parameter δ in (3.6). Second, Figure 4.11 presents the behavior under variability of the uncontrollable loads, in particular when the max/min load bounds increase/decrease. Lastly, Figure 4.12 shows that the global network regret increases when the total number of devices M and the total number of user of the network $N_{\max}$ increase, as expected. In these three cases, we provide the trajectory for the online algorithm with perfect knowledge of the users' function (Online PD) for comparison purposes.

Chapter 5

Learning-Based Demand Response in Buildings

Buildings have been shown to consume approximately 40% of the global energy use, where a large portion is utilized by heating, ventilation, and air-conditioning (HVAC) systems [78]. In this context, the control of HVAC systems plays an important role to maintain thermal comfort and achieving energy saving goals, particularly in commercial buildings. Building control techniques have demonstrated significant improvements related to energy savings [56], where historical data is leveraged to learn detailed models of the building dynamics. The strong coupling between the total energy consumption and the HVAC systems in commercial buildings opens the door to grid-interactive strategies, which enable these assets to provide services to distribution power systems in terms of demand response services or frequency support (see, e.g., [174, 182] and pertinent references therein).

Traditional model predictive control (MPC) strategies have shown promising results in terms of building operation efficiency and cost savings when participating in demand response programs [174]. However, traditional MPC approaches require an accurate model of the building - including the dynamics of the indoor temperature - which usually may not be feasible in real applications due to the building complexity and the dependence on prevailing weather and ambient conditions. Thus, in this work, we present a model-free learning-based predictive control approach that overcomes some of the limitations in traditional MPC approaches for building control.

A GP-based predictive controller (GP-PC) implements a GP-based model that can be used for receding horizon optimal control; this model has the characteristic of providing probabilistic

guarantees, as shown in e.g., [81], [37]. Moreover, GP offers a supervised learning technique that is highly adaptable, and thus attractive in settings where both the environment and the control objectives are time-varying. In applications such as predictive control, the use of GPs allows us to continuously update the GPs model concurrently with the operation of the controller. There are four main advantages of this technique: (i) GPs provide a confidence level of the prediction through the variance. (ii) GPs require a small size of data sets in order to provide meaningful estimates; therefore, they are a sample efficient method. (iii) It is possible to incorporate prior knowledge of the system to improve the learning process via hyperparameters selection or the construction of a particular covariance (kernel) function. (iv) GPs provide a closed-form expression for the posterior mean and variance.

Data-driven techniques for building energy management have been extensively studied, see for example, regression trees [82], Gaussian processes regression [81], reinforcement learning [182], deep reinforcement learning [174], just to name a few. In particular, predictive control approaches based on GPs were considered initially in [87] for controlling the pH neutralization process. In [37] two algorithms for GP-based predictive control were introduced and applied in trajectory tracking problems of a multi-input multi-output nonlinear system with time-varying parameters, and an application of commercial quadrotor is presented in [18]. Moreover, modeling and control of dynamic systems using Gaussian process regression is presented in [88]. We also acknowledge works in the context of discrete zeroth-order methods with application in power systems such as [47, 103].

The main contributions of the work are as follows: *i)* We formulate a GP-based predictive control scheme that is able to strike a balance between operational objectives, demand response objectives, and the thermal discomfort cost. *ii)* We implement a zero-order primal-dual projected-gradient method to solve the optimization problem. *iii)* We show how to train the GP model periodically to adaptively incorporate changes in the environment, e.g., occupants' behaviors or weather conditions. In particular, for the GP characterization, we implement a composite covariance function using a squared exponential kernel and a locally periodic kernel that captures specific

properties of the building dynamics behavior.

In the context of data-driven control for building energy management, to the best of our knowledge, we proposed the first learning-based predictive control for demand response task in a multi-zone building, where the temperature dynamics per zone are estimated via GPs regression, different from the approach in [81] where the total power consumption of the building is learned via GP. As an application example, the proposed algorithm is applied to solve a demand response problem in a five-zone commercial building that interacts with the distribution power system. In this setting, the proposed methodology enables aggregations of distributed energy resources to provide services to the grid (in the form of, e.g., frequency and voltage support) while minimizing operational costs and thermal discomfort.

The remainder of this chapter is organized as follows. Section 5.1 preliminary definitions, and Section 5.2 the problem formulation. Section 5.3 presents the proposed algorithm, while Section 5.4 provides the case study. Section 5.5 concluded the chapter and present further directions.

5.1 Gaussian Processes Regression for Dynamics Learning

For completeness of this section below we recalled the GPs definitions. As shown in Section 2.1.2, with the difference that here we present the definition in terms of a vector input variable.

Consider noisy functional evaluations given by $y = G(\mathbf{z}) + \mathcal{N}(0, \sigma^2)$ of an underlying unknown function $G : \mathbb{R}^{n_z} \to \mathbb{R}$, $\mathbf{z} \in \mathbb{R}^{n_z}$, and n_z the total number of input variables. In this chapter, we propose to learn G via Gaussian Processes (GPs) regression. A GP is a stochastic process and it is characterized by its mean $\mu(\mathbf{z})$ and its covariance function $k(\mathbf{z}, \mathbf{z}^*)$ where $k(\cdot, \cdot)$ is a given kernel, i.e., for any $\mathbf{z}, \mathbf{z}^* \in \mathbb{R}^{n_z}$, $\mu(\mathbf{z}^*) = \mathbb{E}[G(\mathbf{z}^*)]$ and $k(\mathbf{z}, \mathbf{z}^*) = \mathbb{E}[(G(\mathbf{z}) - \mu(\mathbf{z}))(G(\mathbf{z}^*) - \mu(\mathbf{z}^*))]$ [133].

The GP that estimates the unknown function is denoted by $G(\mathbf{z}) \sim \mathcal{GP}(\mu(\mathbf{z}), k(\mathbf{z}, \mathbf{z}^*))$. Given the inputs vectors $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N]^\top$ and the corresponding observed outputs $\mathbf{y} = [y_1, y_2, \ldots, y_N]^\top$ we can defined the training data by $\mathcal{S} = (\mathbf{Z}, \mathbf{y})$. Deriving the conditional posterior distribution of $(G(\mathbf{z}^*)|\mathcal{S}, \mathbf{z}^*)$, we get the key predictive equations for GP regression $G(\mathbf{z}^*)$ for

a new inputs $\mathbf{z}^*$ as

$$\mu(\mathbf{z}^*) = \mathbf{k}(\mathbf{z}^*)^\top (\mathbf{K}_N + \sigma^2 \mathbf{I})^{-1} \mathbf{y}, \tag{5.1a}$$

$$k_N(\mathbf{z}, \mathbf{z}^*) = k(\mathbf{z}, \mathbf{z}^*) - \mathbf{k}(\mathbf{z})^\top (\mathbf{K}_N + \sigma^2 \mathbf{I})^{-1} \mathbf{k}(\mathbf{z}^*) \tag{5.1b}$$

$$\varsigma^2(\mathbf{z}^*) = k_N(\mathbf{z}^*, \mathbf{z}^*), \tag{5.1c}$$

where $\mathbf{k}(\mathbf{z}^*) = [k(\mathbf{z}_i, \mathbf{z}^*)_{i=1}^N]^\top$, $\mathbf{K}_N$ is the positive definite kernel matrix $[k(\mathbf{z}, \mathbf{z}^*)]_{\mathbf{z},\mathbf{z}^* \in \mathbf{Z}}$, and the subscript N indicates the number of data points in $\mathbf{Z}$. The correlation between the points $\mathbf{z}$ and $\mathbf{z}^*$ is indicated through the covariance (or kernel) function $k(\cdot, \cdot)$. A GP approach gives us an estimate of the function that just need the structure of the covariance matrix rather that a fixed structure of the input-output relation, which makes this technique highly flexible with fewer parameters. In particular, the kernel function $k(\mathbf{z}, \mathbf{z}^*)$ selected in this work is a combined covariance function using a squared exponential (SE) kernel $k_{\text{SE}}(\mathbf{z}, \mathbf{z}^*)$ and a periodic kernel $k_{\text{PER}}(\mathbf{z}, \mathbf{z}^*)$, called locally periodic kernel and defined as:

$$k_{\text{SE}}(\mathbf{z}, \mathbf{z}^*) = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{i=1}^{n_z} \frac{(z_i - z_i^*)^2}{l_i^2} \right)$$

$$k_{\text{PER}}(\mathbf{z}, \mathbf{z}^*) = \exp\left(-2 \sum_{i=1}^{n_z} \frac{\sin^2(\pi |z_i - z_i^*| / p)}{\lambda_i^2} \right)$$

$$k(\mathbf{z}, \mathbf{z}^*) = k_{\text{PER}}(\mathbf{z}, \mathbf{z}^*) \times k_{\text{SE}}(\mathbf{z}, \mathbf{z}^*), \tag{5.2}$$

where the hyperparameters set is $\boldsymbol{\theta} = [\sigma_f, \mathbf{l}, p, \boldsymbol{\lambda}]^\top$, $\boldsymbol{\lambda} = [\{\lambda_i\}_{i=1}^{n_z}]^\top$ and $\mathbf{l} = [\{l_i\}_{i=1}^{n_z}]^\top$. The signal variance is represented by σ_f^2, the period p determines the distance between repetitions of the function, and $\mathbf{l}$, $\boldsymbol{\lambda}$ corresponds to the characteristic length-scale for the squared exponential kernel and a periodic kernel, respectively. This combined kernel results in functions that are periodic, but which can slowly vary over time [57]. In this case, $\boldsymbol{\theta}$ is learned by maximizing the likelihood [133], i.e., $\arg\max_{\boldsymbol{\theta}} \Pr(\mathbf{y}|\mathbf{Z}, \boldsymbol{\theta})$. In this application, we have a periodic behavior of the temperature associated with the 24 hours of the day. However, this behavior does not repeat exactly on all days, and thus we add more flexibility to the model by multiplying by a local square exponential kernel to model functions that are only locally periodic, i.e., over time the shape of the function changes following a periodic pattern.

GPs can model nonlinear dynamical systems where autoregressive input and output signals feedback the model as regressors [37]. Key differences between the GP used for regression and the one used for control synthesis [81, 87] is that in a GP regression we have access to all inputs; however, in the GPs used for control, some of the inputs are the control variables that we need to optimize in real-time to meet some performance criteria.

With this definition in place, in the following, we introduce the problem formulation in the context of multi-zone building control.

5.2 Problem Formulation

We consider a multi-zone commercial building with A zones, and we set $\mathcal{A} := \{1, \dots, A\}$. Usually, commercial buildings use a centralized HVAC system for cooling; in this case, the control variables of the HVAC system are the cooling air flow rate to each zone ($u^i, i \in \mathcal{A}$). These variables are controlled to provide a comfortable indoor environment for occupants. For future developments, let $\mathbf{u}_t = [u_t^1, \dots, u_t^A]^\top \in \mathbb{R}^{n_u}$ be the controllable inputs of the system, and $n_u = A$ the total number of control variables.

We let x_t^i denote the indoor temperature of zone i at time t. The vector $\mathbf{x}_t \in \mathbb{R}^A$ collecting the indoor temperatures of all the zones is a state variable, whose dynamics will be estimated via a GP-based method below. The exogenous (uncontrollable) input variables are collected in $\mathbf{w}_t = \{\mathbf{w}_t^i\}_{i\in\mathcal{A}} \in \mathbb{R}^{n_w}$. Specifically, $\mathbf{w}_t^i = [T_t^{\text{out}}, T_t^{\text{da}}, Q_t^{i,\text{int}}, Q_t^{i,\text{sol}}]^\top$, where T_t^{out} is the outside temperature of the building, T_t^{da} is the chiller discharge air temperature, $Q_t^{i,\text{int}}$ is the solar heat gain, and $Q_t^{i,\text{sol}}$ is the internal heat gain.

To quantify the thermal discomfort, we let $\mathcal{D} : \mathbb{R} \to \mathbb{R}$ denote a cost function that measures the temperature deviation from a defined comfort range $[\underline{T}_t^i, \overline{T}_t^i]\, \forall i \in \mathcal{A}$, given by

$$\mathcal{D}(x_t^i) := \begin{cases} \max(x_t^i - \overline{T}_t^i,\, (x_t^i - \overline{T}_t^i)^2), & x_t^i > \overline{T}_t^i, \\ \max(\underline{T}_t^i - x_t^i,\, (\underline{T}_t^i - x_t^i)^2), & x_t^i < \underline{T}_t^i, \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

We model the HVAC power consumption via the function

$$\mathcal{P}(\mathbf{u}_t, \mathbf{w}_t) := \frac{a}{b}(T_t^{\text{out}} - T_t^{\text{da}}) \sum_{i=1}^{A} u_t^i + c \left(\sum_{i=1}^{A} u_t^i \right)^3 + d \tag{5.4}$$

where $a, b, c, d \in \mathbb{R}$ are given constants, and in particular, b represents the coefficient of performance (COP) [105]. Furthermore, to guarantee the proper operation of the HVAC system, the air flow rate per zone must by within a specific range $u_t^i \in [\underline{m}_t^i, \overline{m}_t^i]$, $\forall i \in \mathcal{A}$, $\forall\, t$, and $\sum u_t^i \leq h$ for a given $h > 0$ at all time. In the following, let $\mathbf{z}^i = [x^i, u^i, \mathbf{w}^i]^\top$, $i \in \mathcal{A}$.

Next, we introduce the formulation of the GP-based predictive control (GP-PC) problem. By following the approach presented in [116], we use the *zero-variance* method to estimate the temperature dynamics, where the estimation is defined using the posterior mean (5.1a), and therefore does not propagate uncertainty. Moreover, by using the probabilistic non-parametric properties of the GP regression, we can incorporate the model uncertainties in the cost function while solving the multi-period optimization problem. At each time t, the GP-PC problem is then given by

$$\min_{\{\mathbf{u}_{t+k}, \mathbf{x}_{t+k}\}} \sum_{k=0}^{K-1} [\kappa_1 \mathcal{P}(\mathbf{u}_{t+k}, \mathbf{w}_{t+k}) \Delta k + \rho \|\boldsymbol{\varsigma}(\mathbf{z}_{t+k})\|^2 + \kappa_2 (\mathcal{P}(\mathbf{u}_{t+k}, \mathbf{w}_{t+k}) - P_{t+k}^{\text{ref}})^2 + \kappa_3 \sum_{i=1}^{A} \mathcal{D}(x_{t+1+k}^i)] \tag{5.5a}$$

$$\text{s. to:} \quad x_{t+k+1}^i = \mu(\mathbf{z}_{t+k}^i) \tag{5.5b}$$

$$\sum_{i=1}^{A} u_{t+k}^i \leq h, \tag{5.5c}$$

$$\underline{m}^i \leq u_{t+k}^i \leq \overline{m}^i, \ \forall i \in \mathcal{A}, \ \forall k \in \{0, 1, \ldots, K-1\}, \tag{5.5d}$$

where $x_{t+k+1}^i = \mu(\mathbf{z}_{t+k}^i)\, \forall i \in \mathcal{A}$ are the building temperature dynamics estimated via (5.1a), i.e., we utilize one GP per zone; $\boldsymbol{\varsigma}(\mathbf{z}) = [\varsigma(\mathbf{z}^1), \ldots, \varsigma(\mathbf{z}^A)]^\top$ is the vector of the standard deviation (square root of (5.1c)) per zone; $\mathbf{w}_{t+k}^i$ are non-controllable exogenous inputs, κ_j is a weight factor for each objective of the cost function such that $\sum_{j=1}^{3} \kappa_j = 1$, and K is the control horizon. The penalty ρ helps to select control variables where the model is more confident.

In the following, we present the proposed GP-based learning control algorithm to solve problem (5.5).

Remark 6 In this work, we consider a demand response event for active power tracking. If the objective is to have a demand response event related to a time-of-use (ToU) electricity price, for example, we can modify the cost function (5.5a) to reflect a change in the electricity prices per period of time.

5.3 Gaussian Process-based Learning Control Algorithm

In this section, we present a zero-order primal-dual projected-gradient algorithm to solve problem (5.5). In the following, we drop the subscript t knowing that the procedure is repeated at each time. We first remove the redundant variable $\mathbf{x}_{t+k}$ in the cost function (5.5a) by applying the equation (5.5b). Note in (5.5b) that, for a given current state x_0^i, we have

$$x_1^i = \mu(x_0^i, u_0^i, \mathbf{w}_0^i)$$

$$x_2^i = \mu(x_1^i, u_1^i, \mathbf{w}_1^i) = \mu(\mu(x_0^i, u_0^i, \mathbf{w}_0^i), u_1^i, \mathbf{w}_1^i) \tag{5.6}$$

$$\vdots \tag{5.7}$$

In this iterative manner, for any $0 \leq k \leq K$, x_k^i can be expressed as the function of known current state x_0^i and variables $\{\mathbf{u}_k^i,\ \mathbf{w}_k^i\}_{k=0}^{K-1}$. Therefore, we substitute variable $\mathbf{x}$ by using the state evolution equations (5.7), and denote $J(\mathbf{u})$ as the cost function in (5.5a).

We then rewrite (5.5c) as

$$\mathbf{g}(\mathbf{u}) = [g_0(\mathbf{u}), g_1(\mathbf{u}), \ldots, g_{K-1}(\mathbf{u})]^\top, \tag{5.8}$$

where $g_k(\mathbf{u}) := \sum_{i=1}^{A} u_k^i - h \leq 0$. Letting $\boldsymbol{\lambda} \in \mathbb{R}^K$ denote the dual variables associated with the inequality constraints (5.8), we obtain the Lagrange function

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\lambda}) = J(\mathbf{u}) + \boldsymbol{\lambda}^\top \mathbf{g}(\mathbf{u}). \tag{5.9}$$

The box constraint (5.5d) is considered as hard constraint set on a convex set $\mathcal{U}$, where $\mathcal{U}$ is the Cartesian product between the box constraints per zone in (5.5d). Therefore, the constrained optimization problem (5.5) is converted into the following saddle-point problem

$$\max_{\boldsymbol{\lambda} \geq 0} \min_{\mathbf{u} \in \mathcal{U}} \mathcal{L}(\mathbf{u}, \boldsymbol{\lambda}), \tag{5.10}$$

which can be solved by the projected primal-dual algorithm with the following gradient

$$\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u},\boldsymbol{\lambda}) = \nabla_{\mathbf{u}}J(\mathbf{u}) + \mathbf{H}^{\top}(\mathbf{u})\,\boldsymbol{\lambda}$$

$$\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\mathbf{u},\boldsymbol{\lambda}) = \mathbf{g}(\mathbf{u}),$$

where $\mathbf{H}(\mathbf{u})$ is the Jacobian matrix of $\mathbf{g}(\mathbf{u})$:

$$\mathbf{H}(\mathbf{u}) = \begin{bmatrix} \frac{\partial g_0}{\partial u_1} & \frac{\partial g_0}{\partial u_2} & \cdots & \frac{\partial g_0}{\partial u_{K\times n_u}} \\ \frac{\partial g_1}{\partial u_1} & \frac{\partial g_1}{\partial u_2} & \cdots & \frac{\partial g_1}{\partial u_{K\times n_u}} \\ \vdots & \ddots & \cdots & \vdots \\ \frac{\partial g_{K-1}}{\partial u_1} & \frac{\partial g_{K-1}}{\partial u_2} & \cdots & \frac{\partial g_{K-1}}{\partial u_{K\times n_u}} \end{bmatrix}.$$

Notice that the computation of $\nabla_{\mathbf{u}}J(\mathbf{u})$ requires the gradient of K GPs per each zone as indicated in (5.5b). Therefore, if we use a commercial solver to solve (5.5), e.g., `fmincon` solver in MATLAB or CasADi, we might face computational challenges due to the complexity of the cost function $J(\mathbf{u})$. To reduce the computational complexity, we use the zero-order gradient approximation method with two function evaluations, as presented in [47]. In particular, the estimated gradient of the multi-objective cost function is:

$$\hat{\nabla}_{\mathbf{u}}J(\mathbf{u}) = \frac{1}{2\gamma}\boldsymbol{\xi}\left[J(\mathbf{u}+\gamma\boldsymbol{\xi}) - J(\mathbf{u}-\gamma\boldsymbol{\xi})\right], \tag{5.11}$$

where $\gamma > 0$ is a small scalar and $\boldsymbol{\xi} \in \mathbb{R}^{K\times n_u}$ is a exploration vector with each element corresponding to one control variable per horizon time. Applying the Taylor series expansion, it can be shown that

$$\hat{\nabla}_{\mathbf{u}}J(\mathbf{u}) = \boldsymbol{\xi}\boldsymbol{\xi}^{\top}\nabla_{\mathbf{u}}J(\mathbf{u}) + O(\gamma^2).$$

Candidate exploration signals include independent random variables and sinusoidal signals with different frequencies, chosen such that in expectation or on average, $\boldsymbol{\xi}\,\boldsymbol{\xi}^{\top}$ is an identity matrix; see [47] for details.

Algorithm 3: GP-based primal dual method

Initialization

(a) Step size $\alpha > 0$ and small $\delta > 0$ stopping criteria.

for $t \in \mathcal{T}$ **do**

  if $t = 1$ **then**

    (b) Dual variable $\boldsymbol{\lambda}_t^0 = \mathbf{0}$.

    (c) Controllable variable $\mathbf{u}_t^0 = \underline{\mathbf{m}}^0$ with $\underline{\mathbf{m}}^0 = [{\mathbf{m}_{\min}^0}^\top, \ldots, {\mathbf{m}_{\min}^0}^\top]$, $\underline{\mathbf{m}}_{\min}^0 = \{\underline{m}^i\}_{i \in \mathcal{A}}$.

  else

    (d) Dual variable $\boldsymbol{\lambda}_t^0 = \boldsymbol{\lambda}_{t-1}^0$.

    (e) Controllable variable $\mathbf{u}_t^0 = \mathbf{m}_{t-1}^0$ with $\mathbf{m}_{t-1}^0 = [\mathbf{u}_{0+t}^0, \ldots, \mathbf{u}_{t+K-1}^0]^\top$.

  end

  (f) Exploration signal $\boldsymbol{\xi}^j, j = 0, 1, \ldots$

  for $j = 0, 1, \ldots$ **do**

    (g) Forward exploration $\mathbf{u}_+^j = \mathbf{u}^j + \gamma \boldsymbol{\xi}^j$ and compute $J(\mathbf{u}_+^j)$.

    (h) Backward exploration $\mathbf{u}_-^j = \mathbf{u}^j - \gamma \boldsymbol{\xi}^j$ and compute $J(\mathbf{u}_-^j)$.

    (i) Gradient estimation

$$\hat{\nabla}_{\mathbf{u}} J(\mathbf{u}_t^j) = \frac{1}{2\gamma} \boldsymbol{\xi}^j \left[J(\mathbf{u}_+^j) - J(\mathbf{u}_-^j) \right]$$

    (j) Primal update

$$\mathbf{u}_t^{j+1} = \mathrm{proj}_{\mathcal{U}} \left\{ \mathbf{u}_t^j - \alpha \left(\hat{\nabla}_{\mathbf{u}_t^j} J(\mathbf{u}_t^j) + \boldsymbol{\lambda}_t^j \right) \right\}$$

    (k) Dual update, define $u_{t+k}^{\mathrm{s}} = \sum_{i=1}^{A} u_{t+k}^i$ and $\mathbf{u}_t^{\mathrm{s}} = [u_{t+0}^{\mathrm{s}}, \ldots, u_{t+K-1}^{\mathrm{s}}]^\top$

$$\boldsymbol{\lambda}_t^{j+1} = \left[\boldsymbol{\lambda}_t^j + \alpha \left(\mathbf{u}_t^{\mathrm{s}} - h\mathbf{1} \right) \right]^+ \tag{5.12}$$

    if $|J(\mathbf{u}_{t-1}) - J(\mathbf{u}_t)| \leq \delta$ **then**

      break

    end

  end

end

The proposed approach is summarized in Algorithm 3. After initialization steps (a)-(f), the zero-order gradient is approximated in step (i) under two function evaluations. This approximate gradient $\hat{\nabla}_{\mathbf{u}} J(\mathbf{u})$ is then used in step (j) to perform the projected primal update for control variables $\mathbf{u}$, where $\mathrm{proj}_{\mathcal{U}}\{\ \}$ is the projection operator. The dual variable is updated in (5.12) (step (k)) to enforce the constraint on the total air flow rate (5.5c), where $[\ ]^+$ ensures the dual variable $\boldsymbol{\lambda} \geq 0$. It is worth mentioning that we use a warm-start strategy in our algorithm implementation. In this

GP-based predictive control formulation, problem (5.5) needs to be solved at each instant of time, then we use the previous optimal control variable $\mathbf{u}_{t-1}$ as the initial guess for the next time t.

5.4 Case Study

A five-zone small office model from EnergyPlus is used in this case study, as shown in Figure 5.1. The parameters for the HVAC power consumption in (5.4) are $a = 1$, $b = 3$, $c = 0.0076$, $d = 4.8865$, $h = 10$, $T^{\text{da}} = 13°C$, and control variables ranges are

$$u^i \in \begin{cases} [0.22, 2.2]\,\text{kg/s} & \text{if } i \neq 5 \\ [0.32, 3.2]\,\text{kg/s} & \text{if } i = 5 \end{cases}.$$

Exogenous data $T_k^{\text{out}}, Q_k^{i,\text{int}}, Q_k^{i,\text{sol}}$ is taken from EnergyPlus and correspond to July - August in Austin, TX. To simulate the real building dynamics and test the control outputs from (5.5), we used a reduced-order model (ROM) as presented in [183]. The ROM is developed using an Auto-Regressive model with eXogenous variables (ARX) [48], and the model parameters are learned from building historical data.

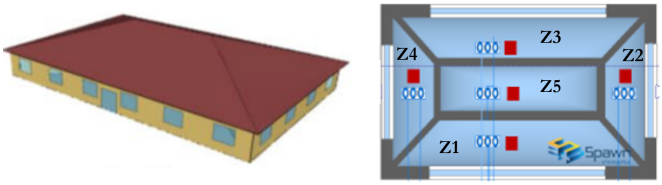


Figure 5.1: DOE commercial prototype building model for a 5-zones small-size office [50].

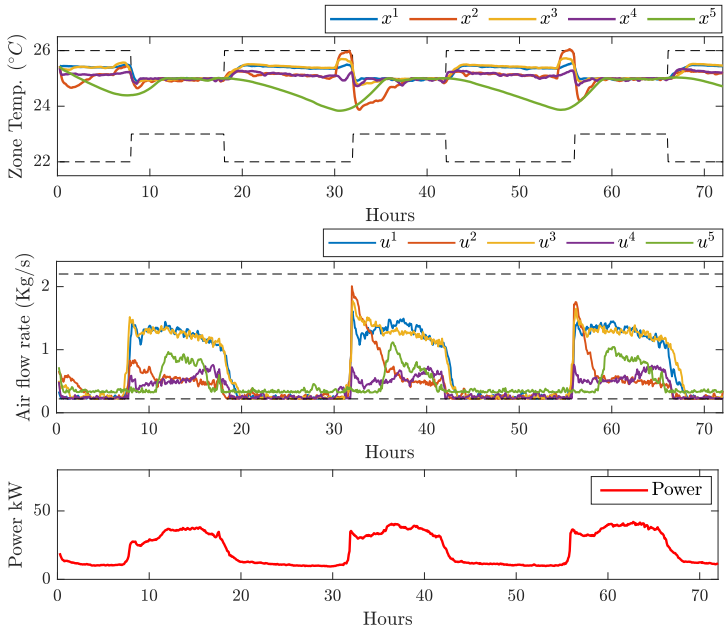
As explained in Section 5.1, we estimated the building temperature dynamics per zone via GPs to solve (5.5). Specifically, we used batch and normalized data from July 1^{st} to 12^{th} to create our training set $\mathcal{S}$ and we test Algorithm 3 on the data corresponding to July 1^{st} to 5^{th} and August 1^{st} to 5^{th}. The hyperparameters $\boldsymbol{\theta}$ and the kernel matrix evaluation in (5.2) were computed by using the Gaussian Processes for Machine Learning (GPML) toolbox [134]. We formulated the GP-PC problem with control horizon of 1 hour and 5 minutes control interval, i.e., $K = 12$. We

evaluated the performance of the proposed approach over 10 days (July 1^{st} to 5^{th}, corresponding to seeing data, and August 1^{st} to 5^{th} to unseeing data), i.e., $\mathcal{T} = \{1, \ldots, 2880\}$.

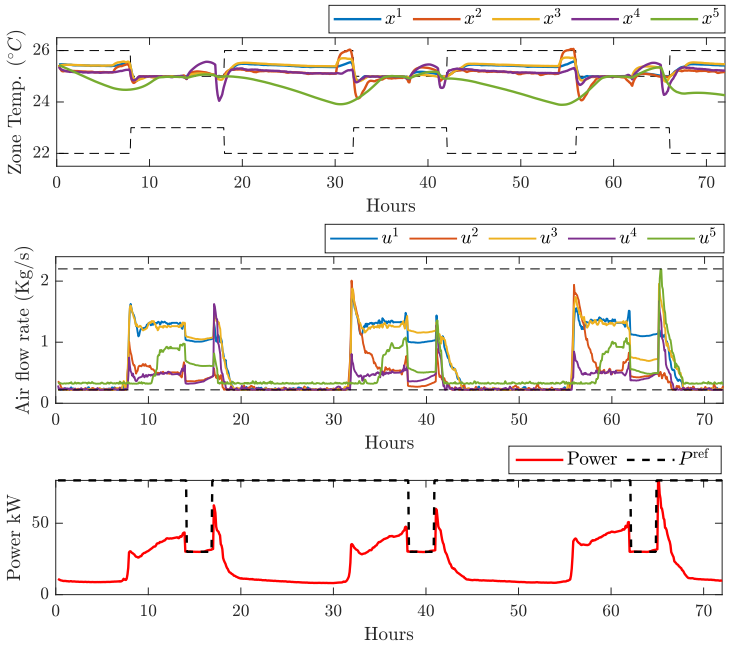
Two scenarios are studied. First, without demand response (DR) event (normal operation), we have $\boldsymbol{\kappa} = [0.2, 0, 0.8]$, i.e, we just consider the energy saving and thermal discomfort objectives. Second, when considering DR events, we have $\boldsymbol{\kappa} = [0, 0.5, 0.5]$, $P_t^{\text{ref}} = 30\text{kW}$ between 14:00 to 16:00 and $\boldsymbol{\kappa} = [0.2, 0.1, 0.7]$, $P_t^{\text{ref}} = 80\text{kW}$ otherwise, i.e, DR event happen within 14:00 to 16:00. Figure 5.2 shows the control performance of GP-PC for building temperature control during normal operation, i.e., considering just energy savings and thermal discomfort objectives, in Figure 5.2(a), and under the DR event in Figure 5.2(b). Both cases are presented over a sample of 3 days (August 1^{st} to 3^{rd}) for the non-DR and DR scenario.

The performance of the GP-PC is compared with two versions of model predictive controller (MPC): MPC-ROM and MPC-LIN, which are based on the exact reduced-order model (ROM) developed in [183, Secction II-B] and a linear approximation obtained from a Taylor series expansion of ROM, respectively. The MPC-ROM uses the same model to compute the optimal control actions and to simulate the building. As explained in [183], the MPC-ROM solves a non-convex optimization problem that may be infeasible for real-world applications. In this case study, we use the MPC-ROM as the best possible reference for the control performance, and MPC-LIN serves as a more realistic implementation to compare the performance of the proposed approach. Moreover, to illustrate the adaptability of GP-PC to model changes, we compare the performance of the proposed controller with the MPC-LIN in a setting where we have an outdated model on the controller. Both MPC-ROM and MPC-LIN are implemented using `fmincon` solver in MATLAB with the interior-point algorithm.

The GP-PC and the two MPCs are compared using 10 testing days (July 1^{st} – 5^{th} and August 1^{st} – 5^{th}). The results are presented in Figure 5.3. These results show that the GP-PC can achieve a better performance than the MPC-LIN where we have an outdated building model with a reduction on average of 13.24% of the total cost for a normal operation scenario and 15.76% for the DR case.



(a) Non-demand response scenario.

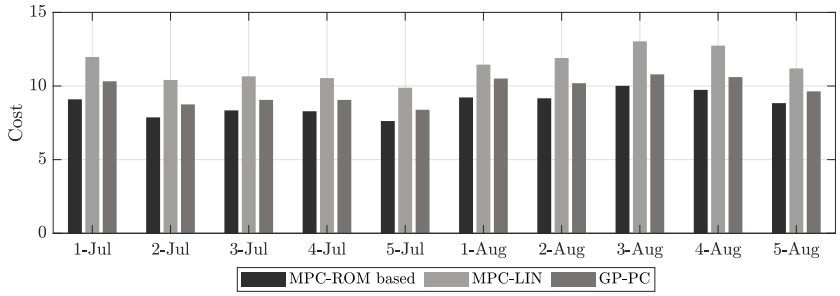


(b) Demand response scenario.

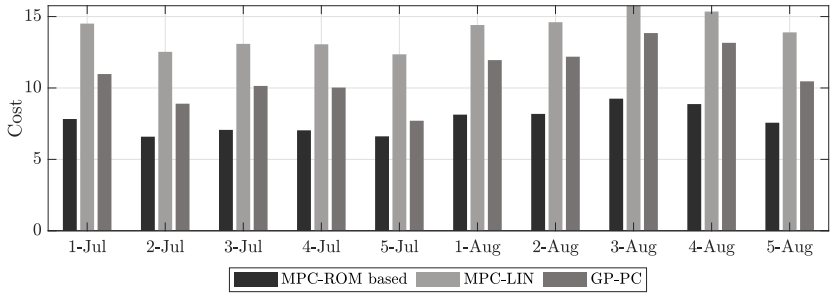
Figure 5.2: Results for the GP-PC performance on August 1$^{\text{st}}$ to 3$^{\text{rd}}$. (a) non-DR scenario: Top plot shows the indoor temperature profiles within the band spans $[23, 25]$°C and $[22, 26]$°C (dashed lines) during occupied and unoccupied periods, respectively; center plot is the air flow mass (m^i) profiles for each zone (maximum limit on dashed line); and bottom plot shows consumed power in kW for the building. (b) DR scenario: Top plot shows the indoor temperature profiles within the bands $[23, 25]$°C and $[22, 26]$°C (dashed lines) during occupied and unoccupied periods, respectively; center plot is the air flow mass (m^i) profiles for each zone (maximum limit on dashed line); and bottom plot shows the consumed building power and the active power reference.

We note that the GP regression model implemented to estimate the building temperature dynamics is using just 12 days of training data. This is an important advantage with respect to other data-driven optimal control approaches, such as the ones based on reinforcement learning, where hundreds of months of data are needed for training the models; see, e.g., [170, 180]. Finally, Figure 5.4 presents the performance of the proposed GP-PC method for the non-DR scenario under 3 options of training data, where we can see that after 12 days of training, there is not a significant improvement in the performance of the controller.

Finally, we report the computational time for this case study. The simulation was performed on a computer with the processor Inter(R) Core(TM) i7-8850H CPU @ 2.60GHz, 32.0 GB of RAM, and the 64-bit Operating System. For a time period of 24 hours ($T = 288$), the computational time per time step is 1 minute for the non-DR scenario and 2.2 minutes for the DR case, on average. It justifies our approach is computationally feasible in the proposed 5 minutes control window.



(a) Non-demand response scenario.



(b) Demand response scenario.

Figure 5.3: GP-PC performance for July 1$^{\text{st}}$ to 5$^{\text{th}}$ and August 1$^{\text{st}}$ to 5$^{\text{th}}$. (a) Non-demand response scenario (b) Demand response scenarios.

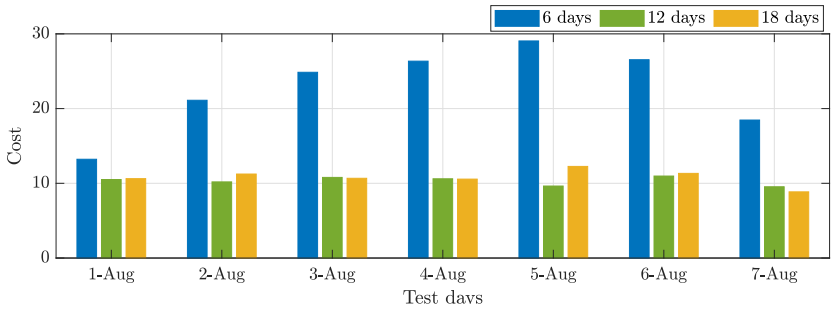


Figure 5.4: GP-PC performance for August 1st to 7th for three options of training data.

5.5 Conclusions

We presented a Gaussian process based predictive control (GP-PC) for a grid-interactive multi-zone commercial building where we showed that the GP-PC is able to control the indoor temperature per zone in a building under normal operation and demand response scenarios. We presented that the GPs regression allows effectively to predict the building temperature dynamics per zone with a small set of training data compare with other model-free techniques. Moreover, the zero-order primal-dual projected-gradient algorithm implemented in the GP-PC approach facilities to overcome computational challenges due to the complexity of the cost function. We corroborated the performance of the proposed GP-PC method in a standard five-zone commercial building.

Chapter 6

Learning of Sensitivities for Transmission & Distribution Systems

Sensitivities factors play an important role in power systems operations and control. For instance, in transmission systems, linear sensitivity distribution factors are traditionally utilized in power systems analysis – for e.g., contingency analysis, generation re-dispatch, and security assessment [184], just to mention a few. Injection shift factors [129, 45] as well as power transfer distribution factors (PTDFs) allow grid operators to estimate line flows in real-time in response to changes in the (net) power injections. At the distribution level, examples include voltage sensitivities (with respect to net power injections) [136], which can be used to adjust droop controllers in real-time [13], or used as a surrogate of the Jacobian matrix in distribution-system analysis [26], and in real-time optimal power flow algorithms [53].

Computation of these sensitivities typically relies on either *model-based* or *measurement-based* approaches. As an example of a model-based method, injection shift factors and the PTDF matrix for transmission systems are typically computed by leveraging the DC approximation [140]; similarly, voltage sensitivities can be computed via linear approximations of the AC power flow equations (see, e.g., [26]). In both cases, model-based approaches require an accurate knowledge of the network topology (including line impedances), and are not dependent on specific operating points of the network [140]. Measurement-based methods leverage data obtained from phase measurement units (PMUs) or Supervisory Control and Data Acquisition (SCADA) systems, to obtain estimates of the sensitivity matrix using, e.g., a least-squares approach or alternative estimation criteria. See, for example, the method proposed in [45] to compute sensitivity distribution factors.

Notably, measurement-based methods do not require a knowledge of the topology and impedances, and they do not rely on pseudo-measurements obtained via power flow solutions.

Approaches based on the least-squares estimation criterion are effective only if one can collect measurements of the net power injections that are "sufficiently rich"; that is, measurements that lead to a regression matrix that is full column rank [45]. In principle, the regression matrix may have a full column rank when the perturbations of the net power injections can be properly designed by the grid operator; for example, by adopting the probing techniques of [22] at some of the nodes or all the nodes. However, an under-determined system may emerge when *(i)* perturbations may not be performed at a sufficient number of nodes (and, thus, the variations of powers are simply due to uncontrollable devices); *(ii)* changes in the power of uncontrollable loads and generation units located throughout the network may lead to correlated measurements [1]; and, *(iii)* when the power network is operating under dynamic conditions due to fluctuations introduced by intermittent renewable generation and uncontrollable loads [159], the operator may not have time to collect enough measurements before the operating point of the network changes (and, thus, the sensitivities change). To address these challenges, this work proposes a robust nuclear norm minimization method [36, 111] to estimate sensitivities from measurements, along with an online algorithm to solve the nuclear norm minimization method with streaming measurements. The proposed approach is motivated by our observation that certain classes of sensitivity matrices can afford a low-rank approximation. For example:

• Figure 6.1 shows the singular values of the PTDF matrices for three different transmission networks; it can be seen that the PTDF matrix can be approximated by a low-rank matrix where only the dominant singular values are retained.

• Figure 6.2 shows the singular values of the sensitivity matrix for the voltage magnitudes with respect to the net injected powers for two different distribution networks; in particular, a real feeder from California was utilized (the feeder has 126 multi-phase nodes, with a total of 366 single-phase points of connection, as described in [20]).

Relative to existing methods based on the least-squares approach, the proposed method: *(i)*

obtains meaningful estimates of the sensitivity matrices with a smaller number of measurements and when the regression model is underdetermined (this is particularly important in time-varying conditions and in case of switches in the topology or switchgear); *(ii)* by leveraging sparsity-promoting regularization functions, the proposed estimator can identify faulty measurements; and, *(iii)* the low-rank approach can handle missing data and asynchronous measurements. The proposed approach can be used to estimate various sensitivity coefficients in a power grid; for example, sensitivity injection shift factors [129, 45, 44] in transmission systems and voltage sensitivities (with respect to power injections) in distribution networks [136]. In the remainder of the chapter, in order to concretely explain the proposed approach, we tailor the exposition to the estimation of the PTDFs.

To adapt to power networks increasingly operating under dynamic conditions (and, hence, having sensitivity matrices that change rapidly over time), the development of real-time algorithms that can estimate the sensitivity matrix on-the-fly from streaming measurements is presented in this chapter. In particular, this work proposes an online proximal-gradient method [52] to solve the nuclear norm minimization problem based on measurements collected from PMUs and SCADA systems at the second or sub-second level. In par with the broad literature on online optimization, convergence results in terms of dynamic regret [72, 80] are offered in this case. We point out that the proposed algorithm is markedly different from the competing alternative [5], and relies on an online proximal-gradient method.

Lastly, it is also worth recognizing related works such as [44], where the AC equations are perturbed in order to derive a closed-form expression of so-called "generalized" injection shift factors. An approach to estimate dynamic distribution factors is introduced in [7], where reduced-order models are used to derive dynamic injection shift factors and generator participation factors. An example of online convex optimization in power systems in presented in [6], for the specific application of estimating load changes in the network.

The remainder of the chapter is organized as follows. Section 6.1 describes the system model and the existing methods used to calculate the PTDF matrix. Section 6.2 presents the proposed

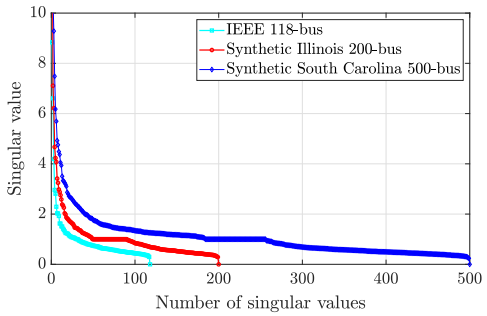


Figure 6.1: Singular values (ordered in decreasing order) of the power transfer distribution factors matrix for three different transmission networks.

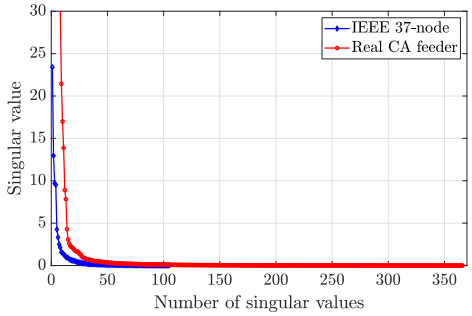


Figure 6.2: Singular values (ordered in decreasing order) of power-voltage magnitude sensitivity matrix for two different multi-phase distribution networks.

low-rank approach. The proposed data-driven online estimation method is presented in Section 6.3. Test cases for transmission and distribution are provided in Section 6.4, and finally, Section 6.5 concludes the chapter.

6.1 Preliminaries

As mentioned in the previous section, the proposed approach can be leveraged to estimate various sensitivity coefficients in a power grid. These include, for example, sensitivity injection shift factors [129, 45] in transmission systems, and voltage sensitivities (with respect to power injections) in distribution networks [136]. In the following, to clearly and concretely explain the proposed approach, we tailor the exposition to the estimation of the power transfer distribution factors matrix.

6.1.1 System Model

Let $\mathcal{N} := \{1, \ldots, n\}$ be the set of nodes where generators and/or loads are located, and let $\mathcal{L} := \{1, \ldots, l\}$ be the set of transmission or distribution lines (or branches). Towards this, let $\Delta p_j \in \mathbb{R}$ represent a change in the net active power injection at node $j \in \mathcal{N}$, around a given point p_j; then, the vector capturing the change in the active power flow on the lines in response to the change of power Δp_j can be approximated as $\mathbf{h}_j \Delta p_j$, where $\mathbf{h}_j \in \mathbb{R}^l$ represent the sensitivity coefficients [129, 45].

Discretize the temporal axis as $\{t_k = kT, k \in \mathbb{N}\}$, with T as a given time interval. Let $\Delta \mathbf{p}_k := [\Delta p_{1k}, \Delta p_{2k}, \ldots, \Delta p_{nk}]^\top$ be the vector of net active power changes collected at time instant t_k at the n nodes, and define the sensitivity matrix as $\mathbf{H}_k := [\mathbf{h}_{1k}\ \mathbf{h}_{2k}\ \ldots\ \mathbf{h}_{nk}] \in \mathbb{R}^{l \times n}$. Then, the vector $\Delta \mathbf{f}_k \in \mathbb{R}^l$ representing the change in the power flow on the lines in the network due to $\Delta \mathbf{p}_k$ can be expressed by [129, 45]

$$\Delta \mathbf{f}_k = \mathbf{H}_k \Delta \mathbf{p}_k, \tag{6.1}$$

where the entry i, j of $\mathbf{H}_k$ represents the sensitivity injection shift factors [45]. Overall, $\mathbf{H}_k$ can be thought as a proxy for the Jacobian of the map $\mathbf{f} = \mathcal{F}(\mathbf{p})$, which yields flows as a function of power injections, calculated at a given point.

By considering m measurements, we can define the matrices $\Delta \mathbf{F}_k = [\Delta \mathbf{f}_{k-m+1}\ \ldots\ \Delta \mathbf{f}_k] \in \mathbb{R}^{l \times m}$, and $\Delta \mathbf{P}_k = [\Delta \mathbf{p}_{k-m+1}\ \ldots\ \Delta \mathbf{p}_k] \in \mathbb{R}^{n \times m}$. Then, the following linear system of equations can be written as

$$\Delta \mathbf{F}_k = \mathbf{H}_k \Delta \mathbf{P}_k. \tag{6.2}$$

Based on (6.2), the following subsection will review existing approaches based on the least-squares method as well as model-based approaches.

Remark 7 To estimate voltage sensitivities in distribution networks, a model similar to (6.2) can be adopted; that is, $\Delta \mathbf{V}_k = \mathbf{H}_k \Delta \mathbf{P}_k$, where $\Delta \mathbf{V}_k$ is a matrix containing measurements of the voltage deviations in response to changes in the net power injections at the nodes. Relevant models for the estimation of the Jacobian matrix can be found in [46].

6.1.2 Existing Methods

6.1.2.1 Least-squares estimation

Assuming that $\Delta\mathbf{P}_k$ is known and measurements (or pseudo-measurements) of $\Delta\mathbf{F}_k$ are available, one possible way to estimate $\mathbf{H}_k$ is via a least-squares criterion. For example, a method similar to [45] can be used, where the injection shift factors for a branch were estimated using PMU measurements obtained in (near) real-time. In particular, borrowing the approach of [45], $\mathbf{H}_k$ can be obtained at time t_k by solving:

$$\mathbf{H}_{LS,k} \in \arg\min_{\mathbf{H}\in\mathcal{H}} \|\Delta\mathbf{F}_k - \mathbf{H}\Delta\mathbf{P}_k\|_F^2 \tag{6.3}$$

where $\|\cdot\|_F$ denotes the Frobenious norm, and $\mathcal{H}$ is a compact set ensuring that each entry (i,j) of the matrix $\mathbf{H}$ satisfies the constraint $h_{min} \leq [\mathbf{H}]_{ij} \leq h_{max}$; that is, $\mathcal{H} = [h_{min}, h_{max}]^{ln}$ (in this case, $h_{min} = -1$ and $h_{max} = 1$). Alternatively, a weighted least-squares method can be utilized when the noise affecting $\Delta\mathbf{F}_k$ is colored or it is not identically distributed across lines. Notice that in (6.3) there are lm measurements and ln unknowns. With this in mind, existing works such as [45] generally assume that $m \geq n$ and that, the matrix $\Delta\mathbf{P}_k$ has a full column rank; with these assumptions, one avoids an underdetermined system and, furthermore, (6.3) has a unique solution. In principle, the matrix $\Delta\mathbf{P}_k$ can have a full column rank when the perturbations $\{\Delta p_{jk}\}$ can be properly designed by the grid operator [22], or when nodes are perturbed in a round-robin fashion. However, this is impractical in a realistic setting (if not infeasible), because the grid operator may not have access to controllable devices at each node of the network; moreover, changes in the power of uncontrollable loads and generation units located throughout the network contribute to $\{\Delta p_{jk}\}$, and this may lead to correlated measurements (therefore, system (6.2) becomes underdetermined). In an underdetermined setting, only a minimum-norm solution would be available using a least-squares criterion, which may provide inaccurate estimates of $\mathbf{H}$ (as corroborated in the numerical results in Section 6.4).

Before presenting the proposed method, we briefly mention a model-based approach.

6.1.2.2 Model-based method

For transmission systems a widely-used model-based approach to calculate the linear sensitivity distribution factor is based on the DC approximation [140]. In particular, by letting $\mathbf{B} \in \mathbb{R}^{n\times n}$ represent the matrix of line series susceptances of the transmission system, one can calculate the changes in the phase angles $\Delta\boldsymbol{\theta}$ by using the following relation:

$$\Delta\mathbf{p}_k = \mathbf{B}\Delta\boldsymbol{\theta}_k. \tag{6.4}$$

Define $\mathbf{X} = \text{diag}(\{-x_{ab}\}) \in \mathbb{R}^{l\times l}$, where x_{ab} represents the line reactance between node a and $b \in \mathbb{N}$, and let $\mathbf{A} \in \mathbb{R}^{l\times n}$ be the branch-bus incidence matrix. Then, using the DC power flow formulation, $\mathbf{f}_k$ can be expressed as the linear relation $\mathbf{f}_k = \mathbf{X}^{-1}\mathbf{A}\boldsymbol{\theta}_k$ [172]. If we want to express the active power flow perturbation $\Delta\mathbf{f}_k$ due to a change in the phase angles $\Delta\boldsymbol{\theta}_k$, we can write $\Delta\mathbf{f}_k$ as $\Delta\mathbf{f}_k = \mathbf{X}^{-1}\mathbf{A}\Delta\boldsymbol{\theta}_k$. By replacing this equation and (6.4) in (6.1) we obtain the model-based relation for the sensitivity matrix as: $\mathbf{H} = \mathbf{X}^{-1}\mathbf{A}\mathbf{B}^{-1}$.

In order to guarantee that the inverse of $\mathbf{B}$ exists, we require that the DC power flow equations for the nodal power balances are linearly independent. Then, taking the node 1 as the slack bus, and denoting as $\mathbf{B}_r \in \mathbb{R}^{(n-1)\times(n-1)}$ and $\mathbf{A}_r \in \mathbb{R}^{l\times(n-1)}$ the reduced matrices, the final sensitivity matrix is given by $\mathbf{H} = [\mathbf{0} \quad \mathbf{X}^{-1}\mathbf{A}_r\mathbf{B}_r^{-1}]$.

With the DC formulation, the sensitivity matrix factors depends only on the topology of the network, and are invariant to changes in the system operation point, such as line outages, load and generation perturbations, etc. At the distribution level, the model-based approach could be used based on, e.g., [26]. In particular, a linearization of the power-flow equations (around a point) is proposed in [26], with the linear coefficients inherently representing the sensitivities. Another possible approach is the one in [3]. In the following, a low-rank method will be presented, which does not require knowledge of network topology or reactances.

6.2 Low-Rank Approach

In this section, we present an approach for the estimation of the matrix $\mathbf{H}$ with the following features: *(i)* it leverages measurements of $\Delta\mathbf{F}_k$ and $\Delta\mathbf{P}_k$ obtained from phasor measurement units (PMUs), supervisory control and data acquisition (SCADA), or other similar sources, rather than relying on a network model; *(ii)* it allows for obtaining meaningful estimates of $\mathbf{H}$ even when (6.2) is underdetermined by leveraging a low-rank approximation of $\mathbf{H}$; *(iii)* when $\Delta\mathbf{P}_k$ is full column rank, it yields an estimation accuracy similar to the least-squares estimator and, *(iv)* can handle missing measurements of flows on some lines (i.e., some entries of $\Delta\mathbf{F}_k$ may be missing).

For simplicity of exposition, we first consider the case where measurements are error-free (then, we consider noisy measurements as well as measurement outliers). Based on the model (6.2), the nearly low-rank property of $\mathbf{H}$ motivates us to consider the following affine rank minimization problem (RMP) [135]:

$$\min_{\mathbf{H}\in\mathcal{H}} \quad \operatorname{rank}(\mathbf{H}) \tag{6.5a}$$

$$\text{s.t. } \operatorname{vec}(\Delta\mathbf{F}_k) = \mathcal{A}(\mathbf{H}), \tag{6.5b}$$

where $\mathcal{H}$ is the convex compact set (in the simplest case, the Cartesian product of box constraints), $\operatorname{vec}(\Delta\mathbf{F}_k) \in \mathbb{R}^p$ where $p := lm$, denotes the vectorized $\Delta\mathbf{F}_k$, and the linear map $\mathcal{A} : \mathbb{R}^{l\times n} \to \mathbb{R}^p$ is defined as: $\mathcal{A}(\mathbf{H}) = \mathbf{A}_{\mathbf{P},k} \operatorname{vec}(\mathbf{H})$, where $\operatorname{vec}(\mathbf{H}) \in \mathbb{R}^d$, $d := ln$, and $\mathbf{A}_{\mathbf{P},k}$ is a matrix of dimensions $p \times d$, appropriately built using the perturbations $\Delta\mathbf{P}_k$. Specifically, matrix $\mathbf{A}_{\mathbf{P},k}$ is the Kronecker product defined by $\mathbf{A}_{\mathbf{P},k} := \Delta\mathbf{P}_k^\top \otimes \mathbf{I}$, where $\mathbf{I}$ is the identity matrix of dimensions $l \times l$.

Unfortunately, the rank criterion in (6.5) is in general NP-hard to optimize; nevertheless, drawing an analogy from compressed sensing to rank minimization, the following convex relaxation of the RMP (6.5) can be utilized [135]:

$$\min_{\mathbf{H}\in\mathcal{H}} \|\mathbf{H}\|_* \tag{6.6a}$$

$$\text{s.t. } \operatorname{vec}(\Delta\mathbf{F}_k) = \mathbf{A}_{\mathbf{P},k} \operatorname{vec}(\mathbf{H}) \tag{6.6b}$$

where $\|\mathbf{H}\|_* := \sum_i \sigma_i(\mathbf{H})$ is the nuclear norm of $\mathbf{H}$, with $\sigma_i(\mathbf{H})$ denoting the ith singular value of $\mathbf{H}$. Interestingly, it was shown in [135] that, if the constraints of (6.6) are defined by a linear transformation that satisfies a restricted isometry property condition, the minimum rank solution can be recovered by the minimization of the nuclear norm over the linear space; see the necessary and sufficient condition in [135]. In terms of optimility conditions, notice that a matrix $\mathbf{H}$ is an optimal solution for (6.6) if there exists a vector $z \in \mathbb{R}^p$ such that $\mathcal{A}^*(z) \in \partial\|\mathbf{H}\|_*$ [135], where $\mathcal{A}^* : \mathbb{R}^p \to \mathbb{R}^{l\times n}$ is the adjoint of $\mathcal{A}$.

The proposed methodology leverages the relaxation (6.6) to estimate the matrix $\mathbf{H}$ from measurements of $\Delta\mathbf{F}_k$ induced by the perturbations in the net power injections $\Delta\mathbf{P}_k$. Assuming that the measurements $\Delta\mathbf{F}_k$ are affected by a zero-mean Gaussian noise (instead of being noise-free), a pertinent relaxation of (6.6) amounts to the following convex program [36]:

$$\min_{\mathbf{H}\in\mathcal{H}} \|\text{vec}(\Delta\mathbf{F}_k) - \mathbf{A}_{\mathbf{P},k}\,\text{vec}(\mathbf{H})\|_2^2 + \lambda\|\mathbf{H}\|_* \tag{6.7}$$

where $\lambda > 0$ is a given regularization parameter that is used to promote sparsity in the singular values of $\mathbf{H}$ (and, hence, to obtain a low-rank matrix $\mathbf{H}$).

Robustness to outliers: We further consider the case where some measurements of $\Delta\mathbf{F}_k$ may be corrupted by outliers. This can be due to, for example, faulty readings of PMUs and micro PMUs, communication errors, or malicious attacks. To this end, we augment the model (6.2) as $\Delta\mathbf{F}_k = \mathbf{H}_k\Delta\mathbf{P}_k + \mathbf{O}_k + \mathbf{E}_k$, where $\mathbf{E}_k$ is a matrix containing (small) measurement errors and $\mathbf{O}_k$ is a matrix containing measurement outliers [185, 111]. When no outliers are present, $\mathbf{O}_k$ is a matrix with all zeros. Based on this augmented model, estimates of $\mathbf{H}_k$ and $\mathbf{O}_k$ can be sought by solving the following convex problem [185, 111]:

$$\min_{\mathbf{H}\in\mathcal{H},\mathbf{O}\in\mathcal{M}} \|\text{vec}(\Delta\mathbf{F}_k) - \mathbf{A}_{\mathbf{P},k}\,\text{vec}(\mathbf{H}) - \text{vec}(\mathbf{O})\|_2^2 + \lambda\|\mathbf{H}\|_* + \gamma\|\text{vec}(\mathbf{O})\|_1, \tag{6.8}$$

where $\|\text{vec}(\mathbf{O})\|_1 = \sum_i |[\text{vec}(\mathbf{O})]_i|$ is the ℓ_1-norm of the vector $\text{vec}(\mathbf{O})$, $\gamma > 0$ is a sparsity-promoting coefficient, and $\mathcal{M}$ are box constraints of the form $O_{min} \le [\mathbf{O}_k]_{ij} \le O_{max}$. Notice that the ℓ_1-norm is the closest convex surrogates to the cardinality function. Once (6.8) is solved, the

locations of nonzero entries in $\mathbf{O}$ reveal outliers across both lines and time; on the other hand, the amplitudes quantify the magnitude of the anomalous measurement. It is important to notice that the parameters λ and γ control the tradeoff between fitting error, rank of $\mathbf{H}$, and sparsity level of $\mathbf{O}$; in particular, when an estimate of the variance of the measurement noise is available, one can follow guidelines for selection of λ and γ similar to the ones proposed in [185].

Missing and asynchronous measurements: It is worth pointing out that the proposed methodology is applicable to the case where some measurements in $\Delta\mathbf{F}_k$ in (6.7) are *missing*. This may be due to communication failures or because measurements are collected at different rates (i.e., they are asynchronous). For the latter, one can take the highest measurement frequency (i.e., the T is smallest inter-arrival time) as a reference frame, and treat measurements that are received less frequently (i.e., with a larger inter-arrival time) with missing entries.

In this case, missing measurements are discarded from the least-squares term in (6.8) [111]. In particular, let $\Omega_k \subseteq \{1, 2, \ldots, p\}$ be a set indicating which measurements in the vector $\text{vec}(\Delta\mathbf{F}_k)$ are available at time t_k; for example, if the measurement for the line 1 is missing at time t_{k-m+1} and t_{k-m+2}, then $\Omega_k \subseteq \{2, 3, \ldots, l, l+2, \ldots, p\}$. Let $\mathcal{P}_{\Omega_k}$ be a time-varying vector sampling operator, which sets the entries of its vector argument not indexed by Ω_k to zero and leaves the other entries unchanged. Then, (6.8) can be reformulated as:

$$\min_{\mathbf{H}\in\mathcal{H},\mathbf{O}\in\mathcal{M}} \|\mathcal{P}_{\Omega_k}\{\text{vec}(\Delta\mathbf{F}_k) - \mathbf{A}_{\mathbf{P},k}\,\text{vec}(\mathbf{H}) - \text{vec}(\mathbf{O})\}\|_2^2 + \lambda\|\mathbf{H}\|_* + \gamma\|\text{vec}(\mathbf{O})\|_1, \tag{6.9}$$

where, of course, missing measurements are not accounted for in the least-squares term.

Remark 8 As an example of another application, to estimate voltage sensitivities in distribution networks, the approach (6.8) can be used based on the model $\Delta\mathbf{V}_k = \mathbf{H}_k\Delta\mathbf{P}_k$, where $\Delta\mathbf{V}_k$ is a matrix containing measurements of the voltage deviations in response to changes in the net power injections at the nodes.

Remark 9 In a multi-phase unbalanced system, the powers measured at each phases of nodes with wye connections and nodes with delta connections are used to form the regression matrix

$\mathbf{\Delta P}_k$ (and, thus, $\mathbf{A}_{\mathbf{P},k}$); flows on each of the phases of lines are utilized to form the measurement matrix $\Delta\mathbf{F}_k$; or, voltage-to-ground measurements at each phase or a node are used to form the matrix $\Delta\mathbf{V}_k$ when voltage sensitivities are to be estimated. The mathematical structure of the problem as well as the solution approach do not change when multi-phase unbalanced system are considered.

Remark 10 The proposed method relies on measurements of the net injected powers at the nodes, and it does not utilize a model of the network and loads; accordingly, the so-called ZIP model is not considered. How to consider ZIP models is outside the scope of this work.

6.3 Data-Driven Online Estimation Method

Based on $\Delta\mathbf{P}_k$ and $\Delta\mathbf{F}_k$, which collect measurements of new power injections and power flows acquired at time steps $k-m+1,\ldots,k$, an estimate of $\mathbf{H}_k$ can be obtained by solving the convex problem (6.8) using existing batch solvers for non-smooth convex optimization problems. When the power network is operating under dynamic conditions, for example, due to swings in the net power due to intermittent renewable generation and uncontrollable loads [159], the sensitivity matrix $\mathbf{H}_k$ may rapidly change over time (since, in general, it depends on the current operating points [45]); in these dynamic conditions, it may not be possible to solve (6.8) sufficiently fast due to underlying computational complexity considerations, and a solution of (6.8) generated by batch solvers can be outdated. That is, by the time the solution is produced, the operating conditions of the network (and, hence, $\mathbf{H}_k$) have changed. Furthermore, batch solvers may not fully take advantage of high-speed measurements collected by PMUs and the SCADA system, since their solution time may be larger than the inter-arrival of the measurements. This aspect motivates the development of an online algorithm that estimates $\mathbf{H}_k$ based on streams of measurements and identifies outliers "on the fly," as explained in this section.

Measurements are assumed to arrive at times $\{t_k = kT, k \in \mathbb{N}\}$, with T the inter-arrival time (e.g, T could be one second or a few seconds [159]); suppose further that measurements are

processed over a sliding window $\mathcal{T}_k = \{t_{k-m+1}, \ldots, t_k\}$. Then, at each instant t_k, the matrix $\mathbf{H}_k$ can be estimated via (6.8), which is re-written here as the following time-varying problem [52]:

$$(\mathbf{H}_k^*, \mathbf{O}_k^*) \in \arg \min_{\mathbf{H}_k \in \mathbb{R}^{l\times n}, \mathbf{O}_k \in \mathbb{R}^{l\times m}} f_k(\mathbf{H}_k, \mathbf{O}_k), \quad \forall\, kT \tag{6.10a}$$

where $f_k(\mathbf{H}_k, \mathbf{O}_k) := s_k(\mathbf{H}_k, \mathbf{O}_k) + g_k(\mathbf{H}_k, \mathbf{O}_k)$,

$$s_k(\mathbf{H}_k, \mathbf{O}_k) := \|\Delta\mathbf{F}_k - \mathbf{H}_k\Delta\mathbf{P}_k - \mathbf{O}_k\|_F^2. \tag{6.10b}$$

$$g_k(\mathbf{H}_k, \mathbf{O}_k) := \lambda_k\|\mathbf{H}_k\|_* + \gamma_k\|\mathrm{vec}(\mathbf{O}_k)\|_1 + \iota_{\mathcal{H}}(\mathbf{H}_k) + \iota_{\mathcal{M}}(\mathbf{O}_k), \tag{6.10c}$$

with $\iota_{\mathcal{H}}(\mathbf{H})$ the set indicator function for the compact set $\mathcal{H}$ and $\iota_{\mathcal{M}}(\mathbf{O})$ the set indicator function for the compact set $\mathcal{M}$. The goal posed here is to develop an online algorithm that can track a solution $\{\mathbf{H}_k^*, \mathbf{O}_k^*\}_{k\in\mathbb{N}}$ and the trajectory of optimal value functions $\{f_k^* := f_k(\mathbf{H}_k^*, \mathbf{O}_k^*)\}_{k\in\mathbb{N}}$ by processing measurements in a sliding window fashion. In the following, let $\mathbf{o}_k = \mathrm{vec}(\mathbf{O}_k)$, and $\mathbf{x}_k = [\mathrm{vec}(\mathbf{H}_k)^\top, \mathbf{o}_k^\top]^\top \in \mathcal{X}_k := \mathcal{H} \times \mathcal{M}$ for brevity. Notice that $s_k(\mathbf{x}_k)$ is closed, convex and proper, with a L_k-Lipschitz continuous gradient at each time t_k; on the other hand, $g_k(\mathbf{x}_k)$ is a lower semi-continuous proper convex function. Lastly, the function attains a finite minimum at a certain $\mathbf{x}_k^*$. Given this particular structure of (6.10), we propose to use an online proximal-gradient algorithm [52] to solve (6.10) under streams of measurements. Assuming that, because of communication delays and computational considerations, one step of the algorithm can be performed within an interval T (which coincides with the inter-arrival rate of the measurements), the online proximal-gradient algorithm amounts to the sequential execution of the following step:

$$\mathbf{y}_k = \mathbf{x}_{k-1} - \alpha\nabla_{\mathbf{x}} s_k(\mathbf{x}_{k-1}) \tag{6.11a}$$

$$\mathbf{x}_k = \mathrm{prox}_{g_k,\mathcal{X}}^{\alpha}\{\mathbf{y}_k\}, \tag{6.11b}$$

where $\alpha > 0$ is the stepsize, and the proximal operator is defined over the non-differentiable function g_k as [15]

$$\mathrm{prox}_g^{\alpha}\{\mathbf{y}\} := \arg\min_{\mathbf{x}}\Big\{g(\mathbf{x}) + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{y}\|^2\Big\}. \tag{6.12}$$

Notice that, if we re-write the function s_k as

$$s_k(\mathbf{x}_k) = \|\Delta\mathbf{f}_k - \mathbf{A}_{\mathbf{Ps},k}\mathbf{x}_k\|^2, \tag{6.13}$$

where $\Delta\mathbf{f}_k = \text{vec}(\Delta\mathbf{F}_k)$, $\mathbf{A}_{\mathbf{Ps},k} = [\mathbf{A}_{\mathbf{P},k}, \mathbf{I}]$, and $\mathbf{x}_k$ defined as before, then, $\nabla_{\mathbf{x}} s_k$ is given by

$$\nabla_{\mathbf{x}} s_k(\mathbf{x}_k) = 2\mathbf{A}_{\mathbf{Ps},k}^{\top}\left(\mathbf{A}_{\mathbf{Ps},k}\mathbf{x}_k - \Delta\mathbf{f}_k\right). \tag{6.14}$$

Furthermore, one can notice that the proximal operator in (6.11b) is separable across the two variables of interest $\mathbf{H}_k$ and $\mathbf{o}_k$, which can therefore be computed separately. In particular, one has that:

$$\mathbf{H}_k = \text{prox}_{\lambda_k\|\cdot\|_*+\iota_{\mathcal{H}}}\{\mathbf{Y}_{H,k}\} \tag{6.15}$$

$$\mathbf{o}_k = \text{prox}_{\gamma_k\|\cdot\|_1+\iota_{\mathcal{M}}}\{\mathbf{y}_{o,k}\} \tag{6.16}$$

with $\mathbf{Y}_{H,k}$ and $\mathbf{y}_{o,k}$ extracted from the stacked vector $\mathbf{y}_k$ in (6.11a). Moreover, (6.16) admits a closed-form solution, which is given by:

$$\mathbf{o}_k = [\mathcal{S}_\gamma(\mathbf{y}_{o,k})]_{o_{min}}^{o_{max}} \tag{6.17}$$

where $[x]_a^b = \max\{\min\{x, b\}, a\}$, and the thresholding operator $\mathcal{S}_\gamma$ is defined as:

$$\mathcal{S}_\gamma(\mathbf{y}) = \max\{|\mathbf{y}| - \gamma\mathbf{1}, \mathbf{0}\} \odot \text{sgn}(\mathbf{y}) = \begin{cases} \mathbf{y} - \gamma\mathbf{1}, & \text{if } \mathbf{y} \geq \gamma\mathbf{1}, \\ 0, & \text{if } |\mathbf{y}| < \gamma\mathbf{1}, \\ \mathbf{y} + \gamma\mathbf{1}, & \text{if } \mathbf{y} \leq -\gamma\mathbf{1}. \end{cases} \tag{6.18}$$

With the previous definitions in place, the online proximal-gradient algorithm for the robust estimation of the sensitivity matrix is tabulated as Algorithm 4.

We stress that, by using Algorithm 4, we can estimate the sensitivity matrix robustly over a sliding window $\mathcal{T}_k$, this adapting to changing operational points of the power system. Leveraging the advantages of the low-rank model, an online estimation with less measurements than an online least-squares method is possible.

Algorithm 4: Online robust estimation of sensitivity matrices

Initialize: $\mathbf{x}_0 = (\mathbf{H}_0, \mathbf{O}_0)$, α, $\alpha \in (0, \frac{2}{L})$.
for $k = m, m+1, \ldots,$ **do**
[**S1**] Collect $\Delta\mathbf{f}_k$ and $\Delta\mathbf{p}_k$
[**S2**] Build $\Delta\mathbf{F}_k$ and $\Delta\mathbf{P}_k$ based on $\{\Delta\mathbf{f}_k, \Delta\mathbf{p}_k\}_{k\in\mathcal{T}_k}$
[**S3**] Compute $\mathbf{y}_k$ via (6.11a)
[**S4**] Update $\mathbf{H}_k$ via (6.15)
[**S5**] Update $\mathbf{o}_k$ via (6.17)
end for

6.3.1 Performance Analysis

In order to analyze the estimation accuracy of the Algorithm 4 performance, the dynamic regret metric is considered here; see, e.g. [72, 80, 52]. In particular, it is defined as:

$$\mathrm{Reg}_k := \sum_{i=1}^{k} \left[f_i(\mathbf{x}_i) - f_i(\mathbf{x}_i^*) \right],$$

where we recall the f_i is the cost function in (6.8) (see also (6.10)). The dynamic regret is an appropriate performance metric for time-varying problems with a cost that is convex, but not necessarily strongly convex [52]. To derive bounds on the dynamic regret, it is first necessary to introduce a "measure" of the temporal variability of (6.10). One possible measure is:

$$\omega_k := \|\mathbf{x}_k^* - \mathbf{x}_{k-1}^*\| \tag{6.19}$$

along with the so-called "path length":

$$\Omega_k := \sum_{i=1}^{k} \omega_i, \qquad \bar{\Omega}_k := \sum_{i=1}^{k} \omega_i^2. \tag{6.20}$$

Recall that the least-squares term $s_k(\mathbf{x}_k)$ is closed, convex and proper, with a L_k-Lipschitz continuous gradient at each time t_k, and that $g_k(\mathbf{x}_k)$ is a lower semi-continuous proper convex function. Then, by using the definitions (6.19)–(6.20) and leveraging bounding techniques similar to [4], the following result can be obtained.

Theorem 4 Suppose that the step size α is chosen such that $\alpha \leq 1/L$, with $L := \max\{L_k\}$. Then, the dynamic regret of Algorithm 4 has the following limiting behavior:

$$\frac{1}{k}\mathrm{Reg}_k = \mathcal{O}(1 + k^{-1}\Omega_k + k^{-1}\bar{\Omega}_k)\,. \tag{6.21}$$

Proof of Theorem 4. Since s_k has a L_k-Lipschitz continuous gradient, then:

$$s_k(\mathbf{x}_k) \leq s_k(\mathbf{x}_{k-1}) + \langle \nabla_{\mathbf{x}} s_k(\mathbf{x}_{k-1}), \mathbf{x}_k - \mathbf{x}_{k-1} \rangle + \frac{L_k}{2} \|\mathbf{x}_k - \mathbf{x}_{k-1}\|^2, \tag{6.22}$$

where $L_k \geq \|\mathbf{A}_{\mathbf{Ps},k}^{\top} \mathbf{A}_{\mathbf{Ps},k}\|$. By using the convexity of s_k we also have that

$$s_k(\mathbf{x}_{k-1}) \leq s_k(\mathbf{x}_k^*) + \langle \nabla_{\mathbf{x}} s_k(\mathbf{x}_{k-1}), \mathbf{x}_{k-1} - \mathbf{x}_k^* \rangle. \tag{6.23}$$

Therefore, putting (6.22) and (6.23) together, one arrives at:

$$s_k(\mathbf{x}_k) \leq s_k(\mathbf{x}_k^*) + \langle \nabla_{\mathbf{x}} s_k(\mathbf{x}_{k-1}), \mathbf{x}_k - \mathbf{x}_k^* \rangle + \frac{L_k}{2} \|\mathbf{x}_k - \mathbf{x}_{k-1}\|^2. \tag{6.24}$$

On the other hand, for the non-differentiable function g_k, we can leverage [15, Theorem 3.36]. Let $\phi_1(\mathbf{z})$, $\phi_2(\mathbf{z})$: $\mathbb{E} \to (-\infty, \infty]$ be proper convex functions, and let $\mathbf{z} \in \text{int}(\text{dom}(\phi_1)) \cap \text{int}(\text{dom}(\phi_2))$. For $\phi(\mathbf{z}) := \phi_1(\mathbf{z}) + \phi_2(\mathbf{z})$, then $\partial\phi(\mathbf{z}) = \partial\phi_1(\mathbf{z}) + \partial\phi_2(\mathbf{z})$. Based on [15, Theorem 3.36], we have that

$$0 \in \partial\Big\{ g_k(\mathbf{x}_k) + \frac{1}{2\alpha} \|\mathbf{x}_k - \mathbf{y}_k\|^2 \Big\} = \partial g_k(\mathbf{x}_k) + \partial\Big\{ \frac{1}{2\alpha} \|\mathbf{x}_k - \mathbf{y}_k\|^2 \Big\},$$

which implies:

$$-\boldsymbol{\varphi} \in \partial g_k(\mathbf{x}_k), \qquad \boldsymbol{\varphi} \in \partial\Big\{ \frac{1}{2\alpha} \|\mathbf{x}_k - \mathbf{y}_k\|^2 \Big\}.$$

Since $\boldsymbol{\varphi} \in \partial\Big\{ \frac{1}{2\alpha} \|\mathbf{x}_k - \mathbf{y}_k\|^2 \Big\}$, the following holds:

$$\frac{1}{2\alpha} \|\mathbf{x}_k - \mathbf{y}_k\|^2 + \langle \boldsymbol{\varphi}, \mathbf{q} - \mathbf{x}_k \rangle - \frac{1}{2\alpha} \|\mathbf{q} - \mathbf{y}_k\|^2 \leq 0 \quad \forall\, \mathbf{q}. \tag{6.25}$$

Furthermore, since (6.25) holds for all $\mathbf{q}$, we can define $\mathbf{q} = \mathbf{y}_k + \alpha\boldsymbol{\varphi}$. Then, (6.25) can be written as

$$\frac{1}{2\alpha} (\|\mathbf{x}_k - \mathbf{y}_k\|^2 + \alpha^2 \|\boldsymbol{\varphi}\|^2) - \langle \boldsymbol{\varphi}, \mathbf{x}_k - \mathbf{y}_k \rangle \leq 0. \tag{6.26}$$

Now using (6.25) and (6.26) we get that,

$$\frac{\mathbf{y}_k - \mathbf{q}}{\alpha} \in \partial g_k(\mathbf{x}_k). \tag{6.27}$$

By using the subgradient defined in (6.27) and (6.11), the following inequality for g_k can be obtained,

$$g_k(\mathbf{x}_k) \leq g_k(\mathbf{x}_k^*) - \frac{1}{\alpha} \langle \mathbf{x}_{k-1} - \mathbf{q}, \mathbf{x}_k^* - \mathbf{x}_k \rangle + \langle \nabla_{\mathbf{x}} s_k(\mathbf{x}_{k-1}), \mathbf{x}_k^* - \mathbf{x}_k \rangle. \tag{6.28}$$

Adding (6.24) and (6.28) we obtained

$$s_k(\mathbf{x}_k) + g_k(\mathbf{x}_k) \leq s_k(\mathbf{x}_k^*) + g_k(\mathbf{x}_k^*) + \frac{L_k}{2}\|\mathbf{x}_k - \mathbf{x}_{k-1}\|^2 + \langle \nabla_{\mathbf{x}} s_k(\mathbf{x}_{k-1}), \mathbf{x}_k^* - \mathbf{x}_k \rangle + \langle \nabla_{\mathbf{x}} s_k(\mathbf{x}_{k-1}), \mathbf{x}_k - \mathbf{x}_k^* \rangle + \frac{1}{\alpha}\langle \mathbf{x}_{k-1} - \mathbf{q}, \mathbf{x}_k - \mathbf{x}_k^* \rangle,$$

and therefore,

$$f_k(\mathbf{x}_k) \leq f_k(\mathbf{x}_k^*) + \left(\frac{L_k}{2} - \frac{1}{\alpha}\right)\|\mathbf{x}_k - \mathbf{x}_k^*\|^2 + \frac{L_k}{2}\|\mathbf{x}_{k-1} - \mathbf{x}_k^*\|^2 + \left(\frac{1}{\alpha} - \frac{L_k}{2}\right)\langle \mathbf{x}_{k-1} - \mathbf{q}, \mathbf{x}_k - \mathbf{x}_k^* \rangle. \tag{6.29}$$

Set $\alpha \leq \frac{1}{\max\{L_k\}}$; in particular, let $\alpha = \frac{1}{\max\{L_k\}} - w^2$, for $w \in \mathbb{R}$. Then:

$$\frac{L_k}{2} - \frac{1}{\alpha} \leq \frac{\alpha \max\{L_k\} - 2}{2\alpha} \leq -\frac{1}{2\alpha} - \frac{w^2 \max\{L_k\}}{2\alpha}. \tag{6.30}$$

Also notice that,

$$\left(\frac{1}{\alpha} - \frac{L_k}{2}\right)\langle \mathbf{x}_{k-1} - \mathbf{q}, \mathbf{x}_k - \mathbf{x}_k^* \rangle \leq \left(\frac{1}{\alpha} - \frac{L_k}{2}\right)\|\mathbf{x}_{k-1} - \mathbf{q}\|\|\mathbf{x}_k - \mathbf{x}_k^*\| \leq \left(\frac{1}{\alpha} - \frac{L_k}{2}\right)\|\mathbf{x}_{k-1} - \mathbf{q}\|(\|\mathbf{x}_k - \mathbf{x}_k^*\| + \omega_k).$$

Based on (6.25) holds for all $\mathbf{q}$, let R be the diameter of $\mathcal{X}$. Therefore,

$$\left(\frac{1}{\alpha} - \frac{L_k}{2}\right)\|\mathbf{x}_{k-1} - \mathbf{q}\|(\|\mathbf{x}_k - \mathbf{x}_k^*\| + \omega_k) \leq \Phi R(R + \omega_k), \tag{6.31}$$

where $\Phi := \frac{1}{\alpha} - \frac{\min\{L_k\}}{2}$. Then, based on (6.30) and (6.31), and neglecting constant terms, we can write (6.29) as,

$$f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*) \leq -\frac{1}{2\alpha}\|\mathbf{x}_k - \mathbf{x}_k^*\|^2 + \frac{1}{2\alpha}\|\mathbf{x}_{k-1} - \mathbf{x}_k^*\|^2 \tag{6.32}$$

$$+\Phi R(R + \omega_k). \tag{6.33}$$

By adding and subtracting $\mathbf{x}_{k-1}^*$ in the last term on the right hand side of (6.33), and adding it from $i = 1, \ldots, k$, we can write the right hand side of the equation as:

$$\sum_{i=1}^{k}\left\{-\frac{1}{2\alpha}\|\mathbf{x}_i - \mathbf{x}_i^*\|^2 + \frac{1}{2\alpha}\|\mathbf{x}_{i-1} - \mathbf{x}_{i-1}^* + \mathbf{x}_{i-1}^* - \mathbf{x}_i^*\|^2\right\}, \tag{6.34}$$

where the second term in (6.34) can be expanded as:

$$\sum_{i=1}^{k}\Big\{-\frac{1}{2\alpha}\|\mathbf{x}_i-\mathbf{x}_i^*\|^2+\frac{1}{2\alpha}\|\mathbf{x}_{i-1}-\mathbf{x}_{i-1}^*\|^2+\frac{1}{\alpha}\|\mathbf{x}_{i-1}-\mathbf{x}_{i-1}^*\|\|\mathbf{x}_i^*-\mathbf{x}_{i-1}^*\|+\frac{1}{2\alpha}\|\mathbf{x}_i^*-\mathbf{x}_{i-1}^*\|^2\Big\}. \tag{6.35}$$

The first two terms in (6.35) correspond a telescoping series; and, by using the definition of ω_k, (6.33) can be rewritten as follows:

$$\sum_{i=1}^{k}[f_i(\mathbf{x}_i)-f_i(\mathbf{x}_i^*)]\leq\frac{1}{2\alpha}\|\mathbf{x}_0-\mathbf{x}_0^*\|^2-\frac{1}{2\alpha}\|\mathbf{x}_k-\mathbf{x}_k^*\|^2 + \frac{1}{\alpha}\sum_{i=1}^{k}\omega_i(\|\mathbf{x}_{i-1}-\mathbf{x}_{i-1}^*\|+\Phi R)+\frac{1}{2\alpha}\sum_{i=1}^{k}\omega_i^2+k\Phi R^2. \tag{6.36}$$

Since $\mathcal{X}$ is compact, we can upper bound $\|\mathbf{x}_k-\mathbf{x}_k^*\|$ by R, and thus the result follows.

Note that:

• When the sensitivity matrix changes over time, Ω_k and $\bar{\Omega}_k$ grow as $\mathcal{O}(k)$. Therefore, $(1/k)\text{Reg}_k = \mathcal{O}(1)$; that is, the sensitivity matrix can be estimated within a bounded error even in the considered online setting [52].

• A no-regret result (i.e., $(1/k)\text{Reg}_k$ asymptotically goes to 0) can not be obtained in general.

• If the sensitivity matrix is constant, then one trivially has that Reg_k approaches 0 asymptotically, thus recovering convergence results for the batch proximal-gradient method.

Finally, it is also worth mentioning that (6.15) may be computed inexactly for large matrices. Results similar to Theorem 4 can be obtained in this case.

Remark 11 In the work, we assume that that $\Delta\mathbf{P}_k$ is either known or can be measured with negligible noise; on the other hand, $\Delta\mathbf{F}_k$ is noisy and may contain outliers. In principle, the proposed approach could be extended to handle noise and outliers in $\Delta\mathbf{P}_k$ by replacing the least-squares term with a total least squares (TLS) criterion; see, for example, [110]. However, the resultant cost function is in this case nonconvex (because of bilinear terms); the challenges rely on the model for the trajectories of the critical points of the cost function.

Remark 12 If a measurement unit is not present at one node, our approach may not be able to estimate the sensitivities associated with that node. The proposed method could be extended

to accommodate a kernel-based matrix completion to impute the sensitivities for nodes without measurement units; see, for example, [68].

6.4 Simulation Results

In this section, the estimation accuracy of the proposed methodology is assessed, for both a batch and online implementation. The following cases are considered:
(i) Two transmission networks: Western Electricity Coordinating Council (WECC) 3-machine 9-bus transmission system[186] and the synthetic South Carolina 500-bus transmission power system model [24].
(ii) Two distribution networks: the IEEE-18 buses[186] and the IEEE-69 buses distribution feeders [186]. MATPOWER is utilized to computed power-flow solutions [186].

6.4.1 Batch Estimation

Fluctuations in active power injection around a given operating point are simulated as in [45]. In particular, the injection at node j, denoted by p_j, is given by $p_j[k] = p_j^0[k] + \sigma_{N1} p_j^0[k]\eta_1 + \sigma_{N2}\eta_2$, where $p_j^0[k]$ is the nominal power injection at node j at instant k, and (η_1, η_2) are random values, where $\eta_1 \sim \mathcal{N}(0, \sigma_{N1})$ and $\eta_2 \sim \mathcal{N}(0, \sigma_{N2})$ for standard deviations $\sigma_{N1} = \sigma_{N2} = 0.1$; see [45] for details. For each time k, we take the difference between consecutive line flow measurements to obtain $\Delta\mathbf{F}_k$. We obtained $\Delta\mathbf{P}_k$ by taking the differences between consecutive values of active power injections at each node. The batch optimization problems (6.7) and (6.8) can also be solved efficiently using the proximal-gradient method (i.e., a batch version of Algorithm 4); see, for example, [38] (and references therein) for standard computational times of proximal-gradient methods. It can also be solved using CVX (available at: `http://cvxr.com/cvx`).

Case 1. The performance of the proposed low-rank based approach is considered for both transmission networks; first, the batch method (6.7) is evaluated, when a decrease of generation occurs at generator 2 for the 9-bus and in generator 9 for the 500-bus transmission system. Figure 6.3 and Figure 6.4 compares the performance of the proposed method with the least-squares ap-

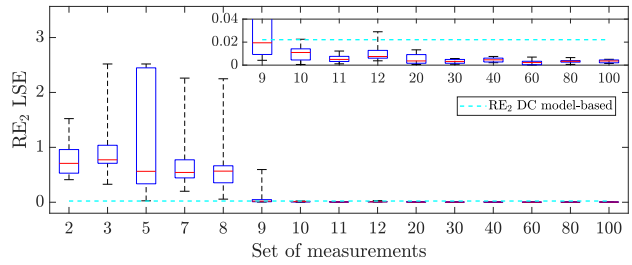
proach [45]. In this case, 10 trials were used, and the relative error (RE) with respect to the bus i is defined by (we use this definition to be consistent with [45])

$$\mathrm{RE}_i = \frac{\|\mathbf{h}_{ik} - \mathbf{h}^*_{ik}\|}{\|\mathbf{h}^*_{ik}\|},$$

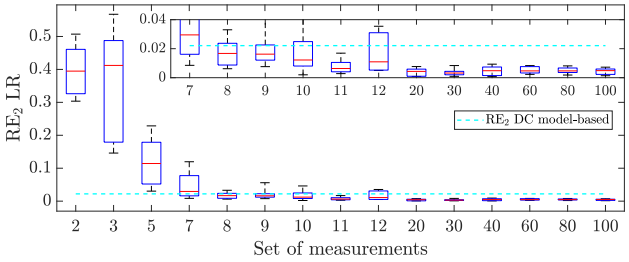
where the actual sensitivity of the lines due the change of generation in bus i is denoted as $\mathbf{h}^*_{ik}$, and $\mathbf{h}_{ik}$ specifies the column of the estimate sensitivity matrix $\mathbf{H}$ for the bus i obtained form the DC model-based, least-squares or low-rank approaches. Figure 6.3 shows that, when the set of measurements is less than 9 (the total number nodes in this case) the least-squares approach does not give an accurate estimation, because it is underdetermined. In the case of the proposed low-rank method, the median of the relative errors can be of just 3% even when we have only 6-7 sets of measurements; the proposed method performs better than the model-based approach via DC approximation once we collect 8 measurements. When more than 9 measurements are collected, the proposed method and the least-squares approach have similar performance as expected. Figure 6.4 shows a similar behavior for the synthetic South Carolina 500-bus transmission systems, where our method is able to estimate the sensitivity matrix from 200 sets of measurements. In both cases, it can be seen that the proposed approach provides accurate results with less measurements than the least-squares approach.

Further, in order to assess the performance of (6.8) in the case of outliers, we replicated the previous case for the 9-bus transmission system but with random outliers in the measurements. Figure 6.5 presents the results for the proposed method and the least-squares approach. Again, the proposed method outperforms the least-squares approach, and provides better estimates than the DC model-based method. In order to assess the performance of (6.9) in the case of missing measurements, we replicated the case for the 9-bus transmission network. Figure 6.6 presents the results for the proposed method and the least-squares approach where the LR method outperforms the LSE approach, in a case when different values of percentages of the data in $\Delta\mathbf{F}_k$ are missing.

Case 2. Two radial distribution feeders are considered, IEEE 18-bus and IEEE 69-bus; again, we first test the batch method (6.7), when an increase in load occurs at bus 5 for the 18-

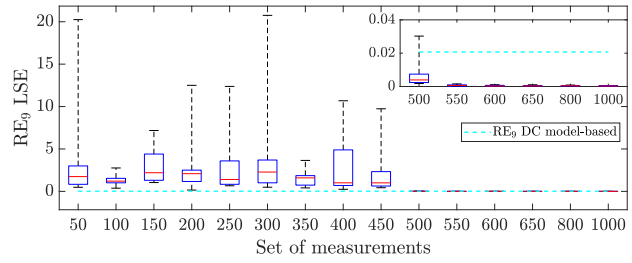


(a)

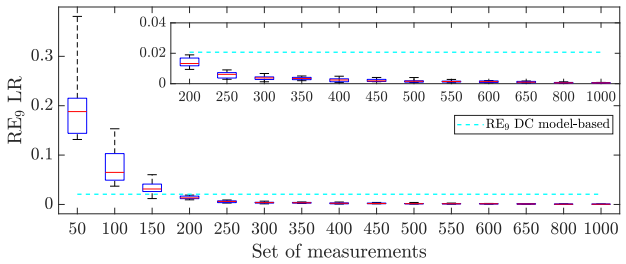


(b)

Figure 6.3: Case 1 - 9-bus: Box plot of the relative error (RE) for the estimation of the sensitivity matrix under different number of measurements: (a) RE for the least square estimator, and (b) RE for the low-rank approach.

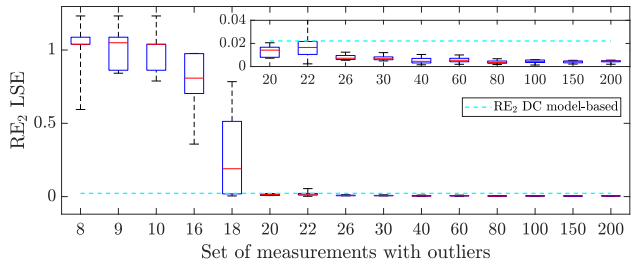


(a)

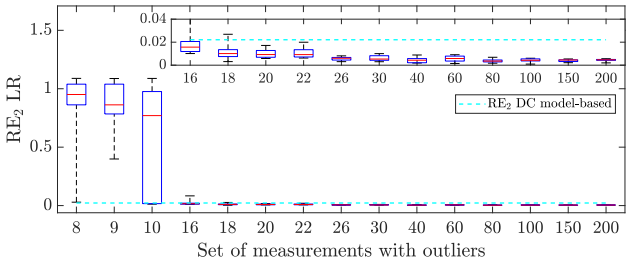


(b)

Figure 6.4: Case 1 - 500-bus: Box plot of the relative error (RE) for the estimation of the sensitivity matrix under different number of measurements: (a) RE for the least square estimator, and (b) RE for the low-rank approach.

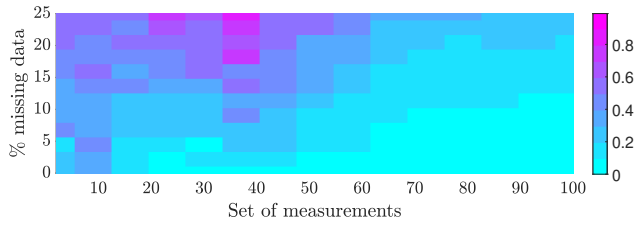


(a)

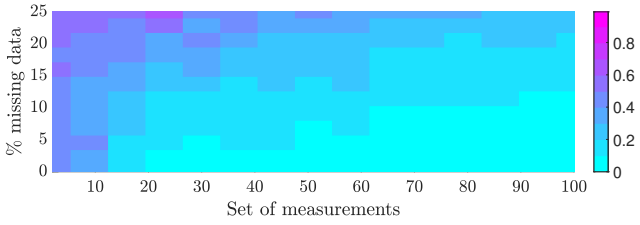


(b)

Figure 6.5: Case 1 - 9-bus: Box plot of the relative error (RE) for the estimation of the sensitivity matrix with outliers: (a) RE for the least square estimator, and (b) RE for the low-rank approach.



(a)



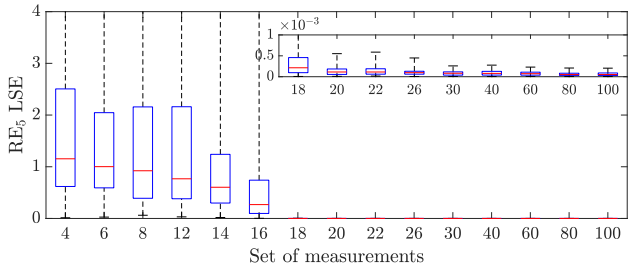
(b)

Figure 6.6: Case - 9-bus: Relative error (RE) over 50 trials for the estimation of the sensitivity matrix under a different number of measurements and different percentages of missing data: (a) RE for the least-square estimator, and (b) RE for the low-rank approach.

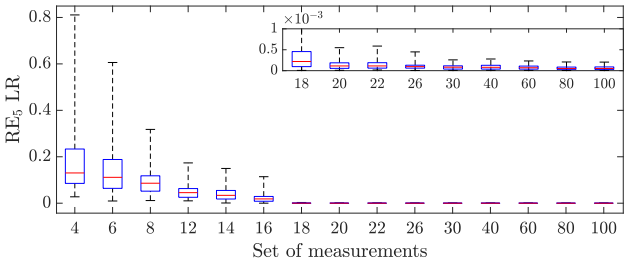
bus and in bus 51 for the 69-bus distribution system. Figure 6.7 presents the results for the RE over 10 trials. Again, it is clear that the proposed approach provides accurate results with less measurements than the least-squares approach by leveraging the intrinsic low-rank nature of the

sensitivity matrix. On the other hand, the least-squares approach provides meaningful estimates only when a sufficient number of linearly independent measurement vectors are collected. Figure 6.8 presents the behavior of the RE for different sets of measurements for the 69-bus distribution system. In this case, in order to have a median RE lower than 10%, the LSE approach requires at least 85 set of measurements while the LR method can achieve this performance under 50 set of measurements.

Moreover, in order to assess the performance of (6.9) in the case of missing and asynchronous measurements, we replicated the case for the 69-bus distribution network. Figure 6.9 presents the results for the proposed method and the least-squares approach where the LR method outperforms the LSE approach, in a case when 2% of the data in $\Delta\mathbf{F}_k$ is missing.

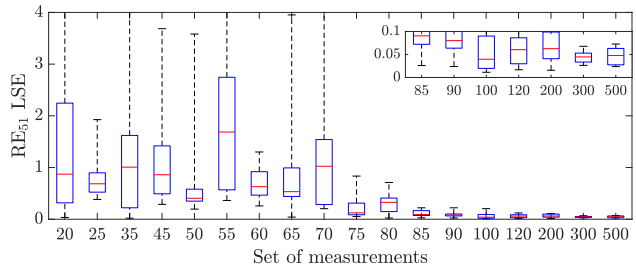


(a)

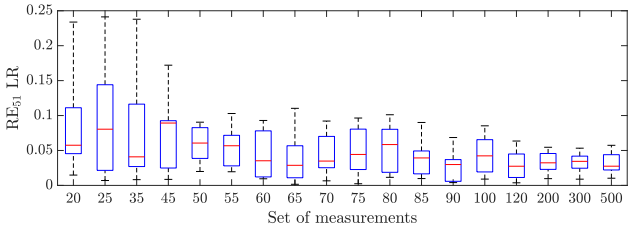


(b)

Figure 6.7: Case 2 - 18-bus: Box plot of the relative error (RE) over 10 trials for the estimation of the sensitivity matrix under different number of measurements: (a) RE for the least square estimator, and (b) RE for the low-rank approach.

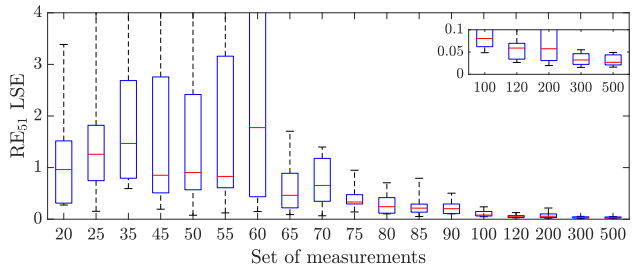


(a)

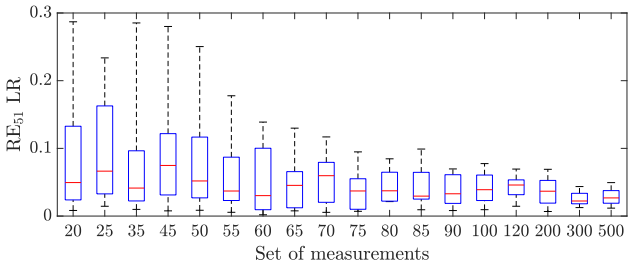


(b)

Figure 6.8: Case 2 - 69-bus: Box plot of the relative error (RE) over 10 trials for the estimation of the sensitivity matrix under different number of measurements: (a) RE for the least square estimator, and (b) RE for the low-rank approach.



(a)



(b)

Figure 6.9: Case 2 - 69-bus: Box plot of the relative error (RE) over 10 trials for the estimation of the sensitivity matrix under different number of measurements and 2% of missing data: (a) RE for the least square estimator, and (b) RE for the low-rank approach.

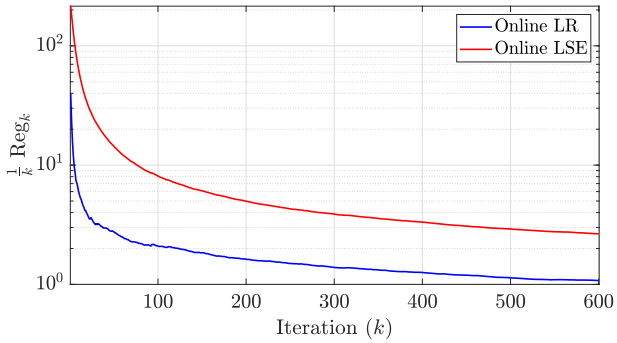


Figure 6.10: Evolution $(1/k)\sum_{i=1}^{k} f(\mathbf{x}_i) - f(\mathbf{x}_i^*)$ for the online robust estimation of the sensitivity matrix in 9-bus transmission system using LR (low-rank) method and LSE (least square estimation) approach.

6.4.2 Online Estimation

As an example of an application of Algorithm 4, we consider an online robust estimation of the sensitivity matrix for the 9-bus transmission network. Relative to the test case presented at Section 6.4.1, the nominal power injections at the nodes are now changing over time as in [53]. Figure 6.10 shows the dynamic regret $(1/k)\text{Reg}_k$, when a window of 18 measurements is used. Based on Theorem 2, in the current setting the limiting behavior of $(1/k)\text{Reg}_k$ is $\mathcal{O}(1)$. Indeed, we can see that an asymptotic error is decreasing with the time index. Figure 6.11 presents the cumulative sum of the relative error (RE) over k, i.e., $(1/k)\sum_{i=1}^{k} \text{RE}_2$, for the online robust estimation of the sensitivity matrix in 9-bus transmission system, when there are changes of topology. In this case, we change the reactance of line 5 at $k = 400$, and the reactance of line 8 at $k = 700$, for the same configuration of case 1. In addition, we test the Algorithm 1 in the 69-bus distribution network. Figure 6.12 shows the results for the cumulative sum of the RE over k for the same configuration of case 2, when a window of 75 measurements is used.

Reference [38] has shown that the proximal gradient method can be used to efficiently solve problems with thousands of variables, and each step can be performed in seconds or at the sub-second level. We used a computer with a processor Inter(R) Core(TM) i7-8850H CPU @ 2.60GHz, 32.0 GB of RAM, 64-bit Operating System. At each step of the proximal-gradient method, the

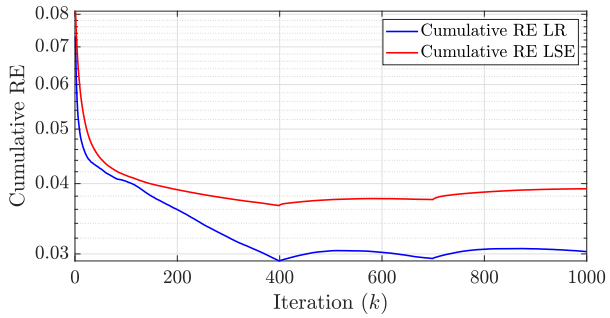


Figure 6.11: Cumulative sum of the relative errors (RE) over k, i.e., $(1/k)\sum_{i=1}^{k} \mathrm{RE}_2$, for the online robust estimation of the sensitivity matrix in 9-bus transmission system, when there are changes of topology ($k = 400$ change reactance of line 5 and $k = 700$ change reactance of line 8), using LR (low-rank) method and LSE (least square estimation) approach.

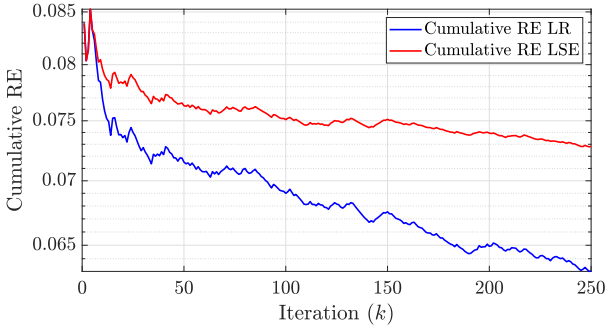


Figure 6.12: Cumulative sum of the relative errors (RE) over k, i.e., $(1/k)\sum_{i=1}^{k} \mathrm{RE}_{51}$, for the online estimation of the sensitivity matrix in 69-bus distribution system using LR (low-rank) method and LSE (least square estimation) approach.

computational time for the calculation of the gradients and the computation of the proximal operator (via CVX) were 0.0005 s, and 1.514 s for the 9-bus system, and 0.046 s and 16.606 s for the 69-bus system. Lower computational times for the proximal operator can be obtained by utilizing a dedicated algorithm. Notice also that the proximal operator can be computer via SVD by removing the constraints on the entries of the matrix [38]; in this case the computational time was 0.019 s for the 9-bus system and 0.020 s for the 69-bus system.

6.5 Conclusions

This chapter proposed a method to estimate sensitivities in a power grid by leveraging a nuclear norm minimization approach as well as sparsity-promoting regularization functions. The

proposed methodology is applicable to the estimation of various sensitivities at both transmission and distribution levels. Relative to a least-squares estimation method, the proposed approach allows to obtain meaningful estimates of the sensitivity matrix even when measurements are correlated. The method can identify outliers due to faulty sensors and is not deterred by missing measurements. An online proximal-gradient algorithm was proposed to estimate sensitivity matrices on-the-fly and enable operators to maintain up-to-date information of sensitivities under dynamic operating conditions.

Chapter 7

Renewable-based Charging for Electric Vehicle

An increasing portion of the world population is expected to live in urban and sub-urban areas, posing formidable challenges [164]. The current urban mobility system is obsolete and it requires drastic changes in order to cope with its main flaws such as congestion, inefficiency, and high carbon footprint. Fossil-fuel vehicles are a major contributor to greenhouse gases and pollutants, which in turn are interlinked with climate change and to more than 8 million deaths each year globally [86]. Vehicle electrification is key towards a radically more sustainable mobility system [31], promising a significant reduction in the environmental impact of the transportation sector. Several ride-sharing alternatives are emerging in the urban mobility sector, driven by these pressing climate and health-related issues, by increasing traffic congestion, and given the trend of younger generations favoring ride-hailing options over car ownership. Indeed, cities worldwide are already experiencing this transformation, observing the rise in popularity of on-demand ride-hailing options in companies such as Uber and Lyft [169, 71]. Ride-hailing platforms have started advertising greener options, such as shared rides and electric vehicle (EV) rides, increasing their appeal. Future mobility trends will also include fleets of autonomous EVs for ride-sharing services to improve both the quality of service (QoS) and sustainability [151, 35, 77].

However, a 100%-electrification of the urban mobility sector – and, in particular, of the ride-sharing services – may come at a cost: with the current modus operandi of the power infrastructure, large numbers of EVs may increase the loading of distribution systems, potentially surpassing the loading capacity in portions of the grid [127, 125]; this, in turn, would compromise the reliability and

increase the fragility of the power infrastructure [31]. Moreover, large swings in the power demand from EVs may impact electricity prices and energy markets. It is therefore of paramount importance to uncover coordination mechanisms between the power network operators and transportation system operators to enable a reliable and effective integration of EVs into the grid at large scale[65, 137].

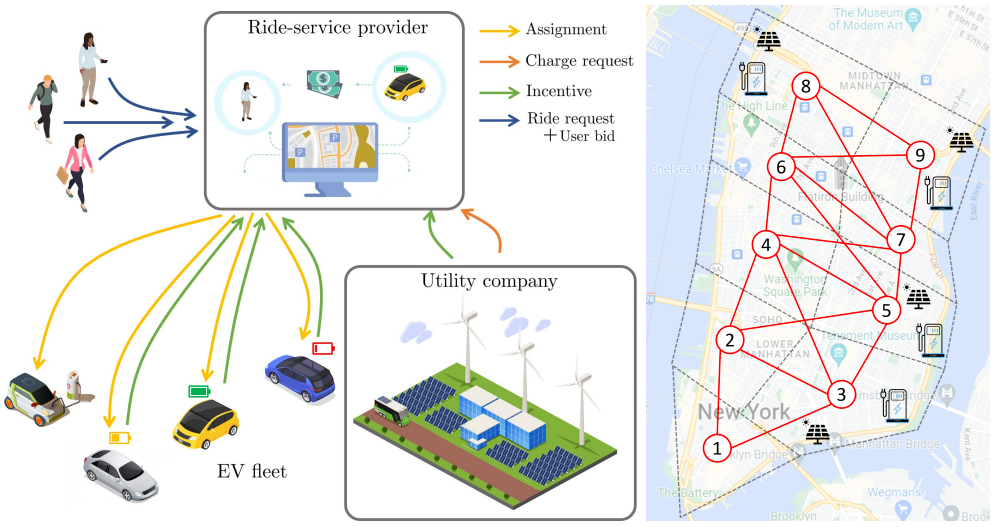
In this context, we consider a scenario where ride-sharing enterprises utilize a 100%-EV fleet, and seek responses to the following key questions: given the potential effect of EV charging on the power infrastructure, how can power utility companies and ride-hailing companies interact to promote vehicle charging in areas with high renewable generation? How can one maximize the utilization of renewables for charging purposes without disrupting the quality of the ride-hailing services? Answering the first question would allow one to systematically integrate EVs at scale with minimal effect on the power grid reliability, and without requiring structural upgrades of distribution feeders and substations to handle the additional power demand. A positive response to the second question would provide evidence for a successful transition away from internal combustion engines in the urban mobility sector. In this work, we provide answers to the questions above by investigating new means for power utility companies to interact with ride-hailing companies in order to promote renewable-based charging directly at locations where renewables are available. Beyond maximizing the use of renewable generation, the aim is also to satisfy the largest number of ride requests and to keep the unoccupied fleet size as small as possible.

Before describing the proposed methodology, we provide a brief overview of existing approaches in the context of mobility-as-a-service (MaaS) and for the coordination between transportation and power systems. Various approaches to tackle problems related to the dispatch of ride-hailing fleets can be found in the literature, including microscopic (possibly stochastic) combinatorial problems [149, 51, 9, 17, 28, 62] and macroscopic network-flow-based formulations[162]. A common topic of research is the interaction between a large fleet of (autonomous) EVs and the power infrastructure in densely populated areas, taking into account factors such as the EV charging requirements[181], fluctuating customer demand, battery degradation, and power system

constraints[137, 162, 30, 8, 150]. It is also well acknowledged that EVs have the potential to benefit the grid by providing[39, 101, 179]: (i) energy storage, serving as distributed power storage unit storing excess energy generated by renewable resources; (ii) load balancing, scheduling the EV charging during off-peak hours; and (iii) ancillary services, such frequency support.

The coupled problem arising from a transportation network together with a power network load balancing has been analyzed via single-level optimization formulations [8], exploiting the benefits of EV charging with renewable energy. Some works have explored hierarchical models such as Nash–Stackelberg–Nash game framework in a network-flow based formulation [107, 131, 66, 41, 69, 175]. Traditional game-theoretic frameworks have analyzed the interaction between EVs and the grid through the regulation of energy prices and EV charging schedules, where the grid acts as a player that sets the cost of the energy, and EVs respond with a charging schedule that optimizes their operation, usually assuming traffic flow models [178]. Other lines of work model the problem of coordinating the charging needs of an EV fleet as a game, seeking a Nash equilibrium [108]. In the case where the EV charging management problem is affected by the volatile nature of renewable generation, Generalized Nash Equilibrium (GNE) approaches have been used to coordinate the EV charging plans with real-time generation profiles [171, 47].

In this chapter, we propose a new mechanism to enable interactions between power utility companies and ride-hailing companies. For the latter, we also consider the case where rides may be shared, namely, multiple ride requests can be served by the same vehicle. The proposed process requires minimal modifications of existing vehicle-ride assignment frameworks for ride-hailing services – where rides are assigned to vehicles based on an assignment problem – and with a little computational and operational burden on the power utility side. The proposed mechanism enables a power utility company to issue *charge requests* that model a financial incentive (tied to specific renewable generation profiles and locations) offered to the ride-hailing company to promote the use of the available renewable energy. This mechanism is qualitatively illustrated in Figure 7.1(a). Drawing from game-theoretic approaches, our strategy involves a bargaining procedure where the power company proposes incentives, and the ride-hailing platform, after receiving charge incentives



(a) (b)

Figure 7.1: (a) Description of the interaction between utility company, ride-service provider, and drivers. The ride-service provider receives ride requests from customers and charge requests from a utility company. The bargaining procedure involves a vehicle-request assignment problem that takes into consideration incentives coming from both parties and bids from customers, to assign available EVs to requests. (b) Case study: Lower Manhattan, New York City, NY, partitioned into 9 regions (links between regions correspond to a reachable site within a 10 minute drive). PV generation is present in 4 regions. This image has been designed using assets from Freepik.com.

and ride requests from customers, assigns vehicles to both ride requests and charge requests. Mathematically, the interaction is in the form of a Gauss-Seidel method where, at each iteration, the power utility company proposes new incentives associated with the charge requests by solving a given optimization problem, and based on the current potential vehicle assignment; subsequently, the ride-service provider issues a new potential assignment based on the new incentives. The performance cost used in the vehicle-request assignment problem involves the minimization of the overall operational cost for the ride-service provider. The cost used by the utility company quantifies the need for the self-consumption of renewable energy resources. We point out that, while this works stresses the renewable generation profile, the same bargaining procedure can be utilized by the utility for general desirable power demand profiles; this opens the door to setups where the ride-hailing company acts as a virtual power plant providing services to the grid at convenient financial conditions.

Our approach retains the microscopic feature of existing assignment problems in the context of MaaS platforms [99, 34, 75]; moreover, following the traditional workflow with limited information sharing, it can be integrated seamlessly into MaaS orchestration services (indeed, as shown in Figure 7.1(a), the utility company can be understood just as an additional "entity" into a vehicle assignment task). At the same time, it naturally models (and enables) an interaction between power systems operations and EV-dominated ride-service providers without resorting to macroscopic (or averaged) flow models, which would be difficult to integrate into existing fleet assignment frameworks. Another important feature of our approach is that it does not rely upon combinatorial problems (therefore, it does not require dedicated software), and can leverage simple optimization algorithms. We test the proposed mechanism and show that it is indeed possible to shift the EV charging during periods of high renewable generation and adapt to intermittent generation. Our approach does not cause a degradation of the QoS for the ride-service provider with respect to ad hoc EV charging strategy.

The remainder of the chapter is organized as follows. Section 7.1 describes the problem formulation and the proposed Gauss-Seidel approach. The case study is described and the results presented in Section 7.2. Finally, Section 7.3 presents the conclusions and further directions.

7.1 Problem Formulation

In this section, we outline the main mathematical framework utilized to develop the proposed bargaining mechanism, and we explain the main implementation. The proposed mechanism involves three entities: a ride-service provider, a fleet of EVs (either with a driver or autonomous), and a power utility company; they are shown in Figure 7.1(a). These three entities interact each time a new assignment has to be made.

In the proposed framework, the power utility company aims to maximize the use of renewable generation at specific portions of the grid, where charging facilities powered by renewable resources are located. The ride-service provider manages the EV fleet to serve ride requests from customers and respond to the power utility company requests while minimizing its overall operational costs.

The goal of the EV drivers is to get ride requests assigned to their EVs.

The proposed mechanism is repeated at given intervals (e.g., $1, 5, 10, ...$ minutes, depending on specific settings) to assign available vehicles to new ride and charge requests. Accordingly, time is discretized as $t \in \{0, 1, 2, \dots\}$ (normalized to integer units). At each given time t, the scheme involves three main steps, illustrated in Figure 7.1(a):

Step 1. Customers send *ride requests* to the ride-service provider, accompanied by a possible additional tip they are willing to offer in order to compete against other customers that are also requesting rides; in parallel, the power utility company sends to the ride-service provider a number of *charge requests*.

Step 2. The EVs, the ride-service provider, and the power utility company start the bargaining procedure, mathematically explained in Algorithm 1 (outlined shortly). This step involves an iterative procedure where the ride-service provider computes potential EV-request assignments, and communicates them to the EVs and power utility company, which in turn provide new incentives to possibly influence the assignment; the interaction then repeats with the ride-service provider re-computing the potential EV-request assignments.

Step 3. Once Step 2 ends, the ride-service provider issues the final assignments.

In the following, we describe the mathematical problem formulation and pertinent algorithms associated with Steps 1–3.

A New Vehicle Assignment Method Based on Renewable and Ride Incentives: We model the transportation network topology as exemplified in Figure 7.1(b). Movements of EVs between geographical areas are described by an undirected graph; the nodes $\mathcal{N} = \{1, 2, \dots, n\}$ of the graph represent n geographical areas (neighborhoods, groups of city blocks, or towns depending on the geographical granularity); two areas are connected through an edge if they can be reached within a given traveling time. Assume that the ride-service provider receives p requests, as illustrated in Figure 7.1(a), and these are indexed in by set $\mathcal{R} = \{1, 2, \dots, p\}$; each ride request includes an origin and destination. The set $\mathcal{C} = \{1, 2, \dots, q\}$ represents the charge requests issued by the utility company; each request is associated with a charging facility (located in one of the areas

$\mathcal{N}$). We define $\mathcal{E} = \mathcal{R} \cup \mathcal{C}$ to be the set of all requests (ride and charge), and we let the set of EVs be $\mathcal{V} = \{1, 2, \ldots, m\}$, where m is the numbers of EVs. At every time slot t, the sets $\mathcal{V}, \mathcal{R}, \mathcal{C}$ are updated based on the current availability of vehicles and the most updated ride and charge requests. For notational readability, in what follows we will drop the dependence of these sets on the variable t.

In the optimization problems described in the following, we have three sets of optimization variables: (i) the variables $x_{ij} \in \{0, 1\}$ are used to describe whether the EV i is assigned or not to the ride or charge request j (with $i \in \mathcal{V}$ and $j \in \mathcal{E}$); these are decision variables of the ride-service provider. (ii) The variables $\{y_{ij}, i \in \mathcal{V}, j \in \mathcal{C}\}$ are decision variables of the power utility company, and represent financial incentives offered to the ride-service provider in order to incentivize the assignment of EVs to issued charge requests. (iii) The set of variables $\{y_{ij}, j \in \mathcal{R}\}$ represents financial incentives computed by each EV i and sent to the ride-service provider in order to obtain the ride request j. In other words, any available EV i has the task to determine the size of the incentive $\{y_{ij}, j \in \mathcal{R}\}$ based on the user's bid for request j, and on the cost of the potential trip, from the current position of EV i to the users' drop-off location. The complete list of parameters and variables is provided in Table 7.1.

With these three groups of variables, the optimization problems associated with the ride-service provider, the EVs, and the power utility company are explained next. We will first outline the three optimization problems, and then explain how these three problems are integral parts of the proposed bargaining mechanism in Step 2.

Linear Assignment at the Ride-Service Provider. We begin by formalizing the optimization problem that is solved at the ride-service provider to assign requests to available vehicles. We recall that we use binary variables $x_{ij} \in \{0, 1\}$ to describe whether EV i is assigned or not to the request j; i.e., if $x_{ij} = 1$, then EV i is assigned to serve the ride/charge request j, otherwise, $x_{ij} = 0$. These are referred to as *assignment variables*. Let c_{ij}, for $i \in \mathcal{V}$ and $j \in \mathcal{R}$, be the fixed cost (in USD, Euro, etc) for the ride-service provider to attend the ride request j using the EV i.

On the other hand, d_{ij} is the cost for the ride-service provider to attend the charge request j via EV i in a given charging facility s. The variables y_{ij} can be seen as a discount price (respectively, a price increase) to induce the EV i to be assigned to (respectively, not to be assigned to) the ride or charge request j.

When the incentives $\mathbf{y} := \{y_{ij}, i \in \mathcal{V}, j \in \mathcal{E}\}$ are given, the operational cost of the ride-service provider can be minimized through the following linear assignment problem:

$$\text{RSP}(\mathbf{y}) \qquad \min_{\{x_{ij}\in\{0,1\}, i\in\mathcal{V}, i\in\mathcal{E}\}} \quad \underbrace{\sum_{i\in\mathcal{V}} \left(\sum_{j\in\mathcal{R}} (c_{ij} - y_{ij}) x_{ij} + \sum_{j\in\mathcal{C}} (d_{ij} - y_{ij}) x_{ij} \right)}_{:= f(\mathbf{x},\mathbf{y})}, \tag{7.1a}$$

$$\text{s.t.} \qquad \sum_{i\in\mathcal{V}} x_{ij} = 1 \quad \forall j \in \mathcal{E}, \tag{7.1b}$$

$$\sum_{j\in\mathcal{E}} x_{ij} \leq 1 \quad \forall i \in \mathcal{V}. \tag{7.1c}$$

where the label "RSP($\mathbf{y}$)" emphasizes that this problem is solved by the ride-service provider once the incentives $\{y_{ij}, i \in \mathcal{V}, j \in \mathcal{E}\}$ are given (they are inputs to the problem). Constraints (7.1b) guarantee that each ride/charge request is assigned to one EV, and constraints (7.1c) ensure that each EV can be assigned to at most one request at time t. The solution to the optimization problem (7.1) minimizes the operational costs of the fleet by optimally assigning the available EVs to ride and charge requests. As in existing linear assignment problems, for the ride requests we consider the cost c_{ij} as the cost of the shortest path between the position of EV i and the pick-up point corresponding to the ride request j; similar arguments apply to d_{ij} for the charge request j corresponding to a charging facility.

We note that (7.1) is a mixed-integer linear program (MILP), which may become computationally burdensome with the increasing of number of vehicles, ride requests, and charge requests. However, one can take advantage of the totally unimodular constraint matrix property of the linear assignment problem (7.1), and show that a continuous relaxation (i.e., substitute $x_{ij} \in \{0, 1\}$ with $x_{ij} \in [0, 1]$) is exact [149]. This, in turn, allows one to leverage standard solvers for linear programs to find an optimal assignment.

Charge Incentives at the Power Utility Company. We consider an optimization problem solved by the utility company to minimize the economic loss due to the unused renewable generation at specific charging facilities. To this end, let $\mathcal{S}$ be the set of charging facilities (located within the neighborhoods or areas $\mathcal{N}$). We let $\{y_{ij}\}_{i\in\mathcal{V},j\in\mathcal{C}} \mapsto U_s(x_{ij}, y_{ij}, \theta_s)$ be a function modeling a financial or operational cost incurred by the power utility company at the charging facility $s \in \mathcal{S}$ for not using power from renewable sources of energy; this function is parametrized by the assignments $\{x_{ij}\}_{i\in\mathcal{V},j\in\mathcal{C}}$ and by additional parameters $\theta_s \in \mathbb{R}^\theta$ that are of interest to the utility company (examples are given shortly in the section Experimental Setup). Moreover, we let $\{y_{ij}\}_{i\in\mathcal{V},j\in\mathcal{C}} \mapsto \rho_s(x_{ij}, y_{ij})$ be a function that keeps track of the total incentives assigned at a charging facility s. With this notation, and for a given (potential) vehicle-request assignment $\mathbf{x}_\mathcal{C} := \{x_{ij},\, i \in \mathcal{V}, j \in \mathcal{C}\}$, the power utility company solves the following problem to compute the financial incentives associated with its charge requests:

$$\text{PUC}(\mathbf{x}_\mathcal{C}) \qquad \min_{\{y_{ij}\in\mathcal{Y}_\mathcal{C}:\, i\in\mathcal{V},j\in\mathcal{C}\}} \quad \sum_{s\in\mathcal{S}} U_s(y_{ij}; \mathbf{x}_\mathcal{C}, \theta_s), \tag{7.2a}$$

$$\text{s.t.} \qquad b_{\min,s} \le \rho_s(y_{ij}; \mathbf{x}_\mathcal{C}) \le b_{\max,s},\ \forall s \in \mathcal{S}. \tag{7.2b}$$

where $\mathcal{Y}_\mathcal{C}$ is a convex set. The label "PUC($\mathbf{x}_\mathcal{C}$)" once again stresses that this is a problem solved by the power utility company, for a given assignment $\mathbf{x}_\mathcal{C}$. Throughout this work, we assume that the function U_s is convex for any fixed x_{ij} and θ_s, while the function ρ_s is linear or affine; consequently, (7.2) is a convex problem.

Ride Incentives at the EV Fleet. We include in our framework an optimization problem associated with each EV, utilized by the EV owner to incentivize the ride-service provider to assign their preferred rides. To this end, let $\mathbf{x}_\mathcal{R} := \{x_{ij},\, i \in \mathcal{V}, j \in \mathcal{R}\}$ the denote the ride-requests assignment, and $\{y_{ij}\}_{j\in\mathcal{R}} \mapsto D_{ij}(y_{ij}; \mathbf{x}_\mathcal{R}, w_{ij})$ be a function modeling a financial cost incurred by the i-th EV when serving the ride assignment described by $\mathbf{x}_\mathcal{R}$, where $w_{ij} \in \mathbb{R}$ parametrizes the cost. Then, each EV i solves the following "best bid problem:"

$$\text{EV-}i(\mathbf{x}_\mathcal{R}) \qquad \min_{\{y_{ij}\in\mathcal{Y}_\mathcal{R}:\, j\in\mathcal{R}\}} \quad \sum_{j\in\mathcal{R}} D_{ij}(y_{ij}; \mathbf{x}_\mathcal{R}, w_{ij}). \tag{7.3}$$

Here, $\mathcal{Y}_{\mathcal{R}}$ describes a convex set of operational constraints, and D_{ij} is assumed to be a convex function for each fixed x_{ij}, w_{ij}. More precisely, in this context, y_{ij} represents the (dis)incentive that the ride-service provider will receive from the EV owner if the EV i is assigned to the ride request j. As a concrete example, the parameter $w_{ij} \in \mathbb{R}$ may represent a fixed (dis)incentive defined as $w_{ij} = h_j - \alpha_{ij} a_{ij}$, where h_j is the bid (or tip) offered by the customer that sent the ride request j (satisfying $0 \leq h_j \leq h_{\max}$), a_{ij} is the cost of reaching the drop-off location of the ride request j starting from the current position of EV i, and $\alpha_{ij} \in [0, 1]$ is a scaling factor decided by the EV owner. As an additional example, in the ride-sharing context, w_{ij} can be given by $w_{ij} = h_j - \alpha_{ij} a_{ij} + \beta b_i$, where $\beta \in [0, 1]$, and b_i represents an additional incentive that depends on the number of currently available seats on EV i.

How do Power Utility Company, Ride-Service Provider, and EVs interact? The optimization problems (7.1)–(7.3) are utilized to compute potential assignments (for given incentives) and incentives (for given potential assignments). We consider a game-theoretic approach where potential assignments and incentives are sequentially updated until they meet a termination criteria. Considering an iterative approach, where $k \in \mathbb{N} \cup \{0\}$ is the iteration index, let $\mathbf{y}^{(k)}$ and $\mathbf{x}^{(k)}$ the assignments and incentives at iteration k; then, our mechanism involves updates of the form $\text{RSP}(\mathbf{y}^{(0)}) \to \mathbf{x}^{(1)} \to \text{PUC}(\mathbf{x}_{\mathcal{C}}^{(1)}), \text{EV-}i(\mathbf{x}_{\mathcal{R}}^{(1)}) \to \mathbf{y}^{(1)} \to \text{RSP}(\mathbf{y}^{(1)}) \to \mathbf{x}^{(2)} \to \text{PUC}(\mathbf{x}_{\mathcal{C}}^{(2)}), \text{EV-}i(\mathbf{x}_{\mathcal{R}}^{(2)}) \to \mathbf{y}^{(2)} \to \dots$ until convergence; the assignments at convergence are then dispatched to the EVs. The notation $\text{RSP}(\mathbf{y}^{(k)})$ means that the problem (7.1) is solved by the ride-service provider to issue a new potential assignment $\mathbf{x}^{(k+1)}$, based on the current incentives $\mathbf{y}^{(k)}$ received from the power utility company and the EVs. Similarly, $\text{PUC}(\mathbf{x}_{\mathcal{C}}^{(k)})$ means that the power utility company computes new incentives $\mathbf{y}_{\mathcal{C}}^{(k+1)}$ by solving (7.2) based on the current potential assignment $\mathbf{x}_{\mathcal{C}}^{(k)}$, and each EV updates its own incentives by solving (7.3). At each round, the ride-service provider sends to the power utility company and EVs new potential assignments and receives from them new incentives. This mechanism, which is in the form of a Gauss-Seidel method [60] and is repeated at every time slot t, is tabulated as Algorithm 5.

Algorithm 5: Gauss-Seidel Best Response-based Algorithm

Initialization

[**I1**] Ride-service provider receives ride requests $\mathcal{R}$ and charge requests $\mathcal{C}$.

[**I2**] Ride-service provider chooses a feasible initial assignment $\mathbf{x}^{(0)}$ and sends it to the power utility company and EVs.

Gauss-Seidel mechanism

for $k = 1, 2, \ldots$ until termination criterion is satisfied **do**

[**S1-a**] Power utility company updates incentives as:

$$\mathbf{y}_{\mathcal{C}}^{(k)} \in \arg\min_{\mathbf{y}} \left\{ \sum_{s\in\mathcal{S}} U_s \left(\mathbf{y}; \mathbf{x}_{\mathcal{C}}^{(k-1)}, \theta_s \right) \mid b_{\min,s} \leq \rho_s(\mathbf{y}; \mathbf{x}_{\mathcal{C}}^{(k-1)}) \leq b_{\max,s} \ \forall s \in \mathcal{S}, \ \mathbf{y} \in \mathcal{Y}_{\mathcal{C}} \right\}.$$

[**S1-b**] Power utility company sends $\mathbf{y}_{\mathcal{C}}^{(k)}$ to ride-service provider.

[**S2-a**] Each EV updates incentives as:

$$\mathbf{y}_{\mathcal{R}}^{(k)} \in \arg \min_{\{y_{ij}\in\mathcal{Y}_{\mathcal{R}}, j\in\mathcal{R}\}} \quad \sum_{j\in\mathcal{R}} D_{ij} \left(y_{ij}; \mathbf{x}_{\mathcal{R}}^{(k-1)}, w_{ij} \right).$$

[**S2-b**] Each EVs send $\mathbf{y}_{\mathcal{R}}^{(k)}$ to ride-service provider.

[**S3-a**] Ride-service provider updates the potential assignment as:

$$\mathbf{x}^{(k)} \in \arg\min_{\mathbf{x}} \left\{ f(\mathbf{x}; \mathbf{y}^{(k)}) \mid \mathbf{x} \in \mathcal{X} \right\},$$

where

$$\mathcal{X} = \left\{ \mathbf{X} \in \mathbb{R}^{m\times p+q} \ : \ x_{ij} \in [0,1], \text{ and } \textstyle\sum_{i\in\mathcal{V}'} x_{ij} = 1, \ \forall j \in \mathcal{E}', \sum_{j\in\mathcal{E}'} x_{ij} = 1, \ \forall i \in \mathcal{V}' \right\}.$$

[**S3-b**] Ride-service provider sends $\mathbf{x}_{\mathcal{C}}^{(k)}$ to utility company and $\mathbf{x}_{\mathcal{R}}^{(k)}$ to EVs.

end for

EV dispatch

[**D1**] Ride-service provider dispatches assignments to EVs.

Note that in step [S3-a] the binary variables $x_{ij} \in \{0, 1\}$ have been relaxed to $x_{ij} \in [0, 1]$; accordingly, the relaxed assignment problem in step [S3-a] is a continuous convex linear problem, which can be solved efficiently. Moreover, in the section Theoretical Foundations, we will show that the relaxation is exact. The steps of the Gauss-Seidel algorithm are repeated until convergence (i.e., when $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$ and $\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)}$) or a maximum number of iterations is reached.

The proposed method is in line with game-theoretic frameworks [60, 64, 91, 141]. Specifically, it can be modeled as a non-cooperative game involving three groups of agents: the ride-service provider, the power utility company, and the EVs. These agents interact to optimize their respective costs. To analyze our method, we initially considered the case where no constraints are imposed on

the power utility company. In this case, we approached the problem from the perspective of Nash equilibrium (NE), drawing analogies with variational inequality (VI) theory as discussed in [60]. However, when we considered the complete problem, it became a Generalized Nash Equilibrium (GNE) problem. In this context, an analogy can be made to quasi-variational inequalities (QVI) [64]. To assess the performance of Algorithm 5, numerical experiments can be conducted, as demonstrated in [128].

Symmetric Linear Assignment Problem

The problem in (7.1) is an asymmetric linear assignment problem[126, 23]. By taking advantage of the totally unimodular constraint matrix property of the linear assignment problems in (7.1), it is well known that their continuous relaxation (i.e., when one substitutes $x_{ij} \in \{0,1\}$ with $x_{ij} \in [0,1]$) is exact [149]. Therefore, we can reformulate the problem (7.1) by following a similar procedure as [149]. First, we add virtual requests $\mathcal{M}_E$ or virtual EVs $\mathcal{M}_V$ such that $\mathcal{E}' = \mathcal{E} \cup \mathcal{M}_E$ and $\mathcal{V}' = \mathcal{V} \cup \mathcal{M}_V$ have the same cardinality, and define $h = \max\{m, p+q\}$. Second, the assignment cost c_{ij} (or d_{ij}) of the virtual EVs in $\mathcal{M}_V$ for all requests in $\mathcal{M}_E$ is set to ∞. The incentives for the virtual EVs are set to 0. Thus, the problem (7.1) is equivalent to the following symmetric linear assignment problem:

$$\min_{x_{ij}\in[0,1]} \quad \sum_{i=1}^{h}\sum_{j=1}^{h}(c_{ij}-y_{ij})x_{ij} + \sum_{i=1}^{h}\sum_{j=1}^{h}(d_{ij}-y_{ij})x_{ij}, \tag{7.4a}$$

$$\text{s.t.} \quad \sum_{i=1}^{h} x_{ij} = 1 \quad \forall j \in \mathcal{E}', \tag{7.4b}$$

$$\sum_{j=1}^{h} x_{ij} = 1 \quad \forall i \in \mathcal{V}'. \tag{7.4c}$$

We then have the following result.

Lemma 3 Problem (7.4) is an exact continuous relaxation of the binary Problem (7.1).

The proof of the result follows from the totally unimodularity of the constraints, or equivalently from the fact that the solutions are vertices of the Birkhoff's polytope. This is a standard result in linear programming. The interested reader is referred to [33, 149].

7.2 Case Study

7.2.1 Experimental Setup

Data Sources: Our case study is based on ride requests from the lower Manhattan area, in New York City, NY. We use real data recorded by the Taxi and Limousine Commission (TLC) [158] on Tuesday, March 1, 2022, between 6:00 and 24:00. The total number of ride requests collected during this time window is 19794, from which we keep a random sample of 2462 to have a number of ride requests that can be handled by a fleet of 100 EVs. The ride requests have been categorized according to their submission time, and grouped into one-minute time slots. Note that the ride requests sub-sampling is due to the fact that we require to keep the simulation of several experiments manageable on a 2.4 GHz Quad-Core Intel Core i5 laptop with 8 GB RAM memory. We obtain approximated power generation profiles for PV systems from the renewable historical data of New York Independent System Operator[118]. As a point of reference, we extracted the PV shape from the renewable energy generated within New York State on a sunny day, and modify it accordingly to represent also a cloudy day. To test our model, we consider three scenarios, corresponding to different weather conditions: sunny day (i.e., maximum PV generation), and cloudy day, which presents two alternatives: cloudy morning and cloudy afternoon. According to NYC OpenData [115], we estimate that about 5 MW of PV peak power will be generated in the part of Manhattan that we are considering, distributed among a total of 39 charging stations. Assuming all PV systems have equal capacity, each station is estimated to produce about 125 kW during peak hours. Since most likely only a fraction of the generated power will be designated to EV charging, and given the smaller fleet size considered in this work, we assume that each charging station can produce 25 kW as peak power. The charging stations are grouped within the 4 regions they belong to by adding their PV capacity. In our graph, this translates to 4 nodes hosting charging facilities, as depicted in Figure 7.1(b). Since the number of charging stations at each of these facilities varies, the maximum PV generation will be different in each region, as one can see in Figure 7.5.

Problem and algorithm setup: Next, we describe the parameters and functions used in

the case study. We consider a fleet of 100 EVs, each with a 50 kWh battery capacity. We assume that each EV consumes 0.1 kWh per minute traveled, and recharges 0.2 kWh per minute spent at any charging station equipped with level 2 EV chargers, i.e., it takes approximately 4 hours to fully charge an empty EV battery. To make our model more realistic, we implement some constraints summarized in the following. EVs whose battery status is up to 2/3 of the full battery capacity cannot attend a charge request. EVs are expected to charge fully once they have accepted a charge request and will therefore not be considered for other assignments in the meanwhile. When they are available, EVs can only accept charge requests coming from the region where they are idling or from neighboring areas located at most one edge distance. EVs must have enough battery to complete the planned trip to be considered as candidates for a request. Ride requests can be attended by EVs located in the near neighborhood, i.e., no further than two edges distance. The travel time to move from one neighboring area to another is 10 minutes. The EV assignment is performed every 1 minute.

If the EV i cannot accommodate the ride request $j \in \mathcal{R}$ or the charge request $j \in \mathcal{C}$, we set $c_{ij} = 10^6$ or $d_{ij} = 10^6$, respectively; this means that the assignment is infeasible[149]. The functions for the optimization problems are defined as follows:

- For the EV:

$$D_{ij}(y_{ij}, w_{ij}) = (y_{ij} - w_{ij})^2, \mathcal{Y}_{\mathcal{R}} = [r_{\min}, r_{\max}],\ \forall i \in \mathcal{V},\ j \in \mathcal{R}.$$

- For the power utility company:

$$U_s(x_{ij}, y_{ij}, \theta_s) = (L(P_{\text{ref,s}}) - \rho_s(x_{ij}, y_{ij}))^2,\ \mathcal{Y}_{\mathcal{C}} = [l_{\min}, l_{\max}],\ \forall s \in \mathcal{S},$$

where $L(P_{\text{ref,s}}) = c_{\text{RER}}\,(P_{\text{ref},s} - v_{\text{ch},s} p_{\text{ch}})$, and $\rho_s(x_{ij}, y_{ij}) = \sum_{i \in \mathcal{V}} \sum_{i \in \mathcal{C}_s} x_{ij} y_{ij},\ \forall s \in \mathcal{S}$. The function L accounts for the loss incurred due to excess of renewables, where $P_{\text{ref},s}$ is the power generated at specific areas where the charging facility s is located, v_{ch} is the number of EVs currently charging at facility s, and p_{ch} is the power delivered to each charging EV. The time-varying price of the generated PV power is given by c_{RER}.

Variable Symbol	Description	Value/Range
$\mathcal{V},\ i \in \mathcal{V}$	set of EVs with cardinality m, index of EV	$m = 100$
$\mathcal{R},\ j \in \mathcal{R}$	set of ride requests with cardinality p, index of ride request	p time-varying
$\mathcal{C},\ j \in \mathcal{C}$	set of charge requests with cardinality q, index of charge request	q time-varying
$\mathcal{E} = \mathcal{R} \cup \mathcal{C}$	set of requests (ride and charge)	$\lvert\mathcal{E}\rvert$ time-varying
$\mathcal{N}$	set of nodes with cardinality n	$n = 9$
$\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$	set of edges	$\lvert\mathcal{L}\rvert = 18$
$\mathcal{S},\ s \in \mathcal{S} \subseteq \mathcal{N}$	set of renewable-powered charging facilities, index of charging facility	$\lvert\mathcal{S}\rvert = 4$
$\mathbb{R}^{m\times p+q} \ni \mathbf{X} = [x_{ij}]$	matrix of optimization variables for the vehicle-request assignment	$x_{ij} \in \{0, 1\}$
$\mathbf{x} = \text{vec}(\mathbf{X}) = [\mathbf{x}_\mathcal{R}; \mathbf{x}_\mathcal{C}]$	vector of optimization variables for ride $\mathbf{x}_\mathcal{R}$ and charge $\mathbf{x}_\mathcal{C}$ requests	$x_{ij} \in \{0, 1\}$
$\mathbb{R}^{m\times p+q} \ni \mathbf{Y} = [y_{ij}]$	matrix of optimization variables for the (dis)incentive problem	$y_{ij} \in \mathcal{Y}$
$\mathbf{y} = \text{vec}(\mathbf{Y}) = [\mathbf{y}_\mathcal{R}; \mathbf{y}_\mathcal{C}]$	vector of optimization variables for EVs $\mathbf{y}_\mathcal{R}$ and utility $\mathbf{y}_\mathcal{C}$ incentives	$y_{ij} \in \mathcal{Y}$
$\mathbb{R}^{m\times p} \ni \mathbf{C} = [c_{ij}]$	matrix of operational cost for ride requests w.r.t. each EV	$c_{ij} \in \mathbb{R}_{\geq 0}$
$\mathbb{R}^{m\times q} \ni \mathbf{D} = [d_{ij}]$	matrix of operational cost for charge requests w.r.t. each EV	$d_{ij} \in \mathbb{R}_{\geq 0}$
$\mathbb{R}^{m\times p} \ni \mathbf{W} = [w_{ij}]$	matrix of fixed (dis)incentives w.r.t. ride requests	$w_{ij} \in \mathbb{R}$
$P_{\text{ref},s}$	surplus of renewable energy at charging facility s	$P_{\text{ref},s} \in \mathbb{R}_{\geq 0}$
v_{ch}	number of EVs currently charging at facility s	$v_{\text{ch}} \in \mathbb{R}_{\geq 0}$
p_{ch}	power delivered to each EV	$p_{\text{ch}} = 12$ kW
c_{RER}	price kWh of the generated renewable power	time-varying
h_j	user's bid for ride request j	$h_j \in [0,\ h_{\max}]$
α_{ij}	weight of fixed incentive w.r.t. EV i attending ride request j	$\alpha_{ij} \in [0, 1]$
a_{ij}	cost from EV i location to request j destination	$a_{ij} \in \mathbb{R}_{\geq 0}$
β	weight of fixed incentive to ride-share	$\beta \in [0, 1]$
b_i	fixed incentive to ride-share in EV i	$b_i \in [0, 4]$
$b_{\min,s}$, $b_{\max,s}$	min/max for the total incentive at charging facility s	$\{b_{\min,s},\ b_{\max,s}\} \in \mathbb{R}$
$r_{\min}$, $r_{\max}$	min/max for the (dis)incentive w.r.t. ride requests	$\{r_{\min},\ r_{\max}\} \in \mathbb{R}$
$l_{\min}$, $l_{\max}$	min/max for the incentive w.r.t. charge requests	$\{l_{\min},\ l_{\max}\} \in \mathbb{R}$

Table 7.1: Definition of sets, variables, and parameters for our vehicle assignment method based on renewable and ride incentives.

7.2.2 Results & Discussion

The following results stem from the mathematical algorithms described in Section 7.1. The experiments consider a fleet of 100 EVs and use the data recorded by the Taxi and Limousine Commission (TLC)[158] on Tuesday, March 1, 2022, between 6:00 and 24:00. In terms of renewable generation, we consider the photovoltaic (PV) power generation profiles extracted from the renewable historical data of New York Independent System Operator[118], as explained in the Data Sources section. To clarify, we use the term ride-hailing below to indicate the case where each vehicle carries only one passenger, as in a taxi service. On the other hand, we use the term ride-sharing to refer to the case where several riders can be picked up along a route by the same vehicle. In line with this terminology, even shared/pooled ride-hailing services are included in this category. The complete description of the simulation setup is provided in Section 7.1. To present the results we define three study cases:

• *Business-as-usual*: the ride-service provider assigns ride requests to an EV fleet. The EVs are exclusively charging when their batteries are empty. The EV charging does not exploit renewable energy resources.

• *Case 1*: the ride-service provider receives and assigns both ride and charge requests, based on the outcome of the process described in Section 7.1, in which the EV fleet and the power utility company engage in an incentive-assignment process.

• *Case 2*: this is a further extension of *case 1*, and includes the option of ride-sharing. In this case, EVs are allowed to pick up more passengers during the trip provided they are willing to ride-share and have a common destination. Each additional passenger can cause a delay of up to 4 minutes.

Case 1 and *case 2* are evaluated under three different weather conditions, named sunny day, cloudy morning, and cloudy afternoon, that correspond to different PV generation profiles. To assess the performance of the proposed strategy in the cases defined above, we introduce two metrics. These are: the *quality of service (QoS)*, defined as the total number of missed ride requests over the total number of received ride requests, and the *power loss (PL)*, defined as the percentage of unexploited renewable power. A summary of the results is presented in Table 7.2. Notice how the QoS always improves with the sharing acceptance, e.g., going from 94.8% to 99.9% on a sunny day, indicating that ride-sharing is critical to enforce high levels of QoS. On the other hand, the amount of unexploited renewable power increases as the willingness to ride-share improves. In this case, the EVs need to charge less often since their total driving time is overall reduced, given the same (or larger) number of accepted ride requests.

To provide some context, we start by considering a fleet consisting of regular fossil-fuel vehicles. Figure 7.2 shows the availability of vehicles over the simulation window, where if an EV is attending a ride request it is labeled as "riding", otherwise, if it is unoccupied it is labeled as "idling". In Section 7.2, we will compare the performance of the fossil-fuel fleet against a 100% EV fleet, given the same amount of received ride requests.

Effects of business-as-usual EV charging on the power infrastructure: Initially, we

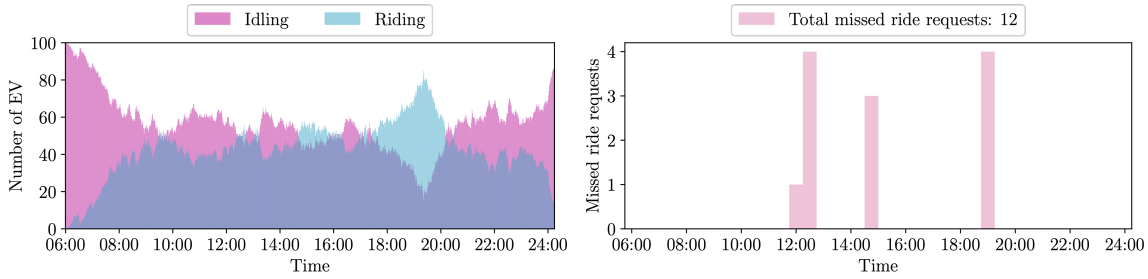


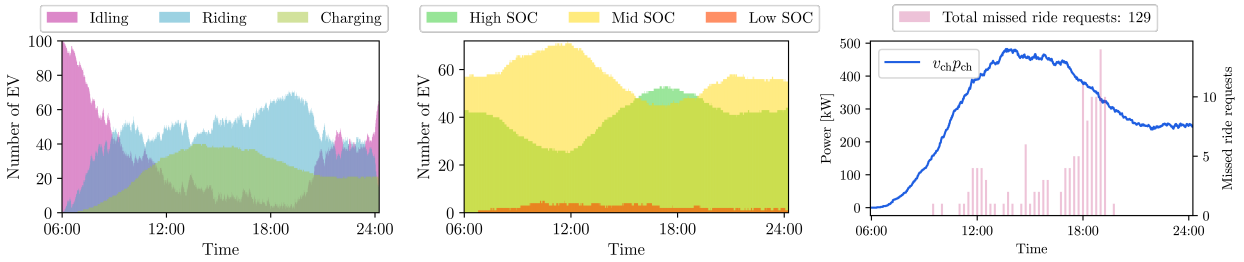
Figure 7.2: Availability of fossil-fuel vehicles during the day (left) and the total number of missed ride requests (right). With a fleet of only fossil-fuel vehicles, the QoS is 99.5%.

		Case 1		**Case 2**							
				100%		**75%**		**50%**		**25%**	
	Metric	QoS	PL	QoS	PL	QoS	PL	QoS	PL	QoS	PL
Sun	Sunny day	94.8%	36.5%	99.9%	45.2%	98.5%	42.0%	97.8%	39.6%	95.8%	37.6%
	Cloudy morning	93.4%	19.0%	99.9%	24.1%	98.7%	22.4%	97.6%	20.4%	94.6%	20.5%
	Cloudy afternoon	93.9%	24.1%	99.3%	28.5%	98.0%	26.6%	96.9%	25.4%	94.9%	24.8%

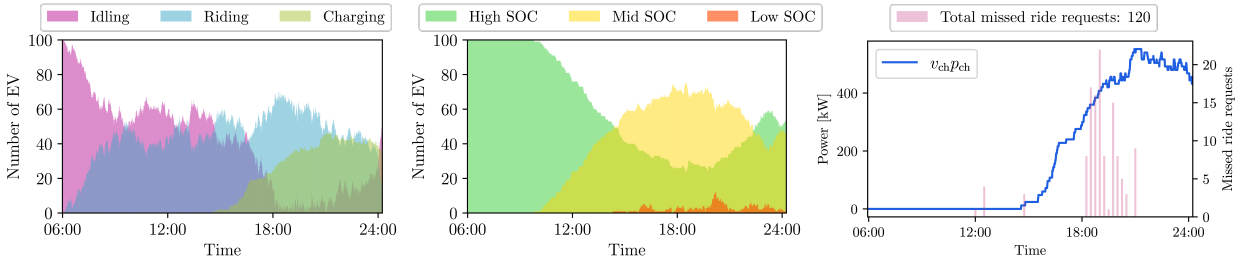
Table 7.2: Quality of service (QoS), and power loss (PL) evaluated over the simulation window. In *case 1*, the ride-service provider assigns ride and charge requests, and the EV fleet and utility company implement the bargaining procedure described in Section 7.1. *Case 2* adds the ride-sharing option to *case 1*; here we assume that 100%, 75%, 50%, and 25% of the customers are willing to ride-share. As a benchmark metric, the QoS for the *business-as-usual* case is 94.8%.

consider a vehicle-request assignment where the ride-service provider has to handle ride requests only. This case, which disregards the presence of renewable energy resources, will serve as a benchmark for later comparison, and represent the *business-as-usual* case. The assignment is performed as described in, e.g., [149], for the available EVs. We assume that the EVs will start charging when their State Of Charge (SOC) is low ($< 10\%$ of the battery capacity), regardless of their location and the time of the day. Figure 7.3 shows the availability of EVs, the SOC trend during the day, and the EV charging profile (i.e., the amount of power used to charge v_{ch} EVs, at a charging rate p_{ch}), with the number of missed ride requests. For convenience, we consider three levels of SOC: high ($> 60\%$), mid (between 10% and 60%), and low ($< 10\%$). We explore two different initial SOC conditions: random and fully charged. For random SOC, we present averaged results over 10 iterations. In this case, without charge requests, the EVs connect to the power grid whenever they are out of battery, generating a peak of power consumption. We show the availability of EVs over the day for both cases in Figure 7.3, where if an EV is busy charging it is

labeled as "charging". In Figure 7.3(a), where the initial SOC of EVs are randomly generated, the peak of the EV charging profile occurs between 13:00 and 17:00, while in Figure 7.3(b), where all the EVs start fully charged at 6:00, the peak is shifted, starting at 20:00. In both cases, we observe that the majority of missed ride requests occurs in the time interval between 18:00 and 20:00.



(a)



(b)

Figure 7.3: Availability of EVs during the day, SOC time-evolution, and charging profiles together with total number of missed ride requests given two different initial SOC conditions: random distribution between 10% and 100% of the battery capacity (a) and fully charged (b). Business-as-usual case. Dark blue line corresponds to the charging profile (i.e., total power used to charge v_{ch} EVs, each receiving p_{ch} kW).

Renewable-based charging and bargaining strategy: We focus now on *case 1*, where the ride-service provider handles both ride and charge requests. In this case, the rider-service provider, the power utility company, and the EVs interact as described in Section 7.1 and as illustrated in Figure 7.1(a). We recall that charge requests are issued by the power utility company. EVs are assigned to feasible requests, depending on their distance to the customers' pickup point or to the charging facilities. The shortest path is assumed to correspond to the lower cost and therefore preferred by the ride-service provider. Moreover, a crucial aspect of our approach is to rely on a bargaining mechanism, in the form of financial incentives, that influences the preferences

of the ride-service provider concerning the assignment of EVs to ride or charge requests. In the bargaining mechanism (a rigorous formulation of the problem can be found in Section 7.1), we augment a linear assignment problem with additional constrained optimization problems, whose objective is to assign a financial incentive to the cost of the ride or charge requests based on the bids from the users and the losses affecting the power utility company. We present averaged results over 10 iterations, where each EV is initially assigned a random SOC between 10% and 100% of the full battery capacity. In Figure 7.4, the top row shows the availability of EVs through the simulation window; the second row displays how the SOC of the EV fleet varies during the day under different weather conditions; the third row shows the total PV generation profile in the lower Manhattan area, compared to the power distributed to charging EVs; and the fourth row presents the incentives for ride and charge requests. Notice how the EV charging schedule now follows the PV generation profile and it is impacted by the weather conditions. Figure 7.5 shows the individual PV generation profiles and the amount of power dispatched in each of the four areas equipped with charging facilities. Then, our method successfully can shift the charging profiles to periods of the day characterized by high renewable generation and avoid the negative impacts on the grid that the wider charging schedule showed in Figure 7.3 can induce.

Regarding the effects of the variability of renewable energy resources on our formulation, we can see in Figures 7.4(e)-7.4(f) the impact on the SOC for two different PV profiles, both affected by significant cloud coverage. In the case of a cloudy morning, EVs cannot fully charge between 9:00 and 13:00, resulting in a larger amount of missed ride requests during the rest of the day (see Figure 7.4(h)), and many more EVs with low SOC by midnight, as compared to the sunny day case. In contrast, the cloudy afternoon profile still allows the EVs to fully charge in the morning, substantially limiting the need for charging in the afternoon and therefore without significantly affecting the final SOC of the EVs. However, in the latter case, slightly more ride requests are missed, as shown in Figure 7.4(i). This behavior can be understood by focusing on the bargaining mechanism for those scenarios. Figure 7.4(k) shows fewer charge request incentives from 8:00 to 12:00, due to a drop in the PV generation in the morning. In the cloudy afternoon scenario, however,

during the same period, the charge request incentives are higher, and therefore, the EV charging is favored over attending ride requests as shown in Figure 7.4(l). The charge request financial incentives depend on the PV profile scenario considered, i.e., sunny days are characterized by a considerable amount of incentives during the day following the consistent PV-power generation, while cloudy days experience a drop in the amount of charge request incentives during the morning and afternoon, respectively.

The benefits of ride-sharing: We explore the impact of ride-sharing in *case 2*, by allowing EVs to pick up more passengers provided that they share a common destination. Passengers can be picked up along the journey, i.e., they do not need to start from the same pick-up area. It is important to stress we are still dealing with a one-to-one vehicle-request assignment but we introduce a new parameter, the EV passenger capacity, to keep track of the passengers' number on board which can be at most 4. Passengers express their willingness to ride-share whenever they submit a request, in exchange for a discounted price and a greater chance to get a ride.

We introduce a new parameter, customers' willingness to ride-share, and analyze how it impacts the results in four different scenarios where 100%, 75%, 50% or 25% of the customers are willing to ride-share. Figure 7.6 is the counterpart of Figure 7.4, this time taking also ride-sharing (75%) into account. Figure 7.7(a) shows how the number of missed ride requests is affected by the customers' willingness to ride-share. For example, assuming that 50% of the passengers would be willing to share rides on a sunny day, the total number of missed ride requests reduces to 55 and to only 3 in the case where all passengers would agree to share rides. This is a substantial improvement, roughly forty times less than the case that does not take ride-sharing into consideration. Figure 7.7(b) also reports how many EVs would have a low or high final SOC at the end of the day, depending on the customers' willingness to ride-share. Here, we can see that when the willingness to ride-share increases also the number of EVs with high SOC at the end of the simulation is larger, confirming that an improved QoS can be achieved even if the need for charging is less, as shown in Table 7.2.

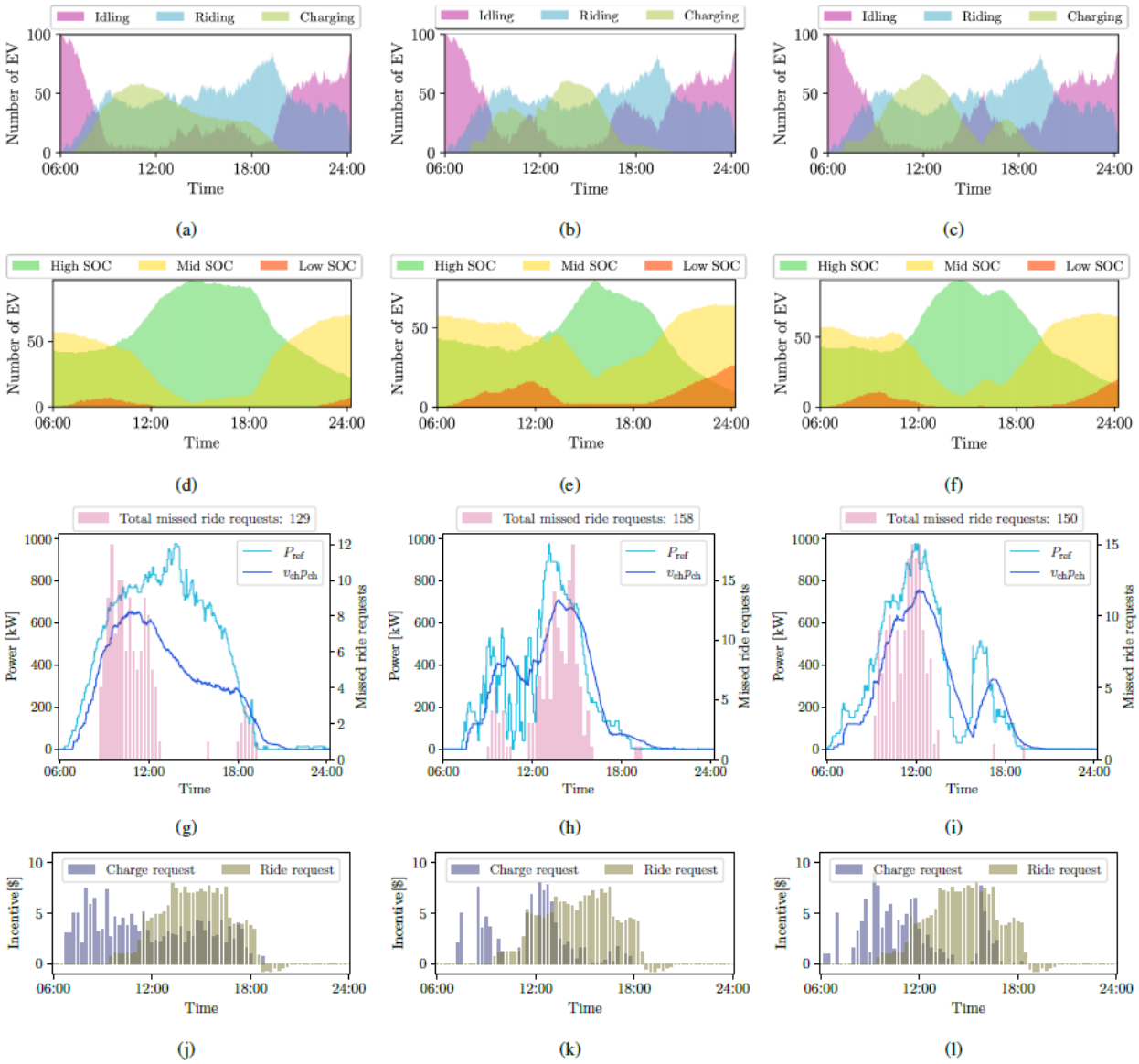


Figure 7.4: Availability of EVs during the day (1st-row), SOC time-evolution (2nd-row), charging profiles together with the total number of missed ride requests (3rd-row) and incentives (4th-row) during the day, given three different weather scenarios: sunny (a)-(d)-(g)-(j), cloudy morning (b)-(e)-(h)-(k), and cloudy afternoon (c)-(f)-(i)-(l). The initial SOC of each EV is randomly set within 10% and 100% of the battery capacity. The PV-generation (P_{ref}) and charging profiles ($v_{\text{ch}}p_{\text{ch}}$) are obtained summing over all charging stations.

On-demand ride-sharing helps to reduce traffic congestion and emissions, and especially in densely populated areas it is paving the way for more sustainable mobility. However, it is still facing a significant amount of skepticism due to many heterogeneous factors, ranging from users'

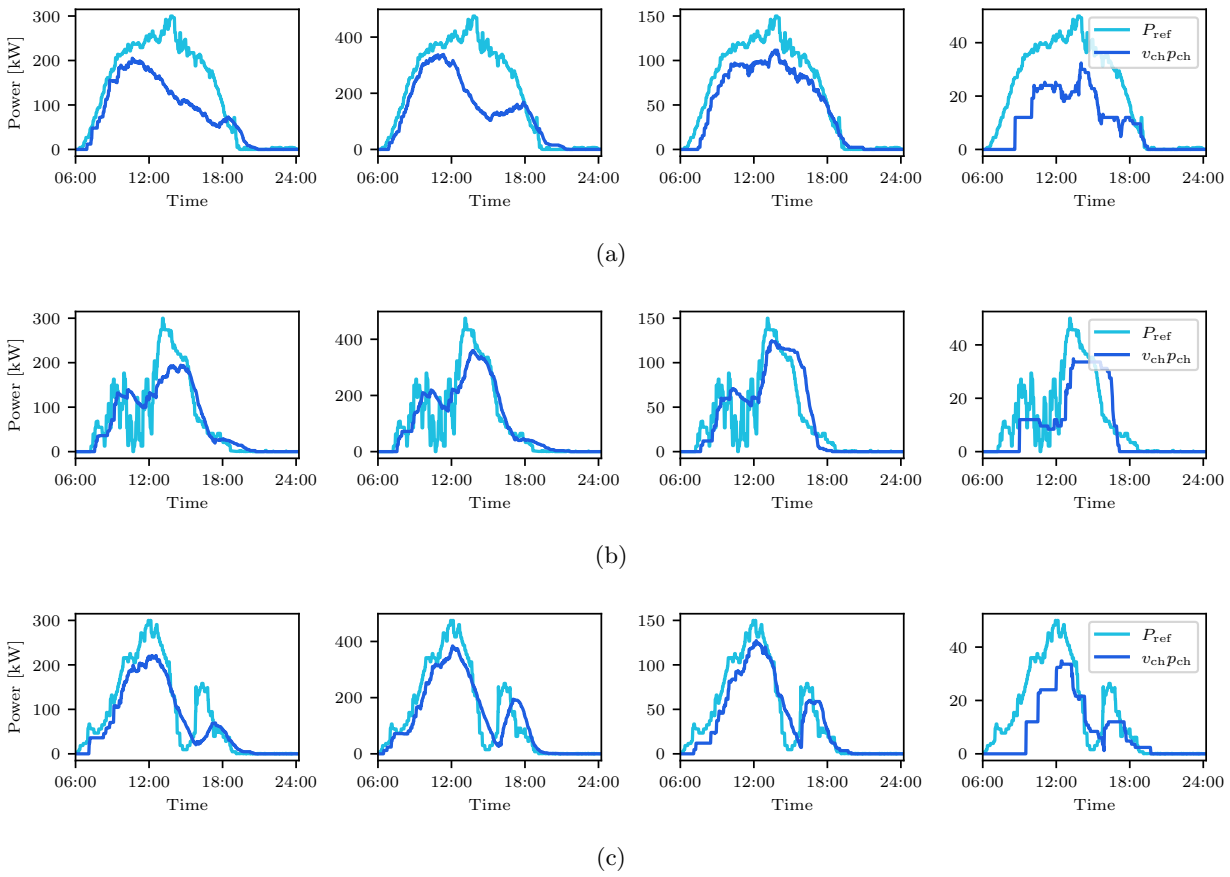


Figure 7.5: Charging profile for each of the 4 regions hosting charging facilities, corresponding (left to right) to node 3, 5, 8 and 9 in Figure 7.1(b), given three different weather scenarios: sunny (a), cloudy morning (b), and cloudy afternoon (c). The curves in dark and light blue correspond to the PV-generation (P_{ref}) and the charging profiles ($v_{\text{ch}}p_{\text{ch}}$), respectively.

preferences to seamless integration into the transportation system[153]. In particular, if EVs are adopted as the main way of transportation, the interaction between the ride-service providers and the distribution power system plays a crucial role in determining the successful operation of the EV fleet.

In this work, we investigate the interaction between a ride-service provider that manages a 100% EV fleet, and a power utility company that operates renewable energy resources at some charging locations. We show that through the proposed bargaining mechanism we achieved an EV charging schedule that maximizes the use of renewable generation and reduces the potential negative effects of the EV charging on the power infrastructure. Moreover, we are able to preserve

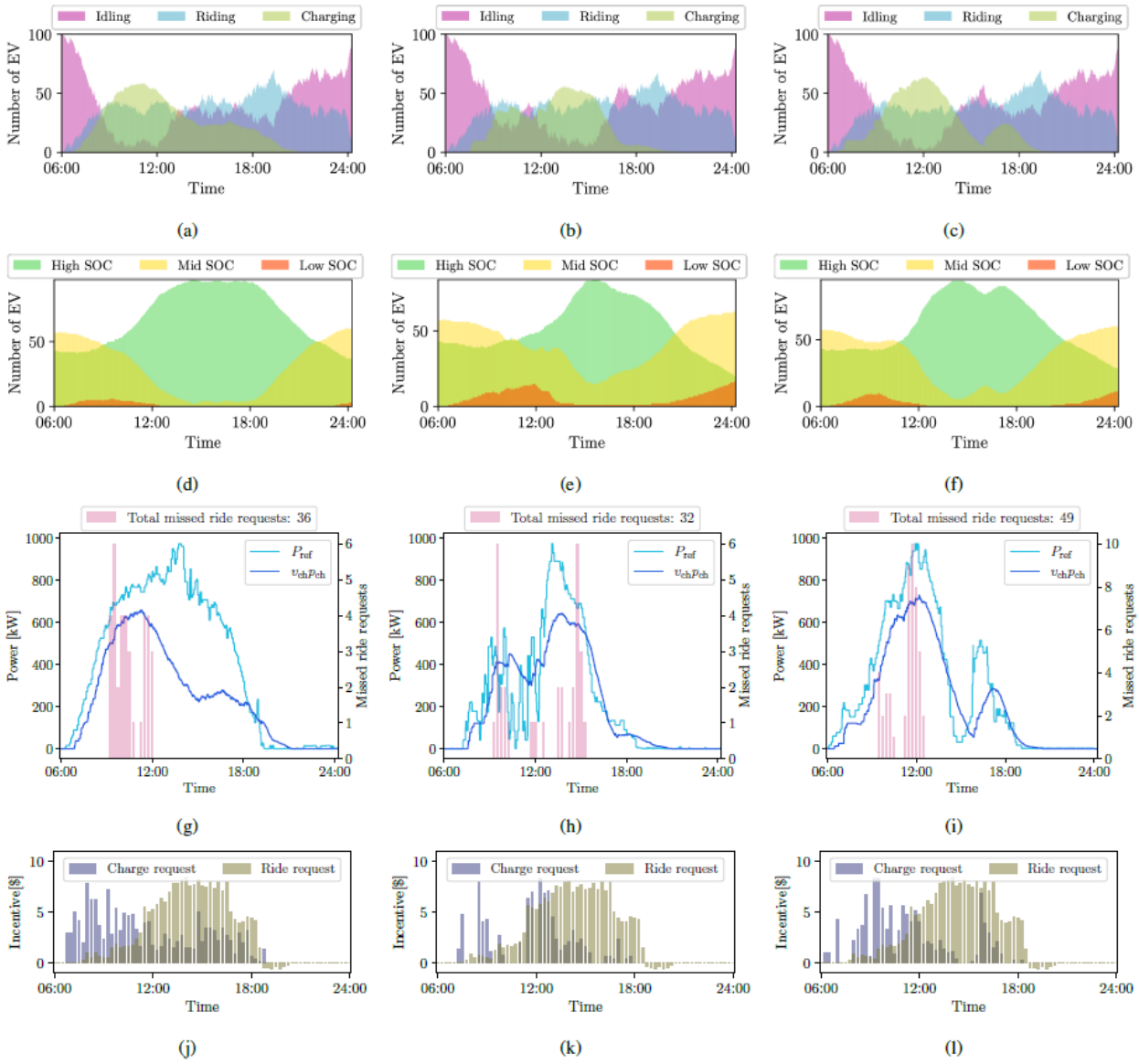


Figure 7.6: Availability of EVs during the day (1st-row), SOC time-evolution (2nd-row), charging profiles together with the total number of missed ride requests (3rd-row) and incentives (4th-row) during the day, given three different weather scenarios: sunny (a)-(d)-(g)-(j), cloudy morning (b)-(e)-(h)-(k), and cloudy afternoon (c)-(f)-(i)-(l). The customers' willingness to ride-share is set here to 75%. The initial SOC of each EV is randomly set within 10% and 100% of the battery capacity. The PV-generation (P_{ref}) and charging profiles ($v_{\text{ch}}p_{\text{ch}}$) are obtained summing over all charging stations.

the QoS of the *businesses-as-usual* case while improving the impact of the EV charging on the grid. The latter is achieved by scheduling EVs to charge via charge requests issued by the power utility company so that the peak of renewable generation is covered, reducing losses for the power utility

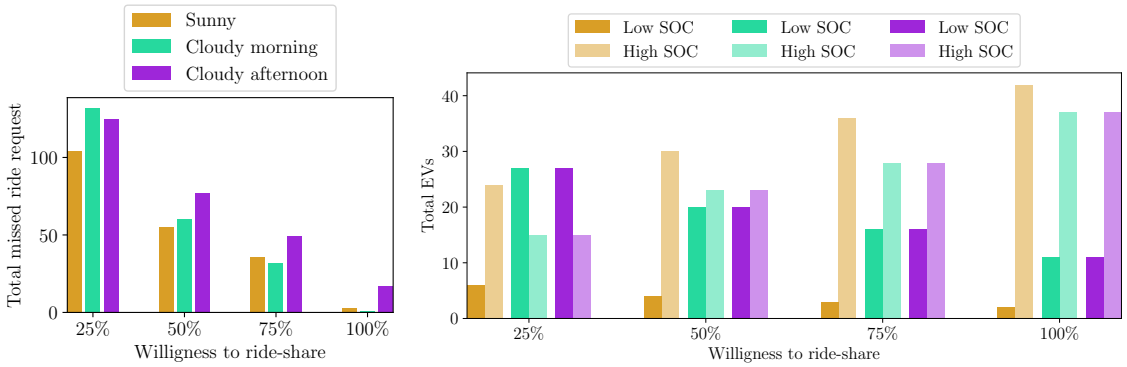


Figure 7.7: Total number of missed ride requests during the day (a) and number of EVs with low/high SOC by the end of the day (b), for given customers' willingness to ride-share and for three different weather scenarios: sunny, cloudy morning, and cloudy afternoon. The initial SOC of each EV is randomly set within 10% and 100% of the battery capacity.

company and improving the power grid load balance. The advantages of our method are multifold: exploit renewable energy resources when available, reduce the need for renewable generation storage at a grid level, and charge a large number of EVs before evening rush hours.

Assuming an increasingly widespread adoption of ride-sharing in the coming years, several implementation strategies have been proposed. To name a few examples representing different perspectives, the design of price mechanisms for ride-sharing [153], a dynamic vehicle-request assignment strategy for autonomous ride-sharing services [9], a linear vehicle-request assignment within a federated optimization architecture[149]. Other approaches design price mechanisms where the charging facilities set prices to control the resource utilization [139], or where the EV charging schedules are influenced by varying energy prices [8]. The coordination of centralized and decentralized strategies for the EV charging is studied in the context of non-cooperative games in works such as [108], where a large EV population is coupled through a common price signal. However, those approaches do not deal with a ride-service provider trying to serve ride requests and respond to utility needs simultaneously.

In this work, we show how to integrate the needs of a power utility company, in the form of maximizing the use of renewable generation, with the users' need to complete trips from an

origin to a destination point. The aim is to find an equilibrium between the objectives of different entities. The reason for establishing a charging schedule can be found in the *business-as-usual* case presented in Figure 7.3, where no charge requests are submitted and the EVs connect to the power grid whenever their SOC is below a certain threshold. Note that in this case the charging profile is wider and shifted towards the evening hours with respect to Figure 7.4, implying worsened grid loading during an already critical moment of the day, according to the data dashboard of New York ISO for the Manhattan area[118]. With our method, *case 1* in the Results section, the charging profile follows the renewable generation profile instead; this contributes to alleviating the so-called "duck curve" by promoting charging during high renewable generation periods. We use 63.47% of the available renewable power to schedule the EV charging (see Figure 7.4) while retaining the same QoS as in the *business-as-usual* case. Additionally, when we introduce the ride-sharing option, as presented in *case 2*, we further improve the QoS while still relying on renewable resources for the EV charging, as presented in Figure 7.6-7.7 and Table 7.2. Our findings indicate that ride-sharing is a must in order to fully benefit from an EV fleet powered by renewables. Improved QoS, a larger number of EVs with high SOC, and less traffic congestion are just some of the multiple advantages of ride-sharing.

Note that in this work the power utility company generates only positive incentives since a charge request can only be sent if renewable power is available. The negative incentives (disincentives) with respect to the ride requests seen in Figure 7.4-7.6 correspond to the scenario where the users did not submit a bid (tip) and/or some EVs sent a negative incentive to the ride-service provider to indicate that they were far away from the user's drop-off point.

For completeness, we simulate a case where we considered a fleet that consists of regular fossil-fuel vehicles, as shown in Figure 7.2. The number of missed ride requests, in this case, that does not include the ride-sharing option, reduces to 12. In order to achieve the same result with EVs, a larger fleet with 25 additional EVs would be needed, or alternatively, the customers' willingness to ride-share should be above 85%. On the other hand, our mobility model is significantly more sustainable since it is entirely powered by renewable energy resources. We emphasize that we can

make the EV charging schedule match the renewable generation profile at each charging facility through our bargaining mechanism, as shown in Figure 7.5. Moreover, we note that our results may further improve in a scenario where fast-charging is adopted. In this work, we take a conservative approach and consider level 2 chargers only, but if we consider level 3 EV chargers, also called DC fast chargers [161], we could achieve a considerable improvement in the QoS. For example, if we assume a charging rate of 1.15 kWh per minute then the time needed to recharge a depleted battery would reduce to 45 minutes, and the performance metrics for *case 1* on a sunny day would improve, resulting in a QoS of 99.8% and a PL of 35.4%.

We acknowledge that our formulation comes with limitations. First, we consider charge requests as an input that depends on the surplus of renewable generation in areas where the charging facilities are located. With coordinated efforts between the ride-service provider and the power utility company, it would be feasible to displace the EV charging peak to a time period with minimal arrival of ride requests, mitigating the adverse effects of charging on the QoS, as currently can be seen in Figure 7.4. However, we note that the proposed approach can be extended to consider the case where the power utility company incentivizes the EVs to provide other services, such as ramping services and load balancing; this is performed by replacing the renewable generation with a different power profile in the proposed method. Second, the performance of our algorithm relies on the tuning of multiple parameters that affect the outcome for the ride-service provider, by improving the QoS, or for the power utility company, by increasing the consumption of renewable generation; then, the balance between the objectives of both companies is not a trivial task. Our algorithm setup depends on external variables such as the user bid, energy prices, initial SOC of EVs, and arrival time of ride and charge requests, thus, the results can differ considerably from one case to another. In particular, we recognize the importance of setting the value for the maximum bid allowed to users, denoted by $h_{\max}$. If users can bid high enough to request a trip, ride requests will be prioritized over charge requests increasing the PL for the power utility company. Otherwise, if users do not offer bids at all, and the financial incentives associated with the charge requests are large enough, the QoS can deteriorate due to the preference of the ride-service provider to prioritize

charging.

7.3 Conclusions

In this chapter, we presented a novel approach based on a game-theoretic framework that aims to balance the objectives of a ride-service provider, an EV fleet, and a power utility company through a bargaining mechanism. The proposed approach influences the EV charging strategy, which in turn controls a financial incentive sent to the ride-service provider. This mechanism is designed to solve a linear assignment problem, determining the optimal way to meet both ride and charge requests. In this work, the interaction between agents was studied using a Gauss-Seidel algorithm.

Chapter 8

Future Directions

In this dissertation, we presented feedback-based first-order algorithms focusing on applications where renewable generation and controllable DERs interact with the grid. In Part I (Chapter 2 and 3) we proposed feedback-based projected gradient descent algorithms to solve a time-varying optimization problem associated with a system modeled with an algebraic map. Specifically, in Chapter 2, we focused on customizing the learning process of the function $U(\mathbf{x})$. Additionally, the incorporation of the learning of the function $C_t(\mathbf{y})$ using similar techniques can also be considered. Analyzing and defining simulation tests that include the learning of function $C_t(\mathbf{y})$ is seen as a potential future direction. Furthermore, studying the learning of time-varying functions during the execution of the online algorithm can be a research direction for this work, as discussed in Section 3.4.

In Chapter 3, we presented an online consensus-based algorithm to solve a time-varying optimization problem associated with a network of systems shared by multiple users. The setting presented in this chapter assumed that the function $\mathbf{x} \mapsto U(\mathbf{x})$ is static but unknown, studying of time-varying functions is a further extension of this work. In this area, our main idea is to leverage the tools presented in [25], where two extensions of the classical GP upper confidence bound (GP-UCB) algorithm were proposed to concurrently learn and maximize a time-varying function. Although, this chapter do not provide an analysis for the case where the time-varying GP-UCB of [25] is utilized in lieu of (2.5a) in our framework, we provided some preliminary numerical simulations to illustrate the idea in Section 3.4. Additionally, in this chapter we assumed that

the function $\mathbf{y} \mapsto C_t(\mathbf{y})$ is known, a further extension of the work presented in Chapters2 and 3 is to analyze and evaluate the performance of our proposed algorithm where this function is also unknown. Finally, we note that the problem formulation and solution approach proposed in this chapter can be naturally extended to the case where the number of users interacting with a device changes over time, a simulation setting to test this scenario can be an extension of the theoretical results presented here.

The Part II of the dissertation is focused on different applications where power systems interact with controllable DERs. In Chapter 5, we presented a Gaussian process based predictive control (GP-PC) for a grid-interactive multi-zone commercial building where we showed that the GP-PC is able to control the indoor temperature per zone in a building under normal operation and demand response scenarios. In this work, we utilized one Gaussian processes (GPs) per zone in the building, employing a multiple input-single output Gaussian process regression approach. A further extension of this work could involve using a multiple input-multiple output Gaussian process regression [32]. Additionally, in a similar vein, exploring the direct learning of the building's power consumption could be achieved through multiple input-single output Gaussian process regression, as demonstrated in this work, but focusing on learning power dynamics instead of temperature dynamics. These two approaches have the potential to significantly reduce the computational time required by the proposed controller in this chapter.

In Chapter 6, we proposed a method to estimate sensitivities in a power grid by leveraging a nuclear norm minimization approach as well as sparsity-promoting regularization functions. In this work, we focused on considering measurements taken at regular intervals, specifically at times $t_{k-m+1}, \ldots, t_k$. However, it is worth exploring the simulation performance when measurements are collected at irregular intervals, which is a potential avenue for further investigation. Furthermore, an extension of this work could involve incorporating reactive power into the linear sensitivity formulation [154], as well as studying topology change identification in transmission and distribution power systems [54].

The final chapter of this dissertation presented a novel approach based on a game-theoretic

framework that aims to balance the objectives of a ride-service provider, an EV fleet, and a power utility company through a bargaining mechanism. Based on the results of this chapter, a potential research direction is to investigate macroscopic network-flow-based approaches for the EV charging schedule problem using bilevel optimization settings. In such problems, the optimality conditions of a sub-problem are incorporated as constraints in another problem. The main problem is referred to as the *outer problem*, while the sub-problem is known as the *inner problem*. In the context of the EV charging schedule problem, it can be formulated as a bilevel optimization problem, where a ride service provider and a utility company interact to achieve their individual objectives. For instance, the ride-service provider aims to manage a fleet of EVs to: (i) maximize profits by setting prices for rides and meeting customer demand at each node, and (ii) minimize operational and charging costs of the EV fleet by optimizing charging and routing decisions. This can be considered the outer problem. On the other hand, the power utility problem is associated with the inner problem and focuses on minimizing losses caused by unused renewable energy.

Exploring bilevel optimization formulations can provide a comprehensive framework to address the interactions and objectives of both the ride-service provider and the power utility company, taking into account factors such as profit maximization, routing, charging decisions, and renewable energy utilization. In this macroscopic model, where vehicles are represented as a continuous quantity, describing the aggregate traffic flow rather than individual EV movements, a further research direction is to analyze the theoretical and/or numerical performance of an algorithm that solves the bilevel problem, combining the individual objectives of the two companies. Furthermore, the problem of routing and charging for EV drivers can be formulated as a resource-constrained shortest-path problem [112], incorporating local marginal prices obtained from a power flow solution [83] in the bilevel optimization formulation. This approach would create a more realistic simulation setting, showcasing the advantages of the proposed method in the context of the power grid.

Bibliography

[1] Abu Abdullah, Ashish Agalgaonkar, and Kashem Muttaqi. Probabilistic load flow incorporating correlation between time-varying electricity demand and renewable power generation. Renewable energy, 55:532–543, 2013.

[2] Robert J. Adler. The Geometry of Random Fields, volume 62. Society for Industrial and Applied Mathematics, SIAM, 1981.

[3] H. Ahmadi, J. R. Martí, and A. Alsubaie. Sensitivity factors for distribution systems. In 2015 IEEE Power Energy Society General Meeting, pages 1–5, July 2015.

[4] Amirhossein Ajalloeian, Andrea Simonetto, and Emiliano Dall'Anese. Inexact online proximal-gradient method for time-varying convex optimization. In 2020 American Control Conference (ACC), pages 2850–2857, 2020.

[5] Albert Akhriev, Jakub Marecek, and Andrea Simonetto. Pursuit of low-rank models of time-varying matrices robust to sparse and measurement noise. Proceedings of the AAAI Conference on Artificial Intelligence, 34(04):3171–3178, Apr. 2020.

[6] A. Al-Digs, B. Chen, S. V. Dhople, and Y. Christine Chen. A data-driven convex-optimization method for estimating load changes. In 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pages 1–5, Nov 2019.

[7] Abdullah Al-Digs, Bo Chen, Sairaj V. Dhople, and Yu Christine Chen. A data-driven convex-optimization method for estimating load changes. In 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pages 1–5, 2019.

[8] Mahnoosh Alizadeh, Hoi-To Wai, Mainak Chowdhury, Andrea Goldsmith, Anna Scaglione, and Tara Javidi. Optimal pricing to manage electric vehicles in coupled power and transportation networks. IEEE Transactions on Control of Network Systems, 4(4):863–875, 2017.

[9] Javier Alonso-Mora, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proceedings of the National Academy of Sciences, 114(3):462–467, 2017.

[10] Sangjae Bae, Sang Min Han, and Scott Moura. System analysis and optimization of human-actuated dynamical systems. In 2018 Annual American Control Conference (ACC), pages 4539–4545, 2018.

[11] Shahab Bahrami, Vincent W. S. Wong, and Jianwei Huang. An online learning algorithm for demand response in smart grid. IEEE Transactions on Smart Grid, 9(5):4712–4725, 2018.

[12] Andrea Bajcsy, Anand Siththaranjan, Claire J. Tomlin, and Anca D. Dragan. Analyzing human models that adapt online. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 2754–2760, 2021.

[13] Kyri Baker, Andrey Bernstein, Emiliano Dall'Anese, and Changhong Zhao. Network-cognizant voltage droop control for distribution grids. IEEE Transactions on Power Systems, 33(2):2098–2108, 2017.

[14] Nicola Bastianello, Liam Madden, Ruggero Carli, and Emiliano Dall'Anese. A stochastic operator framework for inexact static and online optimization. arXiv preprint arXiv:2105.09884, 2021.

[15] A. Beck. First-Order Methods in Optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

[16] Gabriel Behrendt and Matthew Hale. Technical report: A totally asynchronous algorithm for tracking solutions to time-varying convex optimization problems. arXiv preprint arXiv:2110.06705, 2021.

[17] Breno A Beirigo, Rudy R Negenborn, Javier Alonso-Mora, and Frederik Schulte. A business class for autonomous mobility-on-demand: Modeling service quality contracts in dynamic ridesharing systems. Transportation Research Part C: Emerging Technologies, 136:103520, 2022.

[18] Felix Berkenkamp and Angela P. Schoellig. Safe and robust learning control with gaussian processes. In 2015 European Control Conference (ECC), pages 2496–2501, 2015.

[19] Andrey Bernstein, Emiliano Dall'Anese, and Andrea Simonetto. Online primal-dual methods with measurement feedback for time-varying convex optimization. IEEE Transactions on Signal Processing, 67(8):1978–1991, 2019.

[20] Andrey Bernstein and Emiliano Dall'Anese. Real-time feedback-based optimization of distribution grids: A unified approach. IEEE Transactions on Control of Network Systems, 6(3):1197–1209, 2019.

[21] Dimitri P Bertsekas and John N Tsitsiklis. Gradient convergence in gradient methods with errors. SIAM Journal on Optimization, 10(3):627–642, 2000.

[22] Siddharth Bhela, Vassilis Kekatos, and Sriharsha Veeramachaneni. Smart inverter grid probing for learning loads: Part I - identifiability analysis. IEEE Transactions on Power Systems, 34(5):3527–3536, 2019.

[23] J. Bijsterbosch and A. Volgenant. Solving the rectangular assignment problem and applications. Annals of Operations Research, 181:443–462, 2010.

[24] Adam B. Birchfield, Ti Xu, Kathleen M. Gegner, Komal S. Shetye, and Thomas J. Overbye. Grid structural characteristics as validation criteria for synthetic networks. IEEE Transactions on Power Systems, 32(4):3258–3265, 2017.

[25] Ilija Bogunovic, Jonathan Scarlett, and Volkan Cevher. Time-varying gaussian process bandit optimization. In Arthur Gretton and Christian C. Robert, editors, Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, volume 51 of Proceedings of Machine Learning Research, pages 314–323, Cadiz, Spain, 09–11 May 2016. PMLR.

[26] Saverio Bolognani and Florian Dörfler. Fast power system analysis via implicit linearization of the power flow manifold. In 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), pages 402–409. IEEE, 2015.

[27] Saverio Bolognani and Sandro Zampieri. A distributed control strategy for reactive power compensation in smart microgrids. IEEE Transactions on Automatic Control, 58(11):2818–2833, 2013.

[28] Claudia Bongiovanni, Mor Kaspi, Jean-Francois Cordeau, and Nikolas Geroliminis. A machine learning-driven two-phase metaheuristic for autonomous ridesharing operations. Transportation Research Part E: Logistics and Transportation Review, 165:102835, 2022.

[29] Anjan Bose and Thomas J. Overbye. Electricity transmission system research and development: Grid operations. Transmission Innovation Symposium: Modernizing the U.S. Electrical Grid, U.S. Department of Energy, 2021.

[30] Georg Brandstätter, Claudio Gambella, Markus Leitner, Enrico Malaguti, Filippo Masini, Jakob Puchinger, Mario Ruthmair, and Daniele Vigo. Overview of optimization problems in electric car-sharing system design and management. Dynamic perspectives on managerial decision making: Essays in honor of Richard F. Hartl, pages 441–471, 2016.

[31] Cecilia Briceno-Garmendia, Qiao Wenxin, and Foster Vivien. The economics of electric vehicles for passenger transportation. Sustainable Infrastructure Series, 2023.

[32] Wessel Bruinsma, Eric Perim, William Tebbutt, Scott Hosking, Arno Solin, and Richard Turner. Scalable exact inference in multi-output Gaussian processes. In Hal Daumé III and Aarti Singh, editors, Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 1190–1201. PMLR, 13–18 Jul 2020.

[33] Rainer E. Burkard and Eranda Çela. Linear Assignment Problems and Extensions, pages 75–149. Springer US, Boston, MA, 1999.

[34] Luke Butler, Tan Yigitcanlar, and Alexander Paz. Barriers and risks of mobility-as-a-service (MaaS) adoption in cities: A systematic review of the literature. Cities, 109:103036, 2021.

[35] Patrick M. Bösch, Felix Becker, Henrik Becker, and Kay W. Axhausen. Cost-based analysis of autonomous mobility services. Transport Policy, 64:76–91, 2018.

[36] Emmanuel J Candès and Yaniv Plan. Matrix completion with noise. Proceedings of the IEEE, 98(6):925–936, 2010.

[37] Gang Cao, Edmund M-K Lai, and Fakhrul Alam. Gaussian process model predictive control of unknown non-linear systems. IET Control Theory Applications, 11(5):703–713, 2017.

[38] Volkan Cevher, Stephen Becker, and Mark Schmidt. Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics. IEEE Signal Processing Magazine, 31(5):32–43, 2014.

[39] Gautham Ram Chandra Mouli, Mahdi Kefayati, Ross Baldick, and Pavol Bauer. Integrated PV charging of EV fleet based on energy prices, V2G, and offer of reserves. IEEE Transactions on Smart Grid, 10(2):1313–1325, 2019.

[40] Chin-Yao Chang, Marcello Colombino, Jorge Cortés, and Emiliano Dall'Anese. Saddle-flow dynamics for distributed feedback-based optimization. IEEE Control Systems Letters, 3(4):948–953, 2019.

[41] Shibo Chen, Shanshan Feng, Zhenwei Guo, and Zaiyue Yang. Trilevel optimization model for competitive pricing of electric vehicle charging station considering distribution locational marginal price. IEEE Transactions on Smart Grid, 13(6):4716–4729, 2022.

[42] Tao Chen, Bowen Zhang, Hajir Pourbabak, Abdollah Kavousi-Fard, and Wencong Su. Optimal routing and charging of an electric vehicle fleet for high-efficiency dynamic transit systems. IEEE Transactions on Smart Grid, 9(4):3563–3572, 2018.

[43] Tianyi Chen and Georgios B Giannakis. Bandit convex optimization for scalable and dynamic IoT management. IEEE Internet of Things Journal, 6(1):1276–1286, 2018.

[44] Y. C. Chen, S. V. Dhople, A. D. Domínguez-García, and P. W. Sauer. Generalized injection shift factors. IEEE Transactions on Smart Grid, 8(5):2071–2080, Sep. 2017.

[45] Yu Christine Chen, Alejandro D. Domínguez-García, and Peter W. Sauer. Measurement-based estimation of linear sensitivity distribution factors and applications. IEEE Transactions on Power Systems, 29(3):1372–1382, 2014.

[46] Yu Christine Chen, Jianhui Wang, Alejandro D. Domínguez-García, and Peter W. Sauer. Measurement-based estimation of the power flow jacobian matrix. IEEE Transactions on Smart Grid, 7(5):2507–2515, 2016.

[47] Yue Chen, Andrey Bernstein, Adithya Devraj, and Sean Meyn. Model-free primal-dual methods for network optimization with application to real-time optimal power flow. In 2020 American Control Conference (ACC), pages 3140–3147, 2020.

[48] Rohit H. Chintala and Bryan P. Rasmussen. Automated Multi-Zone Linear Parametric Black Box Modeling Approach for Building HVAC Systems. In Dynamic Systems and Control Conference, volume 2. American Society of Mechanical Engineers, 2015.

[49] Marcello Colombino, John W Simpson-Porco, and Andrey Bernstein. Towards robustness guarantees for feedback-based optimization. In IEEE Conference on Decision and Control, pages 6207–6214, 2019.

[50] Drury B. Crawley, Curtis O. Pedersen, Linda K. Lawrie, and Frederick C. Winkelmann. Energyplus: Energy simulation program. ASHRAE Journal, 42:49–56, 2000.

[51] Stella C. Dafermos and Frederick T. Sparrow. The traffic assignment problem for a general network. Journal of Research of the National Bureau of Standards - B. Mathematical Sciences, 73B(2), April-June 1969.

[52] Emiliano Dall'Anese, Andrea Simonetto, Stephen Becker, and Liam Madden. Optimization and learning with information streams: Time-varying algorithms and applications. IEEE Signal Processing Magazine, 37(3):71–83, 2020.

[53] Emiliano Dall'Anese and Andrea Simonetto. Optimal power flow pursuit. IEEE Transactions on Smart Grid, 9(2):942–952, 2018.

[54] Deepjyoti Deka, Scott Backhaus, and Michael Chertkov. Structure learning in power distribution networks. IEEE Transactions on Control of Network Systems, 5(3):1061–1074, 2018.

[55] Rishabh Dixit, Amrit Singh Bedi, Ruchi Tripathi, and Ketan Rajawat. Online learning with inexact proximal online gradient descent algorithms. IEEE Transactions on Signal Processing, 67(5):1338–1352, 2019.

[56] Ján Drgoňa, Javier Arroyo, Iago Cupeiro Figueroa, David Blum, Krzysztof Arendt, Donghun Kim, Enric Perarnau Ollé, Juraj Oravec, Michael Wetter, Draguna L. Vrabie, and Lieve Helsen. All you need to know about model predictive control for buildings. Annual Reviews in Control, 50:190–232, 2020.

[57] David Duvenaud. Automatic model construction with gaussian processes. Doctoral thesis, 2014.

[58] Environmental Protection Agency (EPA). Report on sources of greenhouse gas emissions. `https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions`, 2018. Accessed: 2023-05-11.

[59] Filippo Fabiani, Andrea Simonetto, and Paul J. Goulart. Learning equilibria with personalized incentives in a class of nonmonotone games. In 2022 European Control Conference (ECC), pages 2179–2184, 2022.

[60] Francisco Facchinei and Jong-Shi Pang. Nash equilibria: the variational approach. Convex optimization in signal processing and communications, page 443, 2010.

[61] Mahyar Fazlyab, Cameron Nowzari, George J Pappas, Alejandro Ribeiro, and Victor M Preciado. Self-triggered time-varying convex optimization. In IEEE Conference on Decision and Control, pages 3090–3097, 2016.

[62] Andres Fielbaum, Maximilian Kronmueller, and Javier Alonso-Mora. Anticipatory routing methods for an on-demand ridepooling mobility system. Transportation, 49(6):1921–1962, 2022.

[63] Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '05, page 385–394, USA, 2005. Society for Industrial and Applied Mathematics.

[64] M. Fukushima. Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems. Department of Applied Mathematics and Physics, Faculty of Engineering, 53:99–110, 1992.

[65] Lingwen Gan, Ufuk Topcu, and Steven H. Low. Optimal decentralized protocol for electric vehicle charging. IEEE Transactions on Power Systems, 28(2):940–951, 2013.

[66] Shuang Gao, K. T. Chau, Chunhua Liu, Diyun Wu, and C. C. Chan. Integrated energy management of plug-in electric vehicles in power grid with renewables. IEEE Transactions on Vehicular Technology, 63(7):3019–3027, 2014.

[67] Nikolaos Gatsis and Georgios B. Giannakis. Residential load control: Distributed scheduling and convergence with lost ami messages. IEEE Transactions on Smart Grid, 3(2):770–786, 2012.

[68] Pere Giménez-Febrer, Alba Pagès-Zamora, and Georgios B Giannakis. Matrix completion and extrapolation via kernel regression. IEEE Transactions on Signal Processing, 67(19):5004–5017, 2019.

[69] Sebastián González, Felipe Feijoo, Franco Basso, Vignesh Subramanian, Sriram Sankaranarayanan, and Tapas K. Das. Routing and charging facility location for EVs under nodal pricing of electricity: A bilevel model solved using special ordered set. IEEE Transactions on Smart Grid, 13(4):3059–3068, 2022.

[70] Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. SGD: General analysis and improved rates. In International Conference on Machine Learning, pages 5200–5209. PMLR, 2019.

[71] Krishna Murthy Gurumurthy and Kara M. Kockelman. Modeling Americans' autonomous vehicle preferences: A focus on dynamic ride-sharing, privacy & long-distance mode choices. Technological Forecasting and Social Change, 150:119792, 2020.

[72] Eric C Hall and Rebecca M Willett. Online convex optimization in dynamic environments. IEEE Journal of Selected Topics in Signal Processing, 9(4):647–662, 2015.

[73] Trevor Hastie, Robert Tibshirani, and Jerome. Friedman. The elements of statistical learning. Springer, 2009.

[74] Adrian Hauswirth, Saverio Bolognani, Gabriela Hug, and Florian Dörfler. Optimization algorithms as robust feedback controllers. arXiv preprint arXiv:2103.11329, 2021.

[75] David A Hensher, Corinne Mulley, Chin Ho, Yale Wong, Goran Smith, and John D Nelson. Understanding Mobility as a Service (MaaS): Past, present and future. Elsevier, 2020.

[76] Kenji Hirata, Joao P Hespanha, and Kenko Uchida. Real-time pricing leading to optimal operation under distributed decision makings. In American Control Conference, pages 1925–1932, 2014.

[77] Ji Hyun Hong and Xiaoxi Liu. The optimal pricing for green ride services in the ride-sharing economy. Transportation Research Part D: Transport and Environment, 104:103205, 2022.

[78] International Energy Agency (IEA). Building energy performance metrics. `https://www.iea.org/reports/building-energy-performance-metrics`, 2015. Accessed: 2023-05-11, License: CC BY 4.0.

[79] International Energy Agency (IEA). Power systems in transition: Challenges and opportunities ahead for electricity security. https://iea.blob.core.windows.net/assets/cd69028a-da78-4b47-b1bf-7520cdb20d70/Power_systems_in_transition.pdf, 2020. Accessed: 2023-05-11, License: CC BY 4.0.

[80] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online Optimization : Competing with Dynamic Comparators. In Guy Lebanon and S. V. N. Vishwanathan, editors, Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, volume 38 of Proceedings of Machine Learning Research, pages 398–406, San Diego, California, USA, 09–12 May 2015. PMLR.

[81] Achin Jain, Truong X. Nghiem, Manfred Morari, and Rahul Mangharam. Learning and control using gaussian processes: Towards bridging machine learning and controls for physical systems. In Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS '18, pages 140–149, Piscataway, NJ, USA, 2018. IEEE Press.

[82] Achin Jain, Francesco Smarra, Madhur Behl, and Rahul Mangharam. Data-driven model predictive control with regression trees—an application to building energy management. ACM Trans. Cyber-Phys. Syst., 2(1), January 2018.

[83] Xuegang (Jeff) Ban, Maged Dessouky, Jong-Shi Pang, and Rong Fan. A general equilibrium model for transportation systems with e-hailing services and flow congestion. Transportation Research Part B: Methodological, 129:273–304, 2019.

[84] Zhong-Ping Jiang and Yuan Wang. Input-to-state stability for discrete-time nonlinear systems. Automatica, 37(6):857–869, 2001.

[85] Andrew L. Johnson and Daniel R. Jiang. Shape constraints in economics and operations research. Statistical Science, pages 527 – 546, 2018.

[86] Vohra K., Vodonos A., Schwartz J., Marais E.A., Sulprizio M.P., and Mickley L.J. Global mortality from outdoor fine particle pollution generated by fossil fuel combustion: Results from GEOS-Chem. Environmental Research, 195:110754, 2021.

[87] J. Kocijan, R. Murray-Smith, C.E. Rasmussen, and A. Girard. Gaussian process model based predictive control. In Proceedings of the 2004 American Control Conference, volume 3, pages 2214–2219 vol.3, 2004.

[88] Juš Kocijan. Modelling and control of dynamic systems using gaussian process models, 2016.

[89] Alec Koppel, Felicia Jakubeic, and Alejandro Ribeiro. A saddle point algorithm for networked online convex optimization. IEEE Transactions on Signal Processing, 63(19):5149–5164, 10 2015.

[90] Jayash Koshal, Angelia Nedić, and Uday V Shanbhag. Multiuser optimization: Distributed algorithms and error analysis. SIAM Journal on Optimization, 21(3):1046–1081, 2011.

[91] K. Kubota and M. Fukushima. Gap function approach to the generalized Nash equilibrium problem. Journal of optimization theory and applications, 144:511–531, 2010.

[92] Niels Lassen, Francesco Goia, Stefano Schiavon, and Jovan Pantelic. Field investigations of a smiley-face polling station for recording occupant satisfaction with indoor climate. Building and Environment, 185:107266, 2020.

[93] Soomin Lee and Michael M Zavlanos. Distributed primal-dual methods for online constrained optimization. In 2016 American Control Conference (ACC), pages 7171–7176. IEEE, 2016.

[94] Bruno Lepri, Jacopo Staiano, David Sangokoya, Emmanuel Letouzé, and Nuria Oliver. The tyranny of data? The bright and dark sides of data-driven decision-making for social good. In Transparent data mining for big and small data, pages 3–24. Springer, 2017.

[95] Antoine Lesage-Landry and Duncan S Callaway. Dynamic and distributed online convex optimization for demand response of commercial buildings. IEEE Control Systems Letters, 2020.

[96] Antoine Lesage-Landry and Joshua A Taylor. Setpoint tracking with partially observed loads. IEEE Transactions on Power Systems, 33(5):5615–5627, 2018.

[97] D. Li, W. Chiu, H. Sun, and H. V. Poor. Multiobjective optimization for demand side management program in smart grid. IEEE Transactions on Industrial Informatics, 14(4):1482–1490, April 2018.

[98] Jueyou Li, Chuanye Gu, Zhiyou Wu, and Tingwen Huang. Online learning algorithm for distributed convex optimization with time-varying coupled constraints and bandit feedback. IEEE Transactions on Cybernetics, 52(2):1009–1020, 2022.

[99] Yanying Li and Tom Voege. Mobility as a service (MaaS): Challenges of implementation and policy required. Journal of transportation technologies, 7(2):95–106, 2017.

[100] Lars Lindemann, Alexander Robey, Lejun Jiang, Stephen Tu, and Nikolai Matni. Learning robust output control barrier functions from safe expert demonstrations. arXiv preprint arXiv:2111.09971, 2021.

[101] Liansheng Liu, Fanxin Kong, Xue Liu, Yu Peng, and Qinglong Wang. A review on electric vehicles interacting with renewable energy in smart grid. Renewable and Sustainable Energy Reviews, 51:648–661, 2015.

[102] M. Liu, G. Chowdhary, B. Castra da Silva, S. Liu, and J. P. How. Gaussian processes for learning and control: A tutorial with examples. IEEE Control Systems Magazine, 38(5):53–86, 2018.

[103] Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. IEEE Signal Processing Magazine, 37(5):43–54, 2020.

[104] Sijia Liu, Xingguo Li, Pin-Yu Chen, Jarvis Haupt, and Lisa Amini. Zeroth-order stochastic projected gradient descent for nonconvex optimization. In IEEE GlobalSIP, pages 1179–1183, 2018.

[105] Yang Liu, Nanpeng Yu, Wei Wang, Xiaohong Guan, Zhanbo Xu, Bing Dong, and Ting Liu. Coordinating the operations of smart buildings in smart grids. Applied Energy, 228:2510–2525, 2018.

[106] X. Luo, Y. Zhang, and M. M. Zavlanos. Socially-aware robot planning via bandit human feedback. In 2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS), pages 216–225, 2020.

[107] Si Lv, Sheng Chen, and Zhinong Wei. Coordinating urban power-traffic networks: A subsidy-based Nash–Stackelberg–Nash game model. IEEE Transactions on Industrial Informatics, 19(2):1778–1790, 2023.

[108] Zhongjing Ma, Duncan S. Callaway, and Ian A. Hiskens. Decentralized charging control of large populations of plug-in electric vehicles. IEEE Transactions on Control Systems Technology, 21(1):67–78, 2013.

[109] M. Marchi, J. Bunton, B. Gharesifard, and P. Tabuada. Safety and stability guarantees for control loops with deep learning perception. IEEE Control Systems Letters, 6:1286–1291, 2022.

[110] Ivan Markovsky and Sabine Van Huffel. Overview of total least-squares methods. Signal processing, 87(10):2283–2302, 2007.

[111] G. Mateos and G. B. Giannakis. Load curve data cleansing and imputation via sparsity and low rank. IEEE Transactions on Smart Grid, 4(4), Dec 2013.

[112] A Migdalas. Bilevel programming in traffic planning: Models, methods and challenge. Journal of Global Optimization, 7:381–405, 1995.

[113] Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In IEEE Conference on Decision and Control, pages 7195–7201, 2016.

[114] Sirajum Munir, John A Stankovic, Chieh-Jan Mike Liang, and Shan Lin. Cyber physical system challenges for human-in-the-loop control. In International Workshop on Feedback Computing, 2013.

[115] New York City Open Data. NYC OpenData. `https://data.cityofnewyork.us/Housing-Development/Solar/wq45-53zs`. Accessed: 2023-03-7.

[116] Truong X. Nghiem and Colin N. Jones. Data-driven demand response modeling and control of buildings with gaussian processes. In 2017 American Control Conference (ACC), pages 2919–2924, 2017.

[117] Ivano Notarnicola, Andrea Simonetto, Francesco Farina, and Giuseppe Notarstefano. Distributed personalized gradient tracking with convex parametric models. IEEE Transactions on Automatic Control, 2022.

[118] New York ISO Independent System Operator (NYISO). Real-Time Dashboard, markets. `https://www.nyiso.com/real-time-dashboard`. Accessed: 2023-03-17.

[119] Ana M. Ospina, Nicola Bastianello, and Emiliano Dall'Anese. Feedback-based optimization with sub-weibull gradient errors and intermittent updates. IEEE Control Systems Letters, 6:2521–2526, 2022.

[120] Ana M. Ospina, Yue Chen, Andrey Bernstein, and Emiliano Dall'Anese. Learning-based demand response in grid-interactive buildings via gaussian processes. Electric Power Systems Research, 211:108406, 2022.

[121] Ana M. Ospina and Emiliano Dall'Anese. Data-driven and online estimation of linear sensitivity distribution factors: A low-rank approach, 2023.

[122] Ana M. Ospina, Andrea Simonetto, and Emiliano Dall'Anese. Time-varying optimization of networked systems with human preferences. IEEE Transactions on Control of Network Systems, 10(1):503–515, 2023.

[123] Ana M. Ospina, Andrea Simonetto, and Emiliano Dall'Anese. Personalized demand response via shape-constrained online learning. In 2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), pages 1–6, 2020.

[124] P. Palensky and D. Dietrich. Demand side management: Demand response, intelligent energy systems, and smart loads. IEEE Transactions on Industrial Informatics, 7(3):381–388, 2011.

[125] Nadia Panossian, Matteo Muratori, Bryan Palmintier, Andrew Meintz, Timothy Lipman, and Keith Moffat. Challenges and opportunities of integrating electric vehicles in electricity distribution systems. Current sustainable/renewable energy reports, 9(2):27–40, 2022.

[126] Christos H. Papadimitriou and Kenneth Steiglitz. Combinatorial optimization: algorithms and complexity, pages 1–496. Prentice Hall, Mineola (N.Y.) : Dover, 1998.

[127] Priti Paudyal, Shibani Ghosh, Santosh Veda, Deepak Tiwari, and Jal Desai. EV hosting capacity analysis on distribution grids. In 2021 IEEE Power & Energy Society General Meeting (PESGM), pages 1–5, 2021.

[128] Elisabetta Perotti, Ana M. Ospina, Gianluca Bianchin, Andrea Simonetto, and Emiliano Dall'Anese. Towards the decarbonization of the mobility sector: Promoting renewable-based charging in green ride-sharing, 2023.

[129] J. Peschon, D. S. Piercy, W. F. Tinney, and O. J. Tveit. Sensitivity in power systems. IEEE Transactions on Power Apparatus and Systems, PAS-87(8):1687–1696, Aug 1968.

[130] A Yu Popkov. Gradient methods for nonstationary unconstrained optimization problems. Automation and Remote Control, 66(6):883–891, 2005.

[131] Wenjie Qiao, Yinghua Han, Qiang Zhao, Fangyuan Si, and Jinkuan Wang. A distributed coordination method for coupled traffic-power network equilibrium incorporating behavioral theory. IEEE Transactions on Vehicular Technology, 71(12):12588–12601, 2022.

[132] Salar Rahili and Wei Ren. Distributed continuous-time convex optimization with time-varying cost functions. IEEE Transactions on Automatic Control, 62(4):1590–1605, 2017.

[133] Carl Edward Rasmussen. Gaussian processes for machine learning. In Gaussian processes for machine learning. MIT Press, 2006.

[134] Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (GPML) toolbox. Journal of Machine Learning Research, 11(100):3011–3015, 2010.

[135] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Rev., 52(3):471–501, Aug. 2010.

[136] Valentin Rigoni and Andrew Keane. Estimation of voltage sensitivities in low voltage feeders with photovoltaics. In 2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), pages 1–6, 2018.

[137] Federico Rossi, Ramon Iglesias, Mahnoosh Alizadeh, and Marco Pavone. On the interaction between autonomous mobility-on-demand systems and the power network: Models and coordination algorithms. IEEE Transactions on Control of Network Systems, 7(1):384–397, 2020.

[138] Ernest Ryu and Stephen Boyd. A primer on monotone operator methods survey. Applied and computational mathematics, 15:3–43, 01 2016.

[139] Cesar Santoyo, Gustav Nilsson, and Samuel Coogan. Resource aware pricing for electric vehicle charging. Automatica, 148:110733, 2023.

[140] Peter W. Sauer. On the formulation of power distribution factors for linear load flow methods. IEEE Transactions on Power Apparatus and Systems, PAS-100(2):764–770, Feb 1981.

[141] Gesualdo Scutari, Daniel P. Palomar, Francisco Facchinei, and Jong-Shi Pang. Convex optimization, game theory, and variational inequality theory. IEEE Signal Processing Magazine, 27(3):35–49, 2010.

[142] Daniel D. Selvaratnam, Iman Shames, Jonathan H. Manton, and Mohammad Zamani. Numerical optimisation of time-varying strongly convex functions subject to time-varying constraints. In IEEE Conference on Decision and Control, pages 849–854, 2018.

[143] Shahin Shahrampour and Ali Jadbabaie. Distributed online optimization in dynamic environments using mirror descent. IEEE Transactions on Automatic Control, 63(3):714–725, 2018.

[144] Suzuna Shibasaki, Masaki Inoue, Mitsuru Arahata, and Vijay Gupta. Weak control approach to consumer-preferred energy management. IFAC-PapersOnLine, 53(2):17083–17088, 2020. 21st IFAC World Congress.

[145] Andrea Simonetto. Smooth strongly convex regression. In 2020 28th European Signal Processing Conference (EUSIPCO), pages 2130–2134, 2021.

[146] Andrea Simonetto, Emiliano Dall'Anese, Santiago Paternain, Geert Leus, and Georgios B Giannakis. Time-varying convex optimization: Time-structured algorithms and applications. Proceedings of the IEEE, 108(11):2032–2048, 2020.

[147] Andrea Simonetto, Emiliano Dall'Anese, Julien Monteil, and Andrey Bernstein. Personalized optimization with user's feedback. Automatica, 131, 2021.

[148] Andrea Simonetto and Geert Leus. Double smoothing for time-varying distributed multiuser optimization. In 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pages 852–856, 2014.

[149] Andrea Simonetto, Julien Monteil, and Claudio Gambella. Real-time city-scale ridesharing via linear assignment problems. Transportation Research Part C, 101:208–232, 2019.

[150] Kevin Spieser, Samitha Samaranayake, and Emilio Frazzoli. Vehicle routing for shared-mobility systems with time-varying demand. In 2016 American Control Conference (ACC), pages 796–802, 2016.

[151] Kevin Spieser, Kyle Ballantyne Treleaven, Rick Zhang, Emilio Frazzoli, Daniel Morton, and Marco Pavone. Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in Singapore. Road Vehicle Automation, pages 229–245, 2014.

[152] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. IEEE Transactions on Information Theory, 58(5):3250–3265, 2012.

[153] David-Maximilian Storch, Marc Timme, and Malte Schröder. Incentive-driven transition to high ride-sharing adoption. Nature Communications, 12, 2021.

[154] Samuel Talkington, Daniel Turizo, Santiago Grijalva, Jorge Fernandez, and Daniel K. Molzahn. Conditions for estimation of sensitivities of voltage magnitudes to complex power injections. IEEE Transactions on Power Systems, pages 1–14, 2023.

[155] Yujie Tang, Krishnamurthy Dvijotham, and Steven Low. Real-time optimal power flow. IEEE Trans. on Smart Grid, 8(6):2963–2973, 2017.

[156] Yujie Tang, Zhaolin Ren, and Na Li. Zeroth-order feedback optimization for cooperative multi-agent systems. In 2020 59th IEEE Conference on Decision and Control (CDC), pages 3649–3656, 2020.

[157] Yujie Tang, Junshan Zhang, and Na Li. Distributed zero-order algorithms for nonconvex multiagent optimization. IEEE Trans. on Control of Network Systems, 8(1):269–281, 2020.

[158] Taxi & Limousine Commission (TLC). TLC Trip Record Data. `https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page`. Accessed: 2023-03-7.

[159] Josh A. Taylor, Sairaj V. Dhople, and Duncan S. Callaway. Power systems without fuel. Renewable & Sustainable Energy Reviews, 57:1322–1336, May 2016.

[160] Andrew R Teel, João P Hespanha, and Anantharaman Subbaraman. Equivalent characterizations of input-to-state stability for stochastic discrete-time systems. IEEE Trans. on Automatic Control, 59(2):516–522, 2013.

[161] Hao Tu, Hao Feng, Srdjan Srdic, and Srdjan Lukic. Extreme fast charging of electric vehicles: A technology overview. IEEE Transactions on Transportation Electrification, 5(4):861–878, 2019.

[162] Berkay Turan and Mahnoosh Alizadeh. Competition in electric autonomous mobility-on-demand systems. IEEE Transactions on Control of Network Systems, 9(1):295–307, 2022.

[163] Berkay Turan, César A. Uribe, Hoi-To Wai, and Mahnoosh Alizadeh. Resilient primal–dual optimization algorithms for distributed resource allocation. IEEE Transactions on Control of Network Systems, 8(1):282–294, 2021.

[164] United Nations, Department of Economic and Social Affairs, Population Division. World urbanization prospects: The 2018 revision. (ST/ESA/SER.A/420), page New York: United Nations, 2019.

[165] Roman Vershynin. High-dimensional probability: An introduction with applications in data science. Cambridge University press, 2018.

[166] Mariia Vladimirova, Stéphane Girard, Hien Nguyen, and Julyan Arbel. Sub-weibull distributions: Generalizing sub-gaussian and sub-exponential properties to heavier tailed distributions. Stat, 9(1), Jan 2020.

[167] Qingsi Wang, Mingyan Liu, and Johanna L Mathieu. Adaptive demand response: Online learning of restless and controlled bandits. In 2014 IEEE International Conference on Smart Grid Communications, pages 752–757, 2014.

[168] Xiaojing. Wang and James O. Berger. Estimating shape constrained functions using Gaussian Processes. SIAM/ASA Journal on Uncertainty Quantification, 4(1):1–25, 2016.

[169] Jacob W Ward, Jeremy J Michalek, and Constantine Samaras. Air pollution, greenhouse gas, and traffic externality benefits and costs of shifting private vehicle travel to ridesourcing services. Environmental Science & Technology, 55(19):13174–13185, 2021.

[170] Tianshu Wei, Shaolei Ren, and Qi Zhu. Deep reinforcement learning for joint datacenter and HVAC load control in distributed mixed-use buildings. IEEE Transactions on Sustainable Computing, 6(3):370–384, 2021.

[171] Wei Wei, Feng Liu, and Shengwei Mei. Charging strategies of EV aggregator under renewable generation and congestion: A normalized Nash equilibrium approach. IEEE Transactions on Smart Grid, 7(3):1630–1641, 2016.

[172] Allan J. Wood and Bruce F. Wollenberg. Power Generation, Operation, and Control. 2nd ed. New York: J. Wiley & Sons, 1996.

[173] Hanchen Xu, Hongbo Sun, Daniel Nikovski, Shoichi Kitamura, and Kazuyuki Mori. Learning dynamical demand response model in real-time pricing program. In 2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), pages 1–5. IEEE, 2019.

[174] Shichao Xu, Yixuan Wang, Yanzhi Wang, Zheng O'Neill, and Qi Zhu. One for many: Transfer learning for building HVAC control. Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, Nov 2020.

[175] Jiajia Yang, Thomas Wiedmann, Fengji Luo, Gangui Yan, Fushuan Wen, and Gail Helen Broadbent. A fully decentralized hierarchical transactive energy framework for charging EVs with local DERs in power distribution systems. IEEE Transactions on Transportation Electrification, 8(3):3041–3055, 2022.

[176] Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In International Conference on Machine Learning, 2016.

[177] Xinlei Yi, Xiuxian Li, Lihua Xie, and Karl H Johansson. Distributed online convex optimization with time-varying coupled inequality constraints. IEEE Transactions on Signal Processing, 68:731–746, 2020.

[178] Miyu Yoshihara, Toru Namerikawa, and Zhihua Qu. Non-cooperative optimization of charging scheduling of electric vehicle via Stackelberg game. In 2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), pages 1658–1663, 2018.

[179] Pengcheng You, John Z. F. Pang, and Steven H. Low. Online station assignment for electric vehicle battery swapping. IEEE Transactions on Intelligent Transportation Systems, 23(4):3256–3267, 2022.

[180] Liang Yu, Yi Sun, Zhanbo Xu, Chao Shen, Dong Yue, Tao Jiang, and Xiaohong Guan. Multi-agent deep reinforcement learning for HVAC control in commercial buildings. IEEE Transactions on Smart Grid, 12(1):407–419, 2021.

[181] Hongcai Zhang, Colin J.R. Sheppard, Timothy E. Lipman, Teng Zeng, and Scott J. Moura. Charging infrastructure demands of shared-use autonomous electric vehicles in urban areas. Transportation Research Part D: Transport and Environment, 78:102210, 2020.

[182] Xiangyu Zhang, Yue Chen, Andrey Bernstein, Rohit Chintala, Peter Graf, Xin Jin, and David Biagioni. Two-stage reinforcement learning policy search for grid-interactive building control. IEEE Transactions on Smart Grid, 2022.

[183] Xiangyu Zhang, Rohit Chintala, Andrey Bernstein, Peter Graf, and Xin Jin. Grid-interactive multi-zone building control using reinforcement learning with global-local policy search. In 2021 American Control Conference (ACC), pages 4155–4162, 2021.

[184] Junbo Zhao, Antonio Gómez-Expósito, Marcos Netto, Lamine Mili, Ali Abur, Vladimir Terzija, Innocent Kamwa, Bikash Pal, Abhinav Kumar Singh, Junjian Qi, Zhenyu Huang, and A. P. Sakis Meliopoulos. Power system dynamic state estimation: Motivations, definitions, methodologies, and future work. IEEE Transactions on Power Systems, 34(4):3188–3198, 2019.

[185] Zihan Zhou, Xiaodong Li, John Wright, Emmanuel Candès, and Yi Ma. Stable principal component pursuit. In 2010 IEEE international symposium on information theory, pages 1518–1522. IEEE, 2010.

[186] Ray Daniel Zimmerman, Carlos Edmundo Murillo-Sánchez, and Robert John Thomas. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Trans. on Power Systems, 26:12–19, 2011.