Scalable Learning and Energy Management for Power Grids

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Liang Zhang

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Prof. Georgios B. Giannakis, Advisor

January, 2019

Acknowledgments

There are so many people to whom I wish to acknowledge and thank for their support along the rewarding and enlightening journey of my graduate studies at the University of Minnesota (UMN). Without their help, this thesis would not have been possible.

First and foremost, I would like to express my deepest gratitude to my advisor Professor Georgios B. Giannakis for his continuous support and guidance. Apart from being a role model in the formulation, feedback on, and presentation of original research, he was always open to exploring different areas including network science and deep learning, which I felt attracted to over the past years. By generously offering invaluable guidance and constant encouragement, he has dedicated extensive amounts of time to helping me become a better researcher. For example, thanks to his excellent advice on applying state-of-the-art learning techniques to power system monitoring, I was able to accomplish works on scalable learning for power system state estimation and topology identification, which constitute an important part of this thesis. His vision and enthusiasm have always been a true inspiration for me. He has also provided me with an extraordinary research environment, where I have been lucky enough to discuss and work with many wonderful students and colleagues. Overall, I feel extremely fortunate to have had him as my advisor.

Due thanks go to Professors Yousef Saad, Mingyi Hong and Mehmet Akçakaya for agreeing to serve on my committee, as well as their valuable comments on my research.

I would also wish to give huge credit to all my collaborators during my PhD studies. Particularly, I am greatly indebted to Professor Vassilis Kekatos (now at VaTech), who was patient enough to train me in my first year at UMN. Special thanks to Professors Mehmet Akçakaya and Daniel Romero, as well as Drs. Gang Wang and Pengcheng You, for their significant contributions to our fruitful collaborations. I would like to extend my warmest thanks to current and former members of the SPiNCOM group at UMN: Dr. Brian Baingana, Seth

Barrash, Dimitris Berberidis, Dr. Jia Chen, Tianyi Chen, Professor Emiliano Dall'Anese, Vassilis Ioannidis, Georgios V. Karanikolas, Bingcong Li, Donghoon Lee, Dr. Luis M. López-Ramos, Professor Qing Ling, Dr. Qin Lu, Professor Antonio G. Marqués, Meng Ma, Dr. Athanasios Nikolakopoulos, Dr. Pooya Rezaei, Alireza Sadeghi, Fatemeh Sheikholeslami, Yanning Shen, Professor Konstantinos Slavakis, Panagiotis Traganitis, Dr. Yunlong Wang, Zejing Wang, Professor Yu Zhang, and Professor Hao Zhu, for their company and help. Due credit also goes to the grants that financially support our research.

My sincere gratitude is also due to my family and friends for their spiritual support. In particular, I would like to thank Donghan Feng, Haoji Hu, Lei Li, Lin Liu, Wei Hu, Shaolun Xu, Bo Yang, and Zheng Yan. Special thanks to my parents, who encouraged me to pursue my dreams without too many worries and burden of any kind. Your love and support have been and will always be invaluable to me.

Liang Zhang, Minneapolis, December 5, 2018.

Dedication

To my family, for their unconditional love and support.

Abstract

Contemporary power grids are being challenged by unprecedented levels of voltage fluctuations, due to large-scale deployment of electric vehicles (EVs), demand-response programs, and renewable generation. Nonetheless, with proper coordination, EVs and responsive demands can be controlled to enhance grid efficiency and reliability by leveraging advances in power electronics, metering, and communication modules. In this context, the present thesis pioneers algorithmic innovations targeting timely opportunities emerging with future power systems in terms of learning, load control, and microgrid management. Our vision is twofold: advancing algorithms and their performance analysis, while contributing foundational developments to guarantee situational awareness, efficiency, and scalability of forthcoming smart power grids.

The first thrust to this end deals with real-time power grid monitoring that comprises power system state estimation (PSSE), state forecasting, and topology identification modules. Due to the intrinsic nonconvexity of the PSSE task, optimal PSSE approaches have been either sensitive to initialization or computationally expensive. To bypass these hurdles, this thesis advocates deep neural networks (DNNs) for real-time PSSE. By unrolling an iterative physics-based prox-linear PSSE solver, a novel model-specific DNN with affordable training and minimal tuning effort is developed. To further enable system awareness even ahead of the time horizon, as well as to endow the DNN-based estimator with resilience, deep recurrent neural networks (RNNs) are also pursued for state forecasting. Deep RNNs leverage the long-term nonlinear dependencies present in the historical voltage time series to enable forecasting, and they are easy to implement. Finally, multi-kernel learning based partial correlations accounting for nonlinear dependencies between given nodal measurements are leveraged to unveil connectivity of power grids.

The second thrust leverages the obtained state and topology information to design optimal load control and microgrid management schemes. With regards to EV load control, a decentralized protocol relying on the Frank-Wolfe algorithm is put forth to manage the heterogeneous charging loads. The novel paradigm has minimal computational requirements, and is resilient to lost updates. When higher levels of EV load exceed prescribed voltage limits, the underlying grid needs to be taken into account. In this context, communication-free local reactive power control and optimal decentralized energy management schemes, are developed based on the proximal gradient method and the alternating direction method of multipliers, respectively.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recognized as the most significant engineering achievement of the twentieth century [120], the North American power grid is a complex cyber-physical system with transmission and distribution infrastructure delivering electricity from generators to consumers. Contemporary power grids are facing major challenges related to unprecedented levels of load peaks and voltage fluctuations, with the growing deployment of EVs and distributed renewable generators. The market share of EVs kept increasing over the past years, and achieved a record share of 1.13% in 2017. In total, over one million highway legal EVs were sold in the United States from February 2008 to September 2018 [116]. Growing at annual rates of 10–60% from 2004, renewable energy consumption already occupied 19.3% of the total energy consumption of the world in 2017 [117].

Coping with the challenges brought by the massive integration of EVs and renewable generators, the US Department of Energy envisions modernizing the grid into an intelligent energy system through the deployment of advanced power electronics, as well as communication and metering infrastructure [103]. The overwhelming consensus is that in order to truly transform power networks into reliable and sustainable infrastructures, innovative paradigms in the aforementioned technologies must be coupled with advances in power grid learning, monitoring, and management [44]. As a result, developing scalable and robust algorithms for grid state and topology inference, as well as for allocating resources is of paramount importance [44, 103].

With reference to Fig. 1.1, the present thesis is centered on developing scalable computational methods to markedly enhance monitoring (Chapters 2 and 3), as well as energy management (Chapters 4 and 5) for future smart power grids. The motivation, context, and the contributions

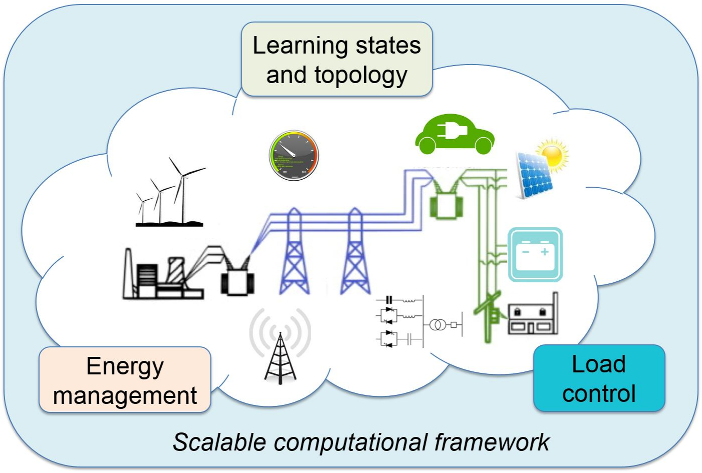

Learning states and topology
Energy management
Load control
Scalable computational framework

Figure 1.1: Cyber-enabled smart grids.

of this thesis are delineated in the ensuing sections.

1.1 Machine Learning for Power Grid Monitoring

Power grid monitoring offers situational awareness by providing estimation of system states and line connections [44]. In this section, typical grid monitoring tasks encompassing state estimation and topology identification are highlighted, along with their challenges. Subsequently, our real-time, accurate, and robust grid monitoring schemes building on the state-of-the-art machine learning techniques, are outlined.

1.1.1 Deep Learning for State Estimation and Forecasting

The supervisory control and data acquisition (SCADA) system periodically collects various types of measurements from buses and lines across the grid; see Fig. 1.2. Given SCADA measurements along with the associated grid parameters, PSSE aims to retrieve the unknown system state, that is, the complex voltages at all buses [64]. Commonly used state estimators include the weighted least-squares (WLS) and least-absolute-value (LAV) ones, which are derived based on the (weighted) ℓ_1- or ℓ_2-loss criteria. To tackle the resultant nonconvex optimization, various solvers have been proposed; see e.g. [118, 64, 27, 109, 107]. However, those optimization-oriented PSSE schemes either require many iterations or are computationally intensive, and they

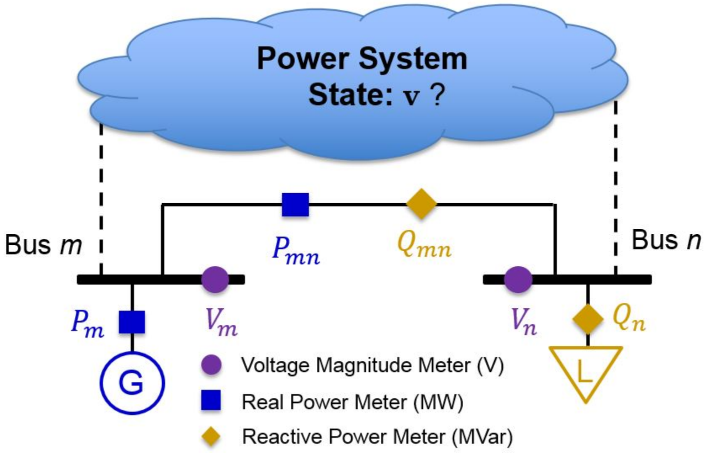


Figure 1.2: SCADA measurement system.

are further challenged by growing dynamics and system size. These considerations motivate innovative approaches for real-time large-scale PSSE.

Toward that end, PSSE using plain feed-forward neural networks (FNNs) has been studied in [7, 123]. Once trained off-line using historical data and/or simulated samples, FNNs can be implemented for real-time PSSE, as the inference entails only a few matrix-vector multiplications. Related approaches using FNNs that 'learn-to-optimize' emerge in wireless communications [100], [92], and outage detection [137]. Unfortunately, past 'plain-vanilla' FNN-based schemes not only suffer from 'vanishing' or 'exploding' gradients for deep nets, but are also model-agnostic, in the sense that they ignore the underlying physics of power grids, therefore lacking physical interpretation and yielding suboptimal performance. To devise NNs in a disciplined manner, recent proposals in computer vision [47, 121] construct deep (D) NNs by unfolding iterative solvers tailored to model-based optimization problems.

In this thesis, we pursue a hybrid approach that combines data with basic physical constraints, to develop model-specific DNNs for PSSE. On the other hand, PSSE by itself is insufficient for fully monitoring the system since it does not account for variations in the states (namely, system dynamics) [64]. In addition, PSSE works (well) only if there are enough measurements achieving system observability, and the grid topology along with the link parameters are precisely known. To address these challenges, power system state forecasting to aid PSSE [30, 111, 136, 90] is well motivated.

Power system state forecasting has so far been pursued via (extended) Kalman filtering and moving horizon approaches in e.g., [26, 73, 48], and also through first-order vector auto-regressive (VAR) modeling [49]. Nonetheless, all the aforementioned state predictors, assume *linear* dynamics; yet in practice, the dependence of the current state on previous (estimated) one(s) is nonlinear and cannot be accurately characterized. To render *nonlinear* estimators tractable, FNN-based state prediction has been advocated with the transition mapping modeled by a single-hidden-layer FNN [30, 31]. Unfortunately, the number of FNN parameters grows linearly with the length of the input sequences, discouraging FNNs from capturing long-term dependencies in voltage time series.

Chapter 2 of the present thesis contributes towards real-time and accurate monitoring of the smart power grids in two directions. First, we advocate model-specific DNNs for PSSE, by unrolling a recently proposed prox-linear SE solver [106]. Toward this goal, we first develop a reduced-complexity prox-linear solver. Subsequently, we demonstrate how the unrolled prox-linear solver lends itself to building blocks (layers) in contemporary DNNs. In contrast to 'plain-vanilla' FNNs, our novel prox-linear nets require minimal tuning efforts, and come naturally with 'skip-connections,' a critical element in modern successful DNN architectures (see [51]) that enables efficient training. Moreover, to enhance system observability as well as enable system awareness ahead of time, we introduce power system state forecasting via deep recurrent NNs (RNNs). Deep RNNs enjoy a fixed number of parameters even with variable-length input sequences, and they are able to capture complex nonlinear dependencies present in time series. Finally, we present numerical tests using real load data on the IEEE 57- and 118-bus benchmark systems to corroborate the merits of the developed deep prox-linear nets and RNNs relative to existing alternatives. The material of Chapter 2 draws from [131, 132, 133].

1.1.2 Multi-kernel Learning for Topology Identification

The aforementioned state estimation along with a number of additional tasks involved in the power network operation require full awareness of the grid topology [44, 125]. Nonetheless, power grids are regularly reconfigured due to faults on power lines, or due to changes induced to minimize losses. These reasons lead to situations where the current grid topology is unknown [44, 4, 64]. In this context, we consider learning grid topologies based on nodal observations.

Recent advances in machine learning and signal processing have introduced new tools for grid topology identification [45]. Leveraging the inherent sparsity of grid connectivity, locational marginal prices are used to identify transmission grid topology [63]. Adopting a linearized power flow model, the grid topology is recovered by factorizing a data matrix of spatio-temporal power injections [75]. The microgrid structure is identified leveraging time delays of communication signals in [32]. Relying on the second-order moments of nodal observations, a systematic learning approach for radial grids is advocated in [28]. The task of grid topology tracking in the presence of switching actions and topology transitions has been also addressed with synchrophsors [19]. These approaches assume information such as communication delays, line quantities, or synchrophasor readings, which are not available in most distribution grids. The linear partial correlation (PC) method as well as its concentration matrix-based variants have also been advocated for power grid topology identification [12, 29]. Unfortunately, the recovery performance is suboptimal especially for meshed (loopy) networks because the nodal voltages collected are typically correlated, and voltage angles/magnitudes are nonlinearly dependent in practice. Although the nonlinear relationships between voltages are well acknowledged, grid topology learning that accounts for nonlinear dependencies has not been investigated yet.

To better capture nonlinear dependencies, we advocate multi-kernel based PCs (MKPCs) for grid (loopy or radial) topology identification in Chapter 3. Instead of adopting the ℓ_2-regularization for MKPC [60], which is computationally easy to deal with, this contribution first formulates the more challenging task of ℓ_p-norm based MKPC. Our generalized nonlinear model is expected to offer improved estimators by exploiting the flexibility in determining the combination of kernels. For example, when $p = 1$, only a few kernels from the user-specified dictionary of kernels can be combined to form the nonlinear estimator. This combination boosts the overall performance of MKPC-based topology learning. Furthermore, an efficient implementaion based on the recently revived Frank-Wolfe algorithm [58] is developed for the ℓ_p-norm regularized MKPC. Besides guaranteed convergence to a global solution, it also features simple closed-form updates. At last, power grid topology identification using ℓ_p-norm regularized MKPCs is highlighted. The results of Chapter 3 have been reported in [130, 129].

1.2 Decentralized Energy Management Schemes

Leveraging our learning-based monitoring modalities established in Chapters 2 and 3, the next-generation power grid will be managed with markedly improved security and efficiency. Turning attention to resource allocation, Chapters 4 and 5 develop scalable energy management schemes for smart microgrids.

1.2.1 Electric Vehicle Charging Coordination

Electric vehicles have received considerable attention from the automotive industry and the government due to their capacity to reduce greenhouse gas emissions, and mitigate oil dependency. However, the overall load profile will be greatly affected by the increasing numbers of EVs. Uncoordinated charging of even 10% penetration of EV loads will notably affect power system operation, giving rise to considerable voltage magnitude fluctuations and unacceptable load peaks [86]. On the other hand, with proper coordination, the sheer volume of EV loads can be controlled to minimize charging costs or perform valley-filling tasks relying on advanced power electronics. To cater scalability to EV fleets and enhance user privacy, decentralized charging control strategies are highly desirable.

Decentralized charging protocols based on congestion pricing schemes similar to those used by Internet Protocol networks can be found in [33]. However, their optimality is not guaranteed. An ad hoc multi-agent based EV charging scheme is devised in [61]. Presuming identical energy requests and plug-in/-out times for all vehicles, a game-theoretic charging scheme attaining Nash equilibrium has been also developed [78]. Iterative schemes based on Lagrangian relaxation are suggested in [42], [95], while [87] builds on the alternating direction method of multipliers (ADMM). A feasible valley-filling charging profile has been proved optimal for any convex cost in [41], where a decentralized protocol based on projected gradient descent is also reported. The spatial coupling of EV charging decisions due to transformer capacity limits is tackled in [77] via a combination of the ADMM and projected gradient descent solvers. Unfortunately, existing optimal charging methods either suffer from slow convergence [42, 95], or they are computationally demanding for EV controllers [41, 87, 77], which hampers their deployment.

The main contribution of Chapter 4 in EV charging coordination is two-fold. First, a decentralized charging protocol is developed based on the Frank-Wolfe method. The devised scheme enjoys fast convergence, especially during the first few iterations. Its closed-form updates pose minimal computational requirements for the vehicle controllers, and the overall computational time is significantly reduced. In particular, our numerical tests demonstrate a 100-times speed-up advantage over existing alternatives. Second, to cope with cyber failures in the communication link between the aggregator and the vehicle controllers, an asynchronous charging scheme is devised based on the randomized-block Frank-Wolfe algorithm. Using our judiciously designed step-size sequences, the asynchronous scheme is shown to converge in the mean at sublinear rates. The results of Chapter 4 are included in [127, 135, 128].

1.2.2 Energy Management under Network Constraints

The EV management schemes of Chapter 4 apply to scenarios where power demands can be transparently supported by the underlying microgrids. Such a scenario arises when the penetration of EVs and renewable sources is relatively low, and is not expected to incur voltage violations. However, as EVs and renewable generators scale up, network-related constraints such as voltage regulation must be enforced by the utility to ensure secure operation of the microgrids. Utility-owned equipment conventionally employed for voltage regulation, such as tap-changing transformers and shunt capacitors, cannot react promptly and efficiently enough to deal with frequent voltage violations caused by renewable generation and EVs [3, 18]. Hence, avoiding under-/over-voltage conditions motivates well advanced reactive power management solutions. To that end, the power electronics of photovoltaic (PV) inverters, EV chargers, and storage devices offer a fast-responding alternative [89, 102]. Catering to the potential lack of communication infrastructure, and the scalability of distributed reactive power resources, local *plug-and-play* reactive power control schemes for voltage regulation, i.e., maintaining bus voltage magnitudes within proper ranges, are highly desirable.

Given that voltage magnitudes depend on grid-wide reactive injections, guaranteeing voltage regulation constraints may be hard to accomplish via purely localized algorithms. In such setups, reactive power management is usually *relaxed* to penalizing voltage magnitude deviations from a desired value and neglecting power losses. Reactive power injections are adjusted proportionally to the local voltage violations in [88]; see also [124] for sufficient conditions guaranteeing its convergence. A similar local control strategy has been shown to minimize a modified voltage regulation cost in [36], while [35] proposes a subgradient-based algorithm. A control rule adjusting the inverter voltage output according to the reactive power flow is reported in [43]. The local control rules proposed in [74] maintain voltage magnitudes within the desired range under the presumption of unlimited reactive power support. Most existing works build on a simplified single-phase grid model. Due to untransposed distribution lines and unbalanced loads though, the equivalent single-phase distribution network may not exist.

In the first part of Chapter 5, we develop local reactive power control schemes for both single- and multi-phase microgrids. Our contribution is on three fronts. First, we provide a unified matrix-vector notation for approximate yet quite accurate multi-phase grid models. Second, localized reactive power control schemes with superior convergence properties are devised based on proximal gradient methods. Third, using a linear approximation for unbalanced multi-phase grids, we reveal an interesting inter-phase coupling pattern across buses. Recall that in single-phase grids, increasing the reactive power injection at any bus raises the nodal voltage magnitudes throughout the grid. In multi-phase grids however, injecting more reactive power into a bus of one phase could result in decreasing voltage magnitudes for the preceding in the positive-sequence ordering phase. It is subsequently shown that in unbalanced scenarios, our reactive power control rule converges to an equilibrium point.

The aforementioned local reactive power control schemes are well suited for microgrids with limited communication capabilities. However, due to lack of information exchange, these local schemes suffer from degraded performance, especially when reactive power resources are limited [66, 138]. Supposing that the communication infrastructure is available in microgrids, various optimal network-constrained energy management schemes have been proposed. Managing microgrids in a centralized fashion is not only computationally challenging, but can also infringe user privacy. Prompted by these considerations, the focus here is on the design of optimal decentralized microgrid energy management schemes that comply with the voltage constraints of the microgrids, as well as the physical limitations of adjustable loads.

Without considering load control, decentralized microgrid management schemes based on ADMM [24, 85], and dual decomposition [22] have been developed; see [79] for a survey. Broadening the aforementioned approaches to encompass control of temporally coupled loads, e.g., EVs, is non-trivial due to the elaborate time-coupling constraints they introduce. Joint generation and EV dispatch under distribution grid limitations has also been considered. Centralized multi-period scheduling schemes are studied under different linear models for multi-phase networks in [25] and [37]. The objective function is confined to be linear, and the optimal solution is found using generic commercial solvers without exploiting the problem structure. A method for heuristically checking network constraint violations after vehicles and distributed generation have been scheduled is reported in [9]. Presuming at most one EV per bus, management under balanced network constraints has been tackled using a water-filling algorithm [20]. Nevertheless, the aforementioned protocols can not be implemented in a decentralized fashion, which hampers their applicability to sizable power networks or large-scale EV fleets. Based on dual decomposition, a distributed energy management strategy for microgrids has been proposed in [97]. However, the problem formulation therein does not account for the total charging requirements of EVs. In addition, the devised scheme requires all local controllers to communicate with the central scheduler per iteration.

The second part of Chapter 5 deals with optimal microgrid energy management accounting for both distributed generation (DG) units and adjustable EV loads. Our contribution is two-fold. First, building on our approximate grid model, multi-period microgrid energy management that incorporates critical constraints related to voltage limits and feeder transformer capacity is formulated as a convex quadratic program. Second, a decentralized algorithm is developed based on ADMM that critically decouples coordination across buses. To cater scalability to EV fleets, the resulting subproblem of charging vehicles connected to the same bus and phase is tackled using the Frank-Wolfe-based charging schemes devised in Chapter 4.

The material of Chapter 5 has been reported in [128, 66].

1.3 Notational Conventions

In the subsequent chapters, the following notation will be used throughout. Lower- (upper-) case boldface letters denote column vectors (matrices), with the exception of line power flow vectors $(\mathbf{P}, \mathbf{Q})$ that are traditionally represented with upper case boldface fonts. Calligraphic symbols are reserved for sets. Symbol $^\top$ stands for transposition. Vectors $\mathbf{0}$ and $\mathbf{1}$ are the all-zeros and all-ones canonical vectors, respectively. Symbol $\|\mathbf{x}\|_2$ denotes the ℓ_2-norm of vector $\mathbf{x}$, while $\lambda_i(\mathbf{X})$ stands for the i-th largest eigenvalue of matrix $\mathbf{X}$. Operator $\operatorname{diag}(\mathbf{x})$ defines a diagonal matrix having $\mathbf{x}$ on its diagonal, whereas $\operatorname{diag}(\mathbf{X})$ is the vector formed by the main diagonal of $\mathbf{X}$. Operator $\operatorname{bdiag}(\{\mathbf{X}_n\})$ defines a block diagonal matrix with $\mathbf{X}_n$s denoting its block entries. Operators $\operatorname{Re}(z)$ and $\operatorname{Im}(z)$ return the real and imaginary part of complex number z. A matrix with non-negative entries is denoted by $\mathbf{X} \geq \mathbf{0}$, while $\mathbf{X} \succeq \mathbf{0}$ indicates a symmetric positive semidefinite matrix.

Chapter 2

Real-time Power System State Estimation and Forecasting

Section 2.1 outlines the basics of PSSE. Section 2.2 introduces our novel reduced-complexity prox-linear solver for PSSE, and advocates the prox-linear net. Section 2.3 deals with deep RNN for state forecasting, as well as shows how state forecasting can aid in turn DNN-based PSSE. Simulated tests are presented in Section 2.4.

2.1 Least-absolute-value Estimation

Consider a power network consisting of N buses that can be modeled as a graph $\mathcal{G} := \{\mathcal{N}, \mathcal{L}\}$, where $\mathcal{N} := \{1, 2, \ldots, N\}$ comprises all buses, and $\mathcal{L} := \{(n, n')\} \in \mathcal{N} \times \mathcal{N}$ collects all lines. For each bus $n \in \mathcal{N}$, let $V_n := v_n^r + jv_n^i$ denote its corresponding complex voltage, and P_n (Q_n) denote the active (reactive) power injection. For each line $(n, n') \in \mathcal{L}$, let $P_{nn'}^f$ $(Q_{nn'}^f)$ denote the active (reactive) power flow seen at the 'forwarding' end, and $P_{nn'}^t$ $(Q_{nn'}^t)$ denote the active (reactive) power flow at the 'terminal' end. To perform power system state estimation or forecasting, suppose M_t system variables are measured at time t. For a compact representation, let $\mathbf{z}_t := [\{|V_{n,t}|^2\}_{n \in \mathcal{N}_t^o}, \{P_{n,t}\}_{n \in \mathcal{N}_t^o}, \{Q_{n,t}\}_{n \in \mathcal{N}_t^o}, \{P_{nn',t}^f\}_{(n,n') \in \mathcal{E}_t^o}, \{Q_{nn',t}^f\}_{(n,n') \in \mathcal{E}_t^o}, \{P_{nn',t}^t\}_{(n,n') \in \mathcal{E}_t^o}, \{Q_{nn',t}^t\}_{(n,n') \in \mathcal{E}_t^o}]^\top$ be the data vector that collects all measured quantities at time t, where sets $\mathcal{N}_t^o$ and $\mathcal{E}_t^o$ signify the locations where the corresponding nodal and line quantities are measured.

Per time slot t, PSSE aims to recover the system state vector $\mathbf{v}_t := [v_1^r, v_1^i, \ldots, v_N^r, v_N^i]^\top \in$

$\mathbb{R}^{2N}$ (expanded in the rectangular coordinates) from generally noisy data $\mathbf{z}_t$. For brevity, the subscript t of $\mathbf{z}_t$ and $\mathbf{v}_t$ will be omitted when discussing PSSE in Sections 2.1 and 2.2. Concretely, the PSSE task can be posed as follows.

Given M measurements $\{z_m\}_{m=1}^M$ and the corresponding matrices $\{\mathbf{H}_m \in \mathbb{R}^{2N \times 2N}\}_{m=1}^M$ obeying the physical model

$$z_m = \mathbf{v}^\top \mathbf{H}_m \mathbf{v} + \epsilon_m, \qquad \forall m = 1, \ldots, M \tag{2.1}$$

our goal is to recover $\mathbf{v} \in \mathbb{R}^{2N}$, where $\{\epsilon_m\}$ account for the measurement noise and modeling inaccuracies.

Adopting the LAV error criterion that is known to be robust to outliers, the ensuing LAV estimate is sought (see e.g., [118])

$$\hat{\mathbf{v}} := \arg \min_{\mathbf{v} \in \mathbb{R}^{2N}} \frac{1}{M} \sum_{m=1}^{M} \left| z_m - \mathbf{v}^\top \mathbf{H}_m \mathbf{v} \right| \tag{2.2}$$

for which various solvers have been developed [2, 57, 64]. In particular, the recent prox-linear solver developed in [106] has well documented merits, including provably fast (locally quadratic) convergence, as well as efficiency in coping with the non-smoothness and nonconvexity in (2.2). Specifically, the prox-linear solver starts with some initial vector $\mathbf{v}_0$, and iteratively minimizes the regularized and 'locally linearized' (relaxed) cost in (2.2), to obtain iteratively (see also [16])

$$\mathbf{v}_{i+1} = \arg \min_{\mathbf{v} \in \mathbb{R}^{2N}} \left\| \mathbf{z} - \mathbf{J}_i(2\mathbf{v} - \mathbf{v}_i) \right\|_1 + \frac{M}{2\mu_i} \left\| \mathbf{v} - \mathbf{v}_i \right\|_2^2 \tag{2.3}$$

where $i \in \mathbb{N}$ is the iteration index, $\mathbf{J}_i := [\mathbf{v}_i^\top \mathbf{H}_m]_{1 \leq m \leq M}$ is an $M \times N$ matrix whose m-th row is $\mathbf{v}_i^\top \mathbf{H}_m$, and $\{\mu_i > 0\}$ is a pre-selected step-size sequence.

It is clear that the per-iteration subproblem (2.3) is a convex quadratic program, which can be solved by means of standard convex programming methods. One possible iterative solver of (2.3) is based on the alternating direction method of multipliers (ADMM). Such an ADMM-based inner loop turns out to entail $2M + 2N$ auxiliary variables, thus requiring the update of $2M + 4N$ variables per iteration [106].

Aiming at a reduced-complexity solver, in the next section we will first recast (2.3) in a Lasso-type form, and subsequently unroll the resultant double-loop prox-linear iterations (2.3)

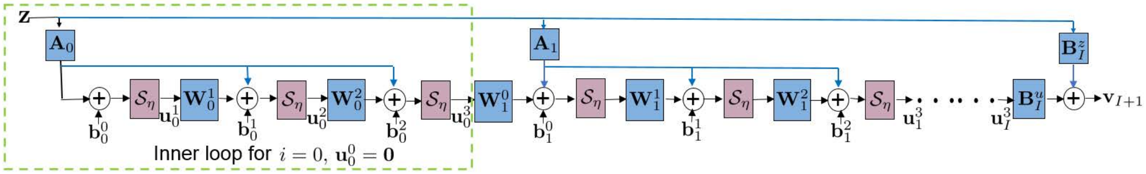


Figure 2.1: Prox-linear net with $K = 3$ blocks.

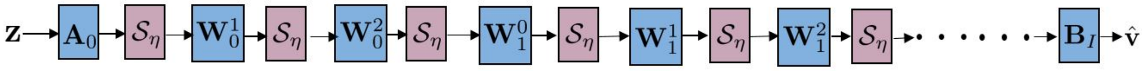


Figure 2.2: Plain-vanilla FNN which has the same per-layer number of hidden units as the prox-linear net.

that constitute the key blocks of our DNN-based PSSE solver.

2.2 The Prox-linear Net

In this section, we will develop a DNN-based scheme to approximate the solution of (2.2) by unrolling the double-loop prox-linear iterations. Upon defining the vector variable $\mathbf{u}_i := \mathbf{J}_i(2\mathbf{v} - \mathbf{v}_i) - \mathbf{z}$, and plugging $\mathbf{u}_i$ into (2.3), we arrive at

$$\mathbf{u}_i^* = \arg\min_{\mathbf{u}_i \in \mathbb{R}^M} \|\mathbf{u}_i\|_1 + \frac{M}{4\mu_i}\|\mathbf{B}_i(\mathbf{u}_i + \mathbf{z}) - \mathbf{v}_i\|_2^2 \tag{2.4}$$

where $\mathbf{B}_i \in \mathbb{R}^{2N \times M}$ denotes the pseudo inverse of $\mathbf{J}_i$ that satisfies $\mathbf{B}_i\mathbf{J}_i = \mathbf{I}_{2N \times 2N}$. The pseudo inverse $\mathbf{B}_i$ exists because PSSE requires $M \geq 2N$ to guarantee system observability.

Once the inner optimum variable $\mathbf{u}_i^*$ is found, the next outer-loop iterate $\mathbf{v}_{i+1}$ can be readily obtained as

$$\mathbf{v}_{i+1} = [\mathbf{B}_i(\mathbf{u}_i^* + \mathbf{z}) + \mathbf{v}_i]/2 \tag{2.5}$$

following the definition of $\mathbf{u}_i$. Interestingly, (2.4) now reduces to a Lasso problem [83], for which various celebrated solvers have been put forward, including e.g., the iterative shrinkage and thresholding algorithm (ISTA) [83].

Specifically, with k denoting the iteration index of the inner-loop, the ISTA for (2.4) proceeds across iterations $k = 0, 1, \ldots$ along the following recursion

$$\begin{aligned} \mathbf{u}_i^{k+1} &= \mathcal{S}_\eta\Big(\mathbf{u}_i^k - \frac{\eta M}{2\mu_i}\mathbf{B}_i^\top \Big[\mathbf{B}_i\Big(\mathbf{u}_i^k + \mathbf{z}\Big) - \mathbf{v}_i\Big]\Big) \\ &= \mathcal{S}_\eta\Big(\mathbf{W}_i^k \mathbf{u}_i^k + \mathbf{A}_i \mathbf{z} + \mathbf{b}_i^k\Big) \end{aligned} \tag{2.6}$$

where $\eta > 0$ is a fixed step size with coefficients

$$\mathbf{W}_i^k := \mathbf{I} - \frac{\eta M}{2\mu_i}\mathbf{B}_i^\top \mathbf{B}_i, \quad \forall k \in \mathbb{N} \tag{2.7a}$$

$$\mathbf{A}_i := -\frac{\eta M}{2\mu_i}\mathbf{B}_i^\top \mathbf{B}_i \tag{2.7b}$$

$$\mathbf{b}_i^k := \frac{\eta M}{2\mu_i}\mathbf{B}_i^\top \mathbf{v}_i, \quad \forall k \in \mathbb{N} \tag{2.7c}$$

and $\mathcal{S}_\eta(\cdot)$ is the so-termed soft thresholding operator

$$\mathcal{S}_\eta(x) := \begin{cases} x - \eta, & x > \eta \\ 0, & -\eta \leq x \leq \eta \\ x + \eta, & x < -\eta \end{cases} \tag{2.8}$$

understood entry-wise when applied to a vector. With regards to initialization, one can set $\mathbf{u}_0^0 = \mathbf{0}$ without loss of generality, and $\mathbf{u}_i^0 = \mathbf{u}_{i-1}^*$ for $i \geq 1$.

Algorithm 1 Reduced-complexity prox-linear solver.

```
Input: Data $\{(z_m, \mathbf{H}_m)\}_{m=1}^M$, step sizes $\{\mu_i\}$, $\eta$, and initialization $\mathbf{v}_0 = \mathbf{1}$, $\mathbf{u}_0^0 = \mathbf{0}$
for $i = 0, 1, \ldots, I$ do
    Evaluate $\mathbf{W}_i^k$, $\mathbf{A}_i$, and $\mathbf{b}_i^k$ according to (2.7)
    Initialize $\mathbf{u}_i^0$
    for $k = 0, 1, \ldots, K$ do
        Update $\mathbf{u}_i^{k+1}$ using (2.6)
    end for
    Update $\mathbf{v}_{i+1}$ using (2.5)
end for
```

The new prox-linear PSSE solver with reduced-complexity is summarized in Algorithm 1. With appropriate step sizes $\{\mu_i\}$ and η, the sequence $\{\mathbf{v}_i\}$ generated by Algorithm 1 converges to a stationary point of (2.2) [106]. In practice, Algorithm 1 often requires a large number K of inner iterations to approximate the solution of (2.4). Furthermore, the pseudo-inverse $\mathbf{B}_i$ has to be computed per outer-loop iteration. These challenges can limit its use in large-scale as well as in real-time applications.

Instead of solving the optimization problem with a (large) number of iterations, recent proposals [47, 121] advocated trainable DNNs constructed by unfolding those iterations, to obtain data-driven solutions. As demonstrated in [47, 114, 121], properly trained unrolled DNNs can achieve competitive performance even with a small number of layers. In the following, we will elaborate on how to unroll our Algorithm 1 to construct DNNs for high-performance PSSE.

Consider first unrolling the outer loop (2.3) up to, say the $(I+1)$-st iteration to obtain $\mathbf{v}_{I+1}$. Leveraging the recursion (2.6), each inner loop iteration i refines the initialization $\mathbf{u}_i^0 = \mathbf{u}_{i-1}^K$ to yield after K inner iterations $\mathbf{u}_i^K$. Such an unrolling leads to a $K(I+1)$-layer structured DNN. Suppose that the sequence $\{\mathbf{v}_i\}_{i=0}^{I+1}$ has converged, which means $\|\mathbf{v}_I - \mathbf{v}_{I+1}\| \leq \epsilon$ for some $\epsilon > 0$. It can then be deduced that $\mathbf{v}_{I+1} = \mathbf{B}_I^u \mathbf{u}_I^* + \mathbf{B}_I^z \mathbf{z}$ with $\mathbf{B}_I^u := \mathbf{B}_I$ and $\mathbf{B}_I^z := \mathbf{B}_I$.

Our novel DNN architecture, that builds on the physics-based Algorithm 1, is thus a hybrid combining plain-vanilla FNNs with the conventional iterative solver such as Algorithm 1. We will henceforth term it 'prox-linear net.' For illustration, the prox-linear net with $K = 3$ is depicted in Fig. 2.1. The first inner loop $i = 0$ is highlighted in a dashed box, where $\mathbf{u}_0^1 = \mathcal{S}_\eta(\mathbf{A}_0\mathbf{z} + \mathbf{b}_0)$ because $\mathbf{u}_0^0 = \mathbf{0}$. As with [47, 114, 121], our prox-linear net can be treated as a trainable regressor that predicts $\mathbf{v}$ from data $\mathbf{z}$, where the coefficients $\{\mathbf{b}_i^k\}_{0\leq i\leq I}^{1\leq k\leq 3}$, $\{\mathbf{A}_i\}_{i=0}^{I}$, $\{\mathbf{W}_i^k\}_{0\leq i\leq I}^{1\leq k\leq 3}$, $\mathbf{B}_I^u$, and $\mathbf{B}_I^z$ are typically untied to enhance approximation capability and learning flexibility. Given historical and/or simulated measurement-voltage training pairs $\{(\mathbf{z}_s, \mathbf{v}_s)\}$, these coefficients can be learned end-to-end using backpropagation [91], possibly employing the Huber loss [54] to endow the state estimates with resilience to outliers.

Relative to the conventional FNN in Fig. 2.2, our proposed prox-linear net features: i) 'skip-connections' (the bluish lines in Fig. 2.1) that directly connect the input $\mathbf{z}$ to intermediate/output layers, and ii) a fixed number (M in this case) of hidden neurons per layer. It has been shown both analytically and empirically that these skip-connections help avoid the so-termed 'vanishing' and 'exploding' gradient issues, therefore enabling successful and efficient training of DNNs [51, 52]. The 'skip-connections' is also a key enabler of the universal approximation capability of DNNs with a fixed number of hidden-neurons per-layer [76].

The only hyper-parameters that must be tuned in our prox-linear net are I, K, and η, which are also tuning parameters required by the iterative optimization solver in Algorithm 1. It is also

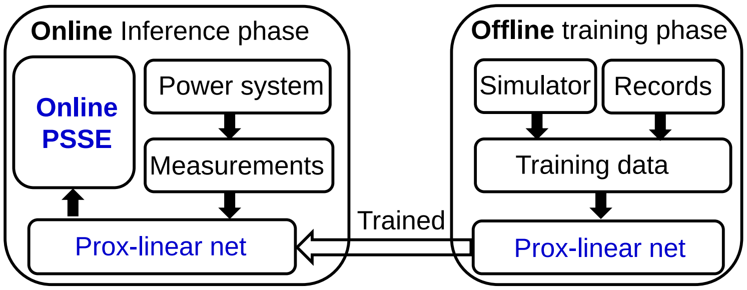


Figure 2.3: Deep prox-linear net based real-time PSSE.

worth pointing out that other than the soft-thresholding nonlinearity (a.k.a. activation function) used in Figs. 2.1 and 2.2, alternative functions such as the rectified linear unit (ReLU) can be applied as well [47]. We have observed in our simulated tests that the prox-linear net with soft thresholding operators or ReLUs yield similar performance. To understand how different network architectures affect the performance, ReLU activation functions are used by default unless otherwise stated.

The flow chart demonstrating the prox-linear net for real-time PSSE is depicted in Fig. 2.3, where the real-time inference stage is described in the left rounded rectangular box, while the off-line training stage is on the left. Thanks to the wedding of the physics in (2.1) with our DNN architecture design, the extensive numerical tests in Section 2.4 will confirm an impressive boost in performance of our prox-linear nets relative to competing FNN and Gauss-Newton based PSSE approaches.

2.3 Deep RNNs for State Forecasting

Per time slot t, the PSSE scheme we developed in Section 2.2 estimates the state vector $\mathbf{v}_t \in \mathbb{R}^{2N}$ upon receiving measurements $\mathbf{z}_t$. Nevertheless, its performance is challenged when there are missing entries in $\mathbf{z}_t$, which is indeed common in a SCADA system due for example to meter and/or communication failures [64]. To enhance our novel PSSE scheme and obtain system awareness even ahead of time, we are prompted to pursue power system state forecasting, which for a single step amounts to predicting the next state $\mathbf{v}_{t+1}$ at time slot $t+1$ from the available time-series $\{\mathbf{v}_\tau\}_{\tau=0}^{t}$ [30]. Analytically, the estimation and prediction steps are as follows

$$\mathbf{v}_{t+1} = \boldsymbol{\phi}(\mathbf{v}_t, \mathbf{v}_{t-1}, \mathbf{v}_{t-2}, \ldots, \mathbf{v}_{t-r+1}) + \boldsymbol{\xi}_t \tag{2.9}$$

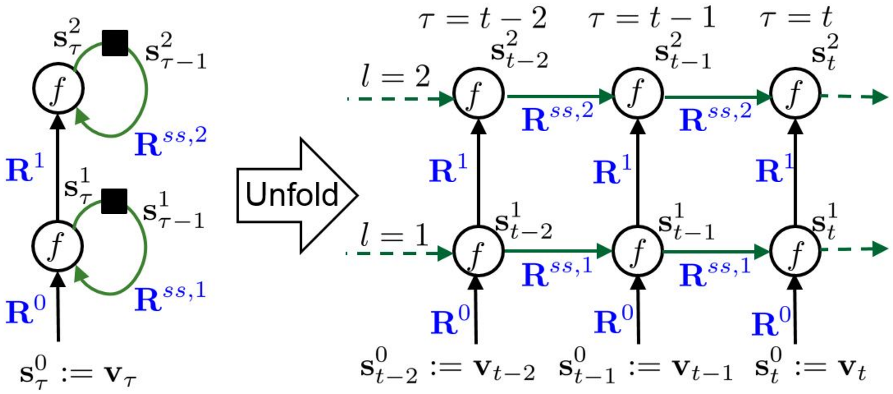


Figure 2.4: An unfolded deep RNN with no outputs.

$$\mathbf{z}_{t+1} = \mathbf{h}_{t+1}(\mathbf{v}_{t+1}) + \boldsymbol{\epsilon}_{t+1} \tag{2.10}$$

where $\{\boldsymbol{\xi}_t, \boldsymbol{\epsilon}_{t+1}\}$ account for modeling inaccuracies; the tunable parameter $r \geq 1$ represents the number of lagged (present included) states used to predict $\mathbf{v}_{t+1}$; and the unknown (nonlinear) function $\boldsymbol{\phi}$ captures the state transition, while $\mathbf{h}_{t+1}(\cdot)$ is the measurement function that summarizes equations in (2.1) at time slot $t+1$. To perform state forecasting, function $\boldsymbol{\phi}$ must be estimated or approximated – a task that we will accomplish using RNN modeling, as we present next.

RNNs are NN models designed to learn from correlated time series data. Relative to FNNs, RNNs are not only scalable to long-memory inputs (regressors) that entail sequences of large r, but are also capable of processing input sequences of variable length [46]. Given the input sequence $\{\mathbf{v}_\tau\}_{\tau=t-r+1}^t$, and an initial state $\mathbf{s}_{t-r}$, an RNN finds the *hidden state*[1] vector sequence $\{\mathbf{s}_\tau\}_{\tau=t-r+1}^t$ by repeating

$$\mathbf{s}_\tau = f(\mathbf{R}^0\mathbf{v}_\tau + \mathbf{R}^{ss}\mathbf{s}_{\tau-1} + \mathbf{r}^0) \tag{2.11}$$

where $f(\cdot)$ is a nonlinear activation function (e.g., a ReLU or sigmoid unit), understood entry-wise when applied to a vector, whereas the coefficient matrices $\mathbf{R}^0$, $\mathbf{R}^{ss}$, and the vector $\mathbf{r}^0$ contain time-invariant weights.

Deep RNNs are RNNs of multiple ($\geq$ 3) processing layers, which can learn compact

[1] Hidden state is an auxiliary set of vector variables not to be confused with the power system state $\mathbf{v}$ consisting of the nodal voltages as in (2.1).

representations of time series through hierarchical nonlinear transformations. The state-of-the-art in numerous sequence processing applications, including music prediction and machine translation [46], has been significantly improved with deep RNN models. By stacking up multiple recurrent hidden layers (cf. (2.11)) one on top of another, deep RNNs can be constructed as follows [84]

$$\mathbf{s}_\tau^l = f\big(\mathbf{R}^{l-1}\mathbf{s}_\tau^{l-1} + \mathbf{R}^{ss,l}\mathbf{s}_{\tau-1}^l + \mathbf{r}^{l-1}\big), \quad l \geq 1 \tag{2.12}$$

where l is the layer index, $\mathbf{s}_\tau^l$ denotes the hidden state of the l-th layer at slot τ having $\mathbf{s}_\tau^0 := \mathbf{v}_\tau$, and $\{\mathbf{R}^l, \mathbf{R}^{ss,l}, \mathbf{r}^l\}$ collect all unknown weights. Fig. 2.4 (left) depicts the computational graph representing (2.12) for $l = 2$, with the bias vectors $\mathbf{r}^l = \mathbf{0}$, $\forall l$ for simplicity in depiction, and the black squares standing for single-step delay units. Unfolding the graph by breaking the loops and connecting the arrows to corresponding units of the next time slot, leads to a deep RNN in Fig. 2.4 (right), whose rows represent layers, and columns denote time slots.

The RNN output can come in various forms, including one output per time step, or, one output after several steps. The latter matches the rth-order nonlinear regression in (2.9) when approximating $\boldsymbol{\phi}$ with a deep RNN. Concretely, the output of our deep RNN is given by

$$\check{\mathbf{v}}_{t+1} = \mathbf{R}^{out}\mathbf{s}_t^l + \mathbf{r}^{out} \tag{2.13}$$

where $\check{\mathbf{v}}_{t+1}$ is the forecast of $\mathbf{v}_{t+1}$ at time t, and $(\mathbf{R}^{out}, \mathbf{r}^{out})$ contain weights of the output layer. Given historical voltage time series, the weights $(\mathbf{R}^{out}, \mathbf{r}^{out})$ and $\{\mathbf{R}^l, \mathbf{R}^{ss,l}, \mathbf{r}^l\}$ can be learned end-to-end using backpropagation [46]. Invoking RNNs for state-space models, the class of nonlinear predictors discussed in [30] is considerably broadened here to have memory. As will be demonstrated through extensive numerical tests, the forecasting performance can be significantly improved through the use of deep RNNs. Although the focus here is on one-step state forecasting, it is worth stressing that our proposed approaches with minor modifications, can be generalized to predict the system states multiple steps ahead.

So far, we have elaborated on how RNNs enable flexible nonlinear predictors for power system state forecasting. To predict $\check{\mathbf{v}}_{t+1}$ at time slot t, the RNN in (2.12) requires ground-truth voltages $\{\mathbf{v}_\tau\}_{\tau=t-r+1}^t$ (cf. (2.9)), which however, may not be available in practice. Instead we can use the estimated ones $\{\hat{\mathbf{v}}_\tau\}_{\tau=t-r+1}^t$ provided by our prox-linear net-based estimator in Section 2.2. In turn, the forecast $\check{\mathbf{v}}_{t+1}$ can be employed as a prior to aid PSSE at time slot $t+1$, by providing the so-termed virtual measurements $\check{\mathbf{z}}_{t+1} := \mathbf{h}_{t+1}(\check{\mathbf{v}}_{t+1})$ that can be readily

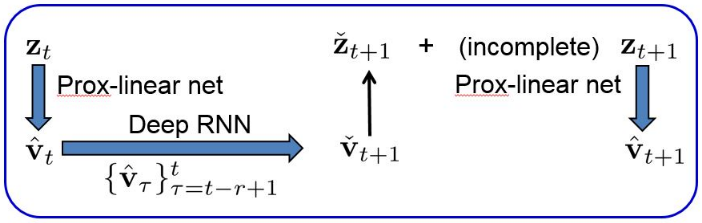

$\mathbf{z}_t$
Prox-linear net
Deep RNN
$\hat{\mathbf{v}}_t$
$\{\hat{\mathbf{v}}_\tau\}_{\tau=t-r+1}^{t}$
$\check{\mathbf{z}}_{t+1}$
+
(incomplete) $\mathbf{z}_{t+1}$
$\check{\mathbf{v}}_{t+1}$
Prox-linear net
$\hat{\mathbf{v}}_{t+1}$

Figure 2.5: DNN-based real-time power system monitoring.

accounted for in (2.2). For example, when there are missing entries in $\mathbf{z}_{t+1}$, the obtained $\check{\mathbf{z}}_{t+1}$ can be used to improve the PSSE performance by imputing the missing values.

Figure 2.5 depicts the flow chart of the overall real-time power system monitoring scheme, consisting of deep prox-linear net-based PSSE and deep RNN-based state forecasting modules, that are implemented at time t and $t+1$. Our novel scheme is reminiscent of the predictor-corrector-type estimators emerging with dynamic state estimation problems using Kalman filters [136, 64]. Although beyond the scope of the present thesis, it is worth remarking that the residuals $\mathbf{z}_{t+1} - \mathbf{h}_{t+1}(\hat{\mathbf{v}}_{t+1})$ along with $\mathbf{z}_{t+1} - \check{\mathbf{z}}_{t+1}$ can be used to unveil erroneous data, as well as changes in the grid topology and the link parameters; see [30] for an overview.

2.4 Numerical Tests

Performance of our deep prox-linear net based PSSE, and deep RNN based state forecasting methods was evaluated using the IEEE 57- and 118-bus benchmark systems. Real load data from the 2012 Global Energy Forecasting Competition (GEFC)[2] were used to generate the training and testing datasets, where the load series were subsampled for size reduction by a factor of 5 (2) for the IEEE 57-bus (118-bus) system. Subsequently, the resultant load instances were normalized to match the scale of power demands in the simulated system. The MATPOWER toolbox [139] was used to solve the AC power flow equations with the normalized load series as inputs, to obtain the ground-truth voltages $\{\mathbf{v}_\tau\}$, and produce measurements $\{\mathbf{z}_\tau\}$ that comprise all forwarding-end active (reactive) power flows, as well as all voltage magnitudes. All NNs were trained using 'TensorFlow' [1] on an NVIDIA Titan X GPU with 12GB RAM, with weights learned by the backpropagation based algorithm 'Adam' (with starting learning rate 10^{-3}) for 200 epochs. To alleviate randomness in the obtained weights introduced by the training algorithms, all NNs

[2] https://www.kaggle.com/c/global-energy-forecasting-competition-2012-load-forecasting/data.

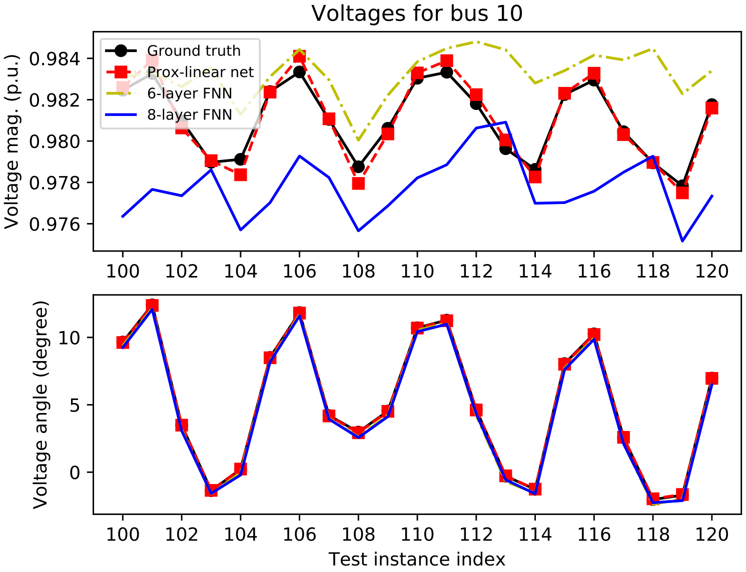


Figure 2.6: Estimation errors in voltage magnitudes and angles of bus 10 of the 57-bus system from test instances 100 to 120.

were trained and tested independently for 20 times, with reported results averaged over 20 runs. For reproducibility, the 'Python'-based implementation of our prox-linear net for PSSE of the 118-bus system is publicly available at https://github.com/LiangZhangUMN/PSSE-via-DNNs.

2.4.1 Simulations for PSSE

To start, the prox-linear net based PSSE was tested, which estimates $\{\hat{\mathbf{v}}_\tau\}$ using $\{\mathbf{z}_\tau\}$. For both training and testing phases, all measurements $\{\mathbf{z}_\tau\}$ were corrupted by additive white Gaussian noise, where the standard deviation for power flows and for voltage magnitudes was 0.02 and 0.01. The estimation performance of our prox-linear net was assessed in terms of the normalized root mean-square error (RMSE) $\|\hat{\mathbf{v}} - \mathbf{v}\|_2/N$, where $\mathbf{v}$ is the ground truth, and $\hat{\mathbf{v}}$ the estimate obtained by the prox-linear net.

In particular, the prox-linear net was simulated with $T = 2$ and $K = 3$. The 'workhorse' Gauss-Newton method, a 6-layer 'plain-vanilla' FNN that has the same depth as our prox-linear net, and an 8-layer 'plain-vanilla' FNN that has roughly the same number of parameters as the prox-linear net, were simulated as baselines. The number of hidden units per layer in all NNs

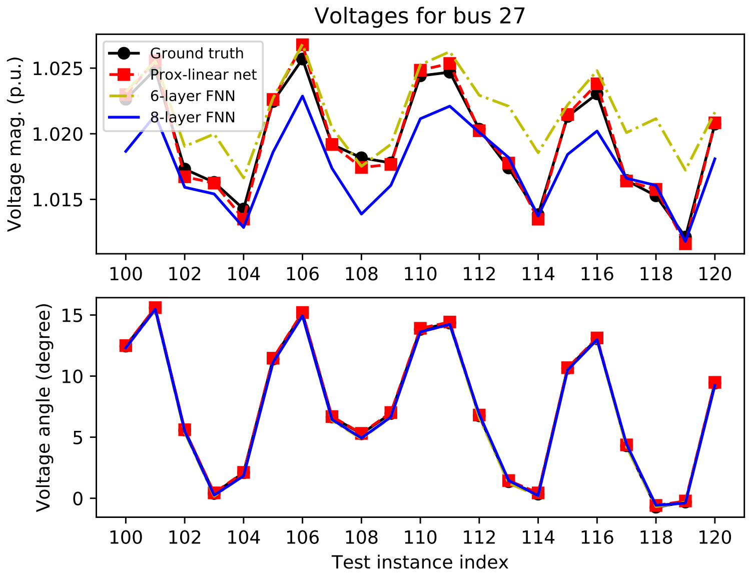


Figure 2.7: Estimation errors in voltage magnitudes and angles of bus 27 of the 57-bus system from test instances 100 to 120.

was kept equal to the dimension of the input, that is, $57 \times 2 = 114$ for the 57-bus system and $118 \times 2 = 236$ for the 118-bus system.

In the first experiment using the 57-bus system, a total of $7,676$ measurement-voltage $(\mathbf{z}_\tau, \mathbf{v}_\tau)$ pairs were generated, out of which the first $6,176$ pairs were used for training, and the rest were kept for testing. The average performance over 20 trials, was evaluated in terms of the average RMSEs over the $1,500$ testing examples for the prox-linear net, Gauss-Newton, 6-layer FNN, and 8-layer FNN, are 3.49×10^{-4}, 3.2×10^{-4}, 6.35×10^{-4}, and 9.02×10^{-4}, respectively. These numbers showcase competitive performance of the prox-linear net. Interestingly, when the number of hidden layers of ‘plain-vanilla’ FNNs increases from 6 to 8, the performance degrades due partly to the difficulty in training the 8-layer FNN.

As far as the computation time is concerned, the prox-linear net, Gauss-Newton, 6-layer FNN, and 8-layer FNN over $1,500$ testing examples are correspondingly 0.0973s, 14.22s, 0.0944s, and 0.0954s, corroborating the speedup advantage of NN-based PSSE over the traditional Gauss-Newton approach. The ground-truth voltages along with the estimates found by the prox-linear net, 6-layer FNN, and 8-layer FNN for bus 10 and bus 27 from test instances 100 to 120, are

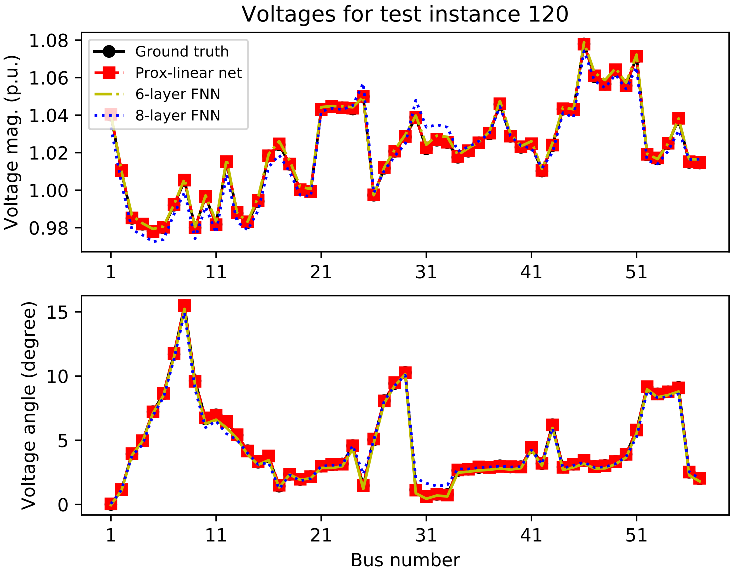


Figure 2.8: Estimation errors in voltage magnitudes and angles of all the 57 buses of the 57-bus system at test instance 120.

shown in Figs. 2.6 and 2.7, respectively. The true voltages and the estimated ones by NNs for all buses on test instance 120 are depicted in Fig. 2.8. Evidently, our prox-linear net based PSSE performs the best in all cases.

The second experiment tests our prox-linear net using the IEEE 118-bus system, where $18,528$ voltage-measurement pairs were simulated, with $14,822$ pairs employed for training and $3,706$ kept for testing. The average RMSEs over $3,706$ testing examples for the prox-linear net, Gauss-Newton, 6-layer FNN and 8-layer FNN, are $\mathbf{2.97\times 10^{-4}}$, 4.71×10^{-2}, 1.645×10^{-3}, and 2.366×10^{-3}, respectively. Clearly, our prox-linear net yields markedly improved performance over competing alternatives in this case (especially as the system size grows large). The Gauss-Newton approach performs the worst due to unbalanced grid parameters of this test system. Interestingly, it was frequently observed that the Gauss-Newton iterations minimize the weighted least-squares (WLS) objective function (resulting a loss smaller than 10^{-6}), but converge to a stationary point that is far away from the simulated ground-truth voltage. This is indeed due to the nonconvexity of the WLS function, for which multiple optimal solutions often exist. Depending critically on initialization, traditional optimization based solvers can unfortunately get stuck

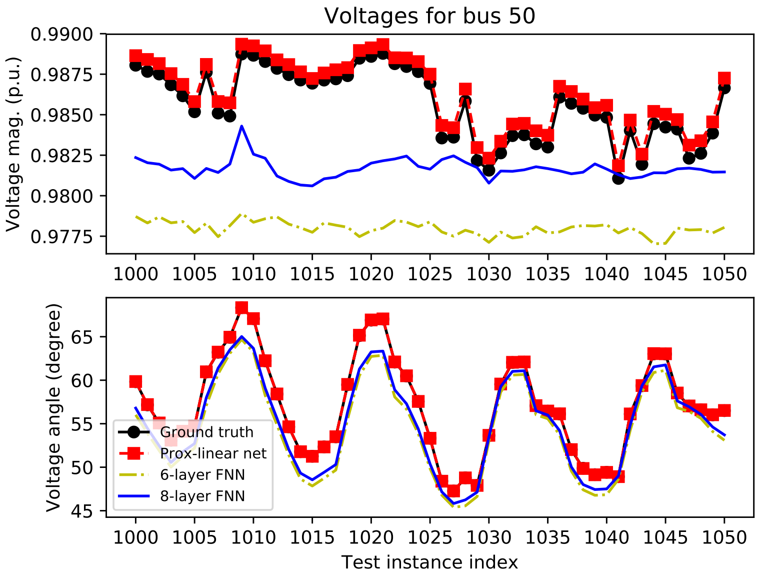


Figure 2.9: Estimation errors in voltage magnitudes and angles of bus 50 of the 118-bus system from instances $1,000$ to $1,050$.

at any of those points. In sharp contrast, data-driven NN-based approaches nicely bypass this hurdle.

In terms of runtime, the prox-linear net, Gauss-Newton, 6-layer FNN, and 8-layer FNN, over $3,706$ testing examples are correspondingly 0.3323s, 183.4s, 0.2895s, and 0.3315s, corroborating again the efficiency of NN-based approaches. The ground-truth voltage along with estimates obtained by the prox-linear net, 6-layer FNN, and 8-layer FNN, for bus 50 and bus 100 at test instances $1,000$ to $1,050$, are depicted in Figs. 2.9 and 2.10, respectively. In addition, the actual voltages and their estimates for the first fifty buses on test instance $1,000$ are depicted in Fig. 2.11. In all cases, our prox-linear net yields markedly improved performance relative to competing alternatives.

2.4.2 Simulations for State Forecating

This section examines our RNN based power system state forecasting scheme. The forecasting performance was evaluated in terms of the normalized RMSE $\|\check{\mathbf{v}} - \mathbf{v}\|_2/N$ of the forecast $\check{\mathbf{v}}$ relative to the ground truth $\mathbf{v}$.

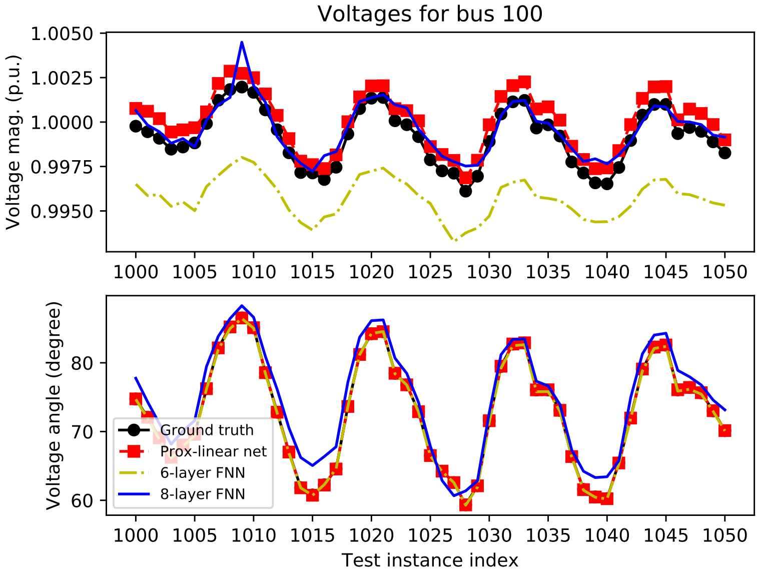


Figure 2.10: Estimation errors in voltage magnitudes and angles of bus 100 of the 118-bus system from instances $1,000$ to $1,050$.

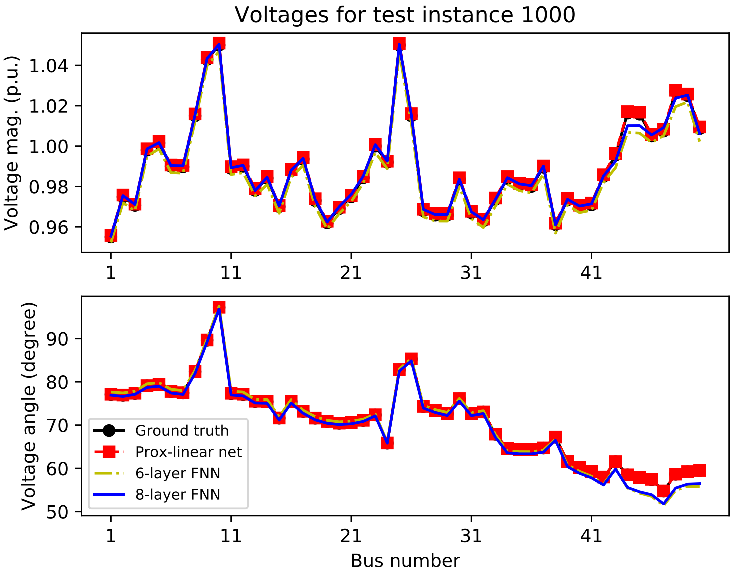


Figure 2.11: Estimation errors in voltage magnitudes and angles of the first 50 buses of the 118-bus system at test instance $1,000$.

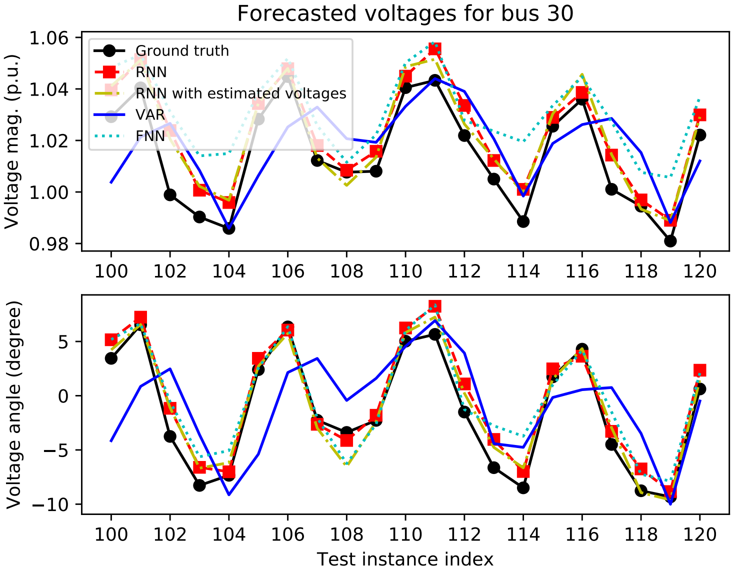


Figure 2.12: Forecasting errors in voltage magnitudes and angles of bus 30 of the 57-bus system from test instances 100 to 120.

Specifically, deep RNNs with $l = 3$, $r = 10$, and ReLU activation functions were trained and tested on the ground-truth voltage time series, and on the estimated voltage time series from the prox-linear net. We will refer to the latter as ‘RNNs with estimated voltages’ hereafter. The number of hidden units per layer in RNNs was kept the same as the input dimension, namely $57 \times 2 = 114$ for the 57-bus system, and $118 \times 2 = 236$ for the 118-bus system. For comparison, a single-hidden-layer FNN (2-layer FNN) [31], and a VAR(1) model [49] based state forecasting approaches were adopted as benchmarks. The average RMSEs over 20 Monte Carlo runs for the RNN, RNN with estimated voltages, 2-layer FNN, and VAR(1) are respectively 2.303×10^{-3}, 2.305×10^{-3}, 3.153×10^{-3}, and 6.772×10^{-3} for the 57-bus system, as well as 2.588×10^{-3}, 2.751×10^{-3}, 4.249×10^{-3}, 6.461×10^{-3} for the 118-bus system. These numbers demonstrate that our deep RNN with estimated voltages offers comparable forecasting performance relative to that with ground-truth voltages. Although both FNN and VAR(1) were trained and tested using ground-truth voltage time-series, they perform even worse than our RNN trained with estimated voltages.

The true voltages and their forecasts provided by the deep RNN, RNN with estimated

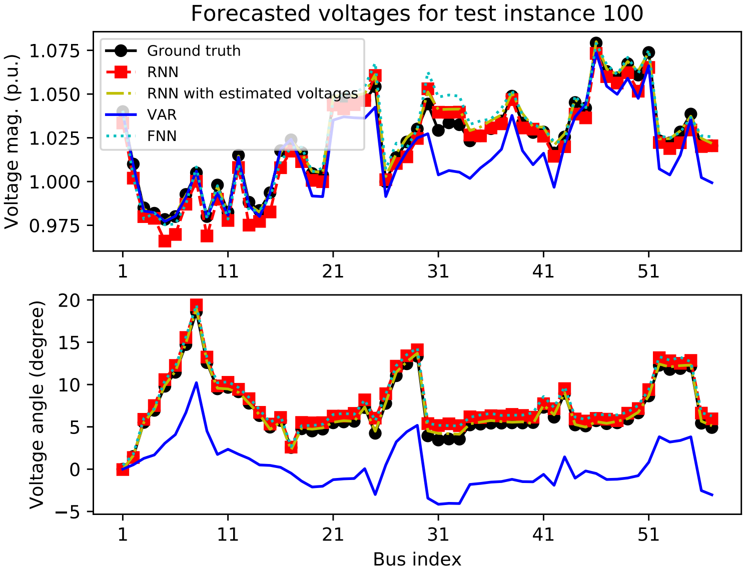


Figure 2.13: Forecasting errors in voltage magnitudes and angles of all the 57 buses of the 57-bus system at test instance 100.

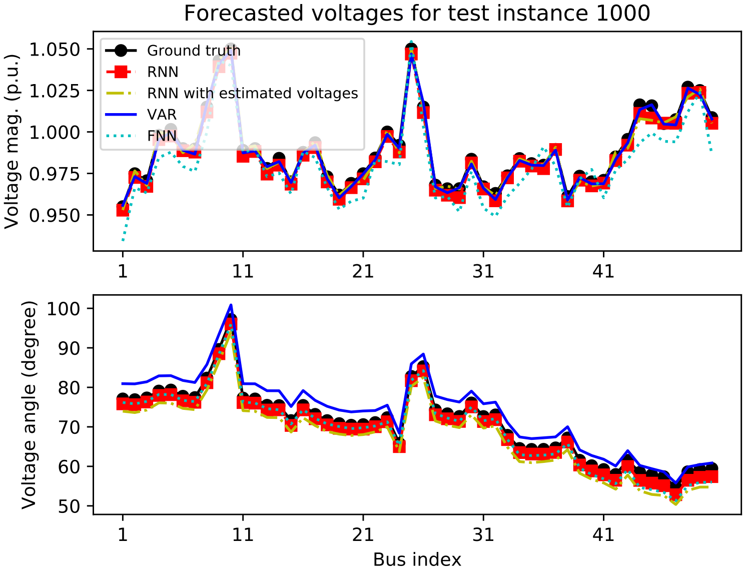


Figure 2.14: Forecasting errors in voltage magnitudes and angles of the first 50 buses of the 118-bus system at instance 1,000.

voltages, 2-layer FNN, and VAR(1) for bus 30 of the 57-bus system from test instances 100 to 120, as well as all buses on test instance 100, are reported in Figs. 2.12 and 2.13, accordingly. The ground-truth and forecast voltages for the first 50 buses of the 118-bus system on test instance $1,000$ are depicted in Fig. 2.14. Curves illustrate that our deep RNN based approaches perform the best in all cases.

Chapter 3

Accurate Grid Topology Identification

Section 3.1 introduces multi-kernel based partial correlations (MKPCs) for topology identification. The multi-kernel ridge regression problem arising in computing MKPCs is tackled efficiently through a tailored Frank-Wolfe scheme in Section 3.2. Section 3.3 specifies MKPC-based topology learning schemes for meshed power grids. Numerical tests are presented in Section 3.4.

3.1 Multi-kernel based Partial Correlations

Consider a power network consisting of N nodes that is modeled as a graph $\mathcal{G} := \{\mathcal{N}, \mathcal{L}\}$. Per node $n \in \mathcal{N}$, let $\mathbf{x}_n := [x_n[1] \dots x_n[T]]^\top$ be its associated time-series measurements. For example, $\mathbf{x}_n$ may consist of the nodal voltages for all time slots $t \in \mathcal{T} := \{1,\, 2,\, \dots,\, T\}$ [12] in the context of power network topology inference. The task of grid connectivity identification can be described as follows: given nodal time-series $\{\mathbf{x}_n\}_{n \in \mathcal{N}}$, infer the network topology, namely recover the edge set $\mathcal{L}$.

To reveal nodal connectivity, the PC method, which is effective in capturing unmediated linear influence between nodes, has been widely used [45, 12, 60]. Letting $\backslash ij$ denote all but the (i, j) nodes, consider the error vector $\tilde{\mathbf{x}}_i := \mathbf{x}_i - \hat{\mathbf{x}}_{i|\backslash ij}$, where $\hat{\mathbf{x}}_{i|\backslash ij}$ is the estimate of $\mathbf{x}_i$ based on $\{\mathbf{x}_n\}_{n \in \backslash ij}$. The empirical PC coefficient between $\mathbf{x}_i$ and $\mathbf{x}_j$ is given by [45]

$$\hat{\rho}_{ij} := \frac{(\tilde{\mathbf{x}}_i - \bar{\tilde{\mathbf{x}}}_i)^\top (\tilde{\mathbf{x}}_j - \bar{\tilde{\mathbf{x}}}_j)}{\|\tilde{\mathbf{x}}_i - \bar{\tilde{\mathbf{x}}}_i\|_2 \|\tilde{\mathbf{x}}_j - \bar{\tilde{\mathbf{x}}}_j\|_2} \tag{3.1}$$

where $\bar{\tilde{\mathbf{x}}}_i := (1/T)\sum_{t=1}^{T} \tilde{x}_i[t]\mathbf{1}$. Having computed all PC coefficients $\{\hat{\rho}_{ij}\}_{(i,j)\in\mathcal{N}\times\mathcal{N}}$, determining whether node i is connected with node j entails a hypothesis test. Specifically, an edge between i and j is declared present if $|\hat{\rho}_{ij}| \geq \tau$, where $\tau > 0$ is a threshold that trades off the relative true positive for the false positive decisions.

The linear PC method assumes that $\hat{\mathbf{x}}_{i|\backslash ij}$ is a linear function of $\{\mathbf{x}_n\}_{n\in\backslash ij}$. Nonetheless, the dependence of $\mathbf{x}_i$ on $\{\mathbf{x}_n\}_{n\in\backslash ij}$ is nonlinear in power grids. To circumvent the limitation of linear estimators, an ℓ_2-norm based MKPC has been studied in [60], which works as follows. Let $\{n_{1\backslash ij}, \ldots, n_{N-2\backslash ij}\}$ enumerate all nodes in $\backslash ij$, and $\boldsymbol{\chi}_{\backslash ij}[t] := \left[x_{n_{1\backslash ij}}[t], \ldots, x_{n_{N-2\backslash ij}}[t]\right]^\top$ collect all measurements at time t. By replacing $\boldsymbol{\chi}_{\backslash ij}[t]$ with its lifted image via a kernel-based feature map $\boldsymbol{\phi}(\boldsymbol{\chi}_{\backslash ij}[t])$, a nonlinear data generation model can be postulated as

$$x_i[t] = \langle \boldsymbol{\phi}(\boldsymbol{\chi}_{\backslash ij}[t]), \boldsymbol{\beta}_i \rangle + \epsilon_i[t] \tag{3.2}$$

where $\boldsymbol{\beta}_i$ is a parameter vector to learn, and $\epsilon_i[t]$ captures modeling inaccuracies. Along the lines of ridge regression, an estimate of $\boldsymbol{\beta}_i$ per node i can be found as

$$\hat{\boldsymbol{\beta}}_i := \arg\min_{\boldsymbol{\beta}} \quad \frac{C}{N}\|\boldsymbol{\xi}_i\|_2^2 + \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 \tag{3.3a}$$

$$\text{s. to} \quad \boldsymbol{\xi}_i = \mathbf{x}_i - \boldsymbol{\Phi}_{\backslash ij}^\top \boldsymbol{\beta} \tag{3.3b}$$

where $\boldsymbol{\Phi}_{\backslash ij} := [\boldsymbol{\phi}(\boldsymbol{\chi}_{\backslash ij}[1]), \ldots, \boldsymbol{\phi}(\boldsymbol{\chi}_{\backslash ij}[T])]$, and $C \geq 0$ is some given constant. Since $\boldsymbol{\phi}(\boldsymbol{\chi}_{\backslash ij}[t])$ has high (potentially infinite) dimension, the dual of (3.3) which has only T variables (as many as the number of scalar equality constraints in (3.3b)) will be used henceforth. Specifically, the dual of (3.3) can be succinctly written as [93]

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^T} \frac{1}{2}\left(-\mu\boldsymbol{\alpha}^\top\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top\mathbf{x}_i - \boldsymbol{\alpha}^\top\mathbf{K}_{\backslash ij}\boldsymbol{\alpha}\right) \tag{3.4}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^T$ denotes the Lagrange multiplier associated with (3.3b), constant $\mu := N/(2C)$, and $\mathbf{K}_{\backslash ij} := \boldsymbol{\Phi}_{\backslash ij}^\top \boldsymbol{\Phi}_{\backslash ij}$. Note that the maximizer of (3.4) can be found in closed form as $\hat{\boldsymbol{\alpha}}_i = (\mathbf{K}_{\backslash ij} + \mu\mathbf{I}_N)^{-1}\mathbf{x}_i$. Appealing to [60, Eq. (4)], the least-squares estimate of $x_i[t]$ can be obtained as

$$\hat{x}_{i|\backslash ij}[t] = \langle \boldsymbol{\phi}(\boldsymbol{\chi}_{\backslash ij}[t]), \hat{\boldsymbol{\beta}}_i \rangle = \langle \boldsymbol{\phi}(\boldsymbol{\chi}_{\backslash ij}[t]), \boldsymbol{\Phi}_{\backslash ij}\hat{\boldsymbol{\alpha}}_i \rangle \tag{3.5}$$

which entails computing inner products between feature vectors $\left\{\langle \boldsymbol{\phi}(\boldsymbol{\chi}_{\backslash ij}[t]), \boldsymbol{\phi}(\boldsymbol{\chi}_{\backslash ij}[t']) \rangle\right\}$, that is oftentimes of high dimensions. Fortunately, such a costly computation can be significantly reduced by invoking the so-termed *kernel trick* [96], which allows computing the wanted inner products in (3.5) through solely evaluating the kernel function for all pairs $(t, t') \in \mathcal{T} \times \mathcal{T}$:

$$\kappa(\boldsymbol{\chi}_{\backslash ij}[t], \boldsymbol{\chi}_{\backslash ij}[t']) = \langle \boldsymbol{\phi}(\boldsymbol{\chi}_{\backslash ij}[t]), \boldsymbol{\phi}(\boldsymbol{\chi}_{\backslash ij}[t']) \rangle.$$

Clearly, the accuracy of the kernel-based estimates in (3.5) depends highly on the selected kernel function $\kappa(\cdot,\cdot)$ [96]. To choose a suitable kernel, multi-kernel learning (MKL) is invoked here [72], which seeks $\kappa(\cdot,\cdot)$ as a nonnegative linear combination of user-defined kernel functions; that is, $\kappa(\cdot,\cdot) := \sum_{m=1}^{M} \theta_m \kappa_m(\cdot,\cdot)$. The coefficients $\{\theta_m \geq 0\}_{m=1}^{M}$ can be deciphered from data via solving [72]

$$\boldsymbol{\theta}^* := \arg \min_{\boldsymbol{\theta} \in \Theta_p} \max_{\boldsymbol{\alpha} \in \mathbb{R}^N} -\mu \boldsymbol{\alpha}^\top \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{x}_i - \sum_{m=1}^{M} \theta_m \boldsymbol{\alpha}^\top \mathbf{K}_{m\backslash ij} \boldsymbol{\alpha} \tag{3.6}$$

where the kernel matrix is given by

$$\mathbf{K}_{m\backslash_{ij}}(t, t') := \kappa_m(\boldsymbol{\chi}_{\backslash ij}[t], \boldsymbol{\chi}_{\backslash ij}[t']),\ \forall (t, t') \in \mathcal{T} \times \mathcal{T} \tag{3.7}$$

the constraint set $\Theta_p := \ \{\boldsymbol{\theta} \in \mathbb{R}^M \,|\, \boldsymbol{\theta} \geq \mathbf{0},\ \|\boldsymbol{\theta}\|_p \leq \Lambda\}$ with $p \geq 1$, and $\Lambda > 0$ is a pre-selected constant. For notational brevity, define also

$$\begin{aligned} F(\boldsymbol{\theta}) &:= \max_{\boldsymbol{\alpha} \in \mathbb{R}^N} \ -\mu \boldsymbol{\alpha}^\top \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{x}_i - \sum_{m=1}^{M} \theta_m \boldsymbol{\alpha}^\top \mathbf{K}_{m\backslash ij} \boldsymbol{\alpha} \\ &= \mathbf{x}_i^\top \Big(\mu \mathbf{I} + \sum_{m=1}^{M} \theta_m \mathbf{K}_{m\backslash ij} \Big)^{-1} \mathbf{x}_i \end{aligned} \tag{3.8}$$

based on which we can rewrite (3.6) as follows

$$\boldsymbol{\theta}^* := \arg \min_{\boldsymbol{\theta} \in \Theta_p} F(\boldsymbol{\theta}). \tag{3.9}$$

Upon obtaining $\boldsymbol{\theta}^*$ and $\mathbf{K}_{\backslash ij} = \sum_{m=1}^{M} \theta_m^* \mathbf{K}_{m\backslash ij}$, it is easy to obtain the estimate (cf. (3.5))

$$\hat{\mathbf{x}}_{i|\backslash ij} = \mathbf{K}_{\backslash ij} (\mu \mathbf{I} + \mathbf{K}_{\backslash ij})^{-1} \mathbf{x}_i. \tag{3.10}$$

Algorithm 2 Frank-Wolfe [38]

```
Initialize $k = 0$, $\mathbf{y}^0 \in \mathcal{Y}$
while stopping_criterion not met do
    Compute $\mathbf{s}^k = \arg\min_{\mathbf{s}\in\mathcal{Y}} \ \mathbf{s}^\top \nabla f(\mathbf{y}^k)$
    Update $\mathbf{y}^{k+1} = (1-\eta_k)\mathbf{y}^k + \eta_k \mathbf{s}^k$
    $k \leftarrow k+1$
end while
```

Our new formulation for multi-kernel ridge regression (MKRR) here generalizes the nonlinear estimator in [60] beyond $p = 2$. This generalization is well motivated because the estimation error $\|\mathbf{x}_i - \hat{\mathbf{x}}_{i|\setminus ij}\|_2$ can be reduced by choosing appropriate p values [69]. When selecting $p = 1$, the nonlinear estimator will rely on only a few kernels from the user-specified dictionary of kernels. In other words, setting $p = 1$ leads to sparse $\boldsymbol{\theta}^*$. Exploiting sparsity turns out to be beneficial in a number of signal processing applications such as tackling the phase retrieval problem [110, 134, 108].

3.2 Efficient Solver for Multi-kernel Ridge Regression

Though the ℓ_p-norm based MKL (3.9) can lead to improved estimation accuracy, solving the resulting ℓ_p-norm constrained optimization problem is typically not an easy task except for a few special cases such as $p = 2$ [69, 129]. For this reason, we put forward an efficient and easy-to-implement solver by leveraging the projection-free Frank-Wolfe algorithm (a.k.a., the conditional gradient method) [58]. As a brief review, the Frank-Wolfe algorithm targets the generic convex problem

$$\mathbf{y}^* \in \arg\min_{\mathbf{y}\in\mathcal{Y}} \ f(\mathbf{y}) \tag{3.11}$$

where f is assumed differentiable, and the feasible set $\mathcal{Y}$ is compact. Listed as Algorithm 3, the Frank-Wolfe solver starts with an arbitrary point $\mathbf{y}^0$, and subsequently iterates between the updates for $k \geq 0$ as follows

$$\mathbf{s}^k \in \arg\min_{\mathbf{s}\in\mathcal{Y}} \ \mathbf{s}^\top \nabla f(\mathbf{y}^k) \tag{3.12a}$$

$$\mathbf{y}^{k+1} = (1-\eta_k)\mathbf{y}^k + \eta_k \mathbf{s}^k \tag{3.12b}$$

where the diminishing step size $\eta_k := 2/(k^\rho + 2)$ with ρ being a predefined constant in $(0.5, 1]$ [135]. It has been shown that the successive iterates $\{\mathbf{y}^k\}$ are always feasible due to $\eta^0 = 1,\ \mathbf{y}^1 = \mathbf{s}^0 \in \mathcal{Y}$, and $\mathbf{s}^k \in \mathcal{Y}$ for $k \geq 1$ [58].

Since $F(\boldsymbol{\theta})$ in (3.9) is convex and differentiable [129], and Θ_p is compact, the ℓ_p-norm regularized MKRR (3.9) is in the form of (3.11) required by the Frank-Wolfe iterations (3.12). To employ (3.12a), the gradient of $F(\boldsymbol{\theta})$ is given by

$$\nabla F(\boldsymbol{\theta}) = [-\hat{\boldsymbol{\alpha}}^\top \mathbf{K}_{1\backslash ij}\hat{\boldsymbol{\alpha}}, \ldots, -\hat{\boldsymbol{\alpha}}^\top \mathbf{K}_{M\backslash ij}\hat{\boldsymbol{\alpha}}]^\top \tag{3.13}$$

with

$$\hat{\boldsymbol{\alpha}} = \Big(\mu \mathbf{I} + \sum_{m=1}^{M} \theta_m \mathbf{K}_{m\backslash ij}\Big)^{-1} \mathbf{x}_i. \tag{3.14}$$

Since $\{\mathbf{K}_{m\backslash ij}\}_{m=1}^{M}$ are positive semidefinite [96], we have $\nabla F(\boldsymbol{\theta}) \leq \mathbf{0}, \forall \boldsymbol{\theta} \in \Theta_p$. Applying (3.12a) to (3.9) yields

$$\mathbf{s}^k \in \arg\min_{\mathbf{s}\in\Theta_p} \mathbf{s}^\top \nabla F(\boldsymbol{\theta}^k) \tag{3.15}$$

which we will show possible to express in closed from. Indeed, upon introducing $\mathbf{g}^k := -\nabla F(\boldsymbol{\theta}^k)$, (3.15) can be equivalently posed as

$$\mathbf{s}^k \in \arg\max_{\mathbf{s}\in\Theta_p} \mathbf{s}^\top \mathbf{g}^k. \tag{3.16}$$

By Hölder inequality and for all $\mathbf{s}^k \in \Theta_p$, it holds that

$$(\mathbf{s}^k)^\top \mathbf{g}^k \leq \|\mathbf{s}^k\|_p \|\mathbf{g}^k\|_q \leq \Lambda \|\mathbf{g}^k\|_q \tag{3.17}$$

where the norms satisfy $1/p + 1/q = 1$. Using further $\mathbf{g}^k = -\nabla F(\boldsymbol{\theta}^k) \geq \mathbf{0}$, it can be readily deduced that the solution to (3.15) is found when equalities in (3.17) are attained

$$s^k[m] = \Lambda \frac{\big(g^k[m]\big)^{q-1}}{\|\mathbf{g}^k\|_q^{q-1}}, \quad 1 \leq m \leq M \tag{3.18}$$

where $q := p/(p-1)$, with $q = \infty$, if $p = 1$. On the other hand, $q = 1$, if $p = \infty$, and in this case, all nonzero entries of $\mathbf{s}^k$ are equal. Upon finding the conditional direction $\mathbf{s}^k$, the algorithm updates the estimate via (3.12b), and iterates until certain convergence criterion is met.

Algorithm 3 Frank-Wolfe based ℓ_p-norm MKRR solver.

```
Input: $\mu$, $\Lambda$, $\eta_k$, $\mathbf{x}_i$, and $\{\mathbf{K}_{m\setminus ij}\}_{m=1}^{M}$
Initialize $\boldsymbol{\theta}^0 = \mathbf{0}$
for $k = 0, 1, \ldots$ do
    Compute $\hat{\boldsymbol{\alpha}}^k$ via (3.14) and $\nabla F(\boldsymbol{\theta}^k)$ from (3.13)
    Find $\mathbf{s}^k$ via (3.18)
    Update $\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \eta_k(\mathbf{s}^k - \boldsymbol{\theta}^k)$
end for
```

The proposed solver for the ℓ_p-regularized MKL (3.9) is summarized as Algorithm 3. Notably, with step size $\eta_k = 2/(k+2)$, it converges to $\boldsymbol{\theta}^*$ at sublinear rate [58], namely

$$F(\boldsymbol{\theta}^k) - F(\boldsymbol{\theta}^*) \leq \mathcal{O}\left(1/k\right).$$

It is also worth stressing that applicability of Algorithm 3 goes beyond MKPC, since any ℓ_p-norm based MKRR can be solved via Algorithm 3.

3.3 Topology Learning for Meshed Networks

Consider a meshed power network of N buses indexed by the set $\mathcal{N} := \{1, 2, \ldots, N\}$. Let $\boldsymbol{\psi}[t] := [\psi_1[t], \ldots, \psi_N[t]]^\top$ collect the voltage angles at all buses at time t. The power grid topology learning problem can be stated as follows: *given voltage angle time-series measurements* $\{\boldsymbol{\psi}[t]\}_{t=1}^{T}$, *identify the grid topology.* Expressed differently, our goal is to infer the connectivity of buses based on T voltage angle-vector measurements.

For any bus pair $(i,j) \in \mathcal{N} \times \mathcal{N}$, define $\mathbf{x}_i := [\psi_i[1], \ldots, \psi_i[T]]^\top$ to collect voltage angle time-series associated with bus i, and likewise $\mathbf{x}_j := [\psi_j[1], \ldots, \psi_j[T]]^\top$. Having selected candidate kernel functions $\{\kappa_m\}_{m=1}^{M}$, one first forms the kernel matrices $\{\mathbf{K}_{m\setminus ij}\}_{m=1}^{M}$ as in (3.7), and learns the best kernel combination $\mathbf{K}_{\setminus ij}^{(l)} = \sum_{m=1}^{M} \theta_{l,m}^* \mathbf{K}_{m\setminus ij}$ using Algorithm 3. For both $\mathbf{x}_i$ and $\mathbf{x}_j$, the next step is to obtain their ℓ_p-norm MKRR based nonlinear estimates $\hat{\mathbf{x}}_{i|\setminus ij} = \mathbf{K}_{\setminus ij}^{(i)}(\mu\mathbf{I} + \mathbf{K}_{\setminus ij}^{(i)})^{-1}\mathbf{x}_i$ and likewise for $\hat{\mathbf{x}}_{j|\setminus ij}$. The estimated PC coefficient $\hat{\rho}_{ij}$ can then be found by means of (3.1), and an edge connecting buses i and j is claimed to be present if $|\hat{\rho}_{ij}| > \tau$. This pairwise hypotheses test is repeated for all pairs $(i,j) \in \mathcal{N} \times \mathcal{N}$. The proposed ℓ_p-regularized MKRR based network topology inference scheme is tabulated in Algorithm 4.

Algorithm 4 Topology learning using ℓ_p-norm MKPC.

```
Input: ${\boldsymbol{\psi}[t]}_{t=1}^{T}$, $\{\kappa_m\}_{m=1}^{M}$, and threshold $\tau > 0$
1: for $(i,j) \in \mathcal{N} \times \mathcal{N}, (i<j)$ do
2:     Form $\{\mathbf{K}_{m\backslash ij}\}_{m=1}^{M}$ via (3.7)
3:     Find $\boldsymbol{\theta}_i^*$ using Algorithm 1 with input $\mathbf{x}_i$
4:     Form $\mathbf{K}_{\backslash ij}^{(i)} = \sum_{m=1}^{M} \theta_{i,m}^* \mathbf{K}_{m\backslash ij}$
5:     Obtain $\hat{\mathbf{x}}_{i|\backslash ij} = \mathbf{K}_{\backslash ij}^{(i)}(\mu\mathbf{I} + \mathbf{K}_{\backslash ij}^{(i)})^{-1}\mathbf{x}_i$
6:     Repeat Steps 3-5 with input $\mathbf{x}_j$ to get $\hat{\mathbf{x}}_{j|\backslash ij}$
7:     Find $\hat{\rho}_{ij}$ using (3.1)
8:     Draw an edge between bus $i$ and bus $j$ if $|\hat{\rho}_{ij}| > \tau$
9: end for
```

3.4 Numerical Tests

This section evaluates the performance of Algorithms 3 and 4 using the voltage angle data from the IEEE 14-bus benchmark [104]. To test the developed methods in a more realistic setting, the real load data from 2012 Global Energy Forecasting Competition [59] were used in all experiments. Specifically, the first ten-day loads of zones 1–14 were normalized to match the scale of active power demands in the IEEE 14-bus system, which is further corrupted by random noise generated from a uniform distribution on [0, 1]. Voltage angle measurements $\{\boldsymbol{\psi}(t)\}_{t=1}^{T}$ across $T = 240$ time slots were then obtained by solving the AC power flow equations using the publicly available MATPOWER toolbox [139]. To obtain a nonlinear voltage angle estimator, a total of 20 kernels were employed to form the dictionary, which consists of 10 polynomial kernels whose orders vary by 1 from 1 to 10, as well as 10 Gaussian kernels with variances distributed uniformly from 0.5 to 5. The regularization coefficients in (3.6) were set as $\mu = 1$ and $\Lambda = 3$.

The first experiment assesses the convergence and computational performance of Algorithm 3. To serve as a benchmark, (3.9) is first equivalently reformulated as a semidefinite program (SDP) by introducing an auxiliary variable $w \in \mathbb{R}$ such that $w \geq F(\boldsymbol{\theta})$, whose global optimum is found by SeDuMi [99]; see also [129]

$$\begin{aligned} \boldsymbol{\theta}^* := \arg \min_{\boldsymbol{\theta}\in\Theta,\, w\geq 0} \quad & w \\ \text{s. to} \quad & \begin{pmatrix} \mu\mathbf{I} + \sum_{m=1}^{M} \theta_m \mathbf{K}_m & \mathbf{x}_i \\ \mathbf{x}_i^\top & w \end{pmatrix} \succeq \mathbf{0}. \end{aligned} \tag{3.19}$$

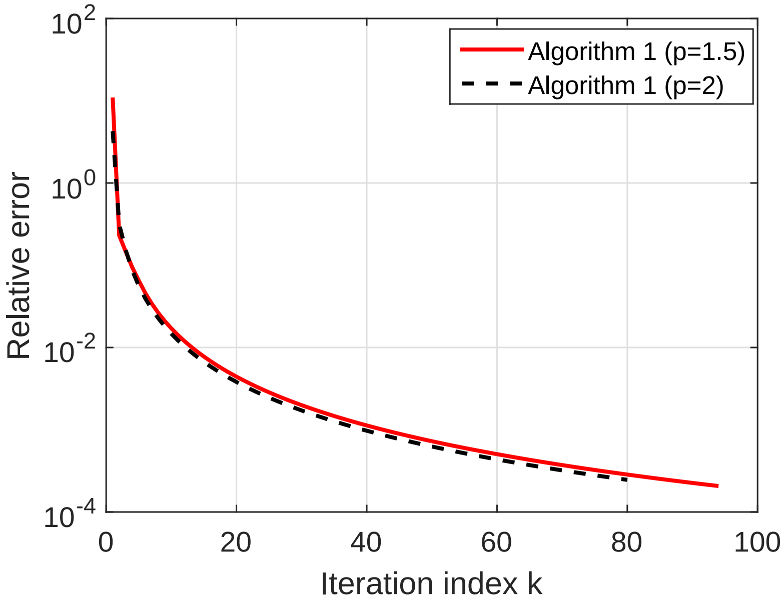


Figure 3.1: Cost convergence for Algorithm 3.

Figure 3.1 depicts the evolution of the relative error $(F(\boldsymbol{\theta}^k) - F(\boldsymbol{\theta}^*))/\boldsymbol{\theta}^*$ of (3.9) with $p = 1.5$ and $p = 2$, where $\{\mathbf{x}_n\}_{n=1}^{2}$ and $\{\mathbf{x}_n\}_{n=5}^{14}$ were used to predict $\mathbf{x}_3$. Algorithm 3 with $\eta_k = 2/(k+2)$ was run until $\|\hat{\boldsymbol{\alpha}}^{k+1} - \hat{\boldsymbol{\alpha}}^{k}\|_2 < 10^{-5}$, or the maximum iteration number 100 was reached. It can be observed that Algorithm 3 converges to the global minimum roughly at rate $\mathcal{O}(1/k)$.

The computation times for Algorithm 3 and SeDuMi are 0.23 and 9.63 secs ($p = 1.5$); and 0.20 and 7.49 secs ($p = 2$). It is clear that Algorithm 3 enjoys a clear speedup advantage over the off-the-shelf solver. This advantage is mainly due to the simple closed-form updates of (3.18), and it is important for the scalability of Algorithm 4 which requires solving (3.9) $N^2 - N$ times.

The second experiment tests the topology recovery performance of Algorithm 4 using the IEEE 14-bus system. Note that including the reference bus renders the covariance matrix of voltage angles non-invertible since the voltage angle at the reference bus remains always 0. Excluding the reference bus, a total of 13 buses and 18 lines are to be recovered. The performance of Algorithm 4 with $p = 1.5$ and $p = 2$ was compared to the performance of linear PC- and concentration matrix-based methods [29] in terms of the empirical receiver operating characteristics (ROC) in Fig. 3.2. For Algorithm 4 and its linear PC counterpart, the ROC curves were

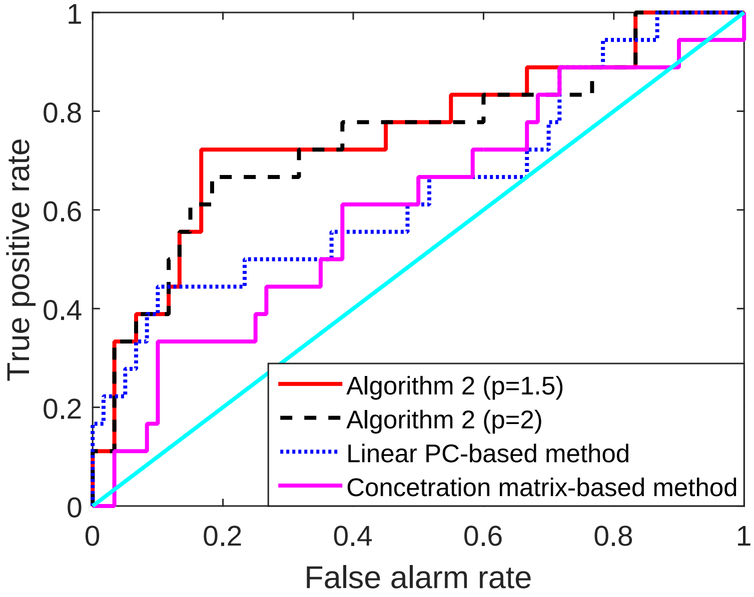


Figure 3.2: ROC curves for topology inference of IEEE 14-bus benchmark from voltage angle data.

obtained using $\{|\hat{\rho}_{ij}|\}_{(i,j)\in\mathcal{N}\times\mathcal{N}}$ as test statistics, whereas for the concentration matrix-based method, entries of the negative concentration matrix were employed as test statistics. The area under the curve (AUC) for Algorithm 4 with $p = 1.5$ and $p = 2$, linear PC-, and concentration matrix-based methods is 0.755, 0.743, 0.646, and 0.604, accordingly. This shows improved recovery performance of the proposed Algorithm 4 relative to existing alternatives. Comparing the red and black curves in Fig. 3.2 further suggests that the grid topology identification can benefit from the ℓ_p-regularized MKPC with p values different from 2.

Chapter 4

Decentralized Electric Vehicle Charging Coordination

Section 4.1 formulates the optimal EV charging control problem. The Frank-Wolfe based optimal decentralized charging scheme and its real-time counterpart are delineated in Section 4.2. Section 4.3 introduces the randomized block Frank-Wolfe (RB-FW) method for solving general convex programs, and applies RB-FW to developing asynchronous charging schemes. Section 4.4 presents numerical tests.

4.1 Electric Vehicle Charging Model

Consider an EV scheduler coordinating the charging of M EVs over a period of T consecutive time slots comprising the set $\mathcal{T} := \{1, \ldots, T\}$. The time slot duration ΔT can range from minutes to an hour, depending on charging specifications, the granularity of load forecasts, as well as communication and computation capabilities. Let $e_m(t)$ denote the energy charge for vehicle m at slot t with $m = 1, \ldots, M$, and $t \in \mathcal{T}$. Supposing that operational slots have equal duration, the terms power and energy will be used interchangeably. The charge $e_m(t)$ can range from zero to its maximum value $\bar{e}_m(t)$. Apparently, a vehicle can be charged only when it is connected to the grid. If $\mathcal{T}_m \subseteq \mathcal{T}$ is the set of time slots (not necessarily consecutive) that vehicle

m is connected to the grid, then for all $t \in \mathcal{T}$ we have

$$\bar{e}_m(t) = \begin{cases} \Delta T \bar{e}_m & , \ t \in \mathcal{T}_m \\ 0 & , \ \text{otherwise} \end{cases}$$

where $\bar{e}_m$ is the maximum charging rate determined by the battery of vehicle m. Let $\mathbf{e}_m := [e_m(1) \ \cdots \ e_m(T)]^\top$ be the charging profile for EV m. Profile $\mathbf{e}_m$ should belong to the compact and convex set

$$\mathcal{E}_m := \{\mathbf{e}_m : \mathbf{e}_m^\top \mathbf{1} = R_m, \ 0 \leq e_m(t) \leq \bar{e}_m(t) \ \forall \, t \in \mathcal{T}\} \tag{4.1}$$

where R_m is the total energy needed by EV m. The latter depends on the initial state of charge, the desired state of charge, and the efficiency of the battery.

Through coordinated charging of EVs, various objectives can be achieved, such as minimizing charging costs or valley-filling. Optimal EV charging can be posed as the ensuing optimization problem [41]

$$\min_{\{\mathbf{e}_m\}_{m=1}^M} \quad C(\{\mathbf{e}_m\}) := \sum_{t=1}^T C_t \left(d(t) + \sum_{m=1}^M e_m(t) \right) \tag{4.2a}$$

$$\text{s.to} \ \ \mathbf{e}_m \in \mathcal{E}_m, \ \forall \, m = 1, \ldots, M \tag{4.2b}$$

where the energy costs $C_t(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ are chosen to be convex and differentiable. For charging cost minimization, $\{C_t\}_{t=1}^T$ can be linear or quadratic [17, 122]; e.g., $C_t(x) = x^2/2$ for all t, when it comes to the valley-filling task. Parameters $\{d(t)\}_{t=1}^T$ capture the base loads for the EV scheduler, which are assumed inelastic and known in advance. To facilitate scheduling, each EV controller is capable of two-way communication and execution of simple computation tasks. Building on the Frank-Wolfe algorithm summarized in Algorithm 3, an optimal decentralized solver for (4.2) is developed and contrasted to state-of-the-art alternatives next.

4.2 Optimal Decentralized Charging Control

Observe that the total number of variables involved in (4.2) is MT. Therefore, although (4.2) is a convex problem, solving it is a non-trivial task, particularly for large EV fleets and/or decreasing control intervals ΔT. To derive a scalable solver, the Frank-Wolfe method summarized in Algorithm 3 is deployed next. In the first Frank-Wolfe step, the gradient of the cost in (4.2) with respect to $\{\mathbf{e}_m\}_{m=1}^M$ must be obtained. Critically, due to the problem structure, the per-vehicle partial gradients of the cost are all *identical* to

$$\nabla_{\mathbf{e}_m} C(\{\mathbf{e}_m\}) = \mathbf{g}, \quad m = 1, \ldots, M.$$

It can be readily checked that at the k-th iteration, the t-th entry of the common partial gradient $\mathbf{g} \in \mathbb{R}^T$ evaluated at $\{\mathbf{e}_m^k\}$ is

$$g^k(t) = \nabla_{e_m^k(t)} C_t \left(d(t) + \sum_{m=1}^{M} e_m^k(t) \right), \; t = 1, \ldots, T. \tag{4.3}$$

Applying (3.12a) to the problem at hand requires solving

$$\{\mathbf{r}_m^k\}_{m=1}^M \in \arg \min_{\{\mathbf{r}_m \in \mathcal{E}_m\}_{m=1}^M} \sum_{m=1}^{M} \mathbf{r}_m^\top \mathbf{g}^k \tag{4.4}$$

which is separable across vehicles. Thus, given $\mathbf{g}^k$, vehicle m needs to solve the linear program

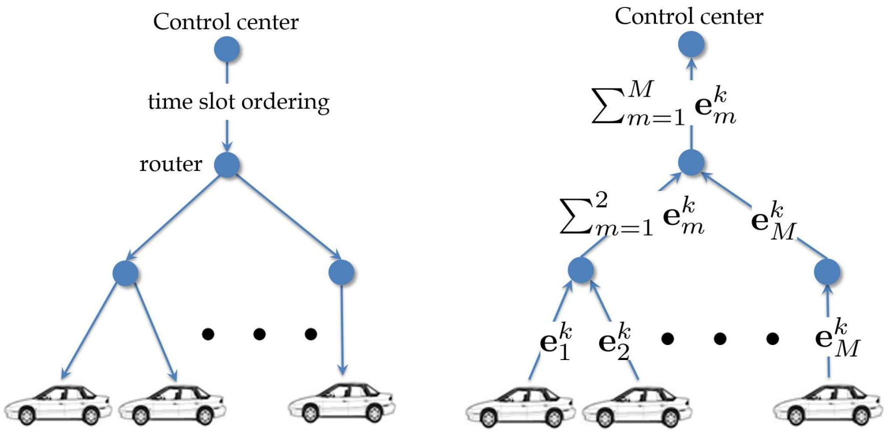
$$\mathbf{r}_m^k \in \arg \min_{\mathbf{r}_m \in \mathcal{E}_m} \mathbf{r}_m^\top \mathbf{g}^k. \tag{4.5}$$

Problem (4.5) involves a linear cost minimized over a weighted budget and box constraints. The key observation here is that due to the aforementioned structure, problem (4.5) can be solved by a simple sorting algorithm [15, Chap. 4]: The entries of $\mathbf{g}^k$ are first sorted in increasing order as

$$g^k(t_1^k) \leq g^k(t_2^k) \leq \ldots \leq g^k(t_T^k). \tag{4.6}$$

Since the problems in (4.5) share vector $\mathbf{g}^k$ for all m, the sorting operation is performed only once by the EV aggregator/charging station. Then, for vehicle m, we need to find the index J_m^k for which

$$\sum_{j=1}^{J_m^k} \bar{e}_m(t_j^k) \leq R_m \text{ and } \sum_{j=1}^{J_m^k+1} \bar{e}_m(t_j^k) > R_m. \tag{4.7}$$



(a) Control center broadcasts time slot pricing ordering (from cheapest to most expensive) to EV controllers.

(b) Summations of charging profiles are transmitted to charging control center.

Figure 4.1: Information exchange for Algorithm 5 at iteration k.

Subsequently, the entries of the minimizer $\mathbf{r}_m^k$ of (4.5) can be computed per vehicle m as

$$r_m^k(t_j^k) = \begin{cases} \bar{e}_m(t_j^k) & , j = 1, \ldots, J_m^k - 1 \\ R_m - \sum_{j=1}^{J_m^k - 1} \bar{e}_m(t_j^k) & , j = J_m^k \\ 0 & , j = J_m^k + 1, \ldots, T \end{cases} . \tag{4.8}$$

The solution in (4.8) simply selects the maximum possible charge during the cheapest time slots in a greedy fashion. Interestingly, finding $\mathbf{r}_m^k$ from (4.8) requires knowing solely the rank order (smallest to largest) rather than the actual entries of the gradient vector $\mathbf{g}^k$.

The second Frank-Wolfe step updates the charging profiles via the convex combinations

$$\mathbf{e}_m^{k+1} = (1 - \eta_k)\mathbf{e}_m^k + \eta_k \mathbf{r}_m^k \tag{4.9}$$

for all vehicles $m = 1, \ldots, M$.

To practically implement (4.3)–(4.9) during iteration k, the charging control center evaluates the cost gradients $\{g^k(t)\}_{t \in \mathcal{T}}$ defined in (4.3), and sorts them to determine the time slot ordering $\{t_1^k, t_2^k, \ldots, t_T^k\}$. This sorting operation can be performed using for example the 'merge sort' algorithm in $\mathcal{O}(T \log T)$ operations [70]. The price ordering of time slots is subsequently broadcast to all EV controllers as shown in Fig. 4.1a. Based on its charging needs $\mathcal{E}_m$, the

Algorithm 5 Decentralized EV scheduling

```
Initialize $\mathbf{e}_m^0 = \mathbf{0}$ for $m = 1, \ldots, M$
for $k = 0, 1, \ldots$ do
    EV scheduler calculates $\mathbf{g}^k$ from (4.3)
    EV scheduler broadcasts $\mathbf{g}^k$ entry ranking to EVs
    Vehicles update $\{\mathbf{e}_m^k\}_{m=1}^M$ via (4.7)–(4.9)
    Profile sums $\sum_{m=1}^M \mathbf{e}_m^{k+1}$ sent to control center
end for
```

m-th EV controller first finds $\mathbf{r}_m^k$ from (4.7)–(4.8) in $\mathcal{O}(T)$. It then updates its charging profile $\mathbf{e}_m^{k+1}$ using (4.9) in $\mathcal{O}(T)$. Note that operations (4.7)–(4.9) can be performed in parallel over the M EV controllers. The updated charging profiles $\{\mathbf{e}_m^{k+1}\}_{m=1}^M$ are communicated back to the charging center, where upon adding the base load $\{d(t)\}$, the center computes the updated cost gradient $\mathbf{g}^{k+1}$, and iterations proceed as tabulated in Algorithm 5. When setting $\eta_k = 2/(k+2)$, the developed solver converges to optimal charging profiles $\{\mathbf{e}_m^*\}$ at the rate [58]

$$C(\{\mathbf{e}_m^k\}) - C(\{\mathbf{e}_m^*\}) \leq \mathcal{O}\left(\frac{1}{k}\right). \tag{4.10}$$

Algorithm 5 not only exhibits provable convergence and low computational cost (namely $\mathcal{O}(T \log T)$ operations) per iteration. It further enjoys two additional advantages. First, the charging center does not require knowing the individual charging profiles $\{\mathbf{e}_m^k\}$, since their summation $\sum_{m=1}^M \mathbf{e}_m^k$ suffices for finding the gradient vector $\mathbf{g}^k$. In an effort to preserve the privacy of EV users, a simple communication protocol can be designed. Information flow can be arranged over a tree graph rooted at the charging center, and vehicle controllers constitute the remaining tree nodes. Each node receives aggregate charging profiles from its downstream nodes, adds them up to its own profile, and forwards the updated aggregate charging profile to its parent node. As a second feature, vehicle controllers do not need to know the precise value of the cost gradient vector $\mathbf{g}^k$, but only the ordering of its entries (current price ordering of time slots). This algorithmic feature lightens the communication load from the charging center to the vehicles, and enhances resiliency to price manipulations and data attacks to the solving scheme.

Algorithm 6 Real-time decentralized EV scheduling

```
Initialize $\mathbf{e}_m^0 = \mathbf{0}$ for all $m \in \mathcal{M}^0$
for $t = 1, 2, \ldots, T$ do
    Scheduler negotiates with EVs in $\mathcal{M}^t$
    for $k = 0, 1, \ldots,$ do
        EV scheduler calculates and broadcasts $\mathbf{g}^k$
        Vehicles update $\{\mathbf{e}_m^k\}_{m \in \mathcal{M}^t}$ via (4.7)–(4.9)
        Profile sums $\sum_{m \in \mathcal{M}^t} \mathbf{e}_m^k$ sent to control center
    end for
    Energy needs $\{R_m^t\}_{m \in \mathcal{M}^t}$ are updated from (4.12)
end for
```

4.2.1 Comparison with Previous Work

The optimal EV charging of (4.2) has been previously studied in [41], where a projected gradient descent (PGD) based solver was developed. Minimizing the non-strongly convex cost in (4.2), the PGD method exhibits a convergence rate of $\mathcal{O}(\frac{1}{k})$ [10]. At iteration k of the PGD method, controller m solves in parallel

$$\mathbf{e}_m^{k+1} := \arg \min_{\mathbf{e}_m \in \mathcal{E}_m} \|\mathbf{e}_m - (\mathbf{e}_m^k - \eta_k' \mathbf{g}^k)\|_2^2 \tag{4.11}$$

for a step size $\eta_k' > 0$. In other words, every EV controller projects vector $(\mathbf{e}_m^k - \eta_k' \mathbf{g}^k)$ onto the simplex $\mathcal{E}_m$, which is a non-trivial task.

On the other hand, each iteration of Algorithm 5 involves closed-form updates, offering high computational efficiency and posing affordable hardware requirements on EV controllers. Although both Algorithm 5 and the PGD solver are decentralized schemes with convergence rate $\mathcal{O}(\frac{1}{k})$, the overall computation time for the former is significantly lower due to its simpler per-iteration updates: The numerical tests in Section 4.4.1 demonstrate that Algorithm 5 provides a 100 times speed-up advantage over the PGD solver and the centralized solver SeDuMi. The SeDuMi solver would be a viable option for tackling (4.2) in a centralized manner after collecting all charging needs $\{(R_m, \mathcal{E}_m)\}$ at the charging center.

4.2.2 Real-time Scheduling

Algorithm 5 requires all EVs to negotiate with the scheduler at the beginning of the control horizon. This presumption can be satisfied for charging coordination in a residential area where

EVs are primarily used for commuting. Nevertheless, there are cases where EVs arrive randomly at a charging station. Then, not all charging needs are known at the beginning of the control horizon; instead, they are revealed to the scheduler in an online manner. To address random EV arrivals, a real-time implementation of Algorithm 5 is pursued next.

At time t, the scheduler negotiates only with the plugged-in EVs comprising the set $\mathcal{M}^t$, while current EV energy demands are denoted by $\{R_m^t\}_{m\in\mathcal{M}^t}$. Ignoring future vehicle arrivals and their energy needs, charging of the EVs in the set $\mathcal{M}^t$ from time t till the latest departure time is coordinated by Algorithm 5. After running Algorithm 5, the vehicles in $\mathcal{M}^t$ are charged by $\{e_m(t)\}_{m\in\mathcal{M}^t}$ during the current time slot t, and their charging demands are updated as

$$R_m^{t+1} = R_m^t - e_m(t) \text{ for all } m \in \mathcal{M}^t. \tag{4.12}$$

The scheduler proceeds to time $t+1$, newly arrived vehicles are taken into account, and the process is repeated.

The overall real-time vehicle scheduling scheme is summarized as Algorithm 6. Algorithm 6 essentially runs Algorithm 5 only for the EVs plugged-in at every time slot. Even though no optimality can be guaranteed for Algorithm 6, this scheme aims greedily for the best instantaneous charging solution while always maintaining feasibility. A related algorithm based on the PGD method is devised in [41].

4.3 Decentralized Charging with Lost Updates

At each iteration, Algorithm 5 requires all vehicles to update their charging profiles according to the current control signal. In practical charging scenarios, vehicle controllers may not be able to update their charging profiles synchronously. That could be the result of failures in the communication links between the aggregator and the vehicle controllers, or due to processing delays in vehicle controllers. In such scenarios, Algorithm 5 has to be modified to guarantee convergence. Assuming the lost updates occur independently at random across iterates and vehicles, the asynchronous charging problem can be handled using our new randomized block Frank-Wolfe (RB-FW) algorithm presented next.

4.3.1 The Randomized Block Frank-Wolfe Algorithm

Our RB-FW method aims at solving the generic problem

$$\begin{aligned} \mathbf{x}^* := \arg\min_{\mathbf{x}\in\mathbb{R}^d} \quad & f(\mathbf{x}) \\ \text{s. to} \quad & \mathbf{x}_1 \in \mathcal{X}_1, \, \ldots, \, \mathbf{x}_{N_b} \in \mathcal{X}_{N_b} \end{aligned} \tag{4.13}$$

where f is a convex and differentiable function, $\mathbf{x}^*$ denotes one globally optimal solution of (4.13), and $\mathbf{x}^\top := [\mathbf{x}_1^\top, \mathbf{x}_2^\top, \cdots, \mathbf{x}_{N_b}^\top]$ with respective feasible sets $\{\mathcal{X}_n\}_{n=1}^{N_b}$ assumed convex and compact. Note that if $N_b = 1$, then (4.13) boils down to (3.11). Thanks to the separable structure of the feasible set $\mathcal{X} := \mathcal{X}_1 \times \ldots \times \mathcal{X}_{N_b}$ of (4.13), the linear oracle in (3.12a) decouples across N_b blocks as

$$\mathbf{s}_n^k = \arg\min_{\mathbf{s}_n\in\mathcal{X}_n} \langle \mathbf{s}_n, \nabla_{\mathbf{x}_n} f(\mathbf{x}^k)\rangle, \quad n = 1, 2, \ldots, N_b \tag{4.14}$$

where $\nabla_{\mathbf{x}_n} f(\mathbf{x}^k)$ comprises the partial derivatives of $f(\mathbf{x})$ with respect to the entries of $\mathbf{x}_n$.

Instead of updating all the N_b blocks in (4.13), RB-FW reduces complexity by updating just B of them, where $B \in \{1, \ldots, N_b\}$ is a pre-selected constant. Let $\mathcal{N}_b := \{1, \ldots, N_b\}$ be the index set of all blocks, and let $\mathcal{B}_k$ be chosen at iteration k uniformly at random among all subsets of $\mathcal{N}_b$ with B elements. The RB-FW solver of (4.13) is summarized as Algorithm 7. In the case of asynchronous vehicle charging, the constant B is determined by the probability of a successful charging profile update, whereas $\mathcal{B}_k$ corresponds to the set of vehicles that successfully updates their charging profiles at iteration k. The only step-size sequence for RB-FW available in the literature is provided in [113], yet the sequence therein does not guarantee feasibility of RB-FW iterates as shown in [135].

To bypass this limitation, we have developed two class of step sizes for RB-FW that offer: (i) guaranteed convergence and feasibility of the iterates along with (ii) flexibility to select a step-size sequence whose decay rate is attuned to the problem at hand. To start with, consider the following general family of diminishing step-size sequences for fixed $\delta \in (0, \alpha]$ with $\alpha := B/N_b$ being the fraction of updated blocks, and decay rate $\rho \in (0.5, 1]$:

$$\boxed{\gamma_k = \frac{2}{\delta k^\rho + 2}, \quad \forall k \geq 0.} \tag{4.15}$$

Algorithm 7 Randomized Block Frank-Wolfe

```
Initialize $k = 0$, $\mathbf{x}^0 \in \mathcal{X}$
while stopping_criterion not met do
    Randomly pick $\mathcal{B}_k \subseteq \mathcal{N}_b$ such that $|\mathcal{B}_k| = B$
    Compute $\mathbf{s}_n^k = \arg\min_{\mathbf{s}_n \in \mathcal{X}_n} \; \mathbf{s}_n^\top \nabla_{\mathbf{x}_n} f(\mathbf{x}^k), \; \forall n \in \mathcal{B}_k$
    Update
        $\mathbf{x}_n^{k+1} = \begin{cases} (1-\gamma_k)\mathbf{x}_n^k + \gamma_k \mathbf{s}_n^k, & \forall n \in \mathcal{B}_k \\ \mathbf{x}_n^k, & \forall n \in \mathcal{N}_b \setminus \mathcal{B}_k \end{cases}$
    $k \leftarrow k + 1$
end while
```

Clearly, the step-size sequence in (4.15) ensures feasibility of the RB-FW iterates since $0 < \gamma_k \leq 1, \; \forall k \geq 0$. To proceed with convergence analysis of RB-FW, two quantities play instrumental roles in the analysis are introduced in order.

The first one is the curvature constant, which for a differentiable $f(\mathbf{x})$ over $\mathcal{X}$ is defined as [21], [58]

$$C_f := \sup_{\substack{\gamma \in [0,1] \\ \mathbf{x}, \mathbf{s} \in \mathcal{X} \\ \mathbf{y} := (1-\gamma)\mathbf{x} + \gamma \mathbf{s}}} \frac{2}{\gamma^2} \left[f(\mathbf{y}) - f(\mathbf{x}) - \langle \mathbf{y} - \mathbf{x}, \nabla f(\mathbf{x}) \rangle \right]. \tag{4.16}$$

C_f is the least upper bound of a scaled difference between $f(\mathbf{y})$ and its linear approximation around $\mathbf{x}$. Throughout, C_f is assumed bounded. This property is closely related to the L-Lipschitz continuity of $\nabla f(\mathbf{x})$ over $\mathcal{X}$, which is defined as

$$\exists L > 0: \;\; \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{s})\| \leq L \|\mathbf{x} - \mathbf{s}\|, \;\; \forall \mathbf{x}, \; \mathbf{s} \in \mathcal{X}. \tag{4.17}$$

If (4.17) holds, it is easy to check that [58, Appendix D]

$$C_f \leq L D_{\mathcal{X}}^2 \tag{4.18}$$

where $D_{\mathcal{X}} := \sup_{\mathbf{x}, \mathbf{s} \in \mathcal{X}} \|\mathbf{x} - \mathbf{s}\|$ is the diameter of $\mathcal{X}$ that is finite for $\mathcal{X}$ compact. Equation (4.18) evidences that C_f is bounded whenever $\nabla f(\mathbf{x})$ is L-Lipschitz continuous over $\mathcal{X}$.

When it comes to RB-FW, the set curvature for an index set $\mathcal{B} \subseteq \mathcal{N}_b$ is commonly used instead of the constant C_f [113]

$$C_f^{\mathcal{B}} := \sup_{\substack{\gamma\in[0,1]\\ \mathbf{x}\in\mathcal{X}\\ \{\mathbf{s}_n\in\mathcal{X}_n\}_{n\in\mathcal{B}}}} \frac{2}{\gamma^2}\Big(f(\mathbf{y}) - f(\mathbf{x}) - \sum_{n\in\mathcal{B}}\langle \mathbf{y}_n - \mathbf{x}_n, \nabla_{\mathbf{x}_n} f(\mathbf{x})\rangle\Big) \tag{4.19}$$

where

$$\mathbf{y}_n := \begin{cases} (1-\gamma)\mathbf{x}_n + \gamma\mathbf{s}_n, & n\in\mathcal{B} \\ \mathbf{x}_n, & n\in\mathcal{N}_b\setminus\mathcal{B} \end{cases}$$

and $\mathbf{y}^\top := [\mathbf{y}_1^\top, \ldots, \mathbf{y}_{N_b}^\top]$. The *expected set curvature* for the uniformly randomly selected $\mathcal{B}$ can thus be expressed as

$$\bar{C}_f^B := \mathbb{E}_{\mathcal{B}}\left[C_f^{\mathcal{B}}\right] = \binom{N_b}{B}^{-1} \sum_{\{\mathcal{B}\colon\, \mathcal{B}\subseteq\mathcal{N}_b,\, |\mathcal{B}|=B\}} C_f^{\mathcal{B}} \tag{4.20}$$

with $\binom{N_b}{B} := N_b!/\big(B!(N_b-B)!\big)$. It is easy to verify that $\bar{C}_f^B \leq C_f$ by observing from (4.16) and (4.19) that $C_f^{\mathcal{B}} \leq C_f, \forall\mathcal{B}\subseteq\mathcal{N}_b$. Note however that $\bar{C}_f^B = C_f^{\mathcal{B}} = C_f$, when $B = N_b$.

The second quantity of interest is the so-termed *duality gap*

$$g(\mathbf{x}) := \sup_{\mathbf{s}\in\mathcal{X}} \ (\mathbf{x}-\mathbf{s})^\top\nabla f(\mathbf{x}), \quad \mathbf{x}\in\mathcal{X} \tag{4.21}$$

whose name stems from Fenchel duality; see [71, Appendix D], [58, Section 2]. Clearly, for the constrained problem (4.13), $\mathbf{x}$ is a stationary point if and only if $g(\mathbf{x}) = 0$. In addition, it holds that $g(\mathbf{x}) \geq h(\mathbf{x})$ [71], where $h(\mathbf{x}) := f(\mathbf{x}) - f(\mathbf{x}^*)$ represents the so-termed primal suboptimality of point $\mathbf{x}$. Thus, $g(\mathbf{x}^k)$ can be used to assess how close is $\mathbf{x}^k$ from being a solution [71], [113]. The ensuing theorem characterizes convergence rate of RB-FW in terms of the primal suboptimality $h(\mathbf{x}^k)$, as well as the minimal expected duality gap until iteration k, defined as [71], [113]

$$g_k := \min_{\kappa\in\{0,1,\ldots k\}} \mathbb{E}\left[g(\mathbf{x}^\kappa)\right]. \tag{4.22}$$

Theorem 1. *For convex $f(\mathbf{x})$, the iterates $\{\mathbf{x}^k\}_{k=1,2,\ldots}$ of Algorithm 7 with step size* (4.15) *satisfy*

$$\mathbb{E}\big[h(\mathbf{x}^k)\big] \leq \frac{4\,(1-\alpha)\,h(\mathbf{x}^0)}{[\delta(k-1)^\rho+2]^2} + \frac{2k\bar{C}_f^B}{[\delta(k-1)^\rho+2]^2} \tag{4.23}$$

and

$$g_k \le \frac{(2\rho+1)^{2\rho+1}(\delta k^\rho + 2)}{\alpha\delta^2(2\rho)^{2\rho}} \cdot \frac{(k+1)\bar{C}_f^B + 2(1-\alpha)h(\mathbf{x}^0)}{k^{2\rho+1}}. \tag{4.24}$$

Proof. See part 4.5.1 of the Appendix. □

Theorem 1 subsumes existing convergence results as special cases. Indeed, when $B = N_b$, one has that $\mathcal{B}_k = \mathcal{N}_b \; \forall k$, which implies that $\bar{C}_f^B = C_f$, and Algorithm 7 reduces to the traditional Frank-Wolfe solver. By selecting $\delta = 1$ and $\rho = 1$, the classical step size $2/(2+k)$ for the Frank-Wolfe method is retrieved. From Theorem 1, the resulting computational bounds are

$$h(\mathbf{x}^k) \le \frac{2kC_f}{(k+1)^2} \le \frac{2C_f}{k+2} \tag{4.25}$$

and

$$g_k \le \frac{27C_f}{4} \cdot \frac{(k+1)(k+2)}{k^3}. \tag{4.26}$$

The resulting convergence rate of $h(\mathbf{x}^k)$ coincides with the one in [58, Theorem 1]. As for g_k, the bound in (4.26) is of the same order as that in [58, Theorem 2].

The decreasing rates of the bounds in Theorem 1 are determined by the decay rates of the step size sequence. The faster γ_k diminishes, the more rapidly the upper bound in Theorem 1 vanishes. However, the sequence in (4.15) decreases at most as fast as $2/(\alpha k + 2)$. To improve the bounds in Theorem 1, a more rapidly vanishing sequence is proposed next. Specifically, consider the sequence

$$\boxed{\gamma_0 = 1, \text{ and } \gamma_{k+1} = \frac{\sqrt{\alpha^2\gamma_k^4 + 4\gamma_k^2} - \alpha\gamma_k^2}{2}, \quad \forall k \ge 0.} \tag{4.27}$$

It is then possible to establish the following proposition.

Proposition 1 (Recursive step size)**.** *If* $\{\gamma_k\}_{k=0,1,\ldots}$ *is chosen as in* (4.27)*, it then holds that*

$$\frac{1}{\alpha k + 1} \le \gamma_k \le \frac{2}{\alpha k + 2}, \quad \forall k \ge 0 \tag{4.28a}$$

$$\gamma_{k+1} \le \gamma_k, \quad \forall k \ge 0. \tag{4.28b}$$

Proof. See part 4.5.2 of the Appendix. □

The upper bound in (4.28a) confirms that the step size in (4.27) vanishes at least as fast as $2/(\alpha k+2)$. Appealing to (4.28a), it holds that $0 \leq \gamma_k \leq 1\, \forall k$, thus ensuring feasibility of the RB-FW iterates. The following computational bounds for RB-FW with γ_k defined in (4.27) can be derived.

Theorem 2. *For convex $f(\mathbf{x})$, the iterates $\{\mathbf{x}^k\}_{k=1,2,\ldots}$ of Algorithm 7 with step size as in* (4.27)*, satisfy*

$$\mathbb{E}\big[h(\mathbf{x}^k)\big] \leq \frac{4(1-\alpha)h(\mathbf{x}^0)}{(\alpha k+2-\alpha)^2} + \frac{2k\bar{C}_f^B}{(\alpha k+2-\alpha)^2} \tag{4.29}$$

and

$$g_k \leq \frac{27(\alpha k+1)}{2\alpha^3 k^3}\left[(k+1)\bar{C}_f^B + 2(1-\alpha)h(\mathbf{x}^0)\right]. \tag{4.30}$$

Proof. See part 4.5.3 of the Appendix. □

To recap, this section put forth two families of step sizes for Algorithm 7 with arbitrary B, namely (4.15) and (4.27). Theorems 1 and 2 establish convergence of Algorithm 7 for these step sizes, which also guarantee feasibility of the iterates. When $\{\gamma_k\}_{k=0,1,\ldots}$ is given by (4.15) with $\delta=\alpha$ and $\rho=1$ or when it is defined as in (4.27), the convergence rates of Algorithm 7 are in the order of $\mathcal{O}\left(1/k\right)$, thus matching those of the traditional Frank-Wolfe algorithm.

Remark 1. The step size of RB-FW can also be chosen through line search, which prescribes

$$\gamma_k = \arg\min_{0\leq\gamma\leq 1} f\left((1-\gamma)\mathbf{x}^k + \gamma\hat{\mathbf{s}}^k\right) \tag{4.31}$$

with $\hat{\mathbf{s}}^{k\top} := [\hat{\mathbf{s}}_1^{k\top}, \ldots, \hat{\mathbf{s}}_{N_b}^{k\top}]$ and

$$\hat{\mathbf{s}}_n^k := \begin{cases} \mathbf{s}_n^k, & n \in \mathcal{B}_k \\ \mathbf{x}_n^k, & n \in \mathcal{N}_b \setminus \mathcal{B}_k. \end{cases}$$

Let $\{\check{\mathbf{x}}^k\}_{k=0,1,\ldots}$ be the iterates generated by Algorithm 7 with γ_k given by (4.31). By (4.37) and (4.31), it holds that

$$\mathbb{E}\big[h(\check{\mathbf{x}}^{k+1})\big] \leq \mathbb{E}\big[h(\check{\mathbf{x}}^k)\big] - \alpha\gamma_t\mathbb{E}\big[g(\check{\mathbf{x}}^k)\big] + \gamma_k^2\bar{C}_f^B/2. \tag{4.32}$$

for any predefined step-size sequence $\{\gamma_k \in [0,1]\}$ [71, 113]. Particularly, (4.32) holds for

$\{\gamma_k := 2/(\alpha k + 2)\}_{k=0,1,\ldots}$. It can then be shown that $\{\check{\mathbf{x}}^k\}_{k=0,1,\ldots}$ satisfy for $k \geq 1$

$$\mathbb{E}\big[h(\check{\mathbf{x}}^k)\big] \leq \frac{4(1-\alpha)h(\mathbf{x}^0)}{(\alpha k + 2 - \alpha)^2} + \frac{2k\bar{C}_f^B}{(\alpha k + 2 - \alpha)^2}$$

and

$$\check{g}_k \leq \frac{27(\alpha k + 2)}{4\alpha^3} \cdot \frac{(k+1)\bar{C}_f^B + 2(1-\alpha)h(\mathbf{x}^0)}{k^3}$$

where $\check{g}_k := \min_{\kappa \in \{0,1,\ldots k\}} \mathbb{E}\left[g(\check{\mathbf{x}}^\kappa)\right]$. The proof follows the steps of the one for Theorem 1. The convergence rate of line-search-based Algorithm 7 therefore remains in the order of $\mathcal{O}(1/k)$. Note however that extra computational cost is incurred for finding γ_k via (4.31).

Remark 2. At this point, it is worth discussing the choice of the step size leading to the fastest convergence in a given problem. Even though the bounds in this section suggest that the more rapid decrease of the step sizes, the quicker decrease of $h(\mathbf{x}^k)$, this is not always the case in practice. This is because step sizes with large decay rates become small after the first few iterations, and small step sizes lead to slow changes in $h(\mathbf{x}^k)$. Conversely, small decay rates tend to yield rapidly decreasing $h(\mathbf{x}^k)$ in the first few iterations since the step sizes remain relatively large. Hence, it is difficult to provide universal guidelines since rapidly or slowly diminishing step sizes may be preferred depending on the specific optimization problem at hand. For example, if optimal solutions lie in the interior of the feasible set, rapidly diminishing step sizes can help reduce oscillations around optimal solutions, thus improving the overall convergence rates. On the other hand, if $f(\mathbf{x})$ is monotone on $\mathcal{X}$, the solution lies on the boundary, which means that no oscillatory behavior is produced and hence, slowly diminishing step sizes will be preferable.

4.3.2 Asynchronous Charging via RB-FW

Under the assumption that the lost updates occur independently at random, asynchronous charging coordination is amenable to RB-FW iterations in Algorithm 7. Let $\mathcal{B}_k$ denote the set of vehicles successfully update their charging profiles at iteration k, our RB-FW-based charging scheme is summarized in Algorithm 8. Let $B := \arg\min_{\kappa=0,1,\ldots} |\mathcal{B}_\kappa|$, then $\alpha = B/M$. Using the step size rule in (4.15) with $\delta = \alpha$ and $\rho = 1$ or the step size in (4.27), the updates of Algorithm 8 remain feasible at all times, while the objective value is guaranteed to converge in expectation with rate $\mathcal{O}\left(\frac{1}{k}\right)$. Moreover, by tuning the decay rate ρ of the step size (4.15), improved convergence can be achieved as will be demonstrated through numerical tests

Algorithm 8 Asynchronous charging control

```
Initialize $k = 0$, and $\mathbf{e}_m^0 \in \mathcal{E}_m$ for $m = 1, \ldots, M$
while stopping_criterion not met do
    EV scheduler evaluates $\mathbf{g}^k$ from (4.3)
    Broadcast entry ranking of $\mathbf{g}^k$ to EVs
    Calculate $\{\mathbf{r}_m^k\}_{m \in \mathcal{B}_k}$ via (4.7) and (4.8)
    Update $\{\mathbf{e}_m^{k+1}\}$ via
        $$\mathbf{e}_m^{k+1} = \begin{cases} (1-\gamma_k)\mathbf{e}_m^k + \gamma_k \mathbf{r}_m^k, & \forall m \in \mathcal{B}_k \\ \mathbf{e}_m^k, & \forall m \in \mathcal{M} \setminus \mathcal{B}_k \end{cases}$$
    Profile sums $\sum_{m=1}^M \mathbf{e}_m^{k+1}$ sent to control center.
    $k \leftarrow k + 1$
end while
```

in Section 4.4.2.

4.4 Numerical Tests

The efficacy of the devised charging schemes were evaluated in this section. The battery capacity of all vehicles was assumed to be 20 kWh. The maximum charging power was fixed to 3.45 kW. According to actual travel survey data [105], the probability density function (pdf) for EV plug-in times in hours is

$$f_{\text{in}}(\tau) = \begin{cases} \mathcal{N}(\mu_{\text{in}} - 24, \sigma_{\text{in}}), & 0 < \tau \leq \mu_{\text{in}} - 12 \\ \mathcal{N}(\mu_{\text{in}}, \sigma_{\text{in}}), & \mu_{\text{in}} - 12 < \tau \leq 24 \end{cases} \tag{4.33}$$

where $\mu_{\text{in}} = 17.47$, $\sigma_{\text{in}} = 3.41$, and $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian pdf with mean μ and variance σ^2. The pdf for plug-out times in hours is

$$f_{\text{out}}(\tau)= \begin{cases} \mathcal{N}(\mu_{\text{out}}, \sigma_{\text{out}}), & 0 < \tau \leq \mu_{\text{out}} + 12 \\ \mathcal{N}(\mu_{\text{out}} + 24, \sigma_{\text{out}}), & \mu_{\text{out}} + 12 < \tau \leq 24 \end{cases} \tag{4.34}$$

where $\mu_{\text{out}} = 8.92$ and $\sigma_{\text{out}} = 3.24$. Moreover, daily travel miles are distributed according to

$$f_{\text{miles}}(y) = \frac{1}{\sqrt{2\pi}\sigma_{\text{miles}} y} \exp\left(-\frac{(\log(y) - \mu_m)^2}{2\sigma_m^2}\right) \tag{4.35}$$

where $\mu_{\text{miles}} = 2.98$ and $\sigma_{\text{miles}} = 1.14$. For each vehicle, the expected state of charge (SOC) was set to 90%. The energy needed per 100 km was $E_{100} = 15$ kWh, and the initial SOC was obtained as $s_m^{\text{soc}} = 0.9 - D_m^{\text{miles}} E_{100}/(100 E_m^B)$, where D_m^{miles} denotes daily travel miles for vehicle m, and E_m^B is the battery capacity of vehicle m. Normalized base load curves with base unit 1000 kW were obtained by averaging the 2014 residential load data from Southern California Edison [98]. The simulation horizon, set from 12:00 pm to 12:00 pm the next day, comprised $T = 96$ time slots. Numerical tests were run using Matlab on an Intel CPU @ 3.6 GHz (32 GB RAM) computer.

4.4.1 Frank-Wolfe for Synchronous Charging

The first experiment simulates Algorithm 5 for managing 59 EVs. The minimizer of (5.45) was obtained via SeDuMi, Algorithm 5, and the PGD solver of [41]. The subproblem (4.11) in PGD was solved by SeDuMi. Algorithm 5 and PGD were run sequentially and they were terminated once the relative cost error became smaller than 10^{-6}. Figure 4.2 shows that the three resultant load curves coincide and feature a flat load valley. Algorithm 5 converged within 0.01 sec, PGD in 12.5 sec, and SeDuMi in 82.47 sec. Problem (4.2) was also solved by the ADMM-based scheme of [87] that converged in 13 sec. Similar to PGD, each ADMM update of the latter involves a quadratic program per vehicle.

Figure 4.3 depicts the cost convergence curves for Algorithm 5 and PGD while scheduling 59 vehicles over $T = 96$ time slots. Figure 4.4 presents the mean running times (averaged across vehicles and iterations) of a single update of Algorithm 5 and PGD and for varying T. Although Algorithm 5 attains a relative cost function error of 10^{-6} at roughly the same rate as PGD, its mean update time is in the order of microseconds, whereas PGD's mean time is in the order of seconds and increases linearly with T. The major computational advantage of Algorithm 5 is the simple update in (4.8).

The real-time Algorithm 6 was subsequently evaluated. By simulating the same 59 EVs used in our first experiment, Algorithm 6 exhibited a load curve almost identical to the one obtained by the offline Algorithm 5 in Fig. 4.2. To amplify the effects of random charging requests, the number of simulated EVs was increased to 120. According to the total load curves shown in Fig. 4.5, the two charging protocols differ slightly around 23:00, thus verifying the efficiency of Algorithm 6. Similar observations have also been made in [41] for uniformly distributed plug-in times. Compared to the PGD protocol of [41], the computational advantage of Algorithm 6

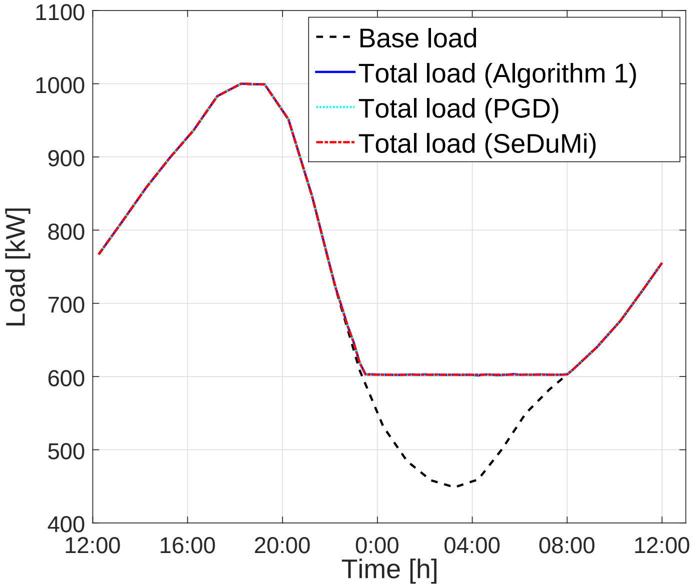


Figure 4.2: Load curves after optimal charging of 59 EVs.

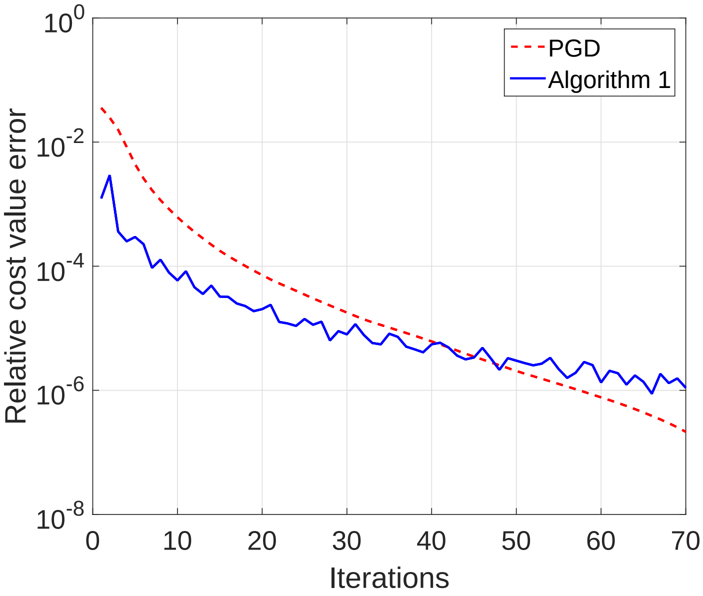


Figure 4.3: Cost convergence for Algorithm 5 and PGD.

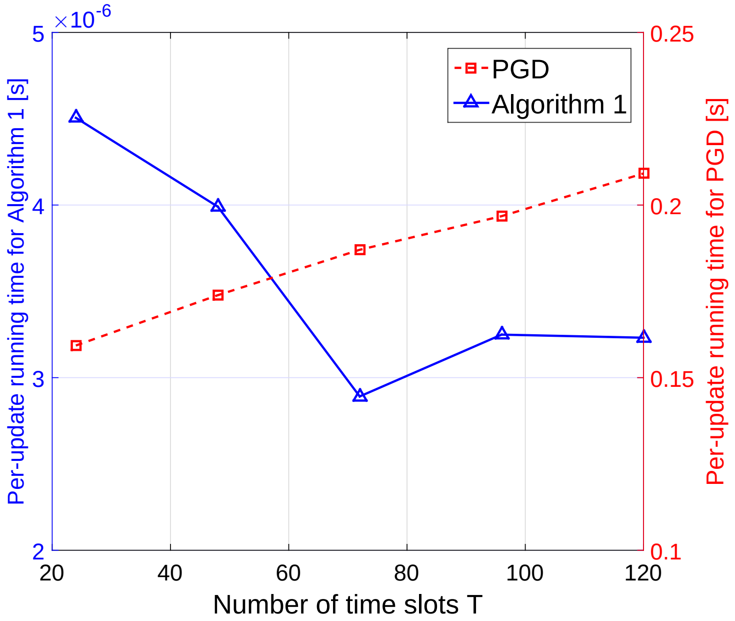


Figure 4.4: Average update time for Algorithm 5 and PGD.

enables faster real-time EV scheduling.

4.4.2 RB-FW for Asynchronous Charging

In this section, asynchronous charging of 63 EVs was simulated. Convergence is assessed in terms of the relative error $\epsilon(\{\mathbf{e}_m^k\}) := \left(f(\{\mathbf{e}_m^k\}) - f(\{\mathbf{e}_m^*\})\right)/f(\{\mathbf{e}_m^*\})$, where $\{\mathbf{e}_m^*\}$ is obtained using the off-the-shelf solver SeDuMi. The following step-size sequences with different decay rates were compared.

$$\begin{aligned}
&(\mathrm{S1}): \quad \gamma_k := \frac{2}{\alpha k + 2} \qquad (4.36)\\
&(\mathrm{S2}): \quad \gamma_k := \frac{\sqrt{\alpha^2\gamma_{k-1}^4 + 4\gamma_{k-1}^2} - \alpha\gamma_{k-1}^2}{2}, \ \gamma_0 = 1\\
&(\mathrm{S3}): \quad \gamma_k := \frac{2}{0.5\alpha k + 2}\\
&(\mathrm{S4}): \quad \gamma_k := \frac{2}{0.5\alpha k^{0.9} + 2}\\
&(\mathrm{S5}): \quad \gamma_k := \frac{2}{0.5\alpha k^{0.8} + 2}.
\end{aligned}$$

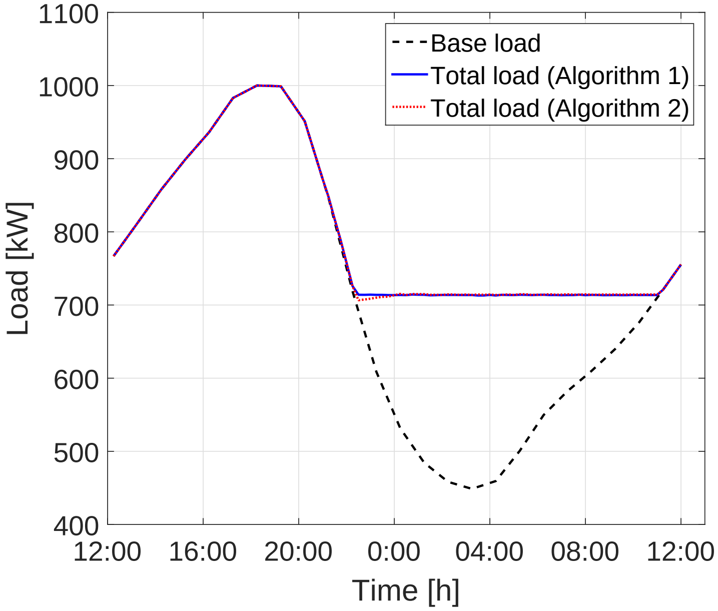


Figure 4.5: Load curves after optimal charging of 120 EVs. The performance degradation using the Algorithm 6 is very small.

S2 is the sequence in (4.27), whereas S1 and S3-S5 are special cases of (4.15). Sequences S1-S5 cover a wide range of decay rates. S2 vanishes faster than S1 [cf. (4.28a)], whereas the decay rates of S3-S5 are smaller than that of S1.

The next experiment assumed that only one vehicle was randomly selected to update its charging profile per iteration. Algorithm 8 with $B = 1$ was run with the step sizes S1-S5 for 1,000 iterations. Fig. 4.6 depicts the evolution of $\epsilon(\{\mathbf{e}_m^k\})$ across the iteration index k for Algorithm 8 with step sizes S1-S5 when $B = 1$. It is observed that the algorithm converges towards a global minimum for all the tested step sizes. In this scenario, the more slowly the step size diminishes, the faster the relative error decreases. Since $B = 1$ and Algorithm 8 is a special instance of Algorithm 7, Fig. 4.6 therefore highlights how randomized single-block FW [71] can benefit from the proposed step sizes. Specifically, the proposed step sizes S3-S5 lead to a much faster convergence than S1.

To assess the effect of increased block size, Algorithm 8 with $B = 10$ was tested. Fig. 4.7 confirms that slowly diminishing step sizes lead to fast convergence in the first few iterations. However, as the iterates approach a minimum, the slowly diminishing step sizes yield larger

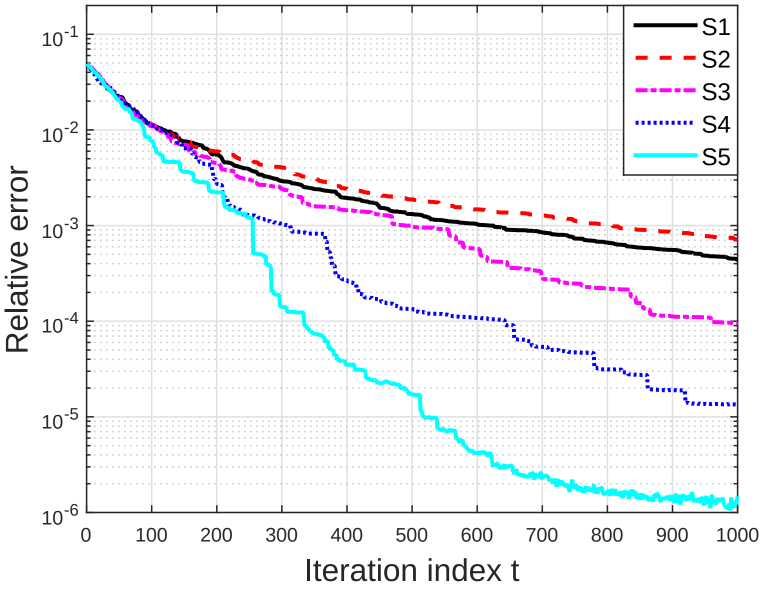


Figure 4.6: Convergence performance of Algorithm 8 with $B = 1$.

oscillations; see e.g. S5 in Fig. 4.7. This phenomenon has already been described in Remark 2. Comparing Figs. 4.6 and 4.7 reveals that considerably less iterations are required to achieve a target accuracy for larger B. For example, about one fifth of iterations are now required for Algorithm 8 with S5 to reach $\epsilon(\{\mathbf{e}_m\}) \leq 10^{-5}$.

The final experiment highlights the impact of B on the convergence of Algorithm 8. Five copies of Algorithm 8, each one with a different step size S1-S5, are executed for 100 independent trials. The minimum value of k such that at least one of these copies satisfies $\epsilon(\{\mathbf{e}_m^k\}) \leq 10^{-5}$ is recorded. Fig. 4.8 represents the sample mean and standard derivation of this minimum k averaged over the 100 trials for different values of B. It is observed that both mean and standard derivation decrease for increasing B. Observe also that the influence of B on the number of iterations decreases for large B. Fig. 4.9 depicts the fraction of trials that each copy of Algorithm 8 is the first among the five copies in achieving $\epsilon(\{\mathbf{e}_m^k\}) \leq 10^{-5}$. This figure reveals that the proposed slowly diminishing step sizes are preferable for small values of B in asynchronous vehicle charging.

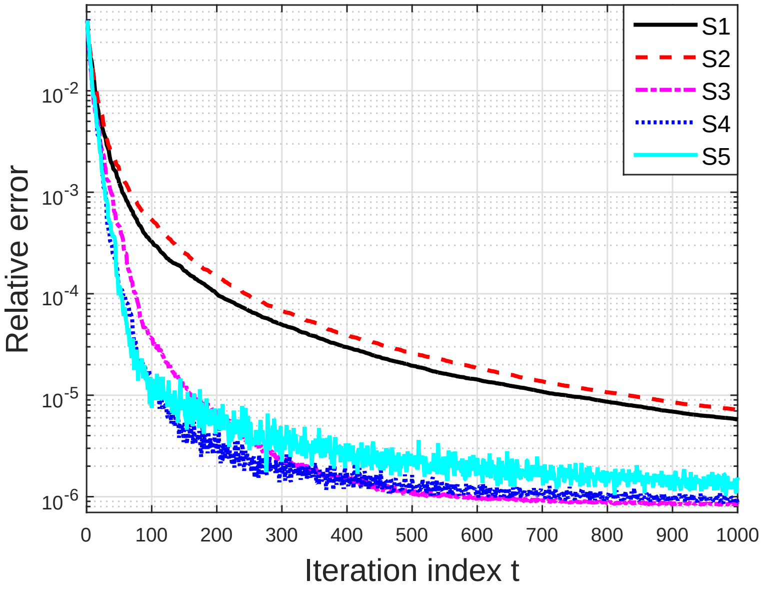


Figure 4.7: Convergence performance of Algorithm 8 with $B = 10$.

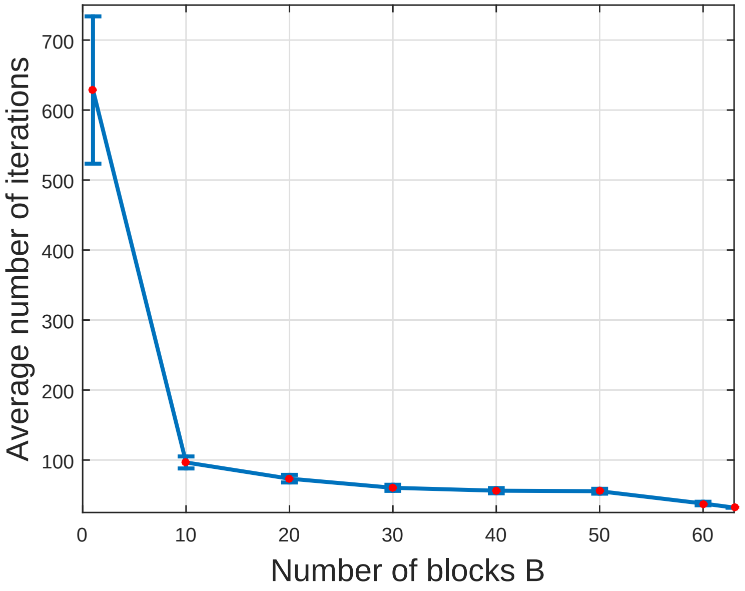


Figure 4.8: Number of iterations to achieve $\epsilon(\mathbf{p}^t) \leq 10^{-5}$.

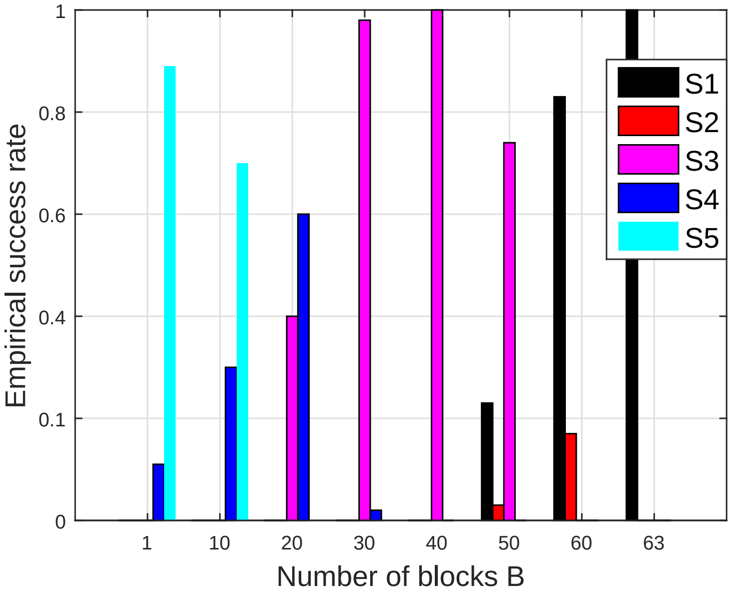


Figure 4.9: Empirical success rate for S1-S5 with different values of B.

4.5 Appendix

Before establishing the proofs for Theorems 1 and 2, the following lemma which quantifies the improvement of $h(\mathbf{x}^k)$ per iteration is adapted from [71, 113].

Lemma 1. *If $\{\mathbf{x}^k\}_{k=0,1,\dots}$ is generated by Algorithm 7 with an arbitrary predefined step-size sequence $\{\gamma_k\}_{k=0,1,\dots}$ satisfying $\gamma_k \in [0,1]\ \forall k$, then it holds that*

$$\mathbb{E}\big[h(\mathbf{x}^{k+1})\big] \le \mathbb{E}\big[h(\mathbf{x}^k)\big] - \alpha\gamma_k \mathbb{E}\big[g(\mathbf{x}^k)\big] + \gamma_k^2 \bar{C}_f^B/2 \tag{4.37}$$

for $k \geq 0$, where the expectation is taken over $\{\mathcal{B}_\kappa\}_{\kappa=0}^k$.

Two lemmas that are useful in establishing the primal convergence as well as the primal-dual convergence of RB-FW for a broad class of step-size sequences are in order.

Lemma 2 (Primal convergence)**.** *If $f(\mathbf{x})$ is convex and $\{\gamma_k\}_{k=0,1,\dots}$ satisfies the following two inequalities:*

$$0 < \gamma_k \le 1, \quad \forall k \ge 0 \tag{4.38a}$$

$$\frac{1-\alpha\gamma_{k+1}}{\gamma_{k+1}^2} \leq \frac{1}{\gamma_k^2}, \quad \forall k \geq 0. \tag{4.38b}$$

the iterates of Algorithm 7 satisfy

$$\mathbb{E}\big[h(\mathbf{x}^k)\big] \leq \frac{1-\alpha\gamma_0}{\gamma_0^2}\gamma_{k-1}^2 h(\mathbf{x}^0) + \frac{k\bar{C}_f^B}{2}\gamma_{k-1}^2, \quad k \geq 1. \tag{4.39}$$

Proof. Since $f(\mathbf{x})$ is differentiable, convexity of $f(\mathbf{x})$ implies that

$$f(\mathbf{x}^k) - f(\mathbf{x}^*) \leq (\mathbf{x}^k - \mathbf{x}^*)^\top \nabla f(\mathbf{x}^k) \tag{4.40}$$

where $\mathbf{x}^*$ denotes any solution to (4.13). Combining (4.21) and (4.40) yields

$$g(\mathbf{x}^k) \geq f(\mathbf{x}^k) - f(\mathbf{x}^*) = h(\mathbf{x}^k) \geq 0. \tag{4.41}$$

Thus, $\mathbb{E}[g(\mathbf{x}^k)] \geq \mathbb{E}[h(\mathbf{x}^k)]$ and (4.37) can be rewritten as

$$\mathbb{E}\big[h(\mathbf{x}^{k+1})\big] \leq (1-\alpha\gamma_k)\mathbb{E}\big[h(\mathbf{x}^k)\big] + \gamma_k^2\bar{C}_f^B/2. \tag{4.42}$$

Dividing both sides of (4.42) by γ_k^2 gives rise to

$$\frac{1}{\gamma_k^2}\mathbb{E}\big[h(\mathbf{x}^{k+1})\big] \leq \frac{1-\alpha\gamma_k}{\gamma_k^2}\mathbb{E}\big[h(\mathbf{x}^k)\big] + \frac{\bar{C}_f^B}{2}. \tag{4.43}$$

Utilizing successively (4.38b) and (4.43) yields

$$\begin{aligned}\frac{1}{\gamma_k^2}\mathbb{E}[h(\mathbf{x}^{k+1})] &\leq \frac{1}{\gamma_{k-1}^2}\mathbb{E}[h(\mathbf{x}^k)] + \frac{1}{2}\bar{C}_f^B \\ &\leq \frac{1-\alpha\gamma_{k-1}}{\gamma_{k-1}^2}\mathbb{E}[h(\mathbf{x}^{k-1})] + \frac{1}{2}\bar{C}_f^B + \frac{1}{2}\bar{C}_f^B \\ &\leq \ldots \leq \frac{1-\alpha\gamma_0}{\gamma_0^2}h(\mathbf{x}^0) + \frac{k+1}{2}\bar{C}_f^B \end{aligned} \tag{4.44}$$

where the last inequality uses $\mathbb{E}[h(\mathbf{x}^0)] = h(\mathbf{x}^0)$. Therefore,

$$\mathbb{E}\big[h(\mathbf{x}^{k+1})\big] \leq \frac{1-\alpha\gamma_0}{\gamma_0^2}\gamma_k^2 h(\mathbf{x}^0) + \frac{k+1}{2}\gamma_k^2\bar{C}_f^B$$

which establishes (4.39). □

Lemma 3 (Primal-dual convergence)**.** *Let* $\{\gamma_k\}_{k=0,1,\ldots}$ *satisfy* (4.38) *and* $\gamma_{k+1} \leq \gamma_k$, $\forall k \geq 0$. *If* $f(\mathbf{x})$ *is convex and* $\{\mathbf{x}^k\}_{k=1,2,\ldots}$ *is generated by Algorithm 7, then for all* $K \in \{1, \ldots, k\}$ *it holds that*

$$g_k \leq \frac{\mathbb{E}\left[h(\mathbf{x}^K)\right]}{\alpha(k-K+1)\gamma_k} + \frac{\bar{C}_f^B \gamma_K^2}{2\alpha\gamma_k} \ . \tag{4.45}$$

Proof. Lemma 1 asserts that

$$\alpha\gamma_k \mathbb{E}[g(\mathbf{x}^k)] \leq \mathbb{E}\big[h(\mathbf{x}^k)\big] - \mathbb{E}\big[h(\mathbf{x}^{k+1})\big] + \gamma_k^2 \bar{C}_f^B/2. \tag{4.46}$$

From $g_k \leq \mathbb{E}[g(\mathbf{x}^k)]$ and (4.46), it follows that

$$\begin{aligned}\alpha g_k \sum_{\kappa=K}^{k} \gamma_\kappa \leq& \alpha \sum_{\kappa=K}^{k} \gamma_\kappa \mathbb{E}[g(\mathbf{x}^\kappa)] \\ \leq& \sum_{\kappa=K}^{k} \left(\mathbb{E}\left[h(\mathbf{x}^\kappa)\right] - \mathbb{E}\left[h(\mathbf{x}^{\kappa+1})\right]\right) + \frac{\bar{C}_f^B}{2} \sum_{\kappa=K}^{k} \gamma_\kappa^2 \\ =& \mathbb{E}\left[h(\mathbf{x}^K)\right] - \mathbb{E}\big[h(\mathbf{x}^{k+1})\big] + \frac{\bar{C}_f^B}{2} \sum_{\kappa=K}^{k} \gamma_\kappa^2 \\ \leq& \mathbb{E}\left[h(\mathbf{x}^K)\right] + \frac{\bar{C}_f^B}{2}(k-K+1)\gamma_K^2 \end{aligned} \tag{4.47}$$

where the last inequality follows from $\mathbb{E}[h(\mathbf{x}^{k+1})] \geq 0$ and $\gamma_{k+1} \leq \gamma_k$. On the other hand, since $\gamma_\kappa \geq \gamma_k$, $\forall \kappa \leq k$, one arrives at

$$\sum_{\kappa=K}^{k} \gamma_\kappa \geq (k-K+1)\gamma_k. \tag{4.48}$$

Finally, (4.45) follows after combining (4.47) with (4.48), and dividing both sides of the resulting inequality by $\alpha(k-K+1)\gamma_k$. □

Upon establishing Lemmas 2 and 3, it is handy to prove Theorem 1 as follows.

4.5.1 Proof of Theorem 1

In order to apply Lemmas 2 and 3, we need to show that the sequence $\{\gamma_k\}$ in (4.15) satisfies (4.38). Obviously, $\{\gamma_k\}$ satisfies (4.38a). To show $\{\gamma_k\}$ satisfies (4.38b), plug (4.15) into the left-hand side of (4.38b), yielding

$$\begin{aligned}\frac{1-\alpha\gamma_{k+1}}{\gamma_{k+1}^2} &= \frac{[\delta(k+1)^\rho+2-\alpha]^2-\alpha^2}{4}\\ &\leq \frac{[\delta(k+1)^\rho+2-\alpha]^2}{4} \leq \frac{[\delta(k+1)^\rho-\delta+2]^2}{4}\end{aligned} \tag{4.49}$$

where the last inequality follows from $\delta \leq \alpha \leq 1$. Consider the auxiliary function $\varphi(x) := (x+c)^\rho - x^\rho - c,\ x \geq 0$ for some constant $c \geq 1$, and its first-order derivative

$$\varphi'(x) = \rho(x+1)^{\rho-1} - \rho x^{\rho-1}, \quad x \geq 0.$$

Since $\rho \leq 1$, it holds that $\varphi'(x) \leq 0$, and thus,

$$\varphi(x) \leq \varphi(0) = c^\rho - c \leq 0, \quad \forall x \geq 0$$

or,

$$(x+c)^\rho - c \leq x^\rho, \quad \forall x \geq 0. \tag{4.50}$$

Multiplying both sides of (4.50) by δ, and setting $c = 1$ and $x = k$ gives rise to

$$0 \leq \delta(k+1)^\rho - \delta \leq \delta k^\rho, \quad \forall k \geq 0. \tag{4.51}$$

Combining (4.49) and (4.51) yields

$$\frac{1-\alpha\gamma_{k+1}}{\gamma_{k+1}^2} \leq \frac{[\delta k^\rho+2]^2}{4} = \frac{1}{\gamma_k^2}$$

which confirms that the sequence $\{\gamma_k\}$ defined in (4.15) satisfies (4.38b).

Expression (4.23) then follows directly by substituting (4.15) into (4.39). To show (4.24), apply Lemma 3 to verify that

$$\begin{aligned}\alpha g_k &\leq \frac{\mathbb{E}\left[h(\mathbf{x}^K)\right]}{\gamma_0^2(k-K+1)\gamma_k} + \frac{\bar{C}_f^B\gamma_K^2}{2\gamma_k} \\ &\leq \frac{(1-\alpha\gamma_0)\gamma_{K-1}^2 h(\mathbf{x}^0)}{\gamma_0^2(k-K+1)\gamma_k} + \frac{K\bar{C}_f^B\gamma_{K-1}^2}{2(k-K+1)\gamma_k} + \frac{\bar{C}_f^B\gamma_K^2}{2\gamma_k} \\ &\leq \frac{(1-\alpha)\,\gamma_{K-1}^2 h(\mathbf{x}^0)}{(k-K+1)\gamma_k} + \frac{\gamma_{K-1}^2\bar{C}_f^B(k+1)}{2\gamma_k(k-K+1)} \\ &\leq \frac{\gamma_{K-1}^2}{k-K+1}\cdot\frac{(k+1)\bar{C}_f^B + 2(1-\alpha)h(\mathbf{x}^0)}{2\gamma_k} \end{aligned} \tag{4.52}$$

for all $K \in \{1,\ldots,k\}$, where the second inequality stems from (4.39) and the third one follows from $\gamma_K \leq \gamma_{K-1}$ and $\gamma_0 = 1$. The next step is to bound the first quotient in the right-hand side of (4.52). To this end, set $c = 2/\delta$ and $x = K-1$ in (4.50) to deduce that

$$\gamma_{K-1} = \frac{2}{\delta(K-1)^\rho + 2} \leq \frac{2}{\delta(K-1+2/\delta)^\rho}. \tag{4.53}$$

Now set $K = \lceil \mu(k+2/\delta) \rceil$, where μ is an arbitrary constant. Since $K \in \{1,\ldots,k\}$, μ needs to satisfy

$$0 < \mu \leq \frac{k}{k+2/\delta}. \tag{4.54}$$

Because

$$\lceil \mu(k+2/\delta) \rceil - 1 + 2/\delta \geq \mu(k+2/\delta) - 2 + 2/\delta \geq \mu(k+2/\delta) > 0$$

it follows from (4.53) that

$$\gamma_{K-1} \leq \frac{2}{\delta\mu^\rho(k+2/\delta)^\rho}.$$

Therefore,

$$\begin{aligned}\frac{\gamma_{K-1}^2}{k-K+1} \leq& \frac{4}{\delta^2(k-K+1)\mu^{2\rho}(k+2/\delta)^{2\rho}} \\ \leq& \frac{4}{\delta^2[k-\mu(k+2/\delta)]\mu^{2\rho}(k+2/\delta)^{2\rho}}.\end{aligned} \tag{4.55}$$

Minimizing the right-hand side with respect to μ in the interval (4.54) yields

$$\frac{\gamma_{K-1}^2}{k-K+1} \leq \frac{4(2\rho+1)^{2\rho+1}}{\delta^2(2\rho)^{2\rho}k^{2\rho+1}} \tag{4.56}$$

for $\mu = \frac{2\rho}{2\rho+1}\frac{k}{k+2/\delta}$.

From (4.52), g_k can be upper bounded as

$$\begin{aligned} g_k \leq & \frac{\gamma_{K-1}^2}{k-K+1} \cdot \frac{(k+1)\bar{C}_f^B + 2(1-\alpha)h(\mathbf{x}^0)}{2\alpha\gamma_k} \\ \leq & \frac{(2\rho+1)^{2\rho+1}(\delta k^\rho + 2)}{\alpha\delta^2(2\rho)^{2\rho}} \cdot \frac{(k+1)\bar{C}_f^B + 2(1-\alpha)h(\mathbf{x}^0)}{k^{2\rho+1}}. \end{aligned}$$

where the second inequality follows from (4.56) and (4.15).

4.5.2 Proof of Proposition 1

To prove (4.28a) by induction, it clearly holds for $k = 0$, and assume that it holds also for a fixed $k \geq 0$. Then, one needs to show that

$$\frac{1}{\alpha k + 1 + \alpha} \leq \gamma_{k+1} \leq \frac{2}{\alpha k + 2 + \alpha}. \tag{4.57}$$

To this end, define the auxiliary function

$$\hat{\varphi}(x) := \frac{\sqrt{\alpha^2 x^4 + 4x^2} - \alpha x^2}{2}, \quad x \geq 0$$

which is monotonically increasing since

$$\hat{\varphi}'(x) = \frac{(\alpha^2 x^2 + 2) - \alpha x\sqrt{\alpha^2 x^2 + 4}}{\sqrt{\alpha^2 x^2 + 4}} > 0.$$

Thus, by the induction hypothesis we have

$$\hat{\varphi}\left(\frac{1}{\alpha k + 1}\right) \leq \gamma_{k+1} = \hat{\varphi}(\gamma_k) \leq \hat{\varphi}\left(\frac{2}{\alpha k + 2}\right). \tag{4.58}$$

Note that

$$1 - \frac{2\alpha}{\alpha k + 2 + \alpha} \leq \left(1 - \frac{\alpha}{\alpha k + 2 + \alpha}\right)^2 = \left(\frac{\alpha k + 2}{\alpha k + 2 + \alpha}\right)^2$$

or, equivalently,

$$
\begin{aligned}
1 &\le \frac{2\alpha}{\alpha k + 2 + \alpha} + \left(\frac{\alpha k + 2}{\alpha k + 2 + \alpha}\right)^2 \\
&= \left(\frac{\alpha}{\alpha k + 2} + \frac{\alpha k + 2}{\alpha k + 2 + \alpha}\right)^2 - \frac{\alpha^2}{(\alpha k + 2)^2}. \qquad (4.59)
\end{aligned}
$$

This inequality implies that

$$
\hat{\varphi}\left(\frac{2}{\alpha k + 2}\right) = \frac{1}{\alpha k + 2}\sqrt{\frac{4\alpha^2}{(\alpha k + 2)^2} + 4} - \frac{2\alpha}{(\alpha k + 2)^2} \le \frac{2}{\alpha k + 2 + \alpha}. \qquad (4.60)
$$

Combining (4.60) with the second inequality in (4.58) proves the second inequality in (4.57).

On the other hand, since $(\alpha k + 1 + \alpha)^2 \ge \alpha(\alpha k + 1 + \alpha) + (\alpha k + 1)^2$, it holds that

$$
\begin{aligned}
\frac{\alpha^2}{(\alpha k + 1)^2} + 4 &\ge \frac{\alpha^2}{(\alpha k + 1)^2} + \frac{4\alpha}{\alpha k + 1 + \alpha} + \frac{4(\alpha k + 1)^2}{(\alpha k + 1 + \alpha)^2} \\
&= \left(\frac{\alpha}{\alpha k + 1} + \frac{2(\alpha k + 1)}{\alpha k + 1 + \alpha}\right)^2. \qquad (4.61)
\end{aligned}
$$

Thus,

$$
\hat{\varphi}\left(\frac{1}{\alpha k + 1}\right) = \frac{1}{2(\alpha k + 1)}\sqrt{\frac{\alpha^2}{(\alpha k + 1)^2} + 4} - \frac{\alpha}{2(\alpha k + 1)^2} \ge \frac{1}{\alpha k + 1 + \alpha}. \qquad (4.62)
$$

Combining (4.62) with the first inequality in (4.58) proves the first inequality in (4.57), thus concluding the proof of (4.28a).

To prove (4.28b), one can also proceed by induction. First, $\gamma_1 \le \gamma_0$ since $\frac{\sqrt{\alpha^2+4}-\alpha}{2} \le 1$. Assuming $\gamma_{k-1} \le \gamma_k$, it follows that $\gamma_k \le \gamma_{k+1}$ since $\hat{\varphi}(x)$ is nondecreasing.

4.5.3 Proof of Theorem 2

To check whether (4.27) meets (4.38), note that (4.38a) follows from (4.28a), whereas (4.27) implies that (4.38b) holds with equality. Because (4.27) satisfies (4.38) and (4.28b), the computational bounds in (4.29) and (4.30) for RB-FW with step-size sequence (4.27) can be derived by plugging (4.28a) into Lemmas 2 and 3.

Inequality (4.29) readily follows from (4.39) and (4.28a). To prove (4.30), note that (4.52) holds because of $\gamma_0 = 1$ and (4.28). Meanwhile, by the second inequality in (4.28a), the step size in (4.27) satisfies (4.56) for $q = \alpha$ and $\rho = 1$, that is

$$\frac{\gamma_{K-1}^2}{k - K + 1} \leq \frac{27}{\alpha^2 k^3}. \tag{4.63}$$

Plugging (4.28a) together with (4.63) into (4.52), yields (4.30).

Chapter 5

Scalable Microgrid Energy Management

This chapter begins with the development of approximate single- and multi-phase microgrid models (Section 5.1). Building on these approximate models, local reactive power control rules for maintaining bus voltage magnitudes within desirable levels are devised in Section 5.2. To improve performance of the aforementioned localized protocols, optimal decentralized energy management schemes that require communication only between neighbors are put forth in Section 5.3. Numerical tests on the IEEE 123-bus feeder are presented in Section 5.4.

5.1 Microgrid Modeling

Approximate models for radial single- and multi-phase grids are presented in this section.

5.1.1 Modeling of Single-phase Grids

Distribution grids are typically operated as radial. A single-phase grid with $N+1$ buses can be modeled by a tree graph $\mathcal{G} = (\mathcal{N}_o, \mathcal{L})$ whose nodes $\mathcal{N}_o := \{0, \ldots, N\}$ correspond to buses, and whose edges $\mathcal{L}$ correspond to distribution lines with cardinality $|\mathcal{L}| = N$. The feeder bus is indexed by $n = 0$, whereas every non-feeder bus $n \in \mathcal{N} = \{1, \ldots, N\}$ has a unique parent bus denoted by π_n. Without loss of generality, nodes can be numbered such that $\pi_n < n$ for all $n \in \mathcal{N}$. For every bus $n \in \mathcal{N}_o$, let v_n be its squared voltage magnitude, and $s_n = p_n + jq_n$ its complex power injection.

The distribution line connecting bus n with its parent π_n is indexed by n. For every line $n \in \mathcal{L}$, let $z_n = r_n + jx_n$, ℓ_n, and $S_n = P_n + jQ_n$ be the line impedance, the squared current magnitude, and the complex power flow sent from the sending bus π_n, respectively. If $\mathcal{C}_n$ is the set of children buses for bus n, the grid can be modeled by the branch flow model [5], [6]

$$s_n = \sum_{n' \in \mathcal{C}_n} S_{n'} - S_n + \ell_n z_n \tag{5.1a}$$

$$v_n = v_{\pi_n} - 2\operatorname{Re}[z_n^\star S_n] + \ell_n |z_n|^2 \tag{5.1b}$$

$$|S_n|^2 = v_{\pi_n} \ell_n \tag{5.1c}$$

for all $n \in \mathcal{N}$, and the initial condition $s_0 = \sum_{n' \in \mathcal{C}_0} S_{n'}$.

For notational brevity, collect all nodal quantities related to non-feeder buses in vectors $\mathbf{p} := [p_1 \cdots p_N]^\top$, $\mathbf{q} := [q_1 \cdots q_N]^\top$, and $\mathbf{v} := [v_1 \cdots v_N]^\top$. Similarly for lines, introduce vectors $\mathbf{r} := [r_1 \cdots r_N]^\top$, $\mathbf{x} := [x_1 \cdots x_N]^\top$, $\mathbf{P} := [P_1 \cdots P_N]^\top$, and $\mathbf{Q} := [Q_1 \cdots Q_N]^\top$. Define further the complex vectors $\mathbf{s} := \mathbf{p} + j\mathbf{q}$, $\mathbf{z} := \mathbf{r} + j\mathbf{x}$, and $\mathbf{S} := \mathbf{P} + j\mathbf{Q}$. According to the approximate *LinDistFlow* model, the grid is described by the linear equations [5], [6]

$$\mathbf{s} = \mathbf{A}^\top \mathbf{S} \tag{5.2a}$$

$$\mathbf{A}\mathbf{v} = 2\operatorname{Re}[\mathbf{Z}^\star \mathbf{S}] - \mathbf{a}_0 v_0 \tag{5.2b}$$

where v_0 is the squared voltage magnitude at the feeder; matrix $\mathbf{Z}$ is defined as $\mathbf{Z} := \operatorname{diag}(\mathbf{z})$; and $\mathbf{A}$ is the *reduced branch-bus incidence matrix* enjoying the following properties.

Proposition 2 ([67]). *The negative of the reduced branch-bus incidence matrix $-\mathbf{A}$ and its inverse $\mathbf{F} := -\mathbf{A}^{-1}$ satisfy:*
(p1) they are both lower triangular with unit eigenvalues;
(p2) $\mathbf{F} \geq \mathbf{0}$; and $\mathbf{F}\mathbf{a}_0 = \mathbf{1}_N$
where $\tilde{\mathbf{A}} = [\mathbf{a}_0\ \mathbf{A}]$ is the full branch-bus incidence matrix.

Equation (5.2a) can be now expressed as $\mathbf{S} = -\mathbf{F}^\top \mathbf{s}$. Substituting the latter into (5.2b), and premultiplying by $-\mathbf{F}$ yields $\mathbf{v} = 2\operatorname{Re}\left[\mathbf{F}\mathbf{Z}^\star\mathbf{F}^\top\mathbf{s}\right] + v_0\mathbf{F}\mathbf{a}_0$. Proposition 2 and the properties of the real part operator provide [5], [36]

$$\mathbf{v} = \mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q} + v_0\mathbf{1}_N \tag{5.3}$$

where $\mathbf{R} := 2\mathbf{F}\,\mathrm{diag}(\mathbf{r})\mathbf{F}^\top$ and $\mathbf{X} := 2\mathbf{F}\,\mathrm{diag}(\mathbf{x})\mathbf{F}^\top$; see also [13] for a linear approximation model relating complex voltages to complex injections. It is well understood that in transmission grids and under regular load conditions and high reactance-to-resistance ratios, the nodal voltage magnitudes are approximately independent of active power injections. On the contrary, the approximate model of (5.3) confirms that voltage magnitudes in distribution grids depend significantly not only on reactive but active injections too. The dependence is roughly linear with the following properties.

Remark 3. Although $\mathbf{F}$ is lower triangular, matrices $\mathbf{R}$ and $\mathbf{X}$ are generally full. Hence, local injection deviations affect voltage magnitudes globally.

Remark 4. Assuming $\mathbf{r}$ and $\mathbf{x}$ to be strictly positive, $\mathbf{R}$ and $\mathbf{X}$ are symmetric and strictly positive-definite by definition; cf. [36] for a more elaborate proof. Further, their inverses are expressed as $\mathbf{R}^{-1} = \frac{1}{2}\mathbf{A}^\top\,\mathrm{diag}^{-1}(\mathbf{r})\mathbf{A}$ and $\mathbf{X}^{-1} = \frac{1}{2}\mathbf{A}^\top\,\mathrm{diag}^{-1}(\mathbf{x})\mathbf{A}$.

Remark 5. Since $\mathbf{F} \geq \mathbf{0}$, it follows readily that $\mathbf{R} \geq \mathbf{0}$ and $\mathbf{X} \geq \mathbf{0}$. Hence, injecting more active or reactive power at any bus raises the voltage magnitudes in the entire grid.

5.1.2 Modeling of Multi-phase Grids

The focus shifts next to modeling multi-phase grids. For ease of exposition, it is first assumed that all buses are served by all three phases. For this reason, system variables are now represented by 3-dimensional vectors. Slightly abusing the notation used in Section 5.1.1, the complex voltages and the power injections at all phases of bus n here are denoted by $\tilde{\mathbf{v}}_n$ and $\mathbf{s}_n$, respectively. Similarly, the complex currents and the complex power flows on all phases of line n are represented by $\tilde{\mathbf{i}}_n$ and $\mathbf{S}_n$, respectively. The coupling across phases provides the multivariate version of Ohm's law:

$$\tilde{\mathbf{v}}_n = \tilde{\mathbf{v}}_{\pi_n} - \mathbf{Z}_n\tilde{\mathbf{i}}_n \tag{5.4}$$

where $\mathbf{Z}_n = \mathbf{Z}_n^\top$ is the phase impedance matrix for line n. If $\mathbf{S}_n$ is the flow on line n seen from bus π_n, the flow received at bus n is

$$\operatorname{diag}(\tilde{\mathbf{v}}_n)\tilde{\mathbf{i}}_n^\star = \mathbf{S}_n - \operatorname{diag}(\mathbf{Z}_n\tilde{\mathbf{i}}_n)\tilde{\mathbf{i}}_n^\star.$$

The multi-phase generalization of (5.1a) reads

$$\mathbf{s}_n = \sum_{n' \in \mathcal{C}_n} \mathbf{S}_{n'} - \mathbf{S}_n + \operatorname{diag}(\mathbf{Z}_n\tilde{\mathbf{i}}_n)\tilde{\mathbf{i}}_n^\star \tag{5.5}$$

for all $n \in \mathcal{N}$. As advocated in [40], to obtain the multi-phase equivalent of the voltage drop equation (5.1b), multiply (5.4) by the complex Hermitian of each side, and maintain only the diagonal of the resultant matrix:

$$\begin{aligned}\operatorname{diag}\left(\tilde{\mathbf{v}}_n\tilde{\mathbf{v}}_n^H\right) =& \operatorname{diag}\left(\tilde{\mathbf{v}}_{\pi_n}\tilde{\mathbf{v}}_{\pi_n}^H\right) - 2\operatorname{Re}\left[\operatorname{diag}\left(\tilde{\mathbf{v}}_{\pi_n}\tilde{\mathbf{i}}_n^H\mathbf{Z}_n^H\right)\right]\\ &+ \operatorname{diag}\left(\mathbf{Z}_n\tilde{\mathbf{i}}_n\tilde{\mathbf{i}}_n^H\mathbf{Z}_n^H\right).\end{aligned} \tag{5.6}$$

The full AC model for this multiphase grid is completed by replicating the definition for flows (5.1c) on a per phase basis. Similar to single-phase grids, the model involves computationally inconvenient quadratic equations, but convex relaxations render the model tractable under appropriate conditions [40].

Alternatively, one may resort to the simpler approximate model of [40]. As for single-phase grids, because $\mathbf{Z}_n$'s have relatively small entries, the last summands in the right-hand sides (RHS) of (5.5) and (5.6) can be dropped. Regarding the second summand in the RHS of (5.6), let us further assume that phase voltages are approximately balanced. By surrogating $\tilde{\mathbf{v}}_n$ by $\tilde{v}_n\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} := [1\ \alpha\ \alpha^2]^\top$ and $\alpha = e^{-j\frac{2\pi}{3}}$, the complex current vector $\tilde{\mathbf{i}}_n$ can be roughly expressed as

$$\tilde{\mathbf{i}}_n^\star \approx \frac{1}{\tilde{v}_{\pi_n}} \operatorname{diag}(\mathbf{S}_n)\boldsymbol{\alpha}^\star \tag{5.7}$$

and the outer product $\tilde{\mathbf{v}}_{\pi_n}\tilde{\mathbf{i}}_n^H$ can be thus replaced by $\boldsymbol{\alpha}\boldsymbol{\alpha}^H \operatorname{diag}(\mathbf{S}_n)$. Let $\mathbf{v}_n$ be the vector of per-phase squared voltage magnitudes for all three phases on bus n

$$\mathbf{v}_n = \begin{bmatrix} v_n^a \\ v_n^b \\ v_n^c \end{bmatrix} = \operatorname{diag}(\tilde{\mathbf{v}}_n \tilde{\mathbf{v}}_n^H).$$

Then, equation (5.6) permits the approximation

$$\mathbf{v}_{\pi_n} - \mathbf{v}_n = 2\operatorname{Re}\left[\operatorname{diag}\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^H \operatorname{diag}(\mathbf{S}_n)\mathbf{Z}_n^H\right)\right]$$

where the argument inside the real operator simplifies as

$$\begin{aligned}\operatorname{diag}\left(\boldsymbol{\alpha}\left(\mathbf{Z}_n \operatorname{diag}(\mathbf{S}_n^\star)\boldsymbol{\alpha}\right)^H\right) &= \operatorname{diag}(\boldsymbol{\alpha})\mathbf{Z}_n^\star \operatorname{diag}(\mathbf{S}_n)\boldsymbol{\alpha}^\star \\ &= \operatorname{diag}(\boldsymbol{\alpha})\mathbf{Z}_n^H \operatorname{diag}(\boldsymbol{\alpha}^\star)\mathbf{S}_n.\end{aligned}$$

The equalities follow from the properties of the diag operator: $\operatorname{diag}(\mathbf{x}\mathbf{y}^H) = \operatorname{diag}(\mathbf{x})\mathbf{y}^\star$ and $\operatorname{diag}(\mathbf{x})\mathbf{y} = \operatorname{diag}(\mathbf{y})\mathbf{x}$. The approximate multi-phase model reads for all $n \in \mathcal{N}$

$$\mathbf{s}_n = \sum_{n' \in \mathcal{C}_n} \mathbf{S}_{n'} - \mathbf{S}_n \tag{5.8a}$$

$$\mathbf{v}_{\pi_n} - \mathbf{v}_n = 2\operatorname{Re}\left[\tilde{\mathbf{Z}}_n^\star \mathbf{S}_n\right] \tag{5.8b}$$

$$\tilde{\mathbf{Z}}_n := \operatorname{diag}(\boldsymbol{\alpha}^\star)\mathbf{Z}_n \operatorname{diag}(\boldsymbol{\alpha}). \tag{5.8c}$$

Building on the grid model of [40], we express (5.8) in a matrix-vector form and study the involved matrices to better understand voltage regulation schemes. Heed that system variables can be arranged either per bus or per phase. For example, the squared voltage magnitudes can be stacked as

$$\check{\mathbf{v}} := \begin{bmatrix} \check{\mathbf{v}}_a \\ \check{\mathbf{v}}_b \\ \check{\mathbf{v}}_c \end{bmatrix} \quad \text{or} \quad \mathbf{v} := \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_N \end{bmatrix} \tag{5.9}$$

where $\check{\mathbf{v}}_\phi := [v_1^\phi \ \ldots \ v_N^\phi]^\top$ for $\phi \in \{a, b, c\}$. Likewise, injections can be arranged in $\check{\mathbf{s}}$ or $\mathbf{s}$, and flows in $\check{\mathbf{S}}$ or $\mathbf{S}$. It can be easily verified that the aforementioned vector pairs are related by

$$\mathbf{v} = \mathbf{T}\check{\mathbf{v}},\ \mathbf{s} = \mathbf{T}\check{\mathbf{s}},\ \mathbf{S} = \mathbf{T}\check{\mathbf{S}} \tag{5.10}$$

for a common permutation matrix $\mathbf{T}$ compactly expressed as

$$\mathbf{T} := \begin{bmatrix} \mathbf{I}_3 \otimes \mathbf{e}_1^\top \\ \vdots \\ \mathbf{I}_3 \otimes \mathbf{e}_N^\top \end{bmatrix} \tag{5.11}$$

where $\mathbf{e}_n$ is the n-th column of $\mathbf{I}_N$. Being a permutation matrix, $\mathbf{T}$ satisfies $\mathbf{T}^{-1} = \mathbf{T}^\top$. Algebraic manipulations postponed for the Appendix show that voltage magnitudes in multiphase grids are related to nodal injections as follows:

Proposition 3. *Based on* (5.8)*, it holds that*

$$\mathbf{v} = \mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q} + v_0\mathbf{1}_{3N} \tag{5.12}$$

where the involved matrices are defined as

$$\mathbf{R} := 2\mathbf{M}\,\mathrm{bdiag}(\{\mathrm{Re}[\tilde{\mathbf{Z}}_n]\})\mathbf{M}^\top \tag{5.13a}$$
$$\mathbf{X} := 2\mathbf{M}\,\mathrm{bdiag}(\{\mathrm{Im}[\tilde{\mathbf{Z}}_n]\})\mathbf{M}^\top,\ \textit{and} \tag{5.13b}$$
$$\mathbf{M} := \mathbf{T}(\mathbf{I}_3 \otimes \mathbf{F})\mathbf{T}^\top \tag{5.13c}$$

and matrices $\tilde{\mathbf{Z}}_n$ have been defined in (5.8c)*.*

Proof of Proposition 3. Collecting (5.8b) for all n yields $\mathbf{v}_\pi - \mathbf{v} = 2\,\mathrm{Re}\left[\mathrm{bdiag}(\{\tilde{\mathbf{Z}}_n\})\check{\mathbf{S}}\right]$, or via the permutations of (5.10), as

$$\mathbf{T}\left(\check{\mathbf{v}}_\pi - \check{\mathbf{v}}\right) = 2\,\mathrm{Re}\left[\mathrm{bdiag}(\{\tilde{\mathbf{Z}}_n\})\mathbf{T}\mathbf{S}\right]. \tag{5.14}$$

Focus first on the LHS of (5.14). Observe that voltage drops between adjacent buses can be alternatively expressed as $\check{\mathbf{v}}_{\pi,\phi} - \check{\mathbf{v}}_\phi = \mathbf{A}\check{\mathbf{v}}_\phi + v_0\mathbf{a}_0$ for $\phi \in \{a, b, c\}$. Stacking the latter equations across all phases yields

$$\check{\mathbf{v}}_\pi - \check{\mathbf{v}} = (\mathbf{I}_3 \otimes \mathbf{A})\check{\mathbf{v}} + v_0 \mathbf{1}_3 \otimes \mathbf{a}_0. \tag{5.15}$$

Regarding the RHS of (5.14), recall that flows are decoupled across phases. The grid topology and (5.8a) dictate that $\check{\mathbf{s}}_\phi = \mathbf{A}^\top \check{\mathbf{S}}_\phi$, or equivalently, $\check{\mathbf{S}}_\phi = -\mathbf{F}^\top \check{\mathbf{s}}_\phi$ for all ϕ. Stacking flows across all phases yields

$$\check{\mathbf{S}} = -(\mathbf{I}_3 \otimes \mathbf{F}^\top)\check{\mathbf{s}}. \tag{5.16}$$

Plugging (5.15)–(5.16) into (5.14), and solving for $\check{\mathbf{v}}$ results in

$$\begin{aligned}\check{\mathbf{v}} =& -2\,\mathrm{Re}\left[(\mathbf{I}_3 \otimes \mathbf{A})^{-1}\mathbf{T}^{-1}\,\mathrm{bdiag}(\{\mathbf{Z}_n\})\mathbf{T}(\mathbf{I}_3 \otimes \mathbf{F}^\top)\check{\mathbf{s}}\right]\\ &- v_0(\mathbf{I}_3 \otimes \mathbf{A})^{-1}(\mathbf{1}_3 \otimes \mathbf{a}_0).\end{aligned} \tag{5.17}$$

Using the facts $\mathbf{A}^{-1} = -\mathbf{F}$, $\mathbf{T}^{-1} = \mathbf{T}^\top$, $\mathbf{F}\mathbf{a}_0 = \mathbf{1}_N$, and properties of the Kronecker product, (5.17) becomes

$$\check{\mathbf{v}} = 2\,\mathrm{Re}\left[(\mathbf{I}_3 \otimes \mathbf{F})\mathbf{T}^\top\,\mathrm{bdiag}(\{\tilde{\mathbf{Z}}_n\})\mathbf{T}(\mathbf{I}_3 \otimes \mathbf{F}^\top)\check{\mathbf{s}}\right] + v_0\mathbf{1}_{3N}.$$

Substituting $\check{\mathbf{s}} = \mathbf{T}^\top\mathbf{s}$ and $\check{\mathbf{v}} = \mathbf{T}^{-1}\mathbf{v}$ proves the claim. □

According to (5.8a), power injections and flows are approximately decoupled across phases. Nonetheless, Proposition 3 asserts that squared voltage magnitudes depend on power injections from all phases. Building on the approximate microgird models (5.3) and (5.12), localized reactive power control schemes for voltage regulation are developed next.

5.2 Localized Reactive Power Control

Posing voltage regulation as an optimal power flow instance leads to a constrained optimization problem. Given that voltage regulation constraints couple reactive injections across the grid, developing localized solutions becomes challenging. To derive such solutions, the voltage regulation goal is relaxed here and posed as the generic minimization problem

$$\min_{\mathbf{q}\in\mathcal{Q}} f(\mathbf{q}) + c(\mathbf{q}) \tag{5.18}$$

where $f(\mathbf{q})$ is the cost of squared voltage magnitudes $\mathbf{v}$ deviating from their nominal value $v_0\mathbf{1}$; $c(\mathbf{q}) = \sum_{n=1}^{N} c_n(q_n)$ is a separable function modeling the potential cost for reactive power compensation; and $\mathcal{Q}$ is the feasible set of reactive injections. Since reactive power injections by DG inverters can be adjusted in real-time, whereas utility-owned voltage regulating equipment responds typically at a slower pace (e.g., every few minutes or hourly), the latter choice is assumed fixed to a value and it will not be considered here. Particular instances of the generic setup in (5.18) are instantiated next for single-phase grids.

5.2.1 Schemes for Single-phase Grids

A viable voltage deviation cost is $f_1(\mathbf{q}) := \frac{1}{2}\|\mathbf{v} - v_0\mathbf{1}\|_2^2$. This cost function tries to keep squared voltage magnitudes close to the nominal value v_0. Assuming $c(\mathbf{q}) = 0$ and $\mathcal{Q} = \mathbb{R}^N$, problem (5.18) boils down to the unconstrained quadratic program

$$\min_{\mathbf{q}} \tfrac{1}{2}\|\mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q}\|_2^2. \tag{5.19}$$

Localized voltage regulation schemes assuming unlimited reactive power support have also been considered in [124] and [74]. Obviously, since $\mathbf{X}$ is invertible, problem (5.19) has the unique minimizer $\mathbf{q}^* = -\mathbf{X}^{-1}\mathbf{R}\mathbf{p}$ that achieves perfect voltage regulation $\mathbf{v}(\mathbf{q}^*) = v_0\mathbf{1}$. Finding $\mathbf{q}^*$ requires knowing the active injections $\mathbf{p}$ over all buses. Using the structure of $\mathbf{R}$ and $\mathbf{X}$, vector $\mathbf{q}^*$ can be alternatively expressed as

$$\mathbf{q}^* = \mathbf{A}^\top \operatorname{diag}\left(\left\{\frac{r_n}{x_n}\right\}\right)\mathbf{P}. \tag{5.20}$$

The entry q_n^* is a linear combination of the active powers flowing in and out of bus n with the related r_n/x_n ratios as coefficients. Although P_n denotes the power flow seen from the sending end of line n, the receiving end will measure approximately $-P_n$ due to the small loss assumption. The minimizer (5.20) can be found in a localized way only if bus n measures power flows on incident lines.

Alternatively, a gradient descent scheme would iteratively update reactive injections over iteration k as

$$\mathbf{q}^{k+1} = \mathbf{q}^k - \mu\mathbf{g}^k \tag{5.21}$$

where $\mu > 0$ is a step size, and $\mathbf{g}^k = \mathbf{X}^\top(\mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q}^k) = \mathbf{X}(\mathbf{v}^k - v_0\mathbf{1})$ is the gradient of $f_1(\mathbf{q})$

at $\mathbf{q}^k$. Unfortunately, such a scheme cannot be implemented in a localized fashion. However, the next proposition asserts that the rule

$$\mathbf{q}^{k+1} = \mathbf{q}^k - \mu(\mathbf{v}^k - v_0\mathbf{1}) \tag{5.22}$$

converges to $\mathbf{q}^*$ for an appropriately small μ.

Proposition 4. *If* $\mu \in \big(0, 2\lambda_{\min}(\mathbf{X})/\lambda_{\max}^2(\mathbf{X})\big)$*, the rule of* (5.22) *converges to the minimizer of* (5.19).

Proof of Proposition 4. The claim is an application of [7, Prop. 2.1]. The Lipschitz constant of $\nabla f_1(\mathbf{q})$ is $\lambda_{\max}^2(\mathbf{X})$. Observe also that $\|\mathbf{g}^k\|_2^2 = (\mathbf{v}^k - v_0\mathbf{1})^\top \mathbf{X}\mathbf{X}^\top(\mathbf{v}^k - v_0\mathbf{1}) \leq \lambda_{\max}^2(\mathbf{X})\|\mathbf{v}^k - v_0\mathbf{1}\|_2^2$, or

$$\|\mathbf{v}^k - v_0\mathbf{1}\|_2 \geq \frac{1}{\lambda_{\max}(\mathbf{X})}\|\mathbf{g}^k\|_2.$$

Note further that $(\mathbf{v}^k - v_0\mathbf{1})^\top\mathbf{g}^k = (\mathbf{v}^k - v_0\mathbf{1})^\top\mathbf{X}(\mathbf{v}^k - v_0\mathbf{1}) \geq \lambda_{\min}(\mathbf{X})\|\mathbf{v}^k - v_0\mathbf{1}\|_2^2$, implying that $(\mathbf{v}^k - v_0\mathbf{1})$ is a descent direction for $f_1(\mathbf{q}^k)$ – although not the steepest one. □

The descent rule in (5.22) scaled by a diagonal matrix has been shown to converge for a more detailed model in [124]. Solving (5.19) may be of little practical use: Distributed generation units may not be installed on every bus and their reactive power resources are limited. Moreover, the power electronics found on a photovoltaic (PV) at bus n have finite apparent power capability s_n, which limits q_n depending on the current active injection (solar generation) as $p_n^2 + q_n^2 \leq s_n^2$ or $|q_n| \leq \sqrt{s_n^2 - p_n^2}$. In general, $\mathbf{q}$ is constrained to lie within the time-varying but known box $\mathcal{Q} := \{\mathbf{q} : \underline{q}_n \leq q_n \leq \overline{q}_n \ \forall n\}$ where $(\underline{q}_n, \overline{q}_n)$ are the associated lower and upper bounds for the reactive power injection in bus n. It will be henceforth assumed that $\underline{q}_n \leq 0 \leq \overline{q}_n$, even though our analysis can be readily extended otherwise too. Buses without reactive power support can be obviously modeled by selecting $\underline{q}_n = \overline{q}_n = 0$.

In this practically pertinent setup where the voltage regulation problem in (5.18) is constrained as $\mathbf{q} \in \mathcal{Q}$, one could try implementing the projected version of (5.22), that is

$$\mathbf{q}^{k+1} = \Big[\mathbf{q}^k - \mu(\mathbf{v}^k - v_0\mathbf{1})\Big]_{\mathcal{Q}} \tag{5.23}$$

where $[\mathbf{x}]_{\mathcal{Q}} := \arg\min_{\mathbf{z}\in\mathcal{Q}} \|\mathbf{x} - \mathbf{z}\|_2$ denotes the projection operator on $\mathcal{Q}$. Unfortunately, this

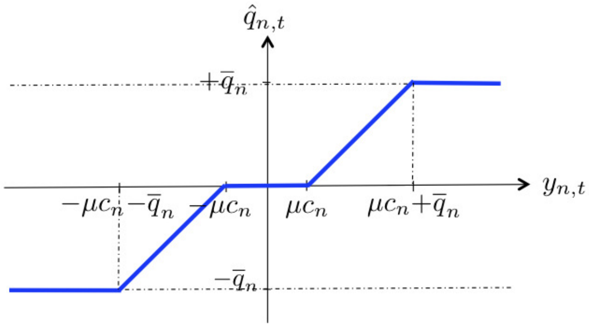


Figure 5.1: The control rule of (5.28).

seemingly meaningful control rule is not guaranteed to converge [11].

A localized voltage regulation scheme can be obtained via a different voltage deviation cost $f(\mathbf{q})$. As advocated in [36], a meaningful choice is the cost

$$f_2(\mathbf{q}) = \frac{1}{2}\|\mathbf{Rp} + \mathbf{Xq}\|^2_{\mathbf{X}^{-1}} \tag{5.24}$$

with the rotated norm defined as $\|\mathbf{x}\|^2_{\mathbf{B}} := \mathbf{x}^\top \mathbf{Bx}$ for $\mathbf{B} \succ \mathbf{0}$. As proved in [67], the cost in (5.24) is equivalent to $f_2(\mathbf{q}) = \frac{1}{4}\sum_{n=1}^N \frac{(v_{\pi_n} - v_n)^2}{x_n}$. Although minimizing $f_2(\mathbf{q})$ over $\mathbf{q} \in \mathcal{Q}$ penalizes scaled voltage magnitude deviations between adjacent buses, obviously, it does not guarantee that voltages will lie within any desired range. Nevertheless, $f_2(\mathbf{q})$ has the important feature that its gradient

$$\nabla f_2(\mathbf{q}) = \mathbf{Rp} + \mathbf{Xq} = \mathbf{v}(\mathbf{q}) - v_0\mathbf{1} \tag{5.25}$$

equals the deviation of squared voltage magnitudes from the nominal, and its n-th entry can be measured locally at bus n.

To deter engaging PV power inverters for negligible voltage deviations, a reactive power compensation cost $c(\mathbf{q})$ should be also considered. Given that negative and positive reactive power injections are equally important, a reasonable option for voltage regulation would be solving the problem

$$\min_{\mathbf{q} \in \mathcal{Q}} \; h_2(\mathbf{q}) = f_2(\mathbf{q}) + c(\mathbf{q}) \tag{5.26}$$

where $c(\mathbf{q}) := \sum_{n=1}^N c_n|q_n|$. Again, due to the strong convexity of $f_2(\mathbf{q})$, problem (5.26) has a unique minimizer in $\mathcal{Q}$. As shown in [67], the minimizer of (5.26) can be found via simple proximal gradient iterations: At iterate k, each bus n measures the quantity

$$y_n^k := q_n^k - \mu(v_n^k - v_0) \tag{5.27}$$

for a step size $\mu > 0$. Voltage magnitude deviations $v_n^k - v_0$ are assumed to be acquired without noise. It then updates its reactive power injection according to the control rule

$$q_n^{k+1} := \mathcal{S}_\mu(y_n^k; c_n, \underline{q}_n, \overline{q}_n). \tag{5.28}$$

The operator $\mathcal{S}_\mu(y; c, \underline{q}, \overline{q})$ is defined as

$$\mathcal{S}_\mu(y; c, \underline{q}, \overline{q}) := \begin{cases} +\overline{q} & , \ y > \overline{q} + \mu c \\ y - \mu c & , \ \mu c < y \leq \overline{q} + \mu c \\ 0 & , \ -\mu c \leq y \leq \mu c \\ y + \mu c & , \ \underline{q} - \mu c \leq y < -\mu c \\ \underline{q} & , \ y < \underline{q} - \mu c \end{cases} \tag{5.29}$$

and is shown in Figure 5.1. The proximal gradient rule can be applied for any other option of convex and separable reactive power compensation cost $c(\mathbf{q}) = \sum_{n=1}^N c_n(q_n)$. Two interesting cases are discussed next.

Remark 6. In the absence of reactive power compensation cost, that is when $c_n(q_n) = 0$ for all n, the operator $\mathcal{S}_\mu(y_n; 0, \underline{q}_n, \overline{q}_n)$ simply projects y_n onto $[\underline{q}_n, \overline{q}_n]$.

Remark 7. Reference [36] considers the reactive power compensation costs $c_n(q_n) = \frac{c_{n2}}{2} q_n^2 + c_n|q_n|$ with positive parameters c_n and c_{n2} for all n. In this case, a proximal gradient scheme would yield the update $q_n^{k+1} := \mathcal{S}_\mu(\hat{y}_n^k; c_n, \underline{q}_n, \overline{q}_n)$, where $\hat{y}_n^k := (1 + \mu c_{n2}) q_n^k - \mu(v_n^k - v_0)$; [cf. (5.27)–(5.28)].

If the step size is selected as $\mu \in \left(0, 2\lambda_{\max}^{-1}(\mathbf{X})\right)$, the control rule of (5.27)–(5.28) converges to the minimizer of (5.26) [83]. Actually, with the optimal step size $\mu = \lambda_{\max}^{-1}(\mathbf{X})$, the convergence rate is linear, but proportional to the condition number $\kappa(\mathbf{X}) := \lambda_{\max}(\mathbf{X})/\lambda_{\min}(\mathbf{X})$ of matrix $\mathbf{X}$. It is worth stressing that $\kappa(\mathbf{X})$ can be relatively large: the Southern California Edison 47-bus grid and the IEEE 34-bus benchmark exhibit $\kappa(\mathbf{X}) = 1.6 \times 10^4$ and 5.5×10^4, respectively [34], [104]. Ways to improve the convergence speed are reviewed next.

5.2.2 Accelerated Control Schemes

A way to speed up proximal gradient schemes is via diagonal scaling. In detail, the reactive injection variables $\mathbf{q}$ can be transformed to $\breve{\mathbf{q}}$ as $\mathbf{q} = \mathbf{D}^{1/2}\breve{\mathbf{q}}$ for a diagonal matrix $\mathbf{D} \succ \mathbf{0}$. Instead of (5.26), we can solve the equivalent problem

$$\min_{\mathbf{D}^{1/2}\breve{\mathbf{q}} \in \mathcal{Q}} h_2(\mathbf{D}^{1/2}\breve{\mathbf{q}})$$

through the proximal gradient method. The update for the transformed variable $\breve{\mathbf{q}}^{k+1}$ is found as the solution to

$$\min_{\mathbf{D}^{1/2}\breve{\mathbf{q}} \in \mathcal{Q}} c(\mathbf{D}^{1/2}\breve{\mathbf{q}}) + \tfrac{1}{2\mu}\|\breve{\mathbf{q}} - (\breve{\mathbf{q}}^k - \mu\mathbf{D}^{1/2}\nabla f_2(\mathbf{D}^{1/2}\breve{\mathbf{q}}^k))\|_2^2$$

and the convergence rate now depends on $\kappa\left(\mathbf{D}^{1/2}\mathbf{X}\mathbf{D}^{1/2}\right)$. Translating the iterations back to the original variables $\mathbf{q}$ yields

$$\mathbf{q}^{k+1} := \arg\min_{\mathbf{q} \in \mathcal{Q}} c(\mathbf{q}) + \tfrac{1}{2\mu}\|\mathbf{q} - (\mathbf{q}^k - \mu\mathbf{D}\nabla f_2(\mathbf{q}^k))\|_{\mathbf{D}^{-1}}^2 \tag{5.30}$$

which is separable across buses as

$$q_n^{k+1} = \arg\min_{\underline{q}_n \leq q_n \leq \bar{q}_n} d_n c_n |q_n| + \frac{1}{2\mu}\left(q_n - u_n^k\right)^2 \tag{5.31}$$

where $u_n^k := q_n^k - \mu d_n(v_n^k - v_0)$. It can be easily verified that the minimizer of (5.30) is provided by the control rule

$$q_n^{k+1} := \mathcal{S}_{\mu d_n}(u_n^k; d_n c_n, \underline{q}_n, \bar{q}_n) \tag{5.32}$$

where d_n is the n-th diagonal entry of the scaling matrix $\mathbf{D}$. The control rule in (5.32) corresponds to the rule of (5.27)–(5.28) with the step size μ adjusted to μd_n per bus n. The scaling matrix $\mathbf{D}$ could be selected to minimize $\kappa\left(\mathbf{D}^{1/2}\mathbf{X}\mathbf{D}^{1/2}\right)$. A relatively simple choice for $\mathbf{D}$ is to assign vector $\operatorname{diag}(\mathbf{X})$ on the main diagonal of $\mathbf{D}^{-1}$. This option sets the diagonal entries of $\mathbf{D}^{1/2}\mathbf{X}\mathbf{D}^{1/2}$ to unity. To meet faster voltage regulation rates, the accelerated voltage regulation scheme of [67] is reviewed and simplified next.

In [67], we derived an accelerated proximal gradient scheme by adapting Nesterov's method [81]. The corresponding control rule was shown to be a simple modification of (5.28) (see [67] for details):

$$q_n^{k+1} := \mathcal{S}_\mu(\tilde{y}_n^k; c_n, \underline{q}_n, \overline{q}_n) \tag{5.33}$$

where variable $\tilde{y}_n^k$ is updated using the y_n^k from (5.27) as

$$\tilde{y}_n^k := (1+\beta_k) y_n^k - \beta_k y_n^{k-1} \tag{5.34}$$

with $\beta_k = \frac{k-1}{k+2}$ for all $k \geq 1$. Compared to (5.28), the control rule in (5.33) introduces memory in calculating $\tilde{y}_n^k$ as a linear combination of y_n^k and y_n^{k-1}. The linear combination coefficients depend on the time-varying β_k that converges to 1. Note that the formula for β_k has been simplified from the one used in [67, Eqs. (20)-(21)].

If the step size is selected as $\mu = \lambda_{\max}^{-1}(\mathbf{X})$ and the sequence β_k is reset to zero every $2\sqrt{\kappa(\mathbf{X})}$ iterations, the reactive control rule of (5.33) converges linearly to an ϵ-optimal cost value within $-\frac{2\log\epsilon}{\log 2}\sqrt{\kappa(\mathbf{X})}$ iterations. For grids with high $\kappa(\mathbf{X})$, this modified scheme offers accelerated convergence. Numerical tests show that even without resetting the sequences and without knowing precisely $\lambda_{\max}(\mathbf{X})$, the rule in (5.33) offers superior performance over both rules (5.28) and (5.32).

Remark 8. The IEEE 1547.8 standard suggests the following reactive power injection rule for the DG at bus n [55]

$$q_n^{k+1} = \tilde{\mathcal{S}}_n(v_0 - v_n^k) \tag{5.35}$$

where the function $\tilde{\mathcal{S}}_n$ is an increasing piecewise linear function similar to the one shown in Figure 5.1. Comparing (5.35) with the control rules of (5.28) or (5.32), suggests that all control rules have similar computational requirements. Nevertheless, the rule in (5.35) can exhibit oscillations as reported in [35], while the schemes presented here exhibit well-understood convergence properties.

5.2.3 Protocols for Multi-phase Grids

Distributed generation and demand-response programs can lead to unbalanced conditions. This section aims at generalizing the schemes of Section 5.2.1 for multi-phase grids. The problem of voltage regulation in multi-phase grids can be posed as in (5.18); yet now the dependence of squared voltage magnitudes on nodal injections is governed by the model in (5.12). Before devising voltage regulation schemes, critical properties of the involved matrices are studied first.

Let us focus on the 3×3 complex matrices $\tilde{\mathbf{Z}}_n$ defined in (5.8). Let $z_n^{ij} = r_n^{ij} + jx_n^{ij}$ be the (i, j)-th entry of $\mathbf{Z}_n$. From the symmetry of $\mathbf{Z}_n$ and the identity $\alpha^2 = \alpha^\star = -\frac{1}{2} + j\frac{\sqrt{3}}{2}$, matrix $\tilde{\mathbf{Z}}_n$ becomes

$$\tilde{\mathbf{Z}}_n = \mathrm{diag}(\boldsymbol{\alpha}^\star)\mathbf{Z}_n \,\mathrm{diag}(\boldsymbol{\alpha}) = \begin{bmatrix} z_n^{11} & \alpha^\star z_n^{12} & \alpha z_n^{13} \\ \alpha z_n^{21} & z_n^{22} & \alpha^\star z_n^{23} \\ \alpha^\star z_n^{31} & \alpha z_n^{32} & z_n^{33} \end{bmatrix}.$$

Therefore, matrix $\mathrm{Im}[\tilde{\mathbf{Z}}_n]$ can be decomposed as

$$\mathrm{Im}[\tilde{\mathbf{Z}}_n] = \tilde{\mathbf{X}}_n + \tilde{\mathbf{R}}_n \tag{5.36a}$$

$$\tilde{\mathbf{X}}_n = \frac{1}{2}\begin{bmatrix} 2x_n^{11} & -x_n^{12} & -x_n^{13} \\ -x_n^{12} & 2x_n^{22} & -x_n^{23} \\ -x_n^{13} & -x_n^{23} & 2x_n^{33} \end{bmatrix} \tag{5.36b}$$

$$\tilde{\mathbf{R}}_n = \frac{\sqrt{3}}{2}\begin{bmatrix} 0 & -r_n^{12} & r_n^{13} \\ r_n^{12} & 0 & -r_n^{23} \\ -r_n^{13} & r_n^{23} & 0 \end{bmatrix} \tag{5.36c}$$

where $\tilde{\mathbf{X}}_n$ is a symmetric matrix ($\tilde{\mathbf{X}}_n = \tilde{\mathbf{X}}_n^\top$) associated to the reactive part of $\mathbf{Z}_n$, and $\tilde{\mathbf{R}}_n$ is an anti-symmetric matrix ($\tilde{\mathbf{R}}_n = -\tilde{\mathbf{R}}_n^\top$) depending on the resistive part of $\mathbf{Z}_n$. Using (5.36), the next fact holds:

Proposition 5. *If $\tilde{\mathbf{X}}_n$ is strictly diagonally dominant with positive diagonal entries, then* $\mathrm{Im}[\tilde{\mathbf{Z}}_n] \succ \mathbf{0}$.

To prove Proposition 5, it suffices to show that the symmetric component of $\mathrm{Im}[\tilde{\mathbf{Z}}_n]$ is strictly positive-definite. If mutual and self-reactances satisfy $2x_n^{ii} > \sum_{j\neq i} |x_n^{ij}|$ for all i, then $\tilde{\mathbf{X}}_n$ is diagonally dominant with positive diagonal entries, and thus, $\tilde{\mathbf{X}}_n \succ \mathbf{0}$. Due to the structure of distribution lines, the aforementioned inequalities are satisfied in general.

The decomposition in (5.36) carries over to $\mathbf{X}$ in (5.13b) as:

$$\mathbf{X} = \mathbf{X}_x + \mathbf{X}_r \tag{5.37}$$

where $\mathbf{X}_x := 2\mathbf{M}\,\mathrm{bdiag}(\{\tilde{\mathbf{X}}_n\})\mathbf{M}^\top$ is symmetric, and $\mathbf{X}_r := 2\mathbf{M}\,\mathrm{bdiag}(\{\tilde{\mathbf{R}}_n\})\mathbf{M}^\top$ is anti-symmetric. Because $\mathbf{M}$ is invertible, matrix $\mathbf{X}$ is positive-definite if and only if matrix $\mathrm{bdiag}(\{\tilde{\mathbf{X}}_n\})$ is, hence leading to the corollary:

Corollary 1. *If* $\mathrm{Im}[\mathbf{Z}_n]$ *is strictly diagonally dominant with positive diagonal entries for all* n*, then* $\mathbf{X} \succeq \mathbf{0}$.

We next elaborate on how bus voltage magnitudes are affected by reactive power injections. According to (5.12), vector $\check{\mathbf{v}}$ is linearly related to reactive injections $\check{\mathbf{q}}$ via the matrix

$$\begin{aligned}\check{\mathbf{X}} &:= \mathbf{T}^\top \mathbf{X}\mathbf{T} \\ &= 2(\mathbf{I}_3 \otimes \mathbf{F})\mathbf{T}^\top\,\mathrm{bdiag}(\{\mathrm{Im}[\tilde{\mathbf{Z}}_n]\})\mathbf{T}(\mathbf{I}_3 \otimes \mathbf{F}^\top).\end{aligned} \tag{5.38}$$

The effect of reactive power injection $\check{\mathbf{q}}_j$ to the squared voltage magnitude $\check{\mathbf{v}}_i$ is described by the (i,j) entry of $\check{\mathbf{X}}$. Let entry i correspond to phase ϕ_i of bus n_i, and entry j to phase ϕ_j of bus n_j. It can be verified that

$$\begin{aligned}\check{\mathbf{X}}_{i,j} &= (\mathbf{e}_{\phi_i} \otimes \mathbf{e}_{n_i})^\top \check{\mathbf{X}}(\mathbf{e}_{\phi_j} \otimes \mathbf{e}_{n_j}) \\ &= (\mathbf{e}_{\phi_i} \otimes \mathbf{f}_{n_i})^\top \mathbf{T}^\top\,\mathrm{bdiag}(\{\mathrm{Im}[\tilde{\mathbf{Z}}_n]\})\mathbf{T}(\mathbf{e}_{\phi_j} \otimes \mathbf{f}_{n_j})\end{aligned}$$

where $\mathbf{f}_k^\top$ is the k-th row of matrix $\mathbf{F}$. By the definition of $\mathbf{T}$ in (5.11), the products $\mathbf{T}(\mathbf{e}_{\phi_i} \otimes \mathbf{f}_{n_i})$ and $\mathbf{T}(\mathbf{e}_{\phi_j} \otimes \mathbf{f}_{n_j})$ can be expressed as $\mathbf{f}_{n_i} \otimes \mathbf{e}_{\phi_i}$ and $\mathbf{f}_{n_j} \otimes \mathbf{e}_{\phi_j}$, respectively. Exploiting the structure of $\mathrm{bdiag}(\{\mathrm{Im}[\tilde{\mathbf{Z}}_n]\})$ and since $\mathbf{F}$ is lower triangular, the entry $\check{\mathbf{X}}_{i,j}$ can be expressed as

$$\check{\mathbf{X}}_{i,j} = \sum_{k=1}^{\min\{n_i,n_j\}} \mathrm{Im}[\tilde{\mathbf{Z}}_k]_{\phi_i,\phi_j}\mathbf{F}_{n_i,k}\mathbf{F}_{n_j,k}. \tag{5.39}$$

Recall that $\mathbf{F}$ has non-negative entries, while for overhead distribution lines the parameters $x_n^{\phi_i\phi_j}$ and $r_n^{\phi_i\phi_j}$ appearing in (5.36b)–(5.36c) are typically non-negative. According to (5.39), three cases can be distinguished:

(c1) If $\phi_i = \phi_j$, $\check{\mathbf{X}}_{i,j} = \sum_{k=1}^{\min\{n_i,n_j\}} x_k^{\phi_i\phi_i}\mathbf{F}_{n_i,k}\mathbf{F}_{n_j,k} > 0$. Thus, as in single-phase grids, injecting more reactive power into a bus raises voltages at all buses in the *same phase*.

(c2) When $(\phi_i, \phi_j) \in \{(a,b),(b,c),(c,a)\}$, it follows that

$$\check{\mathbf{X}}_{i,j} = -\frac{1}{2}\sum_{k=1}^{\min\{n_i,n_j\}}\left(x_k^{\phi_i\phi_i} + \sqrt{3}r_k^{\phi_i\phi_j}\right)\mathbf{F}_{n_i,k}\mathbf{F}_{n_j,k} < 0.$$

Thus, injecting more reactive power into a bus decreases the voltage magnitudes at all buses of the *preceding phase* in the positive sequence ordering.

(c3) If $(\phi_i, \phi_j) \in \{(a,c),(b,a),(c,b)\}$, then

$$\check{\mathbf{X}}_{i,j} = -\tfrac{1}{2}\sum_{k=1}^{\min\{n_i,n_j\}}\left(x_k^{\phi_i\phi_i} - \sqrt{3}r_k^{\phi_i\phi_j}\right)\mathbf{F}_{n_i,k}\mathbf{F}_{n_j,k}.$$

Thus, the effect of reactive power injections into one phase to the voltage magnitudes of the *following phase* depends on the differences $x_k^{\phi_i\phi_i} - \sqrt{3}r_k^{\phi_i\phi_j}$ appearing in the last sum. Actually, if every bus serves all phases and since $|x_k^{\phi_i\phi_i} - \sqrt{3}r_k^{\phi_i\phi_j}| \leq x_k^{\phi_i\phi_i} + \sqrt{3}r_k^{\phi_i\phi_j}$, the effect of one phase to the following phase is less significant than its effect on the previous phase.

An illustration of the influence patterns across phases is shown in Figure 5.2. Evidenced by the previous analysis and Fig. 5.2, decreasing reactive injections in phase b to cater over-voltage conditions on phase b would aggravate possible over-voltage problems on phase a. In this context, voltage regulation becomes even more challenging in multi-phase grids.

Similar to single-phase grids, let us first consider the simplest voltage regulation scenario: The voltage deviation cost is $f_1(\mathbf{q}) = \frac{1}{2}\|\mathbf{v} - v_0\mathbf{1}_{3N}\|_2^2$, there is no reactive power compensation cost $c(\mathbf{q})$, and reactive power is unconstrained:

$$\min_{\mathbf{q}} \tfrac{1}{2}\|\mathbf{Rp} + \mathbf{Xq}\|_2^2. \tag{5.40}$$

Again, the positive-definiteness of $\mathbf{X}$ [cf. Corollary. 1] guarantees the uniqueness of the minimizer in (5.40). Contrary to the single-phase case though, the minimizer of (5.40) cannot be found locally even if bus n measures all phase flows on its incident lines [cf. (5.20)] because of the structure of $\mathbf{R}$ and $\mathbf{X}$.

Similar to single-phase grids, a gradient descent scheme cannot be implemented locally. However, the simple update

$$\mathbf{q}^{k+1} = \mathbf{q}^k - \mu(\mathbf{v}^k - v_0\mathbf{1}) \tag{5.41}$$

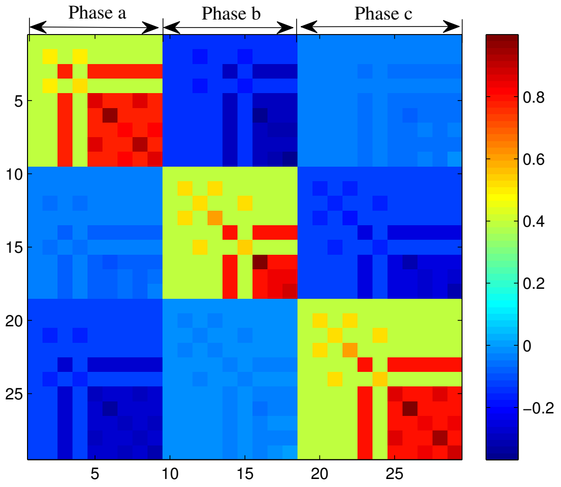


Figure 5.2: Matrix $\check{\mathbf{X}}$ (normalized to unity maximum entry) relating voltage magnitudes $\check{\mathbf{v}}$ to reactive power injections $\check{\mathbf{q}}$ for the IEEE 13-bus grid benchmark depicts the coupling across phases.

still constitutes a convergent rule:

Proposition 6. *If* $\mu \in \left(0, \frac{2\lambda_{\min}(\mathbf{X}_x)}{\lambda_{\max}(\mathbf{X}^\top\mathbf{X})}\right)$*, the rule of* (5.41) *converges to the minimizer of* (5.40)*.*

Proof for Proposition 6. As in Prop. 4, showing Prop. 6 relies on an application of [7, Prop. 2.1]. In this case, the Lipschitz constant for $\nabla f_1(\mathbf{q})$ is $\lambda_{\max}(\mathbf{X}^\top\mathbf{X})$, and it also holds that $\|\mathbf{v}_k - v_0\mathbf{1}\|_2 \geq \lambda_{\min}^{-1/2}(\mathbf{X}^\top\mathbf{X})\|\nabla f_1(\mathbf{q}^k)\|_2$. The critical point in the multi-phase grid case though is that the positive-definiteness of $\mathbf{X}_x$ guarantees that $(\mathbf{v}^k - v_0\mathbf{1})$ is a descent direction, since $(\mathbf{v}^k - v_0\mathbf{1})^\top\nabla f_1(\mathbf{q}^k) = (\mathbf{v}^k - v_0\mathbf{1})^\top\mathbf{X}^\top(\mathbf{v}^k - v_0\mathbf{1}) = (\mathbf{v}^k - v_0\mathbf{1})^\top\mathbf{X}_x(\mathbf{v}^k - v_0\mathbf{1}) \leq \lambda_{\min}(\mathbf{X}_x)\|\mathbf{v}^k - v_0\mathbf{1}\|_2^2$. □

The key point here is that albeit $\mathbf{X}$ is not symmetric, it is still positive-definite under the assumptions of Corollary 1. Hence, rule (5.41) remains a valid descent direction for the unconstrained problem in (5.40).

Reactive power injections are typically constrained in a feasible set $\mathcal{Q}$, and there may also be some reactive power compensation cost $c(\mathbf{q})$. In that case, the update in (5.41) is not practical. Recall that for single-phase grids with constrained reactive resources, the original voltage magnitude deviation function $f_1(\mathbf{q}) = \frac{1}{2}\|\mathbf{v} - v_0\mathbf{1}\|_2^2$ was replaced by a rotated Euclidean norm of the voltage deviations, namely $f_2(\mathbf{q}) = \frac{1}{2}\|\mathbf{X}^{-1/2}(\mathbf{v} - v_0\mathbf{1})\|_2^2$, which resulted in localized updates. The choice of $f_1(\mathbf{q})$ would fail yielding localized solutions in multi-phase grids too. Although $\mathbf{X}$ in multi-phase grids is positive-definite, the lack of symmetry does not allow us to extend the approach with $f_2(\mathbf{q})$.

Nonetheless, let us study the behavior of the control rule of (5.28) under unbalanced conditions. Assume that the DG unit at each bus performs the control rule of (5.27)–(5.28) that can be equivalently expressed as (cf. [67])

$$\mathbf{y}^k = \mathbf{q}^k - \mu(\mathbf{v}^k - v_0\mathbf{1}) \tag{5.42a}$$

$$\mathbf{q}^{k+1} = \text{prox}_{\mu c,\mathcal{Q}}[\mathbf{y}^k] \tag{5.42b}$$

where the proximal operator is defined as

$$\text{prox}_{\mu c,\mathcal{Q}}\,[\mathbf{y}] := \arg\min_{\mathbf{w}\in\mathcal{Q}}\ \mu c(\mathbf{w}) + \tfrac{1}{2}\|\mathbf{w}-\mathbf{y}\|_2^2. \tag{5.43}$$

Compared to the single-phase grid case, the major difference is that now $\mathbf{v}$ is related to $\mathbf{q}$ according to the model in (5.12). The iterates produced by (5.42) satisfy:

$$\begin{aligned}\|\mathbf{q}^{k+1}-\mathbf{q}^k\|_2 &= \|\,\text{prox}_{\mu c,\mathcal{Q}}[\mathbf{y}^k] - \text{prox}_{\mu c,\mathcal{Q}}[\mathbf{y}^{k-1}]\|_2\\ &\leq \|\mathbf{y}^k - \mathbf{y}^{k-1}\|_2 = \|\,(\mathbf{I}-\mu\mathbf{X})\left(\mathbf{q}^k - \mathbf{q}^{k-1}\right)\|_2\\ &\leq \|\mathbf{I}-\mu\mathbf{X}\|_2\|\mathbf{q}^k-\mathbf{q}^{k-1}\|_2.\end{aligned}$$

where the first inequality follows from the non-expansive property of the proximal operator (cf. [83, Prop. 5.1.8]); the equality from (5.42a); and the last inequality from the definition of the maximum singular value. If μ is selected such that $\|\mathbf{I}-\mu\mathbf{X}\|_2 < 1$, then (5.42) constitutes a non-expansive mapping and it therefore converges to the equilibrium point defined by

$$\mathbf{q}^* = \text{prox}_{\mu c,\mathcal{Q}}[\mathbf{q}^* - \mu(\mathbf{v}(\mathbf{q}^*) - v_0\mathbf{1})] \tag{5.44}$$

or, from the definition of the proximal operator in (5.43), by

$$\mathbf{q}^* = \arg\min_{\mathbf{w}\in\mathcal{Q}}\ \mu c(\mathbf{w}) + \tfrac{1}{2}\|\mathbf{w} - [\mathbf{q}^* - \mu(\mathbf{v}(\mathbf{q}^*) - v_0\mathbf{1})]\|_2^2.$$

Vector $\mathbf{q}^*$ is thus defined as the minimizer of an optimization problem, and it cannot be expressed in closed form. Of course, the equilibrium point $\mathbf{q}^*$ does not necessarily coincide with the minimizer of any voltage regulation optimization problem. The next step size range guarantees $\|\mathbf{I} - \mu\mathbf{X}\|_2 < 1$, and therefore convergence of (5.42) (see the appendix for a proof):

Proposition 7. *Let* $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ *be the eigenvalue decomposition of* $\mathbf{X}\mathbf{X}^\top$*. If*

$$\mu \in \left(0, \lambda_{\min}\left(\mathbf{\Lambda}^{-1/2}\mathbf{U}^\top(\mathbf{X} + \mathbf{X}^\top)\mathbf{U}\mathbf{\Lambda}^{-1/2}\right)\right)$$

then $\|\mathbf{I} - \mu\mathbf{X}\|_2 < 1$.

Proof of Proposition 7. By definition, it holds that

$$\|\mathbf{I} - \mu\mathbf{X}\|_2^2 = \lambda_{\max}\left((\mathbf{I} - \mu\mathbf{X})^\top(\mathbf{I} - \mu\mathbf{X})\right) = \lambda_{\max}\left(\mathbf{I} - \mu\mathbf{X}_\mu\right)$$

where $\mathbf{X}_\mu := \mathbf{X} + \mathbf{X}^\top - \mu\mathbf{X}\mathbf{X}^\top$. Guaranteeing $\|\mathbf{I} - \mu\mathbf{X}\|_2 < 1$ is equivalent to satisfying $\lambda_{\max}\left(\mathbf{I} - \mu\mathbf{X}_\mu\right) < 1$, or, simply $1 - \mu\lambda_{\min}\left(\mathbf{X}_\mu\right) < 1$. Granted that $\mu > 0$, the latter is equivalent to ensuring $\mathbf{X}_\mu$ to be a positive-definite matrix, i.e., $\mathbf{X} + \mathbf{X}^\top \succ \mu\mathbf{X}\mathbf{X}^\top$. It can be easily verified that pre/post-multiplying the aforementioned linear matrix inequality by $\mathbf{\Lambda}^{-1/2}\mathbf{U}^\top$/$\mathbf{U}\mathbf{\Lambda}^{-1/2}$ yields the condition imposed on μ by Proposition 7. □

Practical distribution grids do not have all phases at all buses. The previous analysis carries over to such cases, if the related entries of $\mathbf{Z}_n$'s and the corresponding (re)active power injections are set to zero. For the eigendecompositions of $\mathbf{X}\mathbf{X}^\top$ and $\mathbf{X} + \mathbf{X}^\top$, their rows and columns related to non-existing bus-phase pairs are simply removed.

5.3 Distributed Multi-period Energy Management Schemes

The local reactive power control schemes described in 5.2 suffer from degraded performance in multi-phase grids due to the inter-phase coupling. In addition, broadening the aforementioned voltage regulation approaches to encompass control of the temporally correlated active loads, e.g., EVs, is non-trivial. To manage both reactive power and temporally correlated active power, this section first formulates a multi-period energy management program. Subsequently, the multi-period problem is tackled using a decentralized scheme based on the ADMM and Frank-Wolfe algorithms.

5.3.1 Multi-period Scheduling in Multi-phase Microgrids

Consider a multi-phase distribution feeder comprising $N+1$ buses indexed by $n \in \mathcal{N} := \{0, 1, \ldots, N\}$. The distribution grid that is assumed to be functionally radial, can be captured by the approximate power flow model (5.8). For simplicity, only EV loads are taken into account in demand management. Specifically, let $\mathcal{M}_{n,\phi}$ represent the set of vehicles located on phase ϕ of bus n, and $M_{n,\phi} := |\mathcal{M}_{n,\phi}|$. The base active and reactive power loads $\{(\mathbf{d}_n(t), \mathbf{q}_n^d(t))\}$ for all n and t need to be predicted in advance. Active power loads $\mathbf{p}_n^d(t)$ consist of two parts: the base loads $\mathbf{d}_n(t)$ and the EV charging load. If $p_{n,\phi}^d(t)$ and $d_{n,\phi}(t)$ are respectively the total active load and the base load on phase ϕ of bus n, it holds that $p_{n,\phi}^d(t) = d_{n,\phi}(t) + \sum_{m\in\mathcal{M}_{n,\phi}} e_m(t)$, where $e_m(t)$ denote the energy charge for vehicle m at slot t as defined in Chapter 4. The cost $f_0(\mathbf{P}_0(t))$ of power supply from the main grid is convex and known in advance. Variables $\mathbf{p}_n^g(t)$ capture possible dispatchable generation distributed across the feeder, and $f_n^g(\mathbf{p}_n^g(t))$ is the associated convex quadratic cost for all $n \in \mathcal{N}$ and $t \in \mathcal{T}$.

To capture operational constraints, the following limits are introduced. Let $(\underline{p}_{n,\phi}^g, \underline{q}_{n,\phi}^g)$ be the lower, and $(\bar{p}_{n,\phi}^g, \bar{q}_{n,\phi}^g)$ the upper limits for distributed generation at phase $\phi \in \mathcal{P}_n$ of bus n. Define also $(\underline{v}_{n,\phi}, \bar{v}_{n,\phi})$ as the limits of squared voltage magnitudes at phase $\phi \in \mathcal{P}_n$ of bus n, $\bar{S}_n$ as the apparent power flow limits on line n, and $\bar{S}_f$ as the rated capacity of the feeder transformer. The utility company aims to minimize the total operation cost by coordinating generation dispatch and vehicle charging, while respecting operational and charging limitations. The pertinent microgrid management task can be posed as follows:

$$\min \sum_{t\in\mathcal{T}} \left[f_0(\mathbf{P}_0(t)) + \sum_{n\in\mathcal{N}} f_n^g\left(\mathbf{p}_n^g(t)\right) \right] \tag{5.45a}$$

$$\text{over } \{\mathbf{p}_n^g(t), \mathbf{q}_n^g(t), \mathbf{P}_n(t), \mathbf{Q}_n(t), \mathbf{v}_n(t)\}_{n\in\mathcal{N}, t\in\mathcal{T}}, \{\mathbf{e}_m\}$$

$$\text{s.to } \mathbf{p}_n^g(t) - \mathbf{p}_n^d(t) = \sum_{k\in\mathcal{C}_n} \mathbf{P}_k(t) - \mathbf{P}_n(t),\ \forall\, n, t \tag{5.45b}$$

$$\mathbf{q}_n^g(t) - \mathbf{q}_n^d(t) = \sum_{k\in\mathcal{C}_n} \mathbf{Q}_k(t) - \mathbf{Q}_n(t),\ \forall\, n, t \tag{5.45c}$$

$$\mathbf{v}_{\pi_n}(t) - \mathbf{v}_n(t) = \text{Re}\{\bar{\mathbf{Z}}_n(\mathbf{P}_n(t) + j\mathbf{Q}_n(t))\}, \forall n, t \tag{5.45d}$$

$$\underline{p}_{n,\phi}^g \leq p_{n,\phi}^g(t) \leq \bar{p}_{n,\phi}^g,\ \forall\, \phi \in \mathcal{P}_n, n, t \tag{5.45e}$$

$$\underline{q}_{n,\phi}^g \leq q_{n,\phi}^g(t) \leq \bar{q}_{n,\phi}^g,\ \forall\, \phi \in \mathcal{P}_n, n, t \tag{5.45f}$$

$$\underline{v}_n \leq v_{n,\phi}(t) \leq \bar{v}_n,\ \forall\, \phi \in \mathcal{P}_n, n, t \tag{5.45g}$$

$$P_{n,\phi}^2(t) + Q_{n,\phi}^2(t) \leq \bar{S}_n^2,\ \forall\, \phi \in \mathcal{P}_n, n \in \mathcal{N}^+, t \tag{5.45h}$$

$$p_{n,\phi}^d(t) = d_{n,\phi}(t) + \sum_{m \in \mathcal{M}_{n,\phi}} e_m(t), \forall\, \phi \in \mathcal{P}_n, n, t \tag{5.45i}$$

$$\mathbf{e}_m \in \mathcal{E}_m,\ \forall\, m \tag{5.45j}$$

$$(\mathbf{1}^\top \mathbf{P}_0(t))^2 + (\mathbf{1}^\top \mathbf{Q}_0(t))^2 \leq \bar{S}_f^2,\ \forall\, t. \tag{5.45k}$$

Constraints (5.45b)–(5.45d) originate from the power flow model; constraints (5.45e)–(5.45f) enforce generation limits; voltage regulation is guaranteed via (5.45g); apparent power flows are upper bounded by (5.45h); the equalities in (5.45i) define demands across phases and buses; constraint (5.45j) is related to the per-vehicle charging profile; and (5.45k) results from the capacity limit of the feeder transformer.

The cost functions and all the constraints apart from the EV charging constraint in (5.45j) are *separable across time*. The capacity limit in (5.45k) couples flows across phases, while the voltage regulation constraints in (5.45d) and (5.45g) couple variables across buses and phases. For linear and convex quadratic costs, problem (5.45) can be reformulated as a standard quadratically-constrained quadratic program and tackled by standard solvers in a centralized manner. Nonetheless, for increasing grid sizes, longer time horizons $\mathcal{T}$, and/or shorter control periods, tackling (5.45) could be challenging: Phase ϕ of bus n involves five variables $(v_{n,\phi}, p_{n,\phi}, q_{n,\phi}, P_{n,\phi}, Q_{n,\phi})$. Assuming for simplicity that every bus carries all three phases results in a total of $(15N + M)T$ variables. In addition, private information on a per-vehicle basis needs to be collected and processed by the utility. These considerations motivate well the scalable (both in space and time) and the privacy-preserving scheme for solving (5.45) that is pursued next.

5.3.2 ADMM-based Distributed Solver

This section delineates an ADMM-based method for decomposing (5.45) into smaller subproblems. Notably, each subproblem either enjoys a closed-form solution or it can be tackled efficiently by Algorithm 5 presented in Chapter 4. As a brief review, ADMM solves problems of

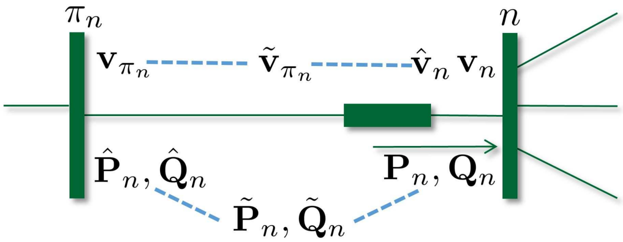

π_n
n
$\mathbf{v}_{\pi_n}$
$\tilde{\mathbf{v}}_{\pi_n}$
$\hat{\mathbf{v}}_n$
$\mathbf{v}_n$
$\hat{\mathbf{P}}_n, \hat{\mathbf{Q}}_n$
$\mathbf{P}_n, \mathbf{Q}_n$
$\tilde{\mathbf{P}}_n, \tilde{\mathbf{Q}}_n$

Figure 5.3: Consensus and duplicate variables in the ADMM scheme. Variables connected by blue dotted lines are constrained to be equal.

the form [14, 53]

$$\min_{\mathbf{x}\in\mathcal{X},\mathbf{z}\in\mathcal{Z}} \{f(\mathbf{x}) + g(\mathbf{z}) : \ \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{z} = \mathbf{b}\} \tag{5.46}$$

where $f(\mathbf{x})$ and $g(\mathbf{z})$ are convex functions; $\mathcal{X}$ and $\mathcal{Z}$ are convex sets; and $(\mathbf{F}, \mathbf{G}, \mathbf{b})$ model the linear equality constraints coupling variables $\mathbf{x}$ and $\mathbf{z}$. In its normalized form, ADMM assigns a Lagrange multiplier $\mathbf{w}$ for the equality constraint and solves (5.46) by iterating over the following three recursions

$$\mathbf{x}^{i+1} \in \arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) + \tfrac{\rho}{2}\|\mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{z}^i - \mathbf{b} + \mathbf{w}^i\|_2^2 \tag{5.47a}$$

$$\mathbf{z}^{i+1} \in \arg\min_{\mathbf{z}\in\mathcal{Z}} g(\mathbf{z}) + \tfrac{\rho}{2}\|\mathbf{F}\mathbf{x}^{i+1} + \mathbf{G}\mathbf{z} - \mathbf{b} + \mathbf{w}^i\|_2^2 \tag{5.47b}$$

$$\mathbf{w}^{i+1} := \mathbf{w}^i + \mathbf{F}\mathbf{x}^{i+1} + \mathbf{G}\mathbf{z}^{i+1} - \mathbf{b} \tag{5.47c}$$

for some $\rho > 0$. At iteration i, the primal and dual residual for (5.47) capturing primal and dual feasibility are defined as

$$\begin{aligned} o_p^i &:= \|\mathbf{F}\mathbf{x}^i + \mathbf{G}\mathbf{z}^i - \mathbf{b}\|_2 \\ o_d^i &:= \rho\|\mathbf{F}^\top\mathbf{G}(\mathbf{z}^i - \mathbf{z}^{i-1})\|_2. \end{aligned} \tag{5.48}$$

Under mild conditions, it has been shown that (o_p^i, o_d^i) converge to zero as the iteration index i goes to infinity, and that the objective function converges to the optimal value [14]. ADMM has been applied to decentralize various power system tasks [24], [62], [85]. Related ideas are adopted here to decouple the spatially-coupled constraints (5.45b)–(5.45d).

To that end, each bus $n \in \mathcal{N}$ maintains a local copy of the variables associated with the squared voltage magnitude of its parent bus, and the power flows feeding its children buses. These auxiliary variables are marked with a hat as $\hat{\mathbf{v}}_n$ and $\{(\hat{\mathbf{P}}_k, \hat{\mathbf{Q}}_k)\}_{k\in\mathcal{C}_n}$. The duplicate variable $\hat{\mathbf{v}}_n$ stored at bus n should agree with the original variable $\mathbf{v}_{\pi_n}$ stored at bus π_n as demonstrated in Fig. 5.3. To decentralize the computations, we further introduce the *consensus* variable $\tilde{\mathbf{v}}_{\pi_n}$, and impose the constraints $\mathbf{v}_{\pi_n} = \tilde{\mathbf{v}}_{\pi_n}$ and $\hat{\mathbf{v}}_n = \tilde{\mathbf{v}}_{\pi_n}$ for all non-leaf buses. The spatially-coupled equality constraints $\mathbf{v}_{\pi_n} = \tilde{\mathbf{v}}_{\pi_n}$ and $\hat{\mathbf{v}}_n = \tilde{\mathbf{v}}_{\pi_n}$ are dualized in the ADMM scheme. Consensus variables marked with a tilde are updated in the second step of ADMM, and all the remaining variables are updated in the first step of ADMM. By repeating this process for the power flow variables and for all $n \in \mathcal{N}$, the physical grid model is decoupled across buses.

We also introduce duplicate variables $\{\tilde{\mathbf{p}}_n^d(t)\}_{n\in\mathcal{N}}$ for net loads to separate the tasks of EV charging and generation dispatch. As detailed later, imposing the constraints $\tilde{\mathbf{p}}_n^d(t) = \mathbf{p}_n^d(t)$ for all n, enables isolating (5.45j) from the rest of the constraints in (5.45); resulting in localized EV charging subproblems that is a special case of (4.2).

For a compact representation define the aggregate variables:

$$\begin{aligned}
\mathbf{x}_n(t) &:= \Big\{\mathbf{v}_n(t), \mathbf{p}_n^g(t), \mathbf{p}_n^d(t), \mathbf{q}_n^g(t), \mathbf{P}_n(t), \mathbf{Q}_n(t)\Big\}\\
\hat{\mathbf{x}}_n(t) &:= \Big\{\hat{\mathbf{v}}_n(t), \{\hat{\mathbf{P}}_k(t), \hat{\mathbf{Q}}_k(t)\}_{k\in\mathcal{C}_n}\Big\}\\
\tilde{\mathbf{z}}_n(t) &:= \Big\{\tilde{\mathbf{v}}_n(t), \tilde{\mathbf{p}}_n^g(t), \tilde{\mathbf{p}}_n^d(t), \tilde{\mathbf{q}}_n^g(t), \tilde{\mathbf{P}}_n(t), \tilde{\mathbf{Q}}_n(t)\Big\}
\end{aligned}$$

for all $n \in \mathcal{N}$ and $t \in \mathcal{T}$. With the newly introduced variables, problem (5.45) can be equivalently expressed as:

$$\min \sum_{t\in\mathcal{T}} \left[f_0(\mathbf{P}_0(t)) + \sum_{n\in\mathcal{N}} f_n^g(\tilde{\mathbf{p}}_n^g(t)) \right] \tag{5.49a}$$

$$\text{over } \{\mathbf{x}_n(t), \hat{\mathbf{x}}_n(t), \tilde{\mathbf{z}}_n(t)\}_{n\in\mathcal{N}, t\in\mathcal{T}},\ \{\mathbf{e}_m \in \mathcal{E}_m\}_{m\in\mathcal{M}},$$

$$\text{s.to } \mathbf{p}_n^g(t) - \mathbf{p}_n^d(t) = \sum_{k\in\mathcal{C}_n} \hat{\mathbf{P}}_k(t) - \mathbf{P}_n(t), \forall\, n \in \mathcal{N}, t \tag{5.49b}$$

$$\mathbf{q}_n^g(t) - \mathbf{q}_n^d(t) = \sum_{k\in\mathcal{C}_n} \hat{\mathbf{Q}}_k(t) - \mathbf{Q}_n(t), \forall\, n \in \mathcal{N}, t \tag{5.49c}$$

$$\hat{\mathbf{v}}_n(t) - \mathbf{v}_n(t) = \mathrm{Re}\{\bar{\mathbf{Z}}_n(\mathbf{P}_n(t) + j\mathbf{Q}_n(t))\}$$

$$\forall n \in \mathcal{N}^+, t \tag{5.49d}$$

Table 5.1: Lagrange multipliers for problem (5.49)

$\mathbf{p}^g_n(t) = \tilde{\mathbf{p}}^g_n(t)$	$\boldsymbol{\lambda}^p_n(t)$	$\mathbf{q}^g_n(t) = \tilde{\mathbf{q}}^g_n(t)$	$\boldsymbol{\lambda}^q_n(t)$
$\hat{\mathbf{P}}_n(t) = \tilde{\mathbf{P}}_n(t)$	$\hat{\boldsymbol{\lambda}}^P_n(t)$	$\mathbf{P}_n(t) = \tilde{\mathbf{P}}_n(t)$	$\boldsymbol{\lambda}^P_n(t)$
$\hat{\mathbf{Q}}_n(t) = \tilde{\mathbf{Q}}_n(t)$	$\hat{\boldsymbol{\lambda}}^Q_n(t)$	$\mathbf{Q}_n(t) = \tilde{\mathbf{Q}}_n(t)$	$\boldsymbol{\lambda}^Q_n(t)$
$\hat{\mathbf{v}}_n(t) = \tilde{\mathbf{v}}_{\pi_n}(t)$	$\hat{\boldsymbol{\lambda}}^v_n(t)$	$\mathbf{v}_n(t) = \tilde{\mathbf{v}}_n(t)$	$\boldsymbol{\lambda}^v_n(t)$
$\mathbf{p}^d_n(t) = \tilde{\mathbf{p}}^d_n(t)$	$\boldsymbol{\lambda}^d_n(t)$	Constraints (5.49l)	$\mu_{n,\phi}(t)$

$$\underline{p}^g_{n,\phi} \leq \tilde{p}^g_{n,\phi}(t) \leq \bar{p}^g_{n,\phi},\ \forall\, \phi \in \mathcal{P}_n, n \in \mathcal{N}, t \tag{5.49e}$$

$$\underline{q}^g_{n,\phi} \leq \tilde{q}^g_{n,\phi}(t) \leq \bar{q}^g_{n,\phi},\ \forall\, \phi \in \mathcal{P}_n, n \in \mathcal{N}, t \tag{5.49f}$$

$$\underline{v}_n \leq \tilde{v}_{n,\phi}(t) \leq \bar{v}_n,\ \forall\, \phi \in \mathcal{P}_n, n \in \mathcal{N}, t \tag{5.49g}$$

$$\tilde{P}^2_{n,\phi}(t) + \tilde{Q}^2_{n,\phi}(t) \leq \bar{S}^2_n,\ \forall\, \phi \in \mathcal{P}_n, n \in \mathcal{N}_+, t \tag{5.49h}$$

$$\begin{aligned} &\mathbf{P}_n(t) = \tilde{\mathbf{P}}_n(t),\ \mathbf{Q}_n(t) = \tilde{\mathbf{Q}}_n(t),\ \mathbf{v}_n(t) = \tilde{\mathbf{v}}_n(t),\\ &\qquad \forall\, n \in \mathcal{N}^+, t \end{aligned} \tag{5.49i}$$

$$\begin{aligned} &\hat{\mathbf{P}}_n(t) = \tilde{\mathbf{P}}_n(t),\ \hat{\mathbf{Q}}_n(t) = \tilde{\mathbf{Q}}_n(t),\ \hat{\mathbf{v}}_n(t) = \tilde{\mathbf{v}}_{\pi_n}(t),\\ &\qquad \forall\, n \in \mathcal{N}^+, t \end{aligned} \tag{5.49j}$$

$$\begin{aligned} &\mathbf{p}^g_n(t) = \tilde{\mathbf{p}}^g_n(t), \mathbf{p}^d_n(t) = \tilde{\mathbf{p}}^d_n(t), \mathbf{q}^g_n(t) = \tilde{\mathbf{q}}^g_n(t),\\ &\qquad \forall\, n \in \mathcal{N}, t \end{aligned} \tag{5.49k}$$

$$\tilde{p}^d_{n,\phi}(t) = d_{n,\phi}(t) + \sum_{m \in \mathcal{M}_{n,\phi}} e_m(t),\ \forall \phi \in \mathcal{P}_n, n, t \tag{5.49l}$$

$$(\mathbf{1}^\top \tilde{\mathbf{P}}_0(t))^2 + (\mathbf{1}^\top \tilde{\mathbf{Q}}_0(t))^2 \leq \bar{S}^2_f,\ \forall t \tag{5.49m}$$

The equality constraints between duplicate variables in (5.49i)–(5.49l) are assigned Langrange multipliers according to Table 5.1. Adopting the ADMM iterates of (5.47) to solve (5.49), variables $\{\mathbf{x}_n(t), \hat{\mathbf{x}}_n(t)\}_{n \in \mathcal{N}, t \in \mathcal{T}}$ and $\{\mathbf{e}_m\}_{m \in \mathcal{M}}$ are updated in the first ADMM step, whereas variables $\{\{\tilde{\mathbf{z}}_n(t)\}_{n \in \mathcal{N}}\}_{t \in \mathcal{T}}$ are updated during the second ADMM step as detailed next.

The *first* step of our ADMM-based scheme is introduced next. Due to the form the generic update (5.47a) takes for the problem at hand, variables $\{\mathbf{x}_n(t), \hat{\mathbf{x}}_n(t)\}_{n \in \mathcal{N}, t \in \mathcal{T}}$ can be updated separately from the EV charging profiles $\{\mathbf{e}_m\}_{m \in \mathcal{M}}$. The updates for these two variable sets are studied next.

Heed that $\{\mathbf{x}_n(t), \hat{\mathbf{x}}_n(t)\}_{n \in \mathcal{N}, t \in \mathcal{T}}$ can be optimized independently across buses and time periods. Nevertheless, for fixed bus and time indices (n, t), variables $\mathbf{x}_n(t)$ and $\hat{\mathbf{x}}_n(t)$ are coupled due to constraints (5.49b)–(5.49d). To simplify the presentation, we drop the time index and consider the canonical subproblems involved for all $t \in \mathcal{T}$. Let $\hat{\mathbf{z}}_n := \left\{\tilde{\mathbf{v}}_{\pi_n}, \{\tilde{\mathbf{P}}_k, \tilde{\mathbf{Q}}_k\}_{k\in\mathcal{C}_n}\right\}$ for bus $n \in \mathcal{N}^+$. Variables $\mathbf{x}_n$ and $\hat{\mathbf{x}}_n$ are updated during the i-th iteration as the minimizers of

$$\begin{aligned}\min_{\mathbf{x}_n,\hat{\mathbf{x}}_n} \quad & \|\mathbf{x}_n - \mathbf{z}_n^i + \boldsymbol{\lambda}_n^i\|_2^2 + \|\hat{\mathbf{x}}_n - \hat{\mathbf{z}}_n^i + \hat{\boldsymbol{\lambda}}_n^i\|_2^2 \\ \text{s.to} \quad & \text{(5.49b) – (5.49d).}\end{aligned} \tag{5.50}$$

For $n = 0$ and due to the power supply cost from the main grid, variables $(\mathbf{P}_0, \mathbf{Q}_0)$ are found as the minimizers of

$$\begin{aligned}\min_{\mathbf{P}_0,\mathbf{Q}_0} \quad & \|\mathbf{x}_0 - \mathbf{z}_0^i + \boldsymbol{\lambda}_0^i\|_2^2 + \|\hat{\mathbf{x}}_0 - \hat{\mathbf{z}}_0^i + \hat{\boldsymbol{\lambda}}_0^i\|_2^2 + \frac{2}{\rho} f_0(\mathbf{P}_0) \\ \text{s.to} \quad & \text{(5.49b) – (5.49c).}\end{aligned} \tag{5.51}$$

Problems (5.50)–(5.51) are linearly-constrained quadratic programs with closed-form minimizers [15].

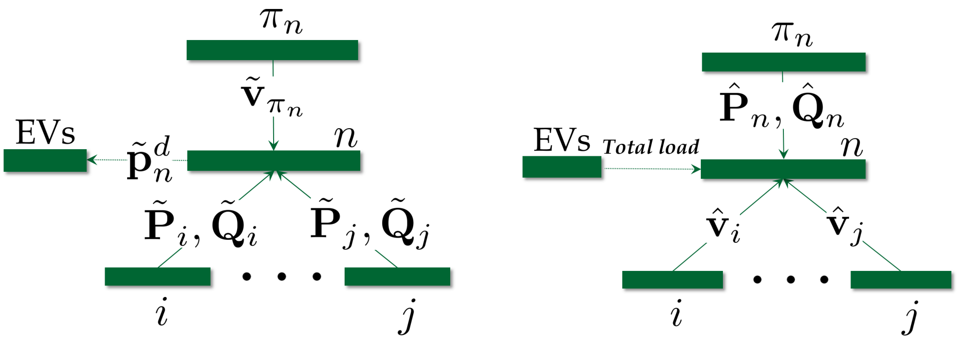
We next focus on updating the vehicle charging profiles $\{\mathbf{e}_m\}_{m\in\mathcal{M}}$ at iteration i. Interestingly, the task of EV charging decouples over buses and phases. The charging profiles for vehicles $m \in \mathcal{M}_{n,\phi}$ can be updated as the minimizers of

$$\min_{\{\mathbf{e}_m\in\mathcal{E}_m\}_{m\in\mathcal{M}_{n,\phi}}} \frac{1}{2}\sum_{t\in\mathcal{T}} \left(l_{n,\phi}^i(t) + \sum_{m\in\mathcal{M}_{n,\phi}} e_m(t) \right)^2 \tag{5.52}$$

where $l_{n,\phi}^i(t) := d_{n,\phi}(t) - \tilde{p}_{n,\phi}^{d,i}(t) + \mu_{n,\phi}^i(t)$ and the Lagrange multiplier $\mu_{n,\phi}^i(t)$ reflects the network constraints. Note (5.52) is actually a special case of (4.2) with $C_t(x) = x^2/2$, $\forall t$. Hence, subproblem (5.52) can be solved using Algorithm 5 introduced in Section 4.2.

In the first step of ADMM, each bus n needs to collect $\tilde{\mathbf{v}}_{\pi_n}$ from its parent and $\{\tilde{\mathbf{P}}_k, \tilde{\mathbf{Q}}_k\}_{k\in\mathcal{C}_n}$ from all its children as depicted in Fig. 5.4a. Meanwhile, each bus n transfers $\tilde{\mathbf{p}}_n^d$ to its EV scheduling center, where the charging profile of EVs are optimized using Algorithm 5.

The *second* step of the ADMM-based scheme is subsequently examined. Finding optimal $\{\tilde{\mathbf{z}}_n(t)\}_{n\in\mathcal{N},t\in\mathcal{T}}$ can be performed independently across buses and time slots. Because of that, time indices are ignored. Every bus n has to solve five sub-problems in parallel, each one associated with the variables $\tilde{\mathbf{p}}_n^g$, $\tilde{\mathbf{p}}_n^d$, $\tilde{\mathbf{q}}_n^g$, $\tilde{\mathbf{v}}_n$, and $(\tilde{\mathbf{P}}_n, \tilde{\mathbf{Q}}_n)$. Firstly, updating $\tilde{\mathbf{p}}_n^g$, $\tilde{\mathbf{p}}_n^d$, $\tilde{\mathbf{q}}_n^g$, and $\tilde{\mathbf{v}}_n$ decouples over phases of bus n too. It can be shown that per phase variables are updated


π_n
$\tilde{\mathbf{v}}_{\pi_n}$
EVs
$\tilde{\mathbf{p}}_n^d$
n
$\tilde{\mathbf{P}}_i, \tilde{\mathbf{Q}}_i$
$\tilde{\mathbf{P}}_j, \tilde{\mathbf{Q}}_j$
i
j
π_n
$\hat{\mathbf{P}}_n, \hat{\mathbf{Q}}_n$
EVs
Total load
n
$\hat{\mathbf{v}}_i$
$\hat{\mathbf{v}}_j$
i
j

(a) First step of ADMM.

(b) Second step of ADMM.

Figure 5.4: Information exchange in the ADMM steps for bus n.

as the minimizers of a univariate convex quadratic function possibly over box constraints. If the generation cost at bus n is $f_n^g(\tilde{\mathbf{p}}_n^g) := \sum_{\phi \in \mathcal{P}_n} a_{n,\phi}(p_{n,\phi}^g)^2 + b_{n,\phi}p_{n,\phi}^g + c_{n,\phi}$ with $a_{n,\phi} \geq 0$, then $\tilde{p}_{n,\phi}^g$ is updated at iteration i by solving

$$\begin{aligned} \min_{\tilde{p}_{n,\phi}^g} \quad & a_{n,\phi}(\tilde{p}_{n,\phi}^g)^2 + b_{n,\phi}\tilde{p}_{n,\phi}^g + \tfrac{\rho}{2}(p_{n,\phi}^{g,i} - \tilde{p}_{n,\phi}^g + \lambda_{n,\phi}^{p,i})^2 \\ \text{s.to} \quad & \underline{p}_{n,\phi}^g \leq \tilde{p}_{n,\phi}^g \leq \bar{p}_{n,\phi}^g. \end{aligned} \tag{5.53}$$

The minimizer of (5.53) is expressed as

$$\tilde{p}_{n,\phi}^{g,i+1} = \left[\frac{\rho(p_{n,\phi}^{g,i} + \lambda_{n,\phi}^{p,i}) - b_{n,\phi}}{2a_{n,\phi} + \rho}\right]_{\underline{p}_{n,\phi}^g}^{\bar{p}_{n,\phi}^g} \tag{5.54}$$

where $[x]_{\underline{x}}^{\bar{x}} := \max\{\underline{x}, \min\{x, \bar{x}\}\}$. The entries of $\tilde{\mathbf{q}}_n^g$ and $\tilde{\mathbf{v}}_n$ are similarly found as

$$\tilde{q}_{n,\phi}^{g,i+1} = \left[q_{n,\phi}^{g,i} + \lambda_{n,\phi}^{q,i}\right]_{\underline{q}_{n,\phi}^g}^{\bar{q}_{n,\phi}^g} \tag{5.55}$$

$$\tilde{v}_{n,\phi}^{i+1} = \left[\frac{\sum_{k \in \mathcal{C}_n}(\hat{v}_{k,\phi}^i + \hat{\lambda}_{k,\phi}^{v,i}) + v_{n,\phi}^i + \lambda_{n,\phi}^{v,i}}{|\mathcal{C}_n| + 1}\right]_{\underline{v}_n}^{\bar{v}_n}. \tag{5.56}$$

The entries of $\tilde{\mathbf{p}}_n^d$ are obtained as the solutions of unconstrained univariate convex quadratic programs as

$$\tilde{p}_{n,\phi}^{d,i+1} = \frac{1}{2}\left(p_{n,\phi}^{d,i} + \lambda_{n,\phi}^{d,i} + d_{n,\phi} + \sum_{m\in\mathcal{M}_{n,\phi}} e_m^i + \mu_{n,\phi}^i\right). \tag{5.57}$$

The optimizations involved in updating the consensus power flow variables $\{\tilde{\mathbf{P}}_n, \tilde{\mathbf{Q}}_n\}_{n\in\mathcal{N}^+}$ decouple across phases. The consensus power flow variables $\{(\tilde{P}_{n,\phi}, \tilde{Q}_{n,\phi})\}_{\phi\in\mathcal{P}_n, n\in\mathcal{N}^+}$ are updated by solving the problems for all $\phi\in\mathcal{P}_n$ and $n\in\mathcal{N}^+$:

$$\begin{aligned} \min_{\tilde{P}_{n,\phi},\tilde{Q}_{n,\phi}} \quad & (\tilde{P}_{n,\phi} - \breve{P}_{n,\phi}^i)^2 + (\tilde{Q}_{n,\phi} - \breve{Q}_{n,\phi}^i)^2 \\ \text{s.to} \quad & \tilde{P}_{n,\phi}^2 + \tilde{Q}_{n,\phi}^2 \leq \bar{S}_n^2 \end{aligned} \tag{5.58}$$

where $\breve{P}_{n,\phi}^i := \frac{1}{2}(P_{n,\phi}^i + \hat{P}_{n,\phi}^i + \lambda_{n,\phi}^{P,i} + \hat{\lambda}_{n,\phi}^{P,i})$, and $\breve{Q}_{n,\phi}^i := \frac{1}{2}(Q_{n,\phi}^i + \hat{Q}_{n,\phi}^i + \lambda_{n,\phi}^{Q,i} + \hat{\lambda}_{n,\phi}^{Q,i})$. Resorting to the KKT conditions for (5.58) shows that its minimizers are

$$\tilde{P}_{n,\phi}^{i+1} := \min\left\{\frac{\bar{S}_n}{\sqrt{(\breve{P}_{n,\phi}^i)^2 + (\breve{Q}_{n,\phi}^i)^2}}, 1\right\}\breve{P}_{n,\phi}^i \tag{5.59a}$$

$$\tilde{Q}_{n,\phi}^{i+1} := \min\left\{\frac{\bar{S}_n}{\sqrt{(\breve{P}_{n,\phi}^i)^2 + (\breve{Q}_{n,\phi}^i)^2}}, 1\right\}\breve{Q}_{n,\phi}^i. \tag{5.59b}$$

The substation power flows are updated as the solution to

$$\begin{aligned} \min_{\tilde{\mathbf{P}}_0,\tilde{\mathbf{Q}}_0} \quad & \|\tilde{\mathbf{P}}_0 - \breve{\mathbf{P}}_0^i\|_2^2 + \|\tilde{\mathbf{Q}}_0 - \breve{\mathbf{Q}}_0^i\|_2^2 \\ \text{s.to} \quad & (\mathbf{1}^\top\tilde{\mathbf{P}}_0)^2 + (\mathbf{1}^\top\tilde{\mathbf{Q}}_0)^2 \leq \bar{S}_f^2 \end{aligned} \tag{5.60}$$

where $\breve{\mathbf{P}}_0^i := \mathbf{P}_0^i + \boldsymbol{\lambda}_0^{P,i}$ and $\breve{\mathbf{Q}}_0^i := \mathbf{Q}_0^i + \boldsymbol{\lambda}_0^{Q,i}$. The minimizers of (5.60) can be computed in closed-form as shown in the ensuing proposition.

Proposition 8. *The optimal solution of problem* (5.60) *is*

$$\tilde{\mathbf{P}}_0^{i+1} := \breve{\mathbf{P}}_0^i - \max\left\{1 - \frac{\bar{S}_f}{\Sigma}, 0\right\}\frac{\mathbf{1}\mathbf{1}^\top\breve{\mathbf{P}}_0^i}{3} \tag{5.61a}$$

$$\tilde{\mathbf{Q}}_0^{i+1} := \breve{\mathbf{Q}}_0^i - \max\left\{1 - \frac{\bar{S}_f}{\Sigma}, 0\right\}\frac{\mathbf{1}\mathbf{1}^\top\breve{\mathbf{Q}}_0^i}{3} \tag{5.61b}$$

Algorithm 9 Decentralized multi-period microgrid energy management

```
Initialize $\mathbf{e}_m^0 = \mathbf{0}$ for all $m$
for $i = 0, 1, \ldots$ do (parallel across all buses)
    Bus $n$ solves (5.50) or (5.51) to obtain $\mathbf{x}_n^{i+1}$ and $\hat{\mathbf{x}}_n^{i+1}$
    Bus $n$ solves (5.52) via Algorithm 5 to update $\mathbf{e}_m^{i+1}$ for $m \in \mathcal{M}_n$
    Bus $n$ updates $\tilde{\mathbf{z}}_n^{i+1}$ from (5.53)–(5.57), (5.59), and (5.61)
    Bus $n$ updates Lagrange multipliers by (5.47c)
end for
```

where $\Sigma := \sqrt{(\mathbf{1}^\top \breve{\mathbf{P}}_0^i)^2 + (\mathbf{1}^\top \breve{\mathbf{Q}}_0^i)^2}$.

Proof of Prop. 8. The Lagrangian function of the convex problem in (5.60) reads $L(\tilde{\mathbf{P}}_0, \tilde{\mathbf{Q}}_0, \nu) = \|\tilde{\mathbf{P}}_0 - \breve{\mathbf{P}}_0^i\|_2^2 + \|\tilde{\mathbf{Q}}_0 - \breve{\mathbf{Q}}_0^i\|_2^2 + \nu \left[(\mathbf{1}^\top \tilde{\mathbf{P}}_0)^2 + (\mathbf{1}^\top \tilde{\mathbf{Q}}_0)^2 - \bar{S}_f^2\right]$. Because (5.60) satisfies Slater's condition (e.g., for $\tilde{\mathbf{P}}_0 = \tilde{\mathbf{Q}}_0 = \mathbf{0}$), strong duality holds [15]. If $(\tilde{\mathbf{P}}_0^*, \tilde{\mathbf{Q}}_0^*, \nu^*)$ are the optimal primal/dual variables, Lagrangian optimality yields:

$$\tilde{\mathbf{P}}_0^* - \breve{\mathbf{P}}_0^i + \nu^* \mathbf{1}\mathbf{1}^\top \tilde{\mathbf{P}}_0^* = \mathbf{0} \tag{5.62a}$$

$$\tilde{\mathbf{Q}}_0^* - \breve{\mathbf{Q}}_0^i + \nu^* \mathbf{1}\mathbf{1}^\top \tilde{\mathbf{Q}}_0 = \mathbf{0}. \tag{5.62b}$$

Premultiplying both sides of (5.62a)–(5.62b) by $\mathbf{1}^\top$ results in:

$$\mathbf{1}^\top \tilde{\mathbf{P}}_0^* = \frac{\mathbf{1}^\top \breve{\mathbf{P}}_0^i}{1 + 3\nu^*} \quad \text{and} \quad \mathbf{1}^\top \tilde{\mathbf{Q}}_0^* = \frac{\mathbf{1}^\top \breve{\mathbf{Q}}_0^i}{1 + 3\nu^*}. \tag{5.63}$$

Complementary slackness yields $\nu^* \left[(\mathbf{1}^\top \tilde{\mathbf{P}}_0^*)^2 + (\mathbf{1}^\top \tilde{\mathbf{Q}}_0^*)^2 - \bar{S}_f^2\right] = 0$, which from (5.63) and dual feasibility provides

$$\nu^* = 0 \text{ or } \nu^* = \frac{1}{3}\left(\sqrt{(\mathbf{1}^\top \breve{\mathbf{P}}_0^i)^2 + (\mathbf{1}^\top \breve{\mathbf{Q}}_0^i)^2} / \bar{S}_f - 1\right). \tag{5.64}$$

The claim follows from primal feasibility, (5.62), and (5.64). □

To implement the second step of ADMM, bus n gathers its copies $(\hat{\mathbf{P}}_n, \hat{\mathbf{Q}}_n)$ from its parent, $\{\hat{v}_k\}_{k \in \mathcal{C}_n}$ from all its children, and the total charging load $\{\sum_{m \in \mathcal{M}_{n,\phi}} \mathbf{e}_m^k\}_{\phi \in \mathcal{P}_n}$ of all the connected EVs as presented in Fig. 5.4b. Then bus n updates $\tilde{\mathbf{z}}_n$ according to (5.53)–(5.57), (5.59), and (5.61).

In the *third* step of our ADMM-based scheme, the Lagrange multipliers are updated according to (5.47c), i.e., every multiplier is equal to its previous value plus the most recent constraint violation. The overall information exchange for the devised algorithm is depicted in Fig. 5.4, while the charging protocol is summarized as Algorithm 9.

5.4 Numerical Tests

Performance of our local reactive power control protocols, and ADMM-based distributed micro-grid energy management schemes was evaluated using the IEEE benchmark systems.

5.4.1 Simulations for Local Reactive Power Control

The voltage regulation schemes presented earlier were evaluated using the IEEE 13-bus feeder, the IEEE 123-bus feeder, and a Southern California Edison (SCE) 47-bus feeder [39], [104]. Solar generation data from the Smart* project and from August 24, 2011 were used [8]. Unless otherwise stated, active power injections were kept fixed over the reactive control period, and PV reactive injections were initialized to zero. Single-phase grids were obtained upon modifying multi-phase grids as in [67]. The global minimizer of (5.18) was obtained using MATLAB and it served as a benchmark. The first experiment simulated an over-voltage scenario on the IEEE 13-bus feeder. The IEEE 13-bus grid exhibits $\kappa(\mathbf{X}) = 716$. Simulating a severe over-voltage violation in the transmission network, the feeder voltage magnitude was set to 1.07 p.u., while the voltage regulator was removed from the system. A 52% PV penetration level was assumed for all buses apart from buses 4, 8, 9, and 12. Loads were fixed to 80% of their peak value, and reactive costs were set to $c_n = 0.0125\text{cent}$/kVar& h for all n.

Three reactive power control rules were tested: (i) the proximal gradient method (PGM) of (5.28); (ii) the proximal gradient method with diagonal scaling (DPGM) in (5.32); and the accelerated proximal gradient method (APGM) of (5.33). The squared voltage magnitudes obtained at three buses are illustrated in Fig. 5.5. For all control rules, the step size was conservatively set to $\mu = 0.1/\lambda_{\max}(\mathbf{X})$. Figure 5.5 demonstrates that APGM has an obvious four-fold speedup advantage over PGM and DPGM. Diagonal scaling does not exhibit any convergence rate advantage over PGM. The latter could be attributed to the fact that the diagonal entries of $\mathbf{X}$ have similar values; hence, matrices $\mathbf{X}$ and $\mathbf{D}^{1/2}\mathbf{X}\mathbf{D}^{1/2}$ with $\mathbf{D} = \text{diag}(\text{diag}(\mathbf{X}))$ have similar condition numbers. On a different note and as expected, the minimizer of (5.26) does not guarantee small voltage magnitude deviations across all buses: Buses 12-13 exhibit

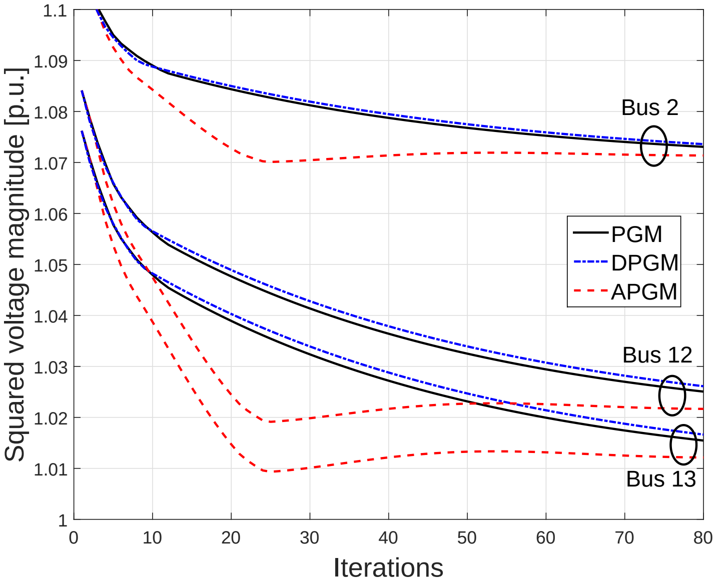


Figure 5.5: Voltages for buses 2, 12, and 13, in the IEEE 13-bus feeder.

small deviations, but the voltage magnitude of bus 2 converges to 1.03.

The second experiment involved the SCE 47-bus grid with $\kappa(\mathbf{X}) = 16,470$ [39]. The capacitor located on bus 32 was ignored. Loads were set to 80% of their peak value with a power factor of 0.8. Five PV generators generating 60% of their capacity were located on buses 13, 17, 19, 23, 24. Distributed PVs with 50% penetration level were further installed on buses 11, 12, 14, 22, 25, 33, 38, 39, 41. The relative cost value error attained by (5.26) using the optimal $\mu = 1/\lambda_{\max}(\mathbf{X})$ is depicted in Fig. 5.6. Apparently, the novel scheme converged at least six to ten times faster than its alternatives.

We further tested the accuracy of the linearized model over the full AC model calculated using the forward-backward algorithm [68]. Voltage magnitudes obtained from PGM and APGM with $\mu = 0.1/\lambda_{\max}(\mathbf{X})$ are presented in Fig. 5.7. The curves suggest that the linearized model of (5.3) offers a good approximation.

To evaluate the control schemes on more realistic conditions, the third experiment used the IEEE 123-bus grid and it also entailed a topology reconfiguration: After twenty control periods (algorithm iterations), the switch between buses 97 and 197 opened, while the switch between buses 151 and 300 closed. Renewable (PV) generation units were located on buses 32, 51, 64, 76, 80, 93, and 114, with capacities 60, 60, 120, 80, 30, 100, and 80 kVA, respectively. The

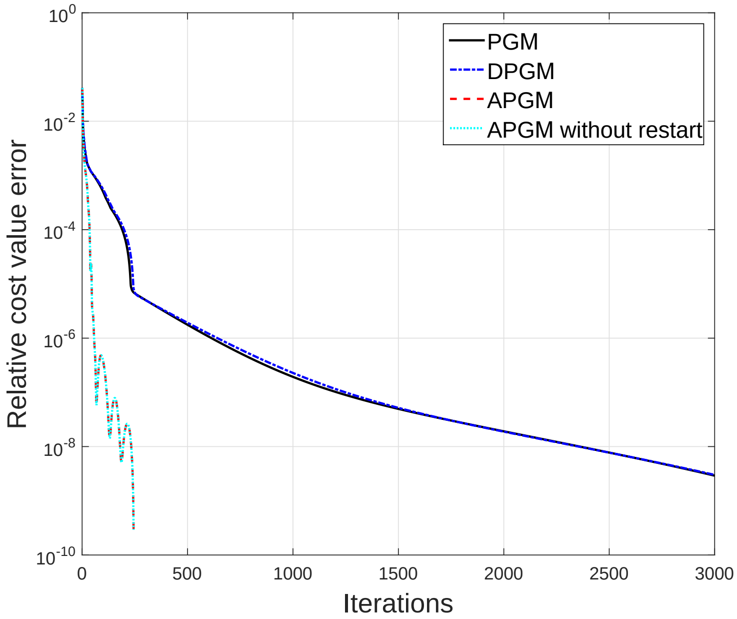


Figure 5.6: Convergence performance for the SCE 47-bus using the optimal μ.

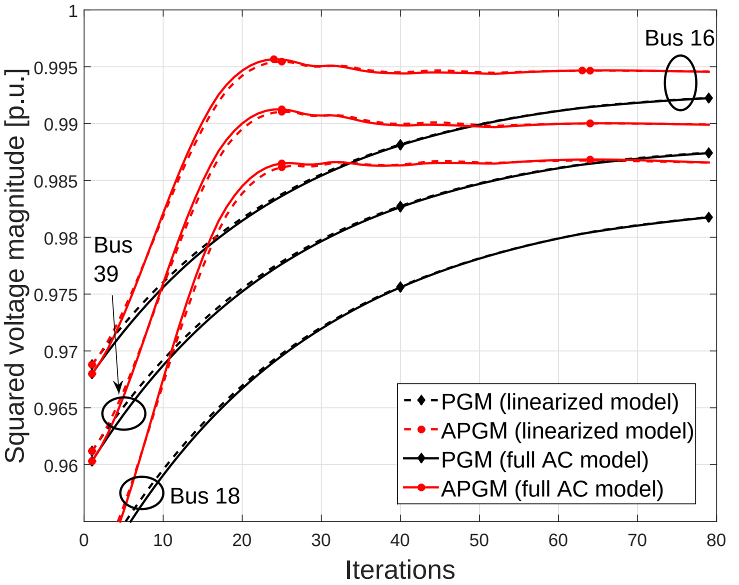


Figure 5.7: Voltages for buses 16, 18, and 39, for the SCE 47-bus feeder.

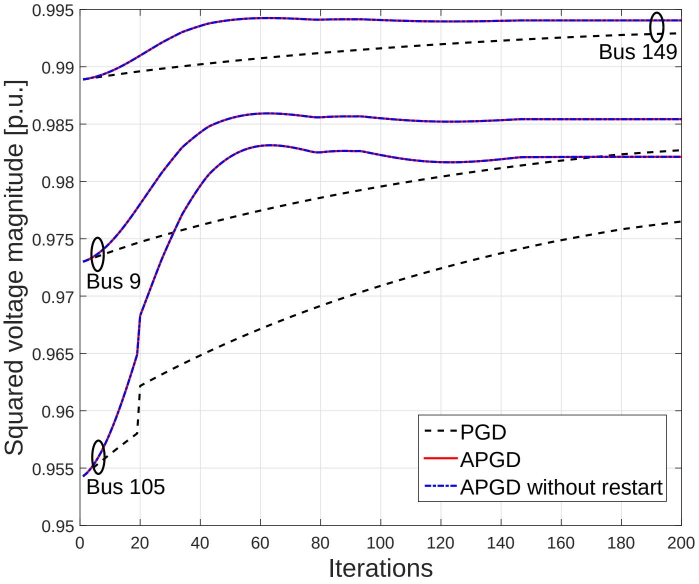


Figure 5.8: Voltages for buses 9, 105, and 149, for the IEEE 123-bus feeder.

condition number for this feeder is $\kappa(\mathbf{X}) = 20,677$. Figure 5.8 presents the squared voltage magnitudes over three representative buses and for $\mu = 0.1/\lambda_{\max}(\mathbf{X})$. Three observations are in order. First, note that the convergence guarantees for all three control schemes hold for any feasible initialization point. Theoretically, a topology change could potentially delay APGM since the parameters β_t in (5.33) are time-increasing. According to the curves, the topology change does not affect significantly the convergence rate of any of the algorithms. Second, the APGM scheme exhibits superior convergence properties over the PGM rule. Third, compared to the experiments on the IEEE 13-bus and the SCE 47-bus feeders, the longer convergence period can be attributed to the larger size of the feeder.

To evaluate the findings of Section 5.2.3, the control rule described by the iterations in (5.42) was tested on the multi-phase IEEE 13-bus system. Loads and PV penetration were selected as in the single-phase experiment presented earlier. The feeder voltage magnitude was fixed to 1, while the step size was set to $\mu = \lambda_{\min}\left(\mathbf{\Lambda}^{-1/2}\mathbf{U}^\top(\mathbf{X}+\mathbf{X}^\top)\mathbf{U}\mathbf{\Lambda}^{-1/2}\right)$. Squared voltage magnitude profiles obtained from (5.42) are plotted in Fig. 5.9. Voltages were calculated using both the linearized model and the full AC model at every iteration. Verifying the findings of [40], the curves indicate that the approximation is quite precise. The control rule converged within 40 iterations. Tests conducted with smaller step sizes exhibited slower convergence, whereas the

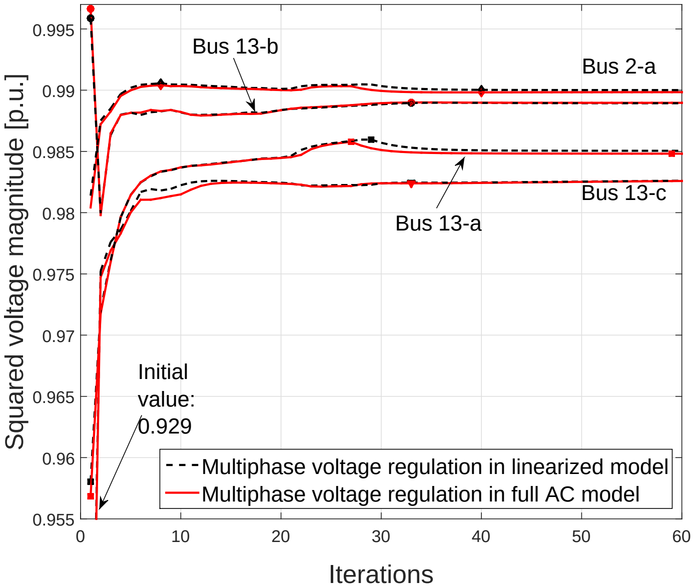


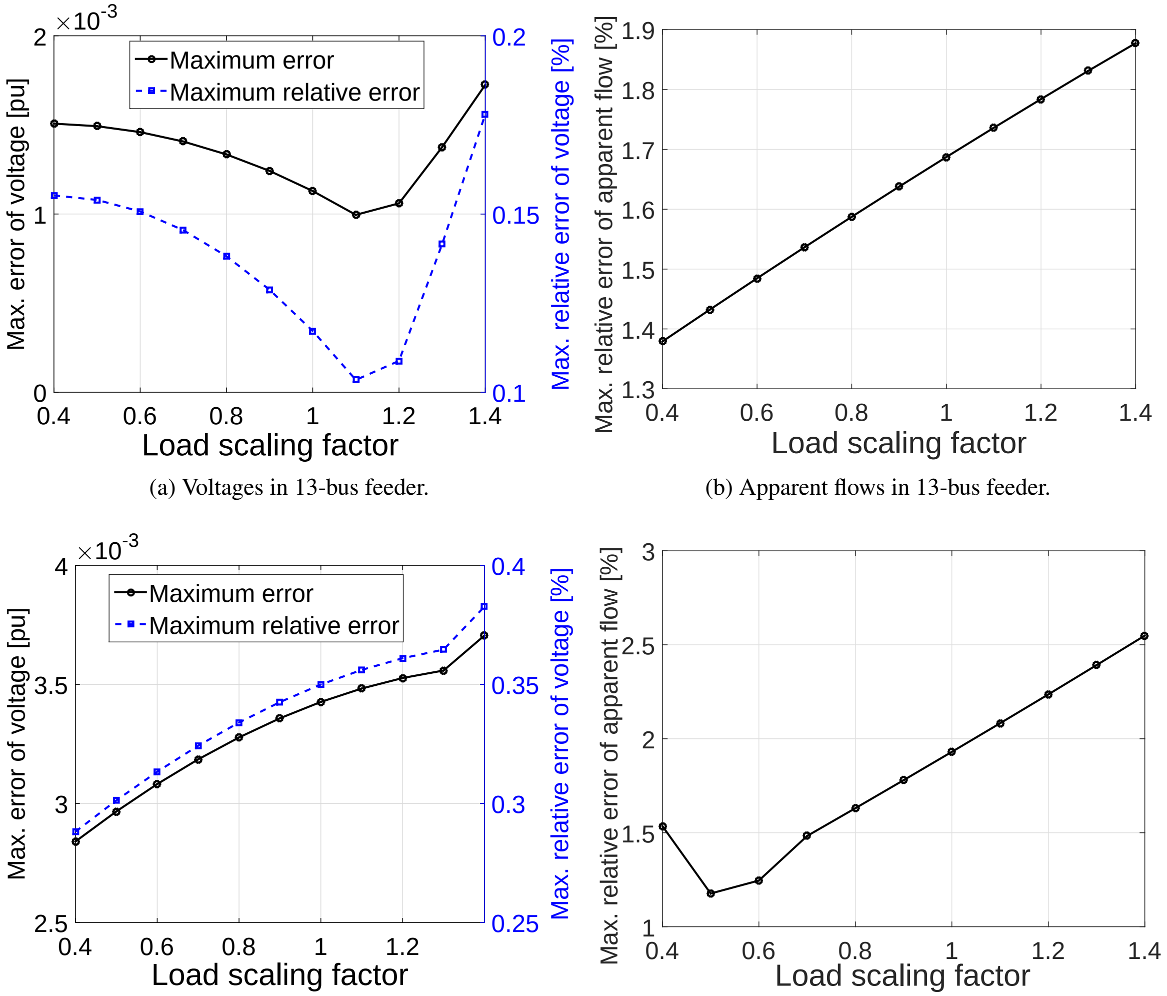
Figure 5.9: Voltage profiles obtained using (5.42) in the original IEEE 13-bus grid.

scheme diverged for a step size $\mu \geq 3.1\lambda_{\min}\left(\mathbf{\Lambda}^{-1/2}\mathbf{U}^{\top}(\mathbf{X}+\mathbf{X}^{\top})\mathbf{U}\mathbf{\Lambda}^{-1/2}\right)$.

Finally, the approximation error incurred by the linearized multi-phase grid model in (5.8) was also examined. The full AC power flow model was solved using the forward-backward sweep algorithm [68]. The approximation error was numerically evaluated by varying active loads from 0.4 to 1.4 times the original peak load while maintaining the power factor constant. The maximum error and the maximum relative error in voltage magnitudes over all phases and buses are shown in Figs.5.10a and 5.10c, respectively for the IEEE 13-bus and the 123-bus feeders. The voltage approximation error is less than $4 \cdot 10^{-3}$ pu. Figures 5.10b and 5.10d depict the maximum relative errors in apparent power flows, which are within 3%.

5.4.2 Simulations for Multi-period Energy Management

Algorithm 9 was first tested using the unbalanced IEEE 13-bus feeder [56]. Voltage magnitudes were constrained within $[\underline{v}, \bar{v}] = [0.97, 1.03]$ pu and the feeder voltage was set to 1 pu. The cost of energy drawn from the main grid was $f_0(\mathbf{P}_0) := \sum_{\phi} f_{0,\phi}(P_{0,\phi})$, where $f_{0,\phi}(P_{0,\phi}) := a_{0,\phi}P_{0,\phi}^2 + b_{0,\phi}P_{0,\phi}$, $a_{0,\phi} =$ 0.1$/(MW)^{2}h, and $b_{0,\phi} = 16$ \$/MWh for all three phases ϕ. Two distributed generators were located at buses 2 and 13. Their generation costs had the same



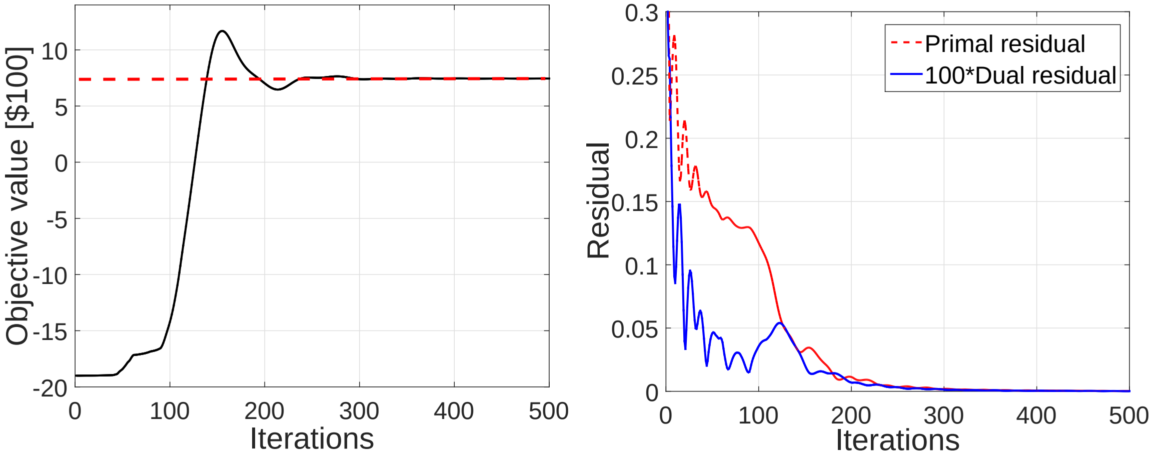
(a) Voltages in 13-bus feeder.

(b) Apparent flows in 13-bus feeder.

(c) Voltages in 123-bus feeder.

(d) Apparent flows in 123-bus feeder.

Figure 5.10: Approximation error between the full AC and the linearized power flow models for increasing loading factor.



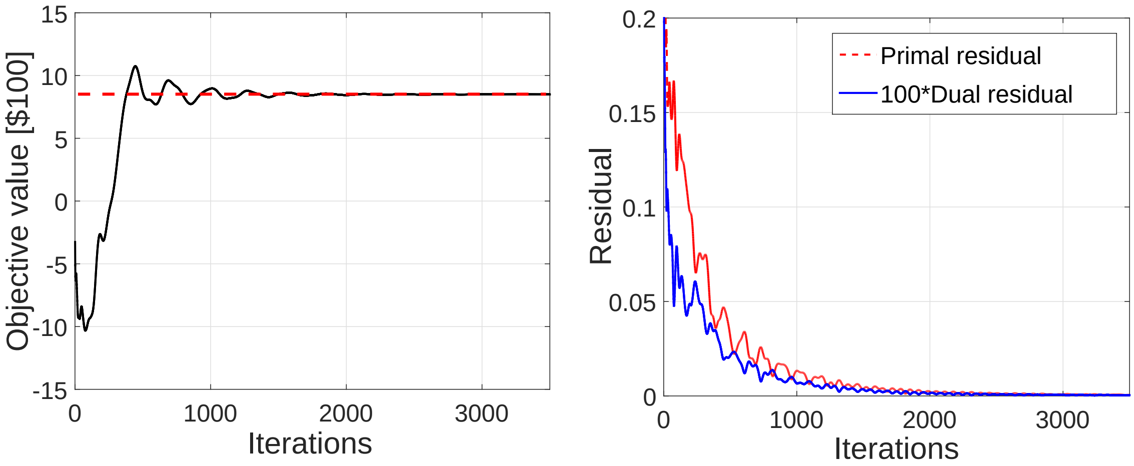
(a) Convergence of objective value.

(b) Primal/dual residual convergence.

Figure 5.11: Convergence performance for the IEEE 13-bus feeder.

functional form as $f_0(\mathbf{P}_0)$ with coefficients $a_{n,\phi} = 0.1$ $/(MW)^{2}h, and $b_{n,\phi} = 8$ $/MWh for all three phases ϕ and $n = 2, 13$. Forty eight EVs were being charged on phases b and c of bus 6 and phase a of bus 13 (16 EVs per location) resulting in a total of 23,328 variables for $T = 96$ time slots. SeDuMi and Algorithm 9 converged to the optimal cost of $744.76 within 454 sec and 4.76 sec, respectively. The convergence of the objective in (5.45) shown in Fig. 5.11a and the evolution of $o_p^i/T\sqrt{N}$ and $o_d^i/T\sqrt{N}$ shown in Fig. 5.11b demonstrate that Algorithm 9 converged within 350 iterations. Elaborating on the running time, the average running time for solving the vehicle scheduling subproblem (5.52) using Algorithm 5 was $8.6 \cdot 10^{-3}$ sec, and the average time spent on the remaining ADMM updates was $5 \cdot 10^{-3}$ sec. The overall computational efficiency of Algorithm 9 is due to the closed-form updates and the use of Algorithm 5 to tackle (5.52).

To evaluate the scalability of Algorithm 9, vehicle numbers were increased to 24 on phase b of bus 6, 24 on phase c of bus 6, and 20 on phase a of bus 13. For this setup, SeDuMi and Algorithm 9 reached the optimal cost of $748.98 in 713 sec and 4.80 sec, respectively. The average running time for tackling (5.52) using Algorithm 5 took $8.7 \cdot 10^{-3}$ sec, and the average time spent on the remaining ADMM updates was $5 \cdot 10^{-3}$ sec as before. For a total of 350 iterations, the total running time for Algorithm 9 was 4.8 sec, slightly larger than the previous case.



(a) Objective value convergence.

(b) Primal/dual residual convergence.

Figure 5.12: Convergence performance for the IEEE 123-bus feeder.

Due to the flexibility of EV loads to be served at night, no voltage constraint was active during the previous experiments. To test the effect of network constraints, the generation cost at bus 13 was increased by setting $a_{13,\phi} = 2\$/(\text{MW})^2\text{h}$, and $b_{13,\phi} = 16\ \$/\text{MWh}$ for all three phases ϕ. Moreover, the capacitor at the same bus was removed. The result was additional active power flowing from the feeder bus thus causing undervoltage scenarios. Sixteen vehicles were assumed on phases b and c of bus 6 and phase a of bus 13. SeDuMi and Algorithm 9 converged to the optimal cost of $881.44 in 562 sec and 12 sec (700 iterations), respectively. The voltage magnitude on phase c of bus 13 was reaching the lower limit of 0.97 pu from time slot 6 to 38. Without considering network constraints, the voltage magnitude on phase c of bus 13 would drop to 0.96 pu, and the total cost would decrease to $861.15.

Algorithm 9 was also tested on the larger unbalanced IEEE 123-bus feeder [56]. Fifteen DG units were placed in the system; while 5, 10, 15, 25 and 5 vehicles were being charged on buses 3, 15, 64, 82, and 102, respectively. In this feeder scenario, SeDuMi was unable to handle the 182,880 variables involved over $T = 96$ slots. The cost convergence and the convergence of $o_p^i/T\sqrt{N}$ and $o_d^i/T\sqrt{N}$ are shown in Fig. 5.12. The optimum cost of $849.43 was attained in 2,000 iterations. The average running times for (5.52) and for the remaining ADMM subproblems were $8.4 \cdot 10^{-3}$ sec and $4.7 \cdot 10^{-3}$ sec, respectively, resulting in 26 sec for Algorithm 9 to converge.

Chapter 6

Summary and Future Directions

Leveraging recent advances in machine learning and large-scale optimization, the present thesis contributes to the development of a scalable computational framework for learning power system states and topologies, coordinating EV load controllers, and managing spatiotemporally distributed microgrid resources. To conclude this thesis, a summary of its main results and possible directions for future research are provided in this final chapter.

6.1 Thesis Summary

Chapter 2 dealt with real-time power system monitoring (estimation and forecasting) by building on data-driven DNN advances. Prox-linear nets were developed for PSSE, that combine NNs with traditional physics-based optimization approaches. Deep RNNs were also introduced for power system state forecasting from historical (estimated) voltages. Our model-specific prox-linear net based PSSE is easy-to-train, and computationally inexpensive. The proposed RNN-based forecasting accounts for the long-term nonlinear dependencies in the voltage time-series, enhances PSSE, and offers situational awareness ahead of time. Numerical tests on the IEEE 57- and 118-bus benchmark systems using real load data illustrate the merits of our developed approaches relative to existing alternatives.

Turning attention to topology monitoring, Chapter 3 advocated topology identification of meshed grids by accounting for nonlinear relationships present among time-courses of power-related quantities measured at different buses. These dependencies were captured through nonlinear partial correlations that rely on the multi-kernel based nonlinear functions of the aforementioned time-courses obtained via ℓ_p-norm regularized regressions. For efficient implementation, the projection-free Frank-Wolfe iterations were invoked to solve the emerging ℓ_p-norm constrained optimization problems. The novel solver features closed-form updates and guaranteed fast convergence. The generalized nonlinear model was subsequently applied to the power grid topology inference application. Simulated tests on the IEEE 14-bus benchmark showcase improved computational efficiency of the projection-free solver, and increased identification accuracy of the proposed grid topology inference scheme.

Leveraging the obtained grid topology and states information, optimal load control and microgrid management schemes were designed. Given that optimal EV charging scales unfavorably with the fleet size and the number of control periods, a decentralized charging protocol based on the Frank-Wolfe method was developed in Chapter 4. Based on numerical tests, the novel charging protocol converged 100 times faster than a competing alternative, while its closed-form updates pose minimal computing requirements to vehicle controllers. To account for random cyber delays, an asynchronous variant was also devised based on our randomized block Frank-Wolfe method. This randomized block-coordinate protocol was shown to converge at sublinear rates upon using our judiciously devised step-size sequences.

To respect voltage limits, while coping with limited communication capabilities in microgrids, communication-free reactive power control schemes were developed in the first part of Chapter 5. For balanced distribution grids, localized (un)constrained schemes were analyzed, while a control scheme based on proximal gradient method was developed. Its computational complexity is comparable to that of the control rule suggested by the IEEE 1547.8 standard [55], yet it enjoys precise convergence guarantees. Adding memory to the control rule further yielded a significantly faster voltage regulation scheme. For unbalanced distribution grids, a linear approximation model revealed a counter-intuitive inter-phase coupling. Based on the properties of the involved matrices, the developed reactive power control rule converges to a well-defined equilibrium point. Numerical tests on benchmark feeders illustrate the superiority of the accelerated scheme over diagonal scaling, and the resilience of all novel schemes against topology reconfigurations.

To further incorporate active power control and enforce feeder transformer limits, microgrid energy management accounting for both DG units and adjustable EV loads was dealt with in the second part of Chapter 5. To achieve scalability, an ADMM-based solver was built leveraging an approximate grid model. The solver features closed-form updates and incorporates the vehicle charging protocol devised in Chapter 4. Numerical tests using real-world data verify the optimality and efficiency of the proposed decentralized schemes.

6.2 Future Research

The results in this thesis open up interesting directions for a number of future research topics, including those dealing with deep learning of grid topologies and online management of spatiotemporally distributed resources. Next, we briefly discuss a couple of topics per subject that we pursue currently.

- **Neural network based PC for topology identification.** In Chapter 3, MKL is adopted to select the optimal kernel(s) from the linear span of a dictionary of prespecified kernels [72]. Nonetheless, the MKL-based approach may not reliably give performance improvements over baseline methods that use simply the best single base kernel [72, 23]. In addition to kernels, NNs can also be employed to capture nonlinear relationships between data [46]. As demonstrated in Chapter 2, the DNN based approaches exhibit improved performance relative to the state-of-the-art methods in state estimation and forecasting. In the context of topology identification, we will develop NN-based PCs, which rely on a neural network to predict $\hat{\mathbf{x}}_{i|\backslash ij}$ based on $\{\mathbf{x}_n\}_{n\in\backslash ij}$. Furthermore, we will compare the performance of NN-based PCs with that of MKL-based PCs in terms of identification accuracy.

- **Multi-task DNNs for joint inference of grid topology and power system states.** For power grids with unknown breaker status [80], both topology identification and state estimation need to be performed to ensure the secure and economical operation of grids. In this context, joint grid topology and state vector inference, a.k.a., generalized state estimation (GSE), has been formulated as a mixed-integer nonlinear program [80]. However, conventional optimization-based GSE solvers including mixed-integer programming are time consuming [80, 65], which hampers running GSE in real time. Learning-based GSE via autoencoders has been explored in [82], where the grid topology is identified using an autoencoder-based classifier, while the states are estimated by a different NN. The DNNs therein are trained using historical data and simulated samples. However, such a scheme not only suffers from the inferior performance of autoencoders, but also overlooks the state-topology coupling. To enhance performance, our idea is to develop a multi-task DNN that performs classification and regression simultaneously, building on the prox-linear net developed in Chapter 2 and our recent work on image segmentation via multi-task DNNs [126]. The novel multi-task model will merit a shared DNN module that extracts features for two different tasks, namely identifying the topology and estimating the power system states.

- **Load balancing in multi-phase microgrids.** The problem (5.45) formulated in Chapter 5 dispatches (re)active power injections under common physical limitations of microgrids. Nonetheless, it does not enforce the phase load to be balanced. As pointed out in [115], the unbalanced charging loads on different phases will cause load imbalance at the substation, which is generally undesirable in power system operation, and may even cause motor damage from excessive heat. Although load balancing constraints are pertinent, they will complicate our problem formulation significantly. Specifically, enforcing power flows across phases to be approximately equal will render the problem nonconvex in the bidirectional power flow scenario. To cope with such nonconvexity, we will introduce auxiliary binary variables to reformulate the load balancing constraints as mixed-integer inequalities. Though the reformulated constraints are still nonconvex, our idea is to tackle them efficiently using Bender's decomposition [94, 119]. To derive scalable dispatch schemes, while respecting the nonconvex load balancing constraints, we will combine Bender's decomposition with our ADMM-Frank-Wolfe (FW) solver devised in Chapter 5. We further expect that such a hybrid ADMM-FW approach will have broader impact by offering an efficient solver for the subproblems arising from Bender's decomposition.

- **Online management under spatio-temporal constraints.** The developed ADMM-FW solver in Chapter 5 requires all DGs and EVs to negotiate with the scheduler at the beginning of a control horizon. However, this presumption may not be satisfied for cases in which the charging needs are revealed to the scheduler in an online manner. Without considering network constraints, an ad-hoc real-time EV scheduling scheme is devised in Chapter 4; see also [101] for centralized online vehicle charging. However, it is not clear how to generalize such schemes to (5.45) due to the presence of spatially coupled constraints (5.45b)–(5.45d) and (5.45i), as well as the temporally coupled constraints (5.45j). Accounting for the randomness associated with DGs and EV arrivals in (5.45), we will investigate optimal online implementations of the proposed ADMM-FW scheme [128] based on the online ADMM [112], and online FW [50] algorithms.

References

[1] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[2] A. Abur and M. K. Celik, “A fast algorithm for the weighted least-absolute-value state estimation (for power systems),” *IEEE Trans. Power Syst.*, vol. 6, no. 1, pp. 1–8, Feb. 1991.

[3] R. Baldick and F. Wu, “Efficient integer optimization algorithms for optimal coordination of capacitors and regulators,” *IEEE Trans. Power Syst.*, vol. 5, no. 3, pp. 805–812, Aug. 1990.

[4] M. Baran and F. Wu, “Network reconfiguration in distribution systems for loss reduction and load balancing,” *IEEE Trans. Power Syst.*, vol. 4, no. 2, pp. 1401–1407, Apr. 1989.

[5] ——, “Optimal capacitor placement on radial distribution systems,” *IEEE Trans. Power Syst.*, vol. 4, no. 1, pp. 725–734, Jan. 1989.

[6] ——, “Optimal sizing of capacitors placed on a radial distribution system,” *IEEE Trans. Power Syst.*, vol. 4, no. 1, pp. 735–743, Jan. 1989.

[7] P. N. P. Barbeiro, J. Krstulovic, H. Teixeira, J. Pereira, F. J. Soares, and J. P. Iria, “State estimation in distribution smart grids using autoencoders,” in *Proc. Intl. Power Eng. and Optimization Conf.*, Shah Alam, Malaysia, Mar. 2014, pp. 358–363.

[8] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, “Smart*: An open data set and tools for enabling research in sustainable homes,” in *Wkshp. on Data Mining Appl. in Sustain.*, Beijing, China, Aug. 2012.

[9] G. Benetti, M. Delfanti, T. Facchinetti, D. Falabretti, and M. Merlo, “Real-time modeling and control of electric vehicles charging processes,” *IEEE Trans. Smart Grid*, vol. 6, no. 3, pp. 1375–1385, May 2015.

[10] D. P. Bertsekas, *Convex Optimization Algorithms*. Belmont, MA: Athena Scientific, 2015.

[11] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computing: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.

[12] S. Bolognani and L. Schenato, “Identification of power distribution network topology via voltage correlation analysis,” in *Proc. IEEE Conf. on Decision and Control*, Florence, Italy, Dec. 2013.

[13] S. Bolognani and S. Zampieri, “On the existence and linear approximation of the power flow solution in power distribution networks,” *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 163–172, Jan. 2016.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learning*, vol. 3, pp. 1–122, Jul. 2010.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY: Cambridge University Press, 2004.

[16] J. V. Burke and M. C. Ferris, “A Gauss-Newton method for convex composite optimization,” *Mathematical Programming*, vol. 71, no. 2, pp. 179–194, Dec. 1995.

[17] Y. Cao, S. Tang, C. Li, P. Zhang, Y. Tan, Z. Zhang, and J. Li, “An optimized EV charging model considering TOU price and SOC curve,” *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 388–393, Mar. 2012.

[18] P. M. S. Carvalho, P. F. Correia, and L. A. Ferreira, “Distributed reactive power generation control for voltage rise mitigation in distribution networks,” *IEEE Trans. Power Syst.*, vol. 23, no. 2, pp. 766–772, May 2008.

[19] G. Cavraro, R. Arghandeh, K. Poolla, and A. Meier, “Data-driven approach for distribution network topology detection,” in *Proc. IEEE PES Soc. General Meeting*, Denver, CO, Jul. 2015.

[20] N. Chen, C. W. Tan, and T. Quek, “Electric vehicle charging in smart grid: Optimality and valley-filling algorithms,” *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 6, pp. 1073–1083, Dec. 2014.

[21] K. L. Clarkson, “Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm,” *ACM Trans. Algorithms*, vol. 6, no. 4, pp. 63–93, Jul. 2010.

[22] A. J. Conejo, E. Castillo, R. Minguez, and R. Garcia-Bertrand, *Decomposition Techniques in Mathematical Programming*. Berlin, Germany: Springer, 2006.

[23] C. Cortes, M. Mohri, and A. Rostamizadeh, “Learning non-linear combinations of kernels,” in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, Dec. 2009.

[24] E. Dall’Anese, H. Zhu, and G. B. Giannakis, “Distributed optimal power flow for smart microgrids,” *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1464–1475, Sep. 2013.

[25] J. de Hoog, T. Alpcan, M. Brazil, D. Thomas, and I. Mareels, “Optimal charging of electric vehicles taking distribution network constraints into account,” *IEEE Trans. Power Syst.*, vol. 30, no. 1, pp. 365–375, Jan. 2015.

[26] A. S. Debs and R. E. Larson, “A dynamic estimator for tracking the state of a power system,” *IEEE Trans. Power App. Syst.*, vol. 89, no. 7, pp. 1670–1678, Sep. 1970.

[27] K. Dehghanpour, Z. Wang, J. Wang, Y. Yuan, and F. Bu, “A survey on state estimation techniques and challenges in smart distribution systems,” *IEEE Trans. Smart Grid*, pp. 1–11, Sep. 2018 (To appear).

[28] D. Deka, M. Chertkov, and S. Backhaus, “Structure learning in power distribution networks,” *IEEE Trans. Control Netw. Syst.*, to appear 2018.

[29] D. Deka, S. Talukdar, M. Chertkov, and M. Salapaka, “Topology estimation in bulk power grids: Guarantees on exact recovery,” *arXiv:1707.01596*, 2017.

[30] M. B. Do Coutto Filho and J. C. Stacchini de Souza, "Forecasting-aided state estimation–Part I: Panorama," *IEEE Trans. Power Syst.*, vol. 24, no. 4, pp. 1667–1677, Nov. 2009.

[31] M. B. Do Coutto Filho, J. C. Stacchini de Souza, and R. S. Freund, "Forecasting-aided state estimation–Part II: Implementation," *IEEE Tran. Power Syst.*, vol. 24, no. 4, pp. 1678–1685, Nov. 2009.

[32] T. Erseghe, S. Tomasin, and A. Vigato, "Topology estimation for smart micro grids via powerline communications," *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3368–3377, Jul. 2013.

[33] Z. Fan, "A distributed demand response algorithm and its application to PHEV charging in smart grids," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1280–1290, Sep. 2012.

[34] M. Farivar, R. Neal, C. Clarke, and S. Low, "Optimal inverter VAR control in distribution systems with high PV penetration," in *Proc. IEEE Power & Energy Society General Meeting*, San Diego, CA, Jul. 2012.

[35] M. Farivar, X. Zhou, and L. Chen, "Local voltage control in distribution systems: An incremental control algorithm," in *Proc. IEEE Intl. Conf. on Smart Grid Commun.*, Miami, FL, Nov. 2015.

[36] M. Farivar, L. Chen, and S. Low, "Equilibrium and dynamics of local voltage control in distribution systems," in *Proc. IEEE Conf. on Decision and Control*, Florence, Italy, Dec. 2013, pp. 4329–4334.

[37] J. Franco, M. Rider, and R. Romero, "A mixed-integer linear programming model for the electric vehicle charging coordination problem in unbalanced electrical distribution systems," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2200–2210, Sep. 2015.

[38] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1–2, pp. 95–110, Mar. 1956.

[39] L. Gan, N. Li, U. Topcu, and S. Low, "On the exactness of convex relaxation for optimal power flow in tree networks," in *Proc. IEEE Conf. on Decision and Control*, Maui, HI, Dec. 2012, pp. 465–471.

[40] L. Gan and S. Low, “Convex relaxations and linear approximation for optimal power flow in multiphase radial networks,” in *Proc. Power Syst. Comput. Conf.*, Wrockaw, Poland, Aug. 2014, pp. 1–9.

[41] L. Gan, U. Topcu, and S. H. Low, “Optimal decentralized protocol for electric vehicle charging,” *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 940–951, May 2013.

[42] N. Gatsis and G. B. Giannakis, “Residential load control: Distributed scheduling and convergence with lost AMI messages,” *IEEE Trans. Smart Grid*, vol. 3, no. 2, pp. 770–786, Jun. 2012.

[43] B. Gentile, J. Simpson-Porco, F. Dorfler, S. Zampieri, and F. Bullo, “On reactive power flow and voltage stability in microgrids,” in *Proc. American Control Conf.*, Portland, OR, Jun. 2014, pp. 759–764.

[44] G. B. Giannakis, V. Kekatos, N. Gatsis, S.-J. Kim, H. Zhu, and B. Wollenberg, “Monitoring and optimization for power grids: A signal processing perspective,” *IEEE Signal Proc. Mag.*, vol. 30, no. 5, pp. 107–128, Sep. 2013.

[45] G. B. Giannakis, Y. Shen, and G. V. Karanikolas, “Topology identification and learning over graphs: Accounting for nonlinearities and dynamics,” *Proc. of the IEEE*, vol. 106, no. 5, pp. 787–807, May 2018.

[46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016, http://www.deeplearningbook.org.

[47] K. Gregor and Y. LeCun, “Learning fast approximations of sparse coding,” in *Proc. Intl. Conf. on Machine Learning*, Haifa, Israel, Jun. 2010.

[48] M. Hassanzadeh and C. Y. Evrenosoğlu, “Power system state forecasting using regression analysis,” in *Proc. IEEE Power & Energy Society General Meeting*, San Diego, CA, USA, Jul. 2012, pp. 1–6.

[49] M. Hassanzadeh, C. Y. Evrenosoğlu, and L. Mili, “A short-term nodal voltage phasor forecasting method using temporal and spatial correlation,” *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3881–3890, Sep. 2016.

[50] E. Hazan and S. Kale, "Projection-free online learning," in *Proc. Intl. Conf. on Machine Learning*, Edinburgh, Scotland, Jun. 2012.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conf. Comput. Vision and Pattern Recognit.*, Las Vegas, NV, Jun. 2016, pp. 770–778.

[52] ——, "Identity mappings in deep residual networks," in *European conf. comput. vision*. Amsterdam, Netherlands: Springer, Oct. 2016, pp. 630–645.

[53] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *Math. Program.*, vol. 162, no. 1-2, pp. 165–199, Mar. 2017.

[54] P. J. Huber, "Robust estimation of a location parameter," *The annals of mathematical stats.*, vol. 35, no. 1, pp. 73–101, Jun. 1964.

[55] *IEEE 1547 Standard for Interconnecting Distributed Resources with Electric Power Systems*, IEEE Std., 2014. [Online]. Available: http://grouper.ieee.org/groups/scc21/1547/1547_index.html

[56] Distribution test feeders. IEEE Power & Energy Society. [Online]. Available: http://nhts.ornl.gov/2009/pub/stt.pdf

[57] R. Jabr and B. Pal, "Iteratively reweighted Least-Squares implementation of the WLAV state-estimation method," *IET Gener. Transm. Dis.*, vol. 151, no. 1, pp. 103–108, Feb. 2004.

[58] M. Jaggi, "Revisiting Frank-Wolfe: Projection-free sparse convex optimization," in *Proc. Intl. Conf. on Machine Learning*, Atlanta, GA, Jun. 2013.

[59] Global energy forecasting competition. Kaggle. 2012. [Online]. Available: https://www.kaggle.com/c/global-energy-forecasting-competition-2012-load-forecasting/data

[60] G. V. Karanikolas, G. B. Giannakis, K. Slavakis, and R. M. Leahy, "Multi-kernel based nonlinear models for connectivity identification of brain networks," in *Proc. IEEE Intl. Conf. Acoust., Speech, Sig. Process.*, Shanghai, China, Mar. 2016.

[61] E. L. Karfopoulos and N. D. Hatziargyriou, "A multi-agent system for controlled charging of a large population of electric vehicles," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1196–1204, May 2013.

[62] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1617–1626, May 2013.

[63] V. Kekatos, G. B. Giannakis, and R. Baldick, "Grid topology identification using electricity prices," in *Proc. IEEE PES Soci. General Meeting*, Washington, DC, Jul. 2014.

[64] V. Kekatos, G. Wang, H. Zhu, and G. B. Giannakis, *PSSE Redux: Convex Relaxation, Decentralized, Robust, and Dynamic Approaches*. in Advances in Power System State Estimation, Ed. Mo El-Hawary; see also *arXiv:1708.03981*, 2017.

[65] V. Kekatos and G. B. Giannakis, "Joint power system state estimation and breaker status identification," in *Proc. North American Power Symposium*, University of Illinois, Urbana-Champaign, IL, Sep. 2012.

[66] V. Kekatos, L. Zhang, G. B. Giannakis, and R. Baldick, "Voltage regulation algorithms for multiphase power distribution grids," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3913–3923, Sep. 2016.

[67] ——, "Accelerated localized voltage regulation in single-phase distribution grids," in *Proc. IEEE Intl. Conf. on Smart Grid Commun.*, Miami, FL, Nov. 2015.

[68] W. H. Kersting, *Distribution System Modeling and Analysis*. Boca Raton, FL: CRC Press, 2002.

[69] M. Kloft, U. Brefeld, P. Laskov, K.-R. Müller, A. Zien, and S. Sonnenburg, "Efficient and accurate ℓ_p-norm multiple kernel learning," in *Proc. Conf. Neural Inf. Process. Syst.*, Vancouver, Canada, Dec. 2009.

[70] D. E. Knuth, *The Art of Computer Programming: Sorting and Searching*. Boston, MA: Pearson Education, 1998.

[71] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher, "Block-coordinate Frank-Wolfe optimization for structural SVMs," in *Proc. Intl. Conf. on Machine Learning*, Atlanta, GA, Jun. 2013.

[72] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Lear. Res.*, vol. 5, pp. 27–72, Jan. 2004.

[73] A. M. Leite da Silva, M. B. Do Coutto Filho, and J. F. De Queiroz, “State forecasting in electric power systems,” *IEE Proc. C. Generation, Transmission & Distribution*, vol. 130, no. 5, pp. 237–244, Sep. 1983.

[74] N. Li, G. Qu, and M. Dahleh, “Real-time decentralized voltage control in distribution networks,” in *Proc. Allerton Conf.*, Allerton, IL, Oct. 2014, pp. 582–588.

[75] X. Li, V. Poor, and A. Scaglione, “Blind topology identification for power systems,” in *Proc. IEEE Smart Grid Commun. Conf.*, Vancouver, Canada, Oct. 2013.

[76] H. Lin and S. Jegelka, “Resnet with one-neuron hidden layers is a universal approximator,” *arXiv:1806.10909*, Jul. 2018.

[77] W.-J. Ma, V. Gupta, and U. Topcu, “On distributed charging control of electric vehicles with power network capacity constraints,” in *Proc. American Control Conf.*, Portland, OR, Jun. 2014.

[78] Z. Ma, D. Callaway, and I. Hiskens, “Decentralized charging control for large populations of plug-in electric vehicles,” in *Proc. IEEE Conf. on Decision and Control*, Atlanta, GA, Dec. 2010.

[79] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, “A survey of distributed optimization and control algorithms for electric power systems,” *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2941–2962, Jul. 2017.

[80] A. Monticelli, “Electric power system state estimation,” *Proc. of the IEEE*, vol. 88, no. 2, pp. 262–282, Feb. 2000.

[81] Y. Nesterov, *Introductory Lectures on Convex Optimization*. Boston, MA: Kluwer, 2004.

[82] R. M. Oliveira, *Training autoencoders for state estimation in smart grids*, M.Sc. Thesis, University of Porto, Portugal, 2017.

[83] N. Parikh and S. Boyd, “Proximal algorithms,” *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, Nov. 2014.

[84] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, “How to construct deep recurrent neural networks,” in *Proc. Intl. Conf. on Learning Representations*, Banff, Canada, Apr. 2014.

[85] Q. Peng and S. H. Low, “Distributed algorithm for optimal power flow on a radial network,” in *Proc. IEEE Conf. on Decision and Control*, Venice, Italy, Dec. 2014, pp. 167–172.

[86] P. Richardson, D. Flynn, and A. Keane, “Optimal charging of electric vehicles in low-voltage distribution systems,” *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 268–279, Feb. 2012.

[87] J. Rivera, P. Wolfrum, S. Hirche, C. Goebel, and H.-A. Jacobsen, “Alternating direction method of multipliers for decentralized electric vehicle charging control,” in *Proc. IEEE Conf. on Decision and Control*, Florence, Italy, Dec. 2013, pp. 6960–6965.

[88] B. Robbins, C. Hadjicostis, and A. Dominguez-Garcia, “A two-stage distributed architecture for voltage control in power distribution systems,” *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1470–1482, May 2013.

[89] K. Rogers, R. Klump, H. Khurana, A. Aquino-Lugo, and T. Overbye, “An authenticated control framework for distributed voltage support on the smart grid,” *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 40–47, Jun. 2010.

[90] W. S. Rosenthal, A. M. Tartakovsky, and Z. Huang, “Ensemble Kalman filter for dynamic state estimation of power grids stochastically driven by time-correlated mechanical input power,” *IEEE Trans. Power Syst.*, vol. 33, no. 4, pp. 3701–3710, Jul. 2018.

[91] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[92] N. Samuel, T. Diskin, and A. Wiesel, “Deep MIMO detection,” in *Proc. SPAWC*, Hokkaido, Japan, Jul. 2017, pp. 1–5.

[93] C. Saunders, A. Gammerman, and V. Vovk, “Ridge regression learning algorithm in dual variables,” in *Proc. Intl. Conf. Machine Learning*, Madison, WI, Jul. 1998.

[94] M. Shahidehpour, H. Yamin, and Z. Li, *Market Operations in Electric Power Systems: Forecasting, Scheduling, and Risk Management*. New York, NY: IEEE-Wiley Interscience, 2002.

[95] C. Shao, X. Wang, X. Wang, and C. Du, “Layered and distributed charge load dispatch of considerable electric vehicles,” *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 1858–1867, Oct. 2015.

[96] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY: Cambridge university press, 2004.

[97] W. Shi, X. Xie, C.-C. Chu, and R. Gadh, “Distributed optimal energy management in microgrids,” *IEEE Trans. Smart Grid*, vol. 6, no. 3, pp. 1137–1146, May 2015.

[98] Southern California Edison dynamic load profiles. Southern California Edison. 2013. [Online]. Available: https://www.sce.com/wps/portal/home/regulatory/load-profiles/.

[99] J. F. Sturm, “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones,” *Optim. Method Softw.*, vol. 11, no. 1-4, pp. 625–653, Jan. 1999.

[100] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, “Learning to optimize: Training deep neural networks for interference management,” in *IEEE Trans. Signal Process.*, vol. 66, no. 20, Oct. 2018, pp. 1039–1050.

[101] W. Tang, S. Bi, and Y. J. Zhang, “Online charging scheduling algorithms of electric vehicles in smart grid: An overview,” *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 76–83, Dec. 2016.

[102] K. Turitsyn, P. Sulc, S. Backhaus, and M. Chertkov, “Options for control of reactive power by distributed photovoltaic generators,” *Proc. of the IEEE*, vol. 99, no. 6, pp. 1063–1073, Jun. 2011.

[103] The smart grid: An introduction. U. S. Department of Energy. 2008. [Online]. Available: http://www.oe.energy.gov/SmartGridIntroduction.htm

[104] Power systems test case archive. Univ. of Washington. 2017. [Online]. Available: http://www.ee.washington.edu/research/pstca/

[105] Federal Highway Administration. US Department of Transportation. [Online]. Available: http://nhts.ornl.gov/2009/pub/stt.pdf.

[106] G. Wang, G. B. Giannakis, and J. Chen, “Robust and scalable power system state estimation via composite optimization,” *IEEE Trans. Smart Grid (Re-revised);* see also *arXiv:1708.06013*, 2017.

[107] G. Wang, A. S. Zamzam, G. B. Giannakis, and N. D. Sidiropoulos, “Power system state estimation via feasible point pursuit: Algorithms and Cramér-Rao bound,” *IEEE Trans. Signal Process.*, vol. 66, no. 6, pp. 1649–1658, Mar. 2018.

[108] G. Wang, L. Zhang, G. B. Giannakis, J. Chen, and M. Akçakaya, “Sparse phase retrieval via truncated amplitude flow,” *IEEE Trans. Signal Process.*, vol. 66, no. 2, pp. 479–491, Jan. 2018.

[109] G. Wang, G. B. Giannakis, J. Chen, and J. Sun, “Distribution system state estimation: An overview of recent developments,” *Front. Inf. Technol. Electron. Eng.*, vol. 20, no. 1, pp. 1–14, Jan. 2019.

[110] G. Wang, G. B. Giannakis, Y. Saad, and J. Chen, “Phase retrieval via reweighted amplitude flow,” *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2818–2833, Jun. 2018.

[111] G. Wang, S.-J. Kim, and G. B. Giannakis, “Moving-horizon dynamic power system state estimation using semidefinite relaxation,” in *Proc. IEEE PES General Meeting*, Washington, DC, Jul. 2014, pp. 1–5.

[112] H. Wang and A. Banerjee, “Online alternating direction method,” in *Proc. Intl. Conf. on Machine Learning*, Edinburgh, Scotland, Jun. 2012, pp. 1119–1126.

[113] Y. Wang, V. Sadhanala, W. Dai, W. Neiswanger, S. Sra, and E. P. Xing, “Parallel and distributed block-coordinate Frank-Wolfe algorithms,” in *Proc. Intl. Conf. on Machine Learning*, New York City, NY, Jun. 2016.

[114] Z. Wang, Q. Ling, and T. Huang, “Learning deep ℓ_0 encoders,” in *Proc. AAAI Conf. on Artificial Intell.*, Phoenix, AZ, Feb. 2016, pp. 2194–2200.

[115] S. Weckx and J. Driesen, “Load balancing with EV chargers and PV inverters in unbalanced distribution grids,” *IEEE Trans. Sustain. Energy*, vol. 6, no. 2, pp. 635–643, Apr. 2015.

[116] Plug-in electric vehicles in the united states. Wikipedia. 2018. [Online]. Available: https://en.wikipedia.org/wiki/Plug-in_electric_vehicles_in_the_United_States

[117] Renewable energy. Wikipedia. 2018. [Online]. Available: https://en.wikipedia.org/wiki/Renewable_energy

[118] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*, 2nd ed. New York, NY: Wiley & Sons, 1996.

[119] L. Wu and M. Shahidehpour, "Accelerating the Benders decomposition for network-constrained unit commitment problems," *Energy Syst.*, vol. 1, no. 3, pp. 339–376, May 2010.

[120] W. A. Wulf, "Great achievements and grand challenges," *The Bridge*, vol. 30, no. 3/4, pp. 5–10, Sep. 2010. [Online]. Available: http://www.greatachievements.org/.

[121] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-net for compressive sensing MRI," in *Proc. Adv. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016.

[122] P. You, S. H. Low, L. Zhang, R. Deng, G. B. Giannakis, Y. Sun, and Z. Yang, "Scheduling of EV battery swapping, II: Distributed solutions," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 4, pp. 1920–1930, Dec. 2018.

[123] A. S. Zamzam, X. Fu, and N. D. Sidiropoulos, "Data-driven learning-based optimization for distribution system state estimation," *arXiv:1807.01671*, 2018.

[124] B. Zhang, A. Dominguez-Garcia, and D. Tse, "A local control approach to voltage regulation in distribution networks," in *Proc. North American Power Symposium*, Manhattan, KS, Sep. 2013.

[125] L. Zhang, D. Feng, J. Lei, C. Xu, Z. Yan, S. Xu, N. Li, and L. Jing, "Congestion surplus minimization pricing solutions when lagrange multipliers are not unique," *IEEE Trans. Power Syst.*, vol. 29, no. 5, pp. 2023–2032, Sep. 2014.

[126] L. Zhang, G. V. Karanikolas, M. Akcakaya, and G. B. Giannakis, "Fully automatic segmentation of the right ventricle via multi-task deep neural networks," in *Proc. IEEE Intl. Conf. Acoust., Speech, Sig. Process.*, Calgary, Canada, Apr. 2018.

[127] L. Zhang, V. Kekatos, and G. B. Giannakis, “A generalized Frank-Wolfe approach to decentralized control of vehicle charging,” in *Proc. IEEE Conf. on Decision and Control*, Las Vegas, NV, Dec. 2016.

[128] ——, “Scalable electric vehicle charging protocols,” *IEEE Trans. Power Syst.*, vol. 32, no. 2, pp. 1451–1462, Mar. 2017.

[129] L. Zhang, D. Romero, and G. B. Giannakis, “Fast convergent algorithms for multi-kernel regression,” in *Proc. IEEE Wkshp. on Statistical Signal Process.*, Mallorca, Spain, Jun. 2016.

[130] L. Zhang, G. Wang, and G. B. Giannakis, “Going beyond linear dependencies to unveil connectivity of meshed grids,” in *Proc. IEEE Wkshp. on Comput. Adv. Multi-Sensor Adaptive Process.*, Curacao, Dutch Antilles, Dec. 2017.

[131] ——, “Real-time power system state estimation and forecasting via deep neural networks,” *arXiv:1811.06146*, Nov. 2018.

[132] ——, “Real-time power system state estimation via deep unrolled neural networks,” in *Proc. Global Conf. on Signal and Info. Process.*, Anaheim, CA, USA, Nov. 2018.

[133] ——, “Power system state forecasting via deep recurrent neural networks,” in *Proc. IEEE Conf. on Acoustics, Speech and Signal Process.*, Brighton, UK, May 2019 (Submitted).

[134] L. Zhang, G. Wang, G. B. Giannakis, and J. Chen, “Compressive phase retrieval via reweighted amplitude flow,” *IEEE Trans. Signal Process.*, vol. 66, no. 19, pp. 5029–5040, Aug. 2018.

[135] L. Zhang, G. Wang, D. Romero, and G. B. Giannakis, “Randomized block Frank-Wolfe for convergent large-scale learning,” *IEEE Trans. Signal Process.*, vol. 65, no. 24, pp. 6448 – 6461, Sep. 2017.

[136] J. Zhao, G. Zhang, Z. Dong, and M. La Scala, “Robust forecasting aided power system state estimation considering state correlations,” *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2658–2666, Jul. 2018.

[137] Y. Zhao, J. Chen, and H. V. Poor, “A learning-to-infer method for real-time power grid multi-line outage identification,” *IEEE Trans. Smart Grid*, pp. 1–10, 2018 (To appear; *see also* arXiv:1710.07818).

[138] H. Zhu and H. J. Liu, “Fast local voltage control under limited reactive power: Optimality and stability analysis,” *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3794–3803, Sep. 2016.

[139] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, “MATPOWER: Steady-state operations, planning and analysis tools for power systems research and education,” *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.