Distributed Optimization Methods for Monitoring and Operating Electric Power Systems

Distributed Optimization Methods for Monitoring and Operating Electric Power Systems

A dissertation presented
by
Ariana Sage Minot
to
The John A. Paulson School of Engineering and Applied Sciences

in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Applied Mathematics

Harvard University
Cambridge, Massachusetts

March 2017

Dissertation Advisors

Professor Na Li and Professor Yue Lu

Author

Ariana Sage Minot

Distributed Optimization Methods for Monitoring and Operating Electric Power Systems

Abstract

Increasing power demands, aging infrastructure, and growth in renewable energy production necessitate new strategies for grid operations. To achieve reliable operations in this setting, it is important to shift away from centralized control paradigms to distributed approaches that 1) allow for quicker solution times by decomposing the problem to be solved in parallel and 2) avoid communication bottlenecks that result from all data and measurements being sent to a centralized location. Since power systems are interconnected systems, decoupling such problems is challenging. This dissertation studies how to improve monitoring and operation of networked systems, specifically electric power grids, through distributed optimization and scientific computing.

The thesis focuses on two areas. In the first part, we design fully distributed algorithms for power system state estimation under both linear and nonlinear measurement models. For the nonlinear setting, we develop a distributed Gauss-Newton method. The main computational burden is solving a series of large, sparse linear systems. Iterative linear solvers based on matrix-splitting techniques are developed to exploit the sparsity pattern induced by the underlying network structure and physical laws of power networks. In addition, a distributed unscented Kalman filter is proposed to improve upon the state estimator by incorporating information from past measurements. Advantages of the proposed distributed approaches include increased scalability in terms of both computation and communication.

The second part of this thesis concerns the operation of power grids, in particular how to determine the optimal set points for the system. The use of primal-dual interior point (PDIP) methods for the optimal power flow (OPF) problem and the security-constrained OPF problem is studied. We design domain decomposition techniques, as well as reduction and reordering schemes, to parallelize the PDIP method by exploiting sparsity structure. Last, distributed energy resources (DERs) subject to uncertainty and various dynamic constraints are increasingly participating in electricity production and demand. A computationally tractable approximation for polyhedral projections is proposed to quantify the aggregate capability of distributed energy resources. This work contributes to the question of how to utilize DERs to improve and support bulk grid operations.

Contents

List of Figures

List of Tables

Citations to Previously Published Work

Most of Chapters 2 and 3 were published as the following papers:

> A. Minot and N. Li. A Fully Distributed State Estimation Using Matrix Splitting Methods. In *2015 American Control Conference (ACC)*, pages 2488–2493, July 2015.

> A. Minot, Y. M. Lu, and N. Li. A Distributed Gauss-Newton Method for Power System State Estimation. *IEEE Transactions on Power Systems*, 31(5):3804–3815, Sept 2016.

Most of Chapter 4 was published as the following paper:

> A. Minot, H. Sun, and D. Nikovski. A Fully Distributed Filtering Scheme for Load-Based Dynamic State Estimation. In *2016 IEEE Power and Energy Society General Meeting (PESGM)*, pages 1–5, July 2016.

Most of Chapter 5 was published as the following paper:

> A. Minot, Y. M. Lu, and N. Li. A Parallel Primal-Dual Interior-Point Method for DC Optimal Power Flow. In *2016 Power Systems Computation Conference (PSCC)*, pages 1–7, June 2016.

Acknowledgments

I would like to express my heartfelt gratitude for my thesis advisors Professor Na (Lina) Li and Professor Yue Lu. Since 2012, Yue has helped me to learn how to critically assess potential problems and equipped me with foundational skills for research. His ability to ask pointed questions that zero in on the main difficulties of a problem has always impressed me. I am grateful for the many years as a student in his group. I started my graduate career hoping to work on power grid problems, but it was not until my third year when I met Lina that this became a reality. I remember excitedly reading her profile as a new faculty hire in a SEAS newsletter and emailing her before she had started at Harvard. I am very thankful to Lina for taking a chance on me and for all her guidance. Lina taught me how to formulate interesting research problems and how to pursue the right directions to make contributions towards solving them. Her passion and expertise in mathematics is an inspiration for me. Beyond research, I am indebted to my advisors for their mentorship.

I would like to thank the other professors who have helped me throughout my graduate career, in particular Professor Kaxiras and Professor Mahadevan who hosted me for research internships, Professor Brenner and Professor Amir for serving on my qualifying committee, and Professor Aziz for being on my defense committee, as well as for discussions about electric grid research since my first visit to Harvard. I would like to acknowledge all the wonderful people I have met over the years through SEAS: Ameya Agaskar, Chenhui Hu, Hongyao Ma, Jie Ding, Chuang Wang, Karen McKinnon, Beth Chen, Taylor Killian, Guannan Qu, Yingying Li, Masoud Badiei, Yannick Meier, Chinwendu Cenyioha, Kathryn Heal, Yu Xiang, Shahin Shahrampour, Hamed Farhadi, Sindri Magnusson, Ahmad Beirami, Zafer Dogan, Hong Hu, and Wangyu Luo. In particular, I am

indebted to my mentor Rehana Patel and co-mentee Yang Chen for their guidance and friendship. I am thankful to IACS which has always felt like a home within SEAS. I am very grateful to Gioia Sweetland, Tricia Jacome, and Lisa Frazier for all of their help. I was lucky to have the opportunity to work with researchers at NREL and MERL during my PhD. I would like to thank Dr. Emiliano Dall'Anese for many helpful discussions and the opportunity to collaborate on exciting problems on distribution systems during my final year of PhD. I would like to acknowledge Dr. Hongbo Sun for his support and warm mentorship. My sincere thanks to the organizations that helped fund my graduate studies: NSF Graduate Research Fellowship Program, Smith Family Fellowship, Blue Waters Graduate Fellowship, IACS Student Scholarship, and Harvard GSAS Merit Fellowship. To Professor Kate Scholberg and Professor Chris Walter, my experience in the Neutrino Group at Duke inspired me to pursue graduate studies. To Dr. Fayard and Dr. Kado, since my time at LAL, it has been my dream to pursue a scientific career in a research laboratory.

This would not be complete without acknowledging my family and friends. To my mother Cynthia, your help these past few months has meant so much. We would have gotten by, but it would have been very tough! To my dad Hal, thank you for always having been interested in what I am "up to" and for lending us mom these past few months for babysitting help. To my brother Zach and sister Alissa, knowing you are nearby has made surviving New England winters much more fun. To my daughter, for providing a new light in my life and a source of joy always in my mind. Finally, I would like to thank my husband Max. Graduating, finding a job, and having a baby all at once has been overwhelming, and having him by my side made all the difference. I could not have done this without his constant support and belief in me.

To Max.

Chapter 1

Introduction

THIS DISSERTATION focuses on distributed algorithms for advancing monitoring capability and operations of the electric power grid. The electric power grid is the critical backbone upon which our water, fuel, and transportation networks depend. The U.S. electric power grid is a complex, interconnected system, composed of over 6 million miles of transmission and distribution lines responsible for delivering power to more than 143 million customers [1]. Increasing power demands, aging infrastructure, and growth in renewable energy production necessitate new strategies for grid operations [4]. At the same time, to ensure safe, reliable operations, electric grid operators must maintain a delicate balance between supply and demand in real-time. Given the size and complexity of the electric grid, such problems present one of the largest engineering challenges of the 21st century. The work of this thesis is motivated by the challenges and opportunities facing the current and future electric grid.

An electric power system is composed of three main entities: the generating plants that produce electricity, the load or demand that consumes power, and the wires or grid that transports it [84]. The electric power grid is divided into two systems: the high-voltage transmission network and a series of lower voltage distribution networks, also known as feeders, which deliver power to customers. A rough dividing line between the two systems is 100 kV [84]. A **substation** serves as the interconnection point between a **distribution feeder** and the **bulk transmission** system.

1.1 Enhancing Monitoring Capability: State Estimation

Since first introduced by Schweppe in 1970 [94], **state estimation** has played an important role in the operation of power systems. State estimation provides a view of real-time power system conditions for the system operator to efficiently and reliably operate the power grid [2]. Improving state estimation advances the capability of online power dispatch, contingency analysis, frequency control, and fault diagnosis. With the growing penetration of renewable energy, it is becoming more demanding to estimate the system state promptly and accurately. There have been increasing research efforts in developing and integrating new sensor technology to create a more advanced state estimation system.

As computational demands increase with the availability of these new measurements, there is a heightened need to develop **distributed** algorithms that allow each aggregate **bus** (*i.e.*, points of connection or nodes where power is drawn or injected into the grid [5]) or control area to have its own processor for local state estimation [52]. Advantages of distributed approaches include increased robustness as well as reduction in computation, communication, and memory requirements per area since each area requires only a subset of the global information. Whether trying to achieve wide-area control between large interconnected areas of the power grid or at the micro-grid level, robust and fully-distributed state estimation will be critical [40].

1.2 Improving Operations: Solving the Optimal Power Flow Problem

The **optimal power flow (OPF) problem** is formulated as an optimization program that determines the amount of power to be generated from each power plant in order to minimize costs (*e.g.*, generation costs, transmission costs, and CO_2 production) under certain physical and operational constraints. OPF is key to grid operations and is solved up to every 5 minutes for minute-ahead planning [13]. The OPF problem was first proposed in 1962 by Carpentier [18]. Since then it has been an active area of study in the power engineering and optimization community, and a variety of approximations and special cases have been studied [73]. For example, one of the main complications in solving OPF is the nonlinear constraints introduced by Kirchoff's Laws.

To this day, there does not exist a method that quickly and robustly solves the full nonlinear OPF problem. However, finding such a method could save on the order of ten billion dollars per year [13]. Obtaining an accurate solution for OPF in quasi-real-time remains a challenge that has been actively researched for over 50 years. Solution methods three to five times faster than existing ones are needed [13]. Furthermore, grid operators must select set points that maintain safe operations in the event of a **contingency**, or equipment failure. These extra security constraints lead to an even more computationally expensive optimization program, known as the **security-constrained optimal power flow (SC-OPF)** problem. In order to achieve gains in computing runtime, it is necessary to use **parallel** and **distributed** computing.

1.3 Future Outlook: Distributed Energy Resources

Operating the electric power grid under a high penetration of renewable energy poses challenges for grid operations due to the non-dispatchable, intermittent nature of wind and solar energy. Reliably operating the grid in this new setting is possible under increased ramping and reserve capacity. Traditionally, the power grid is operated assuming little flexibility in the distribution system. One way of increasing reserve and ramping capability is through **distributed energy resources (DERs)**, such as storage, distributed generation, and responsive loads, at the distribution-system level.

By aggregating the power savings due to many heterogeneous controllable devices (*e.g.*, batteries, rooftop photovoltaics, thermostatically controlled loads), distributed energy resources can contribute significant flexibility to power grid operations [88]. The idea is to coordinate this network of distribution system assets to function as a unified resource, or a *virtual power plant* (VPP) [76]. In order to utilize a distribution system as a VPP, the transmission system operators need to coordinate the devices so that the aggregate **net load** (demand minus production) is controllable and predictable (see Figure 1.1). This is challenging due to the large number of devices which have varying dynamics, operating constraints, and controllability. Therefore, transformative changes are required in how we operate the power grid [84].

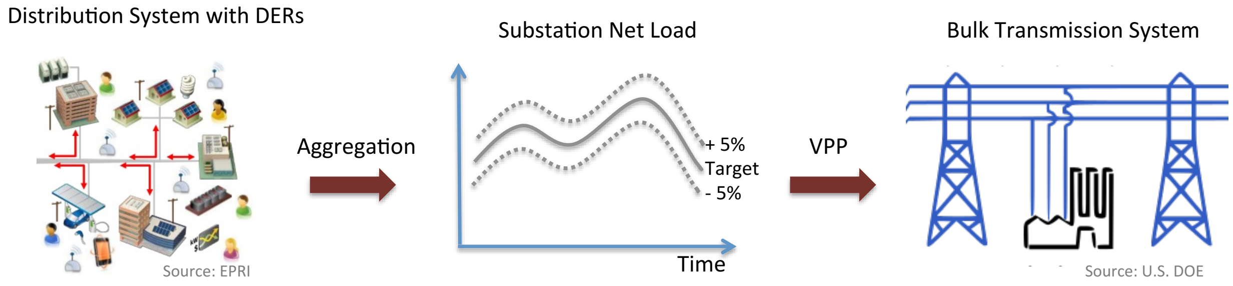


Figure 1.1: A distribution system with heterogeneous controllable devices can act as a virtual power plant (VPP) if the devices can be coordinated so that the aggregated net load at the substation (*i.e.*, connection point between a distribution feeder and the bulk transmission system) stays within a fixed percentage of the targeted value.

1.4 Distributed and Parallel Algorithms

One of the main themes of this dissertaiton is exploring how sparsity induced by the underlying physics and limited connectivity of power networks can be utilized to design efficient distributed algorithms in terms of computation and communication requirements. Distributed algorithms decompose a problem so that it can be solved across multiple computers [6]. It is challenging to develop effective parallel algorithms for interconnected systems, such as electric grids, which do not trivially decouple [95].

In Figure 1.2, a schematic is provided to demonstrate the difference between a **centralized** and **fully distributed** communication scheme. In a centralized communication scheme, the centralized controller receives data or measurements from individual control areas. The centralized controller then performs a computation to solve an optimization problem or determine some control action

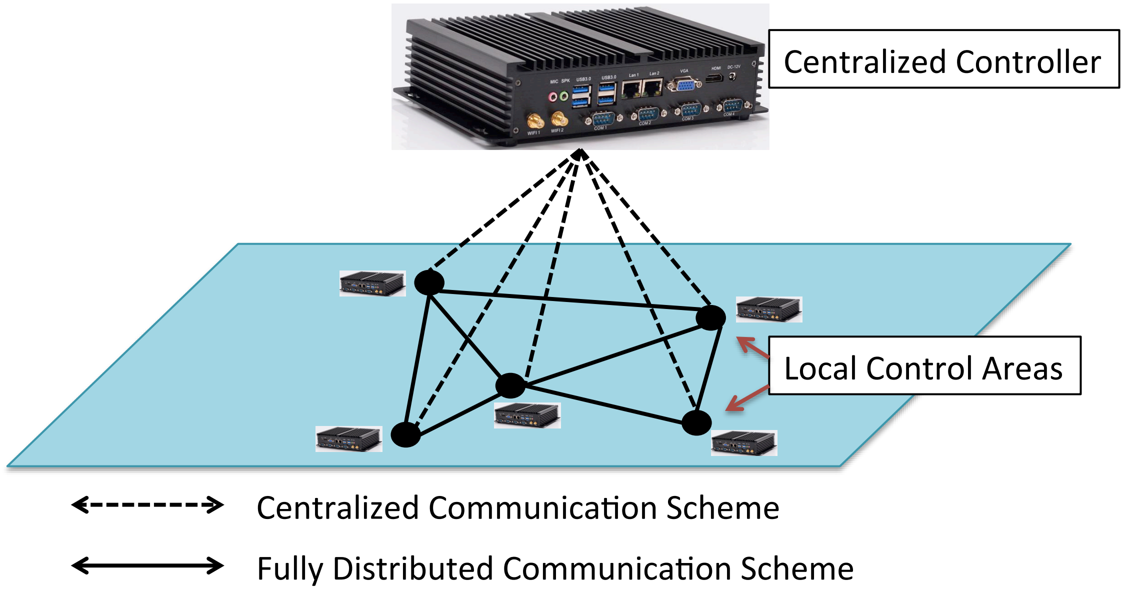

Centralized Controller
Local Control Areas
Centralized Communication Scheme
Fully Distributed Communication Scheme

Figure 1.2: Centralized versus distributed communication schemes.

and finally broadcasts the result back to the local control areas. In contrast, in a fully distributed communication setting, there is no centralized controller. Instead, control areas perform computations locally and rely on communication with their neighbors as shown in the connectivity network. Hybrid communication structures are also possible, where control areas still do some local processing while using a centralized controller to coordinate global information. We refer to algorithms using such hybrid settings (with both local computation and communication, as well as centralized coordination) as **parallel** or **semi-distributed** algorithms. The designation *semi-distributed* is used to signify that the main computational burden of the algorithm is fully distributed, while other portions of the algorithm that are less computationally intensive utilize a centralized controller.

Advantages of distributed processing include scalability. For instance, if the number of neighbors of any given control area stays roughly constant, then distributed methods are highly scalable as the size of the network grows since they utilize only neighbor-to-neighbor communication. Second, distributed algorithms are flexible since they can be designed to allow control areas to pursue local objectives. Lastly, distributed methods enhance reliability of systems since the control areas depend on local and neighboring information rather than global information communicated via a single control center that may experience interruptions in service. The main question addressed in this thesis is how to apply these ideas to problems in power systems.

1.5 Outline of Thesis and Summary of Contributions

Part I: Distributed Monitoring in Electric Power Systems (Chapters 2, 3, 4)

Chapter 2 studies the state estimation problem for power systems under a linear measurement model. A new fully distributed algorithm for power system state estimation is developed based on matrix-splitting techniques. Analytical results include some insight on how to design an optimal matrix splitting to improve convergence. The major merits of the algorithm include its convergence speed and reduced memory requirements.

Chapter 3 extends the ideas developed for linear state estimation to a setting with a nonlinear measurement model. A fully distributed Gauss-Newton algorithm is proposed that utilizes matrix splitting to solve for the Gauss-Newton step, the main computational burden of the algorithm. An implementation using the Message Passing Interface (MPI) [45] protocol is developed, demonstrating the gain in runtime of our distributed algorithm over a centralized implementation.

Chapter 4 examines how to improve upon traditional static state estimators by including measurements from previous time instants. Changes in the system state are induced by changes in load or demand. Unscented Kalman filter techniques are utilized to accomodate a nonlinear measurement model. An approximation to the measurement covariance matrix is proposed to allow for a fully distributed method with limited communication. Numerical experiments verify the advantages of our distributed method in terms of accuracy and communication requirements.

Part II: Distributed Operation of Electric Power Systems (Chapters 5, 6, 7)

Chapter 5 focuses on the linearized optimal power flow (OPF) problem and the application of primal-dual interior point (PDIP) methods. In particular, the problem of how to parallelize PDIP methods to improve computational performance is considered. A semi-distributed PDIP algorithm for the linearized DC OPF is proposed utilizing sparsity induced by the physical laws of power networks. Simulations demonstrate the performance of the proposed method in terms of convergence of the objective function and solution feasibility.

Chapter 6 extends the ideas developed for linear OPF to security-constrained OPF under various power flow formulations. The use of domain decomposition techniques is studied to enhance parallelization not only across buses in the power network but also across contingencies. Results include estimation of the speed up of the proposed distributed method over the centralized setting.

Chapter 7 concerns the operation of distribution systems with distributed energy resources (DERs), and in particular, how to describe their aggregate behavior and capability in terms of the net load achievable at the substation. This problem is formulated in terms of a polyhedral projection using ideas from computational geometry, and a tractable approximation for describing the aggregate net load is proposed based on optimization techniques.

Chapter 2

Fully Distributed Linear State Estimation for Power Systems Using Matrix-Splitting Methods

ACCURATE STATE ESTIMATION is critical for robust and secure operation of the electric power grid. In order to efficiently operate the power grid, the system operator relies on the output from state estimation to assess real-time operating conditions. Advancing state estimation algorithms will facilitate further improvements in state-of-the-art system control functionalities, such as fault diagnosis, oscillation damping, and online power dispatch. In order to realize an electric grid with a high penetration of renewable energy sources, such advanced control capabilities are increasingly important. At the same time, estimating the system state quickly with high accuracy is challenging in such a setting. For these reasons, recently there has been significant interest in using advanced sensor technology to enhance state estimation. One such example is the development of a Wide-Area Measurement System (WAMS) that uses Phasor Measurement Units (PMUs) [22].

Processing these new measurements introduces additional computational demands, motivating the need for state estimation algorithms that execute in a distributed manner. A distributed approach allows control areas, or geographically aggregated buses, to locally estimate their state rather than require a central processer. Furthermore, since each control area utilizes a subset of information from the global system, distributed algorithms increase robustness of state estimation while reducing the computation and memory requirements within a control area. Fully-distributed state estimation methods will be of great use for achieving wide-area control between large interconnected areas of the power grid, as well as successfully monitoring down at the micro-grid level [40].

There have been many research efforts in developing distributed state estimation methods for electric power systems [35, 43, 71]. In hierarchical distributed approaches, state estimation is carried out locally and then information is exchanged with a central processor that coordinates the local estimates to produce a global state estimate. Hierarchical approaches to state estimation in power systems are explored in [32, 34, 56, 57, 60, 61, 100, 109]. One disadvantage of requiring a central coordinator is the potential for communication bottlenecks and reduced robustness.

In contrast, fully distributed state estimation utilizes neighbor-to-neighbor communication rather than relying on a central coordinator. Of recent interest have been gossip-based algorithms for fully distributed state estimation [51, 70, 108]. One potential shortcoming of such methods is that an estimate of the global state is required at each area. For large networks, the memory requirements can be prohibitive. Fully distributed methods requiring only local rather than global estimates per area have recently been proposed: using decomposition methods [17, 19, 25]; applying alternating direction method of multipliers (ADMM) [59]; and information filter-based techniques [97]. The decomposition method in [19] lacks guarantees of the convergence of the distributed state estimates to the estimates obtained by a centralized state estimator. The ADMM approach in [59] guarantees asymptotic convergence. However, possible disadvantages include the computation and storage of additional information in the form of Lagrange multipliers and complications in an asynchronous setting. The method of [97] converges in finite iterations, but the network is assumed to be acyclic and the iteration number for convergence increases linearly with the network size. For large-scale networks, asymptotically convergent methods may be preferable, especially if convergence speed scales independently of the network size.

In this chapter, we explore the use of matrix splitting techniques [102, 106] for developing a new distributed state estimation algorithm that exploits inherent sparse structure in power systems. As a preliminary step, we adopt a linear measurement model, also known as DC state estimation [40]. In Chapter 3, we extend these results to a nonlinear measurement model. In the DC setting, the optimal state estimation problem is equivalent to solving a system of linear equations. Applying matrix splitting to solve this system of equations leads to an iterative solution to the state estimation problem. The choice of matrix splitting is made to ensure convergence of the distributed state estimates. In our algorithm, each bus calculates its own state estimate, and information is exchanged between neighboring nodes. The contributions of this chapter include a new fully distributed method for power system linear state estimation with the following features:

1. The method combines the use of both traditional SCADA (supervisory control and data acquisition) system measurements of power injections and flows with PMU measurements of voltages.

2. The convergence of the distributed estimates to the optimal estimates obtained by a centralized algorithm is guaranteed analytically.

3. Each local bus or area only needs to hold an estimate of its own local states rather than an estimate of the entire global state of the system. This frees up a large amount of communication and memory resources.

The chapter is outlined as follows. In Section 2.1, we provide the mathematical problem statement and introduce the centralized state estimation problem. In Section 2.2, we present our distributed linear state estimation algorithm with analysis of its communication requirements and convergence properties. In Section 2.3, numerical simulations demonstrate the effectiveness of our method.

Notations: We use v_k to denote the k^{th} entry of a vector $\mathbf{v}$. The $(i, j)^{th}$ entry of a matrix $\mathbf{M}$ is given by M_{ij}, and the i^{th} row and j^{th} column are given by $[\mathbf{M}]_\mathbf{i}$ and $[\mathbf{M}]^\mathbf{j}$ respectively. The transpose of a vector or matrix $\mathbf{X}$ is denoted $\mathbf{X^T}$. The matrix inequality $\boldsymbol{M} > 0$ is to be interpreted as each element of $\boldsymbol{M}$ being positive. In contrast, we use $\boldsymbol{M} \succ 0$ to denote that $\boldsymbol{M}$ is positive definite.

2.1 Problem Statement

We consider an interconnected power network, denoted by an undirected graph $(\mathcal{N}, \mathcal{E})$ with a set $\mathcal{N} \triangleq \{1, 2, \ldots, n\}$ of buses and a set $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ of transmission lines connecting the buses. The goal of state estimation in power systems is to infer the unknown voltages (magnitude and phase angle) at each bus from a set of measurements of the system. Figure 2.1 provides an example of the IEEE 14-bus test system with different measurements. In this chapter, we consider the linearized DC state estimation problem, which is a good approximation under the assumptions that 1) the voltage angle differences between neighboring buses are small, 2) the voltage magnitudes are all close to 1 per unit (p.u.), and 3) the transmission lines have negligible resistance [40]. Thus in the setting of this chapter, the voltage magnitudes are assumed to be 1 p.u., leaving only the voltage phase angles, $\boldsymbol{\theta}$, as unknown variables. The objective of state estimation is to infer $\boldsymbol{\theta}$ from the measurements.

There are two typical power measurement systems. One is the traditional SCADA measurements including power flows along transmission lines and power injections at buses; the other system uses PMUs to measure the voltages and currents directly. Since we only consider the DC state estimation where the voltage magnitudes are assumed to be 1 p.u., we assume that the power flow measurements and current measurements are interchangeable and thus only consider power flow measurements. Therefore, we consider the following three measurements: 1) a measurement of power flow along the transmission line between buses, i and j, denoted by $\hat{P}_{ij}$, 2) a measurement of power injection at bus i, denoted by $\hat{P}_i$, and 3) a measurement of phase angle at bus i, denoted by $\hat{\theta}_i$. The cost of measurement units is too prohibitive to deploy enough sensors to take

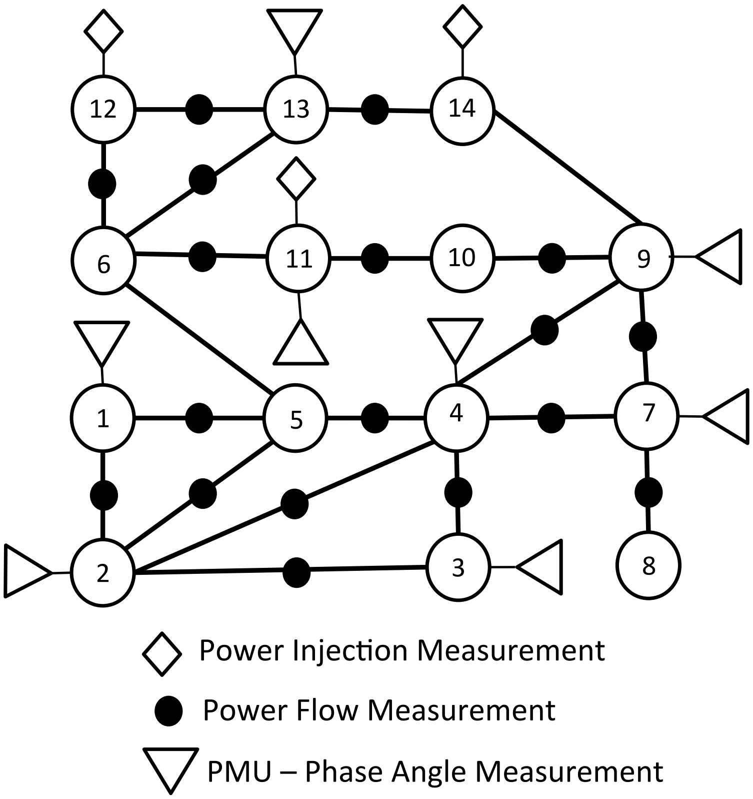

12
13
14
6
11
10
9
1
5
4
7
2
3
8
◇ Power Injection Measurement
● Power Flow Measurement
▽ PMU – Phase Angle Measurement

Figure 2.1: Schematic illustration of IEEE 14-bus test system with location and types of measurements used.

all possible measurements of the system. We denote the set of measurements as $\mathcal{Z}$ and the ordered vector of measurements as $\mathbf{z}$. The DC model linearly relates the measurements $\mathbf{z}$ and unknown voltage phase angles $\boldsymbol{\theta}$,

$$\mathbf{z} = \mathbf{H}\,\boldsymbol{\theta} + \mathbf{e}, \tag{2.1}$$

where $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$. Let the neighbor set of node i be denoted $\mathcal{N}_i$. Denote the susceptance of the transmission line between nodes i and j as B_{ij}. The entries of the measurement matrix $\mathbf{H}$ are given as follows:

1. If $z_k = \hat{P}_{ij}$, then

$$H_{kl} = \begin{cases} B_{ij}, & l = i \\ -B_{ij}, & l = j \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

2. If $z_k = \hat{P}_i$, then

$$H_{kl} = \begin{cases} \sum\limits_{j \in \mathcal{N}_i} B_{ij}, & l = i \\ -B_{ij}, & l = j, j \in \mathcal{N}_i \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

3. If $z_k = \hat{\theta}_i$, then

$$H_{kl} = \begin{cases} 1, & l = i \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

Given an invertible matrix $\mathbf{W}$ that weights the measurements, we seek an estimate of $\boldsymbol{\theta}$ that solves the following optimization problem:

$$\min_{\boldsymbol{\theta}} \, (\mathbf{z} - \mathbf{H}\boldsymbol{\theta})^{\mathbf{T}} \mathbf{W} (\mathbf{z} - \mathbf{H}\boldsymbol{\theta}), \tag{2.5}$$

where one possible choice for $\mathbf{W}$ is the inverse covariance matrix $\mathbf{R}^{-\mathbf{1}}$. The following assumption is made on the weighting matrix:

Assumption 2.1. *The weighting matrix* $\mathbf{W}$ *is invertible and diagonal with entries* $W_{kk} \equiv w_k > 0$.

The solution to (2.5) is given by choosing an estimate $\boldsymbol{\theta}_C^*$ to satisfy the first-order optimality conditions,

$$(\mathbf{H^T W H})\boldsymbol{\theta}_{\mathrm{C}}^* = \mathbf{H^T W z}. \tag{2.6}$$

We make the following assumption.

Assumption 2.2. *(Observability) The measurement matrix* $\mathbf{H}$ *has full column rank.*

With this, we can calculate the unique solution as

$$\boldsymbol{\theta}_C^* = (\mathbf{H^T W H})^{-1}\mathbf{H^T W z}. \tag{2.7}$$

Calculating the centralized solution $\boldsymbol{\theta}_C^*$ depends on full knowledge of $\mathbf{H}$, $\mathbf{W}$, and $\mathbf{z}$. This requires all measurements being sent to a central coordinator, as well as solving the system in (2.6), which can be too large to be practical for wide-area control. These communication and computational burdens present a difficulty for state estimation in large-scale power systems. To address this challenge, we propose a new distributed algorithm for state estimation that requires only local information internal to the local control area of interest and information communicated from neighbors. For ease of exposition, we assume that each bus is a control area with a local processor. Our method can be extended to cases where a control area includes multiple buses in the network as described in [43].

Assuming there is at least one phase angle measurement in the system, we formulate the state estimation problem without using a reference bus as is done in [111]. The algorithm does not require local reference phase angles in each control area since control areas share neighboring estimates. The algorithm does assume that all PMU measurements being used are taken at the same time instant with respect to GPS.

2.2 Distributed Linear DC State Estimation Algorithm

We propose a new state estimation algorithm based on a matrix-splitting technique that allows us to solve (2.6) in a distributed way. This method is inspired by the use of matrix-splitting for developing a distributed Newton method for the Network Utility Maximization (NUM) problem of Wei *et al.* in [106].

2.2.1 Matrix Splitting for Distributed DC State Estimation Problem

First, we introduce the idea behind matrix splitting for solving linear systems [102]. Given a square linear system of n equations, $\mathbf{A}\mathbf{x} = \mathbf{y}$, we seek to write $\mathbf{A}$ as the sum of an invertible matrix, $\mathbf{M}$, and a matrix $\mathbf{N}$, *i.e.* $\mathbf{A} = \mathbf{M} + \mathbf{N}$. Then for an arbitrary $\mathbf{x^0} \in \mathbb{R}^{\mathbf{n}}$, consider the following iterative scheme:

$$\mathbf{x^{t+1}} = -\mathbf{M^{-1}}\mathbf{N}\mathbf{x^t} + \mathbf{M^{-1}}\mathbf{y}. \tag{2.8}$$

The sequence $\{\mathbf{x^t}\}$ converges to its limit $\mathbf{x}^*$ as $t \to \infty$ if and only if the spectral radius [1] of the matrix $\mathbf{M^{-1}}\mathbf{N}$ is strictly less than 1 [102]. In the event that the sequence converges, the limit $\mathbf{x}^*$ is the solution of the system, i.e., $\mathbf{A}\mathbf{x}^* = \mathbf{y}$.

[1] Let $\lambda_1, \ldots, \lambda_n$ be the (real or complex) eigenvalues of a matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$. Then, define the spectral radius $\rho(\boldsymbol{A}) \overset{\text{def}}{=} \max\{|\lambda_1|, \ldots, |\lambda_n|\}$.

To apply this to state estimation, we define

$$\mathbf{A} \overset{\text{def}}{=} \mathbf{H^T W H}. \tag{2.9}$$

We can decompose $\mathbf{A}$ into the sum of a matrix containing its diagonal entries, $\mathbf{D}$, and a matrix containing its off-diagonal entries $\mathbf{E}$. Specifically, let

$$D_{ij} = \begin{cases} A_{ii}, & j = i \\ 0, & j \neq i \end{cases} \tag{2.10}$$

and

$$E_{ij} = \begin{cases} A_{ij}, & j \neq i \\ 0, & j = i \end{cases} \tag{2.11}$$

yielding $\mathbf{A} = \mathbf{D} + \mathbf{E}$. The main challenge is to identify matrices $\mathbf{M}$ and $\mathbf{N}$ such that $\mathbf{A} = \mathbf{M} + \mathbf{N}$ and the spectral radius $\rho(\mathbf{M^{-1}N}) < \mathbf{1}$. To this end, we introduce a diagonal matrix $\bar{\mathbf{E}}$ whose i^{th} diagonal entry equals:

$$\bar{E}_{ii} \equiv \alpha \sum_{j=1}^{n} |E_{ij}|, \tag{2.12}$$

for some positive constant α.

Proposition 2.1. *Let* $\mathbf{M} = \mathbf{D} + \bar{\mathbf{E}}$ *and* $\mathbf{N} = \mathbf{E} - \bar{\mathbf{E}}$. *Then, for* $\alpha \geq \frac{1}{2}$, $\rho(\mathbf{M^{-1}N}) < \mathbf{1}$.

Proof. By Theorem 2.5.3 of [26], to prove that $\rho(\mathbf{M^{-1}N}) < \mathbf{1}$, it is sufficient to show that $\mathbf{M} + \mathbf{N}$ and $\mathbf{M} - \mathbf{N}$ are positive definite. First, it is straightforward to verify that $\mathbf{H^T W H} = \mathbf{M} + \mathbf{N}$ is positive definite given Assumptions 1 and 2.

Second, we show that $\mathbf{M} - \mathbf{N}$ is positive definite. As a corollary to the Gershgorin Circle Theorem [101], it is sufficient to show that $\mathbf{M} - \mathbf{N}$ is strictly diagonally dominant with strictly positive diagonal entries. A matrix $\mathbf{A}$ is diagonal dominant if $|A_{ii}| \geq \sum_{j \neq i} |A_{ij}| \; \forall i$. To show this, we insert our choice for M and N to obtain $\mathbf{M} - \mathbf{N} = \mathbf{D} + \mathbf{2\bar{E}} - \mathbf{E}$. By definition, $\mathbf{D}$ is a diagonal matrix and $D_{ii} = (\mathbf{H^T W H})_{\mathrm{ii}} > 0$. Thus, given that $\alpha \geq \frac{1}{2}$,

$$\begin{aligned}
(\mathbf{D} + \mathbf{2\bar{E}} - \mathbf{E})_{\mathrm{ii}} &= D_{ii} + 2\bar{E}_{ii} - E_{ii} && \text{(2.13a)}\\
&> 2\,\alpha \sum_{j=1}^{n} |E_{ij}| - 0 && \text{(2.13b)}\\
&\geq \sum_{j \neq i} |E_{ij}| && \text{(2.13c)}\\
&= \sum_{j \neq i} |(\mathbf{D} + \mathbf{2\bar{E}} - \mathbf{E})_{\mathrm{ij}}|. && \text{(2.13d)}
\end{aligned}$$

Therefore, $\boldsymbol{M} - \boldsymbol{N}$ is a positive definite matrix. □

2.2.2 Proposed Algorithm and Analysis of Information Communication Requirements

We now study the information required to compute an individual phase angle estimate. Applying the iterative scheme of equation (2.8) to the state estimation problem with our choice of matrix splitting, we obtain

$$\boldsymbol{\theta}^{(t+1)} = -(\mathbf{D} + \bar{\mathbf{E}})^{\mathbf{-1}}(\mathbf{E} - \bar{\mathbf{E}})\boldsymbol{\theta}^{(\mathbf{t})} + (\mathbf{D} + \bar{\mathbf{E}})^{\mathbf{-1}}\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{z} \qquad (2.14)$$

Besides our choice of matrix splitting satisfying conditions necessary for convergence, we also note that the only matrix to invert, namely $(\mathbf{D} + \bar{\mathbf{E}})$, is diagonal and therefore easy to invert distributedly. In order to analyze the information communication of this iterative approach, we study the expression for the estimate of the unknown phase angle at a specific bus. From equation (2.14), we have

$$\theta_i^{(t+1)} = \frac{1}{D_{ii} + \bar{E}_{ii}}(\bar{E}_{ii}\boldsymbol{\theta}_i^{(t)} - [\mathbf{E}]_\mathbf{i}\boldsymbol{\theta}^{(\mathbf{t})} + [\mathbf{H}^\mathrm{T}\mathbf{W}]_\mathbf{i}\mathbf{z}) \tag{2.15}$$

We then utilize the particular structure of the power grid state estimation problem in order to verify the information from neighboring areas needed to calculate $\theta_i^{(t+1)}$. Each node is assumed to know measurements of its own phase angle, power injection, and local power flows. We introduce the following quantities:

$$\sigma_{ij} = \begin{cases} w_k, & z_k = \hat{P}_{ij} \\ 0, & \hat{P}_{ij} \notin \mathcal{Z} \end{cases}, \ \sigma_{i,P} = \begin{cases} w_k, & z_k = \hat{P}_i \\ 0, & \hat{P}_i \notin \mathcal{Z} \end{cases}, \ \sigma_{i,\theta} = \begin{cases} w_k, & z_k = \hat{\theta}_i \\ 0, & \hat{\theta}_i \notin \mathcal{Z} \end{cases}. \tag{2.16}$$

In addition, we define the set of “1-hop" neighbor nodes of node i as nodes that are not direct neighbors but share a common neighbor:

$$\mathcal{N}_i^\dagger \equiv \{j \mid j \notin \mathcal{N}_i, \mathcal{N}_i \cap \mathcal{N}_j \neq \emptyset\}. \tag{2.17}$$

By definition, we have that $\mathcal{N}_i \cap \mathcal{N}_i^\dagger = \emptyset$. For notational simplicity in what follows, let

$$\mathcal{N}_i^* = \mathcal{N}_i \cup \mathcal{N}_i^\dagger. \tag{2.18}$$

With this, we present the final expression for an estimate at an individual bus in terms of the local information available and only the subset of external information needed.

Proposition 2.2. *The updates to the distributed state estimates of equation (2.15) have the follow-*

ing form:

$$\theta_i^{(t+1)} = \frac{1}{A_{ii} + \alpha \sum\limits_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}}^*} |A_{ij}|} \Big\{ (\alpha \sum_{\mathrm{j}\in\mathcal{N}_i^*} |A_{ij}|)\theta_{\mathrm{i}}^{(\mathrm{t})} - \sum_{j\in\mathcal{N}_i^*} A_{ij}\theta_{\mathrm{j}}^{(\mathrm{t})} + \sigma_{i,P}(\sum_{j\in\mathcal{N}_i} B_{ij})\hat{P}_i +$$

$$\sum_{j\in\mathcal{N}_i} B_{ij}(\sigma_{ij}\hat{P}_{ij} - \sigma_{ji}\hat{P}_{ji} - \sigma_{j,P}\ \hat{P}_j) + \sigma_{i,\theta}\ \hat{\theta}_i \Big\} \tag{2.19}$$

where the entries of $A_{ij} =$

$$\begin{cases} \sigma_{i,\theta} + \sum\limits_{j\in\mathcal{N}_i} (\sigma_{ij} + \sigma_{ji} + \sigma_{j,P})B_{ij}^2 + & \\ \sigma_{i,P}\Big(\sum\limits_{j\in\mathcal{N}_i} B_{ij}\Big)^2 , & j = i \\ -(\sigma_{ij} + \sigma_{ji})B_{ij}^2 + \sum\limits_{l\in\mathcal{N}_i\cap\mathcal{N}_j} \sigma_{l,P}B_{il}B_{jl} & \\ -B_{ij}\Big(\sigma_{i,P}\sum\limits_{l\in\mathcal{N}_i} B_{il} + \sigma_{j,P}\sum\limits_{l\in\mathcal{N}_j} B_{jl}\Big) , & j \in \mathcal{N}_i \\ \sum\limits_{l\in\mathcal{N}_i\cap\mathcal{N}_j} \sigma_{l,P}B_{il}B_{jl} , & j \in \mathcal{N}_i^\dagger \\ 0, & \textit{else.} \end{cases} \tag{2.20}$$

Proof. Expanding equation (2.15) using the definitions for matrices $\mathbf{D}, \mathbf{E}$, and $\bar{\mathbf{E}}$ from equations (2.10)-(2.12), we obtain an expression for $\theta_i^{(t+1)}$ soley in terms of entries of the matrices $\mathbf{H}$ and $\mathbf{A} = \mathbf{H}^{\mathbf{T}}\mathbf{W}\mathbf{H}$,

$$\theta_i^{(t+1)} = \frac{1}{A_{ii} + \alpha \sum\limits_{j\neq i} |A_{ij}|} \Big\{ (\alpha \sum_{j\neq i} |A_{ij}|)\theta_i^{(t)} - \sum_{j\neq i} A_{ij}\theta_j^{(t)}$$

$$+ \sum_{k=1}^{m} H_{ki}w_k z_k \Big\} \tag{2.21}$$

Since $\mathbf{W}$ is diagonal, we have that

$$A_{ij} = \sum_{k=1}^{m} w_k H_{ki} H_{kj}. \tag{2.22}$$

From equations (2.2)-(2.4), we see that the topology of the power grid network and the measurement model induces a sparsity on the entries of $\mathbf{A}$. We determine the entries A_{ij} by considering four cases.

1. Case 1: $(i = j)$

   According to equations (2.2)-(2.3), for measurements of power flow on a line incident with node i or a power injection at a neighboring node, $H_{ki} = \pm B_{ij}$, so $H_{ki}^2 = B_{ij}^2$. For measurement of the power injection at node i, then $H_{ki}^2 = (\sum_{j \in \mathcal{N}_i} B_{ij})^2$. According to equation (2.4), for a measurement of the voltage phase angle at node i, $H_{ki}^2 = 1$. All other measurements k in the network will not contribute to the sum in equation (2.22) since $H_{ki} = 0$. Therefore, we see that we need only consider measurements local to node i or power injections at its neighbors. Since in general not all possible power flows, power injections, and voltage phase angles are measured, we use the quantities defined in equation (2.16) to include only the available measurements.

2. Case 2: $(j \in \mathcal{N}_i)$

   Similar logic applies as in Case 1 except in the instance of a power injection at node j, we need the following information about node j's incident edges, $\sum_{l \in \mathcal{N}_j} B_{jl}$. In addition, we must include possible contributions from a power injection at a mutual neighbor of nodes i and j, which does not require any communication of information that is neither local to node

i nor to node j.

3. Case 3: ($j \in \mathcal{N}_i^\dagger$)

   In this case, we consider "1-hop" neighbor nodes of i. If nodes i and j are not direct neighbors but share a neighbor l with a power injection measurement k, then $H_{ki}H_{kl} = B_{il}B_{jl} > 0$, yielding a non-zero entry for A_{ij}.

4. Case 4: Otherwise ($j \notin \mathcal{N}_i^*, j \neq i$)

   If j does not satisfy the conditions for the first three cases, then we must have either H_{ki} or H_{kj} be zero for all possible measurements.

With this, we obtain the entries of A_{ij} in terms of the available measurements, their variances, the network connectivity, and the bus susceptance parameters. Therefore in equation (2.21), instead of requiring *all* entries $\{A_{ij} | j \neq i\}$, we only require $\{A_{ij} | j \in \mathcal{N}_i^*\}$. Furthermore, the only measurements that contribute to the term $\sum_{k=1}^{m} H_{ki} w_k z_k$ are those locally available to node i and power injections at its neighbors. These simplifications provide the final result in equation (2.19). □

From this lemma, we know that if there are only power flow and phase angle measurements at node i (i.e. no power injection measurement), then node i needs to communicate only its current estimate $\theta_i^{(t)}$ to its neighbors. If there is a power injection measured at node i, then the following additional information must be communicated to its neighbors:

1. Value of measured power injection, $\hat{P}_i$

2. The set of neighbors' state estimates $\{\theta_j^{(t)}\}_{j \in \mathcal{N}_i}$ must be shared with each neighbor.

Algorithm 2.1 Matrix-Splitting Based Distributed Linear State Estimation

```
Initialization : Node $i$ has access to local measurements. $\forall\ 1 \leq i \leq n, \theta_i^{(0)} = 0$. If $\hat{P}_i \in \mathcal{Z}$, node $i$ sends $\hat{P}_i, w_{k|z_k=\hat{P}_i}$, and $\{B_{ij}\}_{j\in\mathcal{N}_i}$ to nodes $j \in \mathcal{N}_i$.
for $t := 0$ to $T$ do
    for $i := 1$ to $n$ do
        $\theta_i^{(t+1)} = f(\theta_i^{(t)}, \{\theta_j^{(t)}\}_{j\in\mathcal{N}_i^*})$ from equation (2.19). Node $i$ sends $\theta_i^{(t+1)}$ to nodes $j \in \mathcal{N}_i$.
    end for
    if $\hat{P}_i \in \{z\}$ then
        Node $i$ sends $\{\theta_j^{(t+1)}\}_{j\in\mathcal{N}_i}$ to nodes $j \in \mathcal{N}_i$.
    end if
end for
```

It is noted that only the phase angle estimates change with time and need to be communicated at each time step. Power injections are communicated once during initialization. The line susceptance parameters and measurement weights are assumed to be known a priori.

We illustrate the information exchange for nodes with a power injection measurement using the example in Figure 2.2. Power injections are measured at nodes i and l.

At initialization,

- Node i sends $\hat{P}_i$ to its neighboring nodes j, k, and l.
- Node i receives $\hat{P}_l$ from node l.

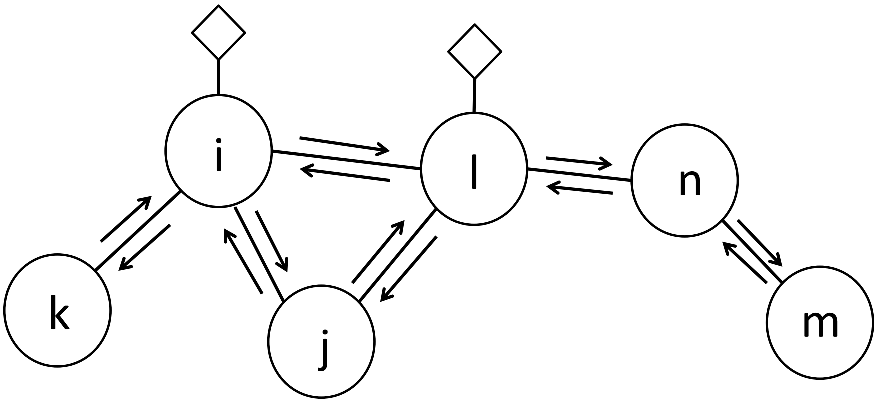


Figure 2.2: Example to illustrate information exchange of Algorithm 2.1. Neighboring nodes are connected via an edge in the graph. There are power injection measurements at nodes i and l, signified by a diamond-shaped sensor. Communication occurs directly between neighbors.

At iteration t, after each node has calculated its next estimate using equation (2.19),

- Node i sends $\theta_i^{(t+1)}, \theta_j^{(t+1)}, \theta_k^{(t+1)}$, and $\theta_l^{(t+1)}$ to its neighboring nodes j, k, and l.
- Node i receives $\theta_j^{(t+1)}, \theta_l^{(t+1)}$, and $\theta_n^{(t+1)}$ from node l. Node i also receives $\theta_k^{(t+1)}$ from node k and $\theta_j^{(t+1)}$ from node j.

2.2.3 Convergence Analysis

Under *Assumption 2*, there is a unique solution given by equation (2.7), which we denote here as $\boldsymbol{\theta}^*$. In Proposition 1, we showed that $\rho(\mathbf{M^{-1}N}) < \mathbf{1}$. Because the iterative scheme in equation (2.8) forms a discrete linear dynamic system, according to (Thm 6.1) [30], the iterative scheme in (2.8) exponentially converges to the solution $\boldsymbol{\theta}^*$. The convergence speed is determined by $\rho(\mathbf{M^{-1}N})$. Formally speaking, we have the following theorem about the convergence:

Theorem 2.1. *The distributed state estimation scheme as described in Algorithm 2.1 exponentially* *converges to the optimal solution in (2.5). The convergence speed is determined by* $\rho(\mathbf{M^{-1}N})$.

2.2.4 Proof of Some Properties of the Spectral Radius

As a starting point towards an analytical analysis of $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$, assume there are only power flow and PMU measurements available. In this setting, we can show that $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ is

1. an increasing function of α and
2. a decreasing function of the number of diagonal blocks in the splitting of $\boldsymbol{A}$.

Without power injection measurements, the entries of $\boldsymbol{A}$ are given by

$$A_{ij} = \begin{cases} \sigma_{i,\theta} + \sum\limits_{l \in \mathcal{N}_i} (\sigma_{P(i,l)} + \sigma_{P(l,i)}) B_{il}^2 \ , & j = i \\ -(\sigma_{P(i,j)} + \sigma_{P(j,i)}) B_{ij}^2 \ , & j \in \mathcal{N}_i \\ 0 \ , & \text{otherwise,} \end{cases} \tag{2.23}$$

where B_{ij} is the susceptance of the transmission line connecting buses i and j.

From this we see that $\boldsymbol{A}$ has positive diagonal entries and non-positive off-diagonal entries. Furthermore, since $\boldsymbol{A}$ is symmetric and positive definite, $\boldsymbol{A}$ is a *Stieljes* matrix. Stieljes matrices have the property that their inverses are non-negative, $\boldsymbol{A}^{-1} \geq \mathbf{0}$, (Cor. 3.24) [102]. From Definition 3.28 in [102], for $n \times n$ real matrices, $\boldsymbol{A}, \boldsymbol{M}$, and $\boldsymbol{N}$, $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ is a regular splitting of the matrix $\boldsymbol{A}$ if $\boldsymbol{M}$ is nonsingular with $\boldsymbol{M}^{-1} \geq \mathbf{0}$ and $\boldsymbol{N} \geq \mathbf{0}$.

Proposition 2.3. *The splitting in Proposition 2.1 is a regular splitting for* $\alpha \geq 0$.

Proof. Since $\boldsymbol{A}$ is a Stieljes matrix, $\boldsymbol{D}$ is also a Stieljes matrix. Upon adding a diagonal matrix with strictly positive entries to a Stieljes matrix, the matrix remains Stieljes. Therefore, for $\alpha \geq 0$, $\boldsymbol{M} = \boldsymbol{D} + \alpha\bar{\boldsymbol{E}}$ is a Stieljes matrix, and we have $\boldsymbol{M}^{-1} \geq \mathbf{0}$ and $\boldsymbol{N} \geq \mathbf{0}$. □

To show the desired properties of $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$, we utilize the following theorem from [102, Theorem 3.32].

Theorem 2.2 (Varga)**.** *Let $\boldsymbol{A} = \boldsymbol{M_1} - \boldsymbol{N_1} = \boldsymbol{M_2} - \boldsymbol{N_2}$ be two regular splittings of $\boldsymbol{A}$, where $\boldsymbol{A}^{-1} \geq \mathbf{0}$. If $\mathbf{0} \leq \boldsymbol{N_1} \leq \boldsymbol{N_2}$, then*

$$0 \leq \rho({\boldsymbol{M_1}}^{-1}\boldsymbol{N_1}) \leq \rho({\boldsymbol{M_2}}^{-1}\boldsymbol{N_2}) < 1. \tag{2.24}$$

With this result, we have the following proposition that analytically characterizes the rate of convergence with respect to the matrix splitting parameter α and with respect to block size.

Proposition 2.4. *In the DC setting and when no power injection measurements are present, $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ is an increasing function of α for $\alpha \geq 0$. The rate of convergence of the iterative algorithm increases as block size increases or equivalently the number of blocks decreases.*

Proof. Let $0 \leq \alpha < \alpha'$ and let $\boldsymbol{N_1} = \alpha\bar{\boldsymbol{E}} - \boldsymbol{E}$ and $\boldsymbol{N_2} = \alpha'\bar{\boldsymbol{E}} - \boldsymbol{E}$, where $\bar{\boldsymbol{E}}$ is defined as in equation (2.12). Then, $\boldsymbol{N_1} < \boldsymbol{N_2}$, so from Theorem 2.2 we have $\rho({\boldsymbol{M_1}}^{-1}\boldsymbol{N_1}) \leq \rho({\boldsymbol{M_2}}^{-1}\boldsymbol{N_2})$. This shows that the rate of convergence increases as α decreases. Consider a splitting with $\boldsymbol{M}_1 = \boldsymbol{D} + \alpha\bar{\boldsymbol{E}}$ and $\boldsymbol{M}_2 = \boldsymbol{D}' + \alpha\bar{\boldsymbol{E}}'$, where $\bar{\boldsymbol{E}}'$ is diagonal with entries given by $\bar{E}'_{ii} = \sum_{j=1}^{n} |A'_{ij}|$. Let

$\boldsymbol{D} = \mathrm{diag}(\boldsymbol{A}_{11}, \ldots, \boldsymbol{A}_{ii}, \ldots, \boldsymbol{A}_{NN})$ and $\boldsymbol{D'} = \mathrm{diag}(\boldsymbol{A}_{11}, \ldots, \boldsymbol{A'}_{ii}, \ldots, \boldsymbol{A}_{NN})$, where

$$\boldsymbol{A}_{ii} = \begin{bmatrix} \boldsymbol{A}_{ii}^{(1,1)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_{ii}^{(2,2)} \end{bmatrix}, \tag{2.25}$$

and

$$\boldsymbol{A'}_{ii} = \begin{bmatrix} \boldsymbol{A}_{ii}^{(1,1)} & \boldsymbol{A}_{ii}^{(1,2)} \\ \boldsymbol{A}_{ii}^{(2,1)} & \boldsymbol{A}_{ii}^{(2,2)} \end{bmatrix}. \tag{2.26}$$

Then, we have $\boldsymbol{N}_1 = \boldsymbol{A} - \boldsymbol{D} - \alpha\bar{\boldsymbol{E}}$ and $\boldsymbol{N}_2 = \boldsymbol{A} - \boldsymbol{D'} - \alpha\bar{\boldsymbol{E}}'$. Since the off-diagonal entries of $\boldsymbol{A}$ are non-positive, by construction $\boldsymbol{A}_{ii} \leq \boldsymbol{A'}_{ii}$, so we have $\boldsymbol{N}_1 \leq \boldsymbol{N}_2$. From Theorem 2.2, we have $\rho({\boldsymbol{M_1}}^{-1}\boldsymbol{N_1}) \leq \rho({\boldsymbol{M_2}}^{-1}\boldsymbol{N_2})$. From equations (2.25) and (2.26), we see that the splitting $\boldsymbol{A} = \boldsymbol{M}_1 + \boldsymbol{N}_1$ results from dividing a given block into two smaller blocks. Since the selection of the block $\boldsymbol{A'}_{ii}$ was arbitrary, we see that decreasing the block size or equivalently increasing the number of blocks decreases the rate of convergence. □

This shows an interesting connection between the rate of convergence and the communication scheme, or degree to which the algorithm is distributed.

2.3 Numerical Results

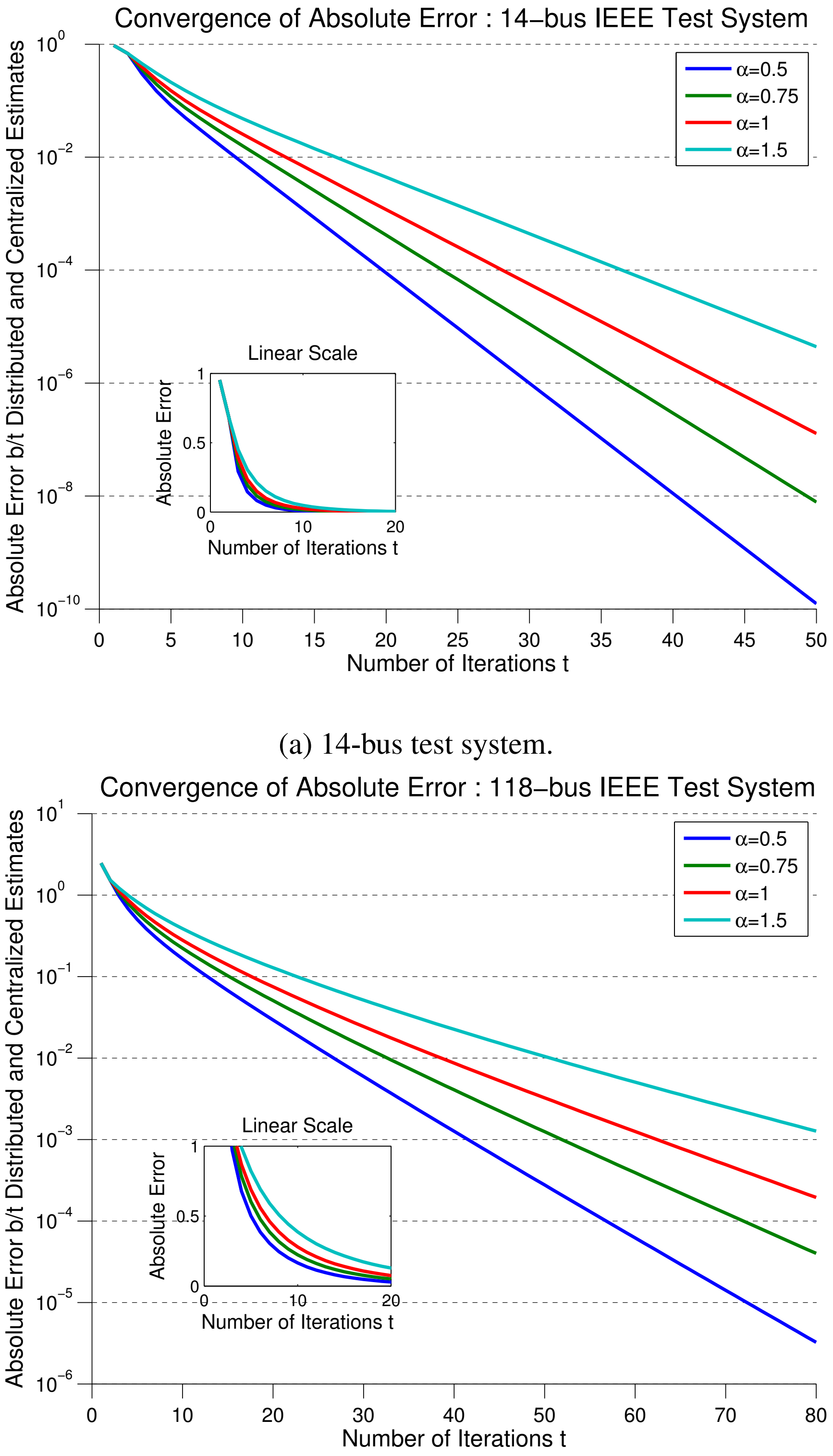
We study the performance of our algorithm by calculating the error between the distributed estimate and the centralized optimal estimate at each iteration t, $\delta^t \equiv ||\boldsymbol{\theta}_d^{(t)} - \boldsymbol{\theta}_c^*||$. Figure 2.3 demonstrates the exponential convergence of the distributed estimates for the IEEE 14-bus and 118-bus systems respectively. We study the convergence for different values of the parameter α

Table 2.1: 118-bus measurement configuration. The power flow measurements are taken at all branches except those listed as missing power flows.

Measurement Type	**Locations (Bus Index)**
Missing Power Flows	(2,12), (9,10), (24,72), (93,94)
Power Injections	1, 2, 3, 5, 18, 19, 21, 22, 25, 33, 34, 35, 36, 55, 56, 69, 71, 72, 77, 80, 81, 86, 88, 90, 98, 106,107 110, 114, 118
Missing voltage phase angles	3, 5, 14, 15, 18, 19, 20, 21, 26, 33, 38, 39, 41, 44, 45, 51, 63, 64, 67, 68, 51, 63, 64, 67, 68, 63, 64, 67, 68, 69, 73, 74, 75, 76, 79, 82, 84, 89, 92, 94, 96, 97, 99, 100, 101, 105, 106, 107

in equation (2.12) and empirically find $\alpha = \frac{1}{2}$ to be optimal as expected from Proposition 2.4. For the 14-bus system (a), using the measurement configuration in Figure 2.1 with $\alpha = \frac{1}{2}$, the convergence is exponential with rate determined by $\rho(\mathbf{M^{-1}N}) = 0.64$. For the 118-bus system (b), using the measurement configuration in Table 2.1 with $\alpha = \frac{1}{2}$, the convergence is exponential with rate determined by $\rho(\mathbf{M^{-1}N}) = 0.87$.

The values of the power flows and power injections in the simulation are on the order of 10^{-1} p.u. The measurements are perturbed by additive zero-mean Gaussian noise $\mathcal{N}(0, \sigma^2)$ with $\sigma = 0.01$. As stated in Theorem 2.1, the convergence of the distributed algorithm is governed by the spectral radius, $\rho(\mathbf{M^{-1}N})$. The value of $\rho(\mathbf{M^{-1}N})$ is sensitive to the measurement configuration, the choice of weighting matrix $\mathbf{W}$, and the parameter α of the matrix splitting. We used the identity matrix for $\mathbf{W}$ in these tests. From Figure 2.3, it is demonstrated that the lower value of $\rho(\mathbf{M^{-1}N})$ in the 14-bus system leads to faster convergence than in the 118-bus system.



(a) 14-bus test system.

(b) 118-bus test system.

Figure 2.3: Convergence is shown for different values of the parameter α from equation (2.12).

2.4 Conclusion

In this chapter, we have proposed a new fully distributed state estimation algorithm based on a matrix splitting iterative approach. Attractive features of the algorithm include the limited amount of information that needs to be shared between neighboring nodes and that each node needs only to store and compute its local estimate. Future research directions include analytically characterizing the spectral radius associated with the matrix splitting in terms of the measurement configuration and extending the algorithm to detect and identify bad data in a distributed manner. In the next chapter, we will generalize the approach to the AC state estimation setting.

Chapter 3

A Distributed Gauss-Newton Method for Power System State Estimation

IN THIS CHAPTER, we extend the results for linear state estimation in Chapter 2 to a nonlinear measurement model. There is substantial literature on developing distributed methods [82] for general optimization problems, such as consensus-based and dual-based gradient and Newton-type methods. However, there has been a lack of work on developing distributed Newton-type methods for state estimation in power grids. Due to the nonconvexity of the power system state estimation problem, we use the Gauss-Newton method [85]. In comparison to gradient-based methods, Newton-type methods are advantageous with respect to convergence rate, which is usually quadratic. The difficulty is that Newton-type methods require solving a linear system at each iteration. This presents a challenge for developing a distributed method since in general global information of the matrix entries is needed. Furthermore, when the system size is large, solving the set of linear equations may be time-consuming or even prohibitive. To overcome this challenge, in our work we explore the use of matrix-splitting techniques [102, 106]. This allows us to exploit inherent sparse structure in power systems in order to calculate the next Gauss-Newton iteration in a distributed way.

A similar method in [72] uses an approximate block Jacobi method, a kind of matrix splitting, for distributed state estimation. Our work is distinct in that we do not ignore the boundary terms containing information about neighboring areas. Without such boundary terms, a distributed algorithm does not require communication. Since the power network is an interconnected system, ignoring such boundary terms leaves out important information. Furthermore, theoretical guarantees for the convergence of the approximate block Jacobi iterative scheme are not provided. The work in this chapter more fully explores some of the ideas first suggested in [72].

In our algorithm, each control area calculates the state estimates of its local buses, and communication is carried out only between neighboring areas. The contributions of this work include a new fully distributed Gauss-Newton method for power system state estimation with the following features:

1. The method incorporates both traditional SCADA (supervisory control and data acquisition) system measurements of power injections and flows, as well as PMU measurements.

2. Each control area only requires local information and a limited exchange of information with neighboring control areas in order to estimate its state, eliminating the need for a central processor.

3. Each control area only needs to hold an estimate toward its own state rather than an estimate of the global state of the system. This saves a large amount of communication and memory resources.

The principal intent of our algorithm is to be applied in a distributed computation environment with information being exchanged between different control areas across potentially large geographic distances. Our algorithm is well-suited to such a setting since it utilizes only neighbor-to-neighbor communication. However, the algorithm is also applicable for running on a parallel computing environment, where measurements are aggregated at a central location [95]. Indeed to test our algorithm, we use a computer cluster where each node in the cluster is treated as a control area. In this case, the communication time is not reflective of a geographically dispersed setting, but it gives a useful indication of how the communication and computation time requirements scale.

The chapter is outlined as follows. In Section 3.1, we present the mathematical problem statement and introduce the application of Newton's method to power system state estimation. In Section 3.2, we present our distributed state estimation algorithm with analysis of its communication requirements. In addition, we discuss the convergence properties of our algorithm. In Section 3.3, numerical simulations are used to demonstrate the effectiveness of our method.

Notations: We use v_k to denote the kth entry of a vector $\boldsymbol{v}$. The (i,j)th entry of a matrix $\boldsymbol{M}$ is given by M_{ij}. The transpose of a vector or matrix $\boldsymbol{X}$ is denoted by $\boldsymbol{X}^T$. The matrix inequality $\boldsymbol{M} > 0$ is to be interpreted as each element of $\boldsymbol{M}$ being positive. In contrast, we use $\boldsymbol{M} \succ 0$ to denote that $\boldsymbol{M}$ is positive definite.

3.1 Problem Statement

We consider a multi-area interconnected power network, denoted by an undirected graph $(\mathcal{N}, \mathcal{E})$ with a set $\mathcal{N} \triangleq \{1, 2, \ldots, n\}$ of buses and a set $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ of transmission lines connecting the buses. The goal of state estimation in power systems is to infer the unknown voltages (phase angle and magnitude) at each bus, $\boldsymbol{x}^T = [\theta_1 \;\; V_1 \ldots \theta_n \;\; V_n]$, from a set of noisy measurements of the system. The power network is partitioned into N non-overlapping regions, called control areas. Decentralized state estimation allows each control area to estimate its local state by exchanging information with neighboring control areas. Figure 3.1 provides an example of the IEEE 14-bus test system under a particular control area partitioning and measurement configuration [61].

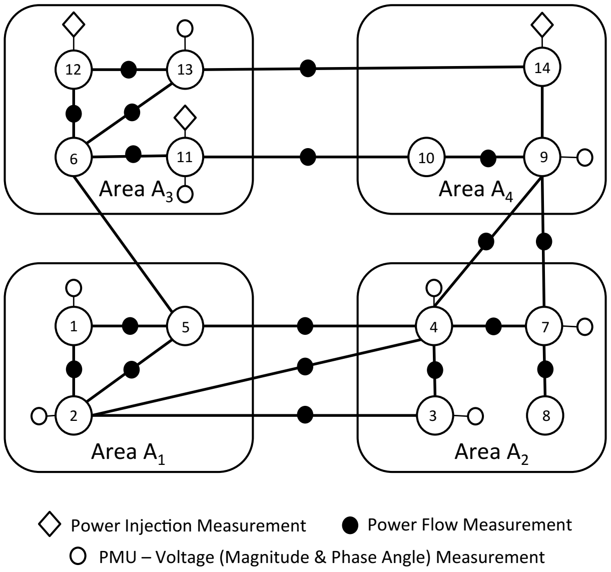


Figure 3.1: Multi-area IEEE 14-bus test system partitioned into $N = 4$ control areas with location and types of measurements used [61].

There are two typical power measurement systems. One is the traditional SCADA measurements including power flows along transmission lines and power injections at buses; the other system uses PMUs to measure the voltages and currents directly. We consider the following measurements: 1) the real and reactive power flow along the transmission line between buses i and j, denoted by $\widehat{P}_{ij}$ and $\widehat{Q}_{ij}$; 2) the real and reactive power injection at bus i, denoted by $\widehat{P}_i$ and $\widehat{Q}_i$; and 3) the voltage phase angle and magnitude at bus i, denoted by $\widehat{\theta}_i$ and $\widehat{V}_i$. Branch current phasor measurements are also available from the PMUs. As a future enhancement, the branch current measurements can be incorporated using techniques from [7,21,62] for developing hybrid state estimators that incorporate both voltage and current phasor measurements with traditional SCADA measurements. SCADA scan rates are approximately once every 2-6 seconds, whereas PMU measurements update about 30 times per second [41]. The intent of our algorithm is to use the most recent measurements available from the PMUs and from the SCADA system for each state estimation run. We consider a static setting rather than a dynamic one, treating each measurement set as a separate snapshot in time. In Chapter 4, we consider the dynamic setting.

Due to the deployment cost of SCADA and PMU sensors, measurements of power flow, power injection, and voltage phasors are only available at a subset of the buses and transmission lines in the system. We denote the set of measurements as $\mathcal{Z}$ and the ordered vector of measurements as $\boldsymbol{z}$. The AC model relates the measurements $\boldsymbol{z}$ and the unknown state $\boldsymbol{x}$,

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{e}, \tag{3.1}$$

where $\boldsymbol{e} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R})$ is zero-mean Gaussian random noise with covariance matrix $\boldsymbol{R}$. AC state estimation solves the following optimization problem $\arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$, where

$$f(\boldsymbol{x}) \overset{\text{def}}{=} (\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}))^T \boldsymbol{W} (\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})). \tag{3.2}$$

The following assumption is made on the weighting matrix, $\boldsymbol{W}$:

Assumption 3.1. *The weighting matrix $\boldsymbol{W}$ is diagonal with entries $W_{kk} \overset{def}{=} w_k > 0$.*

If the inverse covariance matrix is used as the weighting matrix, this is equivalent to assuming that noise from different measurements is independent (*i.e.*, $\boldsymbol{R}$ is diagonal). Our method can be extended to a scenario where the noise values associated with measurements from within a single control area have non-zero covariance.

Given an initial point, $\boldsymbol{x}^{(0)}$, Newton's method uses an iterative scheme to minimize $f(\boldsymbol{x})$:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \Delta\boldsymbol{x}^{(k)}. \tag{3.3}$$

At each iteration k, Newton's method minimizes the second-order approximation to f about $\boldsymbol{x}^{(k)}$. The Newton step, $\Delta\boldsymbol{x}^{(k)}$, is given by solving the following linear system

$$[\nabla^2 f(\boldsymbol{x}^{(k)})]\Delta\boldsymbol{x}^{(k)} = \nabla f(\boldsymbol{x}^{(k)}), \tag{3.4}$$

where ∇f and $\nabla^2 f$ are the Jacobian and Hessian functions of the objective function. Since $f(\boldsymbol{x})$ is a non-convex function, we use the Gauss-Newton method which employs a positive-definite approximation to $\nabla^2 f(\boldsymbol{x})$ [85]. We denote this by $\tilde{\nabla}^2 f(\boldsymbol{x}) = \boldsymbol{J}^T(\boldsymbol{x})\boldsymbol{W}\boldsymbol{J}(\boldsymbol{x})$, where the measurement Jacobian is given by

$$\boldsymbol{J}(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial h_1(\boldsymbol{x})}{\partial \theta_1} & \frac{\partial h_1(\boldsymbol{x})}{\partial V_1} & \cdots & \frac{\partial h_1(\boldsymbol{x})}{\partial \theta_n} & \frac{\partial h_1(\boldsymbol{x})}{\partial V_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial h_m(\boldsymbol{x})}{\partial \theta_1} & \frac{\partial h_m(\boldsymbol{x})}{\partial V_1} & \cdots & \frac{\partial h_m(\boldsymbol{x})}{\partial \theta_n} & \frac{\partial h_m(\boldsymbol{x})}{\partial V_n} \end{pmatrix}. \tag{3.5}$$

Depending on the quality of the approximation to the Hessian, the Gauss-Newton method may not have the same quadratic convergence properties as the Newton method. In cases where $\tilde{\nabla}^2 f(\boldsymbol{x})$ approximates $\nabla^2 f(\boldsymbol{x})$ well, the Gauss-Newton method will show comparable performance to the Newton method without the computational burden of calculating the full objective function Hessian. Using the Gauss-Newton approximation, the Newton update (3.4) becomes

$$\boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{W} \boldsymbol{J}(\boldsymbol{x}) \Delta \boldsymbol{x} = -\boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{W}(\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})), \tag{3.6}$$

where we suppress iteration label k to lighten notation. Solving this linear system yields the Gauss-Newton step, $\Delta \boldsymbol{x}^{(k)}$, needed to produce the next iterate in (3.3). For realistic power systems, this results in a large linear system, that is challenging to solve in real-time. The goal of this work is to solve the linear system (3.6) in a distributed way, utilizing the sparsity of the system to ensure limited communication requirements.

3.2 Distributed Gauss-Newton Algorithm

3.2.1 Overview of Algorithm

The aim of distributed state estimation is for each control area to estimate its local state. This requires a distributed solution to the linear system in (3.6). Rather than using standard direct matrix inversion methods, such as Gaussian elimination (*i.e.*, LU factorization), Gauss-Jordan elimination, or Cholesky factorization [49], we develop an iterative method based on matrix splitting [102]. At every iteration, each control area exchanges information about its local state estimate with neighboring areas. The local state estimate is then updated based on information received from neighboring areas. Though direct methods exactly solve linear systems, up to rounding error, in a finite number of steps, for large systems, they often suffer from prohibitively large storage and computation requirements. In contrast, iterative methods can have a significant advantage over direct methods if they rapidly converge to a sufficiently accurate solution [49].

Control areas have access to their own local measurements and state estimates of local buses but not to system-wide measurements and estimates. Measurements at neighboring buses in other control areas will be relevant to the control area's state estimation (*i.e.*, the measurement has a non-zero partial derivative with respect to at least one of the local states). In particular, a measurement of power flow along a transmission line connecting two control areas requires those control areas to share their bordering bus state estimates with one another. Likewise, for a bus with a power injection connected to another control area, all of that buses neighbors' estimates must be shared with the neighboring control area.

3.2.2 Matrix Splitting for Distributed AC State Estimation

We propose a new multi-area state estimation algorithm based on a matrix-splitting technique that allows us to calculate the Gauss-Newton step, $\Delta \boldsymbol{x}$, in a distributed way. This method is inspired by the use of matrix-splitting for developing a distributed Newton method for the Network

Utility Maximization (NUM) problem of Wei *et al.* in [106]. We introduce the following notation

$$\boldsymbol{A} \overset{\text{def}}{=} \boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{W} \boldsymbol{J}(\boldsymbol{x}) \tag{3.7}$$

$$\boldsymbol{b} \overset{\text{def}}{=} -\boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{W} (\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})). \tag{3.8}$$

As detailed in Chapter 2, writing the linear system in (3.6) as $\boldsymbol{A}\Delta\boldsymbol{x} = \boldsymbol{b}$, the idea behind matrix splitting is to write $\boldsymbol{A}$ as the difference of an invertible matrix, $\boldsymbol{M}$, and a matrix $\boldsymbol{N}$, (*i.e.*, $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$) [102]. Recall for an arbitrary $\Delta\boldsymbol{x}^0 \in \mathbb{R}^n$, the following scheme provides an iterative solution to solving (3.6):

$$\Delta\boldsymbol{x}^{t+1} = \boldsymbol{M}^{-1}\boldsymbol{N}\Delta\boldsymbol{x}^t + \boldsymbol{M}^{-1}\boldsymbol{b}. \tag{3.9}$$

Since the Gauss-Newton method is itself an iterative method for minimizing the weighted least-squares objective function, matrix-splitting introduces an inner-loop of iterations t for each Gauss-Newton (*i.e.* outer-loop) iteration k. Numerical tests in Section 3.3 demonstrate the favorable runtime of our algorithm despite this nested loop structure. A flowchart providing a practical overview of our algorithm is given in Figure 3.2.

Our contribution is to design the matrix splitting so that the iterative scheme in (3.9) converges and is easily distributed. Recall from Chapter 2 the sequence in (3.9) converges if and only if the spectral radius $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ is strictly less than 1. For details of convergence, please see Section 3.2.4. To facilitate distributed processing, each control area should calculate its next Gauss-Newton iterate using local information and a limited amount of communication with neighboring areas. To this end, we consider a splitting of $\boldsymbol{A}$ into a block diagonal matrix, $\boldsymbol{D}$, and an off-diagonal matrix $\boldsymbol{E}$. The entries of $\boldsymbol{D}$ correspond to local information, and the entries of $\boldsymbol{E}$ correspond to

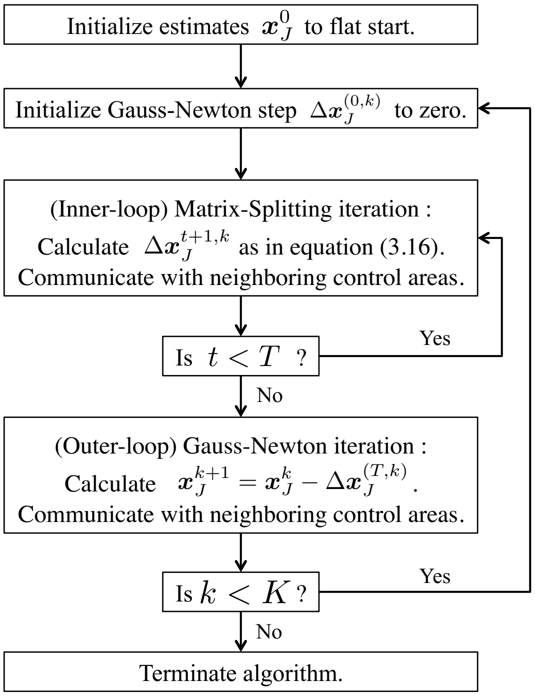

Initialize estimates $\boldsymbol{x}_J^0$ to flat start.
Initialize Gauss-Newton step $\Delta\boldsymbol{x}_J^{(0,k)}$ to zero.
(Inner-loop) Matrix-Splitting iteration :
Calculate $\Delta\boldsymbol{x}_J^{t+1,k}$ as in equation (3.16).
Communicate with neighboring control areas.
Is $t < T$?
Yes
No
(Outer-loop) Gauss-Newton iteration :
Calculate $\boldsymbol{x}_J^{k+1} = \boldsymbol{x}_J^k - \Delta\boldsymbol{x}_J^{(T,k)}$.
Communicate with neighboring control areas.
Is $k < K$?
Yes
No
Terminate algorithm.

Figure 3.2: Outline of proposed state estimation algorithm detailing inner- and outer-loop structure for control area J.

information required from other control areas.

Let the number of buses in control area A_1 be denoted n_1, and without loss of generality, let control area A_1 contain buses with node labels $S_1 = \{1, \ldots, n_1\}$. Similarly let the buses of control area A_2 have labels $S_2 = \{n_1 + 1, \ldots, n_1 + n_2\}$ and so forth for consecutive control areas. The matrix $\boldsymbol{A}$ as defined in (3.7) can be decomposed into the sum of a block-diagonal matrix, $\boldsymbol{D}$, and a matrix containing the remaining off-diagonal entries $\boldsymbol{E}$. The consecutive re-labeling of the node indices described above allows for all entries of $\boldsymbol{A}$ corresponding to buses within the same control area to be contained within a single diagonal block. Given a row index i of $\boldsymbol{x}$, we use $n(i)$ to denote the underlying bus index.[1] Specifically, let

$$D_{ij} = \begin{cases} A_{ij} & \text{if nodes } n(i) \text{ and } n(j) \\ & \text{belong to the same control area} \\ 0 & \text{otherwise} \end{cases}, \tag{3.10}$$

$$E_{ij} = \begin{cases} A_{ij} & \text{if nodes } n(i) \text{ and } n(j) \\ & \text{belong to different control areas} \\ 0 & \text{otherwise} \end{cases}, \tag{3.11}$$

yielding $\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{E}$. The key is to identify matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ such that $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ and the spectral radius $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) < 1$. To ensure convergence, we introduce a diagonal matrix $\bar{\boldsymbol{E}}$ whose ith diagonal entry equals

$$\bar{E}_{ii} \overset{\text{def}}{=} \sum_{j \neq i} |A_{ij}|. \tag{3.12}$$

We propose the following matrix-splitting design

$$\boldsymbol{M} = \boldsymbol{D} + \alpha \bar{\boldsymbol{E}} \tag{3.13}$$

$$\boldsymbol{N} = \alpha \bar{\boldsymbol{E}} - \boldsymbol{E}, \tag{3.14}$$

where α is a scalar parameter. In Proposition 3.1, it is shown for $\alpha \geq \frac{1}{2}$, the matrix-splitting

[1] For example, in a two bus system, $\boldsymbol{x}^T = [\theta_1 \; V_1 \; \theta_2 \; V_2]$, and

$$n(i) = \begin{cases} 1 & \text{if } i = 1, 2 \\ 2 & \text{if } i = 3, 4 \end{cases}.$$

iterative scheme in (3.9) converges to the centralized solution using the splitting designed in (3.13)-(3.14). The centralized solution is given by directly, as opposed to iteratively, solving system (3.6).

3.2.3 Proposed Algorithm and Analysis of Information Communication Requirements

Besides our choice of matrix splitting satisfying conditions necessary for convergence, we also note that the only matrix to invert, namely $(\boldsymbol{D} + \alpha\bar{\boldsymbol{E}})$, is block diagonal and therefore can be inverted locally within each area without any communication between neighboring control areas since the inverse of a block diagonal matrix remains block-diagonal.

We address the question of which information needs to be communicated for each control area to calculate its local Gauss-Newton update, $\Delta\boldsymbol{x}^t = [\Delta\boldsymbol{x}_1^t\ \Delta\boldsymbol{x}_2^t\ \ldots\ \Delta\boldsymbol{x}_N^t]$, where there are a total of N control areas. Similarly, we can partition $\boldsymbol{b}$ from equation (3.8) into different components corresponding to each control area as $\boldsymbol{b} = [\boldsymbol{b}_1\ \ldots \boldsymbol{b}_N]$. To illustrate the local computation required for the Gauss-Newton update, consider a network with two control areas. Then, the matrix-splitting iterative updates in equation (3.9) become

$$\begin{aligned}\begin{bmatrix}\Delta\boldsymbol{x}_1^{(t+1)}\\ \Delta\boldsymbol{x}_2^{(t+1)}\end{bmatrix} &= \begin{bmatrix}\boldsymbol{M}_1^{-1} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{M}_2^{-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{N}_{11} & \boldsymbol{N}_{12}\\ \boldsymbol{N}_{21} & \boldsymbol{N}_{22}\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{x}_1^{(t)}\\ \Delta\boldsymbol{x}_2^{(t)}\end{bmatrix}\\ &+ \begin{bmatrix}\boldsymbol{M}_1^{-1} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{M}_2^{-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\end{bmatrix}\end{aligned}\tag{3.15}$$

In general, control area J has Gauss-Newton step update given by

$$\Delta \boldsymbol{x}_J^{t+1} = \boldsymbol{M}_J^{-1}\big[\sum_{L=1}^{N} \boldsymbol{N}_{JL}\Delta \boldsymbol{x}_L^t + \boldsymbol{b}_J\big]. \tag{3.16}$$

We already saw that $\boldsymbol{M}_J^{-1}$ can be calculated independently in each control area. We consider what information needs to be communicated in order to calculate the matrices $\{\boldsymbol{N}_{JL}\}$ and $\boldsymbol{b}_J$. Let the set of border buses in control area J be denoted Ω_J. These are the buses with a neighbor in another control area. Due to the locality of the measurement functions (whose functional form is given in (3.20)-(3.25) for reference), the matrix $\boldsymbol{N}_{JL}$ will be zero unless control areas J and L are neighboring (*i.e.*, there is a transmission line connecting a bus in J to a bus in L) and otherwise sparse. Non-zero entries of $\boldsymbol{N}_{JL}$ can be attributed to measurements of power flow along transmission lines connecting areas J and L and to measurements of power injections at border buses. Similarly, such measurements are the only non-zero contributions to $\boldsymbol{b}_J$ from other control areas. The sparseness of the $\boldsymbol{N}_{JL}$ matrices allows for a limited communication of information between neighboring control areas. We emphasize that the information needed from other areas to calculate the matrices $\{\boldsymbol{N}_{JL}\}$ and $\boldsymbol{b}_J$ is communicated only once per Gauss-Newton iteration.

The communication exchange is detailed in Algorithm 3.1. It is important to stress that the centralized Gauss-Newton method consists only of the outer loop of Gauss-Newton iterations in equation (3.3) which produces a sequence of estimates using the exact Gauss-Newton step obtained from directly solving (3.6). We refer to these estimates as the centralized estimates, $\{\boldsymbol{x}_*^{(k)}\}$. In contrast, our distributed method iteratively solves (3.6) using the matrix-splitting scheme (*i.e.* inner loop) in (3.9). Let $\Delta\widetilde{\boldsymbol{x}}^{(t,k)}$ be the approximation to the Gauss-Newton step at inner-loop iteration t and outer-loop iteration k, and let the sequence of iterates produced by the distributed method using T inner-loop iterations be given by $\widetilde{\boldsymbol{x}}^{(k+1)} = \widetilde{\boldsymbol{x}}^{(k)} - \Delta\widetilde{\boldsymbol{x}}^{(T,k)}$.

Algorithm 3.1 Distributed Gauss-Newton Method for State Estimation

```
In parallel, each control area $1 \leq J \leq N$ does:
Initialization (Flat Start): $\boldsymbol{x}_J^{(0)} = \{\boldsymbol{\theta}_J^{(0)}, \boldsymbol{V}_J^{(0)}\}$ with $\boldsymbol{\theta}_J^{(0)} = \mathbf{0}$ and $\boldsymbol{V}_J^{(0)} = \mathbf{1}$.
for $k := 0$ to $K$ do
    $\Delta\boldsymbol{x}_J^{(0,k)} = \mathbf{0}$
    for $t := 0$ to $T$ do
        Calculate $\Delta\boldsymbol{x}_J^{(t+1,k)}$ from (3.16).
        for $a \in \Omega_J$ do
            Send $\Delta\boldsymbol{x}_a^{(t+1,k)}$ to neighboring nodes in other control areas.
            if $\widehat{P}_a$ or $\widehat{Q}_a \in \mathcal{Z}$ and finished receiving neighbors updates then
                Send $\{\Delta\boldsymbol{x}_b^{(t+1,k)}\}$ to neighboring nodes $b \in \mathcal{N}_a$ in other control areas.
            end if
        end for
    end for
    $\boldsymbol{x}_J^{(k+1)} = \boldsymbol{x}_J^{k} - \Delta\boldsymbol{x}_J^{(k,T)}$
    for $a \in \Omega_J$ do
        Send $\boldsymbol{x}_a^{(k+1)}$ to neighbors $b \in \mathcal{N}_a$ in other control areas.
        if $\widehat{P}_a$ or $\widehat{Q}_a \in \mathcal{Z}$ and finished receiving neighbors estimates then Send $\{\boldsymbol{x}_b^{(k+1)}\}$ to
    neighboring nodes $b \in \mathcal{N}_a$.
```

```
        end if
    end for
end for
```

We summarize the communication requirements of our algorithm. For clarity of exposition, consider each bus to be a separate control area. Communication between buses in the same control area is considered negligible compared to communication between buses in neighboring control areas. If there are no power injection measurements present at node a, then node a needs to communicate to its neighbors only its current state estimate $\widetilde{\boldsymbol{x}}_a^{(k)}$ at each outer-loop iteration and its Gauss-Newton step estimate $\Delta\widetilde{\boldsymbol{x}}_a^{(t,k)}$ at each inner-loop iteration. We illustrate the additional information exchange for nodes with a power injection measurement using the simple example in Figure 3.3.

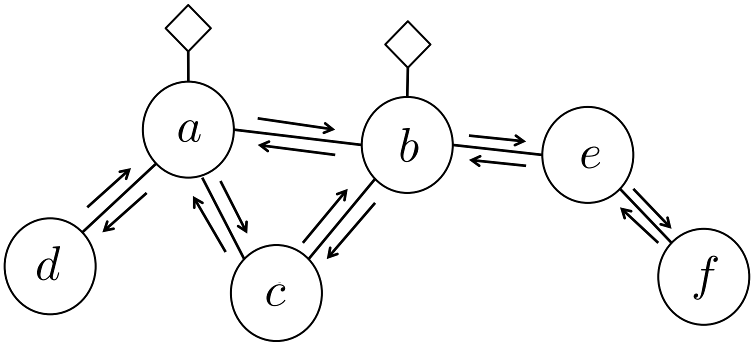


Figure 3.3: Example to illustrate information exchange of Algorithm 3.1. There are power injection measurements at nodes a and b, signified by a diamond-shaped sensor. Communication occurs directly between neighbors.

Let power injections be measured at nodes a and b. At initialization of the Gauss-Newton method,

- Node a sends the value of its real and reactive power injection measurements, $\widehat{P}_a$ and $\widehat{Q}_a$, to its neighbors b, c, and d.

- Node a receives $\widehat{P}_b$ and $\widehat{Q}_b$ from node b.

At outer-loop iteration k,

- Node a sends $\{\widetilde{\boldsymbol{x}}_a^{(k)}, \widetilde{\boldsymbol{x}}_b^{(k)}, \widetilde{\boldsymbol{x}}_c^{(k)}, \widetilde{\boldsymbol{x}}_d^{(k)}\}$ to its neighbors b, c, and d.

- Node a receives $\{\widetilde{\boldsymbol{x}}_b^{(k)}, \widetilde{\boldsymbol{x}}_c^{(k)}, \widetilde{\boldsymbol{x}}_e^{(k)}\}$ from node b. Node a also receives $\widetilde{\boldsymbol{x}}_c^{(k)}$ from node c and $\widetilde{\boldsymbol{x}}_d^{(k)}$ from node d.

At inner-loop iteration t,

- Node a sends $\{\Delta\widetilde{\boldsymbol{x}}_a^{(t,k)}, \Delta\widetilde{\boldsymbol{x}}_b^{(t,k)}, \Delta\widetilde{\boldsymbol{x}}_c^{(t,k)}, \Delta\widetilde{\boldsymbol{x}}_d^{(t,k)}\}$ to its neighbors b, c, and d.

- Node a receives $\{\Delta\widetilde{\boldsymbol{x}}_b^{(t,k)}, \Delta\widetilde{\boldsymbol{x}}_c^{(t,k)}, \Delta\widetilde{\boldsymbol{x}}_e^{(t,k)}\}$ from node b. Node a also receives $\Delta\widetilde{\boldsymbol{x}}_c^{(t,k)}$ from node c and $\Delta\widetilde{\boldsymbol{x}}_d^{(t,k)}$ from node d.

The distributed algorithm uses a finite number of matrix-splitting iterations, T, to calculate the Gauss-Newton step. Due to truncation error, this will not be exactly equal to the Gauss-Newton step, $\Delta\boldsymbol{x}^{(k)}$, given by directly solving (3.6). Therefore, we must distinguish between the centralized estimates, $\{\boldsymbol{x}_*^{(k)}\}$, and the distributed estimates, $\{\widetilde{\boldsymbol{x}}^{(k)}\}$. Let $\boldsymbol{y}_*^{(k)}$ be the solution of the linear system

$$\boldsymbol{J}(\widetilde{\boldsymbol{x}}^{(k)})^T\boldsymbol{W}\boldsymbol{J}(\widetilde{\boldsymbol{x}}^{(k)})\boldsymbol{y}_*^{(k)} = -\boldsymbol{J}(\widetilde{\boldsymbol{x}}^{(k)})^T\boldsymbol{W}(\boldsymbol{z} - \boldsymbol{h}(\widetilde{\boldsymbol{x}}^{(k)})) \tag{3.17}$$

The iterative scheme in equation (3.9) forms a discrete linear dynamic system and therefore exponentially converges to the solution $\boldsymbol{y}_*^{(k)}$ of the linear system in equation (3.17) according to [30, Theorem 6.1], provided $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) < 1$. The convergence speed is determined by $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$.

3.2.4 Proof of Matrix-Splitting Convergence

First, we note the following statements. The sequence $\{\Delta \boldsymbol{x}^t\}$ in (3.9) converges to its limit $\Delta \boldsymbol{x}^*$ as $t \to \infty$ if and only if the spectral radius of the matrix $\boldsymbol{M}^{-1}\boldsymbol{N}$ is strictly less than 1 [102]. Furthermore, if the sequence converges, the limit $\Delta \boldsymbol{x}^*$ is the solution of the system, (i.e., $\boldsymbol{A}\Delta\boldsymbol{x}^* = \boldsymbol{b}$). In order to have the spectral radius $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) < 1$, it is sufficient to have $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N} \succ \mathbf{0}$ and $\boldsymbol{M} + \boldsymbol{N} \succ \mathbf{0}$ [26]. We have the following proposition that ensures convergence of the matrix-splitting iterates in (3.9).

Proposition 3.1. *Given a positive definite matrix $\boldsymbol{A}$, let $\boldsymbol{M} = \boldsymbol{D} + \alpha\bar{\boldsymbol{E}}$ and $\boldsymbol{N} = \alpha\bar{\boldsymbol{E}} - \boldsymbol{E}$ where $\boldsymbol{D}$, $\boldsymbol{E}$, and $\bar{\boldsymbol{E}}$ are given in equations (5.21)-(5.24). Then, for $\alpha \geq \frac{1}{2}$, $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) < 1$.*

Proof. By Theorem 2.5.3 of [26], to prove that $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) < 1$, it is sufficient to show that $\boldsymbol{M} - \boldsymbol{N}$ and $\boldsymbol{M} + \boldsymbol{N}$ are both positive definite. Using the Gauss-Newton method, the formula for $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ in (3.7) is positive definite by construction. We show that $\mathbf{H^T W^{-1} H} = (\mathrm{M} + \mathrm{N})$ is positive definite. Under Assumption 1, the weighting matrix $\mathbf{W}$ is real, symmetric, and positive definite. By the spectral theorem, there exists an orthogonal matrix $\boldsymbol{\mathcal{O}}$ and diagonal matrix $\boldsymbol{\mathcal{D}}$ such that $\mathbf{W^{-1}} = \boldsymbol{\mathcal{O}}^{\mathbf{T}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{O}}$. Let $\mathrm{Q} = \boldsymbol{\mathcal{O}}\mathbf{H}$. Then, we see that for any $\mathbf{x} > \mathbf{0}$,

$$\mathbf{x^T H^T W^{-1} H x} = ||\boldsymbol{Q}\boldsymbol{x}||_{\boldsymbol{\mathcal{D}}}^{\mathbf{2}} > \mathbf{0}, \tag{3.18}$$

where strict positivity comes from the fact that $\mathbf{R}$ is positive definite and thus all diagonal entries of $\boldsymbol{\mathcal{D}}$ are strictly positive. Thus, it is sufficient to show that $\boldsymbol{M} + \boldsymbol{N} = \boldsymbol{D} + 2\alpha\bar{\boldsymbol{E}} - \boldsymbol{E}$ is positive definite. As a corollary to the Gershgorin Circle Theorem [101], we only need to show that $\boldsymbol{D} + 2\alpha\bar{\boldsymbol{E}} - \boldsymbol{E}$ is strictly diagonally dominant with strictly positive diagonal entries. [2] Given that $\alpha \geq \frac{1}{2}$,

$$\begin{aligned}
[\boldsymbol{D} + 2\alpha\bar{\boldsymbol{E}} - \boldsymbol{E}]_{ii} &= D_{ii} + 2\alpha\bar{E}_{ii} \\
&> 2\alpha\sum_{j\neq i}|A_{ij}| && \text{(3.19a)} \\
&\geq \sum_{j\neq i}|[\boldsymbol{D} + 2\alpha\bar{\boldsymbol{E}} - \boldsymbol{E}]_{ij}|, && \text{(3.19b)}
\end{aligned}$$

where (3.19a) follows from the definition of $\bar{\boldsymbol{E}}$ and from the fact that $D_{ii} > 0$. Since $\boldsymbol{A}$ is positive definite, its diagonal entries must be strictly positive. The inequality in (3.19b) follows from (3.19a) since by construction the support of $\boldsymbol{D}$ and $\boldsymbol{E}$ are complimentary, meaning that for all i, j if D_{ij} is nonzero then $E_{ij} = 0$ and vice versa. Furthermore, the contribution from $2\alpha\bar{\boldsymbol{E}}$ in (3.19b) is zero since the off-diagonal terms of $\bar{\boldsymbol{E}}$ are zero. □

3.2.5 Proof of Communication Requirements

The local nature of the measurement equations is key for allowing our algorithm to have limited communication requirements. We introduce the following notations to denote different kinds of measurement equations and emphasize which states that they depend upon. Let the neighbor set of node a be denoted $\mathcal{N}_a$. The exact mathematical form of the non-linear measurement model

[2] A matrix $\boldsymbol{A}$ is strictly diagonal dominant if $|A_{ii}| > \sum_{j\neq i}|A_{ij}|$ for all i.

equations can be found in in [2].

1. If $z_k = \widehat{P}_{ab}$, then

$$h_k(\boldsymbol{x}) = h_{P(a,b)}(\theta_a, V_a, \theta_b, V_b). \tag{3.20}$$

2. If $z_k = \widehat{Q}_{ab}$, then

$$h_k(\boldsymbol{x}) = h_{Q(a,b)}(\theta_a, V_a, \theta_b, V_b). \tag{3.21}$$

3. If $z_k = \widehat{P}_a$, then

$$h_k(\boldsymbol{x}) = h_{a,P}(\theta_a, V_a, \{\theta_b\}_{b\in\mathcal{N}_a}, \{V_b\}_{b\in\mathcal{N}_a}). \tag{3.22}$$

4. If $z_k = \widehat{Q}_a$, then

$$h_k(\boldsymbol{x}) = h_{a,Q}(\theta_a, V_a, \{\theta_b\}_{b\in\mathcal{N}_a}, \{V_b\}_{b\in\mathcal{N}_a}). \tag{3.23}$$

5. If $z_k = \widehat{\theta}_a$, then

$$h_k(\boldsymbol{x}) = h_{\theta,a}(\theta_a). \tag{3.24}$$

6. If $z_k = \widehat{V}_a$, then

$$h_k(\boldsymbol{x}) = h_{V,a}(V_a). \tag{3.25}$$

Applying the iterative scheme of (3.9) to the state estimation problem with our choice of matrix splitting, we obtain

$$\Delta\boldsymbol{x}^{(t+1)} = (\boldsymbol{D} + \boldsymbol{\alpha}\bar{\boldsymbol{E}})^{-1}\left[(\boldsymbol{\alpha}\bar{\boldsymbol{E}} - \boldsymbol{E})\Delta\boldsymbol{x}^{(t)} + \nabla f(\boldsymbol{x})\right]. \tag{3.26}$$

We denote the term within the bracket as:

$$\boldsymbol{y}^{(t)} \overset{\text{def}}{=} (\boldsymbol{\alpha}\bar{\boldsymbol{E}} - \boldsymbol{E})\Delta\boldsymbol{x}^{(t)} + \nabla f(\boldsymbol{x}). \tag{3.27}$$

In terms of the N control areas, equation (3.26) can be written in block-form as

$$\begin{bmatrix} \Delta\boldsymbol{x}_1^{(t+1)} \\ \vdots \\ \Delta\boldsymbol{x}_N^{(t+1)} \end{bmatrix} = \begin{bmatrix} \widehat{\boldsymbol{M}}_1 & & \\ & \ddots & \\ & & \widehat{\boldsymbol{M}}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_1^{(t)} \\ \vdots \\ \boldsymbol{y}_N^{(t)} \end{bmatrix}, \tag{3.28}$$

where $\widehat{\boldsymbol{M}}_1, \ldots, \widehat{\boldsymbol{M}}_N$ are the diagonal blocks of $(\boldsymbol{D}+\alpha\bar{\boldsymbol{E}})^{-1}$ corresponding to the different control areas. In the case that each bus is treated as a control area, the matrix inversion is reduced to the inversion of a scalar number.

We utilize the particular structure of the power grid state estimation problem in order to verify the information from neighboring areas needed to calculate the entries of $\Delta\boldsymbol{x}^{(t)}$. We introduce the following quantities:

$$\sigma_{P(a,b)} = \begin{cases} w_k & z_k = \widehat{P}_{ab} \\ 0 & \widehat{P}_{ab} \notin \mathcal{Z} \end{cases}, \; \sigma_{Q(a,b)} = \begin{cases} w_k & z_k = \widehat{Q}_{ab} \\ 0 & \widehat{Q}_{ab} \notin \mathcal{Z} \end{cases},$$

$$\sigma_{a,P} = \begin{cases} w_k & z_k = \widehat{P}_a \\ 0 & \widehat{P}_a \notin \mathcal{Z} \end{cases}, \; \sigma_{a,Q} = \begin{cases} w_k & z_k = \widehat{Q}_a \\ 0 & \widehat{Q}_a \notin \mathcal{Z} \end{cases},$$

$$\sigma_{a,\theta} = \begin{cases} w_k & z_k = \widehat{\theta}_a \\ 0 & \widehat{\theta}_a \notin \mathcal{Z} \end{cases}, \; \sigma_{a,V} = \begin{cases} w_k & z_k = \widehat{V}_a \\ 0 & \widehat{V}_a \notin \mathcal{Z} \end{cases},$$

where w_k are the diagonal entries of the weighting matrix $\boldsymbol{W}$ (see Assumption 3.1). In addition, we define the set of "1-hop" neighbor nodes as nodes that are not direct neighbors but share a common neighbor, $\mathcal{N}_a^\dagger \overset{\text{def}}{=} \{b \mid b \notin \mathcal{N}_a,\ \mathcal{N}_a \cap \mathcal{N}_b \neq \emptyset\}$. Let $\mathcal{N}_a^* \overset{\text{def}}{=} \mathcal{N}_a \cup \mathcal{N}_a^\dagger$ be the union of the sets of direct and "1-hop" neighbors.

Lemma 3.1. *Let $a := n(i)$ be the node index corresponding to entry i of $\boldsymbol{y}^{(t)}$, and let the set $\boldsymbol{x}_a = (\theta_a, V_a)$ refer to the set of estimates of both the voltage angle and magnitude at bus a at the current Gauss-Newton iteration. The components of the updates to $\boldsymbol{y}^{(t)}$ from equation (3.27) can be reduced to the form given in equation (3.29) for a single bus:*

$$
\begin{aligned}
y_i^{(t)} =& \alpha\Bigg(\sum_{\{j|n(j)\in\mathcal{N}_a^*\}} |A_{ij}|\Bigg)\Delta x_i^{(t)} - \sum_{\{j|n(j)\in\mathcal{N}_a^*\}} E_{ij}\Delta x_j^{(t)} - 2\Bigg\{\sigma_{a,\theta}(\widehat{\theta}_a - \theta_a) + \sigma_{a,V}(\widehat{V}_a - V_a) \\
&+ \sum_{\{j|b\equiv n(j)\in\mathcal{N}_a\}} \Bigg[\sigma_{P(a,b)}\frac{\partial h_{P(a,b)}}{\partial x_i}\big(\widehat{P}_{ab} - h_{P(a,b)}(\boldsymbol{x}_a, \boldsymbol{x}_b)\big) \\
&+ \sigma_{Q(a,b)}\frac{\partial h_{Q(a,b)}}{\partial x_i}\big(\widehat{Q}_{ab} - h_{Q(a,b)}(\boldsymbol{x}_a, \boldsymbol{x}_b)\big) \\
&+ \sigma_{b,P}\frac{\partial h_{b,P}}{\partial x_i}\big(\widehat{P}_b - h_{b,P}(\boldsymbol{x}_b, \{\boldsymbol{x}_c\}_{c\in\mathcal{N}_b})\big) + \sigma_{b,Q}\frac{\partial h_{b,Q}}{\partial x_i}\big(\widehat{Q}_b - h_{b,Q}(\boldsymbol{x}_b, \{\boldsymbol{x}_c\}_{c\in\mathcal{N}_b})\big)\Bigg] \\
&+ \sigma_{a,P}\frac{\partial h_{a,P}}{\partial x_i}\big(\widehat{P}_a - h_{a,P}(\boldsymbol{x}_a, \{\boldsymbol{x}_c\}_{c\in\mathcal{N}_a})\big) + \sigma_{a,Q}\frac{\partial h_{a,Q}}{\partial x_i}\Big(\widehat{Q}_a - h_{a,Q}(\boldsymbol{x}_a, \{\boldsymbol{x}_c\}_c \in \mathcal{N}_a)\Big)\Bigg\}
\end{aligned}
\tag{3.29}
$$

Proof. We determine the sparsity pattern of $\boldsymbol{A}$ as a function of the network connectivity. From the AC measurement model equations, we have that

$$
[\boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{W} \boldsymbol{J}(\boldsymbol{x})]_{ij} = \sum_{k=1}^{m} w_k J_{ki}(\boldsymbol{x}) J_{kj}(\boldsymbol{x}) \tag{3.30}
$$

is nonzero if and only $n(j) \in \mathcal{N}_a^*$. The entries of equation (3.27) simplify to the expression in

(3.29). We note that if power injection measurements are not present, the sparsity is reduced to A_{ij} being nonzero if and only if $n(j) \in \mathcal{N}_a$. □

Furthermore, from the proof of Lemma 3.1, node a in control area J only needs information from nodes in $\mathcal{N}_a^*$ to calculate entries of $\widehat{\boldsymbol{M}}_J$. This specifies the subset of external information needed to calculate the updates to $\Delta\boldsymbol{x}^{(t)}$ in (3.28).

3.2.6 Distributed Implementation Using MPI

The Message Passing Interface (MPI) is a language-independent, standardized protocol that allows one to write programs implementing algorithms that utilize communication of data, or messages, across multiple processors [45]. MPI was originally designed for distributed memory architectures. Objects called communicators are used to specify which processors are allowed to communicate with one another. Messages can be sent collectively to a single processor that acts like a central coordinator, or messages can be sent point-to-point between any two processors. Virtual topologies can be designed to specify which processors can communicate with each other.

We use the MPI *graph topology* functionality to create a communicator that has the same topology as the control area network we consider for the power grid. A separate process is assigned to each control area, and communication is allowed between two control areas if there is a tie-line connecting them. For example, in Figure 3.1, Area 1 can send messages directly to Areas 2 and 3 but not to Area 4. This ensures the algorithm is fully distributed (*i.e.*, that information local to a control area is contained on a single processor and otherwise communicated only to processors corresponding to neighboring control areas).

In point-to-point communications, a message is passed between exactly two MPI processes. The process sending information executes a send operation, and the process receiving information executes a receive operation. There are a variety of such send and receive operations available in MPI. These include blocking or synchronous send and receives, that wait to execute any further instructions until the matching send and receive have been completed. There are also non-blocking send and receive operations, that allow for continued execution on a process before the send and receive have necessarily completed. Non-blocking communication is necessary in certain cases, and care must be taken to ensure that the program is processing the correct data.

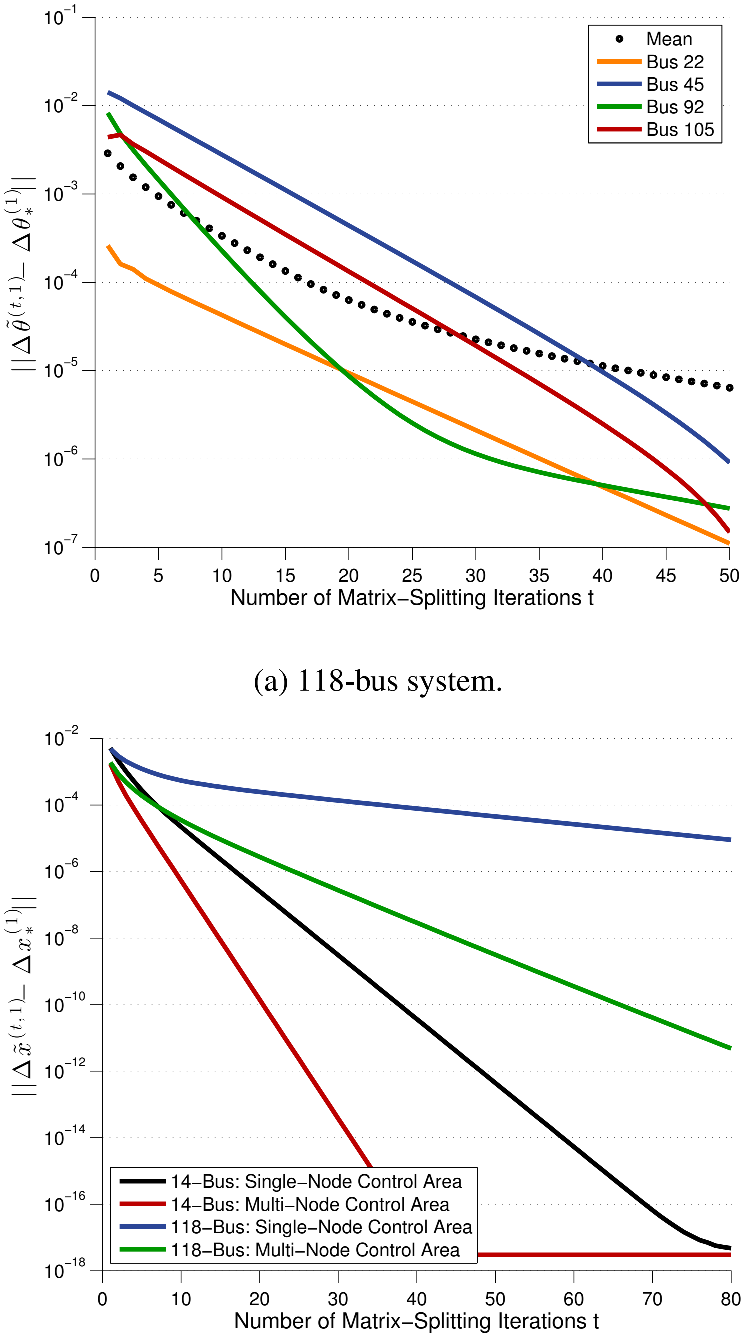
We now detail the use of point-to-point communication in Algorithm 3.1. In Steps 8 and 16, areas need to send their border bus estimates, $\Delta \boldsymbol{x}_a^{(k+1)}$ and border bus Newton step estimates, $\boldsymbol{x}_a^{(k+1)}$, to neighboring control areas. This communication is symmetric since every process sending to another process is also receiving from that process. Therefore, we can use paired send and receive operations ($MPI\_Sendrecv$). For border buses with power injection measurements, extra communication is required in Steps 10 and 17. In this case, the communication is no longer symmetric. In other words, each process sending to another process no longer necessarily receives a message from that process. Since the send operations are no longer perfectly synchronized with the corresponding receive operations, we utilize non-blocking communications (MPI_Isend and MPI_Irecv).

3.3 Numerical Results

We present our numerical results in three sections. First, we study the convergence properties of the matrix-splitting (*i.e.*, inner-loop iterations). Since we are using an iterative method to solve for the Gauss-Newton step, it is important to see how this approximation numerically compares to directly solving for the Gauss-Newton step. Second, we demonstrate the performance of our algorithm in various settings, including a large-scale realistic system with 1,354 buses. Third, we present results towards designing an optimal splitting with respect to the parameter α and to the control area partitioning. Interesting open questions remain in this area.

3.3.1 Convergence of Matrix-Splitting Iterations

To study the convergence of the distributed algorithm with respect to the matrix-splitting (*i.e.*, inner-loop) iterations, we calculate at each matrix-splitting iteration t, the error on the distributed Newton step $||\Delta\widetilde{\boldsymbol{x}}^{(t,k)} - \Delta\boldsymbol{x}_{*}^{(k)}||$. Figure 3.4 demonstrates the exponential convergence of the distributed Gauss-Newton step to the centralized Gauss-Newton step for $k = 1$. The case studies include the IEEE 14-bus and IEEE 118-bus systems. Additionally, we study the convergence under two different communication schemes. In the (“Single-Node Control Area") scheme, each node is considered its own control area. The single-node control area setting is a natural limit for understanding the behavior of the algorithm in terms of the size of the control areas and the degree to which the calculation is distributed. In the (“Multi-Node Control Area") setting, several nodes are grouped into a single control area. From Figure 3.4 (b), the convergence of the (“Multi-Node Control Area") setting is faster than the (“Single-Node Control Area") setting. However,



(a) 118-bus system.

(b) Different communication schemes.

Figure 3.4: These figures show the exponential convergence of the distributed Gauss-Newton step. In (a), the convergence is shown for the Gauss-Newton step update of the phase angles at individual buses. In (b), the convergence of the distributed Gauss-Newton step estimates to the centralized exact Gauss-Newton step is faster in the (“Multi-Node Control Area") setting than in the (“Single-Node Control Area") setting.

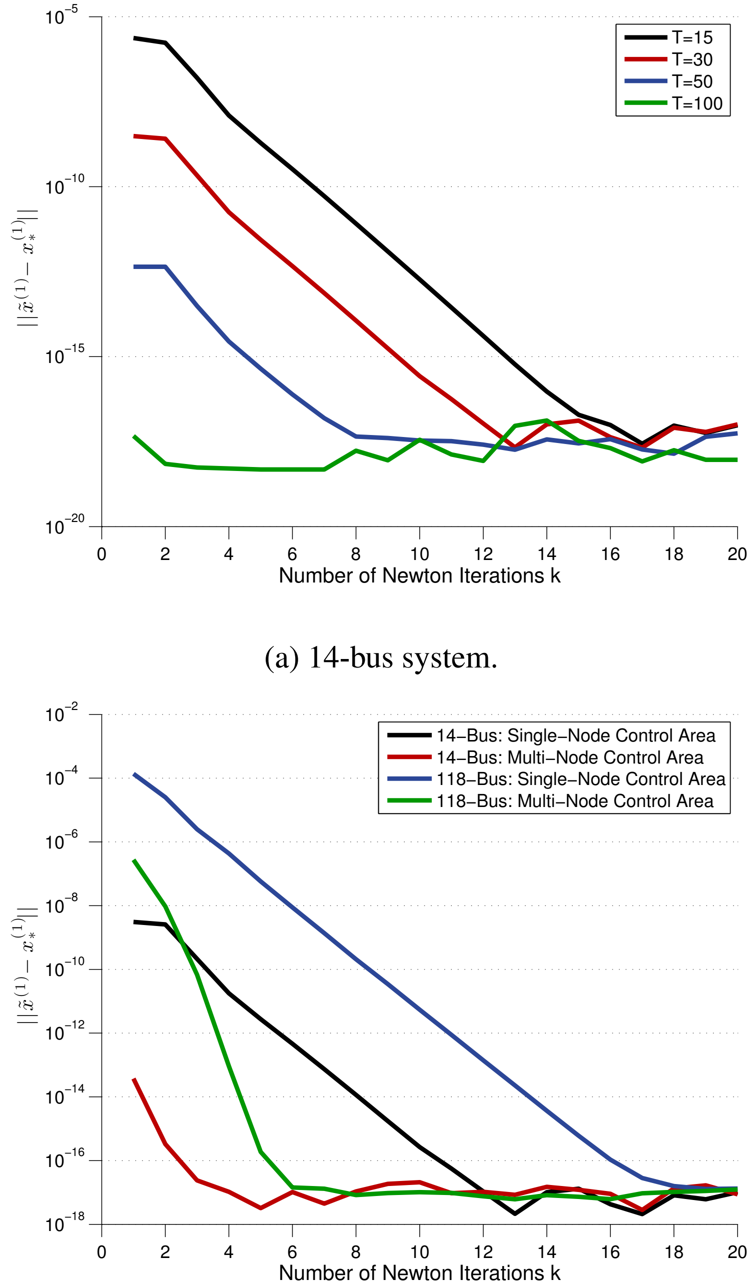
each matrix-splitting iteration is computationally more costly for the (“Multi-Node Control Area") setting, so there is a trade-off between required number of iterations and the computational cost of each iteration.

We use the partitioning of the 14-bus system into the four control areas given in Figure 3.1 [61]. For the 118-bus system, we use the nine control area partitioning from [109]. For the 14-bus system, the measurement configuration used is shown in Figure 3.1. The types and locations of measurements used for the 118-bus system tests are given in Table 2.1. The measurements are perturbed by additive Gaussian noise where the variance of the measurements used was taken from [109]. The inverse covariance matrix $\boldsymbol{R}$ is used as the weighting matrix $\boldsymbol{W}$.

As another measure of the convergence of the matrix-splitting iterations, it is interesting to examine the difference between the centralized and distributed estimates as a function of outer-loop iterations. Figure 3.5 shows the error of the distributed estimates with respect to the centralized estimates. The linear system in equation (3.6) is identical for the distributed and centralized approach only at the first Gauss-Newton (*i.e.*, outer-loop) iteration. This is due to truncation error resulting from using only a finite number, T, of matrix-splitting iterations. Despite propagating an inexact Gauss-Newton step at each Gauss-Newton iteration, the distributed estimates agree with the centralized estimates within several tens of iterations.

3.3.2 Performance

In Figure 3.6, we show the convergence of $f(\boldsymbol{x}^{(k)})$ as a function of the total number of iterations (number of inner-loop iterations times number of outer-loop iterations) for the distributed Gauss-



(a) 14-bus system.

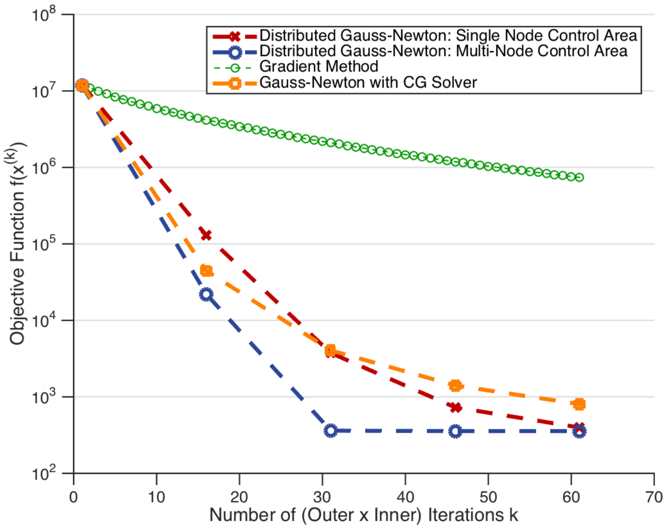
(b) Using $T = 30$.

Figure 3.5: These figures show the difference between the distributed and centralized estimates as a function of the number of Gauss-Newton iterations. In (a), we study this for different values of T, the number of matrix-splitting iterations, for the 14-bus system. We see that for $T = 100$, the distributed and exact Gauss-Newton step agree up to machine precision for all iterations. In (b), we use $T = 30$ iterations and and compare for different network sizes and communication schemes.

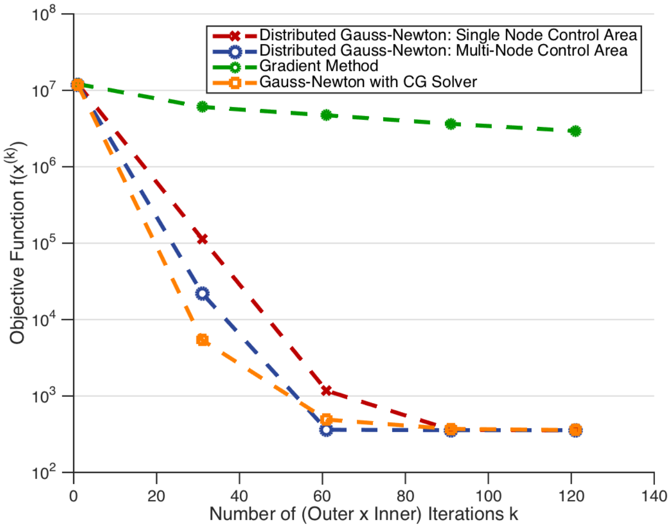
Newton method and gradient method. In Figure 3.6 (a), we use the gradient method with a constant step size, which can be implemented in a fully distributed way. The gradient method with constant step size requires no inner iterations. For example, if $T = 15$ matrix-splitting iterations are used, we compare the value of the objective function after X distributed Gauss-Newton iterations to the value of the objective function after $15 * X$ gradient iterations. In Figure 3.6 (b), we compare the gradient method using a backtracking line search to determine the optimal step-size. We note that the backtracking line search requires central coordination. These plots show the advantage of Newton-type methods over gradient descent with respect to rate of convergence.

In addition, we compare our distributed Gauss-Newton method using the matrix-splitting based solver to the Gauss-Newton method using the conjugate gradient algorithm to solve for the Gauss-Newton step. The conjugate gradient algorithm is a direct method for solving linear systems requiring n steps for solving an $n \times n$ system. However, it is also viewed as an iterative method since in practice satisfactory accuracy can often be achieved in much fewer than n iterations. Furthermore, for sparse systems, the matrix-vector multiplications that are the main computational burden of the conjugate gradient algorithm can be reduced from $O(n^3)$ operations to $O(m)$ operations, where m is the number of nonzero entries. The convergence of the Gauss-Newton method using a conjugate gradient solver for the Newton step is shown in Figure 3.6 (“Gauss-Newton with CG Solver"). We see that our proposed distributed Gauss-Newton solver achieves similar convergence as the CG-based solver. In addition, since conjugate gradient involves dot products of global quantities, a fully distributed implementation requires more significant communication overhead.

Next, we test the performance of our algorithm on a larger, more realistic system. The algo-



(a) Gradient method with constant stepsize.



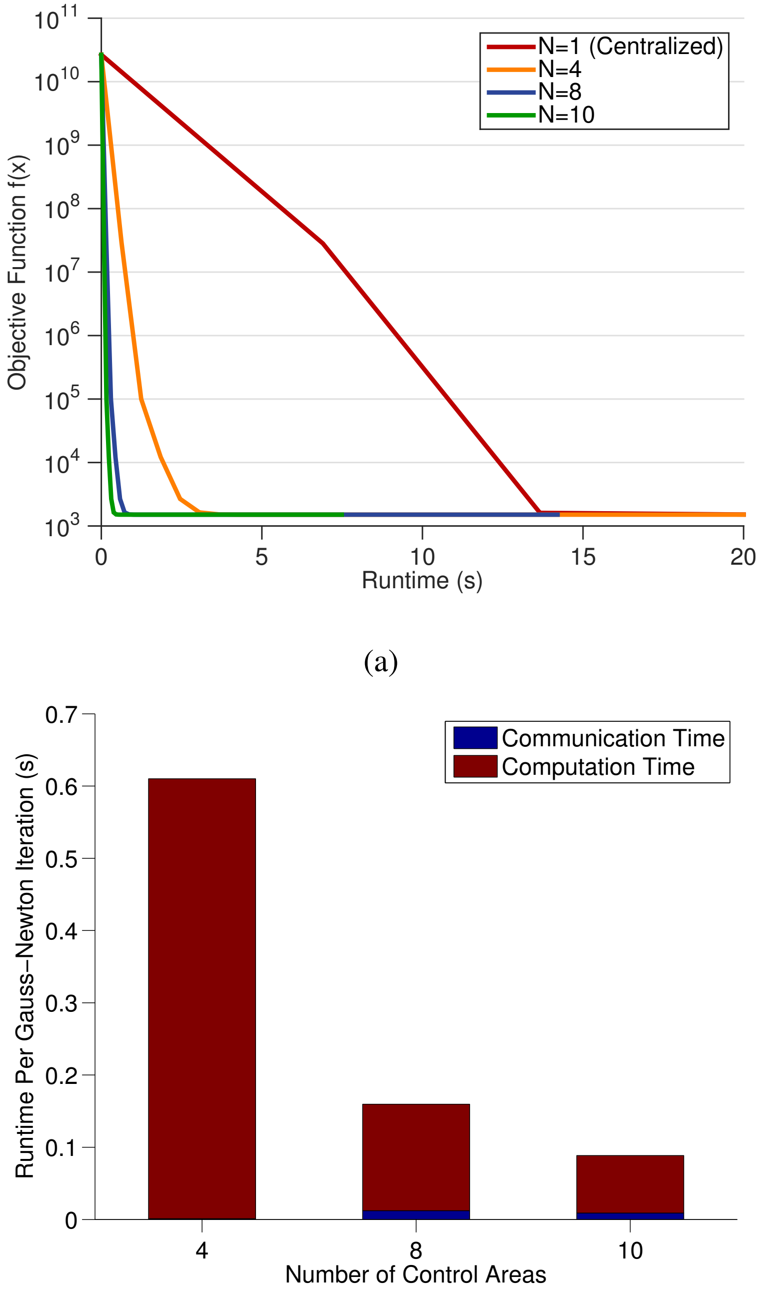
(b) Gradient method with backtracking step-size.

Figure 3.6: The convergence of the objective function for the 118-bus system is shown with respect to the total number of iterations. In a), the gradient method uses a constant stepsize (*i.e.*, no inner iterations), and the distributed Gauss-Newton method and Gauss-Newton method with conjugate gradient (CG) solver use $T = 15$ inner iterations per outer iteration. In b), the gradient method uses a backtracking line search. This requires 30 inner iterations. To compare, we use $T = 30$ matrix-splitting iterations and 30 conjugate gradient iterations in b). Using a backtracking line search, the gradient method is no longer a distributed method.

rithm is implemented in C++ using MPI [45] to run over multiple processors. In Figure 3.7 (a), we show the convergence of the objective function in terms of runtime in seconds for different control area partitionings of the PEGASE 1,354-bus test system [39, 112]. The centralized scheme (using a single control area) is notably slower. Increasing the number of control areas and hence degree of distributivity decreases the runtime.

The breakdown of CPU time spent on computation and communication for a single distributed Gauss-Newton iteration is provided in Figure 3.7 (b). As the number of control areas grows, communication time remains constant as desired, while computation time decreases since the problem solved by each processor is smaller. We expect this behavior since the neighborhood of a control area remains roughly constant as the number of control areas increases. The relative time spent on communication versus computation in Figure 3.7 depends on the network characteristics. For our simulations we used the Blue Waters computing facility [9, 63], which has a peak node injection bandwidth of 9.6 GB/s. If there is no power injection, at each inner iteration a node sends and receives $\Delta\theta$ and ΔV (two double precision numbers or 128 bits) to each of its neighbors, and at each outer iteration each node exchanges θ and V, again 128 bits, with its neighbors. The number of neighbors of a node is roughly 10. If there is a power injection, the bus sends its neighbors' information to its neighbors. Therefore, it takes roughly 1280 bits * 1 s / (9.6 * 10^9 bits) = 1.33 * 10^{-7} s to communicate without power injections and 12,800 bits * 1s / (9.6 * 10^{-9} bits) = 1.33 * 10^{-6} s to communicate with power injections.

For the measurement configuration, we randomly place power injection measurements and PMUs at 10% of the buses and 33% of the buses, respectively. Power flow measurements are



(a)

(b)

Figure 3.7: For a system with 1,354 buses, in a) the convergence of the objective function $f(\boldsymbol{x})$ in terms of runtime in seconds is shown for different control area partitionings. With more control areas (higher degree of distributivity), the runtime needed to converge is smaller. In b), the breakdown of time spent in a single outer-loop iteration for computation versus communication is shown.

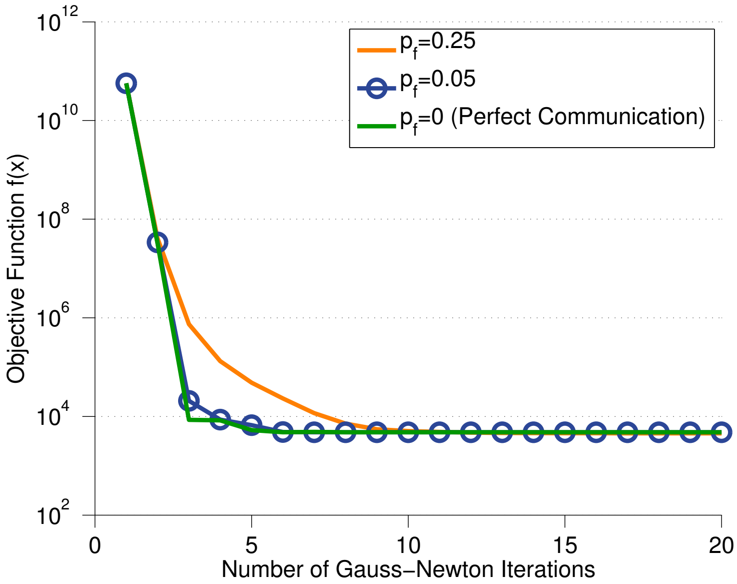


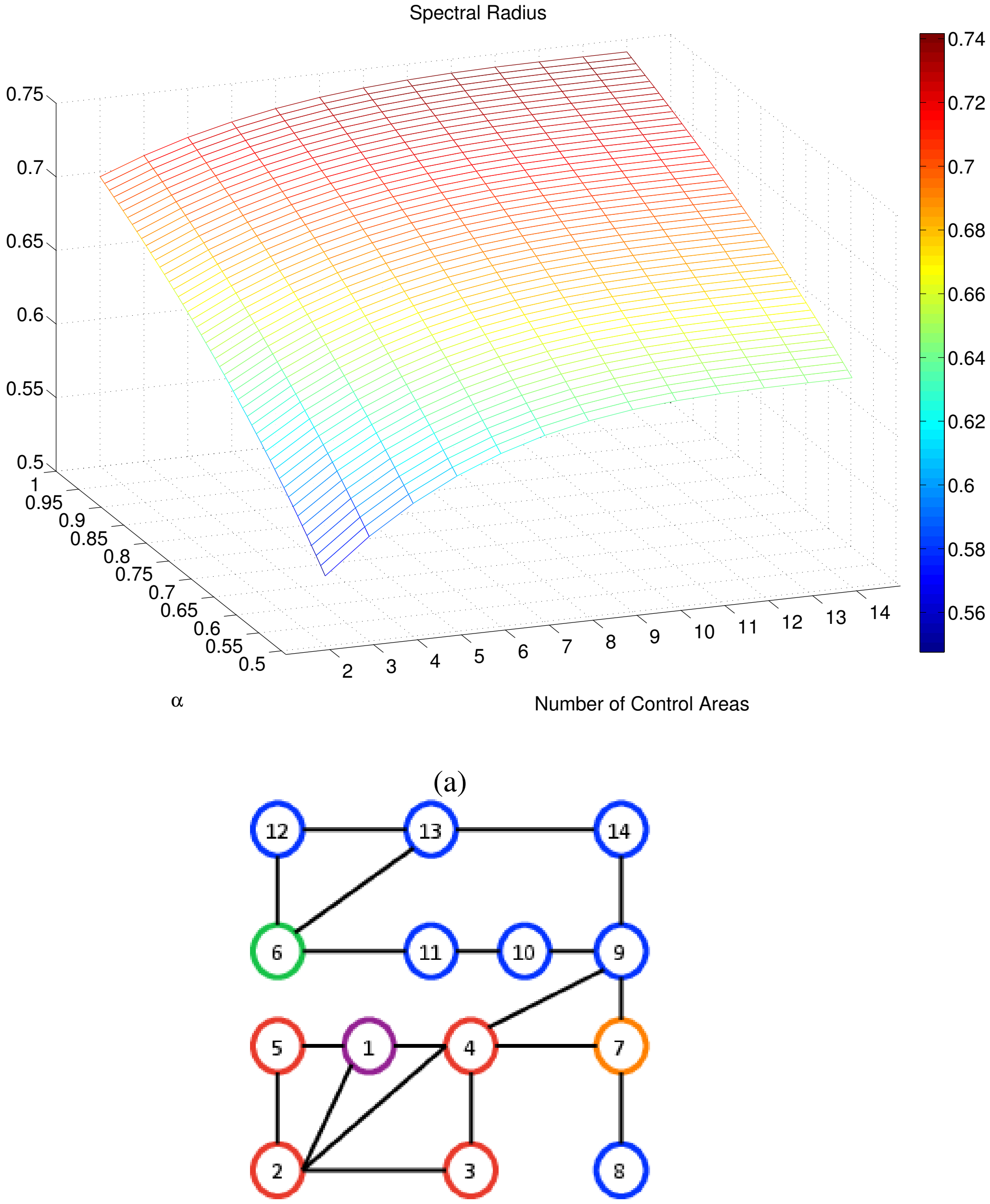
Figure 3.8: Test performance of the 1,354-bus system in presence of communication failure between control areas.

taken at 90% of the transmission lines. To partition the 1,354-bus network into control areas as a pre-processing step, we use a routine written by J. Hespanha [50], that clusters based on spectral factorization.

Additionally, we study the robustness of our algorithm to communication failures between control areas in Figure 3.8. The probability of failure p_f is the probability that a communication failure between two control areas will occur. We see that our algorithm is fairly robust under such communication dropouts and demonstrates a decrease in the rate of convergence as the probability of communication failure increases.

3.3.3 Towards an Optimal Matrix Splitting

Recall from Section 3.2.3 that the convergence of the matrix-splitting iterations is determined by the spectral radius $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$. The value of $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ depends on the way the network is partitioned into control areas and the value of the splitting parameter α from equations (3.13)-(3.14). Optimizing the convergence rate with respect to the partitioning is an interesting open question. Here we present some initial numerical results. Figure 3.9 (a) shows the dependence of the spectral radius $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ on the parameter α and the number of control areas used to partition the network. There are a combinatorial number of ways to partition a network of n nodes into N control areas. For each value of N, the values shown in Figure 3.9 are averaged over all possible integer compositions of n into N parts. For $N = 5$, the control area partitioning achieving the minimal spectral radius is shown in Figure 3.9 (b). It is interesting to note that the optimal communication network is not necessarily the same as the underlying power grid structure as evidenced by the disconnection of node 8 from the rest of its control area. We see that, for each possible partitioning, the spectral radius is minimal at $\alpha = 1/2$. As stated in Section 3.2.3, for $\alpha \geq \frac{1}{2}$, we have $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) < 1$. We note that this is a sufficient and not necessary condition on α, so one might want to try to tune α accordingly.



(a)

(b)

Figure 3.9: (a) Dependence of $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ on α and control area partitioning for the 14-bus test case. Using $N = 14$ control areas, the algorithm is fully-distributed. Using a single control area ($N = 1$), the algorithm is centralized. This plot exhibits the trade-off between the degree to which the calculation is distributed and the rate of convergence. In addition, the minimal values of $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ are achieved at $\alpha = 1/2$. (b) Configuration achieving minimal value of $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ with $N = 5$ control areas is shown.

3.4 Conclusion

In this chapter, we use matrix-splitting techniques to develop a new fully distributed Gauss-Newton algorithm for power system state estimation. We consider the non-linear AC measurement model and a multi-area setting. The method incorporates both traditional SCADA measurements, as well as PMU measurements. The algorithm requires limited sharing of information between neighboring control areas and allows for each control area to store and compute only its local estimate. Numerical experiments verify the convergence properties of this algorithm and the advantage of Newton-type methods for state estimation. Future research directions include investigating how to incorporate distributed bad data detection and network observability into our algorithm.

Chapter 4

Distributed Load-Based Dynamic State Estimation

IN THIS CHAPTER, we propose a fully-distributed algorithm for load-based dynamic state estimation in power systems. Unscented Kalman filter techniques are used to accommodate a nonlinear measurement model. The method requires only local information and a limited amount of information from neighboring areas rather than global, system-wide information. We propose to use the local dependence of the power system measurement equations in order to achieve an approximate decoupling of the load-based dynamic state estimation problem. Our method has reduced computation and communication requirements compared to existing methods.

Accurate state estimation is an integral part of power system operations. In static state estimation (SSE), the system state is inferred using only measurements from the current snapshot in time. The motivation behind dynamic state estimation is to use information from prior measurements in addition to the most recent measurements to make an improved estimate. We consider the state to be the voltage phasors at every bus, which change in time due to changes in load. We refer to this as load-based dynamic state estimation (LB-DSE). The measurement equations relating power flows and power injections to voltages are nonlinear under the AC power flow model. Recently, it has been proposed to use the unscented Kalman filter (UKF) for nonlinear state estimation in power systems [99]. The UKF is an extension of the Kalman filter for nonlinear systems that is based on a statistical linearization proposed in [58]. In contrast to the extended Kalman filter (EKF), the UKF is shown in [99, 105] to be favorable for power system state estimation for its low computational complexity and improved accuracy.

As motivated in Chapter 3, given the large scale of power systems and the advent of wide-area monitoring, distributed approaches to state estimation are increasingly important [95]. Distributed algorithms allow for portions of the grid, referred to as control areas, to locally estimate their state based on limited communication with neighboring control areas. For large enough systems, it can be computationally prohibitive to solve the full centralized state estimation problem within real-time constraints. Decentralized approaches alleviate the computational burden by decomposing the problem into smaller problems per area, as well as improving the system's robustness in the event of communication failures.

The UKF algorithm uses a set of points, consisting of different possible values for the estimates, that are propagated through the nonlinear dynamics. The algorithm relies on the sample covariance of these points, which requires globally assessing all possible correlations between pairs of variables. This makes developing distributed UKF algorithms a difficult problem. We propose a new fully-distributed UKF algorithm for LB-DSE. The main contributions of this work include:

1. The communication and computation requirements of our algorithm are reduced compared to existing works for LB-DSE (*e.g.* [90, 92]). Control areas need only to communicate their border buses' estimates with neighboring control areas once per UKF iteration.

2. Each area only holds an estimate of its local state, reducing memory requirements.

3. Numerical experiments demonstrate competitive performance with respect to the centralized UKF, as well as clear improvement over a distributed UKF with no communication allowed between neighboring areas.

Fully-distributed UKF methods for tracking applications are presented in [104] using the information filter and in [98] using consensus filters. Previous work on decentralized UKFs specifically for power systems includes [90,92,96]. In [92], the authors advocate using the information form of the Kalman filter as in [104]. The method in [104] relies on a centralized processor in the form of an aggregation filter, which processes results from all local areas in order to obtain a single global estimate. The algorithm in [96] proposes a decentralized algorithm for estimation of the generator rotor angle and frequency, rather than the voltage phasors. This algorithm distributes the UKF by considering certain measurements as pseudo-inputs. The authors in [90] apply a consensus-based algorithm to develop a fully-distributed UKF for power system state estimation. At each time step, the method in [90] requires each control area to complete two runs of a UKF with an iterative consensus-based communication stage in between. There are two key differences between our proposed method and that of [90]. First, in our method, communication is done in one-shot at each time step between neighboring areas, making the method highly scalable, and each area locally runs its UKF only *once* per time step. Second, the communication requirements of our method are not based on consensus. The advantage of our method lies in its reduced computation and communication requirements.

4.1 Problem Statement

We consider a multi-area interconnected power network, denoted by an undirected graph $(\mathcal{N}, \mathcal{E})$ with a set $\mathcal{N} \overset{\text{def}}{=} \{1, 2, \dots, n\}$ of buses and a set $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ of transmission lines connecting the buses. We aim to estimate the voltage phase angle and magnitude at every bus, $[\boldsymbol{x}^k]^T = [\theta_1^k \dots \theta_n^k \; V_1^k \dots V_n^k]$ at each time step k. The dynamics of the system are driven by changes in the load. We consider the following measurements: the real and reactive power flows between buses i and j, P_{ij} and Q_{ij}; the real and reactive power injections at bus i, P_i and Q_i; and the voltage phase angle and magnitude at bus i, θ_i and V_i. The power flow and power injection measurements are obtained using the SCADA system, and the voltage phasors are obtained using phasor measurement units (PMUs). The AC power flow model nonlinearly relates the measurements to the underlying state,

$$\boldsymbol{z}^k = \boldsymbol{h}(\boldsymbol{x}^k) + \boldsymbol{e}^k, \tag{4.1}$$

where $\boldsymbol{e}^k$ is a zero-mean Gaussian noise vector with a diagonal covariance matrix $\boldsymbol{R}$ and $\boldsymbol{z}^k \in \mathbb{R}^m$. The goal of dynamic state estimation is to infer $\boldsymbol{x}^k$ using past measurements up to and including time step k.

4.1.1 Preliminaries on System Dynamic Modeling and the Unscented Kalman Filter (UKF)

As proposed in [99], the state vector at time k, $\boldsymbol{x}^k$, is updated according to a linear discrete-time dynamic model

$$\boldsymbol{x}^{k+1} = \boldsymbol{F}_k \boldsymbol{x}^k + \boldsymbol{g}_k + \boldsymbol{q}_k, \tag{4.2}$$

where $\boldsymbol{q}_k$ is the process noise vector and $\boldsymbol{F}_k$ and $\boldsymbol{g}_k$ are constructed in an online fashion using Holt's exponential smoothing technique, which has been studied for state-forecasting in power systems in [67]. The process noise vector $\boldsymbol{q}_k$ is assumed to be zero-mean Gaussian with covariance matrix $\boldsymbol{Q}^k$. Further details on construction of $\boldsymbol{F}_k$ and $\boldsymbol{g}_k$ can be found in [99]. The following assumption used in [99] is also made here:

Assumption 4.1. *The matrices $\boldsymbol{F}_k$ and $\boldsymbol{Q}_k$ are diagonal.*

A qualitative overview of the UKF algorithm is given in Figure 4.1. Like the traditional Kalman filter, the unscented Kalman filter consists of a prediction stage based on the system dynamics and a correction stage based on the measurements. However, unique to the UKF, these stages are applied to a set of sigma points, representing different possible values for the estimated state. The set of sigma points are generated from the current estimate, $\boldsymbol{x}^k$, as follows

$$\begin{aligned} &[\boldsymbol{X}^k]_1 = \boldsymbol{x}^k,\ [\boldsymbol{X}^k]_i = \boldsymbol{x}^k + c[\sqrt{\boldsymbol{P}^k}]_{i-1},\ (i{=}2,\ldots,2n{+}1) \\ &[\boldsymbol{X}^k]_i = \boldsymbol{x}^k - c[\sqrt{\boldsymbol{P}^k}]_{i-2n-1},\ (i{=}2n{+}2,\ldots,4n{+}1) \end{aligned} \tag{4.3}$$

where $\sqrt{\boldsymbol{P}^k}$ is the Cholesky decomposition and c is an adjustable parameter [99].

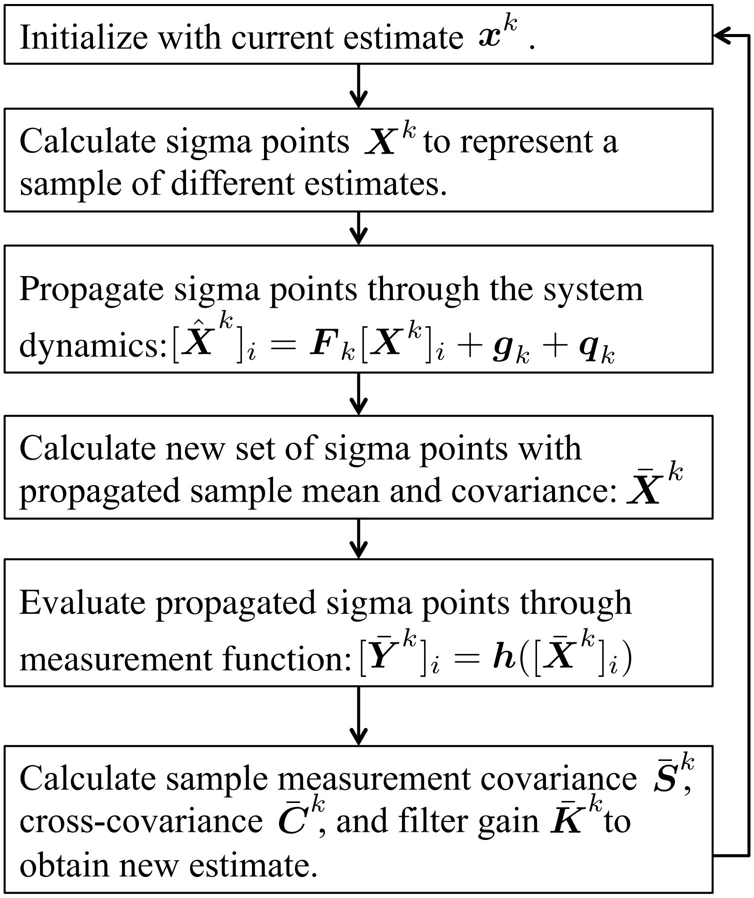

Initialize with current estimate $\boldsymbol{x}^k$.
Calculate sigma points $\boldsymbol{X}^k$ to represent a sample of different estimates.
Propagate sigma points through the system dynamics: $[\hat{\boldsymbol{X}}^k]_i = \boldsymbol{F}_k[\boldsymbol{X}^k]_i + \boldsymbol{g}_k + \boldsymbol{q}_k$
Calculate new set of sigma points with propagated sample mean and covariance: $\bar{\boldsymbol{X}}^k$
Evaluate propagated sigma points through measurement function: $[\bar{\boldsymbol{Y}}^k]_i = \boldsymbol{h}([\bar{\boldsymbol{X}}^k]_i)$
Calculate sample measurement covariance $\bar{\boldsymbol{S}}^k$, cross-covariance $\bar{\boldsymbol{C}}^k$, and filter gain $\bar{\boldsymbol{K}}^k$ to obtain new estimate.

Figure 4.1: Overview of the Unscented Kalman Filter.

4.2 Distributed Filtering Algorithm for Load-Based Dynamic State Estimation

In this section, we present our distributed algorithm for load-based dynamic state estimation (Algorithm 4.1) and detail its limited communication requirements. In each control area, we would like to estimate the local state while still taking into account the effects from the rest of the interconnected power system. Measurements of power flows on tie-lines (*i.e.*, a transmission line that connects buses across two different control areas) and measurements of power injections at border buses (*i.e.*, a bus with a neighbor in another control area) introduce couplings between different control areas. The power network buses are partitioned into N control areas. We say $i \in I$ if bus i belongs to control area I. Let the number of buses in control area I be n_I. Denote the neighbor set of bus i as $\mathcal{N}(i)$, and similarily, let $\mathcal{N}(I)$ be the set of neighboring nodes connected to control area I via a tie-line. We define the measurement set of control area I as follows

$$\begin{aligned} \boldsymbol{z}_I = \{\theta_i, V_i | i \in I\} \cup \{P_{ij}, Q_{ij} | i \in I \text{ or } j \in I\} \cup \\ \{P_i, Q_i | i \in I \text{ or } i \in \mathcal{N}(I)\}. \end{aligned} \tag{4.4}$$

In words, the set of measurements relevant to a control area include voltage phasor measurements at local buses, power flow measurements on internal lines and tie-lines, and power injection measurements at local buses and neighboring buses connected via a tie-line. In sum, $\boldsymbol{z}_I$ is the set of all measurements whose measurement function involves states in $\boldsymbol{x}_I = \{(\theta_i, V_i) | i \in I\}$. We develop our distributed algorithm based on performing a local unscented Kalman filter in each area with the following measurement set

$$\boldsymbol{z}_I = \boldsymbol{h}_I(\boldsymbol{x}_I, \boldsymbol{x}_{\mathcal{N}^*(I)}) + \boldsymbol{e}_I, \tag{4.5}$$

where the dependence of the measurements on the global state $\boldsymbol{x}$ can be simplified to those in control area I and in

$$\begin{aligned} \mathcal{N}^*(I) = &\{j \in \mathcal{N}(I)\} \cup \\ &\{l \in \mathcal{N}(j) \mid l \notin I, j \in \mathcal{N}(I), P_j \in \boldsymbol{z}_I\}. \end{aligned} \tag{4.6}$$

This is due to the functional form of the power system measurement equations, that can be found in [2]. We note that the measurements in (4.5) are not a partitioning of the measurements in (4.1) since a measurement can appear in more than one control area's measurement set (*e.g.*, power flow measurement of a tie-line). As detailed in Algorithm 4.1, each control area estimates its local states, $\boldsymbol{x}_I \in \mathbb{R}^{2n_I}$, and covariance $\boldsymbol{P}_I \in \mathbb{R}^{2n_I \times 2n_I}$. Due to the coupling from the measurements described above, areas need to communicate estimates and measurements as detailed in Step 7 of Algorithm 4.1. We stress that this communication occurs only once per time step in a non-iterative fashion and scales in terms of the number of border buses in a control area.

Algorithm 4.1 Distributed Load Based-Dynamic State Estmimation Algorithm

```
1: Initialize with given state $\boldsymbol{x}_I^0$ and covariance $\boldsymbol{P}_I^0$
2: for $k = 0 : K$ do
3:     Calculate $2(2n_I) + 1$ sigma points:
           $[\boldsymbol{X}_I^k]_1 = \boldsymbol{x}_I^k,$
           $[\boldsymbol{X}_I^k]_i = \boldsymbol{x}_I^k + c_I[\sqrt{\boldsymbol{P}_I^k}]_{i-1},\ (i{=}2,\ldots,2n_I{+}1)$
           $[\boldsymbol{X}_I^k]_i = \boldsymbol{x}_I^k - c_I[\sqrt{\boldsymbol{P}_I^k}]_{i-2n_I-1},\ (i{=}2n_I{+}2,\ldots,4n_I{+}1)$
4:     Propagate each sigma point through the system dynamics:
           $[\hat{\boldsymbol{X}}_I^k]_i = \boldsymbol{F}_I^k[\boldsymbol{X}_I^k]_i + \boldsymbol{g}_I^k,\ i = \{1, \ldots, 4n_I + 1\}$
5:     Calculate predicted state mean and predicted covariance:
           $\bar{\boldsymbol{x}}_I^k = \sum_{i=1}^{4n_I+1} w_i^{m,I} [\hat{\boldsymbol{X}}_I^k]_i$
           $\bar{\boldsymbol{P}}_I^k = \boldsymbol{Q}_I^k + \sum_{i=1}^{4n_I+1} w_i^{c,I} [([\hat{\boldsymbol{X}}_I^k]_i - \bar{\boldsymbol{x}}_I^k)([\hat{\boldsymbol{X}}_I^k]_i - \bar{\boldsymbol{x}}_I^k)^T],$
   where weights $w_i^{c,I}$ and $w_i^{m,I}$ are designed as in [99].
```

```
    Calculate predicted sigma points:
        $[\bar{\boldsymbol{X}}_I^k]_1 = \bar{\boldsymbol{x}}_I^k$ ,
        $[\bar{\boldsymbol{X}}_I^k]_i = \bar{\boldsymbol{x}}_I^k + c_I[\sqrt{\bar{\boldsymbol{P}}_I^k}]_{i-1},\ (i{=}2,\ldots,2n_I{+}1)$
        $[\bar{\boldsymbol{X}}_I^k]_i = \bar{\boldsymbol{x}}_I^k - c_I[\sqrt{\bar{\boldsymbol{P}}_I^k}]_{i-2n_I-1},\ (i{=}2n_I{+}2,\ldots,4n_I{+}1)$
    Communicate with neighboring areas to obtain $\boldsymbol{x}_{\mathcal{N}^*(I)}^k$ and $\{P_i, Q_i | i \in \mathcal{N}(I)\}$.
    Evaluate measurement equations at each predicted sigma point and at neighboring estimates:
        $[\bar{\boldsymbol{Y}}_I^k]_i = \boldsymbol{h}_I\left([\bar{\boldsymbol{X}}_I^k]_i, \boldsymbol{x}_{\mathcal{N}^*(I)}^k\right),\ i = \{1, \ldots, 4n_I + 1\}$
    Calculate the mean, measurement covariance, and measurement state cross-covariance:
        $\boldsymbol{\mu}_I^k = \sum_{i=1}^{4n_I+1} w_i^{m,I}[\bar{\boldsymbol{Y}}_I^k]_i$
        $\boldsymbol{S}_I^k = \boldsymbol{R}_I + \sum_{i=1}^{4n_I+1} w_i^{c,I}[([\bar{\boldsymbol{Y}}_I^k]_i - \boldsymbol{\mu}_I^k)([\bar{\boldsymbol{Y}}_I^k]_i - \boldsymbol{\mu}_I^k)^T]$
        $\boldsymbol{C}_I^k = \sum_{i=1}^{4n_I+1} w_i^{c,I}[([\bar{\boldsymbol{X}}_I^k]_i - \bar{\boldsymbol{x}}_I^k)([\bar{\boldsymbol{X}}_I^k]_i - \bar{\boldsymbol{x}}_I^k)^T]$
    Calculate the filter gain, state estimate, and state covariance at the next time step:
        $\boldsymbol{x}_I^{k+1} = \bar{\boldsymbol{x}}_I^k + \boldsymbol{K}_I^k[\boldsymbol{z}_I^k - \boldsymbol{\mu}_I^k],\ \boldsymbol{K}_I^{k+1} = \boldsymbol{C}_I^k[\boldsymbol{S}_I^k]^{-1}$
        $\boldsymbol{P}_I^{k+1} = \bar{\boldsymbol{P}}_I^k - \boldsymbol{K}_I^k\boldsymbol{S}_I^k[\boldsymbol{K}_I^k]^T$
end for
```

4.2.1 Comparison to Centralized UKF Algorithm

In the centralized version of the UKF algorithm, the correlation between any pair of measurements is calculated. Similar to Step 9 of Algorithm 4.1, the sample measurement covariance matrix in the centralized setting is given by

$$\boldsymbol{S}^k = \boldsymbol{R} + \sum_{i=1}^{4n+1} w_i^c[([\bar{\boldsymbol{Y}}^k]_i - \boldsymbol{\mu}^k)([\bar{\boldsymbol{Y}}^k]_i - \boldsymbol{\mu}^k)^T]. \tag{4.7}$$

However, due to the dependence of the power system measurement equations on a small number of localized states, $\boldsymbol{S}^k$ exhibits a sparsity structure related to the underlying network structure. A numerical example of the sample measurement covariance is given in Figure 4.2. The distributed algorithm takes advantage of the lack of correlation between certain variables. To motivate this, we characterize analytically the sparsity of the measurement covariance in the centralized setting under a linear approximation to the measurement function. We stress that this is only an approximation since the equations are nonlinear but find this to be well-verified in practice.

Due to space constraints, we exclude power injection measurements in this analysis. Define the following set:

$$\begin{aligned} T_a \overset{\text{def}}{=} \{ &\text{ the set of state variables that the functional form} \\ &\text{for measurement } z_a \text{ depends upon.}\} \end{aligned} \tag{4.8}$$

For example, measurement function $z_a = P_{ij}(\theta_i, V_i, \theta_j, V_j)$ depends upon $T_a = \{\boldsymbol{x}_i, \boldsymbol{x}_j\}$.

Proposition 4.1. *Under Assumption 4.1 and excluding power injection measurements, the sparsity*

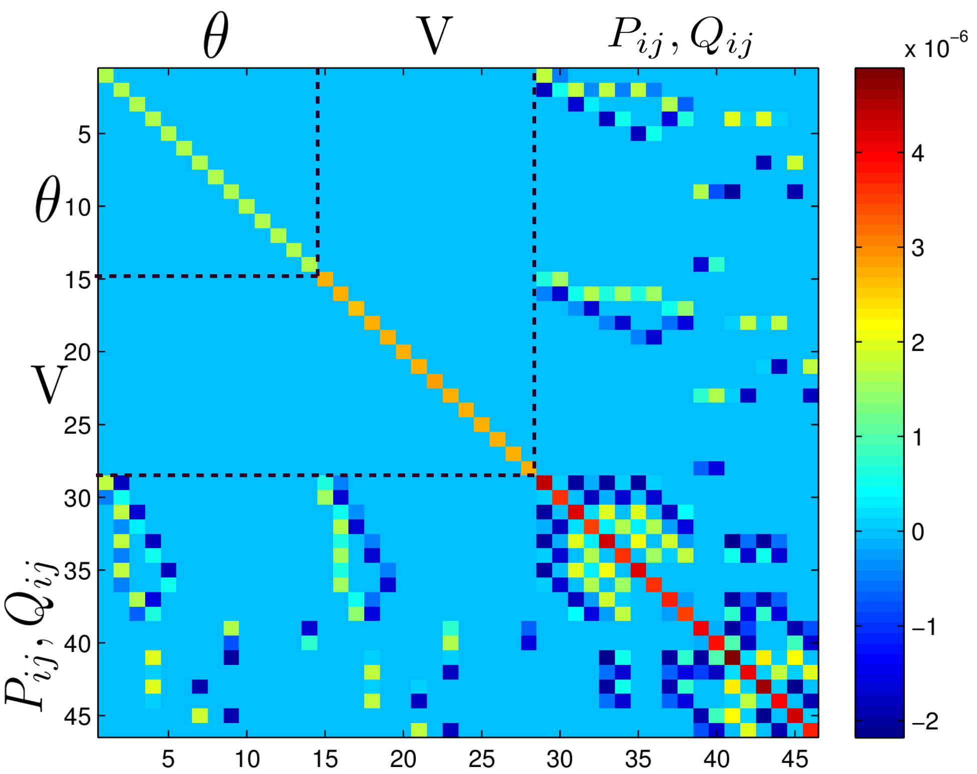


Figure 4.2: Example demonstrating sparsity of the measurement covariance matrix $\boldsymbol{S}^k$ on the IEEE 14-bus test system. Measurements shown include voltage phase angles (yellow diagonal), voltage magnitudes (orange diagonal), and power flows (red diagonal).

of the linearized measurement covariance matrix, $\boldsymbol{S}_*$, in the centralized setting is as follows:

$$[\boldsymbol{S}_*]_{ab} = \begin{cases} s_{ab} \neq 0 & \textit{if } T_a \cap T_b \neq \emptyset \\ 0 & \textit{otherwise} \end{cases}, \tag{4.9}$$

where entry $[\boldsymbol{S}_*]_{ab}$ refers to the sample covariance between measurements z_a and z_b.

Proof. Under Assumption 4.1, the process noise has a diagonal covariance matrix $\boldsymbol{Q}^k$, and the true underlying state $\boldsymbol{x}^k$ is a random vector with Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_x^k, \boldsymbol{Q}^k)$. Then, $\boldsymbol{x}^{k+1} =$

$\boldsymbol{F}_k\boldsymbol{x}^k + \boldsymbol{g}_k + \boldsymbol{q}_k$ remains an uncorrelated random vector, because from Assumption 4.1 the matrix $\boldsymbol{F}_k$ introduces no mixing among the components of $\boldsymbol{x}^k$. The quantity $\boldsymbol{y}^{k+1} = \boldsymbol{h}(\boldsymbol{x}^{k+1})$ is a nonlinear function of the random vector $\boldsymbol{x}^{k+1}$ and thus remains a random vector with some mean $\boldsymbol{\mu}_y$ and covariance $\boldsymbol{S}_*$. By exploiting the functional form of the linearized power system measurement functions we can characterize the sparsity pattern of $\boldsymbol{S}_*$. The time-step index k is dropped for notational convenience. Consider a first-order Taylor expansion of the measurement functions $\boldsymbol{h}(\boldsymbol{x})$ about the point $\boldsymbol{\mu}_x$,

$$\bar{\boldsymbol{h}}(\boldsymbol{x};\boldsymbol{\mu}_x) \overset{\text{def}}{=} \boldsymbol{h}(\boldsymbol{\mu}_x) + \boldsymbol{J}(\boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x), \tag{4.10}$$

where the measurement Jacobian $\boldsymbol{J}(\boldsymbol{x})$ is given by

$$\boldsymbol{J}(\boldsymbol{x}) \overset{\text{def}}{=} \begin{bmatrix} \frac{\partial h_1(\boldsymbol{x})}{\partial \theta_1} & \cdots & \frac{\partial h_1(\boldsymbol{x})}{\partial V_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m(\boldsymbol{x})}{\partial \theta_1} & \cdots & \frac{\partial h_m(\boldsymbol{x})}{\partial V_n} \end{bmatrix}. \tag{4.11}$$

It is a well-known fact that a linear function of a Gaussian random vector remains a Gaussian random vector. Let

$$\bar{\boldsymbol{y}} = \bar{\boldsymbol{h}}(\boldsymbol{x};\boldsymbol{\mu}_x), \tag{4.12}$$

then since $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{Q})$, it follows that $\bar{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{h}(\boldsymbol{\mu}_x) + \boldsymbol{J}(\boldsymbol{\mu}_x)\boldsymbol{\mu}_x, \boldsymbol{J}(\boldsymbol{\mu}_x)\boldsymbol{Q}\boldsymbol{J}(\boldsymbol{\mu}_x)^T)$. Note that the rows of the measurement Jacobian refer to measurements and the columns refer to state variables. Then, the quantity

$$[\boldsymbol{S}_*]_{ab} = [\boldsymbol{J}(\boldsymbol{\mu}_x)\boldsymbol{Q}\boldsymbol{J}(\boldsymbol{\mu}_x)^T]_{ab} = \sum_{c=1}^{n} Q_{cc} J_{ac}(\boldsymbol{\mu}_x) J_{bc}(\boldsymbol{\mu}_x) \tag{4.13}$$

is nonzero if and only if $T_a \cap T_b \neq \emptyset$. □

From Proposition 4.1, we expect the sample measurement covariance matrix to have a localized covariance structure for power system state estimation since the set T_a includes at most a bus and its 2-hop neighbors. In our distributed method, we take advantage of this. We note that the distributed algorithm remains an approximation to the centralized algorithm due to the inability to fuse sigma points from different areas when evaluating the measurement function for tie-line power flows and border bus power injections. Sigma points from different areas cannot be easily fused due to the incompatibility in their dimension and sample covariances. However, as shown in the next section, we can still achieve reasonable performance with respect to the centralized estimates while making gains in computation time, scalability, and robustness.

4.3 Numerical Results

We compare our proposed distributed approach ('Prop. Distr. UKF') to the centralized UKF ('Centr.') and to a distributed UKF that uses no communcation of estimates between neighboring areas ('Distr. UKF No Comm'). This is the same as the ('Distributed UKF without consensus algorithm') used as a benchmark in [90]. To facilitate comparison, we also use the same setup as [90]: the IEEE 14-bus and 118-bus networks with the same control area topologies; simulations of 150 time instants with a linear trend of 20% on all load buses from time instants 50-100, with a random fluctuation of 3%; and the same noise and initialization. In our experiments, PMUs are placed at the border buses and several internal buses. We expect the algorithm in [90] to be more accurate than the proposed method at the cost of increased runtime and communication. As a performance metric, we consider the average absolute estimation error per control area. Let $\widehat{\theta}^j_{I,k}$ and $\widehat{V}^j_{I,k}$ be the estimate of the voltage phase angle and magnitude at bus j in control area I at time step k. The average absolute estimation error for control area I is

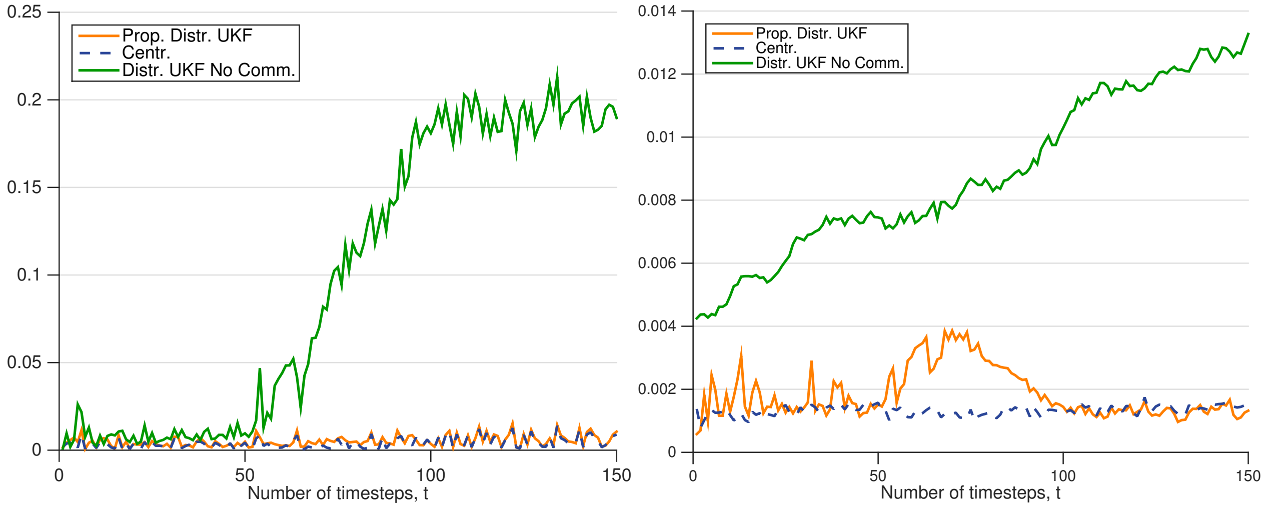
$$\xi^Z_I = \frac{1}{K}\frac{1}{N_I}\sum_{k=1}^{K}\sum_{j=1}^{N_I} |\widehat{Z}^j_{I,k} - Z^j_{I,k}|, \tag{4.14}$$

where K is the total number of time steps, N_I is the number of buses in control area $I,$ and $Z = \theta$ or $Z = V.$

The average absolute error for the 14-bus system with four control areas is given in Table 4.1, and the average absolute error for the 118-bus system with 3 control areas is given in Figures 4.3-4.5. We see that the performance of our method (‘Prop. Distr. UKF’) consistently outperforms the (‘Distr UKF No Comm’) and matches well with the centralized solution for the phase angle errors. For the voltage magnitude errors, our method sees an increase in error as the dynamics start changing but recovers due to the exchange of neighboring estimates unlike (‘Distr. UKF No Comm’). As expected, the centralized algorithm achieves the best performance since it has access to all measurements. This comes at a high communication cost for large systems. Furthermore, we find that the performance of our algorithm improves with the number of PMUs, unlike the (‘Distr. UKF No Comm’) benchmark. Since PMUs are expected to become increasingly prevalent, this is an attractive feature of our algorithm.

Table 4.1: Average estimation errors for IEEE 14-bus system.

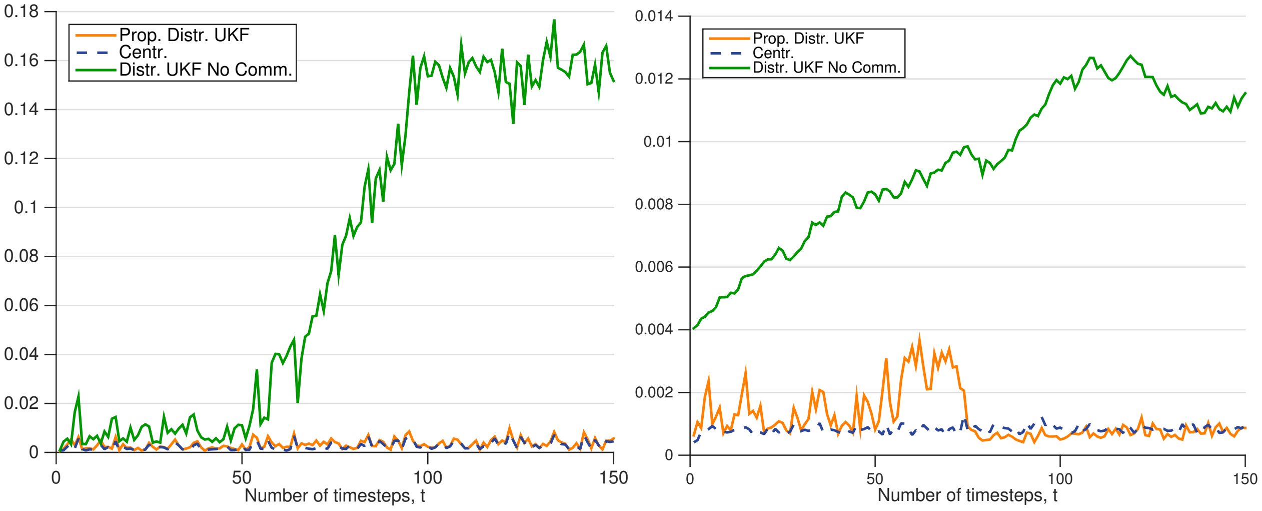
	Distributed UKF No Communication	**Proposed Distributed UKF**	**Centralized**
ξ_1^θ	0.0182	0.0048	0.0016
ξ_2^θ	0.0177	0.0092	0.0047
ξ_3^θ	0.0260	0.0087	0.0042
ξ_4^θ	0.0269	0.0100	0.0048
ξ_1^V	0.0097	0.0032	0.0012
ξ_2^V	0.0088	0.0022	0.0020
ξ_3^V	0.0077	0.0014	0.0008
ξ_4^V	0.0070	0.0016	0.0010



(a) Average Error ξ_1^θ (Area 1)

(b) Average Error ξ_1^V (Area 1)

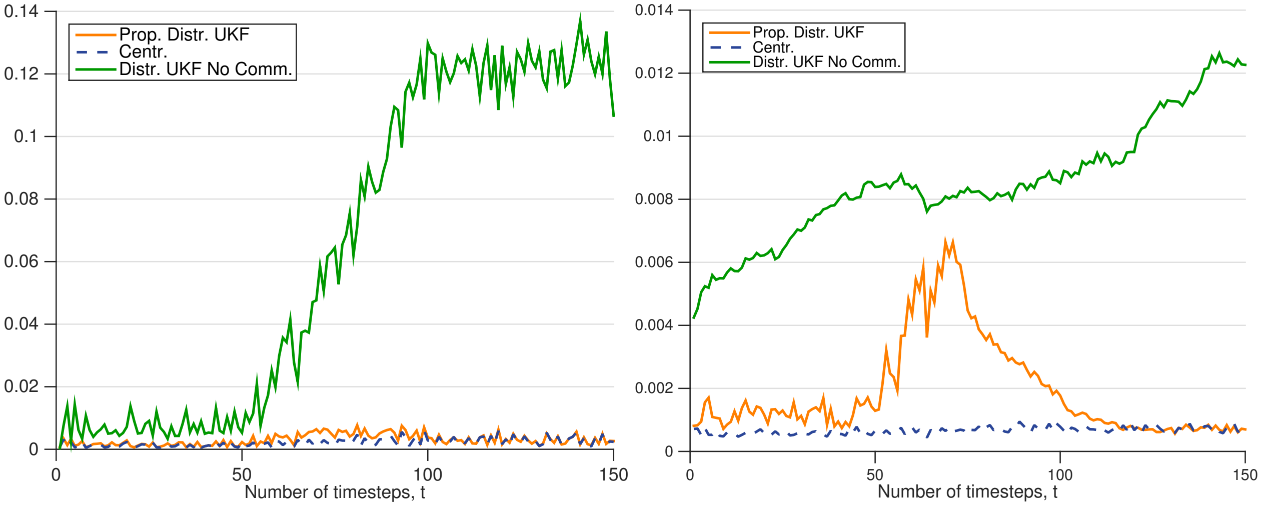
Figure 4.3: The average absolute error in voltage phase angle (a) and voltage magnitude (b) is shown for the 118-bus system for control area 1 using the proposed method, the centralized UKF, and a UKF with no communication.



(a) Average Error ξ_2^θ (Area 2)

(b) Average Error ξ_2^V (Area 2)

Figure 4.4: The average absolute error in voltage phase angle (a) and voltage magnitude (b) is shown for the 118-bus system for control area 2 using the proposed method, the centralized UKF, and a UKF with no communication.



(a) Average Error ξ_3^{θ} (Area 3)

(b) Average Error ξ_3^{V} (Area 3)

Figure 4.5: The average absolute error in voltage phase angle (a) and voltage magnitude (b) is shown for the 118-bus system at control area 3 using the proposed method, the centralized UKF, and a UKF with no communication.

4.4 Conclusion

The proposed method is an approximation to the unscented Kalman filter that allows for quicker computation time and fewer communication requirements than previous methods. The physics of power systems, namely the localized dependence of the measurement equations on voltage states, is utilized to approximately decouple the problem and motivate the use of local UKFs with limited communication. Since our algorithm requires only communication of local information with neighboring areas, the computation time scales very favorably with the network size as it depends on the number of border buses within a control area and not on the total size of the network. The physics of power systems, namely the localized dependence of the measurement equations on voltage states, is utilized to approximately decouple the problem and motivate the use of a decoupled UKF with limited computation and communication. Future research directions include extending this method to the information-filter form and exploring approaches for fusing sigma points for state estimates across control areas.

Chapter 5

Semi-Distributed Algorithms for Linearized Optimal Power Flow

OPTIMAL POWER FLOW (OPF) solves for the values of power generation and responsive loads that optimize a particular objective, such as minimizing operation costs and maximizing user utility, subject to physical and operational constraints. In order for the grid operator to respond immediately and efficiently to fluctuations in load and generation, especially when there is a large penetration of renewable energy, a solution to OPF needs to be available on the order of seconds or minutes. This demands new approaches to solving OPF, that allow for efficient use of parallel computing systems. Such approaches provide a way for different control areas to coordinate with limited and local communication.

Previous parallel approaches to the OPF problem are mostly first-order (*i.e.* gradient-based) methods. As an example, in [66], the authors develop an algorithm based on primal and dual decomposition techniques as a distributed solution to OPF. In [28, 33, 74], the ADMM algorithm is used as a distributed semidefinite programming solver, which is shown to have improved convergence results compared to those in [66]. However, ADMM-based approaches usually achieve a sublinear or linear convergence rate [10]. In [64], a fully distributed algorithm for OPF is proposed based on proximal message-passing algorithm, a version of ADMM.

On the other hand, primal-dual interior point (PDIP) methods, which are Newton-based approaches, demonstrate fast convergence behavior on various optimization problems, including OPF [54], [15]. In [53], a decentralized Newton-based approach based on the unlimited point algorithm is presented. The unlimited point algorithm introduces slack variables whereas PDIP methods successively solve a series of problems parameterized by an increasing barrier parameter. In [53], each control area solves a local optimization problem. In contrast, we decompose the global optimization problem across control areas.

PDIP methods usually achieve a superlinear convergence rate [107]. However, the primal-dual interior point method requires solving a large, sparse linear system at each iteration to calculate the search direction for updating the current estimate. For large systems, such as real-scale power networks with thousands of buses, direct inversion can be prohibitive in terms of computation.

To solve these series of large, sparse systems, we propose iterative, distributed methods based on matrix-splitting [102]. We design the matrix-splitting in order to exploit the sparsity and the topology that are inherent to the problem and the power network structure. Iteratively solving for the search direction introduces a set of inner iterations at each outer PDIP iteration. A centralized controller is used to calculate the step length and termination criteria.

As an initial step, we consider the linearized DC power flow model. In Chapter 6, we consider the extension to nonlinear problem settings. The main contributions of this chapter include:

1. A parallel PDIP method for DC-OPF is proposed. Its convergence rate is demonstrated on several systems and shown to scale favorably as the network size grows.

2. Local processing is used to limit communication requirements with the central processor. The search direction, which is the main computational burden, is calculated in a fully distributed way.

Notations: We use v_k to denote the kth entry of a vector $\boldsymbol{v}$. The (i,j)th entry of a matrix $\boldsymbol{M}$ is given by M_{ij}, and the ith row of a matrix is $[\boldsymbol{M}]_i$. The transpose of a vector or matrix $\boldsymbol{X}$ is denoted by $\boldsymbol{X}^T$. We use $\boldsymbol{M} \succ 0$ to denote that $\boldsymbol{M}$ is positive definite. The $n \times n$ diagonal matrix formed by placing the entries of a vector $\boldsymbol{x} \in \mathbb{R}^n$ on the diagonal is $diag(\boldsymbol{x})$. For a vector-valued function $\boldsymbol{f} : \mathbb{R}^m \to \mathbb{R}^n$,

$$D\boldsymbol{f}(\boldsymbol{x}) \overset{\text{def}}{=} \begin{bmatrix} \nabla f_1(\boldsymbol{x})^T \\ \vdots \\ \nabla f_n(\boldsymbol{x})^T \end{bmatrix}.$$

The neighbor set of node i is $\mathcal{N}(i)$, and the “1-hop" neighbor set (*i.e.*, the neighbors of node i’s neighbors that are not directly connected to node i) of node i is $\mathcal{N}^{\dagger}(i)$. The transmission line l from bus i to bus j is written $l(i,j)$. Let the subset of neighbors of bus i for which i is the “end-bus" of the line be denoted $\mathcal{N}_S(i) \overset{\text{def}}{=} \{j | l(j,i)\}$.

5.1 Problem Statement

5.1.1 DC Power Flow Model

We consider a multi-area interconnected power network, denoted by an undirected graph $(\mathcal{N}, \mathcal{E})$ with a set $\mathcal{N} \triangleq \{1, 2, \ldots, n\}$ of buses and a set $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ of transmission lines connecting the buses. Let the total number of transmission lines be $|\mathcal{E}| = L$. Let $\boldsymbol{f}$ denote the real power flow along every branch in the power network $\boldsymbol{f} \in \mathbb{R}^L$. The power injection profile, $\boldsymbol{p} \in \mathbb{R}^n$, is the value of the net real power injection at every bus. A positive power injection denotes a generation bus, whereas a negative power injection denotes a demand bus. Let $\boldsymbol{\theta} \in \mathbb{R}^{n-1}$ be the voltage phase angle at each bus except bus 1 (the reference bus). In the DC Power Flow model, the power injections and power flows are linearly related to the voltage phase angles [5, 40]:

$$\boldsymbol{p} = \boldsymbol{B\theta} \tag{5.1}$$

$$\boldsymbol{f} = \boldsymbol{DA\theta}. \tag{5.2}$$

Let b_l be the susceptance of line $l(i, j)$ from bus i to bus j, and $\boldsymbol{b}$ the vector of all line susceptances. The bus-susceptance $\overline{\boldsymbol{B}}$ matrix is

$$[\overline{\boldsymbol{B}}]_{ij} = \begin{cases} -b_l, & \text{if line } l \text{ connects buses } i \text{ and } j \\ 0, & \text{if buses } i \text{ and } j \text{ are not directly connected} \\ -\sum_{j \neq i} [\overline{\boldsymbol{B}}]_{ij}, & \text{if } i = j. \end{cases} \tag{5.3}$$

The matrix $\boldsymbol{B}$ in (5.1) is obtained by removing the first column from $\overline{\boldsymbol{B}}$. This results from setting the reference angle at bus 1 to 0. The matrix $\boldsymbol{B}$ encodes the network structure. The line-bus incidence matrix $\overline{\boldsymbol{A}}$ to be of dimension $m \times n$ with $[\overline{\boldsymbol{A}}]_{li} = +1$ and $[\overline{\boldsymbol{A}}]_{lj} = -1$ if the start bus of line l is i and the end bus of line l is j. The matrix $\boldsymbol{A}$ in (5.2) is $\overline{\boldsymbol{A}}$ with the first column removed. The matrix $\boldsymbol{D}$ in (5.2) is an $(m \times m)-$matrix whose lth entry is b_l.

5.1.2 DC-OPF Formulation

In order for our method to work well in a distributed setting (*i.e.*, limited communication requirements), it is important that the system for solving the Newton step has a sparse structure. At the same time, the system matrix should be reasonably conditioned so that the iterative solver is well-behaved. For these two reasons, we utilize the following formulation of DC-OPF:

$$\min_{\boldsymbol{\theta}} f_0(\boldsymbol{\theta}) \tag{5.4}$$

$$\text{s.t. } \underline{\boldsymbol{p}} \leq \boldsymbol{B\theta} \leq \overline{\boldsymbol{p}} \tag{5.5}$$

$$\underline{\boldsymbol{f}} \leq \boldsymbol{DA\theta} \leq \overline{\boldsymbol{f}}. \tag{5.6}$$

The optimization problem minimizes cost subject to line capacity and power injection constraints. The decision variables $\boldsymbol{\theta}$ are the voltage phase angles at each bus, excluding the reference bus. The lower bound on the power flow ensures that it cannot be too large in either direction along a line.

We consider a differentiable, convex objective function. In our simulations in Section 5.3, we use a quadratic cost function of the form

$$f_0(\boldsymbol{\theta}) = (\boldsymbol{B\theta} - \boldsymbol{p}^*)^T \boldsymbol{W} (\boldsymbol{B\theta} - \boldsymbol{p}^*), \tag{5.7}$$

where $\boldsymbol{p}^*$ is a vector whose entries are the nominal power demand for demand buses and zero for generation buses. The weighting matrix $\boldsymbol{W}$ is diagonal and allows to parametrize different generator costs and demand utility functions. This choice of objective function reasonably describes the cost behavior (*e.g.*, deviation from the nominal demand is penalized). The inequality constraints $\boldsymbol{g}(\boldsymbol{\theta}) \leq \mathbf{0}$ are linear and can be written as $\boldsymbol{G}\boldsymbol{\theta} \leq \mathbf{0}$ where:

$$\boldsymbol{G} \overset{\text{def}}{=} \begin{bmatrix} \boldsymbol{B} \\ -\boldsymbol{B} \\ \boldsymbol{D}\boldsymbol{A} \\ -\boldsymbol{D}\boldsymbol{A} \end{bmatrix}. \tag{5.8}$$

In summary, we have the following linear inequality constrained problem,

$$\min_{\boldsymbol{\theta}} \; f_0(\boldsymbol{\theta}) \tag{5.9a}$$

$$\text{s.t. } \boldsymbol{G}\boldsymbol{\theta} \leq \mathbf{0}. \tag{5.9b}$$

We propose using primal-dual interior point (PDIP) methods to solve the optimization problem (5.9a)-(5.9b). First, we will provide a basic overview of interior-point algorithms and then present our semi-distributed PDIP algorithm for DC-OPF.

5.1.3 Preliminaries on Interior-Point Methods for Linear Inequality Constrained Problems

Interior-point methods are used to solve optimization problems with inequality constraints and often demonstrate superlinear convergence behavior [107]. To deal with the inequality constraints, interior-point methods solve a series of equality-constrained problems that are a function of an adaptively changing parameter $\gamma > 0$ [11]. As γ increases, the inequality constraints are more strictly enforced. For each value of γ, Newton's method is applied to solve the modified Karush-Kuhn-Tucker (KKT) equations. For the DC-OPF formulation in (5.9a)-(5.9b), the modified KKT equations are given by

$$\boldsymbol{r}_\gamma(\boldsymbol{x}, \boldsymbol{\lambda}) = \begin{bmatrix} \boldsymbol{r}_{dual} \\ \\ \boldsymbol{r}_{cent} \end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix} \nabla f_0(\boldsymbol{\theta}) + \boldsymbol{G}\boldsymbol{\lambda} \\ \\ -\text{diag}(\boldsymbol{\lambda})\boldsymbol{G}\boldsymbol{\theta} - (1/\gamma)\mathbf{1} \end{bmatrix}, \tag{5.10}$$

where the centrality variables $\boldsymbol{\lambda}$ are dual variables associated with the inequality constraints. Let

$$\boldsymbol{y} \overset{\text{def}}{=} (\boldsymbol{\theta}, \boldsymbol{\lambda}), \;\; \Delta\boldsymbol{y} \overset{\text{def}}{=} (\Delta\boldsymbol{\theta}, \Delta\boldsymbol{\lambda}). \tag{5.11}$$

Primal-dual interior-point methods utilize Newton's method to solve the set of nonlinear equations $\boldsymbol{r}_\gamma(\boldsymbol{y}) = \mathbf{0}$ via the following first-order approximation:

$$D\boldsymbol{r}_\gamma(\boldsymbol{y'})\Delta\boldsymbol{y} = \boldsymbol{r}_\gamma(\boldsymbol{y'}), \tag{5.12}$$

where $\boldsymbol{y'}$ is the value of the previous PDIP iterate. The basic outline of an interior-point algorithm is summarized below [11].

Algorithm 5.1 DC-OPF Interior-Point Algorithm Overview [11]

```
Choose an initial feasible point for inequality constraints.
while termination critera unsatisfied do
    Set parameter $\gamma$ in terms of current iterate, $\boldsymbol{y}$.
    Compute the search direction, $\Delta\boldsymbol{y}$ by solving the linear system (5.12).
    Compute the step length, $d$.
    Compute the next iterate, $\boldsymbol{y}^{+} = \boldsymbol{y} + d\Delta\boldsymbol{y}$.
    Evaluate termination criteria to measure feasibility and optimality of current solution, $\boldsymbol{y}^{+}$.
end while
```

5.2 Semi-Distributed Primal-Dual Interior-Point (PDIP) Method for DC-OPF

Our aim is to develop a semi-distributed primal-dual interior-point (PDIP) algorithm for the OPF formulation in (5.9a)-(5.9b) with limited communication. We partition the buses of the power network into N control areas. There are three challenges to developing a distributed PDIP algorithm for optimal power flow: 1) solving for the search direction $\Delta\boldsymbol{y}$ in Step 4 of Algorithm 5.1, 2) calculating the step length d in Step 5, and 3) evaluating the termination criteria in a parallel way in Step 7.

5.2.1 Fully Distributed Search Direction Calculation

Calculating the search direction involves solving the large, sparse linear system in (5.12). Figure 5.1 is an example of the system matrix for the Newton step calculation.

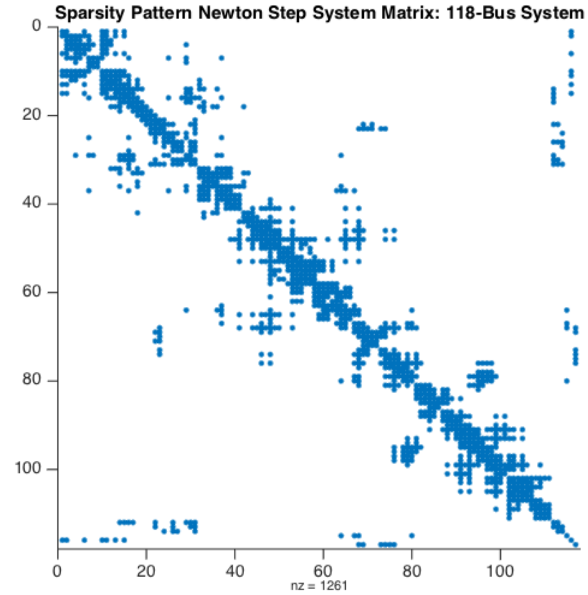


Figure 5.1: The sparsity pattern of the system matrix $\bar{\boldsymbol{C}}$ can be used to develop an efficient iterative inversion method for DC-OPF. Here is an example for a 118-bus system.

Specializing (5.12) for DC OPF, the Newton step requires solving

$$\bar{\boldsymbol{C}} \begin{bmatrix} \Delta\boldsymbol{\theta} \\ \Delta\boldsymbol{\lambda}_{\bar{p}} \\ \Delta\boldsymbol{\lambda}_{\underline{p}} \\ \Delta\boldsymbol{\lambda}_{\bar{f}} \\ \Delta\boldsymbol{\lambda}_{\underline{f}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_{dual} \\ \boldsymbol{r}_{cent,\bar{p}} \\ \boldsymbol{r}_{cent,\underline{p}} \\ \boldsymbol{r}_{cent,\bar{f}} \\ \boldsymbol{r}_{cent,\underline{f}} \end{bmatrix}, \tag{5.13}$$

where the system matrix is

$$\bar{\boldsymbol{C}} \overset{\text{def}}{=} \begin{bmatrix} 2\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B} & \boldsymbol{B}^T & -\boldsymbol{B}^T & (\boldsymbol{D}\boldsymbol{A})^T & -(\boldsymbol{D}\boldsymbol{A})^T \\ -\text{diag}(\boldsymbol{\lambda}_{\bar{p}})\boldsymbol{B} & -\text{diag}(\boldsymbol{B}\boldsymbol{\theta}-\bar{\boldsymbol{p}}) & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \text{diag}(\boldsymbol{\lambda}_{\underline{p}})\boldsymbol{B} & \boldsymbol{0} & -\text{diag}(\underline{\boldsymbol{p}}-\boldsymbol{B}\boldsymbol{\theta}) & \boldsymbol{0} & \boldsymbol{0} \\ -\text{diag}(\boldsymbol{\lambda}_{\bar{f}})\boldsymbol{D}\boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} & -\text{diag}(\boldsymbol{D}\boldsymbol{A}\boldsymbol{\theta}-\bar{\boldsymbol{f}}) & \boldsymbol{0} \\ \text{diag}(\boldsymbol{\lambda}_{\underline{f}})\boldsymbol{D}\boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\text{diag}(\underline{\boldsymbol{f}}-\boldsymbol{D}\boldsymbol{A}\boldsymbol{\theta}) \end{bmatrix}. \tag{5.14}$$

The dual variables associated with the upper bounds on the power injections, $\boldsymbol{B}\boldsymbol{\theta} - \bar{\boldsymbol{p}} \leq \boldsymbol{0}$, are denoted $\boldsymbol{\lambda}_{\bar{p}}$. Similarily, the dual variables for the lower bounds on the power injections, upper bounds on the power flows, and lower bounds on the power flows are denoted $\boldsymbol{\lambda}_{\underline{p}}, \boldsymbol{\lambda}_{\bar{f}}$, and $\boldsymbol{\lambda}_{\underline{f}}$, respectively. The right-hand side quantities are

$$\boldsymbol{r}_{dual} = 2\boldsymbol{B}^T\boldsymbol{W}(\boldsymbol{B}\boldsymbol{\theta} - \boldsymbol{p}^*) + \boldsymbol{B}^T(\boldsymbol{\lambda}_{\bar{p}} - \boldsymbol{\lambda}_{\underline{p}}) + (\boldsymbol{D}\boldsymbol{A})^T(\boldsymbol{\lambda}_{\bar{f}} - \boldsymbol{\lambda}_{\underline{f}}) \tag{5.15}$$

$$\boldsymbol{r}_{cent} \overset{\text{def}}{=} \begin{bmatrix} \boldsymbol{r}_{cent,\bar{p}} \\ \boldsymbol{r}_{cent,\underline{p}} \\ \boldsymbol{r}_{cent,\bar{f}} \\ \boldsymbol{r}_{cent,\underline{f}} \end{bmatrix} = \begin{bmatrix} \text{diag}(\boldsymbol{\lambda}_{\bar{p}})(\boldsymbol{B}\boldsymbol{\theta} - \bar{\boldsymbol{p}}) - \frac{1}{\gamma}\boldsymbol{1} \\ \text{diag}(\boldsymbol{\lambda}_{\underline{p}})(\underline{\boldsymbol{p}} - \boldsymbol{B}\boldsymbol{\theta}) - \frac{1}{\gamma}\boldsymbol{1} \\ \text{diag}(\boldsymbol{\lambda}_{\bar{f}})(\boldsymbol{D}\boldsymbol{A}\boldsymbol{\theta} - \bar{\boldsymbol{f}}) - \frac{1}{\gamma}\boldsymbol{1} \\ \text{diag}(\boldsymbol{\lambda}_{\underline{f}})(\underline{\boldsymbol{f}} - \boldsymbol{D}\boldsymbol{A}\boldsymbol{\theta}) - \frac{1}{\gamma}\boldsymbol{1} \end{bmatrix} \tag{5.16}$$

First, we reduce the system by eliminating the dual variables which only requires inverting a diagonal system and thus preserves sparsity. Eliminating the dual variables, the reduced system is

$$\boldsymbol{C}\Delta\boldsymbol{\theta} = \boldsymbol{w}, \tag{5.17}$$

where the system matrix and right-hand side are

$$\boldsymbol{C} = 2\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B} + \boldsymbol{G}^T\text{diag}(-\boldsymbol{G}\boldsymbol{\theta})^{-1}\text{diag}(\boldsymbol{\lambda})\boldsymbol{G} \tag{5.18}$$

$$\boldsymbol{w} = -\boldsymbol{r}_{dual} - \boldsymbol{G}^T\text{diag}(\boldsymbol{G}\boldsymbol{\theta})^{-1}\boldsymbol{r}_{cent}. \tag{5.19}$$

Since $\text{diag}(\boldsymbol{G}\boldsymbol{\theta})$ is diagonal, $\boldsymbol{C}$ exhibits a sparse structure. A matrix-splitting scheme similar to that developed for state estimation in Chapter 3 can be developed for $\boldsymbol{C}$ in (5.18). Consider decomposing the matrix $\boldsymbol{C}$ into the difference of two matrices $\boldsymbol{M}$ and $\boldsymbol{N}$. Provided the spectral radius $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ is less than one, let the fixed-point of the sequence

$$\Delta\boldsymbol{\theta}^{t+1} = \boldsymbol{M}^{-1}\boldsymbol{N}\Delta\boldsymbol{\theta}^{t} + \boldsymbol{M}^{-1}\boldsymbol{w}, \tag{5.20}$$

be $\Delta\boldsymbol{\theta}^{*}$, then $\boldsymbol{C}\Delta\boldsymbol{\theta}^{*} = \boldsymbol{w}$. The splitting (*i.e.* choice of $\boldsymbol{M}$ and $\boldsymbol{N}$) should be designed so that the matrix-splitting iterates in (5.20) i) converge and ii) are easy to calculate in a distributed way. Without loss of generality, assume the bus indices are consecutively assigned across control areas. The matrix $\boldsymbol{C}$ can be decomposed into the sum of a block-diagonal matrix, $\boldsymbol{D}$, and a matrix containing the remaining off-diagonal entries $\boldsymbol{E}$. The consecutive ordering of the bus indices described above allows for all entries of $\boldsymbol{C}$ corresponding to buses within the same control area to be contained within a single diagonal block. Specifically, let

$$D_{ij} = \begin{cases} C_{ij} & \text{if buses } i \text{ and } j \\ & \text{belong to the same control area} \, , \\ 0 & \text{otherwise} \end{cases} \tag{5.21}$$

$$E_{ij} = \begin{cases} C_{ij} & \text{if buses } i \text{ and } j \\ & \text{belong to different control areas} \, , \\ 0 & \text{otherwise} \end{cases} \tag{5.22}$$

yielding $\boldsymbol{C} = \boldsymbol{D} + \boldsymbol{E}$. We use the following matrix-splitting design

$$\boldsymbol{M} = \boldsymbol{D} + \tau\bar{\boldsymbol{E}},\ \boldsymbol{N} = \tau\bar{\boldsymbol{E}} - \boldsymbol{E}, \tag{5.23}$$

where τ is a scalar parameter and $\bar{\boldsymbol{E}}$ is a diagonal matrix whose ith diagonal entry equals

$$\bar{E}_{ii} \overset{\text{def}}{=} \sum_{j \neq i} |A_{ij}|. \tag{5.24}$$

This is a block-Jacobi scheme modified to be diagonally dominant. We have the following proposition which ensures convergence of the matrix-splitting iterates in (5.20).

Proposition 5.1. *Using the splitting in (5.23), for $\tau \geq \frac{1}{2}$, the iterative updates in (5.20) converge.*

Proof. The sequence $\{\Delta\boldsymbol{\theta}^t\}$ in (5.20) converges to its limit $\Delta\boldsymbol{\theta}^*$ as $t \to \infty$ if and only if the spectral radius of the matrix $\boldsymbol{M}^{-1}\boldsymbol{N}$ is strictly less than 1 [102]. Furthermore, if the sequence converges, the limit $\Delta\boldsymbol{\theta}^*$ is the solution of the system, (i.e., $\boldsymbol{C}\Delta\boldsymbol{\theta}^* = \boldsymbol{w}$). In order to have the spectral radius $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) < 1$, it is sufficient to have $\boldsymbol{C} = \boldsymbol{M} - \boldsymbol{N} \succ \boldsymbol{0}$ and $\boldsymbol{M} + \boldsymbol{N} \succ \boldsymbol{0}$ [26].

To show that $\boldsymbol{C} \succ 0$, first we note that the Hessian of the objective function $f_0(\boldsymbol{\theta}) \overset{\text{def}}{=} (\boldsymbol{B\theta} - \boldsymbol{p}^*)^T\boldsymbol{W}(\boldsymbol{B\theta} - \boldsymbol{p}^*)$ is diagonal and assumed to have strictly positive entries. Then, it is sufficient to show that 1) $\boldsymbol{G}$ has full column rank and 2) diag$(-\boldsymbol{G\theta})^{-1}$diag$(\boldsymbol{\lambda})$ has strictly positive entries. Note from (5.8) that $\boldsymbol{G}$ has full column rank by construction since $\boldsymbol{B}$ is the bus susceptance matrix with the column corresponding to the reference angle eliminated. For each constraint i the entries

$$[\text{diag}(\boldsymbol{G\theta})^{-1}\text{diag}(\boldsymbol{\lambda})]_{ii} = \frac{\lambda_i}{-[\boldsymbol{G}]_i\boldsymbol{\theta}} > 0 \tag{5.25}$$

since $\lambda_i > 0$ and $[\boldsymbol{G}]_i\boldsymbol{\theta} < 0$. Last, $\boldsymbol{M} + \boldsymbol{N}$ is positive definite due to its construction which makes it diagonally dominant. The remaining details follow as in the proof to Proposition 3.1 in Chapter 3. □

Control area I has its search direction update given by

$$\Delta\boldsymbol{\theta}_I^{t+1} = \boldsymbol{M}_I^{-1}\left[\sum_{J\neq I}\boldsymbol{N}_{IJ}\Delta\boldsymbol{\theta}_J^t + \boldsymbol{w}_I\right]. \tag{5.26}$$

The calculation for the search direction $\Delta\boldsymbol{y}$ concludes by using back-substitution of $\Delta\boldsymbol{\theta}$ to calculate $\Delta\boldsymbol{\lambda}$:

$$\Delta\boldsymbol{\lambda} = diag(\boldsymbol{G}\boldsymbol{\theta})^{-1}diag(\boldsymbol{\lambda})(\boldsymbol{r}_{cent} + \boldsymbol{G}\boldsymbol{\theta}). \tag{5.27}$$

In order to distribute the calculation, the dual variable $[\Delta\boldsymbol{\lambda}_{\bar{f}}]_l$ is assigned to bus i, the sending end of line $l(i,j)$. Recall the subset of neighbors of bus i, $\mathcal{N}_S(i) \overset{\text{def}}{=} \{j|l(j,i)\}$. This is the set of neighbors on lines for which i is the receiving end. The information exchange needed to calculate the search direction in a fully-distributed way is summarized in the following proposition.

Proposition 5.2. *In order to calculate $\Delta\boldsymbol{y}_I^{t+1}$ in a fully distributed way via (5.26) and (5.27), area I must receive $\{\theta_j,\ \Delta\theta_j \mid j \in \mathcal{N}^\dagger(I)\}$, $\{[\boldsymbol{\lambda}_{\bar{p}}]_j,\ [\boldsymbol{\lambda}_{\underline{p}}]_j \mid j \in \mathcal{N}(I)\}$, and $\{[\boldsymbol{\lambda}_{\bar{f}}]_j,\ [\boldsymbol{\lambda}_{\underline{f}}]_j \mid j \in \mathcal{N}_S(I)\}$.*

Proof. First, we analyze the communication requirements of the matrix-splitting iterations (5.26). Introduce the following diagonal matrices:

$$\boldsymbol{W}_{\bar{p}} \overset{\text{def}}{=} diag(\boldsymbol{B}\boldsymbol{\theta})^{-1}diag(\boldsymbol{\lambda}_{\bar{p}}) \tag{5.28}$$

$$\boldsymbol{W}_{\underline{p}} \overset{\text{def}}{=} diag(\boldsymbol{B}\boldsymbol{\theta})^{-1}diag(\boldsymbol{\lambda}_{\underline{p}}) \tag{5.29}$$

$$\boldsymbol{W}_{\bar{f}} \overset{\text{def}}{=} diag(\boldsymbol{D}\boldsymbol{A}\boldsymbol{\theta})^{-1}diag(\boldsymbol{\lambda}_{\bar{f}}) \tag{5.30}$$

$$\boldsymbol{W}_{\underline{f}} \overset{\text{def}}{=} diag(\boldsymbol{D}\boldsymbol{A}\boldsymbol{\theta})^{-1}diag(\boldsymbol{\lambda}_{\underline{f}}). \tag{5.31}$$

To characterize the sparsity of $\boldsymbol{C}$, it is useful to consider the following equivalent expression to

(5.18):

$$\boldsymbol{C} = \boldsymbol{B}^T(\boldsymbol{W} - \boldsymbol{W}_{\bar{p}} - \boldsymbol{W}_{\underline{p}})\boldsymbol{B} - \boldsymbol{A}^T\boldsymbol{D}(\boldsymbol{W}_{\bar{f}} + \boldsymbol{W}_{\underline{f}})\boldsymbol{D}\boldsymbol{A}. \tag{5.32}$$

The (i, j)th entry of the first term

$$[\boldsymbol{B}^T(\boldsymbol{W} - \boldsymbol{W}_{\bar{p}} - \boldsymbol{W}_{\underline{p}})\boldsymbol{B}]_{ij} = \sum_{k=1}^{n}[\boldsymbol{W} - \boldsymbol{W}_{\bar{p}} - \boldsymbol{W}_{\underline{p}}]_k[\boldsymbol{B}]_{ki}[\boldsymbol{B}]_{kj} \tag{5.33}$$

is nonzero only if i and j are at most "1-hop" neighbors since $\boldsymbol{B}$ has nonzero entries only for direct neighbors. Similarily, since the sparsity pattern of $\boldsymbol{A}$ only has entries between direct neighbors, the (i, j)th entry of the second term $[\boldsymbol{A}^T\boldsymbol{D}(\boldsymbol{W}_{\bar{f}} + \boldsymbol{W}_{\underline{f}})\boldsymbol{D}\boldsymbol{A}]_{ij}$ is nonzero only if i and j are at most "1-hop" neighbors. Since $[\boldsymbol{C}]_{ij} \neq 0$ only if $j \in \mathcal{N}(i) \cup \mathcal{N}^\dagger(i)$,

$$\sum_{J \neq I} \boldsymbol{N}_{IJ}\Delta\boldsymbol{\theta}_J^t = \sum_{J \in \mathcal{N}(I) \cup \mathcal{N}^\dagger(I)} \boldsymbol{N}_{IJ}\Delta\boldsymbol{\theta}_J^t. \tag{5.34}$$

Next, we analyze the communication requirements of the right-hand side term $\boldsymbol{\omega}_I$ in (5.19). Following a similar argument as above, the term $\boldsymbol{G}^T diag(\boldsymbol{G}\boldsymbol{\theta})^{-1}\boldsymbol{r}_{cent}$ requires information of the primal variables and their updates for at most "1-hop" neighbors. Therefore, area I requires $\{\theta_j,\ \Delta\theta_j \mid j \in \mathcal{N}^\dagger(I)\}$. To calculate $\boldsymbol{r}_{dual}$, area I must receive in addition $\{[\Delta\boldsymbol{\lambda}_{\bar{p}}]_j,\ [\Delta\boldsymbol{\lambda}_{\underline{p}}]_j \mid j \in \mathcal{N}(I)\}$ and $\{[\Delta\boldsymbol{\lambda}_{\bar{f}}]_j,\ [\Delta\boldsymbol{\lambda}_{\underline{f}}]_j \mid j \in \mathcal{N}_S(I)\}$ due to the terms $\boldsymbol{B}^T(\boldsymbol{\lambda}_{\bar{p}} - \boldsymbol{\lambda}_{\underline{p}}) + \boldsymbol{A}^T\boldsymbol{D}(\boldsymbol{\lambda}_{\bar{f}} - \boldsymbol{\lambda}_{\underline{f}})$ in (5.15). Last, we note that there are no additional communication requirements for the back substitution in (5.27). □

5.2.2 Parallel Step Length Calculation

Once the search direction $\Delta \boldsymbol{y}$ has been calculated, the next iterate $\boldsymbol{y}^+$ is produced from the current iterate $\boldsymbol{y}$ by

$$\boldsymbol{y}^+ = \boldsymbol{y} + d\Delta\boldsymbol{y}, \tag{5.35}$$

where d is the step length (Steps 5 and 6, Algorithm 5.1). A full Newton step corresponds to a step length of 1. In order to ensure feasibility (*e.g.*, $\boldsymbol{G\theta}^+ \leq \boldsymbol{0}$ and $\boldsymbol{\lambda}^+ \geq \boldsymbol{0}$), the step length is in general less than 1. One common approach to determining the step length is to use a backtracking line search [11]. To facilitate parallelization of the backtracking line search, we introduce local pre-processing schemes that

- Reduce the size of the information sent and the computation to be performed at the central coordinator.
- Allow for increased privacy by disclosing a surjective function of each control area's information to the central coordinator.

Our proposed parallel backtracking line search is presented in Algorithm 5.2 and based on the centralized one given in [11].

Algorithm 5.2 Parallel Backtracking Line Search Calculation

```
1: Each control area $I$ calculates $d^I_{max} = \min\{1, \min_{i|\lambda_i \in \boldsymbol{y}_I}\{-\lambda_i/\Delta\lambda_i < 0\}\}$ and sends it to the central coordinator.
2: Central coordinator calculates $d_{max} := \min_I\{d^I_{max}\}$ and sends it to the control areas.
```

```
 3: Each area $I$ sets $d_I := d_{max}$.
 4: while $\boldsymbol{G}_I\boldsymbol{\theta}^+ > \boldsymbol{0}$ do $d_I := 0.99 * d_{max}$.
 5: end while
 6: Each area $I$ sends $d_I$ to the central coordinator.
 7: The central coordinator calculates $d := \min_I\{d_I\}$ and sends it to control areas.
 8: Each area $I$ sets $d_I := d$.
 9: while $||[\boldsymbol{r}_\gamma(\boldsymbol{y}^+)]_I||_2 \geq (1 - \alpha d_I)||[\boldsymbol{r}_\gamma(\boldsymbol{y})]_I||_2$ do $d_I := \beta d$
10: end while
11: Each area sends $d_I$ to the central coordinator.
12: The central coordinator calculates $d := \min_I\{d_I\}$ and sends it to all areas.
```

The parameters α and β are chosen *a priori*. The convergence of the centralized version of this scheme in a finite number of steps is guaranteed in [11]. The convergence of the parallel scheme follows easily. Beside communication with the central processor, we consider the neighbor-to-neighbor communication of this algorithm. Information exchange between areas is used to calculate the norm of the residuals in Step 9 of Algorithm 5.2. Let the residual equations local to area I be $[\boldsymbol{r}_\gamma(\boldsymbol{y})]_I$. The information exchange is summarized in the following proposition.

Proposition 5.3. *To calculate $[\boldsymbol{r}_\gamma(\boldsymbol{y})]_I$, area I must receive $\{\theta_j|\ j \in \mathcal{N}^\dagger(I)\}$, $\{[\boldsymbol{\lambda}_{\bar{p}}]_j,\ [\boldsymbol{\lambda}_{\underline{p}}]_j \mid j \in \mathcal{N}(I)\}$, and $\{[\boldsymbol{\lambda}_{\bar{f}}]_j,\ [\boldsymbol{\lambda}_{\underline{f}}]_j \mid j \in \mathcal{N}_S(I)\}$.*

The residuals are contained in $\boldsymbol{\omega}_I$ and therefore the communication requirement analysis follows similarily to the proof of Proposition 5.2. Note that this information is already communicated in calculating the search direction.

5.2.3 Parallel Termination Checking

Criteria measuring the feasibility and optimality of the new iterate are evaluated to determine whether to terminate the algorithm. Again, we consider how local pre-processing can be done to more effectively parallelize the algorithm. Locally, at PDIP iteration k each area I calculates $\hat{\eta}_I = (\boldsymbol{G}\boldsymbol{\theta}_I^k)^T \boldsymbol{\lambda}_I^k$ in parallel. This is then sent to the central coordinator, which sums all contributions to calculate $\hat{\eta} = \sum_{I=1}^{N} \hat{\eta}_I$. Each area has already calculated the squared norm of its local residual vectors $||\boldsymbol{r}_{pri,I}||^2$ and $||\boldsymbol{r}_{dual,I}||^2$ during the backtracking line search in Section 5.2.2. These values are sent to the central coordinator, which calculates $||\boldsymbol{r}_{pri}|| = \sqrt{\sum_{I=1}^{N} ||\boldsymbol{r}_{pri,I}||^2}$ and $||\boldsymbol{r}_{dual}|| = \sqrt{\sum_{I=1}^{N} ||\boldsymbol{r}_{dual,I}||^2}$. The central coordinator checks the termination criterion and broadcasts whether or not to terminate to all control areas.

5.2.4 Overview of Parallel PDIP and Analysis of Communication Requirements

We detail the communication requirements for each of the steps in the proposed semi-distributed PDIP-OPF in Algorithm 5.3.

Algorithm 5.3 Parallel PDIP-OPF for Control Area I

```
Initialization: Set feasible initial point for $\boldsymbol{y}^0$, and set $||\boldsymbol{r}_{pri}||_2, ||\boldsymbol{r}_{dual}||_2$, and $\hat{\eta}$ to $10 * max(\epsilon_{feas}, \epsilon)$.
while $||\boldsymbol{r}_{pri}||_2 > \epsilon_{feas}, ||\boldsymbol{r}_{dual}||_2 > \epsilon_{feas}$, or $\hat{\eta} > \epsilon$ do
    Calculate $\gamma^k := \mu m/\hat{\eta}$
```

```
4:      Calculate $\boldsymbol{M}_I^{-1}\boldsymbol{w}_I$.
5:      Initialize $\Delta\boldsymbol{\theta}_I^t := \mathbf{0}$
6:      while $||\Delta\boldsymbol{\theta}_I^{t+1} - \Delta\boldsymbol{\theta}_I^t|| > \delta$ do
7:          Calculate $\Delta\boldsymbol{\theta}_I^{t+1} = \boldsymbol{M}_I^{-1}[\boldsymbol{w}_I + \sum_{J \neq I} \boldsymbol{N}_{IJ}\Delta\boldsymbol{\theta}_J^t]$
8:      end while
9:      $\Delta\boldsymbol{\theta}_I^k := \Delta\boldsymbol{\theta}_I^{t+1}$
10:     Substitute $\Delta\boldsymbol{\theta}_I^k$ to calculate $\Delta\boldsymbol{\lambda}_I^k$.
11:     Coordinate with central controller to calculate step size $d^k$ using backtracking line search.
12:     Calculate next PDIP iterate $\boldsymbol{y}_I^{k+1} = \boldsymbol{y}_I^k + d^k\Delta\boldsymbol{y}_I^k$.
13:     Coordinate with central controller to calculate $\hat{\eta} = -(\boldsymbol{G\theta}^{k+1})^T\boldsymbol{\lambda}^{k+1}$.
14: end while
```

- In Step 1, no communication is required. The $\boldsymbol{y}$ variables are initialized to a point feasible with respect to the inequality constraints. For example, we initialize the voltage phase angles $\boldsymbol{\theta}$ to zero and the $\boldsymbol{\lambda}$ centrality variables to small positive values. We assume each control area is given the global initial starting point *a priori*.

- In Steps 2 and 3, no communication is needed. The values for the residuals and surrogate duality gap, $\hat{\eta}$, are either set at initialization or calculated below in the previous iteration. The parameter μ is a constant, and $m = 2(n + L)$ is the number of inequality constraints.

- In Steps 5, 6, 9, and 12 all quantities are local, so no communication is needed.

- In Step 7, at most “1-hop" neighbor communication is needed. A fully distributed scheme for calculating Step 4 and its communication requirements (see Proposition 5.2) are given in Section 5.2.1.

- In Step 10, no additional communication is needed as values were communicated at Steps 4 and 7.

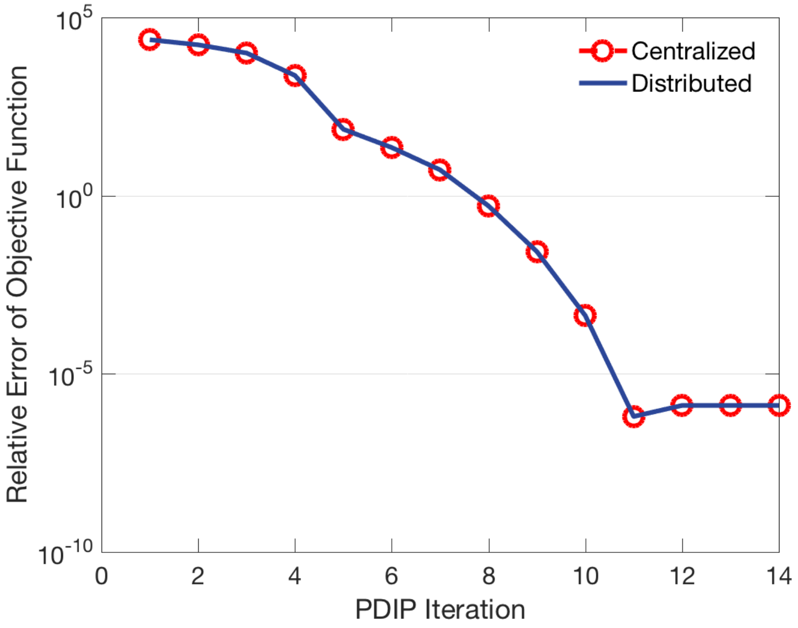
- In Step 11, coordination with a central coordinator is required with local processing as described in Algorithm 5.2. The neighbor-to-neighbor exchange is the same as required in Proposition 5.2 and needs only to be done once per PDIP iteration.

- In Step 13, local pre-processing in Section 5.2.3 is used to reduce the communication and computation with the central processor.

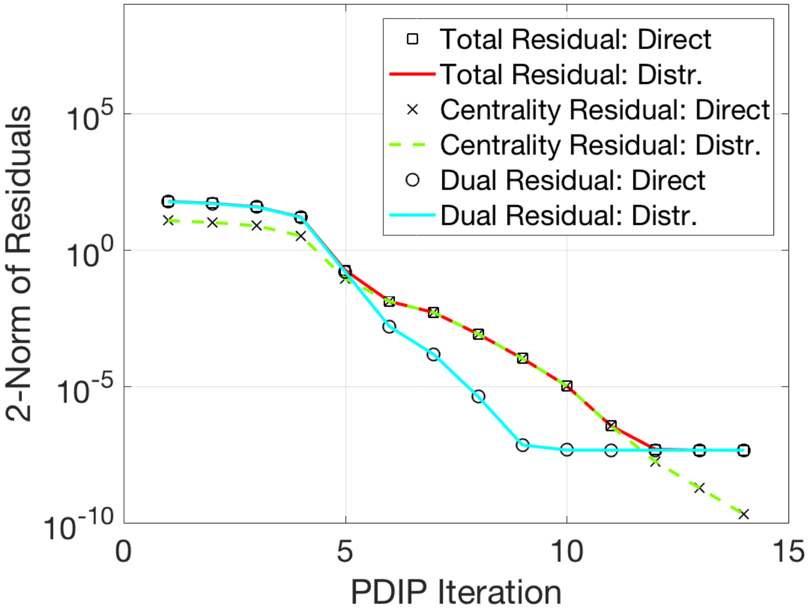
5.3 Numerical Results

In Figures 5.2 and 5.3, we study the performance of the parallel PDIP algorithm for two different size networks, the IEEE 14-bus and 118-bus systems [112]. The convergence using direct inversion (*i.e.*, “Centralized" method) to solve system (5.17) is compared to that using the iterative inversion (*i.e.*, distributed method). Both the centralized and distributed methods converge within roughly 10 PDIP iterations. Moreover, as the network size grows, the number of PDIP iterations needed remain roughly constant, and and the degree to which the network is partitioned can tune the computational cost of each iteration. Using more control areas corresponds to a more distributed configuration with comparable performance. In the simulations, both networks are partitioned into 4 control areas.

In Figures 5.2 and 5.3 (b), the convergence of the centrality residual ensures feasibility of the inequality constraints. The values used for μ, α, and β are 10, 0.01, and 0.3, respectively. The number of inner matrix-splitting iterations used in the distributed method is determined by the following termination criterion, $|\Delta\boldsymbol{\theta}_I^{+} - \Delta\boldsymbol{\theta}_I| < \delta$, where $\delta = 10^{-10}$. For 118-bus system, between 10^5 and 10^6 matrix-splitting iterates are required per iteration. The modified block-Jacobi method allows for a fully distributed calculation of the search direction. However, parallel direct inversion methods for sparse matrices like multifrontal LU decomposition can be an attractive choice when memory and centralized communication is not of high concern.

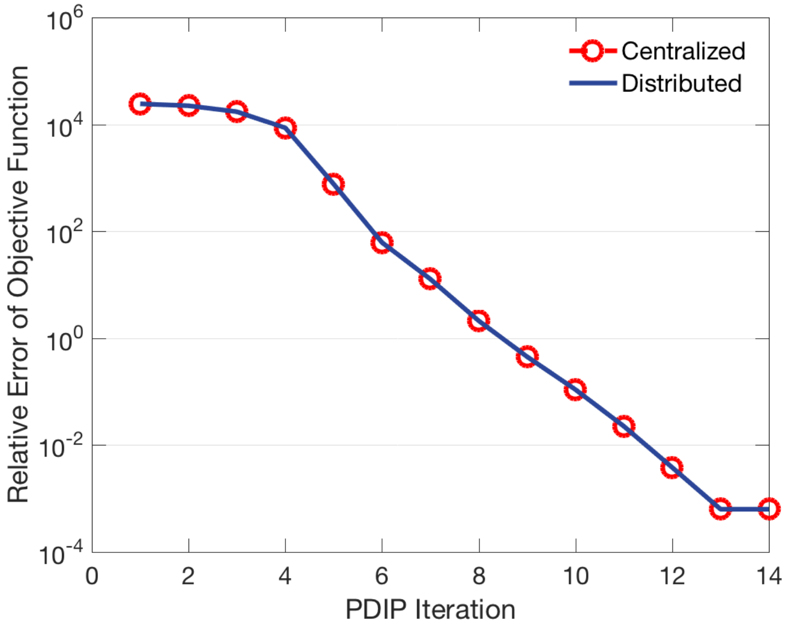


(a) Convergence of Objective Function Relative Error: 14-Bus System

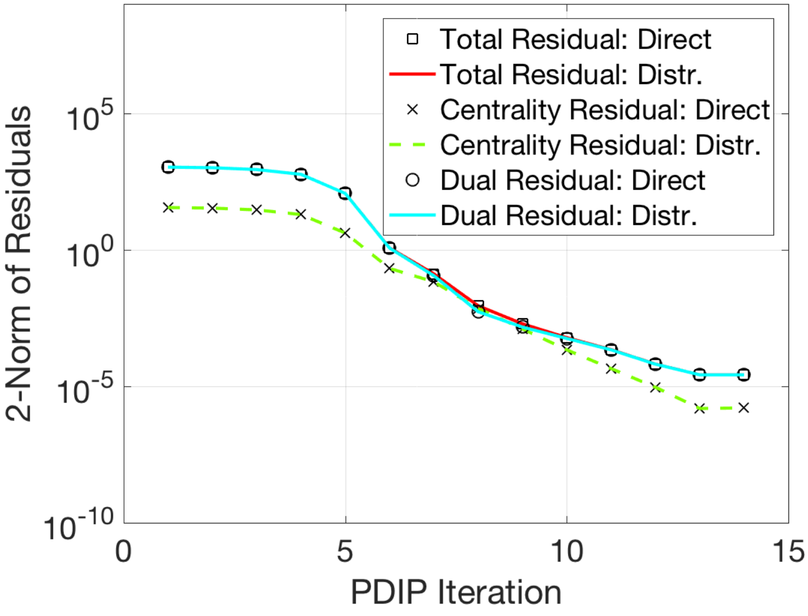


(b) Convergence of Residuals: 14-Bus Study

Figure 5.2: Results are demonstrated on the IEEE 14-bus test system. The relative error of the objective function (a) and residuals (b) converge within 10 PDIP iterations.



(a) Convergence of Objective Function Relative Error: 118-Bus System



(b) Convergence of Residuals: 118-Bus Study

Figure 5.3: Results are demonstrated on the IEEE 118-bus test system. The relative error of the objective function (a) and residuals (b) converge within 13 PDIP iterations.

5.4 Conclusion

We presented a semi-distributed primal-dual interior point method for the linearized optimal power flow problem. The favorable convergence properties of PDIP are demonstrated through numerical experiments, and the limited communication requirements for the Newton step computation are analyzed. While the number of inner-loop iterations required can be high, each inner-loop iteration is a computationally inexpensive arithmetic calculation. If a fully distributed multi-agent setting is of less interest than simply improving runtime, different solvers for the Newton step may be advantageous. This motivates our work in the next chapter on developing parallelized methods for the security-constrained optimal power flow problem.

Chapter 6

Parallel Algorithms for Security-Constrained Optimal Power Flow

SECURITY-CONSTRAINED OPTIMAL POWER FLOW (SC-OPF) seeks an optimal operating point that will be valid in the event of various contingencies, or equipment failures. In the power industry, the "N-1" criteria is often utilized to ensure security of the OPF solution. The "N-1" criteria requires the solution to OPF to be valid in the case of any single failure of equipment. This greatly increases the computational complexity of the optimization problem with respect to the non-security constrained OPF and further motivates the need for fast, robust algorithms.

There are two variations of SC-OPF. *Preventive* SC-OPF is utilized for optimizing over control variables that are not able to automatically respond to contingencies, whereas the *corrective* SC-OPF allows for re-dispatching of control variables. In our work, we consider the *preventive* SC-OPF, where different contingencies are coupled via a common set of control variables.

Early works introducing the SC-OPF problem include [3], in which a nonlinear programming approach is developed; as well as [81], which proposes an algorithm based on mathematical decomposition techniques for corrective SC-OPF that iteratively corrects the “base-case” solution without contingencies. The use of interior point algorithms for SC-OPF was first proposed in [103]. The interior point method is used for solving successive linear programs and studies demonstrate its advantage over the simplex method for SC-OPF.

More recently, there has been a research focus on overcoming the computational difficulties associated with solving SC-OPF problems [14, 16], such as large memory requirements and prohibitively long runtimes, due to the high-dimensionality of the problem. Since in practice only a fraction of contingencies are binding at the optimal point, contingency filtering schemes have been proposed to reduce the dimension of the problem by selecting a subset of contingencies to include in the optimization problem. However, contingency filtering schemes can be difficult to tune, and after filtering the problem may still remain too large to be solved efficiently [14]. Other approaches based on ADMM [20, 87] and Benders decomposition [42] have been proposed to break up the optimization problem into smaller optimization problems. In addition, in [8], the linearized DC SC-OPF problem is decentralized by coordinating different area sub-problems through a pricing mechanism based on the cost of electricity exchanges between adjacent areas.

Beyond ADMM and Benders decomposition, structure-exploiting interior point methods can be devised to solve SC-OPF in a computationally efficient manner. The main computational burden for interior point methods lies in calculating the Newton step or search direction, which involves solving a large, linear system. The Newton step calculation can be effectively decomposed by utilizing structure from the underlying network connectivity, physical laws of power networks, and SC-OPF problem formulation. In [23], the authors propose an interior point method for the linearized DC SC-OPF with a special preconditioning scheme for iteratively solving the Newton step. Similar to our work, they also investigate domain decomposition techniques in solving the Newton step. In [91], the authors design a parallel algorithm for SC-OPF to decompose the computation for subsets of post-contingency states using a GMRES iterative solver.

In this chapter, we present our parallel algorithm for SC-OPF. Our algorithm utilizes two layers of parallelization: 1) across contingencies and 2) across buses. We utilize techniques inspired by domain decomposition and reordering schemes that allow us to reduce the system matrix in parallel while preserving sparsity.

Notations: The number of equality (and inequality) constraints for contingency case k is $n_{eq(k)}$ (and $n_{ineq(k)}$). The ith entry of a vector of equations $\boldsymbol{h}(\boldsymbol{x})$ is $[\boldsymbol{h}(\boldsymbol{x})]_i$. For a function $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u})$, the gradient with respect to $\boldsymbol{x}$ is $\nabla_x \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u})$.

6.1 Problem Statement

To introduce security-constrained OPF, consider the general structure of the OPF problem without security constraints:

$$\min_{\boldsymbol{x},\boldsymbol{u}} f_0(\boldsymbol{x}, \boldsymbol{u}) \tag{6.1a}$$

$$\text{s.t. } \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) \leq \boldsymbol{0} \tag{6.1b}$$

$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{0}, \tag{6.1c}$$

where $\boldsymbol{x}$ are state variables (*e.g.*, bus voltages) and $\boldsymbol{u}$ are control variables (*e.g.*, power injections). The inequality constraints enforce operation constraints, such as transmission line and generation capacities, and the equality constraints enforce power balance at each bus. Under the AC power flow model, the equality constraints are nonlinear. The general structure for the security-constrained OPF is:

$$\min_{\boldsymbol{x}^{(0)},\ldots,\boldsymbol{x}^{(K)},\boldsymbol{u}} \quad f_0(\boldsymbol{u}) \tag{6.2a}$$

$$\text{s.t. } \boldsymbol{g}_u(\boldsymbol{u}) \leq \boldsymbol{0} \tag{6.2b}$$

$$\text{For each contingency } k = 0, \ldots, K:$$

$$\boldsymbol{g}_k(\boldsymbol{x}^{(k)}) \leq \boldsymbol{0} \tag{6.2c}$$

$$\boldsymbol{h}_k(\boldsymbol{x}^{(k)}, \boldsymbol{u}) = \boldsymbol{0}. \tag{6.2d}$$

In our work, we consider line failure contingencies. A set of state variables and constraints are introduced *per* contingency. For example, in the event of a line trip contingency, new power flows and voltages result. The state variables $\boldsymbol{x}^{(0)}$ correspond to the base case without any contingencies. This is a preventative SC-OPF formulation as opposed to a corrective SC-OPF formulation since it requires the state variables under any contingency to remain valid. The control variables $\boldsymbol{u}$ couple the different contingencies. There are various models for the power flow equations leading to different SC-OPF (and OPF) formulations. In the previous chapter, we considered only the DC-OPF problem. In this chapter, we develop our algorithm under three different formulations: 1) DC SC-OPF (linear), 2) Branch Flow SC-OPF (or BR SC-OPF) (convex relaxation), and 3) AC SC-OPF (nonconvex). For all formulations, we assume the objective function $f_0(\boldsymbol{p})$ is convex and a separable function of the power injections. We consider a multi-area interconnected power network, denoted by an undirected graph $(\mathcal{N}, \mathcal{E})$ with a set $\mathcal{N} \triangleq \{1, 2, \ldots, n\}$ of buses and a set $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ of m transmission lines connecting the buses. Under a single line failure contingency k, the set of $(m-1)$ remaining operational lines is $\mathcal{E}_k$.

6.1.1 DC SC-OPF

The DC SC-OPF formulation we consider is a linear optimization program with state variables equal to the bus voltages $\boldsymbol{x} := \boldsymbol{\theta}$ (excluding the reference bus) and control variables set to the power injections $\boldsymbol{u} := \boldsymbol{p}$. Since power injections do not ramp up or down power production or consumption instantaneously, they will remain unchanged immediately after a contingency and are therefore assigned in the *control variables* $\boldsymbol{u}$ [1]. The equality constraints enforce power balance at each bus, which are linear under the DC model, and the inequality constraints are limits on the power injections and bidirectional power flows:

$$\min_{\{\boldsymbol{\theta}^{(k)}\}_{k=0}^{K}, \boldsymbol{p}} \quad f_0(\boldsymbol{p}) \tag{6.3a}$$

$$\text{s.t. } \underline{\boldsymbol{p}} \leq \boldsymbol{p} \leq \bar{\boldsymbol{p}} \tag{6.3b}$$

For each contingency $k = 0, \ldots, K$:

$$\boldsymbol{B}_k \boldsymbol{\theta}^{(k)} - \boldsymbol{p} = \boldsymbol{0} \tag{6.3c}$$

$$\underline{\boldsymbol{f}} \leq \boldsymbol{D}_k \boldsymbol{A}_k \boldsymbol{\theta}^{(k)} \leq \bar{\boldsymbol{f}}, \tag{6.3d}$$

[1] In this context, control variables are not controlled in real-time. Rather the optimization problem SC-OPF determines the optimal set points for the control variable quantities. If a load power injection is not adjustable, this is reflected in the tightness of the inequality constraints (6.3b).

where the matrices $\boldsymbol{A}_k, \boldsymbol{B}_k,$ and $\boldsymbol{D}_k$ are defined as in Section 5.1.1. These matrices depend on the grid connectivity. Since we consider contingencies to be line trips, different contingencies correspond to different line sets, $\mathcal{E}_k$, and different $\boldsymbol{A}_k, \boldsymbol{B}_k,$ and $\boldsymbol{D}_k$. Note that (6.3) is a specialization of the general form (6.2).

6.1.2 Branch Flow (BR) SC-OPF

As a natural next step towards the full nonlinear AC SC-OPF problem, we consider the branch flow (BR) SC-OPF problem. First, we present some preliminaries about the branch flow power model. For radial power networks, the SC-OPF based on the branch flow model can be relaxed to a convex problem through two relaxations. A relaxation is exact, if every solution of the relaxed problem is a solution to the original problem and vice versa. The twice relaxed OPF is exact given some mild conditions [37]. In this section, we introduce the angle-relaxed branch flow model [38].

For each branch $(i, j) \in \mathcal{E}$ the resistance is denoted, r_{ij}, and the reactance, x_{ij}, and for each node i, the shunt conductance is denoted, g_i, and the shunt susceptance, b_i. The squares of the voltage magnitudes at each bus are given by $\boldsymbol{v} = (v_1, \ldots, v_n)$. Similarly, $\boldsymbol{l} = (l_{ij}, (i, j) \in \mathcal{E})$ denotes the squares of the branch current magnitudes. Let $\boldsymbol{P} = (P_{ij}, (i, j) \in \mathcal{E})$ denote the active and $\boldsymbol{Q} = (Q_{ij}, (i, j) \in \mathcal{E})$ the reactive power flow from i to j. The active and reactive power injections at each bus are given by $\boldsymbol{p} = (p_1, \ldots, p_n)$ and $\boldsymbol{q} = (q_1, \ldots, q_n)$. The state variables for each contingency are $\boldsymbol{x}^{(k)} = \left(\boldsymbol{v}^{(k)}, \boldsymbol{l}^{(k)}, \boldsymbol{P}^{(k)}, \boldsymbol{Q}^{(k)}\right)^T$. The control variables are $\boldsymbol{u} = (\boldsymbol{p}, \boldsymbol{q})^T$. We refer to the following convex problem using the relaxation for the branch flow model as the BR

SC-OPF:

$$\min_{\{\boldsymbol{P}^{(k)},\boldsymbol{Q}^{(k)},\boldsymbol{l}^{(k)},\boldsymbol{v}^{(k)}\}_{k=0}^{K},\boldsymbol{p},\boldsymbol{q}} f_0(\boldsymbol{p}) \tag{6.4a}$$

$$\text{s.t. } \underline{\boldsymbol{p}} \leq \boldsymbol{p} \leq \bar{\boldsymbol{p}} \tag{6.4b}$$

$$\underline{\boldsymbol{q}} \leq \boldsymbol{q} \leq \bar{\boldsymbol{q}} \tag{6.4c}$$

For each contingency $k = 0, \ldots, K$:

$\forall j \in \mathcal{N}$:

$$-p_j + \sum_{l:(j,l)\in\mathcal{E}_k} P_{jl}^{(k)} - \sum_{i:(i,j)\in\mathcal{E}_k} \left(P_{ij}^{(k)} - r_{ij} l_{ij}^{(k)}\right) + g_j v_j^{(k)} = 0, \tag{6.4d}$$

$$-q_j + \sum_{l:(j,l)\in\mathcal{E}_k} Q_{jl}^{(k)} - \sum_{i:(i,j)\in\mathcal{E}_k} \left(Q_{ij}^{(k)} - x_{ij} l_{ij}^{(k)}\right) + b_j v_j^{(k)} = 0, \tag{6.4e}$$

$\forall (i,j) \in \mathcal{E}_k$:

$$-v_j^{(k)} + v_i^{(k)} - 2\left(r_{ij} P_{ij}^{(k)} + x_{ij} Q_{ij}^{(k)}\right) + \left(r_{ij}^2 + x_{ij}^2\right) l_{ij}^{(k)} = 0, \tag{6.4f}$$

$$\frac{P_{ij}^2 + Q_{ij}^2}{v_i} - l_{ij} \leq 0 \tag{6.4g}$$

The inequality constraint (6.4g) is a relaxation of the equality constraint:

$$l_{ij} v_i - P_{ij}^2 - Q_{ij}^2 = 0. \tag{6.5}$$

The inequality (6.4g) is convex for $v_i^{(k)} > 0$. In our case, $v_i^{(k)}$ is always positive since it is the squared magnitude of the voltage.

Note that (6.4) is a specialization of the general form in (6.2). The equality constraints (6.4d) -

(6.4f) are linear in the decision variables and can be written as:

$$\begin{bmatrix} \boldsymbol{H}_x^{(k)} & \boldsymbol{H}_u^{(k)} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^{(k)} \\ \\ \boldsymbol{u} \end{bmatrix} = \boldsymbol{0}. \tag{6.6}$$

The linear inequality constraints (6.4b) - (6.4c) and the nonlinear inequality constraints (6.4g) can be summarized as $\boldsymbol{G}_u \boldsymbol{u} \leq 0$ and $\boldsymbol{g}_k(\boldsymbol{x}^{(k)}) \leq \boldsymbol{0}$, respectively.

6.1.3 AC SC-OPF

The AC SC-OPF is a nonconvex optimization program. Improving solutions to AC SC-OPF remains one of the key challenges in power system operations [13]. The state variables are equal to the bus voltage phase angles, $\boldsymbol{\theta}$, and magnitudes, $|\boldsymbol{V}|$. The control variables are the real and reactive power injections $\boldsymbol{p}$ and $\boldsymbol{q}$. The conductance and susceptance of the transmission line between nodes i and j are G_{ij} and B_{ij}, and B_{ij}^{sh} is the shunt susceptance. We consider the following formulation of AC SC-OPF:

$$\min_{\{\boldsymbol{\theta}^{(k)}, |\boldsymbol{V}|^{(k)}\}_{k=0}^{K}, \boldsymbol{p}, \boldsymbol{q}} \quad f_0(\boldsymbol{p}) \tag{6.7a}$$

$$\text{s.t. } \underline{\boldsymbol{p}} \leq \boldsymbol{p} \leq \bar{\boldsymbol{p}} \tag{6.7b}$$

$$\underline{\boldsymbol{q}} \leq \boldsymbol{q} \leq \bar{\boldsymbol{q}} \tag{6.7c}$$

For each contingency $k = 0, \ldots, K$:

$\forall i \in \mathcal{N}$:

$$\sum_{i,j \in \mathcal{N}_i} |V|_i^{(k)} |V|_j^{(k)} (G_{ij} \cos(\theta_i^{(k)} - \theta_j^{(k)}) + B_{ij} \sin(\theta_i^{(k)} - \theta_j^{(k)})) - p_i = 0 \tag{6.7d}$$

$$\sum_{i,j\in\mathcal{N}_i} |V|_i^{(k)}|V|_j^{(k)}(G_{ij}\sin(\theta_i^{(k)}-\theta_j^{(k)}) - B_{ij}\cos(\theta_i^{(k)}-\theta_j^{(k)})) - q_i = 0 \quad (6.7e)$$

$$\forall(i,j)\in\mathcal{E}_k :$$

$$\underline{f_{ij}} \leq -|V|_i^{(k)}|V|_j^{(k)}(G_{ij}\cos(\theta_i^{(k)}-\theta_j^{(k)}) + B_{ij}\sin(\theta_i^{(k)}-\theta_j^{(k)}))$$

$$+ G_{ij}(|V|_i^{(k)})^2 \leq \bar{f_{ij}} \quad (6.7f)$$

$$\underline{q_{ij}} \leq -|V|_i^{(k)}|V|_j^{(k)}(G_{ij}\sin(\theta_i^{(k)}-\theta_j^{(k)}) - B_{ij}\cos(\theta_i^{(k)}-\theta_j^{(k)}))$$

$$- (B_{ij}^{sh} + B_{ij})(|V|_i^{(k)})^2 \leq \bar{q_{ij}} \quad (6.7g)$$

Power injection limits are (6.7b)-(6.7c), and power flow limits are (6.7f)-(6.7g). The inequality constraints enforce power injection and power flow capacity limits. The equality constraints enforce real (6.7d) and reactive (6.7e) power balance at each bus. The method we propose in the next section can be applied to the AC SC-OPF problem. However, unlike the DC SC-OPF and BR SC-OPF formulations, there are no convergence guarantees for the nonlinear AC SC-OPF.

6.2 Two Layer Parallel SC-OPF Algorithm

In SC-OPF, different contingencies are only coupled via the power injection control variables, yielding a sparse system matrix. We design a domain decomposition technique based on this sparsity to parallelize the problem across different contingencies. For each sub-problem associated with a given contingency, additional sparsity structure is given by the physical laws and limited connectivity of power networks. Variable reordering schemes are introduced to reduce the system in parallel while maintaining sparsity after each reduction. We design our method to utilize two

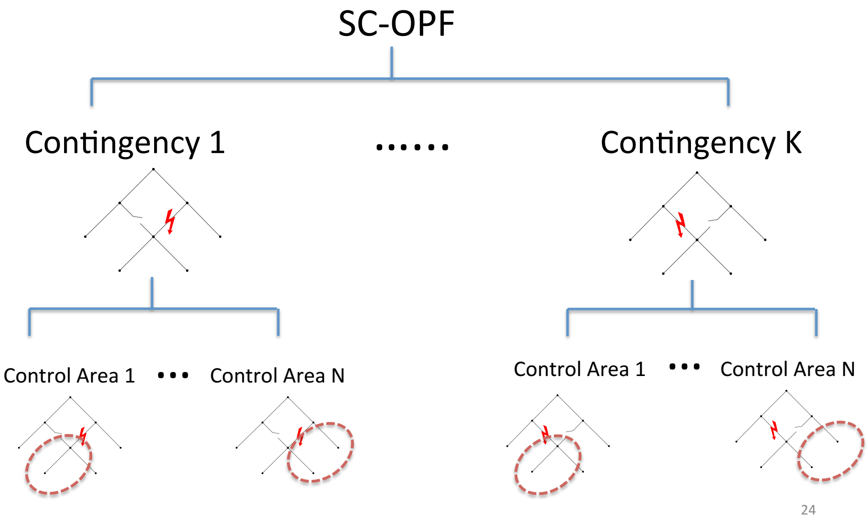


Figure 6.1: The SC-OPF is highly-parallelized, allowing a processor per control area and per contingency.

layers of parallelism: 1) across contingencies and 2) across buses in the network. The parallelization is effective due to the underlying sparsity patterns. A schematic of our parallelization strategy is shown in Figure 6.1.

6.2.1 Preliminaries on Primal-Dual Interior Point (PDIP) Methods for SC-OPF

In contrast to Chapter 5, where we applied PDIP methods to solve DC-OPF, the SC-OPF formulations have both equality and inequality consrtaints. As detailed in Section 6.1.1, the DC SC-OPF has linear equality and inequality constraints, whereas the BR SC-OPF has linear equality and nonlinear inequality constraints (see Section 6.1.2). Last, the AC SC-OPF has both nonlinear equality and inequality constraints (see Section 6.1.3). This leads to different derivations of the

Newton step. We begin with the derivation for the most general case, AC SC-OPF, and show how it specializes for the BR SC-OPF and DC SC-OPF problems. Simplified notation is introduced. For details on the optimization program formulations, we refer the reader back to Sections 6.1.1-6.1.3. Let

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}^{(0)} \\ \vdots \\ \boldsymbol{x}^{(K)} \end{bmatrix}, \ \boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{g}_0(\boldsymbol{x}^{(0)}) \\ \vdots \\ \boldsymbol{g}_K(\boldsymbol{x}^{(K)}) \\ \boldsymbol{g}_u(\boldsymbol{u}) \end{bmatrix}, \ \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \boldsymbol{h}_0(\boldsymbol{x}^{(0)}, \boldsymbol{u}) \\ \vdots \\ \boldsymbol{h}_K(\boldsymbol{x}^{(K)}, \boldsymbol{u}) \end{bmatrix}, \tag{6.8}$$

and $\boldsymbol{y}^T = \begin{bmatrix} \boldsymbol{x}^T & \boldsymbol{u}^T \end{bmatrix}$.

The modified KKT conditions ([11], Section 11.7.1) for problem (6.2) are

$$r_\gamma(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \overset{\text{def}}{=} \begin{bmatrix} \boldsymbol{r}_{dual} \\ \boldsymbol{r}_{cent} \\ \boldsymbol{r}_{pri} \end{bmatrix} = \begin{bmatrix} \nabla f_0(\boldsymbol{u}) + D\boldsymbol{g}(\boldsymbol{x})^T\boldsymbol{\lambda} + D\boldsymbol{h}(\boldsymbol{y})^T\boldsymbol{\nu} \\ -diag(\boldsymbol{\lambda})\boldsymbol{g}(\boldsymbol{x}) - \frac{1}{\gamma}\mathbf{1} \\ \boldsymbol{h}(\boldsymbol{y}) \end{bmatrix}. \tag{6.9}$$

Grouping together both primal and dual variables into $\boldsymbol{z} = (\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\nu})$ and $\Delta\boldsymbol{z} = (\Delta\boldsymbol{y}, \Delta\boldsymbol{\lambda}, \Delta\boldsymbol{\nu})$, the Newton step is obtained by solving the linear equation, $D\boldsymbol{r}_\gamma(\boldsymbol{z})\Delta\boldsymbol{z} = -\boldsymbol{r}_\gamma(\boldsymbol{z})$ or

$$\begin{bmatrix} \nabla^2 f_0(\boldsymbol{u}) + \sum_{i=1}^{n_{ineq}} \lambda_i \nabla^2 g_i(\boldsymbol{y}) & D\boldsymbol{g}(\boldsymbol{y})^T & D\boldsymbol{h}(\boldsymbol{y})^T \\ -diag(\boldsymbol{\lambda})D\boldsymbol{g}(\boldsymbol{y}) & -diag(\boldsymbol{g}(\boldsymbol{y})) & \mathbf{0} \\ D\boldsymbol{h}(\boldsymbol{y}) & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{y} \\ \Delta\boldsymbol{\lambda} \\ \Delta\boldsymbol{\nu} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{r}_{dual} \\ \boldsymbol{r}_{cent} \\ \boldsymbol{r}_{pri} \end{bmatrix}. \tag{6.10}$$

For BR SC-OPF, the equality constraints are linear, and therefore the modified KKT conditions simplify to

$$r_\gamma(\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\nu}) = \begin{bmatrix} \nabla f_0(\boldsymbol{u}) + D\boldsymbol{g}(\boldsymbol{y})^T\boldsymbol{\lambda} + \boldsymbol{H}^T\boldsymbol{\nu} \\ -diag(\boldsymbol{\lambda})\boldsymbol{g}(\boldsymbol{y}) - \frac{1}{\gamma} \\ \boldsymbol{H}\boldsymbol{y} \end{bmatrix}. \tag{6.11}$$

The Newton step calculation requires solving

$$\begin{bmatrix} \nabla^2 f_0(\boldsymbol{u}) + \sum_{i=1}^{n_{ineq}} \lambda_i \nabla^2 g_i(\boldsymbol{y}) & D\boldsymbol{g}(\boldsymbol{y})^T & \boldsymbol{H}^T \\ -diag(\boldsymbol{\lambda})D\boldsymbol{g}(\boldsymbol{y}) & -diag(\boldsymbol{g}(\boldsymbol{y})) & \boldsymbol{0} \\ \boldsymbol{H} & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{y} \\ \Delta\boldsymbol{\lambda} \\ \Delta\boldsymbol{\nu} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{r}_{dual} \\ \boldsymbol{r}_{cent} \\ \boldsymbol{r}_{pri} \end{bmatrix}. \tag{6.12}$$

Last, for DC-SCOPF, both the equality and inequality constraints are linear, yielding,

$$r_\gamma(\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\nu}) = \begin{bmatrix} \nabla f_0(\boldsymbol{u}) + \boldsymbol{G}^T\boldsymbol{\lambda} + \boldsymbol{H}^T\boldsymbol{\nu} \\ -diag(\boldsymbol{\lambda})\boldsymbol{G}\boldsymbol{y} - \frac{1}{\gamma} \\ \boldsymbol{H}\boldsymbol{y} \end{bmatrix}, \tag{6.13}$$

$$\begin{bmatrix} \nabla^2 f_0(\boldsymbol{u}) & \boldsymbol{G}^T & \boldsymbol{H}^T \\ -diag(\boldsymbol{\lambda})\boldsymbol{G} & -diag(\boldsymbol{G}\boldsymbol{y}) & 0 \\ \boldsymbol{H} & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{y} \\ \Delta\boldsymbol{\lambda} \\ \Delta\boldsymbol{\nu} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{r}_{dual} \\ \boldsymbol{r}_{cent} \\ \boldsymbol{r}_{pri} \end{bmatrix}. \tag{6.14}$$

6.2.2 Block-Bordered-Diagonal (BBD) Structure

The different contingencies are coupled only through their mutual dependence on the control variable $\boldsymbol{u}$. We will show that this kind of dependency leads to a block-bordered diagonal (BBD) matrix structure for solving the Newton step,

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{C}_u \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}^{(0)} & \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{C}^{(0)} \\ \mathbf{0} & \boldsymbol{M}^{(1)} & \dots & \mathbf{0} & \boldsymbol{C}^{(1)} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \boldsymbol{M}^{(K)} & \boldsymbol{C}^{(K)} \\ [\boldsymbol{C}^{(0)}]^T & [\boldsymbol{C}^{(1)}]^T & \dots & [\boldsymbol{C}^{(K)}]^T & \boldsymbol{C}_u \end{bmatrix}. \tag{6.15}$$

We design a reordering by contingency and show how it leads to BBD structure. The original ordering is according to (6.8). Let

$$\Delta \boldsymbol{z}^{(k)} = \begin{bmatrix} \Delta \boldsymbol{x}^{(k)} \\ \Delta \boldsymbol{\lambda}^{(k)} \\ \Delta \boldsymbol{\nu}^{(k)} \end{bmatrix}, \ \Delta \boldsymbol{z}^{(u)} = \begin{bmatrix} \Delta \boldsymbol{u} \\ \Delta \boldsymbol{\lambda}^{(u)} \end{bmatrix} \tag{6.16a}$$

$$\boldsymbol{M}^{(k)} = \begin{bmatrix} \sum_{i=1}^{n_{ineq(k)}} \lambda_i^{(k)} \nabla^2 [\boldsymbol{g}_k(\boldsymbol{x}^{(k)})]_i & [D\boldsymbol{g}_k(\boldsymbol{x}^{(k)})]^T & [D_x \boldsymbol{h}_k(\boldsymbol{y}^{(k)})]^T \\ -diag(\boldsymbol{\lambda}^{(k)}) D\boldsymbol{g}^{(k)}(\boldsymbol{x}^{(k)}) & -diag(\boldsymbol{g}^{(k)}(\boldsymbol{x}^{(k)})) & \mathbf{0} \\ D_x \boldsymbol{h}_k(\boldsymbol{y}^{(k)}) & \mathbf{0} & \mathbf{0} \end{bmatrix}, \ k = 0, \dots, K \tag{6.16b}$$

$$\boldsymbol{C}^{(k)} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ D_u \boldsymbol{h}_k(\boldsymbol{y}^{(k)}) & \mathbf{0} \end{bmatrix}, \ \boldsymbol{C}_u = \begin{bmatrix} \nabla^2_{\boldsymbol{u}} \boldsymbol{f}_0(\boldsymbol{u}) + \sum_{i=1}^{n_{ineq(u)}} \lambda_i^{(u)} \nabla^2 g_i^{(u)}(\boldsymbol{u}) & [D\boldsymbol{g}^{(u)}(\boldsymbol{u})]^T \\ -diag(\boldsymbol{\lambda}^{(u)}) D\boldsymbol{g}^{(u)}(\boldsymbol{u}) & -diag(\boldsymbol{g}^{(u)}(\boldsymbol{u})) \end{bmatrix}, \tag{6.16c}$$

where $D_x\boldsymbol{h}_k(\boldsymbol{y}^{(k)})$ is the matrix of gradients with respect to the state variables,

$$D_x\boldsymbol{h}_k(\boldsymbol{y}^{(k)}) \overset{\text{def}}{=} \begin{bmatrix} \nabla_x[\boldsymbol{h}_k(\boldsymbol{y}^{(k)})]_1^T \\ \dots \\ \nabla_x[\boldsymbol{h}_k(\boldsymbol{y}^{(k)})]_{neq(k)}^T \end{bmatrix}. \tag{6.17}$$

Note that for the branch-flow and DC models, $D_x\boldsymbol{h}_k(\boldsymbol{y}^{(k)})$ can be simplified to $\boldsymbol{H}_x^{(k)}$ since the equality constraints are linear. Similarily, for the DC model, $D\boldsymbol{g}_k(\boldsymbol{x}^{(k)})$ can be simplified to $\boldsymbol{G}_k$ since the inequality constraints are linear. Using the following reordering, (6.10) can be reordered as a BBD matrix for the SC-OPF problem:

$$\begin{bmatrix} \boldsymbol{M}^{(0)} & \boldsymbol{0} & \dots & \boldsymbol{0} & \boldsymbol{C}^{(0)} \\ \boldsymbol{0} & \boldsymbol{M}^{(1)} & \dots & \boldsymbol{0} & \boldsymbol{C}^{(1)} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \boldsymbol{M}^{(K)} & \boldsymbol{C}^{(K)} \\ [\boldsymbol{C}^{(0)}]^T & [\boldsymbol{C}^{(1)}]^T & \dots & [\boldsymbol{C}^{(K)}]^T & \boldsymbol{C}_u \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{z}^{(0)} \\ \Delta\boldsymbol{z}^{(1)} \\ \vdots \\ \Delta\boldsymbol{z}^{(K)} \\ \Delta\boldsymbol{z}^{(u)} \end{bmatrix} = -\boldsymbol{r}_*, \tag{6.18}$$

where

$$\boldsymbol{r}_* = \begin{bmatrix} \boldsymbol{r}_0 \\ \boldsymbol{r}_1 \\ \vdots \\ \boldsymbol{r}_K \\ \boldsymbol{r}_u \end{bmatrix}, \boldsymbol{r}_k = \begin{bmatrix} \boldsymbol{r}_{dual}^{(k)} \\ \boldsymbol{r}_{cent}^{(k)} \\ r_{pri}^{(k)} \end{bmatrix} = \begin{bmatrix} [D\boldsymbol{g}_k(\boldsymbol{x}^{(k)})]^T\boldsymbol{\lambda}^{(k)} + [D_x\boldsymbol{h}_k(\boldsymbol{y}^{(k)})]^T\boldsymbol{\nu}^{(\boldsymbol{k})} \\ -diag(\boldsymbol{\lambda}^{(k)})\boldsymbol{g}_k(\boldsymbol{x}^{(k)}) - \frac{1}{\gamma}\boldsymbol{1} \\ [D_x\boldsymbol{h}_k(\boldsymbol{y}^{(k)})]\boldsymbol{x}^{(k)} + [D_u\boldsymbol{h}_k(\boldsymbol{y}^{(k)})]\boldsymbol{u} \end{bmatrix}, \; k = 0, \dots, K \tag{6.19a}$$

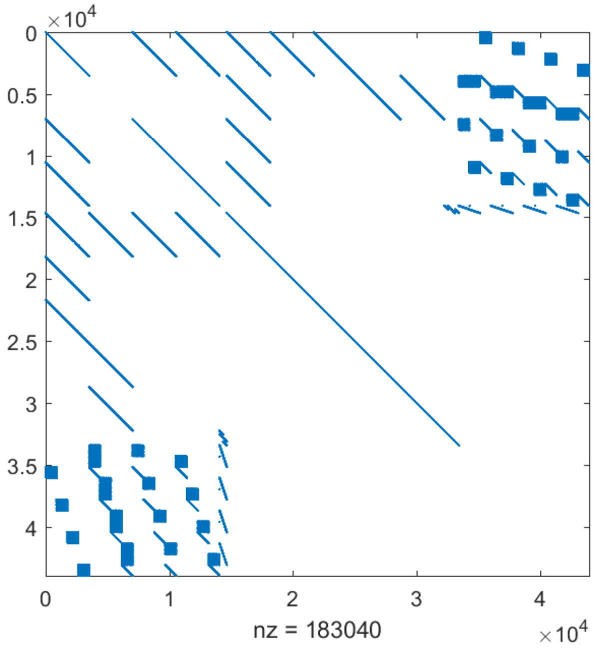


Figure 6.2: Sparsity pattern of Newton step system matrix (6.12) for an 880-bus system under the BR SC-OPF formulation. There are $nz = 183,040$ nonzero entries.

$$\boldsymbol{r}_u = \begin{bmatrix} \nabla_{\boldsymbol{u}} \boldsymbol{f}_0(\boldsymbol{u}) + [D\boldsymbol{g}_u(\boldsymbol{u})]^T \boldsymbol{\lambda}^{(u)} + \sum_{k=0}^{K} [D_u \boldsymbol{h}_k(\boldsymbol{y}^{(k)})]^T \boldsymbol{\nu}^{(k)} \\ -diag(\boldsymbol{\lambda}^{(u)}) \boldsymbol{g}_u(\boldsymbol{u}) - \frac{1}{\gamma}\mathbf{1} \end{bmatrix}. \tag{6.19b}$$

As a numerical example, the sparsity pattern for the Newton step system matrix under the original ordering (6.8) is shown in Figure 6.2 for an 880-bus system under the BR SC-OPF model. The sparsity pattern under the BBD reordering is shown in Figure 6.3.

6.2.3 Layer 1: Parallelizing Across Contingencies

We exploit the BBD structure (6.15) in order to decompose the Newton step calculation into K sub-problems, one for each contingency. The domain decomposition technique applied to solving (6.18) is presented in Algorithm 6.1 [93].

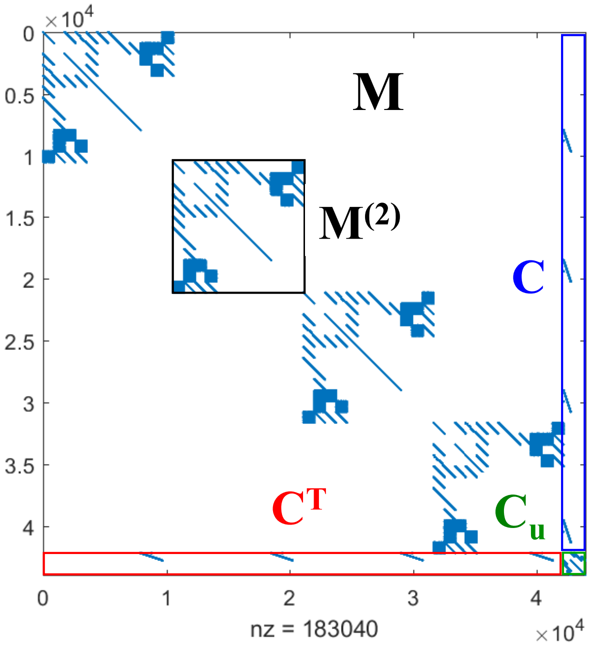


Figure 6.3: Sparsity pattern of Newton step system matrix (6.12) for an 880-bus system under the BR SC-OPF formulation using the Block-Bordered-Diagonal (BBD) reordering in (6.15). This example uses 3 contingencies, plus the base case, leading to 4 diagonal blocks with repeated structure. There are $nz = 183,040$ nonzero entries.

Algorithm 6.1 Block-Gaussian Elimination [93] for Solving 6.18

```
1: In parallel, for each contingency $k$ : Solve $\boldsymbol{M}^{(k)}\tilde{\boldsymbol{C}}^{(k)} = \boldsymbol{C}^{(k)}$ and $\boldsymbol{M}^{(k)}\tilde{\boldsymbol{r}}^{(k)} = \boldsymbol{r}^{(k)}$ for $\tilde{\boldsymbol{C}}^{(k)}$ and $\tilde{\boldsymbol{r}}^{(k)}$.
2: Collect solution $[\tilde{\boldsymbol{C}}]^T = \left[[\tilde{\boldsymbol{C}}^{(0)}]^T \ldots [\tilde{\boldsymbol{C}}^{(K)}]^T\right]$ and $[\tilde{\boldsymbol{r}}]^T = \left[[\tilde{\boldsymbol{r}}^{(0)}]^T \ldots [\tilde{\boldsymbol{r}}^{(K)}]^T\right]$ at central processor.
3: Compute $\tilde{\boldsymbol{r}}_u = \boldsymbol{r}_u - [\tilde{\boldsymbol{C}}]^T\tilde{\boldsymbol{r}}$.
4: Compute $\boldsymbol{S} = \boldsymbol{C}_u - [\boldsymbol{C}]^T\tilde{\boldsymbol{C}}$.
5: Solve $\boldsymbol{S}\Delta\boldsymbol{z}_u = \tilde{\boldsymbol{r}}_u$ for $\Delta\boldsymbol{z}_u$.
```

```
6: For each contingency $k$, compute $\Delta\boldsymbol{z}^{(k)} = \tilde{\boldsymbol{r}}^{(k)} - \tilde{\boldsymbol{C}}^{(k)}\Delta\boldsymbol{z}_u$.
```

The main computational burden of Algorithm 6.1 is Step 1. This step can be parallelized across contingencies without any communication requirements. We do note the results are sent to a centralized processor upon completion. The system in Step 5 is comparatively small compared to the linear solve that is parallelized in Step 1. For example, in the 880-bus test system in Figure 6.3, Step 5 requires solving a $1,770 \times 1,770$ system, whereas Step 1 breaks down solving a $42,212 \times 42,212$ system into four $10,553 \times 10,553$ systems, that are solved in parallel. Furthermore, as the number of contingencies increases, the size of $\boldsymbol{C}_u$ remains constant, while adding a processor per contingency allows the runtime in Step 1 to remain roughly constant.

6.2.4 Layer 2: Parallelizing Across Buses for BR SC-OPF

The system $\boldsymbol{M}^{(k)}$ is still large for power grids with many buses. Furthermore, since these systems demonstrate nice sparsity structure, it is advantageous to exploit this to further increase computational efficiency. The Layer 2 parallelization concerns how to parallelize solving subproblem k in Step 1 in Algorithm 6.1. For the DC SC-OPF, the techniques of Chapter 5 can be applied to the kth subproblem since its sparsity is similar to that of the non-security constrained OPF. Here we extend results from Chapter 5 to the branch flow model for radial, or tree-like, networks. Parallelization across buses for the nonlinear AC model is a topic of future research.

The strategy is to design reordering schemes that 1) allow variable eliminations and backsubtitutions with minimal communication and 2) that maintain sparsity. Instead of solving a large sparse system, runtime is decreased by solving a smaller, still sparse system. From (6.16b), (6.16c), (6.19a), the kth sub-problem can be succinctly written as:

$$\boldsymbol{M}^{(k)} \begin{bmatrix} \tilde{\boldsymbol{C}}^{(k)}_{dual} & \tilde{\boldsymbol{r}}^{(k)}_{dual} \\ \tilde{\boldsymbol{C}}^{(k)}_{cent} & \tilde{\boldsymbol{r}}^{(k)}_{cent} \\ \tilde{\boldsymbol{C}}^{(k)}_{pri} & \tilde{\boldsymbol{r}}^{(k)}_{pri} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & -r^{(k)}_{dual} \\ \boldsymbol{0} & -r^{(k)}_{cent} \\ \boldsymbol{H}^{(k)}_{u} & -r^{(k)}_{pri} \end{bmatrix} \tag{6.20}$$

The sparsity pattern for $\boldsymbol{M}^{(k)}$ in the 880-bus example is shown in Figure 6.4.

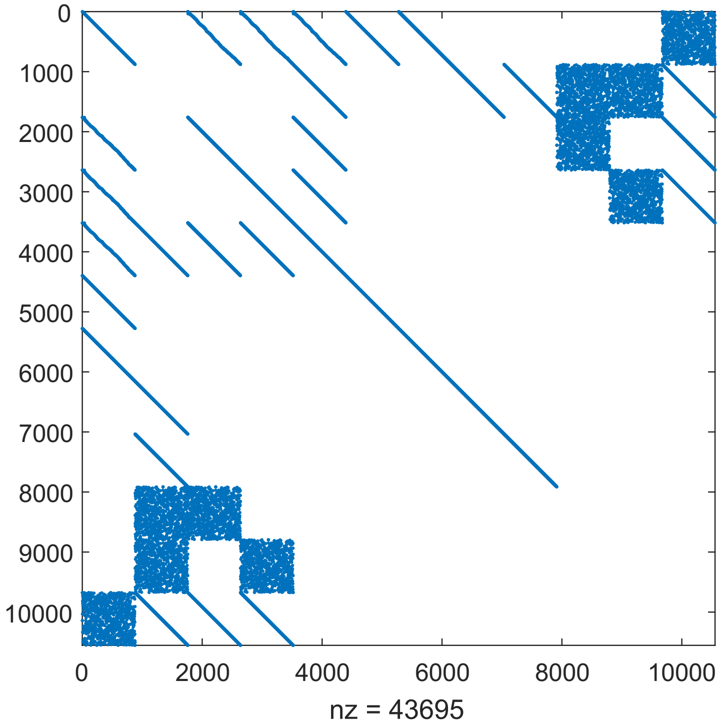


Figure 6.4: The sparsity pattern for the kth sub-problem system matrix $\boldsymbol{M}^{(k)}$ (6.16b) for an 880-bus radial test system under the branch flow model. There are $nz = 43,695$ nonzero entries.

System Reduction 1: The first elimination is of the second set of equations for solving $\tilde{\boldsymbol{C}}^{(k)}_{cent}$ and $\tilde{\boldsymbol{r}}^{(k)}_{cent}$. The reduced system is:

$$\begin{bmatrix} \boldsymbol{Q}_k & [\boldsymbol{H}^{(k)}_{x}]^T \\ \boldsymbol{H}^{(k)}_{x} & 0 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{C}}^{(k)}_{dual} & \tilde{\boldsymbol{r}}^{(k)}_{dual} \\ \tilde{\boldsymbol{C}}^{(k)}_{pri} & \tilde{\boldsymbol{r}}^{(k)}_{pri} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{b}^{(k)} \\ \boldsymbol{H}^{(k)}_{u} & \boldsymbol{r}^{(k)}_{pri} \end{bmatrix}, \tag{6.21}$$

where

$$\boldsymbol{Q}_k = \sum_{i=1}^{n_{ineq(k)}} \left(\lambda_i^{(k)} \nabla^2 [\boldsymbol{g}_k(\boldsymbol{x}^{(k)})]_i - \frac{\lambda_i^{(k)}}{[\boldsymbol{g}_k(\boldsymbol{x}^{(k)})]_i} \nabla [\boldsymbol{g}_k(\boldsymbol{x}^{(k)})]_i [\nabla [\boldsymbol{g}_k(\boldsymbol{x}^{(k)})]_i]^T \right) \tag{6.22}$$

$$\boldsymbol{b}^{(k)} = -\boldsymbol{r}_{dual}^{(k)} - [D\boldsymbol{g}_k(\boldsymbol{x}^{(k)})]^T diag(\boldsymbol{g}_k(\boldsymbol{x}^{(k)}))^{-1} \boldsymbol{r}_{cent}^{(k)}. \tag{6.23}$$

This involves inverting a diagonal matrix, namely $diag(\boldsymbol{g}_k(\boldsymbol{x}^{(k)}))$, that is actually applicable under all three models (DC, BR, AC SC-OPF) considered. Furthermore, this reduction step preserves sparsity. The sparsity of the reduced system matrix (6.21) for the 880-bus test system is shown in Figure 6.5.

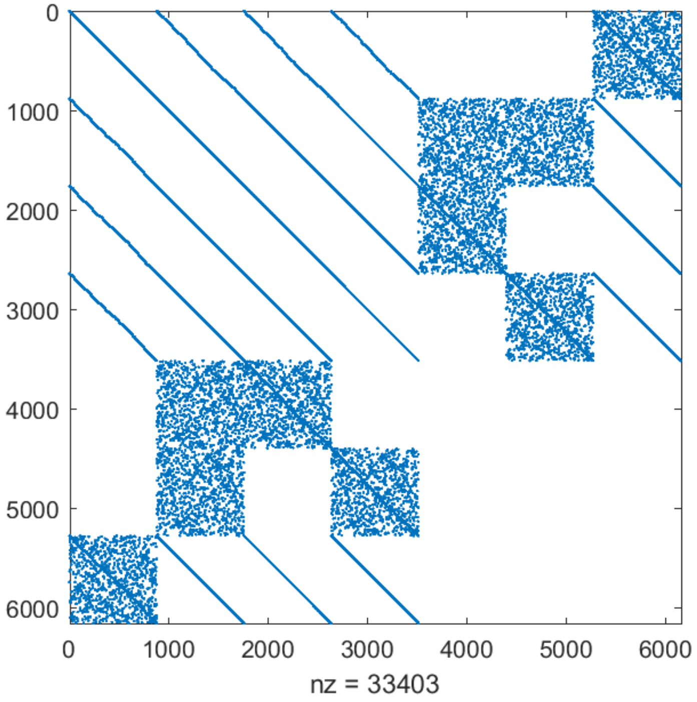


Figure 6.5: The sparsity pattern for the kth sub-problem system matrix after one reduction (6.21) for an 880-bus radial test system under the branch flow model. There are $nz = 33,403$ nonzero entries.

Reordering 1: We design a reordering scheme so that the upper-left block $\boldsymbol{Q}_k$ in (6.21) is block-diagonal under the BR SC-OPF model.

Proposition 6.1. *The decision variables $\boldsymbol{x}^{(k)}$ can be reordered so that $\boldsymbol{Q}_k$ in (6.22) is block-diagonal under the branch flow model for radial networks. The upper bound for the dimension of the blocks is $5 + 3max_i|C_i|$, where $C_i = \{j|(i,j) \in \mathcal{E}_k\}$ are the children of node i.*

Proof. Under the following reordering, the matrix $\boldsymbol{Q}_k$ in (6.22) is block-diagonal. The decision variables associated with a node (*i.e.*, v_i, p_i, q_i) are assigned to $\boldsymbol{x}_i$, and the decision variables associated with an edge (*i.e.*, l_{ij}, P_{ij}, Q_{ij}) are assigned to the sending end of the edge. This choice is not arbitrary but motivated by the fact that the inequality constraints of the form in (6.4g) involve v_i not v_j. The reordered vector $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$ contains n blocks of the form $\boldsymbol{x}_i = (v_i, \boldsymbol{l}_i, \{\boldsymbol{P}_{ij}\}_j, \{\boldsymbol{Q}_{ij}\}_j, p_i, q_i)$. Let $\mathcal{C}_i = \{j|(i,j) \in E\}$ be set of children of node i. Each block $\boldsymbol{x}_i$ belongs to a control area and is of dimension $5 + 3|\mathcal{C}_i|$, and each control area can contain one or more consecutive blocks. Apart from the inequality constraints (6.4g), all other inequalities are simple upper and lower bounds of the form, $\underline{x} \leq x_i \leq \bar{x}$. Since these inequalities are linear, $\boldsymbol{\nabla}^2 g_i = \boldsymbol{0}$ for all i, and since they are separable, $\boldsymbol{\nabla} g_i \boldsymbol{\nabla} g_i^T$ is a diagonal matrix. The inequalities in (6.4g) are the only ones that couple different decision variables, and from the functional form in (6.4g), they couple only the decision variables associated to the same node, $\boldsymbol{x}_i$. Therefore, we have that $\boldsymbol{Q}_k$ is block diagonal. □

The sparsity pattern of the matrix (6.21) under the reordering scheme is shown in Figure 6.6.

System Reduction 2: After reordering, $\boldsymbol{Q}_k$ is block-diagonal and can be inverted in parallel, assigning a processor to one or multiple blocks. Furthermore, this allows for a second system

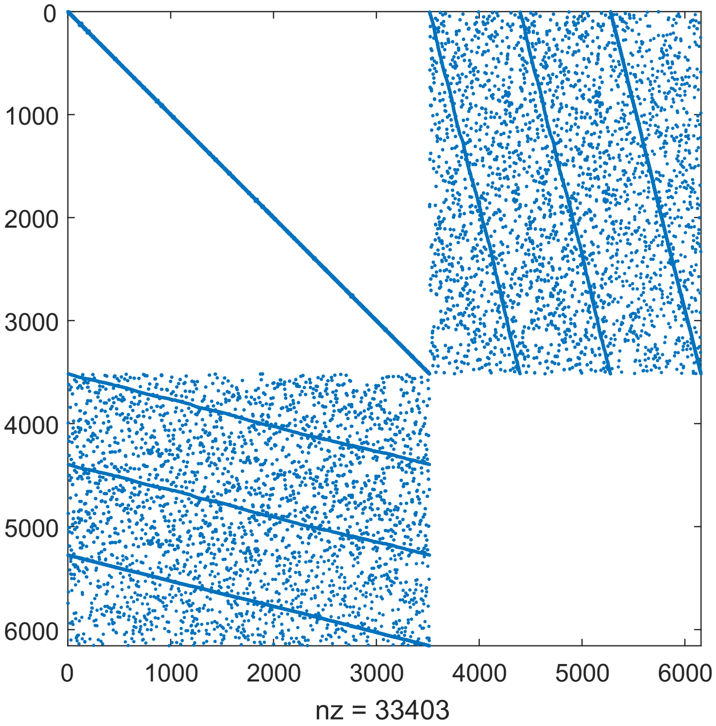


Figure 6.6: The sparsity pattern for the kth sub-problem system matrix after one reduction (6.21) and applying reordering for an 880-bus radial test system under the branch flow model. The upper-left block $\boldsymbol{Q}_K$ is block-diagonal. The blocks are small in size, so it appears almost diagonal. There are $nz = 33,403$ nonzero entries.

reduction while retaining sparsity by eliminating $\tilde{\boldsymbol{C}}_{dual}^{(k)}$ and $\tilde{\boldsymbol{r}}_{dual}^{(k)}$:

$$\boldsymbol{N}_k \begin{bmatrix} \tilde{\boldsymbol{C}}_{pri}^{(k)} & \tilde{\boldsymbol{r}}_{pri}^{(k)} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{H}_u^{(k)} & \boldsymbol{\omega}^{(k)} \end{bmatrix}, \tag{6.24}$$

$$\text{where } \boldsymbol{N}_k = \boldsymbol{H}_x^{(k)}[\boldsymbol{Q}_k]^{-1}[\boldsymbol{H}_x^{(k)}]^T \tag{6.25}$$

$$\boldsymbol{\omega}^{(k)} = [\boldsymbol{H}_x^{(k)}][\boldsymbol{Q}_k]^{-1}\boldsymbol{b}^{(k)} + \boldsymbol{r}_{pri}^{(k)}. \tag{6.26}$$

The sparsity pattern of matrix (6.25) is shown in Figure 6.7. We see that this matrix exhibits a sparse banded structure. Furthermore, $\boldsymbol{N}_k$ is symmetric and positive definite. Different techniques can be used to efficiently solve the remaining, reduced system, such as fully distributed matrix-

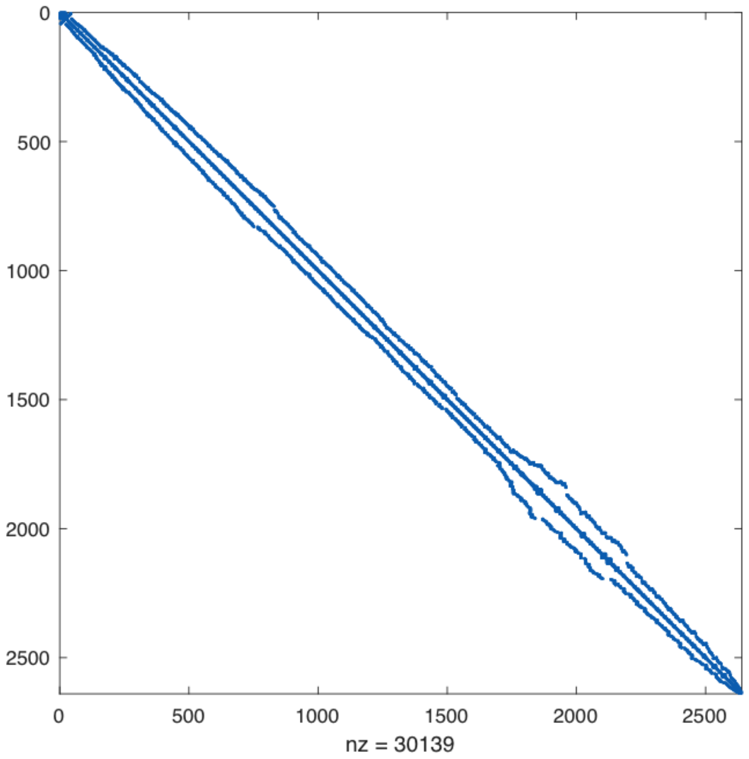


Figure 6.7: The sparsity pattern for the kth sub-problem system matrix after two reductions (6.25) and applying reordering for an 880-bus radial test system under the branch flow model. There are $nz = 30,139$ nonzero entries.

splitting solvers as in Chapter 5, sparse banded direct solvers [29], and red-black coloring [93].

Back-Substitution: To complete solving (6.20), back-substitution is used to calculate $\tilde{\boldsymbol{C}}^{(k)}_{dual}, \tilde{\boldsymbol{C}}^{(k)}_{cent}, \tilde{\boldsymbol{r}}^{(k)}_{dual},$ and $\tilde{\boldsymbol{r}}^{(k)}_{cent}$:

$$\tilde{\boldsymbol{C}}^{(k)}_{dual} = -[\boldsymbol{Q}_k]^{-1}[\boldsymbol{H}^{(k)}_x]^T\tilde{\boldsymbol{C}}^{(k)}_{pri} \tag{6.27a}$$

$$\tilde{\boldsymbol{r}}^{(k)}_{dual} = [\boldsymbol{Q}_k]^{-1}\boldsymbol{b} - [\boldsymbol{Q}_k]^{-1}[\boldsymbol{H}^{(k)}_x]^T\tilde{\boldsymbol{r}}^{(k)}_{pri} \tag{6.27b}$$

$$\tilde{\boldsymbol{C}}^{(k)}_{cent} = -[diag(\boldsymbol{g}^{(k)}(\boldsymbol{x}^{(k)}))]^{-1}diag(\boldsymbol{\lambda}^{(k)})D\boldsymbol{g}^{(k)}(\boldsymbol{x}^{(k)})\tilde{\boldsymbol{C}}^{(k)}_{dual}) \tag{6.27c}$$

$$\tilde{\boldsymbol{r}}^{(k)}_{cent} = -[diag(\boldsymbol{g}^{(k)}(\boldsymbol{x}^{(k)}))]^{-1}(-\boldsymbol{r}^{(k)}_{cent} + diag(\boldsymbol{\lambda}^{(k)})D\boldsymbol{g}^{(k)}(\boldsymbol{x}^{(k)})\tilde{\boldsymbol{r}}^{(k)}_{dual}) \tag{6.27d}$$

For the 880-bus example in Figures 6.3 - 6.7, the dimension of the per-contingency problem is reduced by a factor of 5 and the dimension of the overall SC-OPF problem is reduced by a factor of 16. In addition, the original large, sparse system (see Figure 6.3) has $183,040$ entries, whereas the reduced system (see Figure 6.7) has $30,139$ entries, showing that the reductions do not lead to dense filled-in matrices.

6.3 Numerical Results

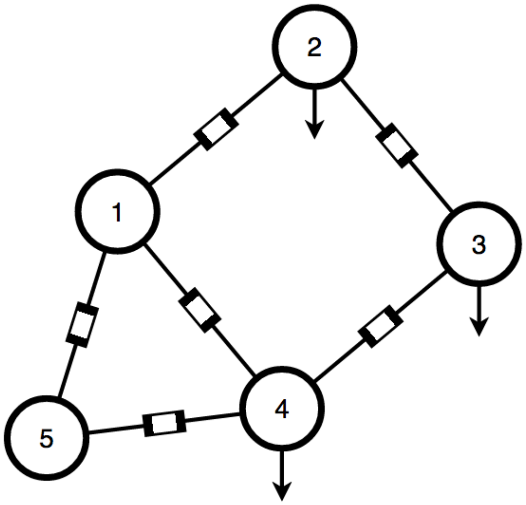
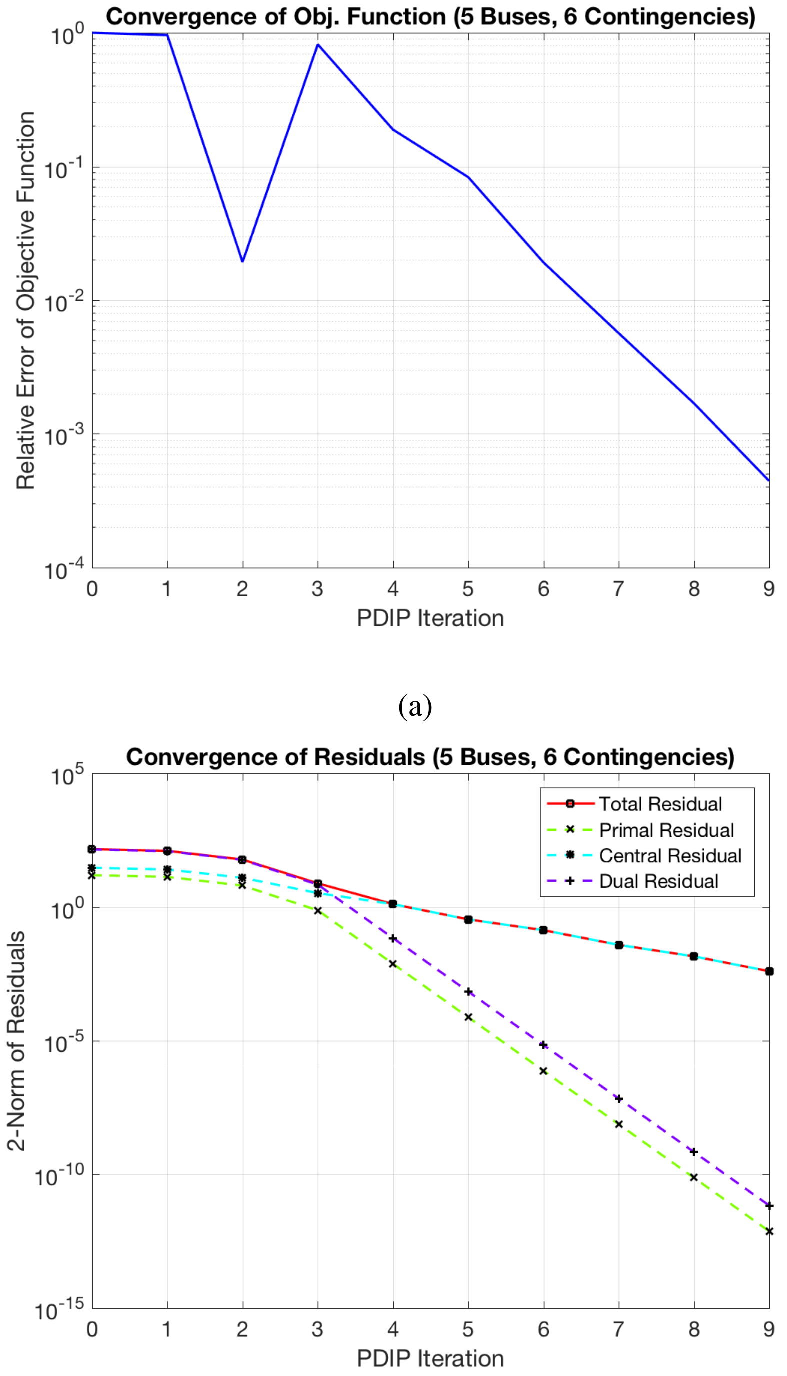


Figure 6.8: A 5-bus system based on [68, 112] for toy studies of DC SC-OPF. Visualization generated using [83].

We evaluate the performance of our algorithm by studying the convergence of the relative error of the objective function with respect to the true minimum. For the DC and BR SC-OPF formulations, the global minimum is calculated using a convex solver, CVX [27], [44]. To ensure the solution is feasible with respect to the constraints, the convergence of the residuals is also shown. For a small 5-bus system (see Figure 6.8), the results using 6 contingencies are shown in Figure 6.9. The contingencies considered are the failure of any one of the 6 transmission lines.



(a)

(b)

Figure 6.9: Performance in terms of the convergence of the relative error of the objective function and in terms of the residual is shown for the 5-bus system of Figure 6.8. The convergence of the relative error is not monotonically decreasing since the intermediate solutions do not satisfy the constraints.

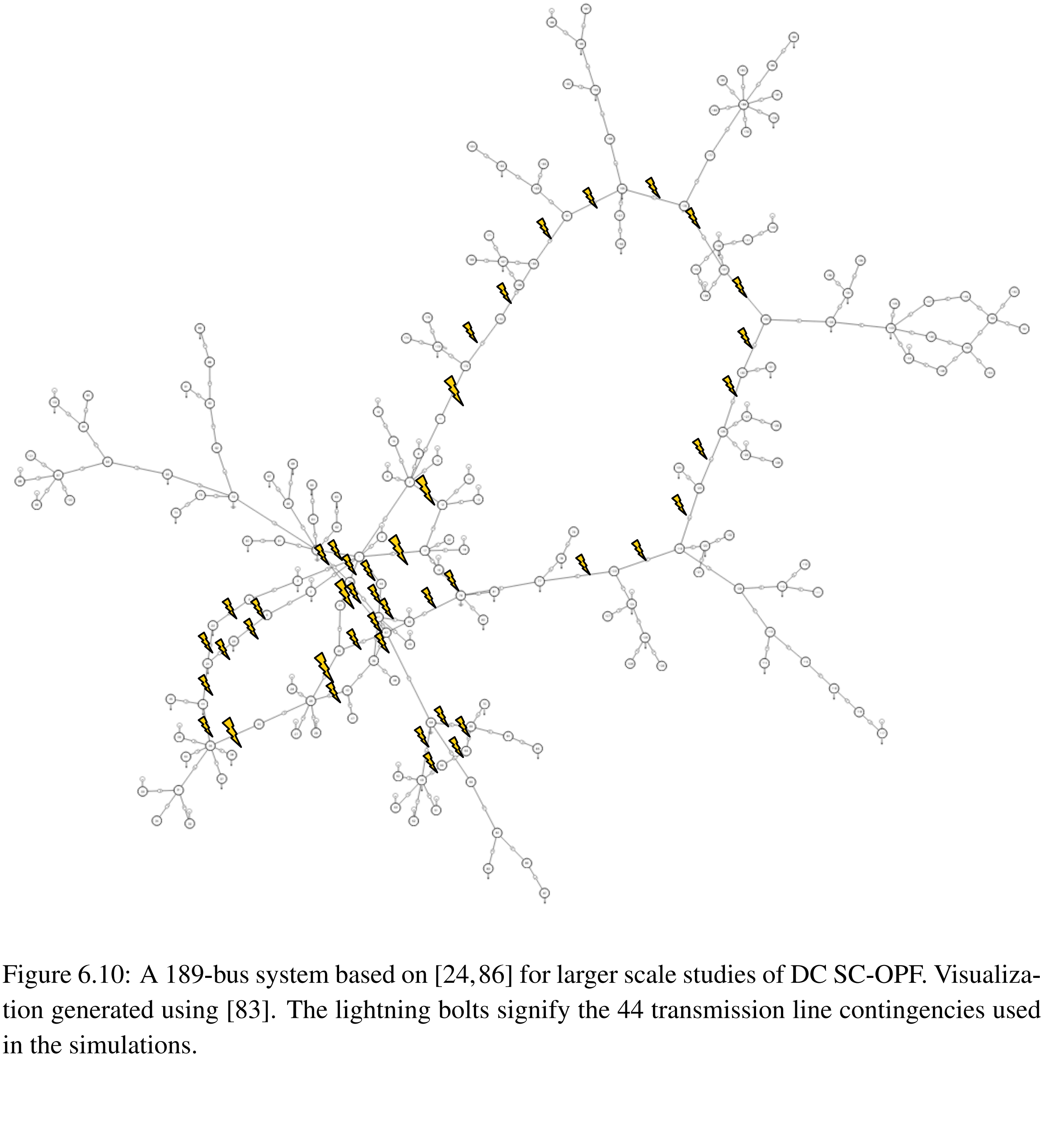


Figure 6.10: A 189-bus system based on [24, 86] for larger scale studies of DC SC-OPF. Visualization generated using [83]. The lightning bolts signify the 44 transmission line contingencies used in the simulations.

The convergence of the relative error and residuals is shown for an 189-bus system (see Figure 6.10) in Figure 6.11. Compared to the small test system that takes 9 iterations to converge, the simulation scales well on the larger 189-bus system with 44 contingencies, requiring around 30 iterations.

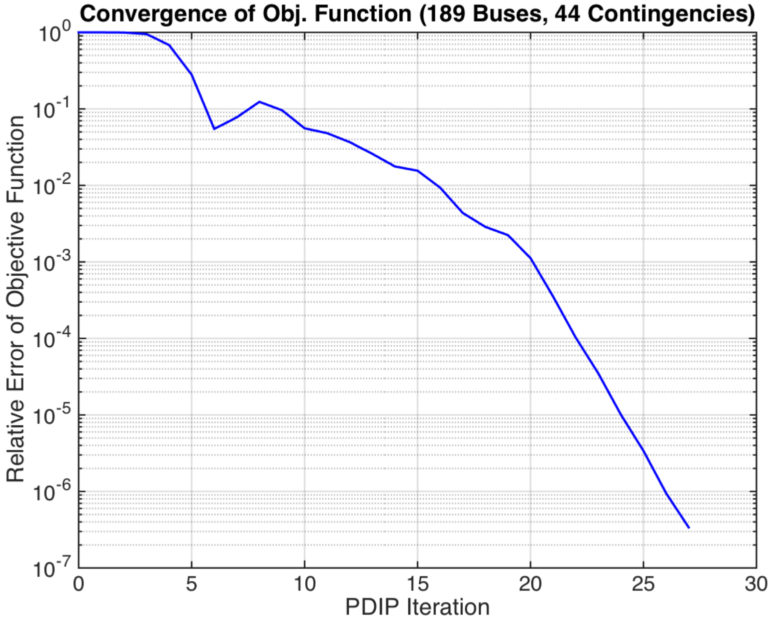
Table 6.1: Runtime Scaling (s) for DC SC-OPF: 189-bus system with 44 contingencies plus base case. Multiple processor runtime calculated without taking into account communication time.

Proposed Method: Layer 1	**Single Processor**	**45 Processors***	**Speedup**
Serial Part	24.83	24.83	1
Parallel Part	39.13	1.21	32.34
Total	63.96	26.04	2.46

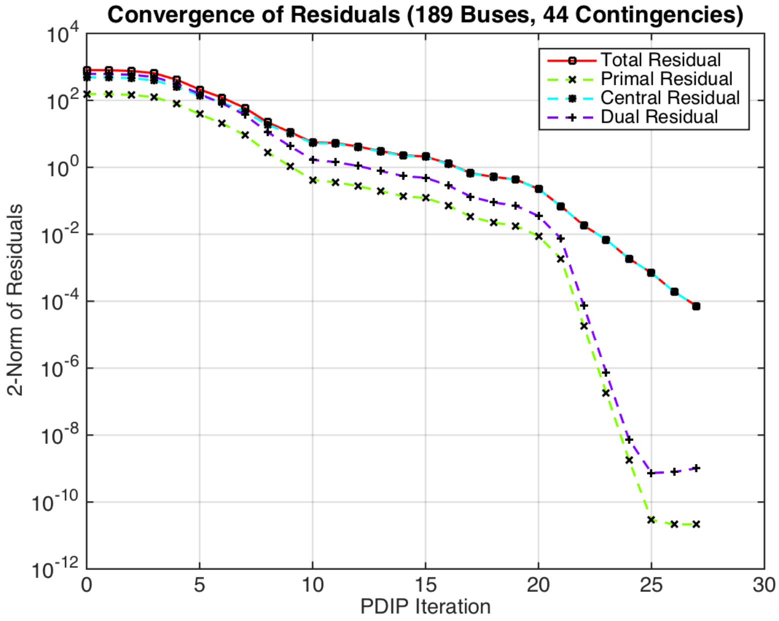
Table 6.2: Runtime Scaling (s) for BR SC-OPF: 880-bus system with 3 contingencies. Multiple processor runtime calculated without taking into account communication time.

Proposed Method: Layers 1&2	**Single Processor**	**64 Processors***	**Speedup**
Serial Part	40.10	40.10	1
Parallel Part	242.83	3.79	64
Total	282.93	43.89	6.4

In Table 6.1, the runtime scaling of the proposed method for computing the Newton step is demonstrated. The parallel part refers to Step 1 of Algorithm 6.1, and the serial part of the method refers to Steps 2 - 6 of Algorithm 6.1. The timing with 45 processors is identical to using a single



(a)

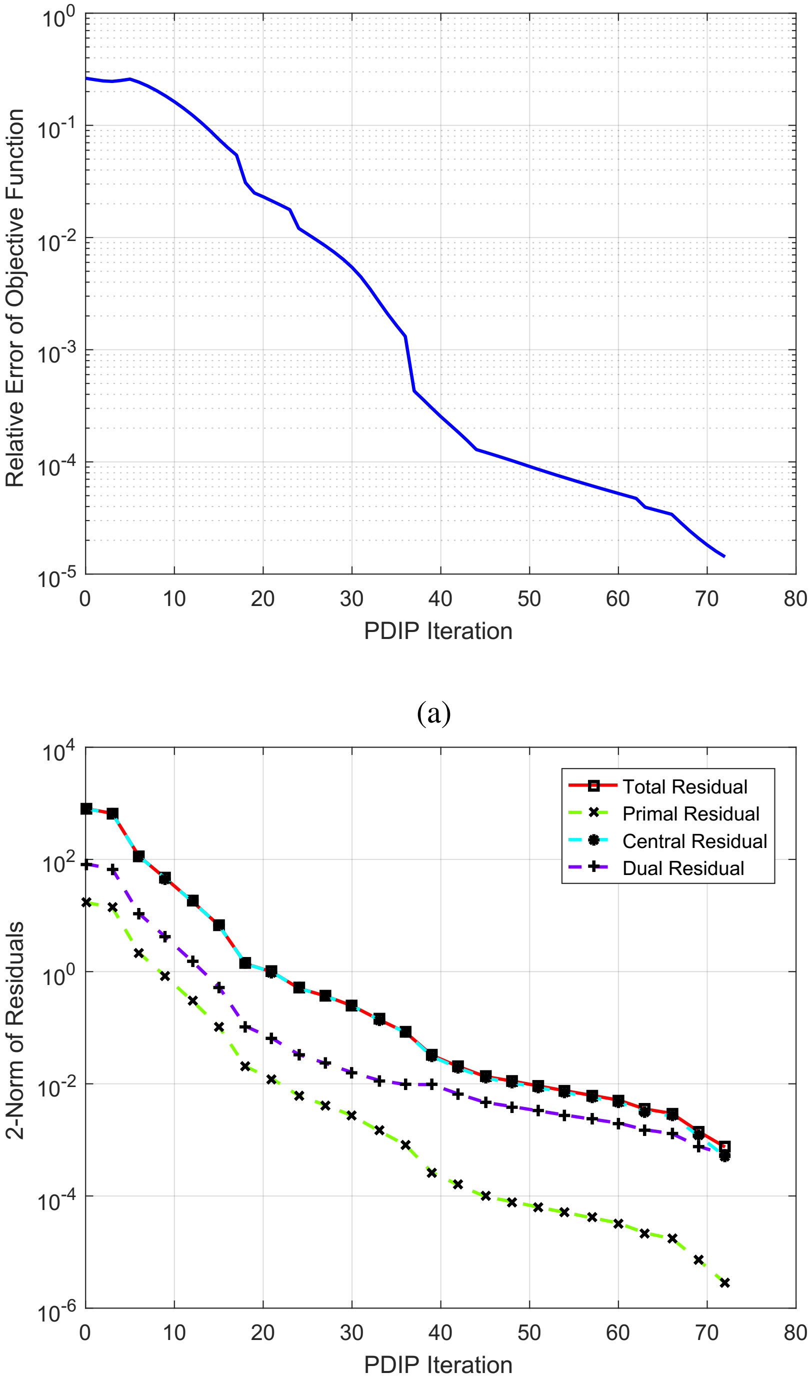


(b)

Figure 6.11: Performance in terms of the convergence of the relative error of the objective function and in terms of the residual is shown for the 189-bus system of Figure 6.10. The convergence of the relative error is not monotonically decreasing since the intermediate solutions do not satisfy the constraints.

processor for the serial part. For the parallel part, we use the maximum time over all contingencies. The timing for multiple processors is estimated excluding communication time, although we note that Step 1 of Algorithm 6.1 is perfectly parallelizable. The results must be synchronized before proceeding to the next step, so we use the maximum time for the parallel part rather than the average. The results in Table 6.1 are averaged across all of the PDIP iterations. We see that the portion of the Newton step algorithm that is parallelizable is roughly 60%, and speedups of roughly 32 and 2.5 are achieved for the parallelizable and total parts, respectively. The timing results in Table 6.1 consider only the Layer 1 parallelization.

Beyond the DC model, we also verify our performance on the BR model. Since the BR SC-OPF is a convex program, we can calculate the global minimum and study convergence of the relative error. In Figure 6.12, we show the convergence of the residual error and residuals, as well as timing results in Table 6.2, for a 880-bus system with 3 contingencies. The timing results in Table 6.2 include both Layer 1 and 2 parallelization. Comparing Tables 6.1 and 6.2, greater speedups can be achieved for the BR SC-OPF since the larger system has greater computational burden for the Newton step calculation, which is parallelized, and the BR SC-OPF allows for further parallelization by exploiting the radial structure of the network.



(a)

(b)

Figure 6.12: Performance in terms of the convergence of the relative error of the objective function and in terms of the residual is shown for an 880-bus radial system under the branch flow model.

6.4 Conclusion

In this chapter, we propose a highly scalable parallel primal-dual interior point (PDIP) method for security-constrained optimal power flow (SC-OPF). We examine three different formulations of SC-OPF: 1) the linearized DC model, 2) the convex relaxation of the BR (branch flow) model, and 3) the nonlinear AC model and the implications these have for parallelization. Two layers of parallelization for the PDIP search direction (Newton step) are designed across contingencies and across buses. The parallelization across contingencies does not require communication and is applicable to all three models. The parallelization across buses requires communication and is specialized for radial power networks. The convergence and timing results demonstrate favorable performance in terms of computational efficiency and accuracy.

Chapter 7

Aggregation of Distribution System-Level Devices

DISTRIBUTED ENERGY RESOURCES (DERS) represent energy production that is connected to the power grid via the distribution system. DERs present both challenges and opportunities for future grid operations. As more and more DERs are connected to the distribution system, there is a need to understand their impact on the operational state of the distribution system, as well as the bulk transmission system. Furthermore, there is an interest to utilize DERs to provide ancillary services and additional flexibility [65]. This includes peak load shedding [31] and voltage regulation [69]. In order to utilize DERs to achieve certain control objectives, their collective behavior needs to be understood.

We propose to quantify this in terms of the net load achievable at the substation over time. Given the feasible region for substation net load, grid operators can incorporate this information into a multi-period system-wide optimal power flow (OPF) to determine an optimal trajectory for the distribution system to follow (see Figure 7.1). Here, the sense of optimal is determined by the objective function of the OPF.

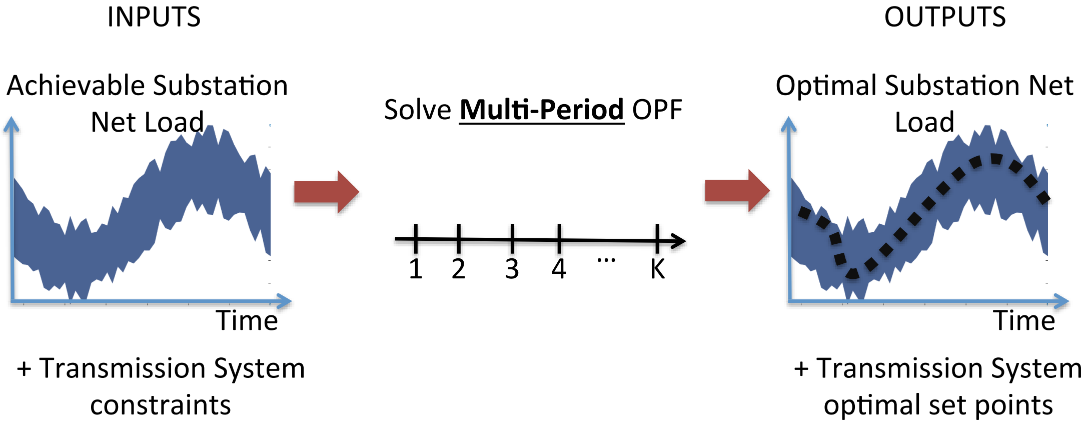

INPUTS
OUTPUTS
Achievable Substation Net Load
Time
+ Transmission System constraints
Solve Multi-Period OPF
1
2
3
4
...
K
Optimal Substation Net Load
Time
+ Transmission System optimal set points

Figure 7.1: The feasibility region is determined by an aggregation problem and utilized in a multi-period OPF for optimizing overall grid operations.

Aggregation of DERs has gained recent interest in the power systems research community. In [48], a set theoretic characterization bounds the aggregate flexibility of a population of heterogeneous thermostatically controlled loads (TCLs) between two different generalized battery models. In [110], the aggregate flexibility of a population of heterogeneous TCLs is formulated as the Minkowski sum of polytopes corresponding to the flexibility of each individual TCL, and linear programs are proposed to approximate the intensive calculation of the Minkowski sum. R-C thermodynamic models are utilized in [75] to model the flexibility in power consumption of a group of commercial building HVAC fans. In addition, a contract design based on economic MPC allows groups of buildings to provide fast regulation services to the grid. In [12], polytopic projections are used to produce a reduced set of constraints, depending only on deviations of tie-line flows, in order to determine and coordinate the flexibility that different TSOs can provide to one another. A two-level hierarchical scheme for incorporating distribution-level resources in a multiple time step OPF is proposed in [89]. The aggregate flexibility of the DERs is approximated by an ellipsoid at each time step using data-driven identification techniques.

The contributions of our approach detailed in this chapter include:

1. Our formulation considers the feasible region over multiple time periods in order to include dynamic and ramping/switching constraints.

2. Voltage constraints, which couple the different DER devices, are included to ensure that the power profiles of the DERs satisfy operational safety considerations for the distribution system.

3. The approach aggregates a general combination of devices whose constraints can be modeled linearly, rather than considering the aggregation of a single kind of device (*e.g.,* only HVACs).

7.1 Problem Statement

Given constraints on both devices and the distribution system bus voltages, the aim is to determine which values for the substation net load are achievable. We formulate this as a problem of polytope projection. Denote the real and reactive power injections at device i at bus j at time step k, $P_{ij}(k)$ and $Q_{ij}(k)$, respectively. The constraints can be organized into two categories: 1) device constraints and 2) network constraints.

- Device constraints include operational box constraints (*e.g.*, a device can draw power within a certain range),

$$\underline{P}_{ij} \leq P_{ij}(k) \leq \bar{P}_{ij}, \tag{7.1}$$

and physical dynamics (*e.g.*, the state of charge of a battery is related to the amount of power it draws or outputs at previous time steps). We assume linear models for the device dynamics. Let the state variable of device i at bus j at time step k be denoted $x_{ij}(k)$. For example, the dynamic state could be the state of charge for a battery or the thermal energy for a TCL. The value of the dynamic state variables depends on the power drawn by the device over time. We assume a linear dynamic model:

$$x_{ij}(k+1) = a_{ij}x_{ij}(k) + b_{ij}P_{ij}(k+1). \tag{7.2}$$

In Section 7.3.1, the coefficients a_{ij} and b_{ij} are identified for a battery and a TCL. In general, there are also operational box constraints on the dynamic variables, $\underline{x}_{ij} \leq x_{ij}(k) \leq \bar{x}_{ij}$ for each time step. Combining this with (7.2), the general form for the device dynamic constraints is:

$$\underline{x}_{ij} - a_{ij}^k x_{ij}(0) \leq \sum_{k'=1}^{k} a_{ij}^{k-k'} b_{ij} P_{ij}(k') \leq \bar{x}_{ij} - a_{ij}^k x_{ij}(0),\ \forall k = 1, \ldots, K. \tag{7.3}$$

- Network constraints introduce coupling between the power injections at different devices. This includes limits on the distribution system bus voltages and power balance at the substation. Linear models are used for both these constraints.

Power System Model. We consider a single-phase distribution feeder with a set $\mathcal{N}$ of $(N+1)$ nodes and a set of edges $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$. The zeroth node represents the substation bus that interfaces with the transmission system (*i.e.*, the secondary of the step-down distribution transformer). The set of all other buses in the feeder minus the substation bus is denoted $\mathcal{N}' = \mathcal{N} \setminus \{0\}$. The voltage (in rectangular coordinates) and complex power injected at bus l is $V_l = V_{re,l} + jV_{im,l}$ and $S_l = P_l + jQ_l$, respectively. Kirchoff's Current Law gives a nonlinear relation between the bus voltages and power injections. We utilize linear approximations for the voltage [46] and power balance constraints [47]:

$$|V|_l \approx |V_l^{nom}| + \sum_{l' \in \mathcal{N}'} R_{ll'} P_{l'} + X_{ll'} Q_{l'},\ \forall l \in \mathcal{N}' \tag{7.4}$$

$$P_0(k) \approx \sum_{l \in \mathcal{N}'} \phi_l P_l(k) \tag{7.5}$$

$$= \sum_{l \in \mathcal{N}'} \phi_l \sum_j P_{lj}(k), \tag{7.6}$$

where $|V|_l$ is the magnitude of the complex voltage V_l. The power injected at a bus is the sum of the power injections at all devices connected to that bus. The magnitude of the nominal voltage (*i.e.*, the point about which the linearization is taken) for bus l is denoted $|V_l^{nom}|$. The coefficients $\{R_{ll'}\}$ and $\{Q_{ll'}\}$ are entries of the real and imaginary parts of the inverse bus admittance matrix, $\boldsymbol{Y}^{-1} = \boldsymbol{R} + j\boldsymbol{Q}$. For details on the derivation and form of the $\{\phi_l\}$ coefficients, we refer the reader to [47].

The linear approximation in (7.4) is used to represent the constraint on the voltage magnitude at each bus in terms of the power injections:

$$|\underline{V}_l| - |V_l^{nom}| \leq \sum_{l' \in \mathcal{N}'} R_{ll'} P_{l'} + X_{ll'} Q_{l'} \leq |\bar{V}_l| - |V_l^{nom}|,\ \forall l.ci \tag{7.7}$$

Since we use linear models for the device and network constraints, the set of allowable power injections across all devices/buses and time steps can be described as a polyhedron:

$$S = \{P_{ij}(k), Q_{ij}(k), P_0(k), Q_0(k) | \text{ constraints } (7.1), (7.3), (7.6), \text{ and } (7.7) \text{ are satisfied}\}. \quad (7.8)$$

In a setting with millions of devices and multiple time steps, S is a very high-dimensional polyhedron with many constraints. It is preferable to provide the system operator a more manageable summary of the capability of the distribution feeder. Towards this aim, we propose to find the projection of S onto the set of power injections at the substation, $\{P_0(k)\}_{k=1}^{K}$, which is a K-dimensional polyhedron, S'. In practice, calculating the projection of high-dimensional polyhedrons is computationally intensive and not practical, especially for use in real-time operations where the result is needed on a rolling basis approximately every 5 minutes. Therefore, tractable approximation methods are needed.

7.2 Inner-Box Approximation

We propose a conservative inner-box approximation for the polyhedral projection S'. The box approximation is defined by a lower and upper bound at each time step:

$$\underline{P}_0(k) \leq P_0(k) \leq \overline{P}_0(k), \ \forall k. \quad (7.9)$$

Advantages of the box approximation are that decoupling time steps allows for easy interpretation in terms of a “feasibility tube” as shown in Figure 7.2.

Let the vector $\boldsymbol{x}$ be the concatenation of the power injections across all devices and time steps.

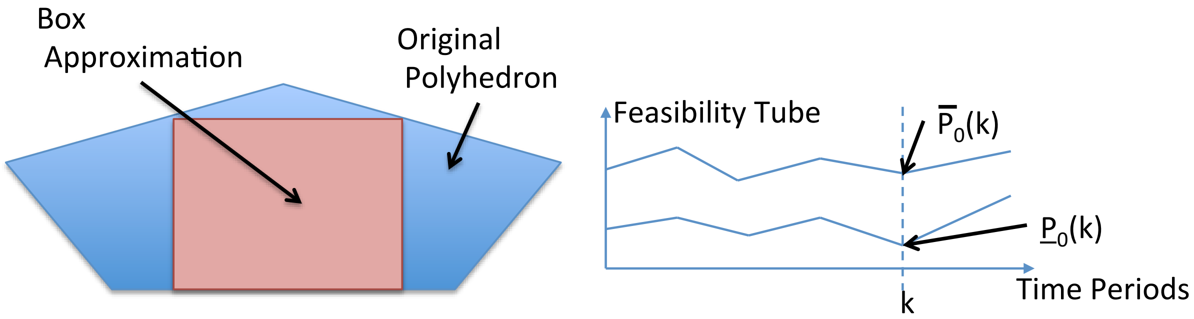


Figure 7.2: The lower and upper bounds of the inner-box approximation define a feasibility tube in time for the real power at the substation.

The inner-box for one time step $(K = 1)$ is found by solving the following optimization problem:

$$\max_{\boldsymbol{x},\tilde{\boldsymbol{x}},\underline{P}_0(1),y} \quad y \tag{7.10a}$$

$$\sum_{i=1}^{N} \phi_i \sum_j P_{ij}(1) = \underline{P}_0(1) \tag{7.10b}$$

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \tag{7.10c}$$

$$\sum_{i=1}^{n} \phi_i \sum_j \tilde{P}_{ij}(1) = \underline{P}_0(1) + y \tag{7.10d}$$

$$\boldsymbol{A}\tilde{\boldsymbol{x}} \leq \boldsymbol{b}, \tag{7.10e}$$

where the extra set of variables $\tilde{\boldsymbol{x}}$ is included since it is necessary to find *a* set of power injections satisfying the constraint at either vertex of the box (*i.e.*,$\underline{P}_0(1)$ and $\underline{P}_0(1) + y$) but not necessarily of the same value. Since the constraints (7.1), (7.3), (7.7) are linear, they can be succinctly written as $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$.

This is generalized to K time steps as,

$$\max \sum_{k=1}^{K} c_k \ln y(k) \tag{7.11a}$$

$$\text{over } \{\boldsymbol{x}_v\}_{v=1,\ldots,2^K}, \{\underline{P}_0(k), y(k)\}_{k=1}^{K} \tag{7.11b}$$

$$\text{s.t. for each vertex } v: \tag{7.11c}$$

$$\sum_{i=1}^{N} \phi_i \sum_j P_{ij}(k) = v(k),\ k = 1, \ldots, K \tag{7.11d}$$

$$\boldsymbol{A}\boldsymbol{x}_v \leq \boldsymbol{b}, \tag{7.11e}$$

where the weights $\{c_k\}$ are designed to favor more flexibility at certain time steps. The volume of the box is $\prod_{k=1}^{K} y(k)$. To obtain an equivalent convex problem, we maximize the log of the volume. There is a family of constraints and decision variables for each vertex of the box. There are 2^K vertices where the kth coordinate of a vertex $v(k) \in \{\underline{P}_0(k), \underline{P}_0(k)+y(k)\}$. In general, all vertices of the box must be included to find the maximum-volume inner-box approximation, however in practice, we find it is numerically sufficient to consider only the two extreme case vertices:

1. All lower bounds: $v(k) = \underline{P}_0(k),\ \forall k$
2. All upper bounds: $v(k) = \bar{P}_0(k),\ \forall k.$

This leads to large computational savings.

7.3 Numerical Results

7.3.1 Simulation Setup

We consider a 13-bus (*i.e.*, $N = 13$) radial network with 5 devices per bus:

1. thermostatically controlled load (TCL)
2. battery

3. fixed/inflexible load

4. flexible load

5. solar panel.

We provide details on the models for the device-specific constraints.

- *Thermostatically Controlled Load (TCL).* Let device j at bus i be a TCL. The TCL devices' power injections are constrained by box constraints of the form (7.1):

  $$\underline{P}_{TCL} \leq P_{ij}(k) \leq \overline{P}_{TLC}. \tag{7.12}$$

  In addition, the thermal energy of the TCL, $x_{ij}(k)$, is related to the power drawn as [110]:

  $$x_{ij}(k) = \alpha x_{ij}(k-1) + \delta(P_{ij}(k) - P_{ij}^{0,TCL}), \tag{7.13}$$

  where $\alpha = e^{-aT}, \delta = (1 - e^{-aT})/a$, T is the sampling period, and $a = 1/(R_{th}C_{th})$, with R_{th} and C_{th} being the thermal resistance and capacitance. The nominal power, $P_{ij}^{0,TCL}$, is defined as the amount of power needed to maintain the desired temperature, θ_r. The thermal energy is constrained as:

  $$\underline{x}_{TCL} \leq x_{ij}(k) \leq \overline{x}_{TCL}, \ \forall k. \tag{7.14}$$

- *Battery.* Let device j at bus i be a battery. The battery is subject to box power constraints of the form:

  $$\underline{P}_{battery} \leq P_{ij}(k) \leq \overline{P}_{battery}. \tag{7.15}$$

The state of charge of the battery, $e_{ij}(k)$, is related to the power drawn by the battery as:

$$e_{ij}(k) = e_{ij}(k-1) + \eta P_{ij}(k), \tag{7.16}$$

where η is the charging/discharging efficiency. This model is modified from [55] assuming symmetric charging and discharging efficiencies. The state of charge is constrained as:

$$\underline{e}_{battery} \leq e_{ij}(k) \leq \overline{e}_{battery}, \ \forall k. \tag{7.17}$$

- *Limited-Flexibility Loads.* Limited-flexibility loads only have the power box constraints in (7.1). The fixed/inflexible load is modeled as a limited-flexibility load by choosing $\underline{P}_{ij} := (1-\epsilon)P^*_{ij}$ and $\overline{P}_{ij} := (1+\epsilon)P^*_{ij}$ for small ϵ, (*e.g.*, $\epsilon = 0.01$ was used in the simulations). The flexible load is modeled similarly but with $\epsilon = 0.3$. The solar panel is modeled as an inflexible load with negative power injections.

One advantage of our method is that it allows for heterogeneous devices. In the simulations, we randomize the device parameters by $\pm 5\%$ the base values provided in Tables 7.1 and 7.2. The nominal voltage magnitude $|V^{nom}_l| = 1$. The numerical values for $\boldsymbol{R}$ and $\boldsymbol{X}$ are from [36].

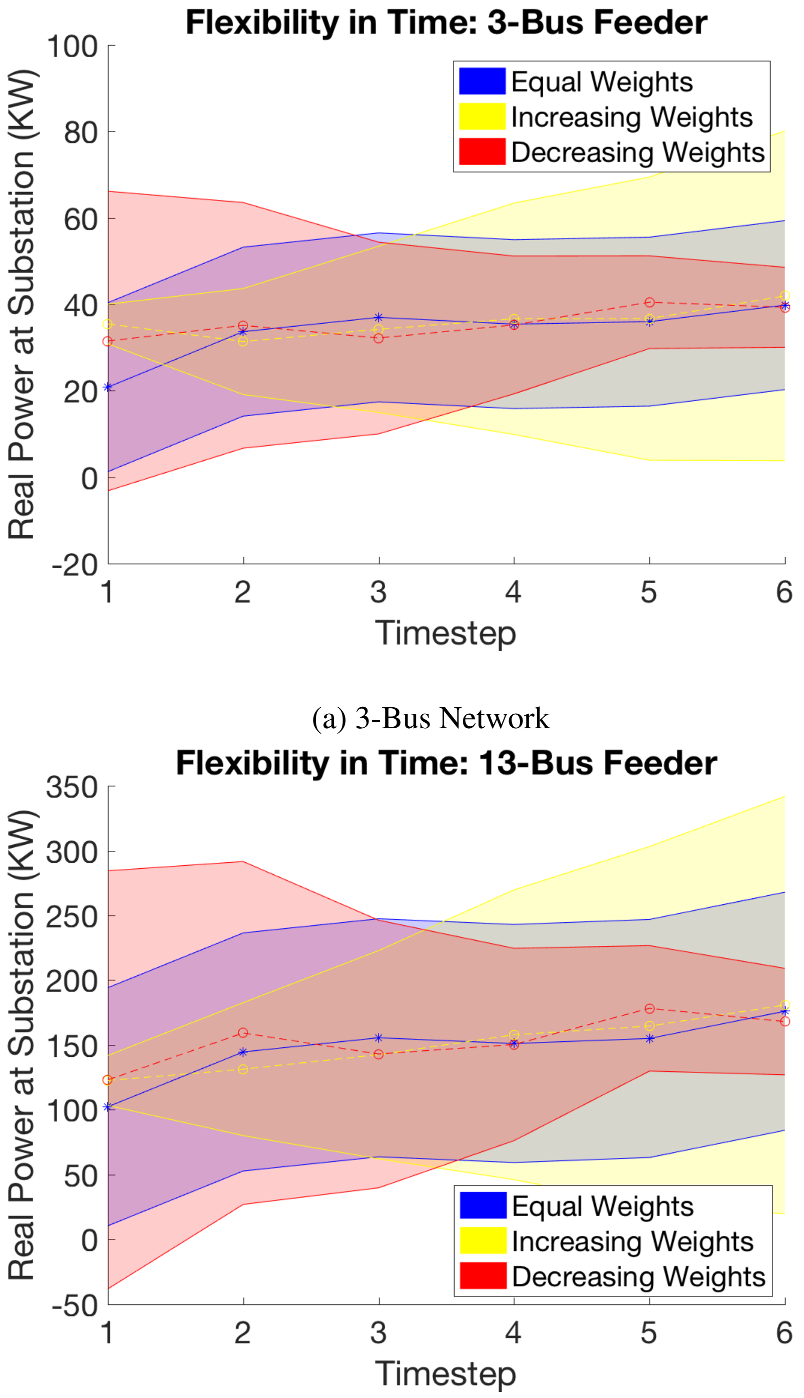
In Figure 7.3, the feasibility tubes for 3-bus and 13-bus networks (with 5 devices per bus as described in Section 7.3.1) are shown. We examine three different cases for the objective function weights $\{c_k\}$ in (7.11a): equal, increasing, and decreasing. As expected the flexibility tube width is fairly uniform across time steps when equal weights are used. The tube width increases in time with increasing weights and decreases in time with decreasing weights. The center of the tube is between 20-40 KW for the 3-bus network, whereas the center is shifted to 100-200 KW for the

Table 7.1: TCL Parameter Values

Parameter	**Value**
$\underline{P}_{TCL}$	0 kW
$\bar{P}_{TCL}$	37 kW
T_a, ambient temp.	32 C
Customer max temp.	24 C
Customer min temp.	20 C
R_{th}, thermal resistance [48]	2000 C/MW
C_{th}, thermal capacitance [48]	0.01, MWh/C
η_{TCL}, performance coeff. [48]	2.5
Desired temperature	22 C

Table 7.2: Battery Storage System Parameter Values

Parameter	**Value**
$\underline{P}_{battery}$	0 MW
$\underline{e}_{battery}$ [55]	32 MWh
$\overline{P}_{battery}$ [55]	8 MW
$\bar{e}_{battery}$	5 MWh
η_c, charging efficiency	1
η_d, discharging efficiency	1



(a) 3-Bus Network

(b) 13-Bus Network

Figure 7.3: Comparing flexibility in 3-bus and 13-bus networks using different objective function weight, $\{c_k\}$, scenarios.

13-bus network since there are more devices and thus more power utilized. To compare the amount of flexibility in different cases, the areas of the tubes are given in Table 7.3.

Table 7.3: Area of Feasibility Tube (kJ)

	Equal Weights	**Increasing Weights**	**Decreasing Weights**
3-Bus Network	81.03	87.00	83.28
13-Bus Network	361.44	366.06	370.77

The runtimes using CVX for different numbers of time steps, K, are compared for the 3-bus and 13-bus networks in Table 7.4. Two different performance metrics are also reported. The vector $\boldsymbol{r}_{ineq}$ is the residual of the inequality constraints, and its maximum entry should be non-positive. The vector $\boldsymbol{r}_{eq}$ is the residual of the equality constraints, and it should be close to zero (or machine precision).

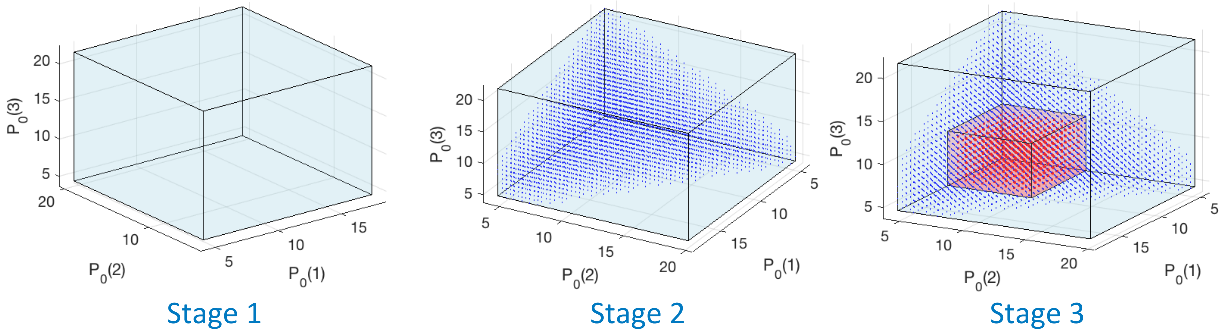
Table 7.4: Runtime Scaling

$\boldsymbol{K}$	$\boldsymbol{N_{bus}}$	**Runtime(s)**	$\boldsymbol{max(r_{ineq})}$	$\boldsymbol{\lvert r_{eq} \rvert}$
4	3	8.8200	0	3.8386e-14
	13	14.8200	0	2.2433e-12
6	3	10.2000	0	2.3623e-13
	13	20.0900	0	3.3915e-12
10	3	8.0900	0	3.0585e-12
	13	38.1500	6.0863e-14	1.0007e-12

7.4 Evaluating the Quality of the Inner-Box Approximation

To evaluate how conservative the inner-box approximation is, we use a three-stage approach:

1. Stage 1: Calculate the minimum-volume outer-box approximation.
2. Stage 2: Obtain a sample of points in the true polygon by sampling the from the outer box approximation and checking if the sampled points lie in the polygon.
3. Stage 3: Obtain the subset of points that are both in the true polygon and in the inner box.



Stage 1

Stage 2

Stage 3

Figure 7.4: In Stage 1, the outer-box approximation (blue box) to the polygon is calculated. In Stage 2, the outer-box is sampled and tested for feasibility to obtain a sample of points from the polygon (blue dots). In Stage 3, the ratio of number of points (red dots) in the inner-box approximation (red box) to the number of points in the polygon is calculated.

This process is shown schematically in Figure 7.4 for a $K = 3$ time step example. The basic idea is to see for a random sample of points from the true polygon what percentage falls within the inner-box approximation. If the sampling is uniform and the percentage captured by the inner box is high, we conclude the inner-box approximation performs well.

The first question is how to obtain a sample of points in the polygon. Let $\boldsymbol{P}_0 = [P_0(1) \dots P_0(K)]$ be a trial vector for the values of the substation real power injection. Recall $\boldsymbol{x}$ is the vector of all the real and reactive power injections across all devices and buses. To verify if $\boldsymbol{P}_0$ is in the polygon S' in (7.8), solving the following linear program checks for feasibility,

$$\min_{\boldsymbol{x}} \quad c \tag{7.18a}$$

$$s.t.\ \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \tag{7.18b}$$

$$\sum_{i=1}^{n} \phi_i \sum_{j} P_{ij}(k) = P_0(k),\ k = 1, \ldots, K \tag{7.18c}$$

where c is some constant. How should one select the trial values for $\boldsymbol{P}_0$? We calculate a minimum-volume outer box approximation, sample points from this box, and use those points as our trial points to test for feasibility in (7.5). The outer box is defined by upper and lower bounds at each time step, $\{\boldsymbol{P}_0 \in \mathbb{R}^K | \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}\}$, where

$$l(k) = \arg \min_{\{P_0(k)\}_{k=1}^K, \boldsymbol{x}} P_0(k) \tag{7.19a}$$

$$s.t.\ \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \tag{7.19b}$$

$$\sum_{i=1}^{n} \phi_i \sum_{j} P_{ij}(k) = P_0(k),\ k = 1, \ldots, K \tag{7.19c}$$

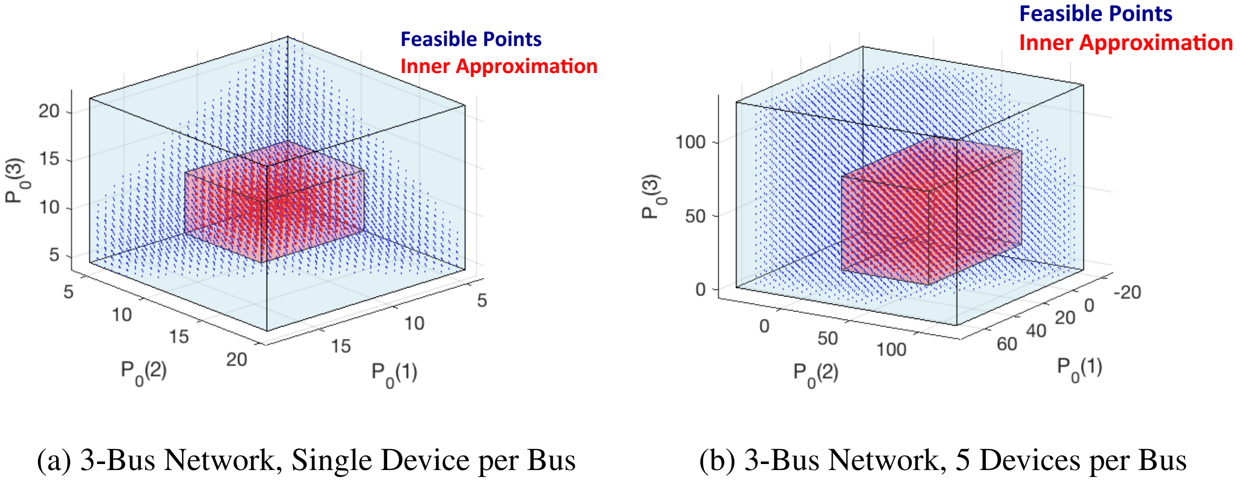
and $u(k)$ is obtained similarly but taking the $\arg\max$. Let N_{outer} be the number of points sampled from the outer box. Let N_{inner} be the number of points remaining from the sample that are contained in the inner box, and let N_{poly} be the number of points remaining from the sample that are contained in the polygon. The quantity

$$r = \frac{N_{inner}}{N_{poly}} \tag{7.20}$$

is a measure of how conservative the approximation is, and

$$r' = \frac{N_{poly}}{N_{outer}} \tag{7.21}$$

is a measure of how loose the outer-box approximation is. The evaluation results for the 3-bus network with a single device at each bus and with 5 devices per bus are shown in Figure 7.5 and Table 7.5. As expected, the quality of the inner-box approximation decreases as K increases.



(a) 3-Bus Network, Single Device per Bus

(b) 3-Bus Network, 5 Devices per Bus

Figure 7.5: The ratio of number of points in the inner approximation, N_{inner} (red dots), to the number of points in the polygon, N_{poly} (blue dots), is a measure of how conservative the approximation is.

Table 7.5: For a 3-bus network, the quality of the inner- and outer-box approximations is given by r and r', respectively. As more grid points (per dimension) are used, the error on these quantities decreases. In the multiple devices per bus case, 5 devices are used as described in Section 7.3.1.

Time Steps K	**# of Grid Points**	**Sampling Efficiency: r'**	**Inner-Box Quality: r**
Single Device			
3	25	0.3295	0.1864
4	12	0.1530	0.0567
4	15	0.1512	0.0627
Multiple Devices			
3	25	0.5386	0.2280

7.5 Conclusion

In this chapter, we developed a computationally tractable method for conservatively approximating the aggregate flexibility of distribution feeders. A main advantage of the inner-box approximation is that it allows for an easy interpretation in terms of the feasibility tube by the system operator. Future research directions include increasing the complexity of the modeling constraints (*e.g.*, asymmetric battery charging efficiencies) and the representation of the feasibility region. An important open question is how to best utilize the characterization of the distribution feeder capability within an online decision making framework for bulk grid operations.

Bibliography

[1] The Future of the Electric Grid: An Interdiscplinary MIT Study. https://mitei.mit.edu/system/files/Electric_Grid_Full_Report.pdf, 2011.

[2] Ali Abur and Antonio Gomez-Exposito. *Power System State Estimation: Theory and Implementation*. CRC Press, 2004.

[3] O. Alsac and B. Stott. Optimal Load Flow with Steady-State Security. *IEEE Transactions on Power Apparatus and Systems*, PAS-93(3):745–751, May 1974.

[4] S. Massoud Amin. Smart Grid: Overview, Issues and Opportunities. Advances and Challenges in Sensing, Modeling, Simulation, Optimization and Control. *Eur. J. Control*, 17(5-6):547–567, 2011.

[5] Arthur R. Bergen and Vijay Vittal. *Power Systems Analysis*. Prentice-Hall Series in Electrical and Computer Engineering. Prentice Hall, 2000.

[6] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[7] T. S. Bi, X. H. Qin, and Q. X. Yang. A Novel Hybrid State Estimator for Including Synchronized Phasor Measurements. *Electric Power Systems Research*, 78(8):1343–1352, August 2008.

[8] P. N. Biskas and A. G. Bakirtzis. A Decentralized Solution to the Security Constrained DC-OPF Problem of Multi-Area Power Systems. In *2005 IEEE Russia Power Tech*, pages 1–7, June 2005.

[9] Brett Bode, Michelle Butler, Thom Dunning, William Gropp, Torsten Hoe-fler, Wen mei Hwu, and William Kramer (alphabetical). *The Blue Waters Super-System for Super-Science.* Contemporary HPC Architectures. Edited by Jeffery Vetter. Chapman and Hall / CRC, 2013.

[10] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.

[11] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[12] M. A. Bucher, S. Chatzivasileiadis, and G. Andersson. Managing Flexibility in Multi-Area Power Systems. *IEEE Transactions on Power Systems*, 31(2):1218–1226, March 2016.

[13] Mary B. Cain, Richard P. O'Neill, and Anya Castillo. History of Optimal Power Flow and Formulations. `http://www.ferc.gov/industries/electric/indus-act/market-planning/opf-papers/acopf-1-history-formulation-testing.pdf`, 2012.

[14] F. Capitanescu, J.M. Ramos, P. Panciatici, D. Kirschen, A.M. Marcolini, L. Platbrood, and L. Wehenkel. State-of-the-Art, Challenges, and Future Trends in Security Constrained Optimal Power Flow. *Electric Power Systems Research*, 81(8):1731–41, 2011.

[15] Florin Capitanescu, Damien Ernst, Mevludin Glavic, and Louis Wehenkel. Interior-Point Based Algorithms for the Solution of Optimal Power Flow Problems. *Electric Power Systems Research*, pages 508–517, April 2007.

[16] Florin Capitanescu, Mevludin Glavic, Damien Ernst, and Louis Wehenkel. Applications of Security-Constrained Optimal Power Flows. In *Modern Electric Power Systems Symposium (MEPS06)*, Wroclaw, Poland, Sep 2006.

[17] E. Caro, A.J. Conejo, and R. Minguez. Decentralized State Estimation and Bad Measurement Identification: An Efficient Lagrangian Relaxation Approach. *Power Systems, IEEE Transactions on*, 26(4):2500–2508, Nov 2011.

[18] J. Carpentier. Contribution à l'Étude du Dispatching Économique. *Bulletin de la Société Française des Électriciens*, 3:431–447, 1962.

[19] J.B. Carvalho and F.M. Barbosa. Distributed Processing in Power System State Estimation. In *10th Mediterranean Electrotechnical Conference (MELECON)*, volume 3, pages 1128–1131, May 2000.

[20] S. Chakrabarti, M. Kraning, E. Chu, R. Baldick, and S. Boyd. Security Constrained Optimal Power Flow via Proximal Message Passing. In *2014 Clemson University Power Systems Conference*, pages 1–8, March 2014.

[21] S. Chakrabarti, E. Kyriakides, G. Ledwich, and A. Ghosh. Inclusion of PMU Current Phasor Measurements in a Power System State Estimator. *IET Generation, Transmission & Distribution*, 4:1104–1115(11), October 2010.

[22] A. Chakrabortty and P. Khargoneka. Introduction to Wide-Area Control of Power Systems. In *American Control Conference*, pages 6773–6785, 2013.

[23] Naiyuan Chiang and Andreas Grothey. Solving Security Constrained Optimal Power Flow Problems by a Structure Exploiting Interior Point Method. *Optimization and Engineering*, 16(1):49–71, 2015.

[24] Carleton Coffrin, Dan Gordon, and Paul Scott. NESTA, The NICTA Energy System Test Case Archive. *CoRR*, abs/1411.0359, 2014.

[25] AJ. Conejo, S. De La Torre, and M. Canas. An Optimization Approach to Multiarea State Estimation. *IEEE Trans. Power Systems*, 22(1):213–221, Feb 2007.

[26] Richard W. Cottle, J. Pang, and R. Stone. *Linear Complementarity Problem.* Academic Press, 1992.

[27] Inc. CVX Research. CVX: Matlab Software for Disciplined Convex Programming, Version 2.0. `http://cvxr.com/cvx`, August 2012.

[28] Emiliano Dall'Anese, Hao Zhu, and Georgios B. Giannakis. Distributed Optimal Power Flow for Smart Microgrids. *IEEE Trans. Smart Grid*, 4(3):1464–1475, 2013.

[29] Timothy A. Davis. *Direct Methods for Sparse Linear Systems* . Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.

[30] J. Demmel. *Applied Numerical Linear Algebra.* SIAM, 1997.

[31] A. D. Dominguez-Garcia and C. N. Hadjicostis. Coordination and Control of Distributed Energy Resources for Provision of Ancillary Services. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 537–542, Oct 2010.

[32] R. Ebrahimian and R. Baldick. State Estimation Distributed Processing. *IEEE Trans. Power Systems*, 15(4):1240–1246, 2000.

[33] T. Erseghe. Distributed Optimal Power Flow Using ADMM. *IEEE Trans. Power Systems*, 29(5):2370–2380, Sept 2014.

[34] Antonio Gomez Exposito, Ali Abur, Antonio de la Villa Jaen, and Catalina Gomez-Quiles. A Multilevel State Estimation Paradigm for Smart Grids. *Proceedings of the IEEE*, 99(6):952–976, 2011.

[35] D.M. Falcao, F.F. Wu, and L. Murphy. Parallel and Distributed State Estimation. *Power Systems, IEEE Transactions on*, 10(2):724–730, May 1995.

[36] M. Farivar, R. Neal, C. Clarke, and S. Low. Optimal Inverter VAR Control in Distribution Systems with High PV Penetration. In *2012 IEEE Power and Energy Society General Meeting*, pages 1–7, July 2012.

[37] Masoud Farivar, Christopher R Clarke, Steven H Low, and K Mani Chandy. Inverter Var Control for Distribution Systems with Renewables. In *IEEE Intl. Conf. on SmartGridComm*, pages 457–462, 2011.

[38] Masoud Farivar and Steven H Low. Branch Flow Model: Relaxations and Convexification Part I. *Power Systems, IEEE Trans.*, 28(3):2554–2564, 2013.

[39] S. Fliscounakis, P. Panciatici, F. Capitanescu, and L. Wehenkel. Contingency Ranking with respect to Overloads in Very Large Power Systems taking into Account Uncertainty, Preventive and Corrective Actions. *Power Systems, IEEE Transactions on*, 28(4):4909–4917, 2013.

[40] Georgios B. Giannakis, Vassilis Kekatos, Nikolaos Gatsis, Seung-Jun Kim, Hao Zhu, and Bruce F. Wollenberg. Monitoring and Optimization for Power Grids: A Signal Processing Perspective. *IEEE Signal Process. Mag.*, 30(5):107–128, 2013.

[41] M. Gol and A. Abur. A Hybrid State Estimator For Systems With Limited Number of PMUs. *Power Systems, IEEE Transactions on*, 30(3):1511–1517, May 2015.

[42] T. Gomez, I. J. Perez-Arriaga, J. Lumbreras, and V. M. Parra. A Security-Constrained Decomposition Approach to Optimal Reactive Power Planning. *IEEE Transactions on Power Systems*, 6(3):1069–1076, Aug 1991.

[43] A. Gomez-Exposito, A. Abur, A. de la Villa Jaen, C. Gomez-Quiles, P. Rousseaux, and T. Van Cutsem. A Taxonomy of Multi-Area State Estimation Methods. *Electric Power Systems Research*, 81:1060–1069, 2011.

[44] M. Grant and S. Boyd. Graph Implementations for Nonsmooth Convex Programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

[45] William Gropp, Ewing Lusk, and Anthony Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface,* 3rd edition. MIT Press, Nov. 2014.

[46] S. S. Guggilam, E. Dall'Anese, Y. C. Chen, S. V. Dhople, and G. B. Giannakis. Scalable Optimization Methods for Distribution Networks With High PV Integration. *IEEE Transactions on Smart Grid*, 7(4):2061–2070, July 2016.

[47] S. S. Guggilam, C. Zhao, E. Dall'Anese, Y. C. Chen, and S. V. Dhople. Primary Frequency Response with Aggregated DERs. In *2017 American Control Conference (ACC)*, May 2017.

[48] H. Hao, B. M. Sanandaji, K. Poolla, and T. L. Vincent. Aggregate Flexibility of Thermostatically Controlled Loads. *IEEE Transactions on Power Systems*, 30(1):189–198, Jan 2015.

[49] Michael T. Heath. *Scientific Computing: An Introductory Survey*. McGraw-Hill Higher Education, 2nd edition, 1996.

[50] João Pedro Hespanha. `grPartition` a MATLAB function for graph partitioning. Available at `http://www.ece.ucsb.edu/~hespanha`, October 2004.

[51] Ying Hu, A Kuh, Tao Yang, and A Kavcic. A Belief Propagation Based Power Distribution System State Estimator. *IEEE Computational Intelligence Magazine*, 6(3):36–46, Aug 2011.

[52] Yih-Fang Huang, Stefan Werner, Jing Huang, Neelabh Kashyap, and Vijay Gupta. State Estimation in Electric Power Grids: Meeting New Challenges Presented by the Requirements of the Future Grid. *IEEE Signal Process. Mag.*, 29(5):33–43, 2012.

[53] G. Hug-Glanzmann and G. Andersson. Decentralized Optimal Power Flow Control for Overlapping Areas in Power Systems. *IEEE Trans. Power Systems*, 24(1):327–336, Feb 2009.

[54] R. Jabr, A. Coonick, and B. Cory. A Primal-Dual Interior Point Method for Optimal Power Flow Dispatching. *IEEE Trans. Power Systems*, 17(3):654–662, August 2002.

[55] R. A. Jabr, S. Karaki, and J. A. Korbane. Robust Multi-Period OPF With Storage and Renewables. *IEEE Transactions on Power Systems*, 30(5):2790–2799, Sept 2015.

[56] Weiqing Jiang, V. Vittal, and G.T. Heydt. A Distributed State Estimator Utilizing Synchronized Phasor Measurements. *Power Systems, IEEE Transactions on*, 22(2):563–571, May 2007.

[57] Weiqing Jiang, V. Vittal, and G.T. Heydt. Diakoptic State Estimation Using Phasor Measurement Units. *Power Systems, IEEE Transactions on*, 23(4):1580–1589, Nov 2008.

[58] S. Julier, J. Uhlmann, and H.F. Durrant-Whyte. A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators. *Automatic Control, IEEE Trans. on*, 45(3):477–482, Mar 2000.

[59] V. Kekatos and G.B. Giannakis. Distributed Robust Power System State Estimation. *IEEE Trans. Power Systems*, 28(2):1617–1626, May 2013.

[60] G. Korres, A. Tzavellas, and E. Galinas. A Distributed Implementation of Multi-Area Power System State Estimation on a Cluster of Computers. *Electric Power Systems Research*, 102:20–32, Sept 2013.

[61] George N. Korres. A Distributed Multiarea State Estimation. *IEEE Trans. Power Systems*, 26(1):73–84, 2011.

[62] G.N. Korres and N.M. Manousakis. State Estimation and Observability Analysis for Phasor Measurement Unit Measured Systems. *IET Generation, Transmission & Distribution*, 6(9):902–913, September 2012.

[63] William Kramer, Michelle Butler, Gregory Bauer, Kalyana Chadalavada, and Celso Medes. *Blue Waters Parallel I/O Storage Sub-system,* High Performance Parallel I/O, Prabhat and Quincey Koziol editors. CRC Publications, Taylor and Francis Group, 2015.

[64] M. Kraning, E. Chu, J. Lavaei, and S. Boyd. Dynamic Network Energy Management via Proximal Message Passing. *Foundations and Trends in Optimization*, 1(2), 2014.

[65] B. Kroposki, C. Pink, R. DeBlasio, H. Thomas, M. Simoes, and P. K. Sen. Benefits of Power Electronic Interfaces for Distributed Energy Systems. In *2006 IEEE Power Engineering Society General Meeting*, pages 8 pp.–, 2006.

[66] A.Y.S. Lam, Baosen Zhang, and D.N. Tse. Distributed Algorithms for Optimal Power Flow Problem. In *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, pages 430–437, Dec 2012.

[67] A.M. Leite da Silva, M.B. Do Coutto Filho, and J.F. de Queiroz. State Forecasting in Electric Power Systems. *Generation, Transmission and Distribution, IEE Proceedings C*, 130(5):237–244, 1983.

[68] F. Li and R. Bo. Small Test Systems for Power System Economic Studies. In *IEEE PES General Meeting*, pages 1–4, July 2010.

[69] H. Li, F. Li, Y. Xu, D. T. Rizy, and J. D. Kueck. Interaction of Multiple Distributed Energy Resources in Voltage Regulation. In *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pages 1–6, July 2008.

[70] Xiao Li and A Scaglione. Advances in Decentralized State Estimation for Power Systems. In *IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pages 428–431, Dec 2013.

[71] C.-H. Lin. A Distributed State Estimator for Electric Power Systems. *Power Systems, IEEE Transactions on*, 7(2):551–557, May 1992.

[72] S.-Y. Lin and C.-H. Lin. An Implementable Distributed State Estimator and Distributed Bad Data Processing Schemes for Electric Power Systems. *Power Systems, IEEE Transactions on*, 9(3):1277–1284, Aug 1994.

[73] S.H. Low. Convex Relaxation of Optimal Power Flow: A Tutorial. In *Bulk Power System Dynamics and Control - IX Optimization, Security and Control of the Emerging Power Grid (IREP), 2013 IREP Symposium*, pages 1–15, Aug 2013.

[74] S. Magnusson, P. C. Weeraddana, and C. Fischione. A Distributed Approach for the Optimal Power-Flow Problem Based on ADMM and Sequential Convex Approximations. *IEEE Transactions on Control of Network Systems*, 2(3):238–253, Sept 2015.

[75] W. Mai and C. Y. Chung. Economic MPC of Aggregating Commercial Buildings for Providing Flexible Power Reserve. *IEEE Transactions on Power Systems*, 30(5):2685–2694, Sept 2015.

[76] Pierluigi Mancarella. MES (Multi-Energy Systems): An Overview of Concepts and Evaluation Models. *Energy*, 65(C):1–17, 2014.

[77] A. Minot and N. Li. A Fully Distributed State Estimation Using Matrix Splitting Methods. In *2015 American Control Conference (ACC)*, pages 2488–2493, July 2015.

[78] A. Minot, Y. M. Lu, and N. Li. A Distributed Gauss-Newton Method for Power System State Estimation. *IEEE Transactions on Power Systems*, 31(5):3804–3815, Sept 2016.

[79] A. Minot, Y. M. Lu, and N. Li. A Parallel Primal-Dual Interior-Point Method for DC Optimal Power Flow. In *2016 Power Systems Computation Conference (PSCC)*, pages 1–7, June 2016.

[80] A. Minot, H. Sun, and D. Nikovski. A Fully Distributed Filtering Scheme for Load-Based Dynamic State Estimation. In *2016 IEEE Power and Energy Society General Meeting (PESGM)*, pages 1–5, July 2016.

[81] A. Monticelli, M. V. F. Pereira, and S. Granville. Security-Constrained Optimal Power Flow with Post-Contingency Corrective Rescheduling. *IEEE Transactions on Power Systems*, 2(1):175–180, Feb 1987.

[82] Angelia Nedic. *Distributed Optimization,* Encyclopedia of Systems and Control, John Baillieul and Tariq Samad editors. Springer London, 2013.

[83] Monash University NICTA. Steady-State AC Network Visualization in the Browser. Available at `http://immersive.erc.monash.edu.au/stac/`, March 2017.

[84] National Academies of Sciences Engineering and Medicine. *Analytic Research Foundations for the Next-Generation Electric Grid*. The National Academies Press, Washington, DC, 2016.

[85] J M Ortega. *Matrix Theory: A Second Course*. Plenum Press, New York, NY, USA, 1986.

[86] Durham University Patrick Mc Nabb. The University of Edinburgh Power Systems Test Case Archive. Available at `http://www.maths.ed.ac.uk/optenergy/NetworkData/`, March 2017.

[87] D. Phan and J. Kalagnanam. Some Efficient Optimization Methods for Solving the Security-Constrained Optimal Power Flow Problem. *IEEE Transactions on Power Systems*, 29(2):863–872, March 2014.

[88] E. Polymenas and S. Meliopoulos. Aggregate Modeling of Distribution Systems for Multi-Period OPF. In *Power Systems Computation Conference (PSCC)*, 2016.

[89] E. Polymeneas and S. Meliopoulos. Aggregate Modeling of Distribution Systems for Multi-Period OPF. In *2016 Power Systems Computation Conference (PSCC)*, pages 1–8, June 2016.

[90] Xiangyun Qing, Hamid Reza Karimi, Yugang Niu, and Xingyu Wang. Decentralized Unscented Kalman Filter Based on a Consensus Algorithm for Multi-Area Dynamic State Estimation in Power Systems. *International Journal of Electrical Power & Energy Systems*, 65:26 – 33, 2015.

[91] Wei Qiu, A. J. Flueck, and Feng Tu. A New Parallel Algorithm for Security Constrained Optimal Power Flow with a Nonlinear Interior Point Method. In *IEEE Power Engineering Society General Meeting, 2005*, volume 1, pages 447–453, June 2005.

[92] G. Rigatos and P. Siano. Distributed State Estimation for Condition Monitoring of Nonlinear Electric Power Systems. In *Industrial Electronics (ISIE), 2011 IEEE Intl. Symp. on*, pages 1703–1708, June 2011.

[93] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.

[94] F. C. Schweppe and J. Wildes. Power System Static-State Estimation, Part I, II, & III. *IEEE Transactions on Power Apparatus and Systems*, PAS-89(1):120–135, Jan 1970.

[95] M. Shahidehpour and Y. Wang. *Communication and Control in Electric Power Systems: Applications of Parallel and Distributed Processing*. Wiley IEEE Press, 2003.

[96] A.K. Singh and B.C. Pal. Decentralized Dynamic State Estimation in Power Systems Using Unscented Transformation. *Power Systems, IEEE Trans. on*, 29(2):794–804, March 2014.

[97] Xin Tai, Zhiyun Lin, Minyue Fu, and Yuanzhang Sun. A New Distributed State Estimation Technique for Power Networks. In *American Control Conference*, pages 3338–3343, 2013.

[98] Ye Tian, Zhe Chen, and Fuliang Yin. Distributed IMM-Unscented Kalman Filter for Speaker Tracking in Microphone Array Networks. *Audio, Speech, and Language Processing, IEEE/ACM Trans. on*, 23(10):1637–1647, Oct 2015.

[99] G. Valverde and V. Terzija. Unscented Kalman Filter for Power System Dynamic State Estimation. *IET Generation, Transmission & Distribution*, 5:29–37(8), January 2011.

[100] Th. Van Cutsem, J.L. Horward, and M. Ribbens-Pavella. A Two-Level Static State Estimator for Electric Power Systems. *Power Apparatus and Systems, IEEE Transactions on*, PAS-100(8):3722–3732, Aug 1981.

[101] R. Varga. *Gershgorin and His Circles*. Springer Series in Computational Mathematics, 2004.

[102] Richard S. Varga. *Matrix Iterative Analysis*. Springer Series in Computational Mathematics. Springer Verlag, Berlin, Heidelberg, Paris, 2000.

[103] L. S. Vargas, V. H. Quintana, and A. Vannelli. A Tutorial Description of an Interior Point Method and Its Applications to Security-Constrained Economic Dispatch. *IEEE Transactions on Power Systems*, 8(3):1315–1324, Aug 1993.

[104] T. Vercauteren and Xiaodong Wang. Decentralized Sigma-Point Information Filters for Target Tracking in Collaborative Sensor Networks. *Signal Processing, IEEE Trans. on*, 53(8):2997–3009, Aug 2005.

[105] Shaobu Wang, Wenzhong Gao, and A.P.S. Meliopoulos. An Alternative Method for Power System Dynamic State Estimation Based on Unscented Transform. *Power Systems, IEEE Trans. on*, 27(2):942–950, May 2012.

[106] Ermin Wei, Asuman E. Ozdaglar, and Ali Jadbabaie. A Distributed Newton Method for Network Utility Maximization. *Conference on Decision and Control*, pages 1816–1821, 2010.

[107] S. Wright. *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics, 1997.

[108] Le Xie, Dae-Hyun Choi, Soummya Kar, and H. Vincent Poor. Fully Distributed State Estimation for Wide-Area Monitoring Systems. *IEEE Trans. Smart Grid*, 3(3):1154–1169, 2012.

[109] L. Zhao and A. Abur. Multiarea State Estimation Using Synchronized Phasor Measurements. *IEEE Trans. Power Systems*, 20(2):611–617, 2005.

[110] L. Zhao and W. Zhang. A Geometric Approach to Virtual Battery Modeling of Thermostatically Controlled Loads. In *2016 American Control Conference (ACC)*, pages 1452–1457, July 2016.

[111] Jun Zhu and A. Abur. Effect of Phasor Measurements on the Choice of Reference Bus for State Estimation. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1–5, June 2007.

[112] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas. MATPOWER: Steady-State Operations, Planning and Analysis Tools for Power Systems Research and Education. *Power Systems, IEEE Transactions on*, 26(1):12–19, 2011.