A Network Graph-based Framework for Modeling, Calculating and Controlling Feasible AC Electric Power Delivery

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Andrew Hsu

B.S., Electrical Engineering, Columbia University
M.S., Electrical & Computer Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

October 2015

Acknowledgments

Firstly, I would like to thank my thesis advisor, Professor Marija Ilić, for her continuous support, motivation, and great ideas. Her guidance pushed me forward and helped me develop my ideas into something concrete and applicable. As a person, her warmth and thoughtfulness made the group like a family, and made the Ph.D. experience much more memorable.

I would also like to sincerely thank Professor Soummya Kar, Professor Zico Kolter, and Dr. Jason Black, for being on my thesis committee and for giving me their feedback and insights. Their questions and comments about my research helped me better understand the theory, as well as see potential new research avenues.

I am grateful for my sponsors, Converteam, Semiconductor Research Corporation, Nexans, and the United States' National Institute of Standards and Technology. Without their support this research would not be possible. I would like thank Sandy Aivaliotis and Eric Hsieh of Nexans for their feedback and discussions about the possible implementations of our research.

I also want to thank my fellow graduate students and friends from Carnegie Mellon University. They are all bright, interesting people who have passion for their research. We have had many good discussions and spent a lot of quality time together.

Lastly I want to thank my family; my parents, my brother, and my grandmother, who were all there with me when I was growing up. They all believed in me and made sure I had what I needed to succeed.

Abstract

An influx of technology has changed the traditional, top-down approach to electric power systems, into an interconnected electrical and informations network with large generators, intermittent renewables, storage, and demand response. The transmission network has seen its fair share of innovation, as new hardware is introduced for both monitoring and control purposes. For example, distributed line rating units (DLRs) are used to determine a more accurate thermal rating for transmission lines. Flexible AC Transmission Systems (FACTS) devices are capable of adjusting the reactance of lines using power electronics switching. Even newer technologies, like distributed series reactances (DSRs) are offering similar adjustment capabilities at lower cost. How can these devices be coordinated to run the electric power system more efficiently and reliably?

This thesis proposes a framework for modeling power systems using a graph representation. This localizes each individual component's behavior and only interconnects the system through the power and voltages of the components, over a network graph that obeys Kirchhoff's laws. Using this framework, a number of new formulations for the power flow problem are introduced.

The first formulation poses the conventional power flow problem, in terms of nodal voltage magnitude and phase angle, as a complex-value domain problem in terms of complex-valued nodal voltages, for systems with only a slack bus and PQ buses.

Secondly, the complex-value domain power flow problem has been reposed in terms of the branch voltages, or voltage difference across the lines of the network. By combining this with the S-E graph model, a π model of transmission lines, a set of power flow equations was formulated in terms of the power transferred through each line, and the voltage across each line.

However, the branch-based formulation of power flow is more difficult to solve using the conventional numerical methods, especially Jacobi method. Therefore, a new optimization-based formulation of power flow is developed, in terms of branch variables. This thesis shows how distributed power flow calculation, performed by smart wires and buses, can be performed through only communication with neighbors. This new formulation is also instrumental in deriving methods for ensuring feasible power delivery.

If there is no power flow solution, there is no equilibrium value for the voltages of the network, and will lead to voltage collapse. To ensure that power is successfully delivered across the network, the components of the network must adjust so that there is a valid power flow solution. This thesis proposes two methods of ensuring feasible power delivery; targeted load shedding, and adjustment of line reactances.

By using Lagrangian relaxation to solve the new optimization-based power flow problem, the set of possible solutions is expanded beyond just the possibly power flow solutions of the network. If there is no valid power flow solution, then the optimization problem can still converge to a solution. The Lagrange multipliers, which correspond to physical constraints, such as nodal power balance, therefore becomes a measure of power mismatch at each bus. Using the numerical results of the Lagrange multipliers, adjustments to load can be made, allowing targeted load shedding.

Another method of reaching a feasible power flow solution is by adjusting the parameters of the lines connecting the generators and loads. It may be less desirable to shed load, which would disconnect customers from power, than to use devices such as DSRs and FACTS devices to change the reactance of lines. By using the mathematical closed form power flow solution of a two-bus power system, the conditions for feasible power delivery across one line can be derived. Extending this to the multi-bus network allows each line to determine whether it should adjust, to reach a valid power flow solution.

Finally, proof of concept simulations show the distributed calculation of power flow, targeted load adjustment and line reactance adjustment on 3 bus, 14 bus, and 45 bus systems. An object-oriented programming platform is used to simulate power systems in an actual distributed environment, with separate Matlab processes communicating between one another. The distributed power flow algorithm has been implemented for the 3 and 14 bus networks on this platform.

Contents

Acknowledgments i

Abstract ii

1 Introduction 1

1.1 Introduction 1

1.1.1 Problem Statement 3

1.1.2 Thesis Contributions 4

2 State-of-the-Art 7

2.1 Industry State-of-the-Art 7

2.1.1 Current System Operation 7

2.1.2 Smart Wire Devices 11

2.2 Theoretical State-of-the-Art 12

2.2.1 Power Flow Calculation 12

2.2.2 Feasible Power Delivery 13

3 Modeling Power Networks Using Graphical Representation 16

3.1 Graph Representation of Power Networks 16

3.1.1 Power Components as Branches and Nodes 16

3.1.2 Directed Graph for Power Flows 20

3.1.3 Non-nodal Spanning Trees 22

3.2 Nodal Power Flow Equations 24

3.2.1 Power Flow Equations 24

3.2.2 Conventional Formulation of Power Flow 26

3.3 Branch-based Power Flow Equations 27

3.3.1 Branch-based AC Power Flow in Complex Variables 27

3.3.2 New Optimization-based Formulation of AC Power Flow 28

3.3.3 Decoupled and DC Power Flow Formulations 31

4 New Numerical Algorithms for Complex-valued Power Flow 32

4.1 New Iterative Algorithms to AC Power Flow in Complex-Valued Domain 32

4.1.1 New Nodal-based AC Power Flow in Complex-Valued Domain 34

4.1.2 New Branch-based AC Power Flow in Complex-Valued Domain 37

4.1.3 AC Power Flow Using New Optimization-based Formulation 39

5 Ensuring Feasible Power Delivery 47

5.1 Feasible Power Delivery Through Wires of the Network 47

5.1.1 Feasible Power Delivery On The 2 Bus Network 47

5.1.2 Feasible Power Delivery On Multi-Bus Networks 50

5.2 Load Shedding to Ensure Feasibility 50

6 New Decoupled and DC Power Flow Formulations 52

6.1 New Decoupled and DC Power Flow Formulations of Real Power Flow 52

6.1.1 Decoupled Power Flow Formulations 52

6.1.2 DC Power Flow Formulations 56

6.2 Solving Decoupled and DC Power Flow 58

6.2.1 Conventional Nodal-based Decoupled Power Flow 58

6.2.2 New Branch-based Decoupled Power Flow 59

6.2.3 New Optimization-based Decoupled Power Flow 62

6.2.4 Conventional Newton Method for Nodal-based DC Power Flow 65

6.3 Ensuring Feasible Power Delivery using the New Decoupled Optimization-based Formulation 69

6.3.1 Targeted Load Shedding 69

6.3.2 Line Reactance Adjustment 69
6.4 Decoupled Distributed Power Flow and Infeasible Power Delivery Simulations 70
6.4.1 2 Bus Example 70
6.4.2 3 Bus Example 72
6.5 IEEE 14 Bus Example 73

7 Simulations and Examples 79
7.1 Power Flow Calculation 79
7.1.1 3 Bus Example 79
7.1.2 14 Bus Example 84
7.1.3 Flores 45 Bus 85
7.2 Targeted Load Shedding and Reactance Adjustment 87
7.2.1 3 Bus Example 87
7.2.2 14 Bus Example 92
7.2.3 Flores 45 Bus 95

8 Implementation Using NIST SGRS 99
8.1 NIST Smart Grid in a Room Simulator 99
8.1.1 Summary of the NIST SGRS Platform 99
8.1.2 SGRS Distributed Power Flow Application 101
8.1.3 NIST SGRS DPF Contribution 103

9 Conclusions and Future Work 106
9.1 Conclusion 106
9.2 Open Questions 107
9.3 Future Work 108

Appendices 110
A.1 Numerical Methods and Conditions for Convergence 110
A.1.1 Newton-(Raphson) Method 110

A.1.2 Jacobi Method 111

Bibliography **112**

List of Figures

2.1 A map of New York state, showing the power delivery path from Niagara to New York City. 8

2.2 A typical P-V curve, showing the maximum possible power transfer. 9

2.3 A thyristor controlled series compensator, a type of FACTS device, made by Siemens. 10

2.4 A distributed series reactance (DSR) device, from Smart Wires Inc. 11

2.5 A power system with a constant impedance load. 13

3.1 A 3 bus system interconnected with 3 lines. 17

3.2 The 3 bus system represented as a graph. Nodal branches are shown as dotted lines. 17

3.3 The pi model, S-E graph of a wire with transmission losses. 20

3.4 S-E graph of the 3 bus system. 21

3.5 The 3 bus system with a spanning tree indicated in solid line, and the extra branch indicated in dotted line. 23

4.1 A flowchart of the information exchange required to execute the distributed Newton method for complex power flow. 46

5.1 The 2 bus network is the base example for deriving the conditions of feasible power delivery. 47

5.2 The transmission line of the 2 bus system represented as an S-E graph pi model. . . 48

6.1 2 bus example with a slack bus and a constant power load. 70

6.2 3 bus example with a slack bus, another generator and a constant power load. 72

6.3 IEEE 14 bus system. The two circled transmission lines were adjusted so that an infeasible power flow for the network was created. 75

7.1 A 3 bus system interconnected with 3 lines. 79

7.2 The IEEE 14 bus network, with 20 lines. 83

7.3 The 45 bus distribution network of Flores island. 92

7.4 The IEEE 14 bus network, with two lines selected to have much higher reactance. . 93

8.1 The objects are connected by communicators, and allocated computation resources by the broker. 100

8.2 The objects and interconnection can be entered through a web browser interface. . . 101

8.3 A flowchart of the information exchange required to execute the distributed Newton method for complex power flow. 102

8.4 The properties and methods of the SGRS distributed power flow bus module. 103

8.5 The properties and methods of the SGRS distributed power flow line module. 104

8.6 A 2 bus example divided into modules for the NIST SGRS simulator. 104

8.7 The communications between modules in the 2 bus example. 105

8.8 A diagram of the processes used to simulate distributed power flow on the 2 bus system. 105

List of Tables

6.1 Optimization based method solution for the 2 bus network: infeasible case 72

6.2 Newton method solution for the 3 bus network: feasible case 73

6.3 Optimization based method solution for the 3 bus network: feasible case 74

6.4 Optimization based method solution for the 3 bus network: infeasible case 74

6.5 Newton method solution for the 3 bus network: infeasible case with adjusted injections 74

6.6 Optimization based method solution for the 3 bus network: infeasible case with adjusted injections 74

6.7 Optimization based method solution for the 3 bus network: infeasible case with adjusted reactances 75

6.8 IEEE 14 bus decoupled results (infeasible, adjusted injections) 77

6.9 IEEE 14 bus decoupled results (infeasible, adjusted line reactances) 78

7.1 AC, Complex Power Flow for the 3 and 14 Bus Examples 85

7.2 85

7.3 Decoupled, Lossy Real Power Flow for the 3 and 14 Bus Examples 86

7.4 86

7.5 Decoupled, Lossless Real Power Flow for the 3 and 14 Bus Examples 87

7.6 87

7.7 AC Power flow solutions of the IEEE 14 bus system with only PQ buses. 88

7.8 Decoupled Lossy and Lossless Power flow solutions of the IEEE 14 bus system. . . . 89

7.9 Results of Jacobi method using nodal formulation of decoupled, lossy power flow on the 14 bus system. 90

7.10 Power flow results of the Flores 45 bus distribution network. 91

7.11 Power flow solutions of the IEEE 14 bus system with adjusted reactances. 94

7.12 Bus power mismatches of the infeasible IEEE 14 bus system with only PQ buses. . 95

7.13 Changes in reactance of the infeasible Flores 45 bus distribution system. 96

7.14 Bus power mismatches of the infeasible Flores 45 bus distribution system. 98

Chapter 1

Introduction

1.1 Introduction

Since the development of alternating current (AC) power transmission, electric power systems have consisted of large generators connected to electrical loads by a system of high voltage transmission lines. This has led power grid operation to a top-down approach, where the generators are controllable, and the transmission system and loads are considered given. However, there is an influx of technology, both on the transmission and load side, which is changing this paradigm. Renewable, intermittent sources of energy, such as wind and solar, are not centralized generators with large inertia, but are typically collections of smaller plants distributed across the power system. Demand response and storage technologies are increasing the amount of control available on loads. Flexible AC Transmission Systems, or FACTS, devices provide the ability to control the impedances of transmission lines, allowing for control of the power flowing through the lines. Accompanying the physical hardware are improvements to communications and computation infrastructure, giving the devices the ability to interact with system operators and each other. These communication and computational abilities lead to efforts to make the electric power systems of the future ”smart”, by not only having newer technologies but also better operation.

The term smart grid has been frequently used to describe the electric power system of the future. Frequently, the term smart refers to the new devices which can collect data from, communicate between, and even control various parts of the electric energy system. Devices such as smart meters, phasor measurement units (PMUs), dynamic line rating units (DLRs) are being introduced to monitor the various aspects of the power grid. FACTS devices, and similar devices such as distributed series reactances (DSRs), allow a certain amount of line flow control over wires that was not available in the past. These technologies, while powerful alone, may become instrumental in improving the way the system is operated, if properly coordinated.

The design of the intelligences of smart devices must be based on the purpose of the application and mathematical models used. To run an electric power system efficiently and reliably, the system must be monitored and controlled. This occurs on different time scales, such as the fast time scale, encompassing dynamic and transient effects over seconds or fractions of a second, steady-state effects that happen over minutes, or scheduling and market events that happen over intervals of tens of minutes, hours, or even longer. Mathematical models of the power system must be chosen such that the inputs, constraints, and variables are properly described, under the correct assumptions given the time scale. Then, numerical methods are applied to calculate the desired outputs, given the mathematical model. If the numerical method can be decomposed into algorithms that can be executed by the various distributed smart devices, then the intelligence of each device can be designed and assigned.

At operating centers, power flow equations have been solved as systems of simultaneous equations, and KVL equations are usually implicit through the use of bus voltage variables (voltage with respect to ground). KCL and constituent relationship equations are usually combined into bus power balance equations. These systems of equations can be solved through numerical methods such as Newton-Raphson method [1], [2]. This requires that the computing platform have access to all relevant information of every component in a system.

In areas such as optimal network resource allocation, and in particular, optimal power flow, methods have been introduced that allow distributed or decentralized calculation. Work by Steven Low, such as [3], [4] feature convex relaxation of the optimal power flow problem, and conditions under which this relaxation is exact are determined. Under these conditions, a guaranteed optimal solution exists, but this is only the case for radial networks, such as typical electrical distribution networks. [5] further adds to this work by relating the convex relaxation solutions to the network topology, allowing for convex relaxation methods to be used in meshed networks.

There have been power flow calculation algorithms, such as *Dist Flow branch equations* [6], that define a line's power balance in terms of the port variables of the line before it, allowing a distributed, but sequential, calculation of power flow on radial networks. This makes it primarily useful on distribution networks, which are frequently radial. *Dist Flow* can also be used for load balancing on distribution networks.

Aside from distributed power flow, there is the question of encountering nonconvergence in power flow algorithms, especially Newton-(Raphson) method. Does that mean there is no solution, or that the numerical method simply cannot reach one? In systems with impedance loads, there is a maximum amount of power that can be transferred through the wires to the loads [7]. This means there are physical reasons for there not being power flow solutions, even if generation can meet demand. Is there a systematic way to address these problems while calculating power flow?

1.1.1 Problem Statement

How do we design autonomous intelligence for smart power system devices (such as smart wires), such that they monitor steady-state behavior of the system, and adjust themselves to ensure feasible power delivery?

Before the intelligence of an individual component is designed, a framework is needed to model both the component and its interaction with the rest of the system. The goal of this work is to provide a framework through which power systems models can be built to address system-wide monitoring and control problems, while integrating various distributed components. This framework uses a graph representation of the power system to model individual devices and their interactions, and the structure of the graph is preserved when numerical algorithms are applied. By combining the graph representation of a power system, with the physical laws that describe power flow, mathematical models of power flow can be formulated.

This thesis first proposes a method of coordinating distributed, smart devices to monitor electric power systems by calculating their steady-state power flow solutions, either autonomously or complementary to the monitoring done by system operators. The intelligence and communications required by each component comes directly from decomposing the mathematical model of power flow such that an algorithm can be applied using only local and exchanged information from neighbors at each component.

This thesis also proposes conditions for feasible power delivery based on these mathematical models. There are conditions derived from lack of power flow convergence, and conditions associated with the constituent relationships of each individual line. These condition allow adjustments of load and line reactance (provided the hardware is capable) to be made such that a power flow solution exists.

1.1.2 Thesis Contributions

The specific contributions of this thesis are as follows:

Branch-based Graph-representation of Electric Power Systems

While the use of graph-representations of power systems is not new [2], using a directed graph of branches that represent power flow is essential to formulating the optimization-based power flow formulation developed in this thesis. This thesis uses the S-E graph [8] model of lossy power delivery through lines. It expresses flow-voltage relations in terms of voltage differences across these lines, instead of in terms of more conventional nodal voltages. Because of this, the graph is called branch-based, or flow branch-based; the branch-based voltages are used explicitly in the calculation of power flow. Similar branch-based formulations used in the past assume negligible power loss and a radial network structure [6]. A general lossy, complex valued power flow problem is formulated using branch-based formulation for the first time.

The advantage of this formulation is two-fold; first, this formulation lends itself to distributed calculation by the lines, or branches, of the network, if they are embedded with smart, computational capabilities. The second is that this formulation also allows a line by line accessment of whether power delivery across the line is feasible. We conclude that when the delivery problem is the main question, this approach is generally a more appropriate one.

Formulation of Power Flow Problem as an Optimization Problem

The power flow problem is typically a simultaneous equations problem, derived from Kirchhoff's laws and the constituent relationships of each line of the network. In this thesis, we formulate it as an optimization problem, where the objective function seeks to minimize the difference between what the power flow through lines would be as a result of Kirchhoff's laws, or network constraints, and the power flow through lines as a function of voltage. In AC and nonlinear decoupled power flow, this results in a new nonlinear objective function with linear constraints [9], [10], [11], [12].

A distributed method for solving optimization problem for flow networks [13] was the inspiration for formulating the power flow as an optimization problem in terms of branch-based flows and voltages. This optimization-based formulation is the driving force in both the distributed calculation algorithm as well as the method on ensuring feasible power delivery.

Ensuring Feasible Power Delivery

This thesis also tackles the problem of feasible power delivery as a result of power flow equations, a distinct problem from combatting congestion resulting from transmission lines' thermal limits. Given generation, load, and a network connecting them through lines with given impedances, can the power be successfully delivered?

A 2 bus example of AC power flow [2] has a closed form power flow solution, and the limit to power flow can be found. In the multi-bus case, it is impossible to write the closed form, so solution existence is not easily inferred.

This thesis proposes two methods of addressing this problem. The first is based on applying our newly posed branch-based optimization-based formulation, which is a more relaxed problem than calculating power flow as simultaneous equations. By using Lagrangian relaxation, the Lagrangian multipliers resulting from Kirchhoff's laws represent power mismatches at buses. Therefore, if the optimization problem converges to non-zero Lagrange multipliers, the result is not a valid power flow solution, but the mismatches relay information about how the network can be adjusted to lead to a valid power flow solution.

The second method is to check the power and voltage variables of each line as the power system is iteratively solved. A 2 bus example is used to develop a feasible power delivery condition, which is checked by each line using local variables. If reactance control is possible, as a result of hardware installed on the lines, then this information can be used to adjust reactances such that feasible power flow conditions are met.

Both methods can be implemented in a distributed, or centralized manner, because the optimization-based power flow formulation can be implemented either using a centralized algorithm, such as Newtons method, or with the proposed distributed algorithm [14].

Setting the Basis for Smart Electric Power Delivery

Finally, we close by pointing out the theoretical and algorithmic contributions in this thesis form the basis for utilizing hardware capable of direct line flow control by adjusting parameters of once passive transmission and distribution wires. As new architectures of smaller electric power systems emerge, methods will be needed to identify the most critical roadblocks to electrical ddelivery of power for given generation sources and given points of power consumption. Being able to understand the ultimate limits to delivery and to install hardware for relaxing these limits in a somewhat component-by-component way is likely to become essential for possible plug-and-play specifications of solutions needed. This thesis provides the first steps in the direction of such modular thinking applied to electric power delivery.

Chapter 2

State-of-the-Art

2.1 Industry State-of-the-Art

2.1.1 Current System Operation

The operators of the generation and transmission system use a number of tools to access and adjust the electric power system. These computer aided tools are typically used by system operators in Energy Management Systems (EMS). The monitoring and control system is called the Supervisory Control And Data Acquisition (SCADA) system. The SCADA network operates as a state estimation system that makes measurements of various portions of the transmission grid, typically every 2 seconds. It then uses these redundant measurements to infer the state of the network, and to detect bad data or topology changes. This set of data is typically not accurate, and does not satisfy power flow equations [15].

The EMS also performs a task called contingency screening. Using the data collected from the system, the EMS does a series of (N-1) reliability tests every 5 minutes. (N-1) refers to every combination of one equipment failure in the entire system. For each of these possibilities, an approximate, so-called DC, power flow is run to determine if any line flows will violate a transmission line hardware limit, such as the thermal limit. Optimal power flow (OPF) is typically used to adjust generation to not violate thermal hardware limits.

In addition, EMS performs a number of simulations, while incrementally increasing the load

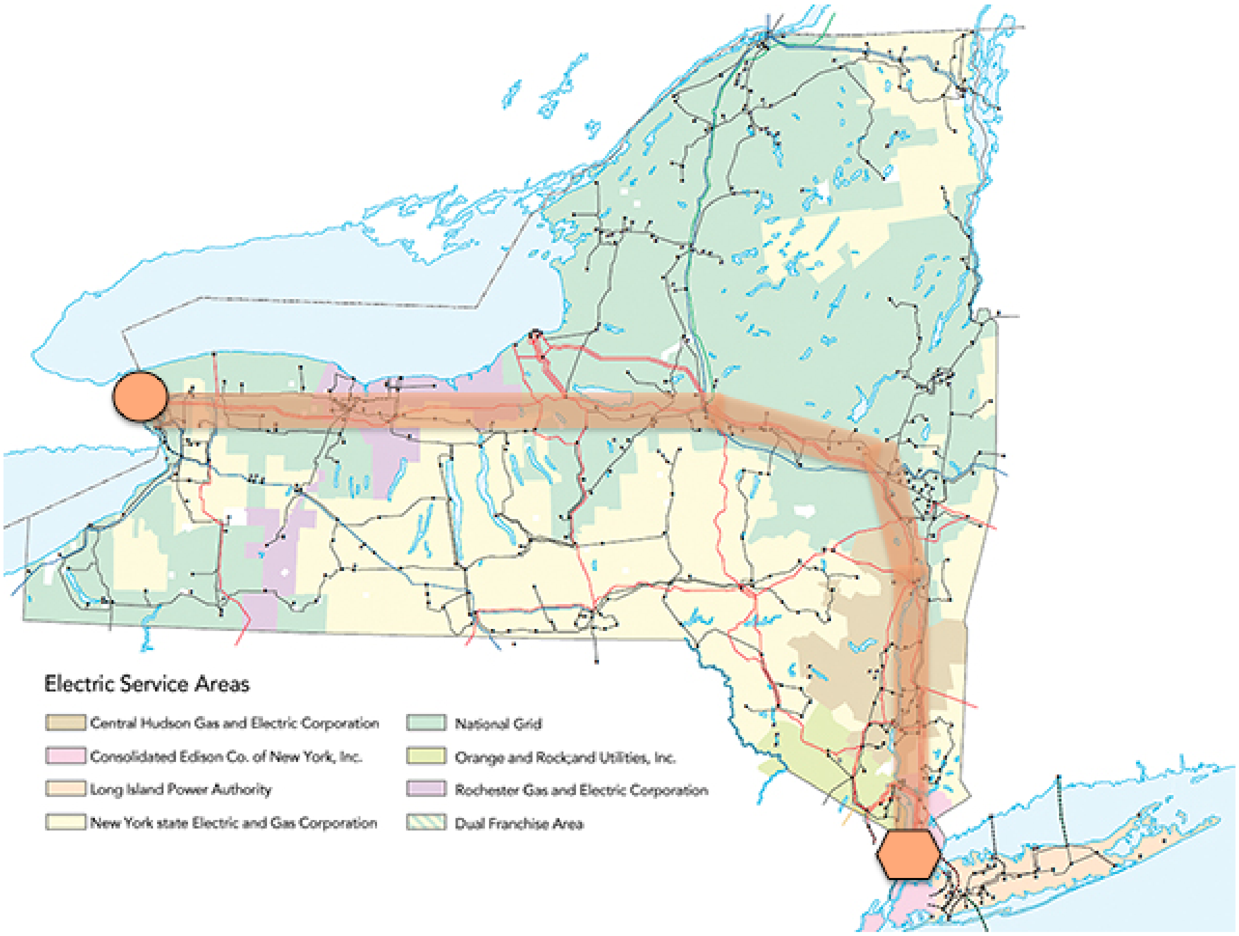

Electric Service Areas
Central Hudson Gas and Electric Corporation
Consolidated Edison Co. of New York, Inc.
Long Island Power Authority
New York state Electric and Gas Corporation
National Grid
Orange and Rockland Utilities, Inc.
Rochester Gas and Electric Corporation
Dual Franchise Area

Figure 2.1: A map of New York state, showing the power delivery path from Niagara to New York City.

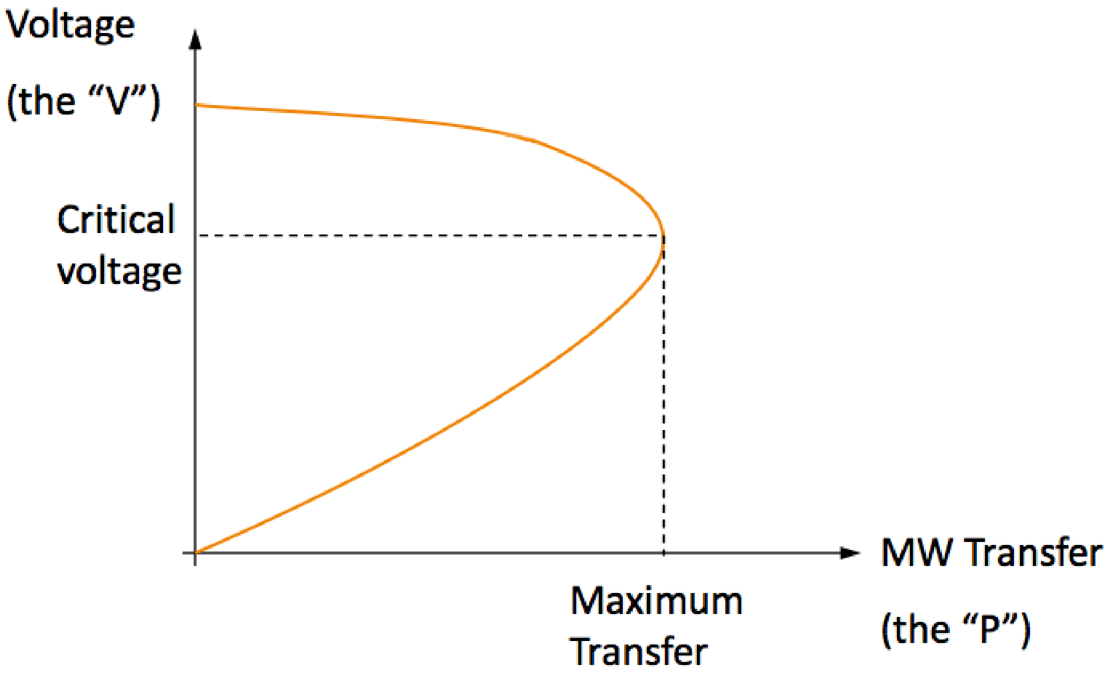


Figure 2.2: A typical P-V curve, showing the maximum possible power transfer.

and generation of the system, in order to determine the maximum power transfer limit between parts of major interest in the system [16], [17]. For example, in the New York electric power grid, the delivery of hydro power from Niagara to New York City would be of major interest. Figure 2.1 shows the path through which the power would be delivered, from Niagara (indicated by the circle) to New York City (indicated by the hexagon). In such a procedure, AC power flow is calculated for the system under normal operating conditions. Then, the load and generation is increased incrementally and AC power flow is calculated again. This is repeated until AC power flow ceases to converge, and it is determined that this is the maximum power transfer that the system operators want to consider feasible, because if AC power flow cannot be calculated, it cannot be guaranteed that a solution exists, and, consequently, that the power can be physically delivered.

This procedure results in so-called P-V curves, which, under certain assumptions(such as constant load power factor), define limits to electrically deliverable power between two points inthe power grid. Figure 2.2 shows a typical P-V curve, and indicates the maximum power transfer limit. Moreover, the P-V curves are obtained for the most limiting (N-1) equipment outages and are used

Figure 2.3: A thyristor controlled series compensator, a type of FACTS device, made by Siemens.

to effectively determine a proxy limit to the transmission line interconnecting the two points. Due to paramter uncertainty concerns, this proxy line flow limit is reduced even further, by up to 13

Power flow is calculated to corroborate the state estimation system, by checking if the topology and line flows make sense. Power flow calculation consists of determining the bus voltages and line flows which result from a given set of generation, loads, line impedances and system topology. This is mathematically posed as conservation of power flow at each bus, expressed in terms of voltages based on the constituent relationships of the lines connected to each bus. These power flow equations are typically real and reactive power balance equations at buses, using the voltage magnitude and phase angle as variables (reactive power balance is omitted for PV buses). Approximations can be made, such that real power is strongly coupled with phase angle, and reactive power is strongly coupled with voltage magnitude, resulting in decoupled power flow equations. Decoupled, real power flow equations can be further approximated through linearization, resulting in so-called DC power flow. This term should not be confused with the DC circuits problem. It has evolved historically in the electric power industry, and, as such, it is routinely used to imply a linearlized, decoupled, real power flow problem.

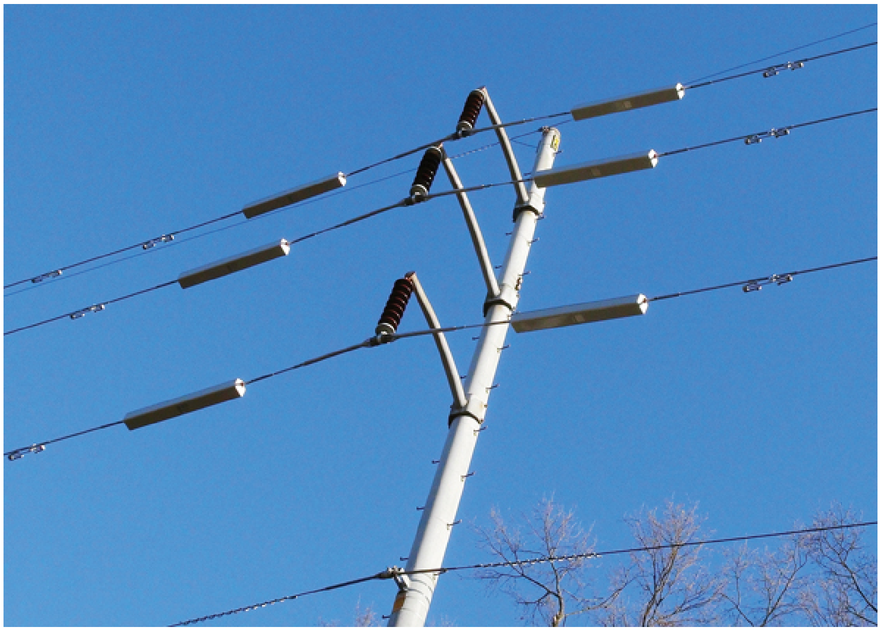


Figure 2.4: A distributed series reactance (DSR) device, from Smart Wires Inc.

2.1.2 Smart Wire Devices

FACTS devices, such as Thyristor Controlled Series Compensators (TCSCs), are capable of adjusting the reactance of a transmission line by controlling inductors and capacitors with fast power-electronics switching [18]. Currently, the potential benefits lie in the economic dispatch area. For example, FACTS devices can address congestion problems by redirecting power down other paths [19]. This is also applicable to technologies by SmartWires Inc., known as ”D-FACTS”, or distributed-FACTS devices, which have similar functions but are distributed and typically less expensive than traditional FACTS devices [20], [21], [22]. The recommended use of these devices are to better utilize system resources, such as dispatch of generation resources, to supply the demand at lowest cost. Shown in Figure 2.3 is a typical TCSC, a FACTS controller installed to control transmission line reactance in a BTS [23]. Also, shown in Figure 2.4 is a recently developed poor-man’s version of FACTS for lower voltage levels, an example of D-FACTS .

Relevant to this thesis is the observation that these harware technologies are becoming mature. However, quantifiable methods for finding the most effective locations and for finding the most effective adjustment logic are at their rudimentary stages. They are by-and-large based on extensive scenario analysis, which uses similar approaches to the ones described for generating P-V curves. Missing is a method for mapping the transfer limit found by the P-V curve onto the best line whose reactance needs to be adjusted to increase the feasible transfer. This problem is very combinatorial and non-unique when tackled in a top-down approach using centralized power flow calculation. This problem is one of the major motivating factors for the bottom-up distributed approach proposed in this thesis.

2.2 Theoretical State-of-the-Art

2.2.1 Power Flow Calculation

Traditionally, the electric power system has been controlled through a top down approach, by a centralized decision maker. The system operator collects information from the state estimator, which takes redundant measurements from points in the network and relays them back to the centralized hub. The system operator must then use the collected data and known parameters and topology of the network to determine the status of the electric power system. This information is also used to determine if the assumed topology makes sense, or if a specific sensor is returning bad data. By keeping an eye on the status of the grid, the operator tries to make sure that there will be no problems in delivering energy generated to its intended consumers.

The conventional method of calculating power flow is the Newton-(Raphson) method [1], [2], which iteratively calculates the solution to nonlinear simultaneous equations by inverting the Jacobian matrix of the equations. Typically, Jacobians are sparse because the elements reflect the interconnections of the network. However, inversion removes the structure, so Newton method requires global information to execute.

Distributed calculation is much easier to perform over a radial network, or a tree graph. [6] is an algorithm that sequentially goes through a radial network to calculate the power flow, and is therefore typically useful on distribution, rather than transmission, networks of the power system. The radial structure allows for optimal power flow (OPF) problems to be posed such that they can be convexified [3], [4], [24]. However, it was recently found that OPF can also be convexified for meshed networks [5], if the rank of the solution to the relaxed problem is 1. Applying convex OPF

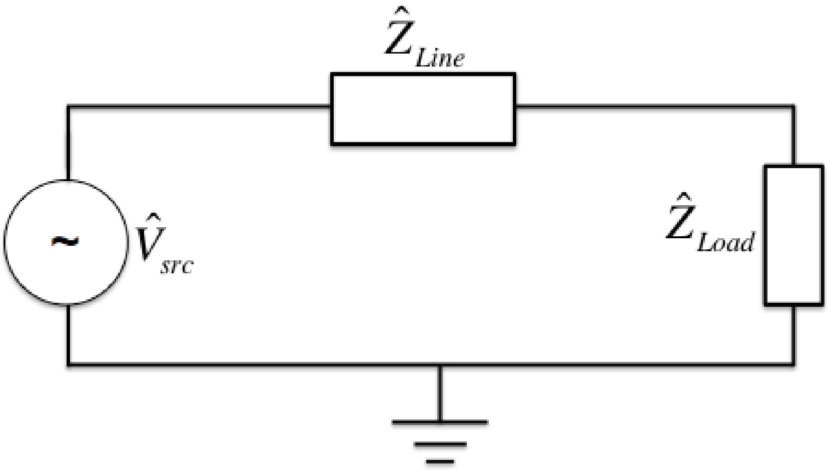


Figure 2.5: A power system with a constant impedance load.

to meshed networks depends on using the branch-based voltages, defined as bus voltage differences across the branches.

2.2.2 Feasible Power Delivery

To ensure power delivery from generation to consumers, there are a couple of conditions which must be satisfied. For any network of impedances interconnected given sources and loads, there is a theoretical limit on the amount of power that can be sent from the sources that will reach the loads. General maximum power transfer results for circuits in which loads are modeled as constant impedance loads exist and can found in [7]. Figure 2.5 shows a two bus power system where the load is a constant impedance. According to [7], the maximum power transfer occurs when $\hat{Z}_{line} = \hat{Z}^*_{load}$, where * denotes the complex conjugate.

Unfortunately, for typical BTS line parameters, the system cannot operate close to the maximum power transfer limit because of typical high voltage resulting at the receiving end [2], [25]. This voltage occurs because the maximum power transfer condition, $\hat{Z}_{line} = \hat{Z}^*_{load}$, requires load to be compensated by shunt capacitance so that it compensates for reactive power losses occuring in typically inductanive transmission lines.

In addition, there are hardware limits, such as the thermal limit of transmission lines. This refers to an operating limit on the power transfer, or current that can flow through a transmission line before it physically fails, such as from overheating.

Currently, the system operators do contingency screening, looking for critical cases where the power flow solution of a network might not exist if one component were to fail. They also use the idea of voltage stability, which calculates test cases where the generation and load are slowly increased. This is to determine at which power transfer level the power flow solution does not converge, and gives a conservative limit on the maximum power transfer. Without a power flow solution, the voltage does not have a stable equilibrium point [26], [27]. When reaching that boundary, such as if load were increased, it is possible to see a slow but then rapid decline in voltage magnitude, corresponding to reaching the edge of the stable voltage region.

Theoretical works, such as in [7], try to pose the problem of maximum power transfer mathematically. However, the model used considers a power system with complex voltage sources, and impedance loads. In conventional power systems, most generators are modeled as PV, or constant real power and voltage magnitude, and loads are modeled as PQ, or constant real and reactive power. [2] introduces a two bus case with PQ load, and gives a range for power transfer within which the voltage magnitude is real (and therefore the solution exists).

Current theoretical state of the art, however, does not give an obvious way to adjust line based smart devices, such as FACTS devices to ensure a feasible power flow solution. The closest is to pose OPF without allowing generation to be dispatched, and considering them as given. This is because the power flow equations have traditionally been posed in terms of nodal variables, more specifically the voltage phase angle at each node.

We propose that a new approach be used, that considers the power flow equations in terms of branch variables, such as the phase angle difference across each line. We also propose a way to solve the power flow equations as an optimization problem instead of solving the conventional formulation of algebraic equations. The structure of the optimization problem lends itself to decomposition, allowing for a distributed iterative approach that treats both buses and lines in the network as agents. This allows the line agents to analyze the power flow and voltage of a line, and determine if the maximum power transfer limit would be violated on a line by line basis, thus allowing line based controllers, such as FACTS devices, to be adjusted. This would complement the ability to check line thermal hardware limits.

Chapter 3

Modeling Power Networks Using Graphical Representation

3.1 Graph Representation of Power Networks

3.1.1 Power Components as Branches and Nodes

An electric power system consists of generators and loads, interconnected by a series of wires. All of these components can be represented as branches in a directed graph. The buses of the electric power system, the points of interconnection between generators, loads and wires, are the nodes of the directed graph. The direction of each branch serves as reference for power flow direction and voltage orientation, with the positive voltage terminal on the sending end of each branch.

Consider a 3 bus power system, depicted in Figure 3.1. The system consists of one generator, two loads, and three transmission lines, or wires. Figure 3.1, like most power systems diagrams, assumes that the generators and loads are connected to ground, and therefore does not explicitly show this. The graphical representation of the network, including the ground, or neutral line, is shown in Figure 3.2. The generator and loads are shown as dotted lines, as branches from each respective bus to ground. In the graphical representation, buses are now abstract connection points, and can be considered as nodes of the graph.

If the full graph representation has $N + 1$ nodes, and L branches, then N branches are required

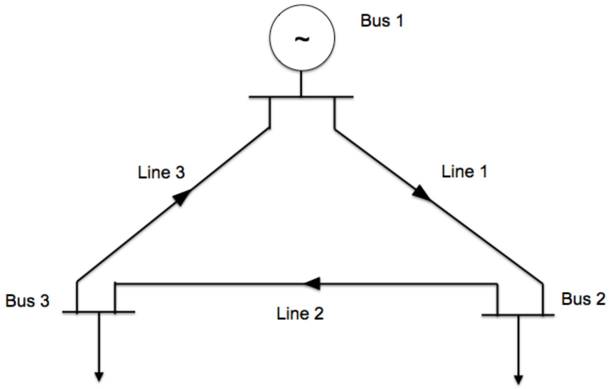


Figure 3.1: A 3 bus system interconnected with 3 lines.

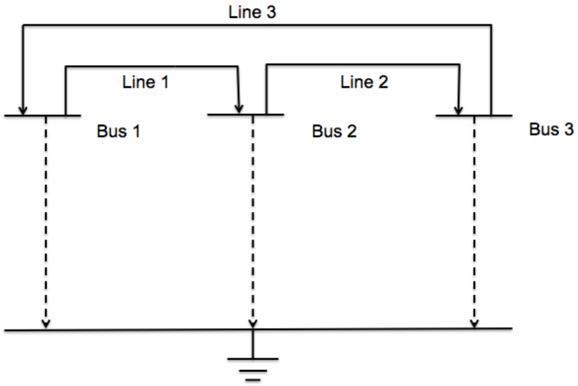


Figure 3.2: The 3 bus system represented as a graph. Nodal branches are shown as dotted lines.

to span the graph. If the $(N+1)^{th}$ node is ground, then the generators and loads form a set of N branches that span the graph. This normal tree, or spanning tree, is referred to as the nodal branches, which connects each node of the graph to the ground node. The rest of the branches of the graph correspond to the wires from the physical power system; they interconnect all the nodes without traversing the ground node. In the physical world, power delivery is achieved through these wires, while ground serves as an electron sink or source. The branches delivering power will be referred to as flow branches.

A directed graph is used, so an incidence matrix can store the interconnection data of the network. The incidence matrix is an $N \times L$ matrix $A_a = [a_{i,k}]$, where

$$a_{i,k} = \begin{cases} 1 & \text{if node } k \text{ incident at } i\text{, arrow away from } i \\ -1 & \text{if node } k \text{ incident at } i\text{, arrow into } i \\ 0 & \text{if node } k \text{ not incident at } i. \end{cases}$$

Because each branch necessarily connects two distinct nodes, each column contains exactly one 1 and one -1 term, the rest being zeros. Therefore, a row of A_a could be removed without losing irretrievable information. This reduced matrix, A_a^T, is used to describe Kirchhoff's current law (KCL) and voltage law (KVL) of the network. For the graph shown in Figure 3.2,

$$A_a = \begin{bmatrix} 1 & 0 & -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 & -1 \end{bmatrix}$$

.

The rows correspond to nodes 1, 2, 3, and ground, from top to bottom. The columns correspond to the branches of line 1, 2, 3, and then the nodal branches from nodes 1, 2, and 3, from left to right. A_a^{red} is A_a with the ground node, or bottom row, removed.

KCL can be expressed as

$$A_a^{red} i_b = 0, \tag{3.1.1}$$

where i_b are the currents through all the branches of the graph. Equation (3.1.1) states that all currents at each node sum to zero. If the branches are subdivided into flow and nodal branches, KCL becomes

$$A_a^{red} i_b = \begin{bmatrix} A_f & A_n \end{bmatrix} \begin{bmatrix} i_f \\ i_n \end{bmatrix} = 0, \tag{3.1.2}$$

where the matrix A_f is the portion of the incidence matrix related to flow branches, and A_n is the portion related to nodal branches. Because nodal branches correspond one-to-one with graph nodes, $A_n = I$, because nodal branches point towards ground. In the 3 bus example,

$$A_f = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$$

.

Similarly, KVL becomes

$$(A_a^{red})^T e_n = e_b, \tag{3.1.3}$$

where e_n are the voltages across the nodal branches, and e_b are the currents through all branches. Equation (3.1.3) states that all voltages in loops sum to zero. The one-to-one relationship between nodes and nodal branches means that Equation (3.1.3) can be reduced to

$$A_f^T e_n = e_f, \tag{3.1.4}$$

where e_f are the voltages across flow branches. The removed rows, $A_n e_n = e_n$, are redundant since $A_n = I$.

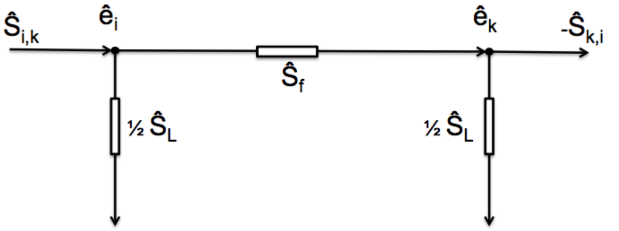


Figure 3.3: The pi model, S-E graph of a wire with transmission losses.

3.1.2 Directed Graph for Power Flows

In the study of electric power systems, the flow of power, instead of current, is used to calculate the steady-state power flow of the network [2]. This is because electricity consumption of loads is typically specified as constant power. The branches of the directed graph do not account for power losses without modeling the system slightly differently than with currents; either each wire is represented as two, opposite directional branches, or a S-E graph representation of the wire is used [8].

The S-E graph is a pi model of a physical wire, where the transmitted power flows across the wire while the half of total power loss leaves the wire from either end. Figure 3.3 shows an S-E graph of a wire connecting nodes i and k. The flows and voltages are complex, using phasors to represent the variables of an AC power system. The flows from either end of the wire, $\hat{S}_{i,k}$ and $\hat{S}_{k,i}$, are the same as in the dual branch model. However, the π branches consist of $\hat{S}_f$ and $\hat{S}_L$, the flow and loss components of the wire. $\hat{S}_f$ is what is actually transmitted from i to k, and $\hat{S}_L$ is divided between two branches that flow to ground. $\hat{e}_i$ and $\hat{e}_k$ are the nodal voltages of nodes i and k.

The S-E graph is used for the remainder of the thesis because the transmitted and lost power are clearly defined and mapped to branches. The dual branch model is equivalent, and the losses and flow can be calculated as a result of $\hat{S}_{i,k}$ and $\hat{S}_{k,i}$. The complex power loss is

$$\hat{S}_L = \hat{S}_{i,k} + \hat{S}_{k,i}, \tag{3.1.5}$$

and the transmitted power is

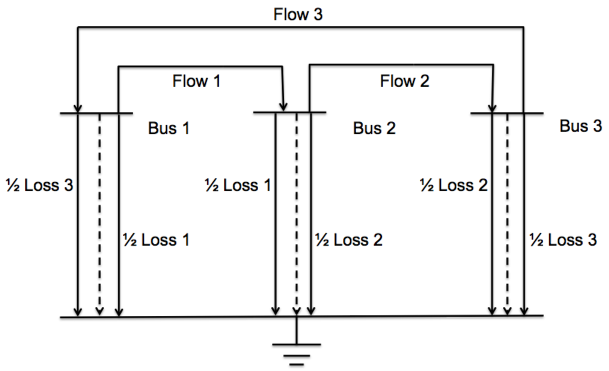


Figure 3.4: S-E graph of the 3 bus system.

$$\hat{S}_f = \hat{S}_{i,k} - \frac{1}{2}\hat{S}_L = -\hat{S}_{i,k} + \frac{1}{2}\hat{S}_L. \tag{3.1.6}$$

The 3 bus system with S-E graph representation of wires is shown in Figure 3.4. The transmitted power branches occupy the same positions as the original flow branches, but the loss branches are additional nodal branches added to the graph. The incidence matrix would therefore be

$$A_a^{red} = \begin{bmatrix} A_f & A_l & A_n \end{bmatrix}, \tag{3.1.7}$$

where

$$A_l = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

.

The first three columns of A_L correspond to the loss branches of the sending ends of wires 1, 2, and 3, and the last three columns correspond to the loss branches of the receiving ends. Because each pair of loss branches accounts for half the power loss, they are always equal in power. Therefore, power flow conservation, the power analog of KCL, can be written as

$$A\hat{S} = \begin{bmatrix} A_f & A_l \end{bmatrix} \begin{bmatrix} \hat{S}_f \\ \frac{1}{2}\hat{S}_L \\ \frac{1}{2}\hat{S}_L \end{bmatrix} = \begin{bmatrix} A_f & A_L \end{bmatrix} \begin{bmatrix} \hat{S}_f \\ \frac{1}{2}\hat{S}_L \end{bmatrix} = -A_n\hat{S}_n, \tag{3.1.8}$$

where

$$A_L = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

.

The A_L incidence sub-matrix treats the losses, $\hat{S}_L$ as one set of losses, instead of pairs of branches. This allows less branch variables to be used in computation, but perserves power flow conservation at each node. A_L is constructed simply by adding the first set of columns of A_l with the second half, creating a matrix with half as many columns. At each node, the power of flow and loss branches balance the power from nodal branches. Because $A_n = I$, each node has the power from one nodal branch. These branch powers, $\hat{S}_n$, can be thought of as the power to or from the load or generator at that node.

Because the voltages across loss branches are nodal voltages, they do not contribute linearly independent equations to KVL. Therefore, the KVL equations remain as in Equation (3.1.4).

3.1.3 Non-nodal Spanning Trees

Even though the set of nodal branches form a spanning tree, it is not the only possible normal tree of a graph. Kirchhoff's theorem states that number of spanning trees in a graph is the determinant of the reduced graph Laplacian. The Laplacian is the negative node-to-node adjacency matrix whose diagonal terms are the positive number of degrees that each node has [28]. The Laplacian is reduced by a specific node, by removing the corresponding row and column. The reduced Laplacian, L_a, is also calculable using A_a^{red}, as

$$L_a = A_a^{red}(A_a^{red})^T. \tag{3.1.9}$$

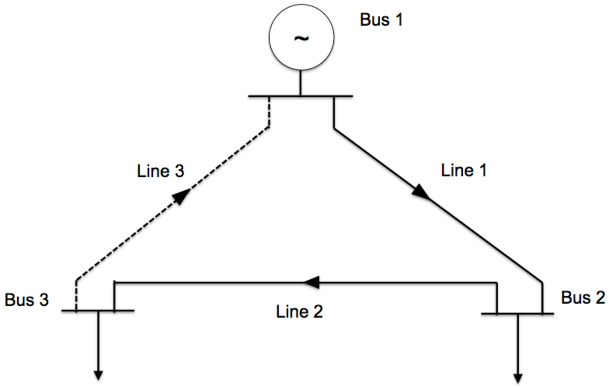


Figure 3.5: The 3 bus system with a spanning tree indicated in solid line, and the extra branch indicated in dotted line.

While it is possible to have a number of spanning trees that include both nodal and flow branches, it is important to focus on formulating spanning trees from just the set of flow branches. This is because KVL must be explicitly defined when using flow branch voltages as variables, unless the wires of the electric power system form a radial network.

Without considering ground node, the minimum number of lines needed to connect N nodes is $N-1$. In an electric power system with L wires, $N-1$ wires form a spanning tree, and the $L-N+1$ remaining wires form loops in the graph. Figure 3.5 shows the 3 bus example with lines 1 and 2 indicated with solid lines, and line 3 indicated in a dotted line. There are three buses, or nodes, and only two wires are required to connect all the nodes. There are three possible spanning trees, (1,2), (1,3), (2,3), and (1,2) is the spanning tree depicted in Figure 3.5. Spanning trees can be constructed using depth and breadth first search, as well as more sophisticated algorithms, designed to increase parallelism [29], [30]. For small examples, a spanning tree can be determined via inspection.

Consider the case where flow branches are renumbered such that

$$e_f = \begin{bmatrix} e_{tree} \\ e_{mesh} \end{bmatrix}, \tag{3.1.10}$$

where e_{tree} are voltages across flow branches that form a preselected spanning tree, and e_{mesh}

consists of voltages across remaining flow branches, which must form meshes.

Equation (3.1.4) gives the relationship between voltage differences across nodal branches and flow branches. Matrix A_a^{red} can be subdivided such that

$$\begin{bmatrix} A_{f,tree}^T \\ A_{f,mesh}^T \end{bmatrix} e_n = \begin{bmatrix} e_{tree} \\ e_{mesh} \end{bmatrix}. \tag{3.1.11}$$

In power flow calculation, one bus is a reference bus, maintaining a voltage at $1\angle 0$ p.u. Therefore, there are $N-1$ nodal branch voltages to be solved. Because there are $N-1$ branches that form a spanning tree, a linear relationship between e_n and e_{tree} can be established,

$$e_n = (A_{f,tree}^T)^{-1} e_{tree}, \tag{3.1.12}$$

and this can be substituted into Equation (3.1.11), to obtain

$$e_{mesh} = A_{f,mesh}^T (A_{f,tree}^T)^{-1} e_{tree}, \tag{3.1.13}$$

which defines the relationship between voltages of non-spanning tree branches and spanning tree branches. Equation (3.1.13) specifically states that each voltage of a non-spanning tree branch is a sum of spanning tree branches (with appropriate orientations). Equation (3.1.13) can be further rearranged to

$$\begin{bmatrix} A_{f,mesh}^T (A_{f,tree}^T)^{-1} & -I \end{bmatrix} \begin{bmatrix} e_{tree} \\ e_{mesh} \end{bmatrix} = A_{kvl} e_f = 0, \tag{3.1.14}$$

which is KVL expressed in terms of flow branch voltages, e_f. $A_{f,tree}$ is a $(N-1) \times (N-1)$ matrix, $A_{f,mesh}$ is a $(L-N+1) \times (N-1)$ matrix, and A_{kvl} is a $(L-N+1) \times L$ matrix.

3.2 Nodal Power Flow Equations

3.2.1 Power Flow Equations

The steady-state power flow through an electric power system is determined by Kirchhoff's laws and constituent relationships of the wires. The generation and loads are given as constant power, making nodal branch flows a given number. The AC power flow examples of this work will not include PV buses, which constraints the voltage magnitude and real power, instead of the complex power of a nodal branch.

Power conservation at nodes, stated in Equation (3.1.8) is

$$A\hat{S} = \begin{bmatrix} A_f & A_L \end{bmatrix} \begin{bmatrix} \hat{S}_f \\ \frac{1}{2}\hat{S}_L \end{bmatrix} = -\hat{S}_n, \tag{3.2.1}$$

and voltage conservation in loops, stated in Equation (3.1.14), is

$$\begin{bmatrix} A^T_{f,mesh}(A^T_{f,tree})^{-1} & -I \end{bmatrix} \begin{bmatrix} e_{tree} \\ e_{mesh} \end{bmatrix} = A_{kvl}e_f = 0. \tag{3.2.2}$$

Equations (3.2.1) and (3.2.2) describe how the flows and voltages of each branch interact with the rest of those in the network. Constituent relationships describe the relationship between the flow and voltage of a specific branch [2]. This is based on the physical properties of the wire, such as Ohm's law. In the general AC transmission case, the power sent from node i toward k is

$$\hat{S}_{i,k} = \hat{E}_i(\hat{E}_i - \hat{E}_k)^*\hat{Y}^*_{i,k}, \tag{3.2.3}$$

where $\hat{Y}^*_{i,k}$ is the admittance from i to k, and $\hat{E}_i$ and $\hat{E}_k$ are the nodal voltages of nodes i and k, respectively. The reference bus voltage is given, typically as $1\angle 0$. $\hat{E}$ is used to indicate the complex nodal voltages, instead of e_n, the generalized voltage notation. Using the S-E graph model of wires,

$$\hat{S}_{L:i,k} = |\hat{E}_i - \hat{E}_k|^2\hat{Y}^*_{i,k}, \tag{3.2.4}$$

and

$$\hat{S}_{f:i,k} = \frac{1}{2}(\hat{E}_i + \hat{E}_k)(\hat{E}_i - \hat{E}_k)^*\hat{Y}^*_{i,k}, \tag{3.2.5}$$

are derived from Equation (3.1.5), (3.1.6) and (3.2.3). Because there is power loss when transmitting power, there are two sets of constituent relationships per wire. The constituent relationships, Equations (3.2.4), (3.2.5), and Kirchhoff's laws, Equations (3.2.1), (3.2.2), form the power flow equations that must be solved to find a feasible operating point for a given electric power system.

3.2.2 Conventional Formulation of Power Flow

Conventional formulation of power flow equations combines the constituent relationship equations with flow conservation equations. By using nodal voltages, the KVL equations need not be explicitly stated, because the nodal voltages always form a spanning tree [1], [2]. Equations (3.2.1) and (3.2.3) are combined to obtain

$$\mathrm{diag}(\hat{E})A_f\mathrm{diag}(Y^*)A_f^T\hat{E}^* = -\hat{S}_n, \tag{3.2.6}$$

where Y is the vector of admittances of each wire, and diag(v) is an operation that creates a diagonal matrix, where the elements of vector v are the diagonals of the matrix. diag(M), when applied to matrix M, creates a diagonal matrix using the diagonal of M. This results in power flow equations, expressed in terms of complex voltages, for systems without PV generators.

The conventional formulation breaks the complex variables into real and imaginary parts, and solves for the voltage in polar form, in terms of voltage angle and magnitude. For node i, the real and reactive power balance equations are

$$P_i = \sum_{k\in\{i,k\}} G_{i,k}(E_i^2 - E_iE_k\cos(\delta_i - \delta_k)) - B_{i,k}E_iE_k\sin(\delta_i - \delta_k), \tag{3.2.7}$$

and

$$Q_i = \sum_{k\in\{i,k\}} -B_{i,k}(E_i^2 - E_iE_k\cos(\delta_i - \delta_k)) - G_{i,k}E_iE_k\sin(\delta_i - \delta_k), \tag{3.2.8}$$

respectively. E without hat notation indicates the magnitude of $\hat{E}$ and δ indicates the angle of $\hat{E}$. P_i and Q_i are the nodal power flows from the nodal branch to node i. They are the real and imaginary parts of $\hat{S}_n$. G and B are the real and imaginary parts of Y, the conductance and susceptance of the wires, respectively.

For the purposes of this work, the polar form voltage equations will not be used except in decoupled approximations of power flow equations. This is because in branch formulation, if $\hat{V}_{i,k} = \hat{E}_i - \hat{E}_k$, then $V_{i,k}\angle\phi_{i,k} = E_i\angle\delta_i - E_k\angle\delta_k$, but is not true that $V_{i,k} = E_i - E_k$ or $\phi_{i,k} = \delta_i - \delta_k$. KVL does not make sense for the AC power flow equation unless expressed in complex voltage.

3.3 Branch-based Power Flow Equations

The purpose of formulating power flow in terms of flow branch voltages, rather than nodal branch voltages, is to determine which physical wire or portion of the electric power system needs to be adjusted based on feasibility of power delivery. This is based on the 2 bus example of AC power flow, where the closed form solution of the voltage can be derived [2]. When the constituent relationships of wires are nonlinear, it is possible to have no solutions to the set of power flow equations based on given loads, generation and wire impedances [14].

3.3.1 Branch-based AC Power Flow in Complex Variables

The power flow equations, defined in Equation (3.2.6), can be reformulated using complex branch voltages, $\hat{V}$, which are analogous to e_f, the voltages across flow branches on the general graph. Therefore, KVL would apply,

$$A_f^T \hat{E} = \hat{V}. \tag{3.3.1}$$

so that $\hat{V}$ can be substituted for $\hat{E}$ in the AC power flow equations, Equation (3.2.6). This results in the set of equations

$$\text{diag}(\hat{E}) A_f \text{diag}(Y^*)\text{diag}(\hat{V}^*) = -\hat{S}_n, \tag{3.3.2}$$

which leaves a nodal voltage vector, $\hat{E}$. If a set of spanning tree branches are selected such that $\hat{V} = \begin{bmatrix} \hat{V}_{tree}^T & \hat{V}_{mesh}^T \end{bmatrix}^T$, then Equation (3.1.12) can be applied,

$$\hat{E} = (A_{f,tree}^T)^{-1} \hat{V}_{tree}, \tag{3.3.3}$$

and Equation (3.3.2) is

$$\operatorname{diag}((A_{f,tree}^T)^{-1}\hat{V}_{tree})A_f\operatorname{diag}(Y^*)\operatorname{diag}(\hat{V}^*) = -\hat{S}_n, \tag{3.3.4}$$

which only specifies power flow in terms of voltages across flow branches of the spanning tree.

However, the number of voltages in $\hat{E}$ is $N-1$, and the number of voltages in $\hat{V}$ is L. Equation (3.3.4) only contains $N-1$ equations, but L variables. The other $L-N+1$ equations come from explicitly expressing KVL,

$$A_{kvl}\hat{V} = 0. \tag{3.3.5}$$

3.3.2 New Optimization-based Formulation of AC Power Flow

In pursuit of distributed algorithms for power flow calculation, an optimization-based formulation was developed [9], [10], [11], [12], based on work in optimization for graphs whose branches represent resource flow [13]. The basic concept is to pose power flow as minimizing the difference between the power flows and losses of the branches, and the powers as a function of the flow branch voltage resulting from constituent relationships. These objectives are subject to Kirchhoff's nodal flow conservation and loop voltage conservation.

The constituent relationships of the AC case, Equations (3.2.4) and (3.2.5), are summed to form the objective function, in the optimization problem,

$$\begin{aligned} \underset{\hat{S},\hat{V}}{\text{minimize}} \quad & \sum_{i,k\in\{1,L\}} |\hat{S}_{f:i,k} - \frac{1}{2}\hat{Y}_{i,k}^*(2\bar{E}_i - \hat{V}_{i,k})\hat{V}_{i,k}^*|^2 + \\ & \qquad\qquad |\hat{S}_{L:i,k} - \hat{Y}_{i,k}^*\hat{V}_{i,k}\hat{V}_{i,k}^*|^2 \\ \text{subject to} \quad & \\ & \begin{bmatrix} A_f & A_L \end{bmatrix} \begin{bmatrix} \hat{S}_f \\ \hat{S}_L \end{bmatrix} = A\hat{S} = -\hat{S}_n, \\ & A_{kvl}\hat{V} = 0. \end{aligned} \tag{3.3.6}$$

The $\bar{E}_i$ term is equivalent to the $\hat{E} = (A_{f,tree}^T)^{-1}\hat{V}_{tree}$ term found in Equation (3.3.4). The overscore indicates that this is to be treated as a given value, rather than a variable, during the iterative algorithm. The nodal voltage is used in the optimization formulation in Equation (3.3.6) because that value can still be obtained locally by computing hardware at the wire itself, whereas $(A_{f,tree}^T)^{-1}\hat{V}_{tree}$ requires the voltages across other flow branches.

The KVL constraints are stated explicitly in Equation (3.3.6), but in practice, the voltages of non-spanning tree flow branches can be computed by updating the nodal voltages during the applied iterative algorithm. In the course of numerical calculation, nodal voltages can be updated through a step-wise process, where each flow branch of the spanning tree updates the nodal voltage of the node farther from the slack bus, using the nodal voltage closer to the slack bus. These nodal designations can be assigned while the spanning tree creation algorithm is running through the network. For example, if depth or breadth first search is used to create a spanning tree starting from the slack bus node, then as it propagates outwards the flow branches can store the identity of the node that it is responsible for updating.

The optimization problem in Equation (3.3.6) is a relaxed version of power flow; while zero is the absolute minimum, there are possible local minima. Lagrangian relaxation is used to solve the optimization problem, by first finding the Lagrangian,

$$\begin{aligned}\mathcal{L}(\hat{S},\hat{V},\lambda) = &\sum_{i,k\in\{1,L\}} (|\hat{S}_{f:i,k} - \frac{1}{2}\hat{Y}_{i,k}^{*}(2\bar{E}_i - \hat{V}_{i,k})\hat{V}_{i,k}^{*}|^2 \\ &+ |\hat{S}_{L:i,k} - \hat{Y}_{i,k}^{*}\hat{V}_{i,k}\hat{V}_{i,k}^{*}|^2) + \lambda^T(A\hat{S} + \hat{S}_n)^{*} + \\ &\lambda^{*T}(A\hat{S} + \hat{S}_n) = 0.\end{aligned} \tag{3.3.7}$$

The KVL constraints are not explicitly expressed, because doing so would require coordination around each linearly independent loop, or mesh, of the network. The spanning tree branches are used to update nodal voltages, allowing the non-spanning tree branches to use the difference of nodal voltages at their ends to update their voltage values.

Unlike traditional Lagrangian relaxation, a λ and λ^* term are used in order to create a real valued Lagrangian. Based on the complex Taylor expansion [31], it can be shown that this results in a complex Hessian without all-zero rows, that result from taking the partial derivative of both the variables and their conjugates.

The optimality conditions are satisfied when the gradient of $\mathcal{L}(\hat{S},\hat{V},\lambda)$ is zero. They form a set of simultaneous equations that need to be solved to find a potential power flow solution. Let F and L be vectors of length L, where each element corresponds to a flow branch connecting nodes i and k, such that

$$L_{i,k}(\hat{S}_{i,k},\hat{V}_{i,k}) = \hat{S}_{L:i,k} - \hat{Y}^*_{i,k}\hat{V}_{i,k}\hat{V}^*_{i,k}, \tag{3.3.8}$$

and

$$F_{i,k}(\hat{S}_{i,k},\hat{V}_{i,k}) = \hat{S}_{f:i,k} - \frac{1}{2}\hat{Y}^*_{i,k}(2\bar{E}_i - \hat{V}_{i,k})\hat{V}^*_{i,k}, \tag{3.3.9}$$

so that the Lagrangian can be simplified to

$$\begin{aligned} &\mathcal{L}(\hat{S},\hat{V},\lambda) = F^T F^* + G^T G^* + \\ &\lambda^T (A\hat{S} + \hat{S}_n)^* + \lambda^{*T}(A\hat{S} + \hat{S}_n) = 0. \end{aligned} \tag{3.3.10}$$

The optimality conditions are

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \hat{S}} = \operatorname{diag}(F^*)\frac{\partial F}{\partial \hat{S}} + \operatorname{diag}(F)\frac{\partial F^*}{\partial \hat{S}} + \operatorname{diag}(G^*)\frac{\partial G}{\partial \hat{S}} \\ + \operatorname{diag}(G)\frac{\partial G^*}{\partial \hat{S}} + A^T(\lambda^*) = 0, \end{aligned} \tag{3.3.11}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \hat{V}} = \operatorname{diag}(F^*)\frac{\partial F}{\partial \hat{V}} + \operatorname{diag}(F)\frac{\partial F^*}{\partial \hat{V}} + \\ \operatorname{diag}(G^*)\frac{\partial G}{\partial \hat{V}} + \operatorname{diag}(G)\frac{\partial G^*}{\partial \hat{V}} = 0, \end{aligned} \tag{3.3.12}$$

and

$$\frac{\partial \mathcal{L}}{\partial \lambda} = A\hat{S}^* + \hat{S}_n = 0. \tag{3.3.13}$$

By setting up the Lagrangian as a real valued function of complex variables, the partial derivatives with respect to $\hat{S}^*$, $\hat{V}^*$, and λ^* are equal to the conjugates of those respective partial derivatives,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \hat{S}^*} &= (\frac{\partial \mathcal{L}}{\partial \hat{S}})^* = 0,\\ \frac{\partial \mathcal{L}}{\partial \hat{V}^*} &= (\frac{\partial \mathcal{L}}{\partial \hat{V}})^* = 0,\\ \frac{\partial \mathcal{L}}{\partial \lambda^*} &= (\frac{\partial \mathcal{L}}{\partial \lambda})^* = 0.\end{aligned} \tag{3.3.14}$$

To solve the optimization problem, a numerical algorithm is used to solve the optimality conditions, Equations (3.3.11), (3.3.12) and (3.3.13). When they are equal to zero, their conjugates are also equal to zero, resulting in the complex gradient being equal to zero.

3.3.3 Decoupled and DC Power Flow Formulations

There are approximations that can be made to the AC power flow equations that make them easier to calculate. One of the most frequently used is assuming decoupling between the real and reactive power balance, and that real power is highly correlated with voltage angle and frequency, while reactive power is correlated with voltage magnitude [2]. Appendix 6.1 includes the derivations of the nodal and branch-based power flow formulations under the decoupled and DC models.

Chapter 4

New Numerical Algorithms for Complex-valued Power Flow

4.1 New Iterative Algorithms to AC Power Flow in Complex-Valued Domain

Numerical algorithms for solving power flow havehad a long history [1]. They have matured over time and form the basis for routinely used computer applications in each utility EMS. In this chapter, we briefly summarize today's numerical algorithms, which are fundamentally inspired using nodal power flow equations. We then introduce possible formulations of numerical algorithms for solving the newly introduced branch-based optimization formulation of the power flow problem.

The most general power flow problem allows for modeling generators as PV elements, except for the so-called "slack" bus, which is modeled as an ideal voltage source whose nodal voltage magnitude and phase angle are given, and used as the reference voltage for the system. Loads are modeled as constant real and reactive power elements whose nodal voltages are unknown. Following the golden rule of power flow formulation, for any type of component, only two out of four key variables (nodal voltage magnitude E, nodal voltage phase angle δ_n, real power P and reactive power Q), can be specified. The other two are the result of computing power flow solution.

In today's power flow formulations, generators (other than slack) have their real power and voltage magnitude specified; this means that once the coupled power flow is solved, the resulting generator variables will be nodal voltage phase angle and generator reactive power. Similarly, the slack bus has nodal voltage magnitude and phase angle specified. Once the power flow is solved, the resulting real and reactive power out of the slack bus can be computed.

This reasoning leads to the complex set of algebraic equations defined by stating real power flow balance equations at all geerator buses, and real and reactive power balance equation at all loads. The resulting coupled AC power flow formulation based on thee constraints takes on the form

$$P_i = \sum_{m \in \{i,m\}} (\frac{E_i^2}{Z_{i,m}} \cos(\zeta_{i,m}) - \frac{E_i E_m}{Z_{i,m}} \cos(\delta_{i,m} - \zeta_{i,m})), \tag{4.1.1}$$

for $i = 1, ...N_l + N_g$, and

$$Q_i = \sum_{m \in \{i,m\}} (\frac{E_i^2}{Z_{i,m}} \sin(\zeta_{i,m}) - \frac{E_i E_m}{Z_{i,m}} \sin(\delta_{i,m} - \zeta_{i,m})), \tag{4.1.2}$$

for $i = 1, ...N_l$, where N_l is the number of load buses, N_g is the number of generator buses, $Z_{i,m}$ and $\zeta_{i,m}$ are the magnitude and phase angle of the impedance across line (i, m), $\delta_{i,m}$ is the phase angle difference across line (i, m), E_i is the nodal voltage magnitude of bus i, and P_i and Q_i are the total real and reactive powers of the lines entering bus i.

The physically implementable solution of Equations (4.1.1) and (4.1.2) must satisfy the following inequality constraints:

$$\begin{aligned} &P_{Gi}^{min} < P_{Gi} < P_{Gi}^{max}, \\ &Q_{Gi}^{min} < Q_{Gi} < Q_{Gi}^{max}, \\ &E_{Gi}^{min} < E_{Gi} < E_{Gi}^{max}, \\ &E_{Li}^{min} < E_{Li} < E_{Li}^{max}, \\ &P_{i,m} \leq P_{i,m}^{max}, \end{aligned} \tag{4.1.3}$$

where P_{Gi} is the real power of each generator bus, Q_{Gi} is the reactive power of each generator bus, E_{Gi} is the voltage of each generator bus, E_{Li} is the voltage of each load bus, and $P_{i,m}$ is the power flowing through line (i, m).

The constraints of Equation (4.1.3) are usually checked after the power flow calculation is done. Only optimization takes account explicitly for thee inequality constraints.

The routinely used numerical algorithms for solving these coplex algebraic equations, Newton-(Raphson) and Jacobi methods, are briefly reviewed in Appendix A.1 of this thesis. Notably, all existing numerical methods for power flow solving are formulaed in real-valued domain.

In this thesis we consider a particular case of power flow problem in which a general meshed network has only one generator (slack bus), and all loads are modeled as PQ buses. In other words, PV generators are not considered in this thesis.

It is shown in what follows that for this particular class of power systems, it becomes possible to formulate numerical algorithms in complex-valued domain. This formulation of numerical algorithms in a complex-valued domain represents one of the contributions to this thesis, as introduced next.

4.1.1 New Nodal-based AC Power Flow in Complex-Valued Domain

The Newton and Jacobi methods are used to iteratively solve power flow equations. Conventionally, Newton method is used to solve power flow equations in terms of nodal voltage. However, it is also possible to apply these algorithms to the branch voltage formulations as well. Newton and Jacobi method and their convergence conditions are described in Appendix A.1.

Newton Method of Solving Nodal-based AC Power Flow in Complex Valued Domain

Recall equation (3.2.6), which models the AC power flow equations, written in terms of complex voltages, for the case of power grids without PV generators. This can be reexpressed as the function,

$$f(\hat{E}) = \mathrm{diag}(\hat{E}) A_f \mathrm{diag}(Y^*) A_f^T \hat{E}^* + \hat{S}_n = 0, \tag{4.1.4}$$

Then, partial derivatives with respect to $\hat{E}$ and its conjugates can be calculated,

$$\frac{\partial f(\hat{E})}{\partial \hat{E}} = J(\hat{E}) = \mathrm{diag}(A_f \mathrm{diag}(Y^*) A_f^T \hat{E}^*), \tag{4.1.5}$$

and

$$\frac{\partial f(\hat{E})}{\partial \hat{E}^*} = J_c(\hat{E}) = \text{diag}(\hat{E})A_f\text{diag}(Y^*)A_f^T. \tag{4.1.6}$$

By using the complex Taylor expansion, the expression for the Newton step, $\Delta\hat{E}^{(k)}$, is

$$J\Delta\hat{E}^{(k)} + J_c\Delta(\hat{E}^{(k)})^* = f(\hat{E}^{(k)}), \tag{4.1.7}$$

where k is the iterative step number. The $\Delta(\hat{E}^{(k)})^*$ term can be eliminated by adding the equation,

$$-J_c(J^*)^{-1}(J_c^*\Delta\hat{E}^{(k)} + J^*\Delta(\hat{E}^{(k)})^* = f^*(\hat{E}^{(k)})), \tag{4.1.8}$$

to Equation (4.1.7). Equation (4.1.8) is the conjugate of Equation (4.1.7) multiplied by $-J_c(J^*)^{-1}$. After the equations are added, what is left is

$$(J - J_c(J^*)^{-1}J_c^*)\Delta\hat{E}^{(k)} = f(\hat{E}^{(k)}) - J_c(J^*)^{-1}f^*(\hat{E}^{(k)}). \tag{4.1.9}$$

Therefore, the new expression for the Newton step, calculated in complex-valued domain, is

$$\Delta\hat{E}^{(k)} = (J - J_c(J^*)^{-1}J_c^*)^{-1}(f(\hat{E}^{(k)}) - J_c(J^*)^{-1}f^*(\hat{E}^{(k)})). \tag{4.1.10}$$

The complex valued nodal voltages, $\hat{E}$, are updated as

$$\hat{E}^{(k+1)} = \hat{E}^{(k)} + \Delta\hat{E}^{(k)}. \tag{4.1.11}$$

New Jacobi Method of Solving Nodal-based AC Power Flow

For a given node i, the power balance is

$$f_i(\hat{E}) = \hat{S}_i - \hat{E}_i \sum_{m\in\{i,m\}} (\hat{E}_i - \hat{E}_k)^* Y_{i,k}^* = 0. \tag{4.1.12}$$

In order to calculate $\hat{E}$, the Jacobi method iterate is calculated using a solved term $(\hat{E}_i^{(k:s)})$, by deriving a closed form expression using Equation (A.1.6). First, an expression can be found in terms of $\hat{E}_i^{(k:s)}$ and $|\hat{E}_i^{(k:s)}|^2$.

$$|\hat{E}_i^{(k:s)}|^2(\sum_{m\in\{i,m\}} Y^*_{i,m}) - \hat{E}_i^{(k:s)}(\sum_{m\in\{i,m\}} Y^*_{i,m}\hat{E}_m^{(k)}) - \hat{S}_n = 0 \tag{4.1.13}$$

For the complex variable problem of the following form,

$$a_1\hat{z}\hat{z}^* + a_2\hat{z} + a_3 = 0, \tag{4.1.14}$$

$\hat{z}^*$ in terms of $\hat{z}$ is

$$\hat{z}^* = \frac{1}{a_1 a_2^*}(a_1^* a_2\hat{z} + a_1^* a_3 - a_1 a_3^*). \tag{4.1.15}$$

By substituting $\hat{z}^*$ back into Equation (4.1.14), a quadratic complex polynomial equation in terms of $\hat{z}$ is found,

$$a\hat{z}^2 + b\hat{z} + c = 0, \tag{4.1.16}$$

where

$$\begin{aligned} a &= a_1^* a_2, \\ b &= (-a_1 a_3^* + a_1^* a_3 + a_2 a_2^*), \\ c &= a_2^* a_3. \end{aligned} \tag{4.1.17}$$

The quadratic equation gives a solution for $\hat{z}$,

$$\hat{z} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{4.1.18}$$

Therefore, to solve Equation (4.1.13), let

$$\begin{aligned} a_1^{(k)} &= \begin{bmatrix} \vdots \\ \sum_{m\in\{i,m\}} Y^*_{i,m} \\ \vdots \end{bmatrix} = \text{vect}(\text{diag}(A_f Y^* A_f^T)), \\ a_2^{(k)} &= \begin{bmatrix} \vdots \\ \sum_{m\in\{i,m\}} Y^*_{i,m}\hat{E}_m^{(k)} \\ \vdots \end{bmatrix} = \\ &(A_f Y^* A_f^T - \text{diag}(A_f Y^* A_f^T))(\hat{E}^{(k)})^*, \\ a_3^{(k)} &= -\,\hat{S}_n, \end{aligned} \tag{4.1.19}$$

where the function vect(M) creates a vector using the elements of diagonal matrix M. Then, applying Equation (4.1.17) and (4.1.18) gives a closed form solution for $\hat{E}^{(k:s)}$,

$$\hat{E}^{(k:s)} = \frac{-b^{(k)} \pm \sqrt{(b^{(k)})^2 - 4a^{(k)}c^{(k)}}}{2a^{(k)}}. \tag{4.1.20}$$

Then, the Jacobi iterative update is

$$\hat{E}^{(k+1)} = (1-\omega)\hat{E}^{(k)} + \omega\hat{E}^{(k:s)}. \tag{4.1.21}$$

4.1.2 New Branch-based AC Power Flow in Complex-Valued Domain

New Newton Method for Branch-based AC Power Flow

Equation (3.3.2) is AC power flow formulated in terms of branch voltages. This is rearranged into $f(\hat{V}) = 0$, where

$$f(\hat{V}) = \begin{bmatrix} \text{diag}(\bar{E})A_f\text{diag}(Y^*)(\hat{V}^*) + \hat{S}_n \\ A_{kvl}\hat{V}^* \end{bmatrix} = 0. \tag{4.1.22}$$

$\hat{V}$ can be left in terms of nodal voltage, instead of flow branch voltages, and Newton method still converges. At each iteration, the nodal voltages, $\bar{E}$, can be calculated as an algebraic result of the voltages across a flow branch spanning tree, as in Equation (3.3.3). $A_{kvl}\hat{V}^* = 0$ is equivalent to $A_{kvl}\hat{V} = 0$, allowing all of Equation (4.1.22) to be in terms of the conjugate of $\hat{V}$. Therefore, the Jacobian with respect to $\hat{V}$, $J(\hat{V})$ is equal to zero. The Jacobian with respect to $J_c(\hat{V}^*)$ is

$$J_c(\hat{V}) = \begin{bmatrix} \text{diag}(\bar{E})A_f\text{diag}(Y^*) \\ \\ A_{kvl} \end{bmatrix}. \tag{4.1.23}$$

Because $J(\hat{V}) = 0$, Equation (A.1.3) becomes

$$J_c((\hat{V}^{(k)})^*)\Delta(\hat{V}^{(k)})^* = f((\hat{V}^{(k)})^*), \tag{4.1.24}$$

where k is the iterative step. Then the Newton step is

$$\Delta\hat{V}^{(k)} = [J_c^{-1}((\hat{V}^{(k)})^*)f((\hat{V}^{(k)})^*)]^*. \tag{4.1.25}$$

The iterative update is then

$$\hat{V}^{(k+1)} = \hat{V}^{(k)} + \Delta\hat{V}^{(k)}. \tag{4.1.26}$$

New Jacobi Method for Branch-based AC Power Flow

Equation (4.1.22) can also be solved using Jacobi method. It can be expressed as

$$f(\hat{V}) = \begin{bmatrix} \text{diag}(\bar{E})A_f\text{diag}(Y^*) \\ \\ A_{kvl} \end{bmatrix} (\hat{V}^*) + \begin{bmatrix} \hat{S}_n \\ \\ 0 \end{bmatrix} = 0. \tag{4.1.27}$$

The solved value of branch flow voltages , $\hat{V}^{(k:s)}$, is

$$\hat{V}^{(k:s)} = -D_J^{-1}(C_J(V\hat{^{(k)}})^* + \begin{bmatrix} \hat{S}_n \\ \\ 0 \end{bmatrix}), \tag{4.1.28}$$

where

$$D_J = \text{diag}(\begin{bmatrix} \text{diag}(\bar{E}^{(k)})A_f\text{diag}(Y^*) \\ \\ A_{kvl} \end{bmatrix}), \tag{4.1.29}$$

and

$$C_J = \begin{bmatrix} \mathrm{diag}(\bar{E}^{(k)})A_f\mathrm{diag}(Y^*) \\ A_{kvl} \end{bmatrix} - \mathrm{diag}(\begin{bmatrix} \mathrm{diag}(\bar{E})A_f\mathrm{diag}(Y^*) \\ A_{kvl} \end{bmatrix}), \tag{4.1.30}$$

so that D_J and C_J are the diagonal and off-diagonal terms of $J_c(\hat{V})$.

As with the Newton method, $\bar{E}^{(k)}$ can be calculated as a result of Equation (3.3.3) at each iteration. Then, each Jacobi iterate is

$$\hat{V}^{(k+1)} = (1-\omega)\hat{V}^{(k)} + \omega\hat{V}^{(k:s)}. \tag{4.1.31}$$

4.1.3 AC Power Flow Using New Optimization-based Formulation

Newton Method for Optimization-based AC Power Flow

The optimality conditions of the optimization-based power flow formulation, Equations (3.3.11), (3.3.12), and (3.3.13) , are a set of simultaneous equations that can be solved numerically. The optimality conditions are

$$f(\hat{S},\hat{V},\lambda) = \begin{bmatrix} \partial\mathcal{L}/\partial\hat{S} \\ \partial\mathcal{L}/\partial\lambda \\ \partial\mathcal{L}/\partial\hat{V} \end{bmatrix} = 0. \tag{4.1.32}$$

The Jacobian of $f(\hat{S},\hat{V},\lambda)$, which is equal to the Hessian of the Lagrangian, is

$$H(\hat{S},\hat{V},\lambda) = \nabla f(\hat{S},\hat{V},\lambda) = \begin{bmatrix} \frac{\partial^2\mathcal{L}}{\partial\hat{S}^2} & \frac{\partial^2\mathcal{L}}{\partial\hat{S}\partial\lambda} & \frac{\partial^2\mathcal{L}}{\partial\hat{S}\partial\hat{V}} \\ \frac{\partial^2\mathcal{L}}{\partial\lambda\partial\hat{S}} & \frac{\partial^2\mathcal{L}}{\partial\lambda^2} & \frac{\partial^2\mathcal{L}}{\partial\lambda\partial\hat{V}} \\ \frac{\partial^2\mathcal{L}}{\partial\hat{V}\partial\hat{S}} & \frac{\partial^2\mathcal{L}}{\partial\hat{V}\partial\lambda} & \frac{\partial^2\mathcal{L}}{\partial\hat{V}^2} \end{bmatrix}. \tag{4.1.33}$$

When deriving the block submatrix components of the Hessian, some are found to have certain structures, based on the forms of the constituent relationships F and L from Equations (3.3.8) and (3.3.9). For example, the partial derivatives with respect to complex power flows are

$$\begin{aligned}\frac{\partial L}{\partial \hat{S}} = \frac{\partial L^*}{\partial \hat{S}^*} = \frac{\partial F}{\partial \hat{S}} = \frac{\partial F^*}{\partial \hat{S}^*} = \overrightarrow{1},\\ \frac{\partial L^*}{\partial \hat{S}} = \frac{\partial L}{\partial \hat{S}^*} = \frac{\partial F^*}{\partial \hat{S}} = \frac{\partial F}{\partial \hat{S}^*} = \overrightarrow{0},\end{aligned} \tag{4.1.34}$$

where $\overrightarrow{m}$ of a scalar, m, represents a column vector whose elements are all m.

Because only spanning tree flow branches are updated, partial derivatives with respect to $\hat{V}_{mesh}$ are not considered, and only the $\hat{V}_{tree}$ portion of $\hat{V}$ is included in the Hessian. The rows and columns of the Hessian are reduced to not include $\hat{V}_{mesh}$. The partial derivatives with respect to complex spanning tree voltages have symmetry with respect to their conjugate counterparts,

$$\begin{aligned}\frac{\partial L}{\partial \hat{V}_{tree}} &= (\frac{\partial L^*}{\partial \hat{V}^*_{tree}})^* = -\text{diag}(Y^*_{tree})\hat{V}^*_{tree}\\ \frac{\partial L}{\partial \hat{V}^*_{tree}} &= (\frac{\partial L^*}{\partial \hat{V}_{tree}})^* = -\text{diag}(Y^*_{tree})\hat{V}_{tree}\\ \frac{\partial F}{\partial \hat{V}_{tree}} &= (\frac{\partial F^*}{\partial \hat{V}^*_{tree}})^* = \frac{1}{2}\text{diag}(Y^*_{tree})\hat{V}^*_{tree}\\ \frac{\partial F}{\partial \hat{V}_{tree}} &= (\frac{\partial F^*}{\partial \hat{V}^*_{tree}})^* = \frac{1}{2}\text{diag}(Y^*_{tree})(2\bar{E} - \hat{V}^*_{tree})\end{aligned} \tag{4.1.35}$$

where $\bar{E}$ are the nodal voltages at the beginning of each flow branch, corresponding to branch order in $\hat{V}_{tree}$. The Hessian with respect to the variables $\hat{S}, \hat{V}, \lambda$ has the structure,

$$H(\hat{S}, \hat{V}, \lambda) = \begin{bmatrix} 0 & 0 & \frac{\partial(F^*+L^*)}{\partial \hat{V}} \\ 0 & 0 & 0 \\ (\frac{\partial(F^*+L^*)}{\partial \hat{V}})^T & 0 & \frac{\partial^2(FF^*+LL^*)}{\partial \hat{V}^2} \end{bmatrix}. \tag{4.1.36}$$

However, as with the complex Newton method, the partial derivatives with respect to the conjugate variables need to be considered. The full complex Hessian is

$$H_{full} = \nabla \begin{bmatrix} f(\hat{S}, \hat{V}, \lambda) \\ f(\hat{S}^*, \hat{V}^*, \lambda^*) \end{bmatrix} = \begin{bmatrix} H & H_{c1} \\ H_{c2} & H_d \end{bmatrix}, \tag{4.1.37}$$

where

$f(\hat{S}^*, \hat{V}^*, \lambda^*)$ are the partial derivatives of the Lagrangian with respect to the conjugate variables, and $f(\hat{S}^*, \hat{V}^*, \lambda^*) = f(\hat{S}, \hat{V}, \lambda)^*$, because the Lagrangian is a real function, formed by products and sums of conjugates. The other block submatrices of Equation (4.1.37) are

$$Hc(\hat{S},\hat{V},\lambda) = \begin{bmatrix} I & A^T & \frac{\partial(F^*+L^*)}{\partial\hat{V}^*} \\ A & 0 & 0 \\ (\frac{\partial(F+L)}{\partial\hat{V}})^T & 0 & \frac{\partial^2(FF^*+LL*)}{\partial\hat{V}\hat{V}^*} \end{bmatrix}, \tag{4.1.38}$$

$$H_{c2}(\hat{S},\hat{V},\lambda) = \begin{bmatrix} I & A^T & \frac{\partial(F+L)}{\partial\hat{V}} \\ A & 0 & 0 \\ (\frac{\partial(F^*+L^*)}{\partial\hat{V}^*})^T & 0 & \frac{\partial^2(FF^*+LL*)}{\partial\hat{V}\hat{V}^*} \end{bmatrix}, \tag{4.1.39}$$

$$H_d(\hat{S},\hat{V},\lambda) = \begin{bmatrix} 0 & 0 & \frac{\partial(F+L)}{\partial\hat{V}^*} \\ 0 & 0 & 0 \\ (\frac{\partial(F+L)}{\partial\hat{V}^*})^T & 0 & \frac{\partial^2(FF^*+LL*)}{\partial\hat{V}^{*2}} \end{bmatrix}, \tag{4.1.40}$$

It should be noted that the Lagrangian was constructed in Equation (3.3.7) so that the term $\lambda^{*T}(A\hat{S}+\hat{S}_n)$ has the conjugate of the Lagrange multiplier associated to it. This creates the submatrix $\begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix}$ in the H_{c1} and H_{c2} blocks, and removes A and A^T terms from the H and H_d blocks. This rearrangement is necessary to carry out the distributed method for solving the optimization-based formulation. To carry out the standard Newton method, the Newton step at iteration k is

$$\begin{aligned} &\Delta \begin{bmatrix} \hat{S}^{T(k)} & \lambda^{T(k)} & \hat{V}^{T(k)} & (\hat{S}^{T(k)})^* & (\lambda^{T(k)})^* & (\hat{V}^{T(k)})^* \end{bmatrix}^T \\ &= -H_{full}^{-1}(\hat{S}^{(k)},\hat{V}^{(k)},\lambda^{(k)}) \begin{bmatrix} f(\hat{S}^{(k)},\hat{V}^{(k)},\lambda^{(k)}) \\ f(\hat{S}^{*(k)},\hat{V}^{8(k)},\lambda^{*(k)}) \end{bmatrix}. \end{aligned} \tag{4.1.41}$$

This can also be expressed as

$$\begin{bmatrix} H & H_{c1} \\ H_{c2} & H_d \end{bmatrix} \Delta \begin{bmatrix} \hat{S}^{(k)} \\ \lambda^{(k)} \\ \hat{V}^{(k)} \\ (\hat{S}^{(k)})^* \\ (\lambda^{(k)})^* \\ (\hat{V}^{(k)})^* \end{bmatrix} \begin{bmatrix} f(\hat{S}^{(k)}, \hat{V}^{(k)}, \lambda^{(k)}) \\ f(\hat{S}^{*(k)}, \hat{V}^{8(k)}, \lambda^{*(k)}) \end{bmatrix}. \tag{4.1.42}$$

By subtracting the conjugate of the lower rows of Equation (4.1.42), similar to Equation (4.1.9), the conjugate Newton steps can be cancelled out and the Newton steps are

$$\begin{aligned} &\Delta \begin{bmatrix} \hat{S}^{T(k)} & \lambda^{T(k)} & \hat{V}^{T(k)} \end{bmatrix}^T \\ &= -(H - H_{c1}{H_{c2}}^{-1} H_d)^{-1} (f - H_{c1}{H_{c2}}^{-1} f^*). \end{aligned} \tag{4.1.43}$$

Other algebraic combinations are possible to solve for $\Delta \begin{bmatrix} \hat{S}^{T(k)} & \lambda^{T(k)} & \hat{V}^{T(k)} \end{bmatrix}^T$, but the matrix inversions in Equation (4.1.43) need to be considered. In the distributed case, decompositions are used that approximate the off-diagonal terms of each submatrix of H_{full} as zeros. Therefore, only H_{c1} and H_{c2} would remain invertible.

The Newton update is then

$$\begin{bmatrix} \hat{S}^{(k+1)} \\ \lambda^{(k+1)} \\ \hat{V}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \hat{S}^{(k)} \\ \lambda^{(k)} \\ \hat{V}^{(k)} \end{bmatrix} + \Delta \begin{bmatrix} \hat{S}^{(k)} \\ \lambda^{(k)} \\ \hat{V}^{(k)} \end{bmatrix}. \tag{4.1.44}$$

New Jacobi method for Solving Optimization-based AC Power Flow in Complex-Valued Domain

Jacobi method applied to the optimization-based power flow formulation will never meet the strict diagonal dominance criteria for convergence. This is a result of the zeros along the diagonal of the Hessian, which is the Jacobian of the optimality condition equations. These zeros correspond to the Lagrange multiplier, λ. In the linear case of DC power flow, these diagonal zeros prevent the

Jacobi method from being used, because the diagonal of the Jacobi matrix cannot be inverted to calculate the next iteration of Jacobi method. A method, called distributed Newton method, will be used to solve the optimization-based power flow formulation in a distributed way.

New Distributed Newton Method for Solving Optimization-based AC Power Flow in Complex-Valued Domain

The distributed Newton method, demonstrated by [13] for optimization on networks that obey conservation of flow, is adapted for networks that also have voltage across branches in [9], [10], [11] and [12]. The core idea is to decompose the Hessian matrix, such that each branch or node can update its own variables with only information from itself or its directly connected neighbors.

Consider the four submatrices of the complex Hessian, H_{full}. A decomposition based on approximate Newton directions can be used [36], which assumes that the coupling between groups of variables, in this case power flow and voltage, is weak. Therefore, the off-diagonal terms of H, H_{c1}, H_{c2}, and H_d are neglected. H and H_d are only left with their lower right submatrices, which are $(N-1)\times(N-1)$ diagonal matrices. H_{c1} and H_{c2} are left with their lower right submatrices, and their upper left submatrices are both $\begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix}$.

After using decomposition through approximate Newton directions, Equation (4.1.43) becomes two separate subproblems,

$$\begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix} \Delta \begin{bmatrix} \hat{S}^{(k)} \\ \lambda^{(k)} \end{bmatrix} = - \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \hat{S}} - \frac{\partial \mathcal{L}}{\partial \hat{S}^*} \\ \frac{\partial \mathcal{L}}{\partial \lambda} - \frac{\partial \mathcal{L}}{\partial \lambda^*} \end{bmatrix}, \tag{4.1.45}$$

and

$$\begin{aligned} &\Delta \hat{V}^{(k)} = \\ &(D - D_{c1} {D_{c2}}^{-1} D_d)^{-1} \left(\frac{\partial \mathcal{L}}{\partial \hat{V}} - D_{c1} {D_{c2}}^{-1} \frac{\partial \mathcal{L}}{\partial \hat{V}^*}\right). \end{aligned} \tag{4.1.46}$$

where

$$\begin{aligned} D &= \frac{\partial(FF^* + LL^*)}{\partial \hat{V}^2}, \\ D_{c1} = D_{c2} &= \frac{\partial(FF^* + LL^*)}{\partial \hat{V}\hat{V}^*}, \\ D_d &= \frac{\partial(FF^* + LL^*)}{\partial \hat{V}^{*2}}. \end{aligned} \tag{4.1.47}$$

Because the elements of vectors F and L are only functions of a specific flow branch's power flow, loss, and voltage across the branch, the matrices D, D_{c1}, D_{c2} and D_d are diagonal.

It is shown in [13] that Equation (4.1.45) can be solved without inverting $\begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix}$. First, the Newton step for $\lambda^{(k)}$ is found,

$$\Delta\lambda^{(k)} = (AA^T)^{-1}(\frac{\partial\mathcal{L}}{\partial\lambda} - \frac{\partial\mathcal{L}}{\partial\lambda^*} - A(\frac{\partial\mathcal{L}}{\partial\hat{S}} - \frac{\partial\mathcal{L}}{\partial\hat{S}^*})). \tag{4.1.48}$$

Inverting the matrix would make Equation (4.1.48) require global information from the network to complete, but the He's Homotopy method [37] is used to approximate the inverse of AA^T with a zeroth order approximation, the diagonal of AA^T. Knowing that A is an incidence matrix that accounts for both flow and loss branches, the diagonal of AA^T, the Laplacian, can be deduced using the degree of connectivity at each node. Therefore, Equation (4.1.48) can be approximated using only local information. Then, $\Delta\lambda^{(k)}$ is used to calculate $\Delta\hat{S}^{(k)}$,

$$\Delta\hat{S}^{(k)} = -((\frac{\partial\mathcal{L}}{\partial\hat{S}} - \frac{\partial\mathcal{L}}{\partial\hat{S}^*}) + A^T\Delta\lambda^{(k)}). \tag{4.1.49}$$

If Equations (4.1.48) and (4.1.49) are examined row by row, and if inverse of AA^T is approximated using a diagonal matrix, then it is found that each element of $\Delta\lambda^{(k)}$ corresponds to a node, and only requires information from its neighboring branches and itself, and the elements of $\Delta\hat{S}^{(k)}$ each correspond to a flow branch, and only requires information from its neighboring nodes and itself.

Equation (4.1.48) can be expressed in terms of each individual node. For node i, $\Delta\hat{\lambda}_i$ is calculated as

$$\Delta\hat{\lambda}_i = \frac{5}{4}(M_i)(\sum_{m=1}^{M_i}(A_f(i,m)\hat{S}_{f:i,m} + \frac{1}{2}\hat{S}_{L:i,m}) - A_f(i,(i,m))F_{i,m} - \frac{1}{2}L_{i,m}) - \hat{S}_{n:i}), \qquad (4.1.50)$$

where M_i is the number of lines connected to bus i, b_i is the complex power injection at bus i, and $A_f(i,m)$ is -1 if the branch flow points into node i, and 1 if the branch flow points out of node i. $\Delta\hat{\lambda}_i$ is then used to calculate the iterative update at step k, of $\hat{\lambda}_i$:

$$\hat{\lambda}_i^{(k+1)} = \hat{\lambda}_i^{(k)} + \Delta\hat{\lambda}_i^{(k)}. \qquad (4.1.51)$$

Each flow branch then transmits the value of $\Delta\hat{\lambda}_i^{(k)}$ and $\hat{\lambda}_i^{(k+1)}$ to the wires connected to bus i. Each line updates its own flow and loss as

$$\Delta\hat{S}_{f:i,m}^{(k)} = F_{i,m}^{(k)} + \Delta\hat{\lambda}_i^{(k)} - \Delta\hat{\lambda}_m^{(k)}, \qquad (4.1.52)$$

and

$$\Delta\hat{S}_{L:i,m}^{(k)} = L_{i,m}^{(k)} + \Delta\hat{\lambda}_i^{(k)} + \Delta\hat{\lambda}_m^{(k)}. \qquad (4.1.53)$$

Using $\Delta\hat{S}^{(k)}$, $\Delta\lambda^{(k)}$ and $\Delta\hat{V}^{(k)}$, the iterative algorithm updates,

$$\begin{bmatrix} \hat{S}^{(k+1)} \\ \lambda^{(k+1)} \\ \hat{V}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \hat{S}^{(k)} \\ \lambda^{(k)} \\ \hat{V}^{(k)} \end{bmatrix} + \omega\Delta \begin{bmatrix} \hat{S}^{(k)} \\ \lambda^{(k)} \\ \hat{V}^{(k)} \end{bmatrix}, \qquad (4.1.54)$$

where $0 < \omega \leq 1$ is a damping factor to aid convergence.

Figure 4.1 shows a flowchart of the iterative distributed Newton method for calculating complex power flow. The threshold variable, ϵ, is used to determine when the Newton step is sufficiently small, resulting in convergence of the algorithm. It is also possible to use the value of the optimality conditions, $f(\hat{S}, \hat{V}, \lambda)$, as a threshold condition, because they should be equal to zero at convergence.

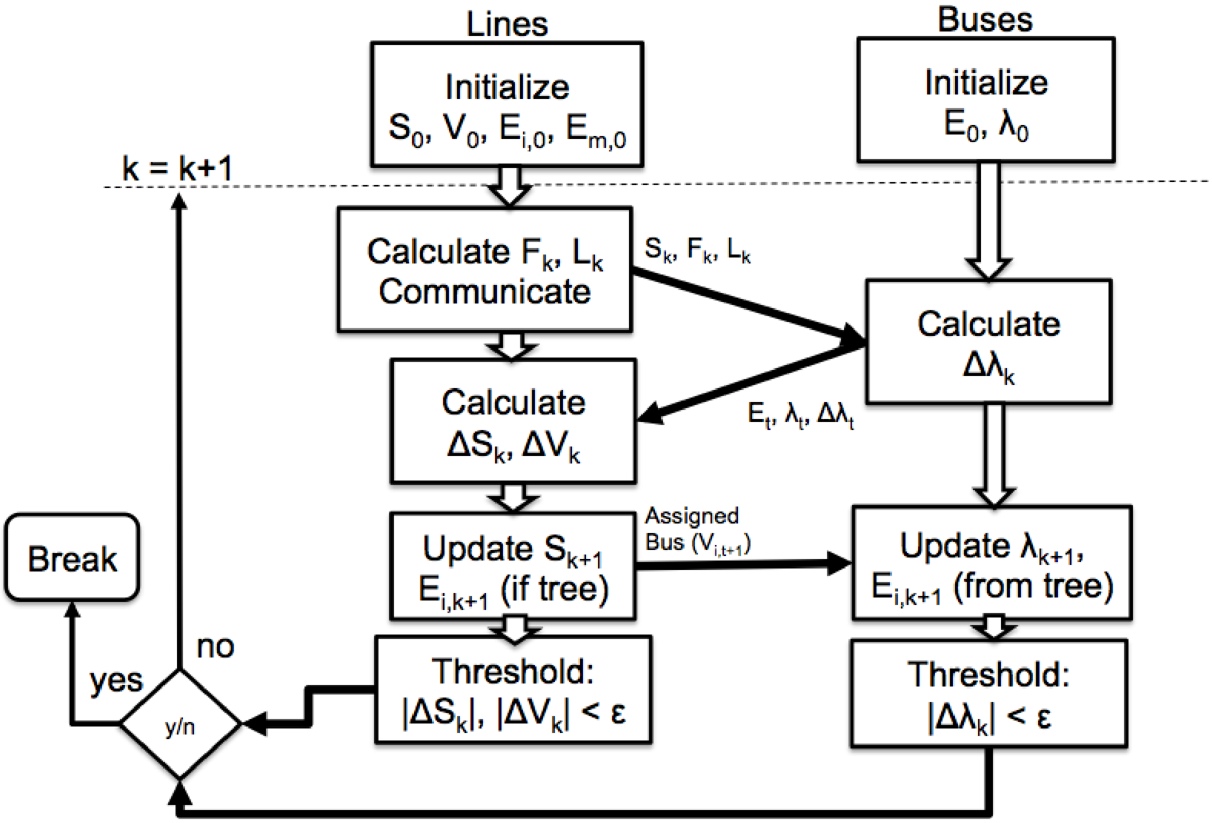


Figure 4.1: A flowchart of the information exchange required to execute the distributed Newton method for complex power flow.

Chapter 5

Ensuring Feasible Power Delivery

5.1 Feasible Power Delivery Through Wires of the Network

5.1.1 Feasible Power Delivery On The 2 Bus Network

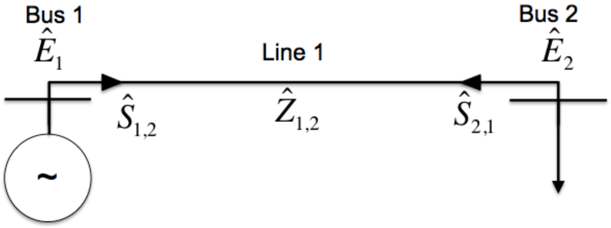


Figure 5.1: The 2 bus network is the base example for deriving the conditions of feasible power delivery.

The basic premise of feasible power delivery can be demonstrated in a 2 bus network, with a generator at one end and a constant real and reactive power load at the other. Figure 5.1 shows the system diagram of the 2 bus network. The voltages of buses 1 and 2 are $\hat{E}_1$ and $\hat{E}_2$, respectively. The connecting transmission line has impedance $\hat{Z}_{1,2}$. The power flow from buses 1 and 2 towards the other are

$$\begin{aligned} \hat{S}_{1,2} &= \hat{Y}^*_{1,2}\hat{E}_1(\hat{E}_1 - \hat{E}_2)^*, \\ \hat{S}_{2,1} &= \hat{Y}^*_{1,2}\hat{E}_2(\hat{E}_2 - \hat{E}_1)^*, \end{aligned} \tag{5.1.1}$$

where $\hat{S}_{1,2}$ is the power flow at the sending end, bus 1, towards bus 2, and $\hat{S}_{2,1}$ is the power flow at the sending end, bus 2, towards bus 1, and $\hat{Y}_{1,2} = \hat{Z}_{1,2}^{-1}$ [1]. $\hat{S}_{1,2}$ is not equal to $-\hat{S}_{2,1}$ because some power is lost through the transmission line. Transmission losses are accounted for using the S-E graph model, a pi circuit model where half of the line loss leaves the system through the legs of the pi circuit [8].

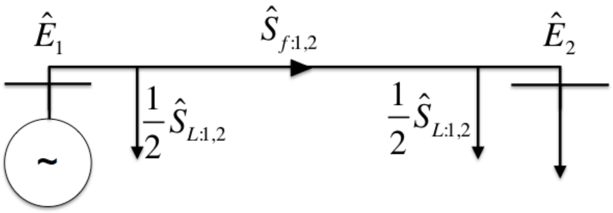


Figure 5.2: The transmission line of the 2 bus system represented as an S-E graph pi model.

Figure 5.2 shows the 2 bus system with the S-E graph model of the transmission line. The transmitted power, $\hat{S}_{f:1,2}$, is the amount of power that is delivered through the line. The power loss, $\hat{S}_{l:1,2}$, is removed from the network through the legs of the pi graph to ground. The constituent relationships of the transmitted power and lost power are

$$\hat{S}_{f:1,2} = \frac{1}{2}\hat{Y}_{1,2}^{*}(\hat{E}_1 + \hat{E}_2)(\hat{E}_1 - \hat{E}_2)^{*} = \frac{1}{2}\hat{Y}_{1,2}^{*}(2 * \hat{E}_1 - \hat{V}_{1,2})\hat{V}_{1,2}^{*}, \tag{5.1.2}$$

and

$$\hat{S}_{l:1,2} = \hat{Y}_{1,2}^{*}(\hat{E}_1 - \hat{E}_2)(\hat{E}_1 - \hat{E}_2)^{*} = \hat{Y}_{1,2}^{*}\hat{V}_{1,2}\hat{V}_{1,2}^{*}, \tag{5.1.3}$$

where $\hat{V}_{1,2}$ is the voltage difference across the line, equal to $(\hat{E}_1 - \hat{E}_2)$. Consider the case where $\hat{E}_1$ and the load at bus 2, $-\hat{S}_{2,1}$ are specified. The power consumed by the load is balanced by $\hat{S}_{f:1,2}$ and $\hat{S}_{l:1,2}$ at bus 2, resulting in

$$-\hat{S}_{2,1} = \hat{S}_{f:1,2} - \frac{1}{2}\hat{S}_{l:1,2} = \hat{Y}_{1,2}^{*}(\hat{E}_1\hat{V}_{1,2}^{*} - |\hat{V}_{1,2}|^2). \tag{5.1.4}$$

$\hat{V}_{1,2}$ can be solved for in terms of $\hat{E}_1$, $\hat{Z}_{1,2}$ and $\hat{S}_{1,2}$. First, multiply both sides of Equation (5.1.4) by $\hat{Z}_{1,2}^{*}$ and separate the real and imaginary parts of $\hat{Z}_{1,2}^{*}\hat{S}_{2,1}$, so that

$$\mathrm{Re}\{\hat{Z}_{1,2}^{*}\hat{S}_{2,1}\} = E_1 V_{1,2}\cos(\delta_1 - \theta_{1,2}) - V_{1,2}^2, \tag{5.1.5}$$

and

$$\mathrm{Im}\{\hat{Z}_{1,2}^{*}\hat{S}_{2,1}\} = E_1 V_{1,2}\sin(\delta_1 - \theta_{1,2}), \tag{5.1.6}$$

where E_1 and $V_{1,2}$ are the magnitudes of the complex bus voltage at bus 1, and complex voltage across line 1,2, respectively. δ_1 and $\theta_{1,2}$ are the phase angles of the complex bus voltage at bus 1, and complex voltage across line 1,2, respectively. The sine and cosine terms can be squared, and then added, to remove the phase angle terms,

$$\begin{aligned} 4E_1^2V_{1,2}^2(\cos^2(\delta_1 - \theta_{1,2}) + \sin^2(\delta_1 - \theta_{1,2})) &= 4E_1^2V_{1,2}^2 \\ &= \mathrm{Im}\{\hat{Z}_{1,2}^{*}\hat{S}_{2,1}\}^2 + (\mathrm{Re}\{\hat{Z}_{1,2}^{*}\hat{S}_{2,1}\} + V_{1,2}^2)^2, \end{aligned} \tag{5.1.7}$$

leaving Equation (5.1.7) in terms of the voltage magnitudes, and complex power at the sending end of bus 2. This equation is a quadratic function with respect to $V_{1,2}$,

$$\begin{aligned} 0 =& V_{1,2}^4 + 2\,\mathrm{Re}\{\hat{Z}_{1,2}^{*}\hat{S}_{2,1}\}V_{1,2}^2 + \mathrm{Re}\{\hat{Z}_{1,2}^{*}\hat{S}_{2,1}\}^2 \\ &+ \mathrm{Im}\{\hat{Z}_{1,2}^{*}\hat{S}_{2,1}\}^2 - 4E_1^2V_{1,2}^2, \end{aligned} \tag{5.1.8}$$

which can be solved through the quadratic equation,

$$V_{1,2}^2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2ac} \tag{5.1.9}$$

where

$$\begin{aligned} a =& 1, \\ b =& 2\,\mathrm{Re}\{\hat{Z}_{1,2}^{*}\hat{S}_{2,1}\} - 4E_1^2, \\ c =& \mathrm{Re}\{\hat{Z}_{1,2}^{*}\hat{S}_{2,1}\}^2 + \mathrm{Im}\{\hat{Z}_{1,2}^{*}\hat{S}_{2,1}\}^2 = |\hat{Z}_{1,2}^{*}\hat{S}_{2,1}|^2. \end{aligned} \tag{5.1.10}$$

The magnitude of voltage across transmission line 1,2, $V_{1,2}$ must be a real number. Therefore, the term under the square root of the quadratic equation,

$$\begin{aligned} b^2 - 4ac =& 16E_1^4 - 16\,\mathrm{Re}\{\hat{Z}_{1,2}^*\hat{S}_{2,1}\}E_1^2 + \\ & 4\,\mathrm{Re}\{\hat{Z}_{1,2}^*\hat{S}_{2,1}\}^2 - 4|\hat{Z}_{1,2}^*\hat{S}_{2,1}|^2, \end{aligned} \tag{5.1.11}$$

must be nonnegative. Given a complex voltage at one end of a transmission line, and the power flow, loss, and impedance of the transmission line, the success of transferring that power across the line depends on if $4E_1^2 - 4\,\mathrm{Re}\{\hat{Z}_{1,2}^*\hat{S}_{2,1}\}E_1 + \mathrm{Re}\{\hat{Z}_{1,2}^*\hat{S}_{2,1}\}^2 - |\hat{Z}_{1,2}^*\hat{S}_{2,1}|^2 \geq 0$ for the given set of values. If this condition is not fulfilled, it is not possible to support a complex voltage at the load, even if there is sufficient generation to supply the load.

5.1.2 Feasible Power Delivery On Multi-Bus Networks

Equation (5.1.11) gives the condition for a source to feasibly deliver a certain amount of complex power through a line to the other end. In the multi-bus case, this condition needs to be fulfilled at every line, and must be checked for the nodal voltage of both ends of the line. After voltage and power variables are updated at each iteration of distributed Newton method, the conditions

$$4|\hat{E}_i|^4 - 4|\hat{E}_i|^2\,\mathrm{Re}\{\hat{\sigma}_{i,m}\} + \mathrm{Re}\{\hat{\sigma}_{i,m}\}^2 - |\hat{\sigma}_{i,m}|^2 \geq 0 \tag{5.1.12}$$

and

$$4|\hat{E}_m|^4 - 4|\hat{E}_m|^2\,\mathrm{Re}\{-\hat{\sigma}_{i,m}\} + \mathrm{Re}\{-\hat{\sigma}_{i,m}\}^2 - |-\hat{\sigma}_{i,m}|^2 \geq 0 \tag{5.1.13}$$

are checked. If both are satisfied, then the combination of voltages and power flows still produce a feasible solution. If one or both conditions are not met, then corrective action from the FACTS device can be taken. The reactance of the line must be lowered in order to make it easier to deliver power across the line. The algorithm can propose a lower reactance, and then recalculate $\hat{\sigma}_{i,m}$ to check if feasibility conditions would be satisfied. This can be repeated until no more reactance control is available or until there is no more infeasibility problem.

5.2 Load Shedding to Ensure Feasibility

The first two optimality conditions, with respect to power flow and loss, are

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \hat{S}_f} =&F^* + A_f\lambda^* = 0,\\ \frac{\partial \mathcal{L}}{\partial \hat{S}_l} =&L^* + \frac{1}{2}A_L\lambda^* = 0,\end{aligned} \tag{5.2.1}$$

because all the elements of $\frac{\partial F}{\partial \hat{S}_f} = \frac{\partial L}{\partial \hat{S}_l} = 1$, and $\frac{\partial F^*}{\partial \hat{S}_f} = \frac{\partial LF^*}{\partial \hat{S}_l} = 0$. When the Newton method converges, the optimality conditions, G, are sufficiently close to zero. However, this does not mean that power flow equation are fulfilled. As shown in Equation (5.2.1), the mismatches of the constituent relationships, F and L, are not equal to zero if $A_f\lambda^*$ and $\frac{1}{2}A_L\lambda^*$ are not equal to zero. The optimization-based power flow is capable of converging to solutions that fulfill the optimality conditions without satisfying the original power flow equations, and in cases where there are no solutions to power flow, the optimization-based algorithms can still converge to a solution with nonzero Lagrange multipliers.

If the conjugate of Equation (5.2.1) is taken, then

$$\begin{bmatrix} F \\ \\ L \end{bmatrix} = \begin{bmatrix} -A_f\lambda \\ \\ -\frac{1}{2}A_L\lambda \end{bmatrix} = -A^T\lambda. \tag{5.2.2}$$

To balance out the mismatches F and L, which correspond to lines, the change in power injection at each bus is

$$\hat{S}_{n,\text{mismatch}} = -AA^T\lambda. \tag{5.2.3}$$

So if a given system with bus power injections of $\hat{S}_n$ does not return a valid power flow solution, then the simulation can be reattempted using the new bus injections $\hat{S}_n - \hat{S}_{n,\text{mismatch}}$. This gives an idea of at least how much load must be shed, and at which buses, in order to ensure feasible power delivery.

Chapter 6

New Decoupled and DC Power Flow Formulations

6.1 New Decoupled and DC Power Flow Formulations of Real Power Flow

6.1.1 Decoupled Power Flow Formulations

The real power and voltage phase angle decoupling assumption is frequently used to simplify the calculation of power flow [2]. This can be accompanied by the assumption that voltage magnitude is well controlled, and close to 1 p.u. at each bus of the system. Under these assumption, the decoupled real power formulation can be used to model systems with both PQ and PV buses, because buses are only modeled as constant real power.

Conventional Nodal-based Decoupled Real Power Flow

For node $i \in \{1, N-1\}$, the power flow equations are

$$P_i = \sum_{m \in \{i,m\}} G_{i,k}(1 - \cos(\delta_i - \delta_m)) - B_{i,k} \sin(\delta_i - \delta_m), \tag{6.1.1}$$

where the only terms that need to be solved are δ. This is referred to as the decoupled, lossy, real power flow. Applying the S-E graph results in the power loss and flow terms,

$$P_L = 2G_{i,m}(1 - \cos(\delta_i - \delta_m)), \tag{6.1.2}$$

and

$$P_f = -B_{i,m}\sin(\delta_i - \delta_m). \tag{6.1.3}$$

These constituent relationships give real power flow and loss as a nonlinear function of nodal voltage angles.

A further approximation can be made that neglects the resistive elements of the wires. This results in $G = 0$, and, for node $i \in \{1, N-1\}$,

$$P_i = \sum_{k \in \{i,k\}} -B_{i,k}\sin(\delta_i - \delta_k). \tag{6.1.4}$$

Interestingly, the S-E graph formulation is well suited to this approximation, resulting in $P_L = 0$ in place of Equation (6.1.2), and no change to P_f from Equation (6.3.2). This is because the real power loss is dependent on the resistance/conductance, while the transmitted real power is based on the reactance of the wire.

Power flow conservation in terms of real power is

$$AP = \begin{bmatrix} A_f & A_L \end{bmatrix} \begin{bmatrix} P_f \\ \frac{1}{2}P_L \end{bmatrix} = -P_n, \tag{6.1.5}$$

where P_n are the real powers through nodal branches. KVL for complex voltages does not make physical sense when only applied to voltage angles. However, when examining Equations (6.1.2) and (6.3.2), the constituent relationships of flow and loss are both nonlinear function of $\delta_i - \delta_k$, the difference of nodal voltage angles at either end of the wire connecting i and k. Therefore, a set of voltages angles, δ_f, can be used as variables when solving for the voltage solution in the decoupled cases. δ_f are the differences between nodal voltage angles at either ends of a flow branch, rather than the actual angles of the voltages across flow branches. These angle differences follow the KVL equations,

$$A_f\delta_n = \delta_f, \tag{6.1.6}$$

and

$$A_{kvl}\delta_f = 0, \tag{6.1.7}$$

where δ_n are the nodal voltage angles. Because the conventional formulation uses nodal voltages, KVL is not explicitly stated, and flow conservation and constituent relationships are combined into

$$\begin{bmatrix} A_f & A_L \end{bmatrix} \text{diag}(\begin{bmatrix} -B \\ G \end{bmatrix}) \begin{bmatrix} \sin(A_f\delta_n) \\ \overrightarrow{1} - \cos(A_f\delta_n) \end{bmatrix} = -P_n, \tag{6.1.8}$$

where $\overrightarrow{1}$ is a vector of ones, and the sine and cosine terms are element-wise functions of the vector $A_f\delta_n$. Equation (6.1.8) is the most general form of real, decoupled power flow, expressed in terms of nodal voltage angles, δ_n. The lossless case can be expressed by eliminating the rows and columns associated with A_L, G, and $\overrightarrow{1} - \cos(A_f\delta_n)$. A frequently used linearization, called DC power flow, assumes that for small angle values, $\sin(\delta) \approx \delta$. The DC model of real power flow is

$$A_f\text{diag}(B)A_f^T\delta_n = -P_n, \tag{6.1.9}$$

where $A_f\text{diag}(B)A_f^T$ is called the admittance matrix, a Laplacian matrix weighted by the admittances of flow branches.

New Branch-based Decoupled Real Power Flow

As with the AC case, the decoupled case is reformulated using flow branch voltages. However, instead of the actual voltage angle, the voltage angle differences between nodal voltages on either end of a flow branch is used, δ_f. By using the equivalence in Equation (6.1.6), the power flow equations,

$$\begin{bmatrix} A_f & A_L \end{bmatrix} \text{diag}(\begin{bmatrix} -B \\ G \end{bmatrix}) \begin{bmatrix} \sin(\delta_f) \\ \overrightarrow{1} - \cos(\delta_f) \end{bmatrix} = -P_n, \tag{6.1.10}$$

are obtained. As with Equation (3.3.4) in the AC case, there are only $N-1$ equations, so KVL, Equation (6.1.7) is required,

$$A_{kvl}\delta_f = 0.$$

New Optimization-based Formulation of Decoupled Power Flow

The optimization-based formulation of decoupled power flow is similar but it only uses real variables. The optimization problem of the decoupled, lossy case is formulated in much the same way as in the AC case,

$$\begin{aligned}
\underset{P,\delta}{\text{minimize}} \quad & \sum_{i,k\in\{1,L\}} (P_{f:i,k} + B_{i,k}\sin(\delta_{i,k}))^2 + \\
& (P_{L:i,k} - 2G_{i,k}(1-\cos(\delta_{i,k})))^2 \\
\text{subject to} \quad & \\
& \begin{bmatrix} A_f & A_L \end{bmatrix} \begin{bmatrix} P_f \\ P_L \end{bmatrix} = AP = -P_n,
\end{aligned} \tag{6.1.11}$$

$A_{kvl}\hat{\delta} = 0$ is calculated algebraically instead of as a constraint. The lossless formulation can be derived by removing G, P_L, and A_L.

The Lagrangian of Equation (6.1.11) is

$$\begin{aligned}
\mathcal{L}(P,\delta,\lambda) = \sum_{i,k\in\{1,L\}} & (P_{f:i,k} + B_{i,k}\sin(\delta_{i,k}))^2 + \\
(P_{L:i,k} - 2G_{i,k}&(1-\cos(\delta_{i,k})))^2 + \lambda^T(AP+P_n).
\end{aligned} \tag{6.1.12}$$

The optimality conditions are

$$\frac{\partial\mathcal{L}}{\partial P} = 2(P + \begin{bmatrix} \operatorname{diag}(B)\sin(\delta) \\ -2\operatorname{diag}(G)(\overrightarrow{1} - \cos(\delta)) \end{bmatrix}) + A\lambda, \tag{6.1.13}$$

$$\frac{\partial\mathcal{L}}{\partial\delta} = 2\operatorname{diag}(\begin{bmatrix} \operatorname{diag}(B)\cos(\delta) \\ -2\operatorname{diag}(G)(\sin(\delta)) \end{bmatrix})(P + \begin{bmatrix} \operatorname{diag}(B)\sin(\delta) \\ -2\operatorname{diag}(G)(\overrightarrow{1} - \cos(\delta)) \end{bmatrix}), \tag{6.1.14}$$

and

$$AP + P_n = 0. \tag{6.1.15}$$

6.1.2 DC Power Flow Formulations

Conventional Nodal-based DC Power Flow

A frequently used linearization, called DC power flow, assumes that for small angle values, $\sin(\delta) \approx \delta$. The resistance is also assumed to be negligible. The DC model of real power flow through each flow branch is

$$\text{diag}(B)A_f^T\delta_n = P_f. \tag{6.1.16}$$

Then, this expression for P can be substituted into Equation (3.1.8), resulting in

$$A_f\text{diag}(B)A_f^T\delta_n = -P_n, \tag{6.1.17}$$

where $A_f\text{diag}(B)A_f^T$ is called the admittance matrix, a Laplacian matrix weighted by the admittances of flow branches.

New Branch-based DC Power Flow

By linearizing Equation 6.1.10, the conservation of flow equation can be obtained,

$$A_f\text{diag}(B)\delta_f = P_n, \tag{6.1.18}$$

which can be combined with kvl,

$$A_{kvl}\delta_f = 0,$$

resulting in the power flow equations,

$$\begin{bmatrix} A_f \text{diag}(B) \\ \\ A_{kvl} \end{bmatrix} \delta_f = \begin{bmatrix} P_n \\ \\ 0 \end{bmatrix}. \tag{6.1.19}$$

New Optimization-based DC Power Flow Formulation

The linearized version of Equation (6.1.11) is

$$\begin{aligned} &\underset{P_f,\delta}{\text{minimize}} \quad \sum_{i,k\in\{1,L\}} (P_{f:i,k} + B_{i,k}\delta_{i,k})^2 + \\ &\text{subject to} \\ &\qquad A_f P_f = -P_n, \\ &\qquad A_{kvl}\hat{\delta} = 0. \end{aligned} \tag{6.1.20}$$

$A_{kvl}\hat{\delta} = 0$ is calculated algebraically instead of as a constraint. The Lagrangian from this optimization problem is

$$\begin{aligned} \mathcal{L}(P_f, \delta, \lambda) = \sum_{i,k\in\{1,L\}} (P_{f:i,k} + B_{i,k}\delta_{i,k})^2 + \\ +\lambda^T (A_f P_f + P_n). \end{aligned} \tag{6.1.21}$$

The optimality conditions are

$$\frac{\partial \mathcal{L}}{\partial P_f} = 2(P_f + \text{diag}(B)\delta) + A_f \lambda, \tag{6.1.22}$$

$$\frac{\partial \mathcal{L}}{\partial \delta} = 2\text{diag}(\text{diag}(B)(P_f + \text{diag}(B)\delta)), \tag{6.1.23}$$

and

$$A_f P_f + P_n = 0. \tag{6.1.24}$$

6.2 Solving Decoupled and DC Power Flow

6.2.1 Conventional Nodal-based Decoupled Power Flow

Newton Method for Nodal-based Decoupled Power Flow

The decoupled, real power flow equations are

$$\begin{aligned} f(\delta_n) =& A_f[G - \mathrm{diag}(G)\cos B(A_f^T\delta_n) - \\ & \mathrm{diag}(B)\sin(A_f^T\delta_n)] + P_n = 0. \end{aligned} \tag{6.2.1}$$

The Jacobian is

$$\begin{aligned} f'(\delta_n) =& A_f[\mathrm{diag}(G)\mathrm{diag}(\sin B(A_f^T\delta_n)) - \\ & \mathrm{diag}(B)\mathrm{diag}(\cos(A_f^T\delta_n))]A_f^T = 0. \end{aligned} \tag{6.2.2}$$

The Newton step at step k is

$$\Delta\delta_n^{(k)} = f'^{-1}(\delta^{(k)})f(\delta_n^{(k)}), \tag{6.2.3}$$

and the iterative updates are

$$\delta_n^{(k+1)} = \delta_n^{(k)} + \Delta\delta_n^{(k)}. \tag{6.2.4}$$

Jacobi Method for Nodal-based Decoupled Power Flow

The power conservation at node i is

$$\begin{aligned} f_i(\delta_n) = & \sum_{m\in\{i,m\}} [G_{i,m}(1 - \cos(\delta_i - \delta_m)) - \\ & B_{i,m}\sin(\delta_i - \delta_m)] + P_i = 0. \end{aligned} \tag{6.2.5}$$

To derive an expression can be derived for δ_i at iterative step k, first rearrange Equation (6.2.5) such that

$$P_i + \sum_{m\in\{i,m\}} G_{i,m} = a_i\cos(\delta_i^{(k:s)}) + b_i\sin(\delta_i^{(k:s)}), \tag{6.2.6}$$

where

$$\begin{aligned} a_i &= \sum_{m\in\{i,m\}} [G_{i,m}\cos(\delta_m^{(k)}) + B_{i,m}\sin(\delta_m^{(k)})] \\ b_i &= \sum_{m\in\{i,m\}} [B_{i,m}\cos(\delta_m^{(k)}) - G_{i,m}\sin(\delta_m^{(k)})]. \end{aligned} \tag{6.2.7}$$

In matrix form, a and b are

$$\begin{aligned} a =&(A_f\text{diag}(G)A_f^T - \text{diag}(A_f\text{diag}(G)A_f^T))\cos(\delta^{(k)}) \\ &+ (A_f\text{diag}(B)A_f^T - \text{diag}(A_f\text{diag}(B)A_f^T))\sin(\delta^{(k)}), \\ b =&(A_f\text{diag}(B)A_f^T - \text{diag}(A_f\text{diag}(B)A_f^T))\cos(\delta^{(k)}) \\ &- (A_f\text{diag}(G)A_f^T - \text{diag}(A_f\text{diag}(G)A_f^T))\sin(\delta^{(k)}), \end{aligned} \tag{6.2.8}$$

Then, consider the right-hand side of Equation (6.2.16) as a phasor that can be converted into polar form, where

$$\begin{aligned} r_i &= \sqrt{a_i^2 + b_i^2}, \\ \chi_i &= \tan^{-1}(\frac{b_i}{a_i}), \end{aligned} \tag{6.2.9}$$

and then $\delta_i^{(k:s)}$ is

$$\delta_i^{(k:s)} = \sin^{-1}(\frac{1}{r_i}(P_i + \sum_{m\in\{i,m\}} G_{i,m})) + \chi_i. \tag{6.2.10}$$

The Jacobi iteration of δ_i is

$$\delta_i^{(k+1)} = (1-\omega)\delta_i^{(k)} + \omega\delta_i^{(k:s)}. \tag{6.2.11}$$

6.2.2 New Branch-based Decoupled Power Flow

Newton Method for New Branch-based Decoupled Power Flow

The power flow equations of decoupled, real power flow, in terms of flow branch angle differences, are

$$f(\delta_f) = \begin{bmatrix} \begin{bmatrix} A_f & A_L \end{bmatrix} \operatorname{diag}(\begin{bmatrix} -B \\ G \end{bmatrix}) \begin{bmatrix} \sin(\delta_f) \\ \overrightarrow{1} - \cos(\delta_f) \end{bmatrix} + P_n \\ A_{kvl}\delta_f \end{bmatrix} = 0, \tag{6.2.12}$$

and the Jacobian is

$$f'(\delta_f) = \begin{bmatrix} \begin{bmatrix} A_f & A_L \end{bmatrix} \operatorname{diag}(\begin{bmatrix} -B \\ G \end{bmatrix}) \begin{bmatrix} \operatorname{diag}(\cos(\delta_f)) \\ \operatorname{diag}(\sin(\delta_f)) \end{bmatrix} \\ A_{kvl} \end{bmatrix} . \tag{6.2.13}$$

The Newton step is simply

$$\Delta\delta_f^{(k)} = -f'^{-1}(\delta_f^{(k)})f(\delta_f^{(k)}), \tag{6.2.14}$$

and the Newton iterative update is

$$\delta_f^{(k+1)} = \delta_f^{(k)} + \Delta\delta_f^{(k)}. \tag{6.2.15}$$

Jacobi Method for New Branch-based Decoupled Power Flow

Assuming that a proper spanning tree is selected, and that flow branches are ordered such that $\delta_f = [\delta_{tree}^T, \delta_{mesh}^T]^T$, the upper $N-1$ rows of Equation (6.2.12) are

$$\begin{aligned} f_{tree}(\delta_f) = &\begin{bmatrix} A_{f,tree} & A_{f,mesh} \end{bmatrix} \operatorname{diag}(-B) \sin(\begin{bmatrix} \delta_{tree} \\ \delta_{mesh} \end{bmatrix})+ \\ &\begin{bmatrix} A_{L,tree} & A_{L,mesh} \end{bmatrix} \operatorname{diag}(G)(\overrightarrow{1} - \cos(\begin{bmatrix} \delta_{tree} \\ \delta_{mesh} \end{bmatrix}))) \\ &+ P_n = 0, \end{aligned} \tag{6.2.16}$$

and the lower $L-N+1$ rows are

$$f_{mesh}(\delta_f) = A_{kvl} \begin{bmatrix} \delta_{tree} \\ \delta_{mesh} \end{bmatrix} = 0. \tag{6.2.17}$$

By applying Equation (3.1.13), the Jacobi iterate for non-spanning tree flow branches at step k is

$$\delta_{mesh}^{(k:s)} = A_{f,mesh}^T (A_{f,tree}^T)^{-1} \delta_{tree}^{(k)}. \tag{6.2.18}$$

Equation (6.2.16) is rearranged to

$$\begin{aligned} &A_{f,tree}\text{diag}(B_{tree})\sin(\delta_{tree}) + A_{f,mesh}\text{diag}(B_{mesh})\sin(\delta_{mesh}) + \\ &A_{f,tree}\text{diag}(G_{tree})\cos(\delta_{tree}) + A_{f,mesh}\text{diag}(G_{mesh})\cos(\delta_{mesh}) \\ &= P_n + A_L G. \end{aligned} \tag{6.2.19}$$

Let the non-spanning tree terms be aggregated into

$$\begin{aligned} C_J = &- A_{f,mesh}\text{diag}(B_{mesh})\sin(\delta_{mesh}) - \\ &A_{f,mesh}\text{diag}(G_{mesh})\cos(\delta_{mesh}) + P_n + A_L G, \end{aligned} \tag{6.2.20}$$

and let

$$\begin{aligned} D_s =& A_{f,tree}\text{diag}(B_{tree}), \\ D_c =& A_{f,tree}\text{diag}(G_{tree}), \end{aligned} \tag{6.2.21}$$

so that Equation (6.2.19) can be expressed as

$$D_s \sin(\delta_{tree}) + D_c \cos(\delta_{tree}) = C_J, \tag{6.2.22}$$

where D_s and D_c are square matrices. This is because $A_{f,tree}$ and $A_{f,mesh}$ are $(N-1)\times(N-1)$. In order to create the Jacobi iterate, each element of δ_{tree} needs to be expressed in terms of the other elements, so δ_{tree} does not need to be solved explicitly. Therefore, if C_s and C_c are the off-diagonals of D_s and D_c, respectively, then at step k

$$\begin{aligned} &\text{diag}(D_s)\sin(\delta_{tree}^{(k:s)}) + \text{diag}(D_c)\cos(\delta_{tree}^{(k:s)}) \\ &= C_J - C_s \sin(\delta_{tree}^{(k)}) - C_c \cos(\delta_{tree}^{(k)}). \end{aligned} \tag{6.2.23}$$

By using the polar form expression for phasors, Equation (6.2.24) becomes

$$\begin{aligned}&\operatorname{diag}(D_s)\sin(\delta_{tree}^{(k:s)})+\operatorname{diag}(D_c)\cos(\delta_{tree}^{(k:s)})\\&=\operatorname{diag}(r)\sin(\delta_{tree}^{(k:s)}+\chi),\end{aligned}\tag{6.2.24}$$

where

$$\begin{aligned}&\operatorname{diag}(r)=(\operatorname{diag}(D_s)^2+\operatorname{diag}(D_c)^2)^{\frac{1}{2}},\\&\chi=\tan^{-1}(\operatorname{vect}(D_s^{-1}D_c)).\end{aligned}\tag{6.2.25}$$

As a result, $\delta_{tree}^{(k:s)}$ is

$$\begin{aligned}\delta_{tree}^{(k:s)}=&\sin^{-1}(\operatorname{diag}(r)^{-1}[C_J-C_s\sin(\delta_{tree}^{(k)})\\&-C_c\cos(\delta_{tree}^{(k)})])-\chi.\end{aligned}\tag{6.2.26}$$

Then, let $\delta_f^{(k:s)}=\begin{bmatrix}\delta_{tree}^{T(k:s)} & \delta_{mesh}^{T(k:s)}\end{bmatrix}^T$, and

$$\delta_f^{(k+1)}=(1-\omega)\delta_f^{(k)}+\omega\delta_f^{(k:s)}.\tag{6.2.27}$$

6.2.3 New Optimization-based Decoupled Power Flow

Newton Method for New Optimization-based Decoupled Power Flow

Equations (6.1.13), (6.1.14) and (6.1.15) form the optimality conditions of the optimization-based formulation of decoupled power flow. They are equal to zero at the optimum,

$$f(P,\delta,\lambda)=\begin{bmatrix}\partial\mathcal{L}/\partial P\\ \partial\mathcal{L}/\partial\lambda\\ \partial\mathcal{L}/\partial\delta\end{bmatrix}=0.\tag{6.2.28}$$

The Hessian is

$$H(P,\delta,\lambda)=\nabla f(\hat{S},\hat{V},\lambda)=\begin{bmatrix}\frac{\partial^2\mathcal{L}}{\partial P^2} & \frac{\partial^2\mathcal{L}}{\partial P\partial\lambda} & \frac{\partial^2\mathcal{L}}{\partial P\partial\delta}\\ \frac{\partial^2\mathcal{L}}{\partial\lambda\partial P} & \frac{\partial^2\mathcal{L}}{\partial\lambda^2} & \frac{\partial^2\mathcal{L}}{\partial\lambda\partial\delta}\\ \frac{\partial^2\mathcal{L}}{\partial\delta\partial P} & \frac{\partial^2\mathcal{L}}{\partial\delta\partial\lambda} & \frac{\partial^2\mathcal{L}}{\partial\delta^2}\end{bmatrix},\tag{6.2.29}$$

which becomes

$$H(P,\delta,\lambda) = \begin{bmatrix} 2I & A^T & H_c \\ A & 0 & 0 \\ H_c^T & 0 & H_\delta \end{bmatrix}, \tag{6.2.30}$$

The off-diagonal blocks are

$$H_c = \mathrm{diag}(\begin{bmatrix} 2\mathrm{diag}(B)\cos(\delta) \\ -4\mathrm{diag}(G)\sin(\delta) \end{bmatrix}), \tag{6.2.31}$$

and its transpose, h_c^T.

The bottom right diagonal block is

$$H_\delta = \\ \mathrm{diag}\begin{bmatrix} -2\mathrm{diag}(B)\sin(\delta)(P_f + \mathrm{diag}(B)\sin(\delta)) + 2\mathrm{diag}(B)^2\cos(\delta) \\ -4\mathrm{diag}(G)\cos(\delta)(P_L - \mathrm{diag}(G)\cos(\delta)) - 8\mathrm{diag}(G)^2\sin(\delta) \end{bmatrix}. \tag{6.2.32}$$

The non-spanning tree flow branch voltages need to be removed, because they are algebraically dependent on the spanning tree voltages. Therefore, the rows and columns corresponding to δ_{mesh} are removed. The power flows through those lines do not need to be removed.

Because the decoupled problem only uses real valued variables, the Newton step at iteration k is

$$\Delta \begin{bmatrix} P^{(k)} \\ \lambda^{(k)} \\ \delta^{(k)} \end{bmatrix} = -H^{-1}f, \tag{6.2.33}$$

and

$$\begin{bmatrix} P^{(k+1)} \\ \lambda^{(k+1)} \\ \delta^{(k+1)} \end{bmatrix} = \begin{bmatrix} P^{(k)} \\ \lambda^{(k)} \\ \delta^{(k)} \end{bmatrix} + \Delta \begin{bmatrix} P^{(k)} \\ \lambda^{(k)} \\ \delta^{(k)} \end{bmatrix}. \tag{6.2.34}$$

New Distributed Newton Method for Optimization-based Decoupled Power Flow

As with the AC formulation, the decomposition based on approximate Newton directions is used on the Hessian, from Equation (6.2.30). The result is the distributed Hessian,

$$H_{distr}(P,\delta,\lambda) = \begin{bmatrix} 2I & A^T & 0 \\ A & 0 & 0 \\ 0 & 0 & H_\delta \end{bmatrix}, \tag{6.2.35}$$

and the Newton step can be solved through

$$\begin{bmatrix} 2I & A^T & 0 \\ A & 0 & 0 \\ 0 & 0 & H_\delta \end{bmatrix} \begin{bmatrix} \Delta P \\ \Delta\lambda \\ \Delta\delta \end{bmatrix} = - \begin{bmatrix} \partial\mathcal{L}/\partial P \\ \partial\mathcal{L}/\partial\lambda \\ \partial\mathcal{L}/\partial\delta \end{bmatrix}. \tag{6.2.36}$$

Equation (6.2.36) is rearranged into two subproblems that solve for each Newton step at iteration k,

$$\Delta\delta^{(k)} = -H_\delta^{-1}\frac{\partial\mathcal{L}}{\partial\delta}, \tag{6.2.37}$$

and

$$\begin{bmatrix} 2I & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta P^{(k)} \\ \Delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} \partial\mathcal{L}/\partial P \\ \partial\mathcal{L}/\partial\lambda. \end{bmatrix} \tag{6.2.38}$$

Once again, by using the method presented in [13], the Newton steps are calculated as

$$\Delta\lambda^{(k)} = (2AA^T)^{-1}(\frac{\partial\mathcal{L}}{\partial\lambda} - 2A\frac{\partial\mathcal{L}}{\partial P}), \tag{6.2.39}$$

and

$$\Delta P^{(k)} = -\frac{1}{2}(\frac{\partial\mathcal{L}}{\partial P} + A^T\Delta\lambda^{(k)}). \tag{6.2.40}$$

After the Newton steps are calculated, the variables can all be updated.

$$\begin{bmatrix} P^{(k+1)} \\ \lambda^{(k+1)} \\ \delta^{(k+1)} \end{bmatrix} = \begin{bmatrix} P^{(k)} \\ \lambda^{(k)} \\ \delta^{(k)} \end{bmatrix} + \omega\Delta \begin{bmatrix} P^{(k)} \\ \lambda^{(k)} \\ \delta^{(k)} \end{bmatrix}. \tag{6.2.41}$$

Once again, ω is a damping factor to aid convergence. In the real valued case, the convergence of Newton steps using the decomposed Hessian, compared to the original Hessian, is dependent on the condition

$$\rho(I - H_{distr}^{-1} H) < 1, \tag{6.2.42}$$

where $\rho(M)$ indicates the spectral radius of matrix M. If the condition in Equation (6.2.42) is met, then the Newton method using the decomposed Hessian or Jacobian reaches the same solution as the algorithm using the unaltered Hessian or Jacobian [36], [38].

6.2.4 Conventional Newton Method for Nodal-based DC Power Flow

Because Equation (6.1.17) is a set of linear equations, the Newton step at iteration k is simply

$$\Delta\delta_n^{(k)} = -(A_f \text{diag}(B) A_f^T)^{-1}(A_f \text{diag}(B) A_f^T \delta_n^{(k)} + P_n), \tag{6.2.43}$$

where $A_f \text{diag}(B) A_f^T$ is the Jacobian matrix.

$\delta_n^{(k)}$ is iteratively updated using the scheme

$$\delta_n^{(k+1)} = \delta_n^{(k)} + \Delta\delta_n^{(k)}. \tag{6.2.44}$$

Conventional Jacobi Method for Nodal-based DC Power Flow

The distributed numerical method of solving Equation (6.1.17) is to use Jacobi method. If D is defined as the diagonal terms of $A_f \text{diag}(B) A_f^T$, and C is defined as the off-diagonal terms of $A_f \text{diag}(B) A_f^T$ [34], then

$$(D + C)\delta_n = -P_n, \tag{6.2.45}$$

and the iteration steps can be developed as

$$\delta_n^{(k+1)} = D^{-1}(b - C\delta_n^{(k)}). \tag{6.2.46}$$

If Equation (6.2.46) is examined row by row, it is shown that each iteration of δ_n at each bus depends on its own bus injection, as well as δ_n at the neighboring buses. Since D is a diagonal matrix, its inverse is also diagonal, which does not change the structure of $-P_n - C\delta_n$. Because $A_f\text{diag}(B)A_f^T$ has the structure of a weighted Laplacian, the off-diagonal terms are only nonzero when buses are connected to other buses through lines.

Newton Method for New Branch-based DC Power Flow

The branch-based DC power flow equations are

$$\begin{bmatrix} A_f\text{diag}(B) \\ A_{kvl} \end{bmatrix} \delta = \begin{bmatrix} -P_n \\ 0 \end{bmatrix}. \tag{6.2.47}$$

Solving Equation (6.2.47) using Newton method results in the following Newton step at iteration k:

$$\Delta\delta^{(k)} = \begin{bmatrix} A_f\text{diag}(B) \\ A_{kvl} \end{bmatrix}^{-1} \left(\begin{bmatrix} A_f\text{diag}(B) \\ A_{kvl} \end{bmatrix} \delta^{(k)} - \begin{bmatrix} -P_n \\ 0 \end{bmatrix} \right), \tag{6.2.48}$$

and δ is iteratively updated as

$$\delta^{(k+1)} = \delta^{(k)} + \Delta\delta^{(k)}. \tag{6.2.49}$$

Jacobi Method for New Branch-based DC Power Flow

Equation (6.2.47) can also be solved using Jacobi method. If D_δ is the matrix of the diagonal terms of $\begin{bmatrix} A_f\text{diag}(B) \\ A_{kvl} \end{bmatrix}$, and C_δ is the matrix of the off-diagonal terms of $\begin{bmatrix} A_f\text{diag}(B) \\ A_{kvl} \end{bmatrix}$, the Jacobi method iteration steps are

$$\delta^{(k+1)} = D_\delta^{-1}(\begin{bmatrix} -P_n \\ 0 \end{bmatrix} - C_\delta \delta^{(k)}). \tag{6.2.50}$$

However, the Jacobian, $\begin{bmatrix} A_f \text{diag}(B) \\ A_{kvl} \end{bmatrix}$ is never strictly diagonally dominant if the system is meshed, because the terms of A_{kvl} are ones, negative ones, and zeros. Even if the network were radial, the matrix $A_f \text{diag}(B)$ would not necessarily be strictly diagonally dominant. The Jacobi method would be unreliable for solving power flow in terms of flow branch voltages.

Newton Method for New Optimization-based DC Power Flow

The Lagrangian of the optimization-based DC power flow equations, from Equation (6.1.21), is

$$\mathcal{L}(P, \delta, \lambda) = \sum_{i,k \in \{1,L\}} (P_{f:i,k} + B_{i,k}\delta_{i,k})^2 + \, + \lambda^T (A_f P_f + P_n).$$

As with the optimality condition of decoupled power flow, Equations (6.2.28, the DC power flow optimality condition are

$$f(P_f, \delta, \lambda) = \begin{bmatrix} \partial\mathcal{L}/\partial P_f \\ \partial\mathcal{L}/\partial\lambda \\ \partial\mathcal{L}/\partial\delta \end{bmatrix} = \begin{bmatrix} 2(P_f + \text{diag}(B)\delta) \\ A_f P_f + P_n \\ 2\text{diag}(B)(P_f + \text{diag}(B)\delta) \end{bmatrix} = 0. \tag{6.2.51}$$

The full Hessian is

$$H(P, \delta, \lambda) = \begin{bmatrix} 2I & A_f^T & \text{diag}(B) \\ A_f & 0 & 0 \\ \text{diag}(B) & 0 & \text{diag}(B)^2 \end{bmatrix}, \tag{6.2.52}$$

but with the rows and columns corresponding to nonspanning tree voltage phase angles removed, so that there are only $N-1$ phase angle variables. This reduces the number of terms in the bottom right diagonal block, $\text{diag}(B)^2$, as well as removes the corresponding rows or columns of $\text{diag}(B)$ from the upper right and lower left blocks.

Let

$$x = \begin{bmatrix} P_f \\ \lambda \\ \delta \end{bmatrix}, \tag{6.2.53}$$

then the Newton step for the optimization-based power flow at step k is expressed as

$$\Delta x^{(k)} = -(H^{(k)})^{-1} f^{(k)}. \tag{6.2.54}$$

Then, Newton method is iteratively updated based on

$$x^{(k+1)} = x^{(k)} + \alpha \Delta x^{(k)}, \tag{6.2.55}$$

New Distributed Newton Method for Optimization-based DC Power Flow

By using approximate Newton directions [36], the Hessian of the decomposed problem is

$$H(P, \delta, \lambda) = \begin{bmatrix} 2I & A_f^T & 0 \\ A_f & 0 & 0 \\ 0 & 0 & \operatorname{diag}(B)^2 \end{bmatrix}, \tag{6.2.56}$$

This decouples the updates of P_f and λ with the updates of δ. The subproblem of P_f and λ becomes

$$\begin{bmatrix} 2I & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta P_k \\ \Delta \lambda_k \end{bmatrix} = - \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial P_f} \\ \frac{\partial \mathcal{L}}{\partial \lambda} \end{bmatrix}. \tag{6.2.57}$$

The δ subproblem becomes

$$\Delta P_k = -(\operatorname{diag}(B)^2)^{-1} \frac{\partial \mathcal{L}}{\partial P_f}, \tag{6.2.58}$$

which can be solved with local information, because $\operatorname{diag}(B)^2$ is diagonal and the inverse can be found by inverting each line's own susceptance value.

The form of the first subproblem, from Equations (6.2.57) is in the same form as the Hessian of the optimization problem found in [13], and the Newton steps are

$$\Delta\lambda^{(k)} = (A_f(2I)^{-1}A_f^T)^{-1}(\frac{\partial\mathcal{L}}{\partial\lambda} - A_f(2I)^{-1}\frac{\partial\mathcal{L}}{\partial P_f}), \tag{6.2.59}$$

and

$$\Delta P_f^{(k)} = -(2I)^{-1}(\frac{\partial\mathcal{L}}{\partial P_f} + A^T\Delta\lambda^{(k)}). \tag{6.2.60}$$

6.3 Ensuring Feasible Power Delivery using the New Decoupled Optimization-based Formulation

6.3.1 Targeted Load Shedding

The main principles described in Section 5.1 apply to the decoupled real power flow formulation in a similar way as in the AC formulation. Targeted load shedding is accomplished by calculating a set of nodal mismatches based on the Lagrange multiplier, λ. However, the decoupled optimality conditions with respect to P, Equation (6.1.13), has a factor of 2 resulting from taking the partial derivative of the square of a function. This is because the real-value domain objective function is squared, instead of products of conjugates in the complex-valued domain. Therefore, the mismatch is simply

$$P_{n:mismatch} = \frac{1}{2}AA^T\lambda. \tag{6.3.1}$$

6.3.2 Line Reactance Adjustment

Because the decoupled formulation assumes that voltage magnitude is well managed, the infeasible solution problem focuses on the voltage phase angles. The S-E graph model of lines becomes extremely useful, because only the power transferred, rather than loss, is important to power delivery. This is coupled with the fact that transferred power, $P_{f:i,m}$, is only a function of the sine, but not cosine, of the phase angle difference across the line, $\delta_{i,m}$. By solving for $\delta_{i,m}$ using Equation (6.3.2), the equation

$$\delta_{i,m} = \sin^{-1}(\frac{-P_f}{B_{i,m}}) \tag{6.3.2}$$

is obtained, which has a clearly defined range. For real values of $\frac{P_f}{B_{i,m}}$, the magnitude of $\delta_{i,m}$ cannot exceed 1. Therefore, the condition to check at every line is simply $|\frac{P_f}{B_{i,m}}| \leq 1$. If this condition is satisfied, then $\delta_{i,m}$ has a feasible solution. If the condition is not satisfied, then reactance should be incremented lower, which raises susceptance, to try to meet the condition.

6.4 Decoupled Distributed Power Flow and Infeasible Power Delivery Simulations

This section includes distributed power flow simulations for the decoupled real power formulation. These results have been published in [9], [10], and [14].

6.4.1 2 Bus Example

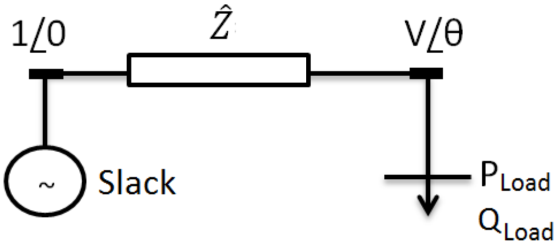


Figure 6.1: 2 bus example with a slack bus and a constant power load.

Consider the 2 bus system featured in Figure (6.1). In this two bus case, closed form solutions allow for the direct calculation of the line flow limit of the network. For this 2 bus network, let $Z = 0 + 0.1jp.u.$, which means $B = -10$, $G = 0$, and let $P_{Load} = -1$, which means 1 p.u. of power is being consumed at the load. The system has the following parameters: $A = -1$, $b = P_{Load} = -1$.

In this simple case, the line flow limit based on the closed form solution is $P_{f,max} = \pm B = \pm 10$. Since we know that power balance at node two must be fulfilled, we know the actual flow is only 1 p.u., so this system is feasible. We can also calculate the phase angle at bus 2, by using $\theta = \sin^{-1}(\frac{-P_{Load}}{B})$. This is calculated to be $\theta = \sin^{-1}(-0.1) = -5.7392$ degrees.

Simulations were conducted with thresholds at 0.0001 p.u. and starting angles and flows at 0 p.u. Both conventional Newton Raphson method and the optimization based distributed method reach the same result as the closed form solution with initial angle 0. However, results differ in the infeasible cases. For example, if $Z = 0 + 10j$, then $B = -0.1$ and $G = 0$. If the load is still $P_{Load} = -1$, then we know we have an infeasible case because the maximum transfer limit is $P_{f,max} = \pm B = \pm 0.1$. In addition, if we attempt to solve $\theta = \sin^{-1}(\frac{-P_{Load}}{B})$, then we get the inverse sine of -10, which does not have a real solution. This means that no phase angle satisfies the power flow equations.

In the infeasible case, Newton Raphson method does not converge and continues to iterate until stopped. However, the optimization method does converge, and it converges to a solution where the objective function is not zero.

Table 6.1 depicts the solution reached in the infeasible case. In this infeasible 2 bus example, λ was found to be 2.2. A and A^T are equal to -1, because there is only one line entering one node. Therefore $P_{n:mismatch} + P_n$ is equal to 0.1. Based on the closed form solutions, this amount is possible to transfer across the transmission line when $B = -0.1$. The closed form solution is $\theta = \sin^{-1}(\frac{-P_{Load}}{B}) = 90$ degrees. Both conventional and distributed optimization simulations confirm that this is the solution. This may not be a good engineering solution, since a load of 0.1 is generating instead of consuming. However, it gives some bounds to work with when deciding how to adjust the system, and the concept of adjusting injections is illustrated in this simple case.

In the 2 bus case, adjusting the line reactance is very trivial. $|\frac{-P_{Load}}{B}|$ must be less than 1, so B must be increased until that is the case. If P_{Load} is -1, then $|B|$ must be at least 1.

Table 6.1: Optimization based method solution for the 2 bus network: infeasible case

Angles in degrees, power in p.u,				
δ init.	δ final	P_f init.	P_f final	λ
0	90	0	1	2.2

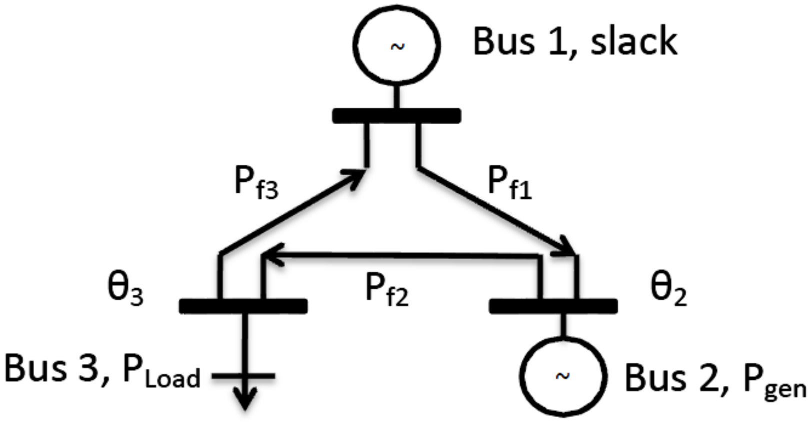


Figure 6.2: 3 bus example with a slack bus, another generator and a constant power load.

6.4.2 3 Bus Example

For this 3 bus example, $A = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$, $b = \begin{bmatrix} P_{gen} \\ P_{Load} \end{bmatrix}$, and $A_{kvl} = \begin{bmatrix} -1 & -1 & -1 \end{bmatrix}$.

Let the 3 bus system have the following parameters: $P_{gen} = 2$, $P_{Load} = -2$, all $G = 0$ and all $B = -10$.

Table 6.2 shows the simulation results for the conventional Newton method on the 3 bus case, and Table 6.3 shows the simulation results for the optimization based method. In the feasible case, both methods converge to almost the same solution. It should be noted that λ and μ were on the order of magnitude of 10^{-6} for the optimization based method.

Consider the infeasible case where all $B = -0.1$. Using intuition, it is realized that this case is infeasible because both paths from the generator to the load cannot transfer enough power for there to be a power flow solution. Using these new line parameters, conventional Newton method fails to converge.

Table 6.4 shows the simulation results for the optimization based method after running the infeasible case. In this, $\lambda = \begin{bmatrix}-1.1225 & 1.1224\end{bmatrix}^T$ and $\mu = 0.0666$. $P_{n:mismatch}$ is calculated to be $\begin{bmatrix}-1.8240 & 1.8240\end{bmatrix}^T$, and the new adjusted injections are $b_{new} = \begin{bmatrix}0.1760 & -0.1760\end{bmatrix}^T$.

Table 6.6 and Table 6.5 show simulation results for the newly adjusted system. The phase angles from either algorithm only deviate by around 1 degree, while the flows deviate by less than 0.01 p.u. However, both methods converged, and the Lagrange multipliers from the optimization method were on the order of magnitude of 10^{-5}. Therefore, the infeasible network has been adjusted to become feasible by shedding generation and load.

We also run the infeasible case while allowing for an unlimited amount of adjustment to the susceptances of the lines. Table 6.7 shows the solution if the algorithm is run with the adjustments made to reactances between iterations.

In this case, the Lagrange multipliers are on the order of magnitude of 10^{-8}. The new line susceptances are $B_{1,2} = -1.0752$, $B_{2,3} = -1.9856$ and $B_{3,1} = -1.0752$. The original infeasible system was exaggerated with very large line reactances, and therefore small susceptances, so therefore both the injection and reactance adjustment methods made large changes to the network. However, in cases that are close to being feasible, it may only require very small adjustments in order to ensure feasibility.

Table 6.2: Newton method solution for the 3 bus network: feasible case

Angles in degrees, power in p.u,				
θ_1	θ_2	$P_{f,1}$	$P_{f,2}$	$P_{f,3}$
3.8283	-3.8283	-0.6677	1.3324	-0.6677

6.5 IEEE 14 Bus Example

Figure 6.3 depicts the IEEE 14 bus system. The system parameters can be found at [39]. Both the centralized Newton method and the optimization based distributed method found the power

Table 6.3: Optimization based method solution for the 3 bus network: feasible case

Angles in degrees, power in p.u,				
θ_1	θ_2	$P_{f,1}$	$P_{f,2}$	$P_{f,3}$
3.8283	-3.8283	-0.6677	1.3323	-0.6676

Table 6.4: Optimization based method solution for the 3 bus network: infeasible case

Angles in degrees, power in p.u,				
θ_1	θ_2	$P_{f,1}$	$P_{f,2}$	$P_{f,3}$
306.3745	-306.3759	-0.6418	1.2180	-0.6417

Table 6.5: Newton method solution for the 3 bus network: infeasible case with adjusted injections

Angles in degrees, power in p.u,				
θ_1	θ_2	$P_{f,1}$	$P_{f,2}$	$P_{f,3}$
-53.1301	53.1301	-0.0800	0.0660	-0.0800

Table 6.6: Optimization based method solution for the 3 bus network: infeasible case with adjusted injections

Angles in degrees, power in p.u,				
θ_1	θ_2	$P_{f,1}$	$P_{f,2}$	$P_{f,3}$
-52.5129	52.4806	-0.079369	0.096643	-0.079335

flow solution to within 0.01 degree of each other, in terms of phase angle differences. However, the conventional solver only required 4 iterations while the distributed method took 126 Newton steps, and 2395 total steps, which means that an enhanced convergence rate is necessary to make this algorithm practical for industrial use. When homotopy perturbation method was used to estimate the Lagrange multipliers, which eliminated the Jacobi steps, the total number of Newton steps

Table 6.7: Optimization based method solution for the 3 bus network: infeasible case with adjusted reactances

Angles in degrees, power in p.u,				
θ_1	θ_2	$P_{f,1}$	$P_{f,2}$	$P_{f,3}$
-25.4026	25.3995	-0.46123	1.5388	-0.46117

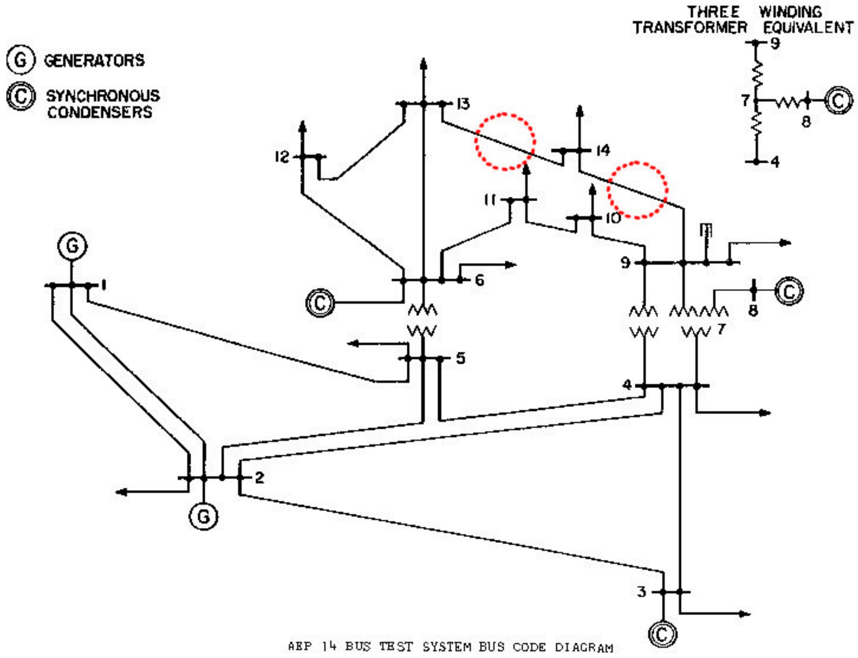


Figure 6.3: IEEE 14 bus system. The two circled transmission lines were adjusted so that an infeasible power flow for the network was created.

stayed the same at 126. This puts the number of steps in a similar range to the conventional Newton method.

An infeasible case was created by increasing the reactances of lines 17 and 20 by a factor of 100. These two lines connect to bus 14, which results in it being impossible to supply the load at bus 14. The conventional solver fails to find a solution to the infeasible case within 10000 iterations. The optimization based method finds a solution at which μ is on the order of 10^{-4} and λ is also close to 10^{-4}, except for λ_{14}, which was 0.0669. This makes sense because we expect that bus 14 cannot have its load served. By using $\frac{1}{2}AA^T\lambda$, we can adjust and try the new injections to see if the system is feasible.

Table 6.8 shows the results of running the infeasible system with adjusted injections. Lines 17 and 20 had phase angle differences of close to 90 degrees, which would result in maximum power being transferred through these lines. Besides lines 17 and 20, the angles between centralized and distributed solutions did not deviate by more than 0.01 degrees. Lines 17 and 20 had phase angles which deviated by around 1.45 degrees. This system is barely feasible and it is not desirable for phase angle differences of lines to be so high. However, this gives an upper limit on generation and load which can guide the system operator to a feasible and desirable power flow solution. The distributed method took 171 Newton steps and 3250 total steps while the conventional Newton method only took 18 steps in order to solve for the adjusted system. When homotopy perturbation method was used, the total number of Newton steps stayed the same at 171. This puts the number of steps in a similar range to the conventional Newton method.

Table 6.9 shows the results of running the infeasible system with adjusted reactances. The algorithm for adjusting line reactances changed the susceptance of line 17 from -0.0369 to -0.0875 and the susceptance of line 20 from -0.0287 to -0.0618. This resulted in the reactance of line 17 to be adjusted from 27 p.u. to 11 p.u. and for the reactance of line 20 to be adjusted from 34 p.u. to 17 p.u. This once again led to a system which was transferring close to the maximum limits of lines 17 and 20, with phase angles close to 90 degrees across those two lines. The distributed method took 162 Newton steps and 3079 total steps (including Jacobi steps) while the conventional Newton method only took 116 steps in order to solve for the adjusted system. When homotopy perturbation method was used, the total number of Newton steps stayed the same at 162. This puts the number of steps in a similar range to the conventional Newton method.

The algorithm for adjusting line reactances changed the susceptance of line 17 from -0.0369 to -0.0875 and the susceptance of line 20 from -0.0287 to -0.0618.

Table 6.8: IEEE 14 bus decoupled results (infeasible, adjusted injections)

Angles in degrees		
Line number	Phase angle difference (Centralized)	Phase angle difference (Distributed)
1	5.7282	5.7270
2	10.0970	10.0949
3	8.6796	8.7430
4	6.0168	6.0157
5	4.3689	4.3678
6	-2.6628	-2.6633
7	-1.6479	-1.6477
8	3.2681	3.2668
9	4.9873	4.9859
10	6.0519	6.0497
11	0.7447	0.7446
12	1.3386	1.3381
13	1.6653	1.6648
14	0	-0.0001
15	1.7192	1.7188
16	0.3973	0.3971
17	90.1951	88.7419
18	-0.2358	-0.2361
19	0.3267	0.3266
20	89.1130	87.6608

Table 6.9: IEEE 14 bus decoupled results (infeasible, adjusted line reactances)

Angles in degrees		
Line number	Phase angle difference (Centralized)	Phase angle difference (Distributed)
1	5.8214	5.8201
2	10.2805	10.2781
3	8.7389	8.7382
4	6.1293	6.1279
5	4.4591	4.4581
6	-2.6097	-2.6102
7	-1.6702	-1.6700
8	3.4402	3.4388
9	5.2492	5.2471
10	6.2286	6.2263
11	0.7922	0.7920
12	1.3668	1.3662
13	1.7298	1.7294
14	0	-0.0001
15	1.8090	1.8086
16	0.3798	0.3796
17	90.3086	91.3279
18	-0.2784	-0.2786
19	0.3631	0.3632
20	89.2696	90.2894

Chapter 7

Simulations and Examples

7.1 Power Flow Calculation

7.1.1 3 Bus Example

Figure 7.1 depicts a simple 3 bus power network. The node at bus 1 is taken as the slack bus, with a voltage of $1\angle 0$. The incidence matrix of nodes and flow branches, A_f, is

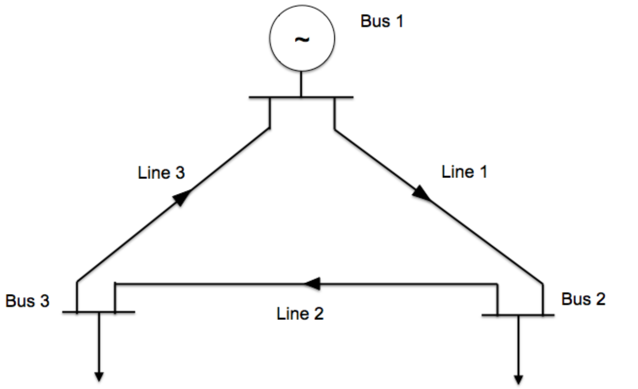


Figure 7.1: A 3 bus system interconnected with 3 lines.

$$A_f = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix},$$

and

$$A_L = |A_f| = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix},$$

The complex nodal power injections at buses 2 and 3 are

$$\hat{S}_n = \begin{bmatrix} \hat{S}_{n,2} \\ \hat{S}_{n,3} \end{bmatrix} = - \begin{bmatrix} 1.5 + 0.057j \\ -1 + 0.036j \end{bmatrix},$$

and the impedance of each transmission line is $0.001 + 0.1j$, so the vector of flow branch admittances is

$$Y = \begin{bmatrix} Y_{1,2} \\ Y_{2,3} \\ Y_{3,1} \end{bmatrix} = \begin{bmatrix} 0.100 - 9.999j \\ 0.100 - 9.999j \\ 0.100 - 9.999j \end{bmatrix}.$$

These parameters are from the 3 bus example in [35], except with an added resistive element, 0.001 p.u., to the flow branches.

The spanning tree selected for the 3 bus examples are flow branches 1 and 2. Therefore,

$$A_{tree} = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix},$$

and

$$A_{mesh} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

This comes from the matrix A_f, reduced by the first row, which corresponds to the slack bus. A_{tree} and A_{mesh} is used to construct A_{kvl},

$$A_{kvl} = \begin{bmatrix} -1 & -1 & -1 \end{bmatrix}.$$

This A_{kvl} matrix matches the directions of the flow branches of the 3 bus network, which form the only loop in the graph.

Matpower 5.0 [40] was used to verify the power flow solutions reached by the algorithms presented in this work. Matpower uses Newton method to solve for the complex nodal voltages in terms of their polar components, the magnitude and phase angle. The complex power balance at each node is separated into real and imaginary components, creating a system of equations in terms of real valued variables.

The AC power flow solution of the 3 bus case is

$$\hat{E}_2 = 1.001\angle 3.818^\circ,$$

$$\hat{E}_3 = 1.000\angle -0.959^\circ.$$

The algorithms started each nodal voltage at $1\angle 0$, each flow branch voltage difference at zero, and flows and losses at zero. Lagrange multipliers, λ, also started at zero. Each algorithm was iteratived until each element of their respective vector functions, $f(x^{(k)})$, was within 10^{-3} p.u. of zero. In the case of Newton and Jacobi method, $f(x^{(k)})$ corresponds to the fulfillment of the power flow equations. In the case of distributed Newton method for the optimization-based formulation, $f(x^{(k)})$ is the gradient of the Lagrangian.

Another possible convergence condition is to stop iterating when the change in variables between iterations is sufficiently close to zero. However, this may not result in the same accuracy between different methods. For example, in order for Jacobi method to satisfy power flow equations to 10^{-3}, the change of variables between iterations needed to be 10^{-5} in the 3 bus and 14 bus cases. A scaling factor $\omega = 0.5$ was used for Jacobi method, and $\omega = 0.8$ was used for distributed Newton method. The voltage solutions to branch formulations were converted back to nodal voltages using $\hat{E} = (A_{tree}^T)^{-1}\hat{V}$.

Table 7.2 shows the number of iterations required for each algorithm, using the nodal, branch, and branch optimization-based formulations of power flow. Newton method converges for all three formulations, requiring 7 or less iterations to solve the 3 bus case using all three formulations. The distributed Newton method required 12 iterations to converge for the 3 bus case.

The Jacobi method converges for the nodal, but not branch, formulation of AC power flow. The Jacobian of the nodal formulation meets the strict diagonal dominance criteria, but the Jacobian of the branch formulation does not. Upon further examination, if a system is meshed, then the A_{kvl} rows of the Jacobian with respect to the branch formulation of power flow only has 1, 0, or -1 as possible elements, so there is no column that can contain a dominant element. There may be radial networks where the Jacobian can be arranged such that it is strictly diagonally dominant.

The decoupled, lossy power flow solution is

$$\delta_2 = 3.8240^\circ,$$

$$\delta_3 = -0.9549^\circ,$$

and the decoupled, lossless power flow solution is

$$\delta_2 = 3.8238^\circ,$$

$$\delta_3 = -0.9560^\circ,$$

In this particular example, the change in voltage angles from adding resistive losses is very low, because the resistance is much lower than the reactance. Tables 7.4 and 7.6 show the iterations required in the decoupled lossy, and lossless cases, respectively. Similar to the AC formulations, the decoupled formulations took very few iterations to converge. Only 2 or 3 iterations were required to solve the 3 bus case, both in the lossy and lossless cases. Jacobi method with the nodal formulation of decoupled power flow converges in 8 iterations forthe lossless case. However, in the lossy case, the Jacobi method does not reach within 10^{-3} degrees of the solution even when the change in variables between each iteration is as low as 10^{-20}, which takes 124 iterations. Instead, if the change between iterations is 10^{-5}, around the same as at convergence of the lossless case, then the voltage angles are 10^{-1} degrees away from the solution after 24 iterations. Therefore, Tables 7.4

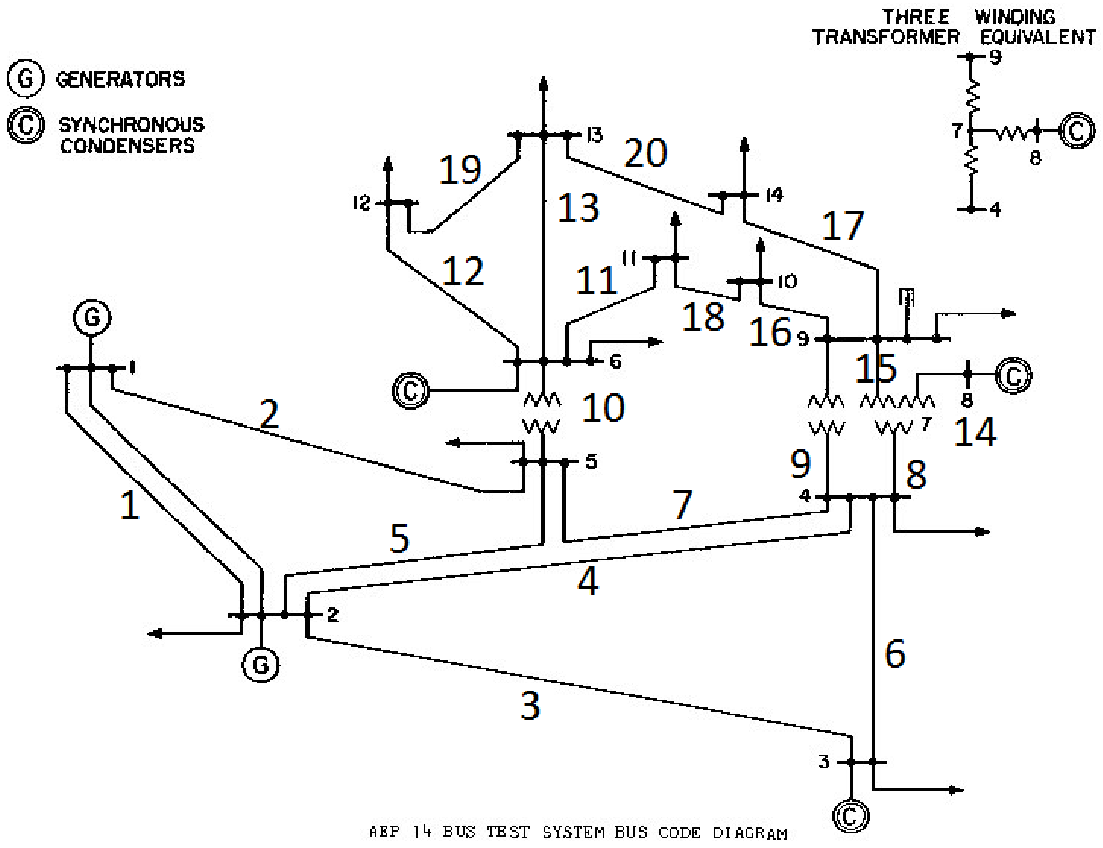


Figure 7.2: The IEEE 14 bus network, with 20 lines.

lists the number of iterations as 24, because the algorithm does not diverge, but has diminishing returns on convergence after reaching within 10^{-1} degrees of the solution.

Unlike in the AC case, usage of the Jacobi method on the branch formulation of decoupled power flow resulted in an infeasible answer. The voltage phase angles resulted in complex values during the iterative algorithm. In Matlab, this comes from taking the inverse sine of values of magnitude greater than 1. The Jacobian in the decoupled case also has the matrix A_{kvl}, and is therefore not diagonally dominant. Therefore the Jacobi method implemented on the branch formulation of decoupled power flow does not meet convergence conditions. The distributed Newton method required 6 iterations to converge in the 3 bus case, both for the lossy and lossless formulations.

7.1.2 14 Bus Example

Figure 7.2 shows the IEEE 14 bus network [39]. Power flow simulations were performed after adjusting the system such that PV buses were made into PQ buses with zero reactive power drawn, and with the real power injection remaining the same. Additionally, shunt capacitances were removed. This creates a graph whose flow branches are reactive and resistive, and whose nodal branches are constant complex power sources (sources in the sense that the value of power is given).

Table 7.7 gives the power flow solution of the AC formulation of power flow for the IEEE 14 bus network. This solution is verified using Matpower. The algorithms that converged, converged to within 10^{-3} p.u., and 10^{-2} degrees of this solution. The number of iterations for each numerical algorithm and formulation is shown on Table 7.2.

As with the 3 bus case, the Newton method took a small number of iterations, between 7 and 14 for all three formulation of AC power flow. However, the distributed methods, Jacobi method and distributed Newton method, used hundreds of iterations to reach a sufficiently close solution. The Jacobi method, using nodal formulation of AC power flow, took 534, and the distributed Newton method took 355 iterations to reach a sufficiently close solution. A large increase in the number of iterations occurs between the 3 and 14 bus examples when using distributed algorithms.

Table 7.8 lists the lossy and lossless solution to decoupled power flow on the 14 bus network. The decoupled formulations of 14 bus network have similar results to the AC formulation. Newton method takes only 3 steps to converge using the nodal and branch formulations of decoupled power flow, and 10 and 14 iteration for the distributed Newton method, for the lossless and lossy formulations, respectively. The Jacobi method does not converge using the branch formulation, and takes around 300 iterations using the nodal formulation. The distributed Newton method takes a little over 100 iterations.

Additionally, the accuracy of the Jacobi method is even lower than in the 3 bus case, deviating from the solution by as much as a degree. Table 7.9 shows the nodal phase angles obtained from iterating the Jacobi method until variable differences between iterations were smaller than 10^{-5} degrees, requiring 338 iteration. This set of voltage phase angles only satisfies power flow equation up to a tolerance of 10^{-1} p.u., which is very inaccurate. If iterated until differences between

Table 7.1: AC, Complex Power Flow for the 3 and 14 Bus Examples

Number of Iterations to Solve AC, Complex Power Flow

	Newton Method	Jacobi Method	Distr. Newton Method
Nodal Formulation	3 bus: 4, 14 bus: 7	3 bus: 24, 14 bus: 534	NA
Branch Formulation	3 bus: 7, 14 bus: 10	DC	NA
Opt.-based Formulation	3 bus: 4, 14 bus: 14	NA	3 bus: 12, 14 bus: 355

NA: Not Applicable, IA: Infeasible Answer, DC: Doesn't Converge

Table 7.2

iterations were smaller than 10^{-20}, then the solution would not become more accurate, even after 1791 iterations.

7.1.3 Flores 45 Bus

The 45 bus example is a distribution network that supplies the power on Flores Island, one of the islands of the Azores archipelago [41]. In this example, the renewable wind resources will be neglected. This leaves a radial network of distribution lines connected to a central generator bus. Figure 7.3 shows a system diagram, with bus 1 being the slack bus, and the location of the generator. There are 4 main sets of distribution lines originating from the generator; the Santa

Table 7.3: Decoupled, Lossy Real Power Flow for the 3 and 14 Bus Examples

Number of Iterations to Solve Decoupled, Lossy Real Power Flow

	Newton Method	Jacobi Method	Distr. Newton Method
Nodal Formulation	3 bus: 2, 14 bus: 3	3 bus: 8, 14 bus: 338	NA
Branch Formulation	3 bus: 3, 14 bus: 3	IA	NA
Opt.-based Formulation	3 bus: 3, 14 bus: 10	NA	3 bus: 6, 14 bus: 118

NA: Not Applicable, IA: Infeasible Answer, DC: Doesn't Converge

Table 7.4

Cruz I, Santa Cruz II, Ponta Delgada and Airport lines. Santa Cruz I includes buses 2 to 6, Santa Cruz II includes buses 7 to 16, the Airport line includes buses 17 to 40, and Ponta Delgada includes buses 41 to 45. The normally open switches and circuit breakers are assumed to be in their normal positions.

The power flow solution to the Flores 45 bus network is found under normal conditions, and the bus voltages are shown on Table 7.10. The centralized optimization-based Newton method took 9 iterations, and the distributed Newton method took 561 iterations.

Table 7.5: Decoupled, Lossless Real Power Flow for the 3 and 14 Bus Examples

Number of Iterations to Solve Decoupled, Lossless Real Power Flow

	Newton Method	Jacobi Method	Distr. Newton Method
Nodal Formulation	3 bus: 2, 14 bus: 3	3 bus: 8, 14 bus: 327	NA
Branch Formulation	3 bus: 3, 14 bus: 3	IA	NA
Opt.-based Formulation	3 bus: 3, 14 bus: 10	NA	3 bus: 6, 14 bus: 110

NA: Not Applicable, IA: Infeasible Answer, DC: Doesn't Converge

Table 7.6

7.2 Targeted Load Shedding and Reactance Adjustment

7.2.1 3 Bus Example

Simulations were carried out on 3 bus network shown in Figure 7.1, with the two loads' complex powers increased by a factor of 10. This new system is infeasible, and Matpower fails to find a power flow solution. The centralized and distributed Newton method using the optimization-based algorithm converge in 45 and 227 iterations, respectively. The reactance of line 1,2 is reduced from 0.1 to 0.03 p.u., the reactance of line 2,3 is reduced from 0.1 to 0.07 p.u., and the reactance of line 1,3 is reduced from 0.1 to 0.06 p.u. The bus voltages are $0.92\angle -10.62°$ p.u. on bus 2 and

Bus No.	Voltage
1	$1\angle 0°$
2	$0.902\angle -4.794°$
3	$0.799\angle -15.221°$
4	$0.821\angle -11.658°$
5	$0.840\angle -9.653°$
6	$0.758\angle -19.421°$
7	$0.767\angle -17.221°$
8	$0.767\angle -17.221°$
9	$0.742\angle -20.453°$
10	$0.734\angle -20.853°$
11	$0.741\angle -20.399°$
12	$0.737\angle -21.121°$
13	$0.730\angle -21.262°$
14	$0.710\angle -22.890°$

Table 7.7: AC Power flow solutions of the IEEE 14 bus system with only PQ buses.

Bus No.	Lossy $\delta(^\circ)$	Lossless $\delta(^\circ)$
1	0	0
2	-5.4189	-5.5070
3	-13.9024	-14.1033
4	-11.3448	-11.5329
5	-9.6837	-9.8707
6	-15.7545	-15.9411
7	-14.8298	-15.0181
8	-14.8298	-15.0181
9	-16.6623	-16.8507
10	-17.0035	-17.1923
11	-16.6360	-16.8255
12	-17.0593	-17.2541
13	-17.5252	-17.5538
14	-18.7236	-18.7530

Table 7.8: Decoupled Lossy and Lossless Power flow solutions of the IEEE 14 bus system.

Bus No.	Jacobi Nodal Lossy $\delta(^\circ)$
1	0
2	-5.8185
3	-14.5590
4	-11.9518
5	-10.2656
6	-16.3805
7	-15.4443
8	-15.4433
9	-17.2835
10	-17.6257
11	-17.2625
12	-17.7019
13	-18.0046
14	-19.2037

Table 7.9: Results of Jacobi method using nodal formulation of decoupled, lossy power flow on the 14 bus system.

Bus No.	V_m (p.u.)	V_a (deg.)	Bus No.	V_m (p.u.)	V_a (deg.)
1	1	0			
2	0.9967	-0.1046	24	0.9624	-1.2530
3	0.9961	-0.1231	25	0.9622	-1.2609
4	0.9954	-0.1441	26	0.9621	-1.2631
5	0.9948	-0.1630	27	0.9621	-1.2635
6	0.9943	-0.1658	28	0.9705	-0.9263
7	1.000	-0.0000	29	0.9703	-0.9271
8	0.9965	-0.1110	30	0.9702	-0.9304
9	0.9932	-0.1277	31	0.9630	-1.1745
10	0.9927	-0.1330	32	0.9630	-1.1767
11	0.9925	-0.1367	33	0.9627	-1.1863
12	0.9924	-0.1398	34	0.9623	-1.1944
13	0.9921	-0.1422	35	0.9632	-1.1655
14	0.9920	-0.1438	36	0.9627	-1.1843
15	0.9919	-0.1445	37	0.9623	-1.1967
16	0.9919	-0.1445	38	0.9604	-1.2614
17	0.9854	-0.4516	39	0.9601	-1.2723
18	0.9723	-0.8659	40	0.9599	-1.2752
19	0.9689	-0.9764	41	0.9962	-0.0746
20	0.9666	-1.0517	42	0.9956	-0.0869
21	0.9658	-1.1002	43	0.9948	-0.1019
22	0.9641	-1.1984	44	0.9944	-0.1141
23	0.9628	-1.2406	45	0.9947	-0.1033

Table 7.10: Power flow results of the Flores 45 bus distribution network.

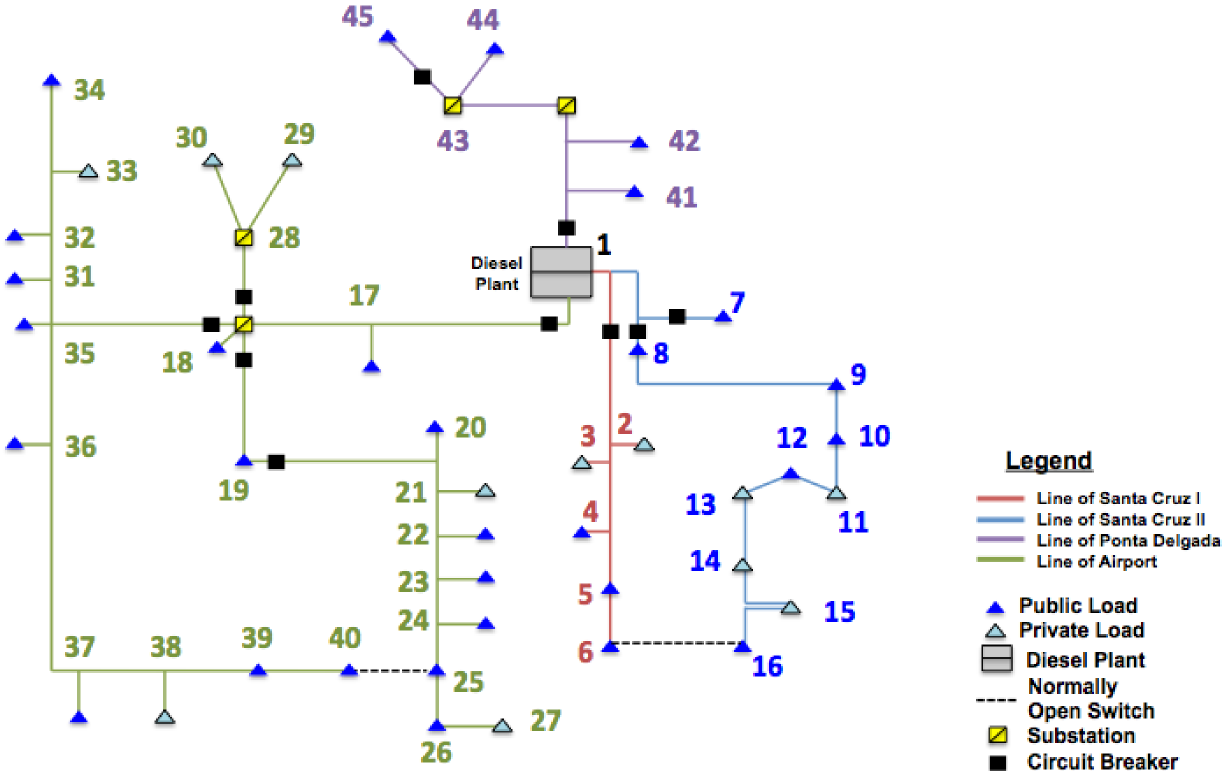


Figure 7.3: The 45 bus distribution network of Flores island.

$0.91\angle 11.44°$ p.u. on bus 3. Magnitude of λ is less than 10^{-2}. This results in a system whose λ is less than 10^{-2} in magnitude.

If the line reactances are not adjusted, the centralized and distributed Newton methods take 156 and 164 iteration, respectively, to converge. The resulting mismatch, $AA^T\lambda$, is 4.71 + 1.82j p.u. at bus 2 and -5.27 + 3.38j p.u. at bus 3. By applying these mismatches to the bus injections, a new feasible system is created. This system is solved in 99 iteration through the centralized Newton method, and in 257 iterations through the distributed Newton method. The new bus voltages are $0.89\angle -35.55°$ p.u. on bus 2 and $1.02\angle -2.11°$ p.u. on bus 3. Magnitude of λ is less than 10^{-2}.

7.2.2 14 Bus Example

Simulations were carried out on the IEEE 14 bus network [39], shown in Figure (7.4). The two lines indicated by red circles in Figure (7.4) were changed so that the reactances are a hundred times

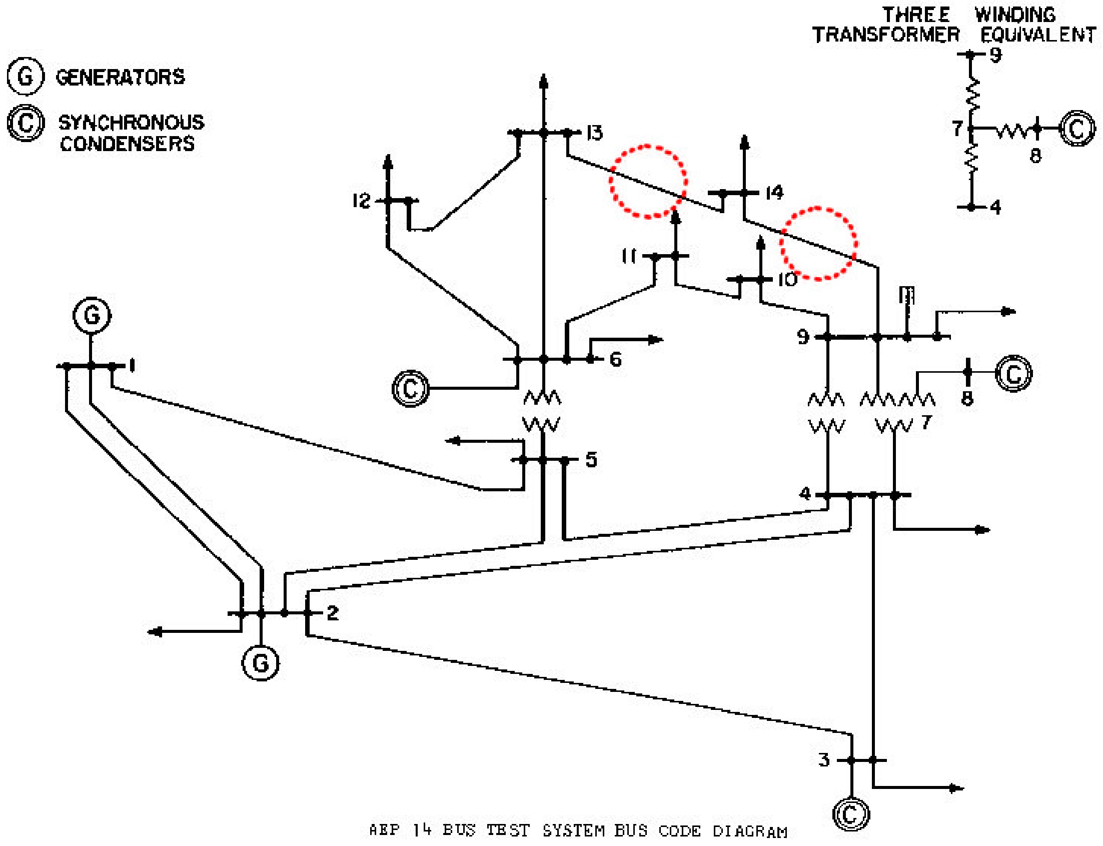


Figure 7.4: The IEEE 14 bus network, with two lines selected to have much higher reactance.

higher than their original values. The reactance of line 9 to 14 is changed from 0.27 to 27 p.u. and the reactance of line 13 to 14 is changed from 0.34 to 34 p.u. This is to create a situation where power cannot be delivered to bus 14, which is only connected to the rest of the network through the two adjusted lines.

This new network was entered into the Matpower solver [40], and the solver fails to converge. This is expected, because unreasonably high reactances were picked so that power could not be delivered. This creates the situation where the whole system has no power flow solution, because of one portion of the network.

The distributed Newton method algorithm takes 341 iterations to arrive at a solution. The changes in reactance of the transmission lines are -22 p.u. for the line connecting buses 9 and 14, and -34 p.u. for the line connecting buses 13 and 14. The new reactance values of these lines are

Bus No.	V_m (p.u.)	V_a (deg.)
1	1	0
2	0.894	-4.755
3	0.784	-15.454
4	0.803	-11.691
5	0.824	-9.642
6	0.722	-20.268
7	0.737	-17.587
8	0.737	-17.587
9	0.706	-21.111
10	0.698	-21.606
11	0.705	-21.225
12	0.696	-22.152
13	0.686	-22.199
14	0.520	-45.400

Table 7.11: Power flow solutions of the IEEE 14 bus system with adjusted reactances.

5.04 p.u. and 0.80 p.u., respectively. The centralized Newton method using the optimization-based formulation does not converge in this case. Keeping the off-diagonal terms of the Hessian results in the Hessian being close to singular or badly scaled in Matlab after 85 iterations.

After lowering the reactance values, the distributed solver was able to reach a power flow solution, shown on Table 7.11. The bus voltage magnitude becomes very low and the bus voltage angle deviates greatly from that of its neighbors. The distributed solution is within 10^{-3} p.u. of fulfilling Kirchhoff's laws and constituent relationships.

When the reactances of the network are not adjusted, the value of bus mismatches is not zero. The centralized Newton method fails to converge after 10,000 iterations, and the distributed Newton method also fails to converge, unless the damping factor is lowered to 0.1. Then the

Bus No.	$AA^T\lambda$
1	0
2	0.0001 + 0.0004j
3	0.0005 + 0.0004j
4	0.0007 + 0.0007j
5	0.0004 + 0.0008j
6	0.1489 - 0.0959j
7	0.0004 + 0.0009j
8	0.0002 + 0.0003j
9	-0.0183 - 0.0263j
10	0.0005 + 0.0006j
11	0.0006 + 0.0006j
12	-0.1712 + 0.1149j
13	-0.3135 + 0.1661j
14	0.1384 + 0.0444j

Table 7.12: Bus power mismatches of the infeasible IEEE 14 bus system with only PQ buses.

algorithm converges after 8219 iterations, and Table 7.12 lists the complex power mismatches at each bus. When the mismatches are applied to the buses and the algorithm is executed, neither the centralized nor the distributed Newton methods converge.

7.2.3 Flores 45 Bus

Consider the Flores 45 bus distribution network, where the loads are increased by a factor of 10. Matpower fails to find a solution, and the centralized and distributed Newton methods converge in 94 and 5866 iterations, respectively.

λ had magnitudes of up to 0.1 p.u. at buses across the network at convergence, even with large reactance adjustments. The lines connecting buses 1 to 17, 17 to 18, and 18 to 35, and all

Line	ΔX (p.u.)
1-17	-0.1270 - 0.0700j
17-18	-0.1140 - 0.0620j
18-19	-0.0650 - 0.0350j
18-28	-0.1920 - 0.1050j
18-35	-0.1870 - 0.1030j
19-20	-0.0440 - 0.0240j
20-21	-0.0170 - 0.0170j
21-22	-0.0380 - 0.0360j
22-23	-0.0320 - 0.0170j
23-24	-0.0160 - 0.0080j
24-25	-0.0290 - 0.0160j
25-26	-0.0160 - 0.0080j
26-27	-0.0160 - 0.0080j
28-29	-0.0930 - 0.0100j
28-30	-0.0410 - 0.0110j
31-32	-0.0070 - 0.0070j
32-33	-0.0550 - 0.0300j
33-34	-0.1080 - 0.0370j
35-31	-0.0240 - 0.0230j
35-36	-0.0200 - 0.0140j
35-37	-0.0260 - 0.0140j
37-38	-0.1620 - 0.0890j
38-39	-0.0320 - 0.0170j
39-40	-0.0180 - 0.0050j

Table 7.13: Changes in reactance of the infeasible Flores 45 bus distribution system.

the other lines along line of Airport were adjusted such that their reactances were less than 10^{-3} p.u., two orders of magnitude lower than their starting values. 10^{-3} p.u. was also the increment of adjustment used in the reactance adjustment algorithm. λ was greater than $10^{-3} p.u.$, the threshold of convergence of optimality condition for Newton method, on buses 17 through 40. As with the targetted load shed, the line of Airport portion of the network was causing the infeasible power delivery problem.

Using the maximum amount of reactance control allowed, the system could not be adjusted to alleviate the infeasible power delivery. Therefore, in a practical application, some load along line of Airport must be shed to achieve feasible power delivery. However, if the resistance of lines could be decreased, then λ can be decreased to the order of 10^{-5}. Table 7.13 shows the changes in reactance of the lines that are adjusted. The resistances and reactances of the lines along line of Airport are all adjusted to below 10^{-2} p.u., which allows feasible power delivery.

If the reactance and resistance are not adjusted, the centralized and distributed Newton methods converge in 13 and 825 iterations, respectively. Table 7.14 shows the bus mismatches, $AA^T\lambda$, found by the optimization-based algorithms. By adjusting the buses 17 through 40 by the mismatch amounts, a new power flow solution is found, and λ has a magnitude of less than 10^{-3}.

Bus No.	$AA^T\lambda$
17	-0.0008 - 0.0004j
18	-0.0164 - 0.0057j
19	0.0010 + 0.0006j
20	0.0068 + 0.0032j
21	-0.0040 - 0.0009j
22	0.0214 + 0.0131j
23	0.0873 + 0.0292j
24	0.0886 + 0.0279j
25	0.0399 + 0.0129j
26	0.0428 + 0.0136j
27	0.0210 + 0.0066j
28	0.0032 + 0.0013j
29	0.0010 + 0.0004j
30	0.0011 + 0.0004j
31	0.0224 + 0.0080j
32	0.0224 + 0.0082j
33	0.0220 + 0.0078j
34	0.0161 + 0.0053j
35	0.0533 + 0.0198j
36	0.0275 + 0.0097j
37	0.0238 + 0.0087j
38	0.0334 + 0.0106j
39	0.0350 + 0.0112j
40	0.0233 + 0.0073j

Table 7.14: Bus power mismatches of the infeasible Flores 45 bus distribution system.

Chapter 8

Implementation Using NIST SGRS

8.1 NIST Smart Grid in a Room Simulator

8.1.1 Summary of the NIST SGRS Platform

Carnegie Mellon University, in collaboration with the United States' National Institute of Standards and Technology (NIST), is developing a Smart Grid in a Room Simulator (SGRS), a platform for simulating arbitrarily complex power system models with a scalable, distributed design. The basis of the simulator is the object oriented approach to developing physical and information models into implementable modules that exchange relevant data.

Each class definition is defined by the researchers, who must write the internal logic and data exchange based on their own mathematical models, which depend on their particular application. The platform is flexible such that the researcher can write whatever methods, data structures, and time scales desired. The current iteration of class definitions are written in Matlab, and are compiled into executables so that the source code need not be shared. These class definitions are used to construct objects, which can simulate physical components, such as generators, loads, or the transmission system, or information processes, such as the electricity market, or system operator. These objects are executed on individual Matlab processes, and can be assigned to different computers connected through the internet, to simulate the affect of separate entities communicating between each other.

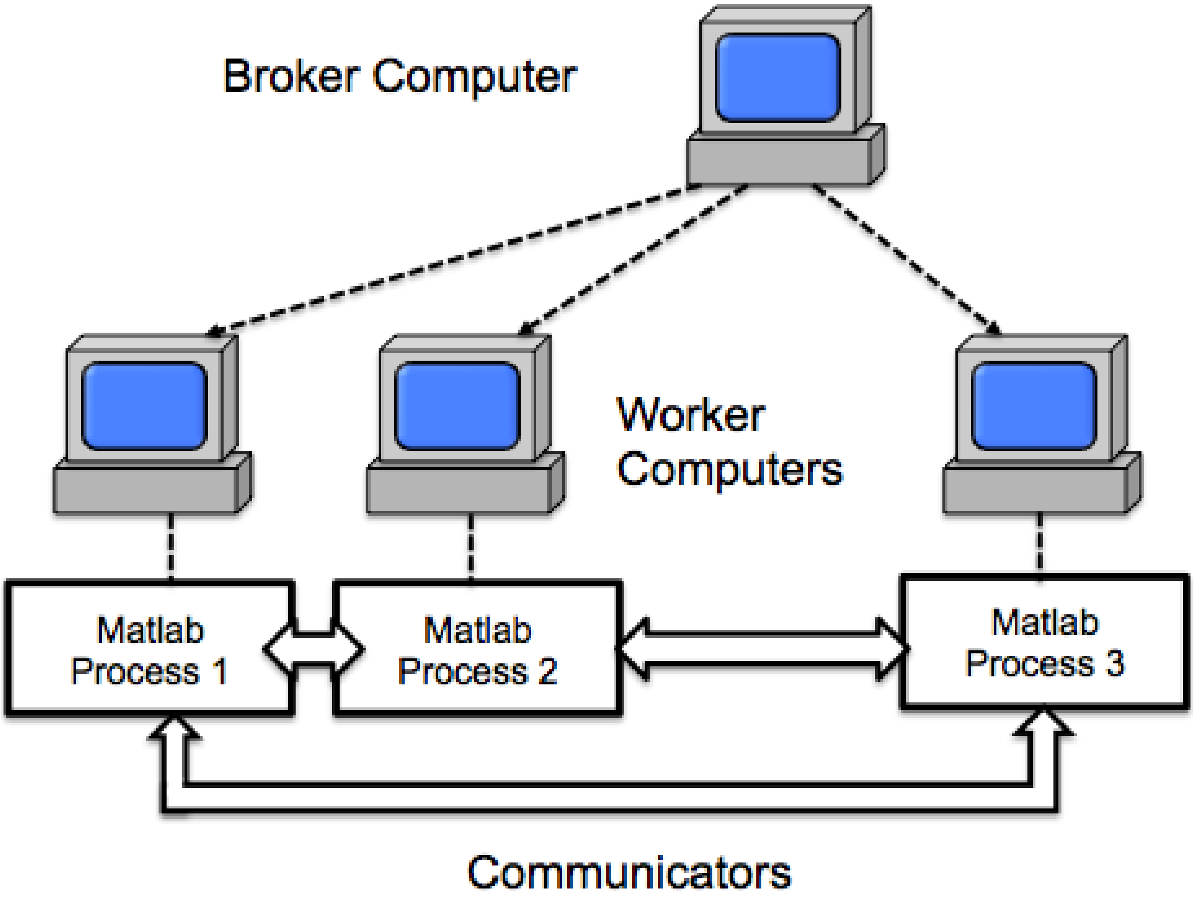


Figure 8.1: The objects are connected by communicators, and allocated computation resources by the broker.

These object-based modules are supported by SGRS communication objects, which can transmit any Matlab struct between any objects without requiring the researcher going into programming details, such as needing to know IP addresses or opening sockets. An SGRS program, called the ”broker”, coordinates the creation and assignment of communicators, so that the modules are already connected by the time the simulation starts. The broker also allocates computational resources to the modules, so that multiple processors and computers can be used to parallelize the distributed modules’ computations. Figure 8.1 illustrates an example where the broker assigns worker computers to execute Matlab processes, which are interconnected by communicators. The role of communicators is to simulate both actual communications between components, as well as exchange data required to simulate the physics of the power system.

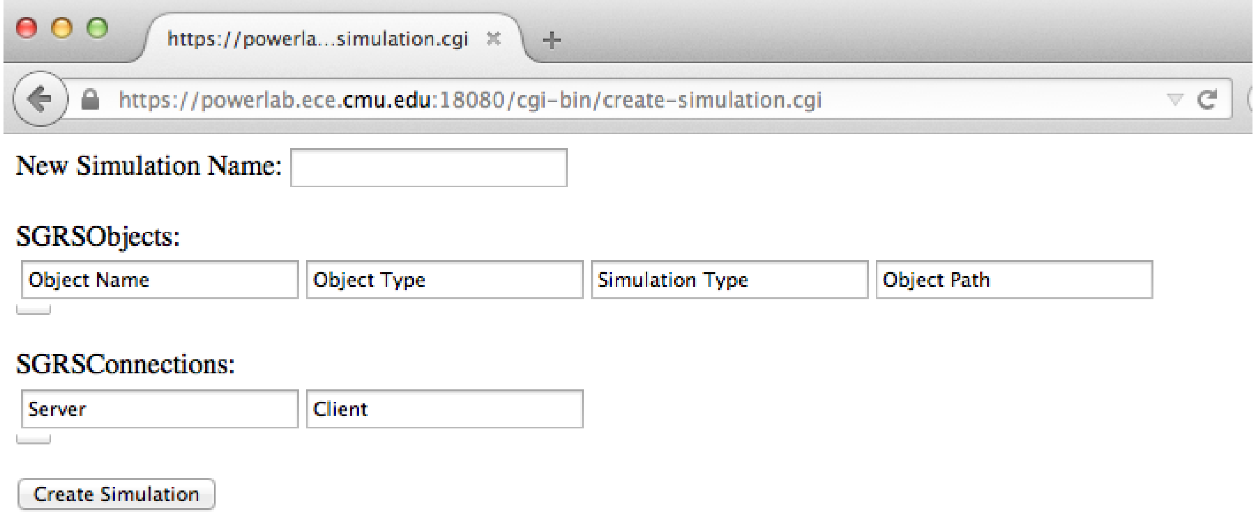


Figure 8.2: The objects and interconnection can be entered through a web browser interface.

The current implementation of the SGRS simulator platform also has a MySQL database, as well as a browser interface, where the module names, file paths, and communications interconnections can be entered. Figure 8.2 shows the current web browser page for creating a simulation. There are efforts in development to automatically create simulations from data entered into a power systems format file, such as PSSE format. There is also a GUI being built that will populate such a format file based on user entries.

8.1.2 SGRS Distributed Power Flow Application

The algorithm proposed in this thesis is to be performed by distributed smart components such as FACTS devices, DSRs, DLRs, etc. The calculations need to take place at the lines and buses of the network, and each line has communicates with their neighboring buses, and vice versa. The information exchange of the algorithm is mapped out in Figure 8.3.

The operations carried out in Equations (4.1.50) to (4.1.53) must be programmed into the class definitions for module creation. Figures 8.4 and 8.5 describe the contents of the bus and line class definitions from the SGRS distributed power flow algorithm, respectively.

Classes consist of properties and methods, and one method must be the class's constructor. The

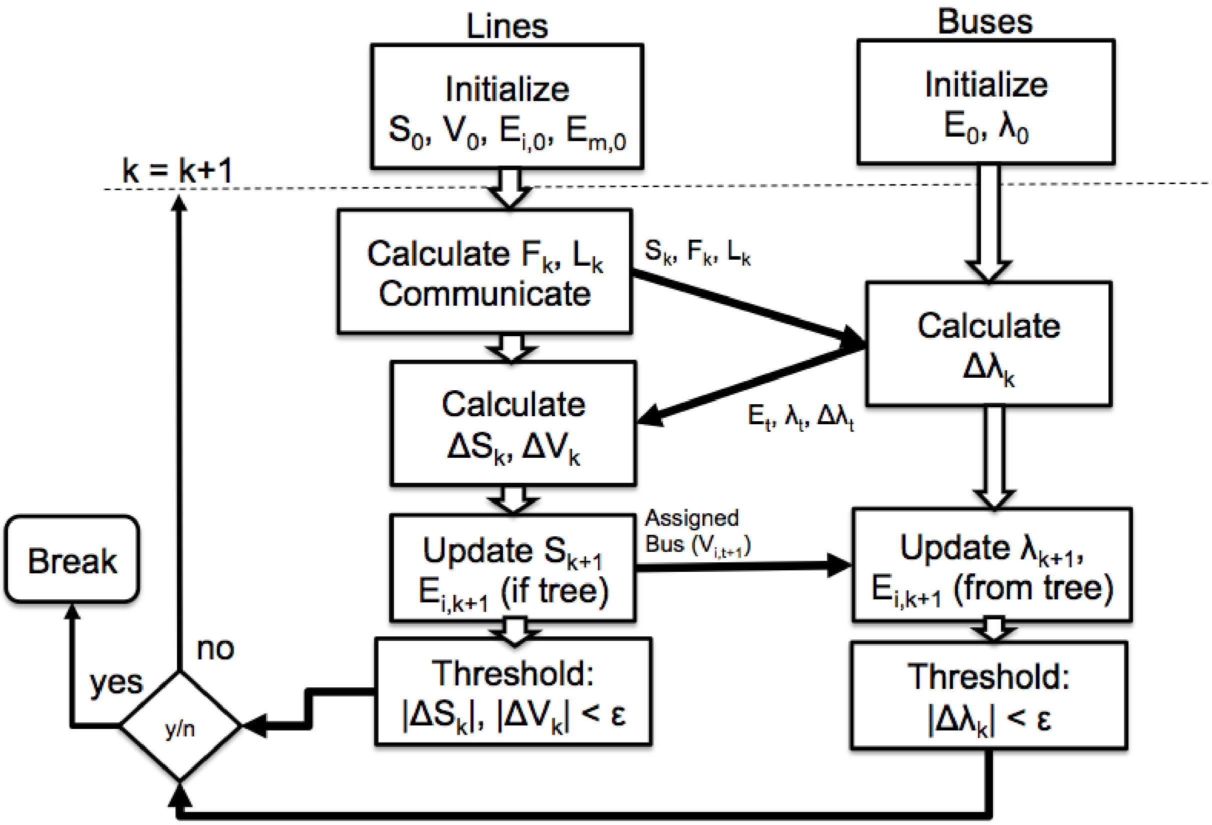
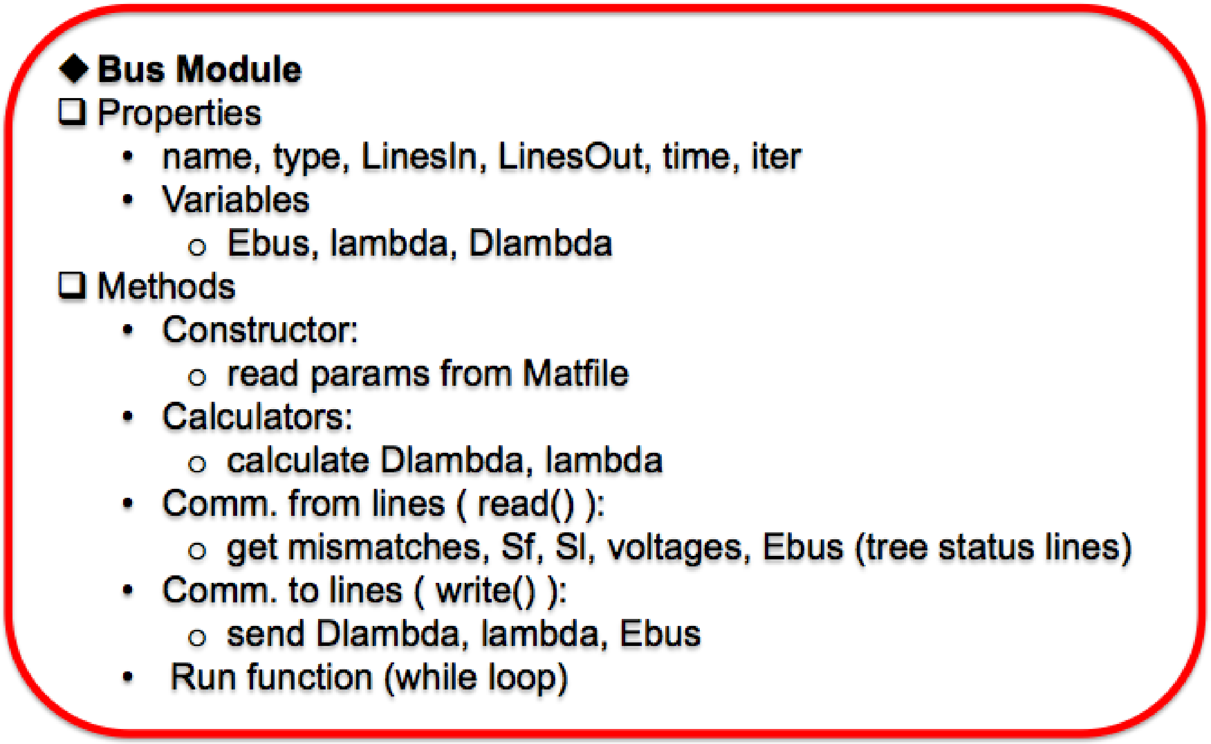


Figure 8.3: A flowchart of the information exchange required to execute the distributed Newton method for complex power flow.

properties are data that is stored locally by the object. All objects store the name and type of the module, and the simulated time. Objects of the distributed power flow simulator store the names of neighboring objects; buses store the names of lines oriented in and out, and lines store the names of the buses at the sending and receiving end. Both the bus and line classes have constructors that read the parameters and initial conditions from Matlab *.mat* files. They also have *run* functions, which are while loops that contain the logical functions of the module. There are also custom *read* and *write* functions, which access the communicators associated with the object and allow information to be sent or received.

Consider a 2 bus power system, shown in Figure 8.6. There are two buses and one transmission line. They are each considered a module, and are named *Bus1*, *Bus2* and *Line1*. Their types are *BusSlack*, *Line*, and *BusPQ*, which describes their behavior.



◆ **Bus Module**

❑ Properties

- name, type, LinesIn, LinesOut, time, iter
- Variables
  - Ebus, lambda, Dlambda

❑ Methods

- Constructor:
  - read params from Matfile
- Calculators:
  - calculate Dlambda, lambda
- Comm. from lines (read()):
  - get mismatches, Sf, Sl, voltages, Ebus (tree status lines)
- Comm. to lines (write()):
  - send Dlambda, lambda, Ebus
- Run function (while loop)

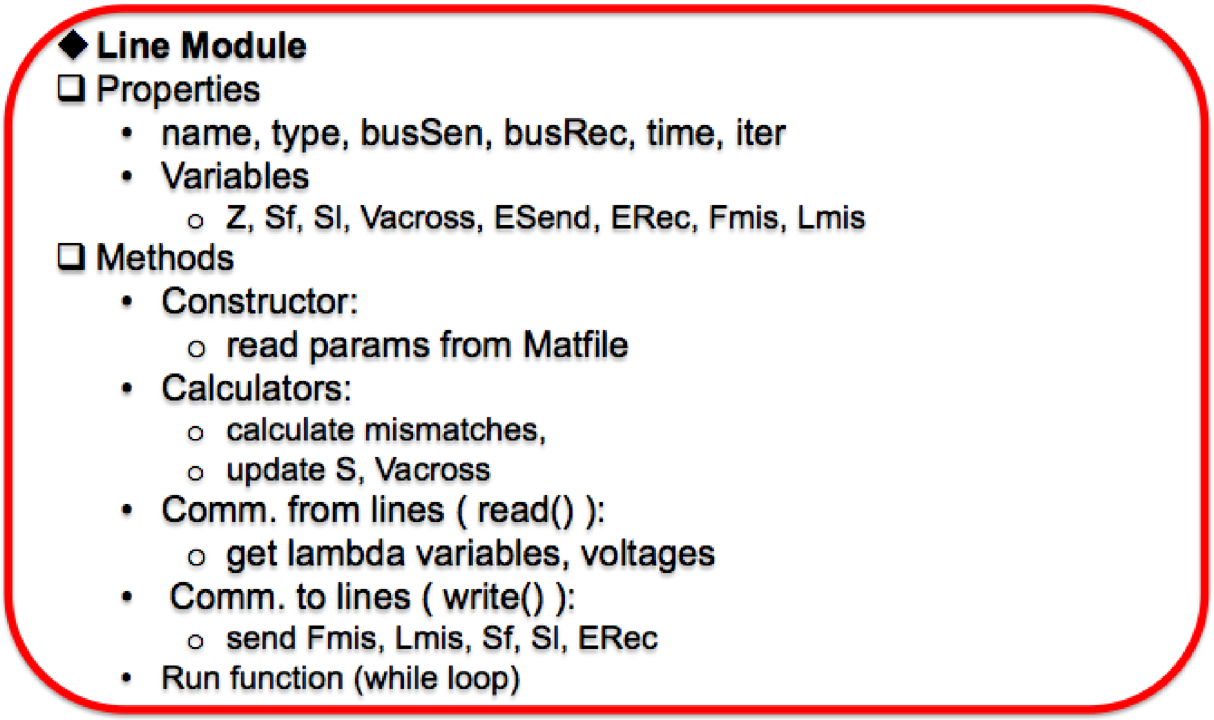
Figure 8.4: The properties and methods of the SGRS distributed power flow bus module.

Figure 8.7 is a table of communications objects required for the 2 bus example. Currently, the communications objects only create a one way communications channel, so each interconnection requires two entries, with reversed destination and sources.

Figure 8.8 illustrates the Matlab processes that would be activated to simulate the distributed power flow algorithm on the 2 bus system. The three processes are started by the broker, and each instantiate an object after accessing the compiled bus or line class definition. Then, each object accesses the locally stored parameter data that matches the object's name. Then, as in Figure 8.1, the broker connects the objects by constructing communicator objects. Then the objects run their *run* functions, and execute the algorithm.

8.1.3 NIST SGRS DPF Contribution

The distributed power flow (DPF) objects and scripts developed for NIST SGRS will be documented and archived for future users. It currently includes four class definitions; a slack bus, a PQ



◆ **Line Module**

❑ Properties

- name, type, busSen, busRec, time, iter
- Variables
  - o Z, Sf, Sl, Vacross, ESend, ERec, Fmis, Lmis

❑ Methods

- Constructor:
  - o read params from Matfile
- Calculators:
  - o calculate mismatches,
  - o update S, Vacross
- Comm. from lines (read()):
  - o get lambda variables, voltages
- Comm. to lines (write()):
  - o send Fmis, Lmis, Sf, Sl, ERec
- Run function (while loop)

Figure 8.5: The properties and methods of the SGRS distributed power flow line module.

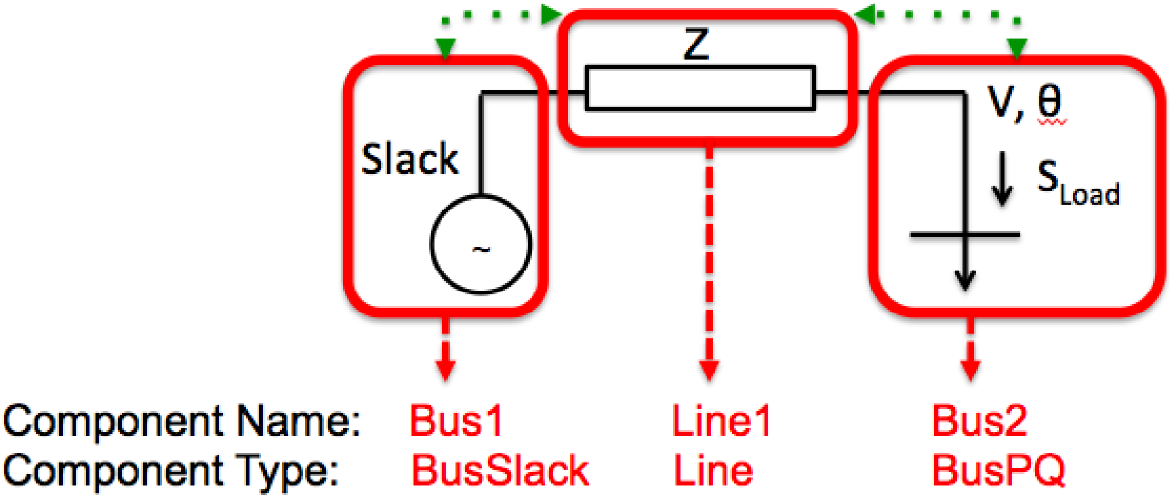
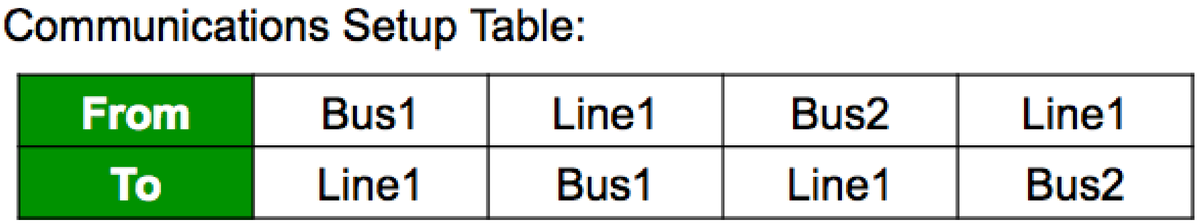


Figure 8.6: A 2 bus example divided into modules for the NIST SGRS simulator.

bus, a passive line and a reactive line. It is possible to combine the Matlab code into two class definitions that are distinguished by the *type* property used to construct the object. The current



Communications Setup Table:

From	Bus1	Line1	Bus2	Line1
To	Line1	Bus1	Line1	Bus2

Figure 8.7: The communications between modules in the 2 bus example.

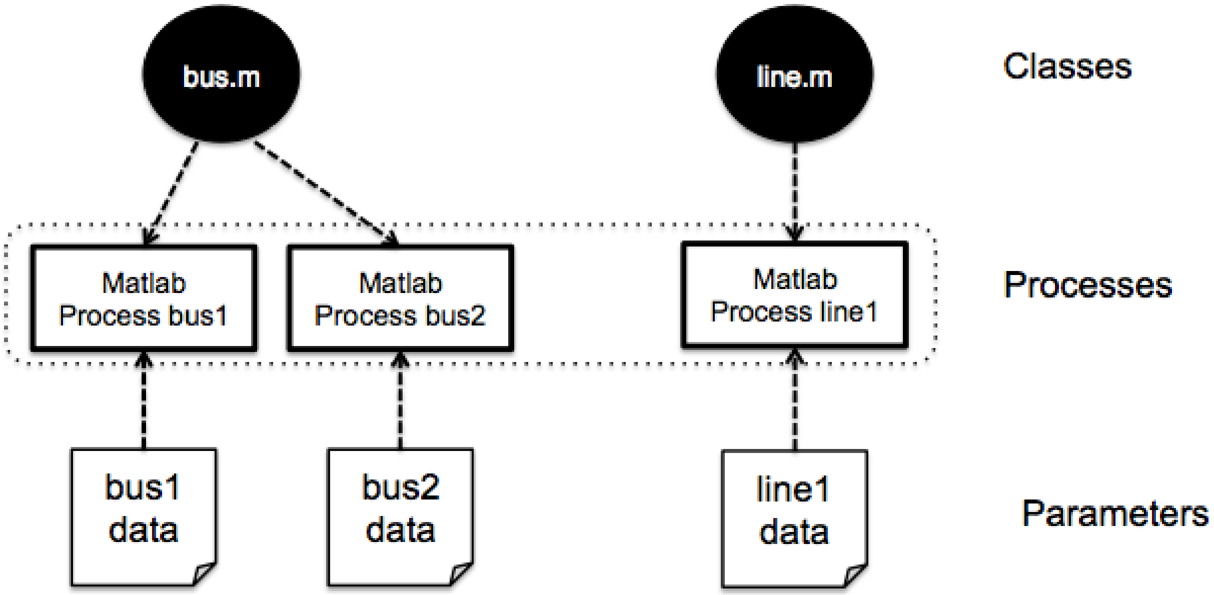


Figure 8.8: A diagram of the processes used to simulate distributed power flow on the 2 bus system.

implementation can be executed in the browser interface.

Chapter 9

Conclusions and Future Work

9.1 Conclusion

This thesis proposes a framework to analyze power systems problems with respect to the penetration of distributed, smart devices, especially transmission/distribution line technologies such as flexible AC transmission systems (FACTS), dynamic line rating units (DLRs), and distributed series reactance (DSRs). The aim is to determine if intelligence can be designed for these devices such that the efficiency and reliability of the power system can be improved. Algorithms in this work are shown to calculate AC power flow without relying on a system operator knowing global information of the network, but by local calculators on buses and lines, communicating with their neighbors and computing autonomously.

The models used to develop these algorithms are derived by formulating steady state power flow problems using a directed graph representation of power flows and voltages. The branches of these directed graphs are used to model components of the power system, while the nodes represent the points of connection, or buses, of the network. The local behavior of components, such as constituent relationships of lines, is integrated through the rest of the system through the graph's incidence matrix, which gives the structure of the graph. Using this structure, an optimization-based mathematical model of power flows is derived that can be decomposed so that distributed calculation is possible.

In addition to calculating power flow, this framework is used to tackle the problem of infeasible power delivery, which results from a lack of power flow solution. The optimization-based formulation provides two approaches to this problem. The Lagrange multipliers that result from solving the optimization-based power flow problem can be nonzero, indicating that the solution is not a power flow solution, but a solution to the optimality conditions of the problem. However, because the structure of the network is preserved in the optimality conditions, the Lagrange multipliers can be used to calculate power mismatches at each bus, allowing targeted load shedding.

The second approach is to use the infeasible power delivery condition of the 2 bus AC power flow example to determine if each line on the system can support power flow through itself, given the voltages at either end. This condition can be checked iteratively during the distributed algorithm, so that lines can adjust their own reactances if FACTS or other hardware is available. This can save resources in cases of peak demand, because the most vulnerable lines would be identified, which is not possible using contemporary power flow algorithms, such as using Newton-(Raphson) method to solve for the nodal voltages.

Both methods of addressing feasible power delivery can be implemented in a centralized or distributed setup, either through the system operators or by smart devices.

9.2 Open Questions

The possible implementations of these methods on real world systems depend on the roles of centralized system operators and distributed smart devices in the future. Conventional transmission level operators may find new tools derived from algorithms for ensuring feasible power delivery. The results of these calculations can guide operators to the critical locations of the network when there is a danger of voltage collapse.

Currently, lower voltage levels, such as distribution networks or microgrids, may not have the control centers required to perform centralized decision making for the network. There may be a need for autonomous or self-adjusting networks of smart devices to perform calculations or even self adjust to ensure feasible power delivery.

The use of branch-based voltages may be useful in a wider range of electrical engineering applications, such as in vehicles such as ships and aircraft. The graph-representation can be used even for DC models instead of AC. The optimization-based formulation is useful for relaxing a simultaneous equations problem, such as finding electrical equilibrium points.

There are many practical open questions towards implementation, especially for the distributed algorithms, such as operating in an asynchronous environment, and recognizing convergence of the global algorithm using only local information. Some tasks require a sequential sweep of a network, such as developing the spanning tree, and this may not be viable in a distributed way on much larger systems.

9.3 Future Work

The optimization-based power flow currently solves AC power flow for systems with a slack bus and PQ buses. This is because the nodal injections are all given complex values. The algorithm needs to be extended to include PV buses, which constrain real power and voltage magnitude, instead of real and reactive power. This is easier to resolve in the conventional AC power flow formulation, where nodal real and reactive power balance are separated into two real valued functions, and the voltage magnitude and phase angle are separate real valued variables. However, this is harder to distribute, because the constituent relationships are functions of nodal voltage magitudes, but also sines and cosines of phase angle differences across flow branches.

Further examination can be applied to the convergence properties of the distributed Newton method, as well as checking other numerical methods to solving the decomposed subproblems of the optimization-based AC power flow. As the algorithm is implemented in an actual distributed setting, such as the NIST SGRS simulator platform, the damping factors may have to be adjusted autonomously by each local component, to either ensure convergence or speed up the algorithm. The convergence properties may also be addressed using preconditioning of the Hessian matrix [42].

The theoretical work about ensuring feasible power delivery through targeted load shed and line reactance control need to be refined such that resources be allocated optimally. Changing the optimization function of the optimization-based algorithm or adding minimal coordination may be explored. In the Flores Island 45 bus case, the adjustment of line reactance isn't enough to guarantee feasible power delivery if the load increases by a factor of 10. Therefore, some optimal compromise between load shedding and reactance adjustment is required.

Appendices

A.1 Numerical Methods and Conditions for Convergence

A.1.1 Newton-(Raphson) Method

The Newton-(Raphson) method is an iterative method for solving nonlinear simultaneous equations. This is commonly used to solve the conventional, nodal formulation of power flow [2]. Newton method uses the slope of the function, or Jacobian, in the multivariate case, to estimate where the zeros of the functions are [32].

Consider a nonlinear vector function $f(x) = 0$, the corresponding Newton step at iterative step k is

$$\Delta x^{(k)} = -(f'(x^{(k)}))^{-1} f(x^{(k)}) \tag{A.1.1}$$

where $f'(x^{(k)})$ is the Jacobian of the function at step k. Using this Newton step, an iterative update,

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)}, \tag{A.1.2}$$

can be performed until either $|\Delta x^{(k)}|$ is sufficiently close to zero, or $|f(x^{(k)})|$ is sufficiently close to zero. The threshold condition used in this work is the latter, or the closeness to satisfying power flow equations. In the case of complex variables, the Taylor expansion is [31],

$$\frac{\partial f(z^{(k)})}{\partial z^{(k)}} \Delta z^{(k)} + \frac{\partial f(z^{(k)})}{\partial (z^{(k)})^*} \Delta (z^{(k)})^* = f(z^{(k)}) \tag{A.1.3}$$

The Kantorovich theorem gives a sufficient condition for convergence [33],

$$h_0 := \gamma\omega_0 \leq \frac{1}{2}, \tag{A.1.4}$$

where

$$\gamma \geq ||f'(x^{(0)})^{-1}f'(x^{(0)})|| \tag{A.1.5}$$

and ω_0 is the Lipschitz constant of $f'(x^{(k)})$.

A.1.2 Jacobi Method

The Jacobi method is a method for solving systems of equations that updates each variable as a function of the variables it is locally coupled with. When solving linear systems of equations with Jacobi method, the algorithm separates the system matrix into diagonal and off-diagonal elements, and iterates by only inverting the diagonal, thus preserving the system structure [34]. To contrast, the Newton method, which inverts the Jacobian matrix, loses system structure, especially when the Jacobian is sparse.

Jacobi method can be generalized to nonlinear systems of equations. For $f(x) = 0$, where $x = [x_1, x_2, ...x_i...x_{m-1}, x_m]$,

$$x_i^{(k:s)} = f_i(x_1^{(k)}, x_2^{(k)}, ...x_{i-1}^{(k)}, x_{i+1}^{(k)}, ...x_m^{(k)}), \tag{A.1.6}$$

where k is the iterative step number, and m is the length of vector m. $x_i^{(k:s)}$ refers to the solved value for x_i at step k. This is then combined with $x_i^{(k)}$ to get the iterative update,

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \omega x_i^{(k:s)}, \tag{A.1.7}$$

where $0 < \omega \leq 1$. The Jacobi method can be executed in a distributed way, in cases where $x_i^{(k:s)}$ only depends on variables from its own component or directly neighboring components. In power flow, the nodal voltage updates depend on the nodal voltages one branch away. The convergence condition of nonlinear Jacobi method is strict diagonal dominance, or strict block diagonal dominance, of the Jacobian matrix $f'(x^{(k)})$ [35].

Bibliography

[1] Stott, B. ”Review of Load Flow Calculation Methods,” Proceedings of the IEEE, vol. 62, No. 7, July 1974

[2] Ilić, M and Zaborszky, J. *Dynamics and control of large electric power systems*, New York; Chichester : Wiley, 2000

[3] Gan, L and Low, S. H., ”Optimal Power Flow in Direct Current Networks. IEEE Transactions on Power Systems,” 29 (6). pp. 2892-2904, 2014

[4] Low, S. H., ”Convex Relaxation of Optimal Power Flow Part I: Formulations and Equivalence. IEEE Transactions on Control of Network Systems,” 1 (1). pp. 15-27. ISSN 2325-5870, 2014

[5] Madani, R., Sojoudi, S., and Lavaei, J., ”Convex Relaxation for Optimal Power Flow Problem: Mesh Networks,” IEEE Transactions on Power Systems, vol. 30, no.1, pp. 199-211, 2015

[6] Baran, M., Wu, F., ”Network reconfiguration in distribution-systems for loss reduction and load balancing.” IEEE Transactions on Power Delivery,4, 14011407, 1989

[7] Desoer, C.A., ”A maximum power transfer problem,” Circuits and Systems, IEEE Transactions on , vol.30, no.10, pp.757,758, Oct 1983

[8] Ilić-Spong, M.; Zaborszky, J., ”A Different Approach to Load Flow,” Power Apparatus and Systems, IEEE Transactions on , vol.PAS-101, no.1, pp.168,179, Jan. 1982

[9] Ilić, M. and Hsu, A. ”Toward Distributed Contingency Screening Using Line Flow Calculators and Dynamic Line Rating Units (DLRs)” , Hawaiian International Conference of System Sciences (HICSS). Maui, Hawaii, January 2012.

[10] Hsu, A. and Ilić, M. "Distributed Newton Method For Computing Real Decoupled Power Flow In Lossy Electric Energy Networks" , North American Power Symposium (NAPS), 2012 , vol., no., pp.1-7, 9-11 Sept. 2012.

[11] Ilić, M. and Hsu, A. "General Method For Distributed Line Flow Computing With Local Communications In Meshed Electric Networks", Patent application 20130024168, January 24, 2013

[12] Hsu, A., Ilic, M. "Calculating Complex Power Flow In AC Electric Power Networks Using Smart Wires", To be submitted, 2015

[13] Jadbabaie, A., Ozdaglar, A.E., and Zargham, M., "A distributed newton method for network optimization. " In Proceedings of CDC, 2736-2741. 2009

[14] Hsu, A., Ilic, M. "Ensuring Feasible Power Delivery Using An Optimization-based Power Flow Model", Techcon 2013, Austin, Tx. Sept 2013

[15] Monticelli, A., "Electric power system state estimation," Proceedings of the IEEE , vol.88, no.2, pp.262,282, Feb. 2000

[16] "Steady-State Power System Security Analysis with PowerWorld Simulator, S6: Voltage Stability Using PV Curves", PowerWorld Corporation, 2014. Powerpoint Presentation

[17] Z. de Souza, F. W. Mohn, "Using PV and QV curves with the meaning of static contingency screening and planning", Electric Power Systems Research, Volume 81, Issue 7, Pages 1491-1498, 2011

[18] Bindal, R. K. "A Review of Benefits of FACTS Devices in Power System," International Journal of Engineering and Advanced Technology, Vol. 3, No. 4, ISSN 2249-8958, April 2014

[19] Glanzmann, G., Andersson, G., "Using FACTS devices to resolve congestions in transmission grids," CIGRE/IEEE PES, 2005. International Symposium , vol., no., pp.347,354, 7-7 Oct. 2005

[20] Kreikebaum, F.; Das, D.; Yang, Y.; Lambert, F.; Divan, D., ”Smart Wires A distributed, low-cost solution for controlling power flows and monitoring transmission lines,” in Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES , vol., no., pp.1-8, 11-13 Oct. 2010

[21] Mohammadi, J.; Hug, G.; Kar, S., ”A benders decomposition approach to corrective security constrained OPF with power flow control devices,” in Power and Energy Society General Meeting (PES), 2013 IEEE , vol., no., pp.1-5, 21-25 July 2013

[22] Wolf, G., ”T & D World: Dynamic Monitoring”, Smartwires.com, < *http://www.smartwires.com/category/uncategorized/page/4/* >, Accessed October, 2015

[23] Siemens Series Compensation System, Directindustry.com, < *http://www.directindustry.com/prod/siemens-em-transmission-solutions/product-32878-723953.html* >, Accessed October, 2015

[24] Lavaei, J.; Tse, D.; Baosen Zhang, ”Geometry of power flows in tree networks,” in Power and Energy Society General Meeting, 2012 IEEE , vol., no., pp.1-8, 22-26 July 2012

[25] Calvaer, A. J., ”On the Maximum Loading of Active Linear Electric Multiports.” Proc. IEEE, 71: 282-283, 1983

[26] ”Voltage Stability and Reactive Power Planning”, Entergy Transmission Planning Summit, New Orleans, LA, July 8, 2004. Powerpoint Presentation

[27] Dobson, I., Glavitsch, H., Liu, C. C., Tamura, Y. ”Voltage collapse in power systems” Circuits and Devices Magazine, IEEE, 8(3), 40-45. 1992

[28] Maurer, S. , ”Matrix Generalizations of Some Theorems on Trees, Cycles and Cocycles in Graphs,” SIAM Journal on Applied Mathematics, 30:1, 143-148, 1976

[29] Bader, D., Cong, G. ”A fast, parallel spanning tree algorithm for symmetric multiprocessors (SMPs),” Journal of Parallel and Distributed Computing, no. 65, pp. 994-1006, 2005

[30] Gallager, R., Humblet, P., ”A Distributed Algorithm for Minimum-Weight Spanning Trees,” Journal ACM Transactions on Programming Languages and Systems (TOPLAS), vol. 5, issue 1, pp. 66-77, 1983

[31] Sorber, L., Barel, M. V., ”Unconstrained Optimization of Real Functions in Complex Variables,” SIAM Journal on Optimization 22:3, 879-898 , 2012

[32] Boyd S. and Vandenberghe L. *Convex Optimization,* Cambridge University Press, Cambridge, UK, 2004

[33] Deuflhard, P., ”Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms,” Springer Berlin Heidelberg: Springer Series in Computational Mathematics, vol. 35, 2011

[34] Salkuyeh, D. K. ”Generalized Jacobi and Gauss-Seidel Methods for Solving Linear System of Equations,” NUMERICAL MATHEMATICS, A Journal of Chinese Universities (English Series), issue 2, vol. 16: 164-170, 2007

[35] Ili-Spong M., Katz, N., ”Block diagonal dominance for systems of nonlinear equations with application to load flow calculations in power systems,” Mathematical Modelling, Volume 5, Issue 5, Pages 275-297, 1984

[36] Conejo, A.J., Nogales, F. J., Prieto, F.J., ”A Decomposition Procedure Based On Approximate Newton Directions,” Mathematical Programming, Volume 93, Issue 3 , pp 495-515, Sept. 2002

[37] Kermati, B. ” An Approach To The Approximation Of The Inverse Of A Square Matrix By He’s Homotopy Perturbation Method,” Italian Journal Of Pure And Applied Mathematics. Pages 117-124, November 2011

[38] Chung, F. *Spectral graph theory*, Providence, RI: American Math. 1992

[39] Power Test Systems Case Archive. Available Online: http://www.ee.washington.edu/research/pstca/pf14/pg_t

[40] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, Matpower: SteadyState Operations, Planning and Analysis Tools for Power Systems Research and Education, Power Systems, IEEE Transactions on, vol. 26, no. 1, pp. 1219, Feb. 2011.

[41] Ilić, M and Le, X. *Engineering IT-Enabled Sustainable Electricity Servces*, New York, New York; Springer, 2013

[42] M. Benzi, ”Preconditioning Techniques for Large Linear Systems: A Survey”, Journal of Computational Physics, Volume 182, Issue 2, Pages 418-477, 2002