DISTRIBUTED STATE ESTIMATION FOR SMARTER ELECTRIC POWER GRIDS

SHIVAM SAXENA

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE AND ENGINEERING

YORK UNIVERSITY

TORONTO, ONTARIO

MAY 2015

Abstract

The focus of this thesis is to design and implement distributed and decentralized state estimation (SE) algorithms for smart Electric Power Grids (EPGs). These algorithms are applied to two different types of EPGs: 1) modern, deregulated transmission networks that include advanced wide-area monitoring systems, and; 2) smart distribution networks with high penetration of distributed and renewable generation (DG) configured of microgrids. Microgrids are capable of cutting off from the main grid and operating autonomously in the islanded mode of operation in case of emergency situations. SE in such systems is complex because of highly non-linear system dynamics, slow and corrupted measurement updates, as well as the sheer computational complexity of the estimation algorithms. The contribution of this thesis is to explore the design and implementation of a reduced-order, distributed particle filter for state estimation in EPGs. Knowledge of the EPG state is necessary for EPG control, optimization, and emergency troubleshooting.

Dedicated to: Avni

You coming into our lives gave us everything we ever wanted. Wishing you all the best health, happiness, and success Avni.

"I've missed more than 9000 shots in my career. I've lost almost 300 games. 26 times, I've been trusted to take the game winning shot and missed. I've failed over and over and over again in my life. And that is why I succeed."

Michael Jordan

Acknowledgements

Dr. Amir Asif: My favorite part of this whole process will always be sitting in your office during our weekly meetings. The procedure was always the same. You would attempt to explain some complex mathematical theorem and as always I would not understand. You would then go through endless iterations of the same thought, but you would twist it, turn it and make it simpler each time until I finally understood. But you never gave up on me, and I'll always remember that. Your endless patience, good faith and work ethic always inspired me to put my head down and keep working harder everyday. Thank you for giving me your time and knowledge. I hope I have honored it.

Dr. Hany Farag: Dr. Farag, you have helped me with every possible facet of this thesis. From designing and coding the initial Farag 3 Bus Microgrid, to supervising my final year project, to correcting and improving my writing for the papers we published together. Your endless support and guidance are much appreciated. In particular, thank you for teaching me about the smart grid - this contribution of yours is something I will always value forever.

Ma: We made it Ma! I remember that when I was working in the industry I never wanted to do a Masters, but somehow you made me promise I would do one. *Aaj woh sapna pura hota hai Ma.* Thank you for sacrificing so much for all of us - you work so hard, keep our lives organized and give us everything you have. I would be nowhere without your relentless positive energy, guidance, and support. This one's for you Ma. Now don't make me promise to do a PhD!

Dad: Dad you've taught all of us so much over the years, but the number one thing you've taught me is how to respond to challenges. To always give it your best effort, no matter what the circumstances. To always fight hard no matter what your up against. Most importantly, to do so with respect, honor, and a sense of humor. You are an example we all try to follow Dad. Thank you for your teachings, your jokes, and always making me laugh.

Didi: My second mother. Didi, with you around I never had to worry about anything. Thank you for raising me with love, compassion, integrity, courage and respect. Thank you for being that one person who I could always look up to. You are the benchmark that we all strive to touch - whether its academics, whether its music, it could be anything. You mean everything to me. Thanks Didi.

Colleagues: I would really like to take time to appreciate and acknowledge my colleagues in the Signal Processing Lab at York University. Without you this thesis could not have been possible, and I will cherish all our memories together. Special thanks to Albina Rahim, Farzana Yasmeen, Arash Mohammadi and Mohammad Sajjadieh.

Table of Contents

List of Figures

List of Acronyms and Symbols

AC	Alternating Current
BDF	Backward Differentiation Formulas
CPF	Centralized Particle Filter
DPF	Reduced-order, Distributed Particle Filter
DC	Direct Current
DG	Distributed Generator
DKF	Decentralized Kalman Filter
EIF	Extended Information Filter
EKF	Extended Kalman Filter
EnKF	Ensemble Kalman Filter
EMS	Energy Management System
EPG	Electrical Power Grid
GPS	Global Positioning System
MATLAB	Computer software specializing in handling arrays
NDF	Numerical Differentiation Formulas

ODE	Ordinary Differential Equation
ode15s	A stiff ODE solver subroutine used in MATLAB
pdf	Probability Density Function
PF	Particle Filter
PMU	Phasor Measurement Unit
RMSE	Root Mean Square Error
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SE	State Estimation
SNR	Signal-to-Noise Ratio
UI	User Interface
UIF	Unscented Information Filter
UKF	Unscented Kalman Filter
WLS	Weighted Least Squares
B	Susceptance
$\boldsymbol{d}$	Forcing term
$\boldsymbol{F}$	Set of State Equations related to $\boldsymbol{X}$
$\boldsymbol{G}$	Set of observation Equations related to $\boldsymbol{z}$
G	Conductance
$\mathcal{G}_n$	n nodes in neighbouring subsystem

$\{I_{BD_l}, I_{BQ_l}\}$	Line currents in DQ frame
$\{I_{LD_m}, I_{LQ_m}\}$	Load currents in DQ frame
$\{I_{od}, I_{oq}\}$	Output current of DG in dq frame
$\hat{\jmath}$	Imaginary number $\sqrt{-1}$
$\boldsymbol{K}$	Kalman Gain
N_s	Number of particles
n_{sub}	Number of subsystems
n_x	Number of State Variables
P_g	Active Power Generated by DG
$P_n^{(l)}$	Covariance value of n shared states at subsystem l
q	Proposal distribution
$\boldsymbol{Q}$	Process covariance noise
Q_g	Reactive Power Generated by DG
$\boldsymbol{R}$	Process covariance noise
$S^{(l)}$	Subsystem at index l
V	State variable: Voltage
$\{V_{bD_p}, V_{bQ_p}\}$	Nodal voltages in DQ frame
$W_i^{(l)}$	Weight of particle i at subsystem l
$\{W_i\}_{i=1}^{N_s}$	Weights associated with particles
$\boldsymbol{X}$	State Vector

x_n	State Variable n
$X_n^{(\text{fuse})}$	Fused state value of n shared states
Y	Y-Bus
$\boldsymbol{z}$	Observation Vector
$\boldsymbol{z}(1:k)$	Measurements from index 1 to k
z^*	Actual measurement
$\aleph_n$	Neighbourhood of bus n
δ	DG Angle
$\boldsymbol{\zeta}$	Observation Noise
θ	State variable: Phase Angle
$\mu_n^{(l)}$	Mean value of n shared states at subsystem l
$\boldsymbol{\xi}$	Process Noise
ω	State variable: Generator speed
$\mathbb{X}_i^{(l)}$	Particle i at subsystem l
$\{\mathbb{X}_i\}_{i=1}^{N_s}$	Complete set of particles with index i

Chapter 1 - Thesis Overview

1.1 Motivation

It is often taken for granted that at a flick of a switch, there will always be a reliable and uninterrupted source of electrical power. This was not the case, however, during the Northeast American power blackout of 2003 which left over 50 million consumers without power for more than 48 hours. The blackout claimed 10 lives and caused countless injuries, while also totaling an economic loss of approximately 6 billion dollars. The Northeast blackout ranks as the largest blackout faced by the North American people and the seventh largest on a worldwide scale.

In the aftermath of the blackout, the IEEE Power Engineering Society set up a committee to understand the major causes of the system failure and provide recommendations for preventing blackouts in the future [1]. It was found that the starting point of the power outage was an incidental contact between a major high voltage transmission line and unmaintained tree branches which had grown beyond municipal city limits. The contact caused the line to fault. System operators of the EPG are usually notified of such abnormal events by a locally installed Energy Management System (EMS). In particular, the

EMS provides a tool termed as State Estimation (SE), which continually monitors the state and security of the grid. However, at the time of the incident, the SE module of the EMS was inactive due to a software bug and thus the information of the faulted line was not relayed to the system operator. If a robust, efficient and accurate state estimator was in place, the damage caused by the blackout could have been greatly limited.

Furthermore, the infrastructure of the EPG has a number of fundamental issues which include: fossil fuel based power plants which emit massive carbon emissions into the atmosphere, ageing technology which will soon not be able to keep up with increasing power demand, as well as a centralized, hierarchical infrastructure, which does not align well with the current deregulated electricity market sector. Driven by the urgent need to develop cheaper, cleaner, efficient and sustainable electric power grids, the electric power industry is currently undergoing a profound paradigm change towards a smarter grid setup. A smart grid represents a vision for digital upgrades of electric power transmission and distribution. The key to the smart grid utilization is enabling advanced control, communication, computing and monitoring technologies for shuttling numerous amounts of information back and forth between the electric utility sector and its customers. The distributed nature of restructured power systems and the new applications of monitoring and control techniques introduce a different set of indices for measuring the reliability of electric power systems.

The focus of this thesis is to design and implement distributed and decentralized state estimation algorithms for: 1) modern restructured (deregulated) transmission networks with advanced wide-area monitoring systems (i.e., where many fast-information gathering and processing devices such as phasor measurement units (PMUs) are installed), and; 2) smart distribution networks with high penetration of distributed and renewable generation (DG) microgrids that are capable of operating in an islanded mode of operation.

State estimation for these extremely dynamic networks are complex due to high non-linearity in the system model, corrupted measurements often with sources of noise that are non-Gaussian and highly correlated, and the computational complexity of large power networks. This thesis discusses the design and implementation of the particle filter, distributed particle filter, and reduced-order particle filter to address these key challenges [2]-[4]. The particle filter is well equipped to handle non-linear system models and its reduced-order, distributed implementation significantly lowers the overall computational complexity of the centralized particle filter. Both the centralized and distributed particle filters are implemented on two types of test EPGs: the IEEE 14 Bus and a custom 8 Bus microgrid. As part of the modeling and simulation of microgrids, this thesis also contributes a dynamic software toolbox that is used for rapid prototyping of custom microgrids. This is designed as a drag and drop application in which the user can design a microgrid and simulate its behaviour.

1.2 Thesis Structure

The following bulleted list provides a summary of the thesis in terms of its organization on a chapter by chapter basis:

- Chapter 2 explains the structure of a typical electrical power grid and how its components work together to deliver power. In addition, it contains information on smart power grids and recent initiatives in the area followed by an introduction to microgrids.

- Chapter 3 provides an introduction to state estimation techniques that include the Kalman filter for systems with linear state dynamics and the Particle Filter for non-linear systems corrupted with non-Gaussian noise. This chapter also contains a literature review of state estimation algorithms, which have been implemented previously in the field of electric power grids (EPGs).

- Chapter 4 introduces the concept of distributed state estimation techniques. Reduced-order distributed estimation techniques are also discussed and applied to the particle filter. A sample example is also included to illustrate how a reduced-order configuration can be derived for a large system.

- Chapters 5 and 6 apply both the centralized and distributed particle filter to modern, deregulated transmission systems as well as islanded microgrids. The objective is to derive a near-optimal distributed implementation of the distributed particle filter and show that the performance follows that of the centralized implementation at lower computational complexity, with much reduced latency, and higher immunity to failure. The pseudocode for the microgrid system model and flowcharts for the applied filters are also provided in this chapter.

- Chapter 7 concludes the thesis and presents directions for future work that can be considered as an extension of this research.
- The Appendix contains supplementary information to the thesis and is split into 3 parts. In the first part, the toolbox for the modeling and simulation of islanded microgrids is discussed in detail. The software framework, architecture, and simulation results are presented. The second and third parts of the Appendix provides the network configuration and component data for both the modern, deregulated transmission networks as well as the islanded microgrids.

1.3 Thesis Assumptions

The following bulleted list describes the set of assumptions that the thesis is based on:

- The focus of the thesis is on the development of distributed state estimation techniques for systems with non-linear dynamics (such as modern, deregulated power grids and islanded microgrids). The state equations representing both systems are presented as dynamical, time-invariant equations. Effectively, this means that the network topology of the system is static and does not support changes in network configurations. In order to incorporate dynamic topology identification with the implementation of the estimation techniques discussed in this thesis, the topology assessment should take place at the beginning of every iteration of the estimation process. The system equations can then be altered to reflect the changes in topology found by the assessment. The distributed particle filter implementations proposed in the thesis do not change and are generalizable to dynamical systems. Techniques for dynamic topology identification can be found in [5]-[7].

- In the implementation of the distributed, reduced-order particle filter, it is assumed that consensus for the estimates of shared states between subsystems is achieved within 2 successive observations. To include support for situations with intermittent/irregular consensus convergence in this implementation of the distributed, reduced-order particle filter, see [55].

- For the implementations of both the centralized particle filter and the distributed, reduced-order particle filter, the signal-to-noise ratio is kept constant within each Monte Carlo simulation. The goal is to show that the state estimations results from centralized and distributed filters are similar. Under such a scenario, the difference between the real and estimated state values decrease over time as new observations are received and incorporated in the filters. Real systems are often corrupted with a variable amount of noise. In order to add support for a randomized SNR, a random number generator can be used to generate a random number within a designated upper and lower bound. This random number can then be assigned to the SNR parameter at the beginning of every iteration of the estimation process.

Chapter 2 - Introduction to Electrical Power Grids

2.1 Structure of Conventional Electric Power Grids

An electrical power grid (EPG) is a complex system, which generates, transmits, and distributes electricity to a variety of end users over large geographical distances. The EPG can be thought of as a combination of three interconnected networks: the *generation* network, *transmission* network, and the *distribution* network as shown in Fig. 2.1. The primary objective of the generation network is to generate electric power through the use of generators. Generators are housed in industrial facilities known as power stations and are used to convert mechanical, chemical, or nuclear energy into electrical energy. They accomplish this by using an already existing energy source to provide the mechanical energy to rotate a turbine. The turbine is connected to a shaft which consists of powerful magnets that are tightly wrapped in wire. From the Faraday electromagnetic induction law, it is known that electric charges can be induced in a moving electric conductor when immersed in a magnetic field. As such, current is *induced* through the wire because of the rotation of the magnets.

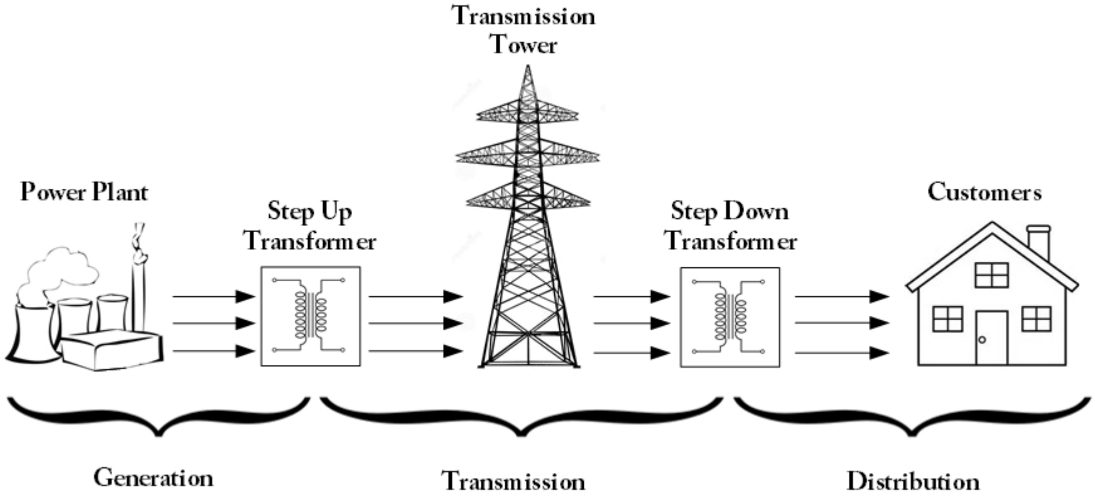


Figure 2.1: The Electrical Power Grid.

The energy sources used to provide energy to the generator can be renewable or non-renewable. Most power stations use non-renewable, thermal sources such as coal, oil and natural gas to power the generators. The thermal source is burned to heat up a reservoir of water, which subsequently evaporates into steam. The steam propels the turbine blades, which in turn rotates the shaft. The constant rotation of the shaft within the magnetic field (provided by the magnets surrounded by heavy coils of copper wire) produces a steady current. The machinery that provides the mechanical energy to propel the turbine is referred to as the *prime mover.*

The process of burning thermal sources, however, is very harmful for the environment due to the fact that it releases carbon dioxide into the atmosphere. This is a major motivating factor for researchers to shift towards the use of renewable energy sources, which have minimum impact on the environment. These energy sources are primarily wind, solar and hydro. The integration of these power sources into the smart grid setup will be discussed further in the later sections of this chapter.

The electricity produced from commercial power plants is typically at a fixed frequency and consists of 3 phases, with each phase being offset by 120 degrees with respect to each other. Once produced, the electricity flows along transmission lines to a transmission substation. Using step-up transformers, the voltage of the electricity is stepped up to extremely high voltages that may reach up to 765 kV. Stepping up the voltage of the electricity is necessary as it results in lower energy losses due to the resistance in the line.

Using these high voltage overhead power lines, electricity then travels from the transmission substations to the distribution substations located near the consumers. The distribution substations step down the voltage to usable levels (e.g., the typical 110V and 240V levels used for household appliances) and distribute power locally. Large factories which have higher power needs may consume power from special distribution substations and use their own step down transformers to step down the voltage to ranges between 4 kV to 69 kV.

In order to control and supervise the generation, transmission, and distribution of electricity, an Energy Management System (EMS) is used. The EMS is a SCADA (Supervisory Control and Data Acquisition) based system, which provides communication channels through Remote Terminal Units (RTUs) for supervisory and control purposes. RTUs are used to measure electrical observations throughout the grid. These measurements include: active and reactive power flows as well as voltage and current magnitudes. The measurements are then typically telemetered to the Data Acquisition module of the EMS and used by the State Estimator to provide an estimate of the state. System operators use the EMS to monitor and optimize the state of the electric power grid, as well as to provide real-time troubleshooting when an important system component fails (Fig. 2.2).

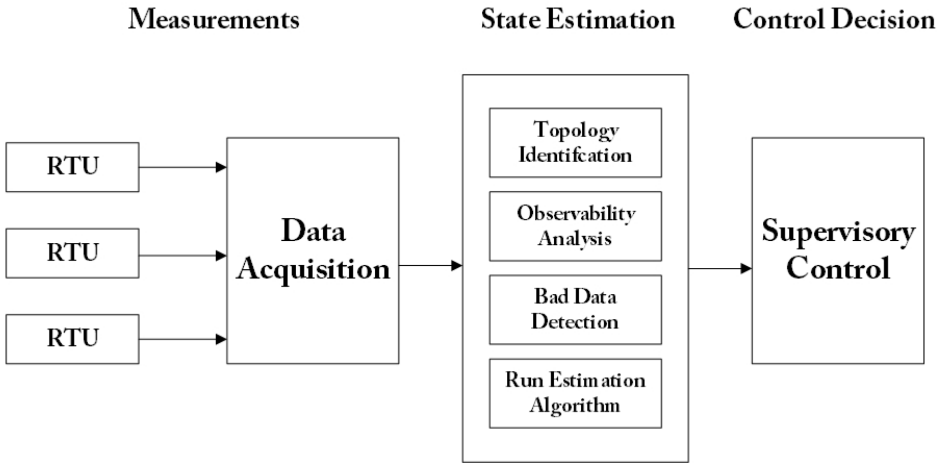


Figure 2.2: State Estimation Within an Energy Management System.

One of the most complex requirements of the EPG is to balance the supply and power demand. It is not economically feasible to store electricity in large amounts within the grid and therefore it must be consumed soon after it is produced [8]. As such, the EPG is driven completely by the demand of its consumers. System operators are required to use the EMS to schedule the power production of the generators in accordance to the demand. This is typically done on an hourly basis and can be predicted from 24 hour forecasts based on historical data. The prediction of the power demand is referred to as the *demand curve*.

2.2 Modern Deregulated Power Grids

Traditionally, a single utility has been responsible for all three networks, (i.e., generation, transmission, and distribution), with the notion being that one entity could better oversee the power demand within the area and also manage resources and prices more effectively and efficiently. This entity could build and produce large scale power plants and cut back on costs by using its own transmission and distribution systems. While being the sole electricity provider in the area makes the system much more reliable, the price of electricity may vary depending on the distributors proximity to the power plant. Historically, distributors closer to generation stations have paid slightly less than those who are located far away [9].

Deregulation of the power system allows the generation network to be "unbundled" from the transmission and distribution networks and also introduces competition in the power generation market. Two significant technological improvements have facilitated this change. New power generation schemes have shifted from large, fossil fuel based generators to smaller, high efficiency gas turbines. This allows smaller companies with viable generation plans to compete in an industry that was once monopolistic. Secondly, improvement in communication methods and transmission infrastructure has allowed power to be transmitted across significant distances in a reliable and efficient manner. Proximity to a power plant does not necessarily imply cheaper electricity.

2.3 Smart Power Grids and its Initiatives

There exist many problems with the current setup of the EPG. First, the infrastructure and machinery of the EPG is ageing rapidly and many key internal components are reaching their end-of-life date. With the worldwide demand for power growing at a steady pace (North American power demand is projected to increase by up to 28% by the year 2040 [10]), the already overloaded power grid is being made to stretch even further. Second, power production still relies heavily on burning fossil fuels that causes severe environmental issues. Third, the centralized and hierarchical structure of the EPG is inefficient, unscalable and vulnerable to failure.

As such, the promise of an electrical power network, which is decentralized, automated, and distributed has come to realization with the concept of the smart grid. The smart grid seeks to decentralize the massive power grid into a network of smaller, more manageable subsystems, each of which services a smaller demand for power. To that end, a great deal of research is being put into using renewable energy sources as viable production alternatives in order to cut down on carbon emissions. Aggregating these energy sources and using their combined power output is known as distributed generation (DG). The aforementioned smaller grids are known as microgrids, which generate power locally through distributed generation and distribute it among local loads as illustrated in Fig. 2.3.

The smart grid brings with it many initiatives in additions to the ones mentioned above. A short list includes: increased reliability, increased efficiency, as well as increased market awareness. Advanced communication channels and devices help the smart grid form a wide area monitoring system to help troubleshoot issues. In addition, the advent of Phasor Measurement Units (PMUs) has provided exciting possibilities with regards to monitoring the grid. PMUs operate at very high sampling rates; often taking 30 measurements per second, which are all synchronized using a built in Global Positioning System (GPS).

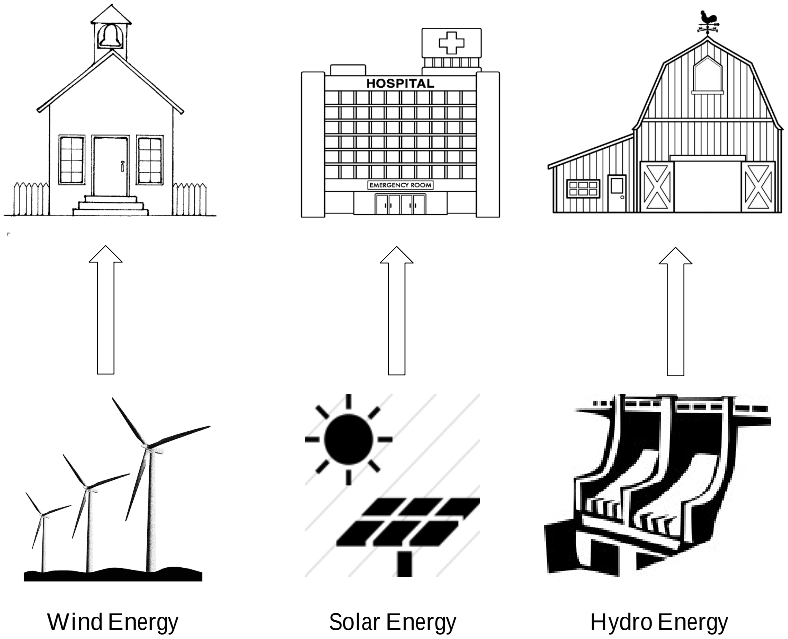


Figure 2.3: Distributed Generation.

This is an enormous upgrade over the RTUs used in the legacy grid, which are not synchronized (causing issues in discretized state estimation algorithms) and are sampled every 2 to 4 seconds [12]. Using the PMU data as observations can help state estimation algorithms provide faster and more accurate estimates. In the case of autonomous control, these devices will help the smart grid *self-heal* in case of an emergency situation, thus making the system more reliable. Additionally, smart metering promotes two-way communication between the consumer and the utility provider, allowing users to see electricity consumption and rates in real time. A user may elect to allow utilities to have control access to smart appliances in their home in order to consume power when the rates are low. This allows the functional smart grid to be more efficient, as well as providing an increased amount of market rate awareness to the user.

2.4 Microgrids

Microgrids are extremely important to the smart grid because they reinforce two major goals of the smart grid concept: that they are distributed and decentralized in nature. Microgrids are distributed because they use distributed generators to aggregate renewable energy sources in order to provide services for local loads. In particular, these microgrids break the centralized scheme of legacy EPGs by providing power from geographically distributed locations. Microgrids are decentralized because they are capable of cutting off from the main grid and operating autonomously in the *islanded* mode at any point in time. This setup addresses many key concerns faced with the current (legacy) EPG. The more power generated locally by a collection of microgrids, the less power needs to be generated by fossil fuel based generators. Furthermore, if failure of internal components within the main grid take power away from critical loads such as hospitals and fire stations, microgrids can be used to deliver power in the islanded mode.

Fig. 2.4 shows a schematic of a simple microgrid network, which shows two DGs, one load, and two transmission lines.

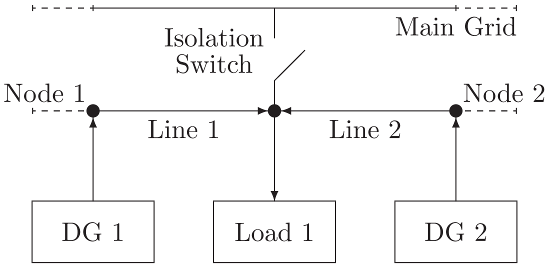


Figure 2.4: An Example of a Simple Microgrid.

The DGs are responsible for generating power through renewable energy sources known as micro-sources. These micro-sources can be: photovoltaic panels, fuel cells, wind turbines, and hydro turbines among others. The generated power then travels over the transmission lines and is consumed by the load. Each microgrid has an isolation switch, which facilitates its connection to the main grid. With the isolation switch turned on, the microgrid is in grid-connected mode and it injects or absorbs power to and from the main grid. When the switch is off, the microgrid enters the islanded mode, where it is cut off from the main grid and services only its local power demand [13].

Distributed generation technology has matured finally to an extent where microgrids are functioning independently in locations all over the world [14]. Furthermore, the energy sources used for the DGs are low cost, low voltage, high reliability, and most importantly, have minimum impact on the environment [15]. The majority of micro-sources produce direct current (DC) and are converted to alternating current (AC) by a voltage source inverter. Microgrids powered by DGs that are using voltage source inverters are classified as inverter based microgrids. *Specifically, this thesis will explore state estimation for* *inverter based microgrids operating in the islanded mode.*

The main features of smart grids, and by extension, microgrids, are denoted by the seven principles of smart grid philosophy [16]. These principles include: self-healing, active participation by consumers, protection against physical and cyber attack, power quality, adapting all generation and storage options, enabling new products, services and markets, and performance optimization. In order to achieve these targets, dynamic, efficient, and accurate decision making is required, especially in the case of autonomous control decisions. As such, the role of state estimation in aiding the efficiency and reliability of microgrids cannot be understated. Without accurate and timely knowledge of the state of the power network, decisions involving the operation, control, and efficiency of the microgrid will be negatively affected.

Chapter 3 - Non-linear State Estimation

This chapter introduces the topic of state estimation. In Section 3.1, a brief background review is presented, which covers the definitions and vocabulary used in state estimation. Sections 3.2 and 3.3 introduce the linear Kalman filter and non-linear particle filter, respectively, while Sections 3.4 and 3.5 provide a literature survey of state estimation methods applied to electric power grids. The SE techniques in this chapter are all assumed to be centralized in nature. In this infrastructure, all recorded measurements are processed at a single location solely responsible for evaluating the state estimate.

3.1 State Estimation

The *state* of a dynamical system is represented by a set of variables, which models and describes the entire system as it evolves over time. These state variables are used to predict the system's current and future behaviour. Examples of state variables in practical systems include: the value of voltage and current in electrical circuits, and position, velocity, and acceleration in motion based mechanical systems. All state variables are collected together into a vector referred to as the state vector. If the value of the state vector is known, the overall state of the system can be known. Eq. (3.1) shows the equation for a general discrete-time state vector

$$\boldsymbol{X}(k) = \{x_1(k), x_2(k), x_3(k)...x_{n_x}(k)\}, \tag{3.1}$$

where n_x is the number of state variables and index k represents the discretized time.

The evolution of state variables with time is modeled by a set of state equations, which are used to compute the respective values of the variables. These state equations are typically difference equations for discrete-time systems and differential equations for continuous-time systems. Since the thesis deals with non-linear systems, the state equations used in this thesis are discrete, non-linear, ordinary differential equations (ODEs). The system model for a particular system can be derived from collecting the state equations together as a typical system of equations. Therefore, simulating the behaviour of a system can be accomplished by implementing the system model, which is achieved by solving the system of equations at discrete points of time. Knowing that a dynamical system's state evolution over time/iteration k depends on the state of the previous iteration $k-1$, the state model can be denoted as

$$\boldsymbol{X}(k) = \boldsymbol{F}(\boldsymbol{X}(k-1), \boldsymbol{\xi}(k)), \tag{3.2}$$

where vector function $\boldsymbol{F}$ represents the collection of state equations for the state variables and $\boldsymbol{\xi}$ represents uncertainty or randomness in the system model. Parameter $\boldsymbol{\xi}$ is referred to as *process* noise and is often modeled with a known probability distribution.

As a system evolves over time, it may become important to obtain accurate estimates of its states. Often, these states are not directly observable. However, there exist observations or measurements that are mathematically related to the state. These observations are often noisy and corrupt. The observations can be collected in a vector as follows

$$\mathbf{z}(k) = \boldsymbol{G}(\boldsymbol{X}(k), \boldsymbol{\zeta}(k)), \tag{3.3}$$

where vector function $\boldsymbol{G}$ represents the collection of measurement equations and $\boldsymbol{\zeta}$ represents observation uncertainty (observation noise). This vector is referred to as the observation model.

The challenge of state estimation is to provide accurate state estimates by *inferring* the state from noisy measurements. Specifically, the goal is to build a posterior probability density function (pdf) of the state based on incoming measurements and other information including: the theoretical state model, an estimate of the initial state of the system, and the probability distribution of the uncertainties in both the state and observation models. This is referred to as the Bayesian approach to dynamic state estimation [17]. The joint pdf of the state that is conditional on both observations and the initial state can be denoted as

$$P(\boldsymbol{X}(k) \mid \boldsymbol{z}(1:k), \boldsymbol{X}(0)), \tag{3.4}$$

where $\boldsymbol{z}(1:k)$ represents all observations from iteration index 1 to k and $\boldsymbol{X}(0)$ represents the initial value of the state vector at $k = 0$. Effectively, this computation provides the pdf of the state based or conditioned on all previously available observations up to iteration k and the initial state of the system. In this thesis, the initial state will be assumed to be known and will subsequently be omitted from the notation.

The computation to estimate state $\boldsymbol{X}(k)$ is achieved in two steps using the principles of Bayesian statistics and recursion. In the prediction step, a rough estimate of the state vector is computed using all previously available observations up to $k-1$. This is also known as a *prior* estimate of the state, and is given by

$$P(\boldsymbol{X}(k)|\, \boldsymbol{z}(1:k{-}1)) = \int P(\boldsymbol{X}(k)|\, \boldsymbol{X}(k{-}1))P(\boldsymbol{X}(k{-}1)|\, \boldsymbol{z}(1:k-1))d\boldsymbol{X}(k{-}1), \tag{3.5}$$

where $P(\boldsymbol{X}(k) \mid \boldsymbol{X}(k{-}1))$ is the state transition model given in Eq. (3.2) and $P(\boldsymbol{X}(k{-}1) \mid \boldsymbol{z}(1:k-1))$ is the recursively calculated filtering distribution of the estimate in the previous iteration.

In the update step, the estimate made in the prediction step is updated with the new measurement at iteration k to obtain the *posterior* pdf of the state

$$P(\boldsymbol{X}(k) \mid \boldsymbol{z}(1:k)) = \frac{P(\boldsymbol{z}(k) \mid \boldsymbol{X}(k))P(\boldsymbol{X}(k) \mid \boldsymbol{z}(1:k-1))}{P(\boldsymbol{z}(k) \mid \boldsymbol{z}(1:k-1))}, \tag{3.6}$$

where $P(\boldsymbol{z}(k) \mid \boldsymbol{X}(k))$ is obtained from the observation model of Eq. (3.3) and $P(\boldsymbol{X}(k) \mid \boldsymbol{z}(1:k-1))$ is the prior state estimate given in Eq. (3.5). The denominator can be further simplified as follows

$$P(\boldsymbol{z}(k) \mid \boldsymbol{z}(1:k-1) = \int \boldsymbol{P}(\boldsymbol{z}(k \mid \boldsymbol{X}(k))\boldsymbol{P}(\boldsymbol{X}(k) \mid \boldsymbol{z}(1:k-1)). \tag{3.7}$$

All pdf's on the right hand side of the equations are now known. Thus, the state estimate can be found from the posterior pdf, which is obtained after recursively implementing these equations with respect to time index k. Sections 3.2 and 3.3 implement the prediction and update steps for state estimation in systems that are both linear and non-linear.

3.2 Linear Estimation Algorithms - Kalman Filter

The Kalman filter originates from filters that follow the aforementioned Bayesian approach and is the optimal state estimation choice for state models that are linear in nature [2]. It contains a set of mathematical equations, which recursively provide estimates for the state of a system in the past, present, and future (forecasting). The equations for the Kalman Filter are shown below [18].

For a linear state model, the state equation, Eq. (3.2), and observation equation, Eq. (3.3), are given by

State Model:

$$\boldsymbol{X}(k) = \boldsymbol{F}(k)\boldsymbol{X}(k-1) + \boldsymbol{\xi}(k), \tag{3.8}$$

Observation Model:

$$\boldsymbol{z}(k) = \boldsymbol{G}(k)\boldsymbol{X}(k) + \boldsymbol{\zeta}(k), \tag{3.9}$$

where $\boldsymbol{F}(k)$ is the state matrix and $\boldsymbol{G}(k)$ is the measurement matrix. The process noise $\boldsymbol{\xi}(k)$ is assumed Gaussian with zero mean and covariance matrix $\boldsymbol{Q}(k)$. Likewise, the observation noise is Gaussian with zero mean and covariance matrix $\boldsymbol{R}(k)$.

Prediction step:

Prior State Estimate:

$$\boldsymbol{X}(k \mid k-1) = \boldsymbol{F}(k)\boldsymbol{X}(k-1 \mid k-1) \tag{3.10}$$

Prior Covariance Prediction:

$$\boldsymbol{P}(k \mid k-1) = \boldsymbol{F}(k)\boldsymbol{P}(k-1 \mid k-1)[\boldsymbol{F}(k)]^T + \boldsymbol{Q}(k) \tag{3.11}$$

Measurement Prediction Covariance:

$$\boldsymbol{S}(k \mid k-1) = \boldsymbol{G}(k)^T\boldsymbol{P}(k \mid k-1)\boldsymbol{G}(k) + \boldsymbol{R}(k) \tag{3.12}$$

Kalman Gain:

$$\boldsymbol{K}(k) = \boldsymbol{P}(k \mid k-1)\boldsymbol{G}(k)\boldsymbol{S}(k \mid k-1)^{-1} \tag{3.13}$$

Measurement update:

Updated State Estimate:

$$\boldsymbol{X}(k \mid k) = \boldsymbol{X}(k \mid k-1) + \boldsymbol{K}(k)(\boldsymbol{z}(k) - [\boldsymbol{G}(k)]^T\boldsymbol{X}(k \mid k-1)) \tag{3.14}$$

Updated Error Covariance:

$$\boldsymbol{P}(k \mid k) = [I - \boldsymbol{K}(k)[\boldsymbol{G}(k)]^T]\boldsymbol{P}(k \mid k-1). \tag{3.15}$$

In the prediction step, the objective is to compute a crude estimate for the state and error covariance matrix (also known as the prior estimate). The prior state estimate is computed using the mathematical state model using the estimate given at the end of the previous iteration, while the prior error covariance is computed similarly using the previous estimate of the covariance. Both these estimates are used in the next step (measurement update). First, a constant known as the Kalman gain is calculated and is subsequently used to arrive at an updated estimate for the state based on the difference between the incoming measurement and prior estimate. The error covariance is also updated, and these posterior estimates are used recursively in the prediction step for the next iteration.

3.3 Non-linear Estimation Algorithm - Particle Filter

Two significant assumptions limit the Kalman filter. The dynamical system must be linear and the pdf of the state must remain Gaussian at every iteration. Practical, real-world systems are most often non-linear, however, and applying the Kalman filter to these systems leads to linearization error [19]. A general closed form solution does not exist for expressing the pdf for non-linear, non-Gaussian systems [20].

The particle filter uses Monte Carlo methods in order to approximate the posterior pdf by constructing it using random samples known as particles. In the very first iteration of the estimation process, each particle is created by generating N_s random state vectors based on the initial state value. These particles are then propagated through time using the state model and a measurement update step is utilized to process incoming measurements. The relative likelihood of the particle is then calculated and is given by a numerical weight. At the end of the iteration, new particles are generated based on the relative likelihood of all the particles. It must be noted that the particle filter does not require the state or observation models to be linear, nor does it require that the process or measurement noise be Gaussian. The general algorithm for the particle filter is presented below [21].

Consider a system with a state vector comprised of n_x states. For the state variable $x(k)$, the total number of particles are denoted by N_s, and the complete set of particles and its respective weights are denoted by the terms $\{\mathbb{X}_i(k)\}_{i=1}^{N_s}$ and $\{W_i(k)\}_{i=1}^{N_s}$ respectively, where i represents the index of the particle. A sample particle is therefore a vector and is represented as

$$\mathbb{X}_i(k) = \{x_{1,i}(k), x_{2,i}(k), x_{3,i}(k) ... x_{n_x,i}(k)\}. \tag{3.16}$$

The particles can be propagated through the system model at every iteration k as

$$\mathbb{X}_i(k) = \boldsymbol{F}(\mathbb{X}_i(k-1), \boldsymbol{\xi}_i(k-1)) \; ... \; (i = 1 : N_s). \tag{3.17}$$

The state values can then be used to update the observation model for each particle using $\boldsymbol{G}(\mathbb{X}_i(k), \zeta(k))$. Once the actual measurement $\boldsymbol{z}^*$ at k is available, the weight for each particle can be calculated by comparing these two values. The relative likelihood W_i is computed by comparing the value obtained from the observation model with the actual measurement $\boldsymbol{z}^*$

$$\begin{aligned} W_i(k) &= \boldsymbol{P}[(\boldsymbol{z}(k) = \boldsymbol{z}^*) \mid (\boldsymbol{X}(k) = \mathbb{X}_i(k))], \\ &= \boldsymbol{P}[\boldsymbol{\xi}(k) = \boldsymbol{z}^* - \boldsymbol{G}(\mathbb{X}_i(k))], \\ &\sim \frac{1}{2\pi^{m/2}|R|^{1/2}} \exp\left\{ \frac{-[\boldsymbol{z}^* - \boldsymbol{G}(\mathbb{X}_i(k))]^T \times R^{-1}[\boldsymbol{z}^* - \boldsymbol{G}(\mathbb{X}_i(k))]}{2} \right\}, \end{aligned} \tag{3.18}$$

where R is the error covariance of the observation noise. For simplicity, Eq. (3.18) is written for the case when the observation noise is Gaussian but is generalizable for other pdfs.

These weights are then normalized to the unit value of 1 as follows

$$W_i(k) = \frac{W_i(k)}{\sum_{j=1}^{N_s} W_j(k)}. \tag{3.19}$$

A well known issue with the particle filter is the degeneracy of the particles, where a small number of particles become dominant with time and have relatively higher weights than the rest. To avoid such a situation, a procedure known as resampling is introduced, in which new particles are randomly generated from the distribution of the weights of the particles. Some often used techniques are: residual sampling, systematic resampling, and stratified sampling. As the number of particles used approaches a high value, the pdf of the resampled particles approaches the pdf $P(\boldsymbol{X}(k)|\boldsymbol{z}(k))$ [17]. The particle filter works well for non-linear systems and is accurate when the number of particles are high, but the accuracy comes at the cost of higher computational complexity at $\mathcal{O}(n_x^2 N_s)$ floating point operations. The choice of the number of particles is often a tradeoff between speed and accuracy.

3.4 Literature Survey of State Estimation in Conventional Power Grids

The primary purpose of an EPG is to generate and distribute electrical energy to a variety of end users in a reliable and efficient manner. The EPG uses state estimation (SE) to monitor the electric grid in order to see if it is in a stable condition. SE is also used to optimize power flows, detect line faults, and provide forecasting for power failures. In general, SE attempts to produce an estimate of the *state* of the EPG by using measurements of electrical quantities within the grid. The state variables that are estimated in EPGs typically include the complex voltage and phase angles at every bus (a connection point on the grid). These variables are traditionally difficult to measure directly, but given a set of measurements which are electrically related, the SE algorithm can *infer* the state from the measurements. Therefore, given a set of measurements/observations (power injections, active/reactive power flows between buses) and knowing the overall network topology, an estimate can be made of the state [8].

Performing state estimation for power transmission networks is an active research topic since the 1970's and has its roots in *static* state estimation [22]. The main impact of this research is the fact that it differentiates state estimation from conventional load flow calculations, which were previously used for management purposes such as power flow optimization. The static state estimator is designed to handle some degree of uncertainty in the measured observations due to calibration and communication noise among other factors. However, this estimation technique assumes that the network is in a quasi-static state. Due to the ever changing nature of load and generation patterns, modeling the dynamics of the network as quasi-static is not an accurate assumption. In particular, due to sudden load/generation changes due to component failures (generator and line faults), the static nature of the estimator does not capture the true system dynamics.

Furthermore, the static nature of the estimator means that it needs to be run in its entirety for the next time instant and that it discards any previous estimate made, relying entirely on a fresh set of observations that may be severely corrupted or missing entirely. Such an approach is suboptimal compared to an estimator that incorporates both past and present information systematically. The first dynamic state estimator, proposed in 1970, improves on these two facts by using its previous estimate recursively in its next estimate [23]. In case a measurement is not available, the predicted estimate is substituted for the measurement. Given the continuous evolution of the states with time in EPGs, tracking the states dynamically improves the accuracy of the estimate. In [23], the Weighted Least Squares (WLS) method is used as the state estimation technique. The WLS method minimizes the weighted sum of the square of the error between the measurement and the estimate. Although the WLS algorithm is easy to implement, the solution is quite sensitive to noise in measurements [24].

Although the estimator in [23] is dynamic, test simulations are still performed in [23] with a quasi-static power system model. Physical modeling of the generators and a more complete model of the network is developed in [25]. This paper proposes the Kalman filter as the state estimation algorithm of choice. For linear models, the Kalman filter has been the optimal algorithm used to predict states from noisy measurements. However, the EPG is a highly non-linear system, and as such the Kalman filter is not an accurate solution [26]. When applied to non-linear systems, the Kalman filter incurs significant linearization error and converges slowly.

In [27], a non-linear state estimation approach is proposed by introducing a modified Extended Kalman Filter (EKF) technique. The classical EKF linearizes the state model about an operating point using the Taylor Series expansion and then uses the original Kalman filter to predict the new estimate. While an improvement over the Kalman filter, the EKF faces stability and convergence issues when linearizing highly non-linear system models. In addition, very sudden load/generation changes cause significant nonlinearities in the observation model. As such, the observation model is kept non-linear and the performance of the filter is improved over the Kalman Filter.

The advent of Phasor Measurement Units (PMUs) provides researchers with a new hope for accurately monitoring the EPG. PMUs are very fast monitoring devices that capture voltage and current waveforms at points within the EPG. State estimation techniques, in particular, could use the fast transmission speed of PMU data to provide better estimates. As such, [28] uses the EKF on a more dynamic model of a synchronous generator by including the rotor angle ω as part of the state vector. The state estimator uses the PMU data to predict the state as well as proposing a novel version of the EKF, which predicts the state estimate with unknown input measurements that are usually considered constant.

Building on this work, the Unscented Kalman Filter (UKF) is applied to the above system model [29]. The UKF does not linearize the state and observation models and instead creates sigma points around the mean of the state from which deterministic sampling is used to arrive at the best estimate [19]. As such, the UKF fares better than the EKF when it comes to highly non-linear systems. In [30], the UKF is tested on multiple bus systems such as IEEE-14, IEEE-30, and IEEE-57 test bus systems, while in [31], it is tested on the WSCC 9 bus system. In [32], the Ensemble Kalman Filter (EnKF) is used as the state estimation technique. The EnKF also does not linearize the state and observation models, but uses Monte Carlo methods to propagate the distribution of the states through points called ensembles. This technique is closely related to the Particle Filter (PF), however, both UKF and EnKF assume the additive noise to be Gaussian. The particle filter has no such constraints and the noise can be non-Gaussian and colored.

All of the aforementioned approaches are tested on networks with a centralized architecture. In other words, all observations are collected at a central location called a fusion center, which then uses the state estimation algorithm of choice to provide an estimate of state for every bus. This is the most accurate solution; however, for large systems this is not feasible for several reasons. First, the task is computationally expensive. Second, busses located far away from the fusion center take more time to transmit their observation which introduces latency into the system. Third, the system is extremely vulnerable to failure since it has a single point of failure at the fusion center.

The aforementioned reasons have sparked an interest in a more distributed and decentralized approach to state estimation. The fusion center is disbanded, and instead, the network is partitioned into subsystems. Each subsystem is given some processing resources and is responsible only for computing its local estimates, thus reducing computational complexity. Information sharing can be done between neighbours, thus eliminating the need for a fusion center. With power deregulation occurring in North America and operators having to coordinate and oversee power transactions over long distances, a distributed architecture is a far more attractive option than the centralized approach. Work on linear distributed approaches is explored in [33]-[36], while work on non-linear distributed approaches is pursued in [37]-[39].

In [37], a decentralized kalman filter (DKF) is used to estimate the state of a power network in which only the load model is kept non-linear. The dynamics of the electrical power converter are neglected and a linearized state model is used in its place. As discussed previously, a linearized model is not an accurate assumption for a model which is highly dynamic. In this case of distributed non-linear networks, the DKF will incur significant errors and converge slowly when it assimilates observations from different subsystems. An even simpler power distribution network is explored in [38] whereby an extended information filter (EIF) and unscented information filter (UIF) are proposed. The EIF assigns an EKF at each subsystem to perform local filtering and uses a fusion rule to achieve consensus within the network. This state model comprises of 5 busses and 3 observation nodes. This model is non-linear, however, all five state equations are identical with the only variation among them being the Gaussian noise added to each state. As in [37], the proposed filter must be expanded to a more realistic EPG for verification.

In [39], semidefinite programming is used to estimate the state of the IEEE-30 and IEEE-118 bus systems. This approach fuses traditional non-linear telemeter observations with linear PMU measurements in the overall observation model to achieve *hybrid* state estimation. However, this approach has the worst case computational complexity of $O(N^{4.5} \log 1/\epsilon)$, where ϵ is the solution accuracy greater than 0. In comparison, the worst case complexity of the reduced-order, distributed particle filter is lower at $n_{\text{sub}} \times \text{O}((n_x/n_{\text{sub}})^2 N_s) \approx \text{O}(n_x^2 N_s/n_{\text{sub}})$, where n_x is the number of state variables, N_s is the number of particles, and n_{sub} is the number of subsystems. The reduced-order, distributed particle filter is therefore a more computationally feasible approach for extremely large systems such as EPGs. For the purposes of comparing the merits of the particle filter to other well-known estimation approaches (EKF, UKF, EnKF), a comparative study is done in [40]. Furthermore, comparisons of the distributed particle filter with the distributed Extended Kalman Filter and other distributed estimation techniques are covered in [41]-[43].

3.5 Literature Survey of State Estimation in Islanded Microgrids

Although research has been done in distribution networks and microgrid state estimation in general, state estimation on islanded microgrids is a very fresh topic of research in which not much work has yet been done. Two significant papers exist in this area of research. In [44], state estimation is performed on microgrids operating in both islanded and grid-connected mode. However, a linear state model is used and a Weighted Least Squares estimation technique is employed which can be sensitive to outliers. In [45], the optimal Kalman filter is used; however the model is again linearized about an operating point. Non-linear modeling in microgrids is essential in order to capture its very quick system dynamics. Furthermore, with the DGs interfaced to the network through power electronic converters and lacking physical inertia, the overall system is susceptible to oscillations from network disturbances [13]. Non-linear state estimation in microgrids is required to provide reliable, quick, and accurate state estimates in order to anticipate these disturbances.

This thesis explores the implementation of the reduced-order, distributed particle filter in both modern transmission networks and microgrids. The particle filter is well equipped to handle the non-linearities of the system model and does not assume the process and measurement noise vectors to be Gaussian. Up to the author's knowledge, this research is the first of its kind in the field of microgrids operating in the islanded mode.

Chapter 4 - Distributed Non-linear State Estimation

This chapter introduces the concept of distributed state estimation. In Section 4.1, the practical infeasibility of centralized state estimation techniques for large scale and/or geographically distributed systems is discussed to motivate the introduction of a distributed framework. Section 4.2 illustrates the distributed framework with a sample example, while Section 4.3 outlines the algorithm for the distributed, reduced-order particle filter

4.1 Motivation

The linear Kalman Filter and non-linear particle filter discussed in Chapter 3 were assumed to be implemented within a centralized infrastructure. That is, all sensor nodes transmit their observations to a single, centrally located processing unit, which then performs the state estimation task (Fig. 4.1). This processing resource is known as the fusion center. Theoretically, aggregating all observations at the fusion center provides the most accurate result since the fusion center has access to the complete set of measurements

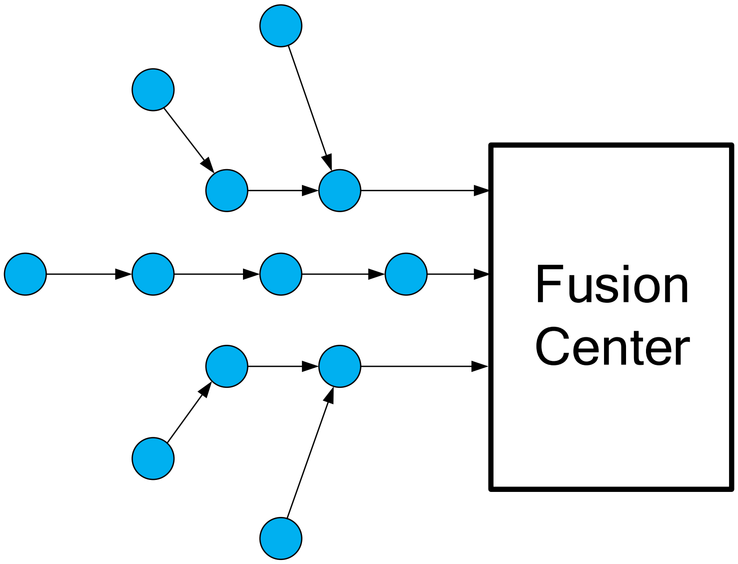


Figure 4.1: Centralized State Estimation Architecture.

and, as such, can produce a better estimate for the state. Practically, this approach is not preferred for a variety of reasons.

For systems that are large scale and have sensor nodes distributed over very large geographical distances, a centralized architecture is problematic for a number of reasons. Nodes which are far away from the fusion center cannot transmit their observations to the fusion center in a timely manner. In such cases, these nodes transmit their observations to other nodes in their neighbourhoods till the information is propagated to the fusion center. While adding significant overhead and latency into the system, this disrupts the overall energy balance within the SE communication system. Since the nodes that are closer to the fusion center transmit additional information, these nodes consume more energy than the more distant nodes causing an imbalance in the energy consumption of the nodes.

This singularity of the system also brings with it a multitude of issues. As seen in the previous chapter, state estimation algorithms can be intensive and require a large amount of computational resources, especially for systems with large dimensional state vectors. With just one central resource processing all the measurements and performing state estimation under hard real time constraints, the task becomes very computationally complex and requires a large number of computational resources. Furthermore, the system has a single point of failure. This vulnerability is significant, especially in the case of systems such as electrical power grids. A system failure of any magnitude within the EPG has the potential to lead to catastrophic failure of the magnitude of the blackout faced in 2003.

Specifically with regards to the electrical industry, the focus has shifted from a hierarchical, centralized infrastructure to a more modular and distributed approach. The traditional design of EPGs involved a monopolistic approach with massive power plants serving a large number of customers within the generation area. With deregulated markets breaking this monopoly, smaller vendors with distributed generation plans have entered the market. As such, transmission system operators regularly oversee power transactions over very large geographical distances. A centralized state estimation framework for EPGs is unscalable and does not fit with the ongoing paradigm shift in electric power distribution and decentralization.

Distributed state estimation seeks to eliminate the fusion center from the framework. Instead, each sensor node is given some computing resources to perform state estimation. Each sensor node communicates only with its neighbouring nodes to exchange information, observations, and estimates. Neighbouring nodes that share observations and have common states with other nodes implement a data fusion rule in order to achieve a consistent estimate. This rule helps achieve consensus for state estimates throughout the network. Fig. 4.2 shows a distributed state estimation architecture, where the fusion center has been disbanded and instead nodes exchange two-way information with their neighbours.

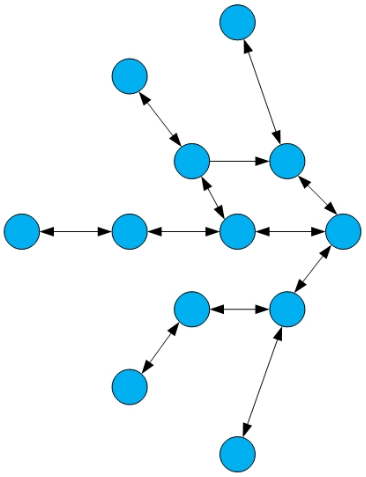


Figure 4.2: An Example Distributed State Estimation Architecture.

4.2 Distributed, Reduced-order Estimation Configuration

Distributed state estimation can be divided into two categories: full-order and reduced-order implementations. In the full-order implementation, each node maintains an estimate of all the state variables in the network. A consensus step is used to combine the local estimates of all state variables derived at the constituent nodes to achieve a degree of consistency for the overall state estimate [46]-[47]. This approach does not work well with large scale systems with a high order of state variables since the computational complexity of maintaining the estimates of all states at each node is extremely high. Secondly, the amount of information exchanges with neighbouring nodes also increases communication overhead in the full-order implementation.

In the reduced-order approach, the overall network is spatially partitioned into subsystems which estimate only a subset of the overall state vector based on the observations made within the subsystems.. As such, each subsystem maintains a reduced-order state model of its constituent nodes. It may be the case that for implementing the reduced-order state model for a particular subsystem, the subsystem may need state variables which it does not observe directly. These state variables are collected in a forcing vector $\boldsymbol{d}$, whose value, if needed, is borrowed from the neighboring nodes where the forcing states comprising $\boldsymbol{d}$ are being estimated. Consensus and data fusion steps are only required for those nodes which share some states. The reduced-order approach, therefore, cuts down on both the computational complexity as well as the number of information exchanges within the network. This thesis focuses on the distributed, reduced-order state estimation implementation.

Formally, the network is spatially decomposed or partitioned into n_{sub} subsystems, where each subsystem is denoted by $S^{(l)}$ and l is the index of the subsystem. As such, the reduced-order state model at a particular subsystem is thus obtained from the overall state model as follows

$$S^{(l)} : \boldsymbol{X}^{(l)}(k) = \boldsymbol{F}^{(l)}(\boldsymbol{X}^{(l)}(k-1), \boldsymbol{d}^{(l)}(k-1)) + \boldsymbol{\xi}^{(l)}(k). \tag{4.1}$$

Similarly, the local observation vector can also be obtained as follows

$$S^{(l)} : \boldsymbol{z}^{(l)}(k) = \boldsymbol{G}^{(l)}(\boldsymbol{X}^{(l)}(k)) + \boldsymbol{\zeta}^{(l)}(k). \tag{4.2}$$

Fig. 4.3 shows a test power distribution network containing 5 busses that have been partitioned into 3 subsystems [53]. In this network, the busses are represented by bold bars, while their connecting paths are denoted by thin lines. The state model for the overall system is given below.

$$\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \end{bmatrix} = \begin{bmatrix} 1+\sin(.04*\pi x_1(k-1))+x_2(k-1)) \\ 1+\sin(.04*\pi x_2(k-1))+x_4(k-1)) \\ 1+\sin(.04*\pi x_1(k-1))+x_3(k-1)) \\ 1+\sin(.04*\pi x_4(k-1))+x_5(k-1)) \\ 1+\sin(.04*\pi x_5(k-1))+x_4(k-1)) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\xi}_1(k-1) \\ \boldsymbol{\xi}_2(k-1) \\ \boldsymbol{\xi}_3(k-1) \\ \boldsymbol{\xi}_4(k-1) \\ \boldsymbol{\xi}_5(k-1) \end{bmatrix} \qquad (4.3)$$

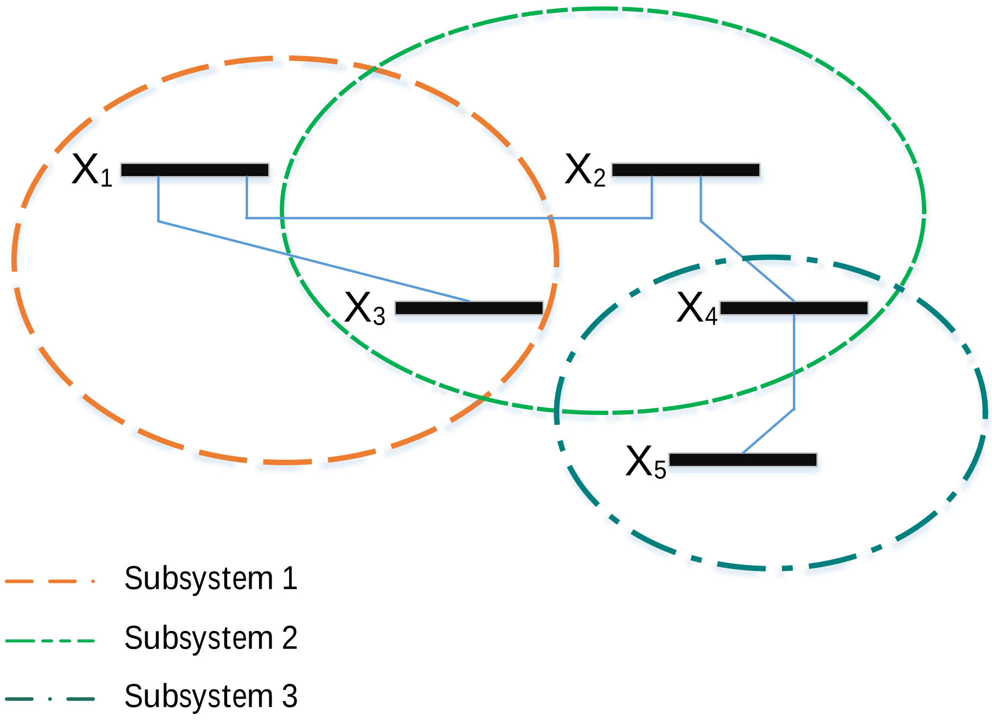


Figure 4.3: Network Partitioned Into 3 Subsystems.

The state model of every subsystem, its forcing terms, as well as the shared terms between the subsystems are provided below. The forcing terms for each subsystem have been underlined for emphasis.

Subsystem 1 - $S^{(1)}$

$x_1(k) = 1 + \sin(.04 * \pi x_1(k-1)) + \underbrace{x_2(k-1))}$

$x_3(k) = 1 + \sin(.04 * \pi x_1(k-1)) + x_3(k-1))$

$\boldsymbol{d}^{(1)}(k) = \{x_2(k)\}$

Subsystem 2 - $S^{(2)}$

$x_2(k) = 1 + \sin(.04 * \pi x_2(k-1)) + x_4(k-1))$

$x_3(k) = 1 + \sin(.04 * \pi \underbrace{x_1(k-1))} + x_3(k-1))$

$x_4(k) = 1 + \sin(.04 * \pi x_4(k-1)) + \underbrace{x_5(k-1))}$

$\boldsymbol{d}^{(2)}(k) = \{x_1(k), x_5(k)\}$

Subsystem 3 - $S^{(3)}$

$x_4(k) = 1 + \sin(.04 * \pi x_4(k-1)) + \underbrace{x_5(k-1))}$

$x_5(k) = 1 + \sin(.04 * \pi x_5(k-1)) + x_4(k-1))$

No forcing term for Subsystem 3

Note x_3 is shared between $S^{(1)}$ and $S^{(2)}$ and x_4 is shared between $S^{(2)}$ and $S^{(3)}$.

With these equations clearly defined, the state estimation algorithms covered previously in Chapter 3 can be suitably modified to incorporate the reduced-order implementation approach. The reduced-order state model is computed based on the estimate made by the subsystem of the previous iteration as well as any borrowed forcing terms from any neighbouring subsystem using Eq. (4.1). Observations for the subsystems are also local and based on Eq. (4.2). A mathematical data fusion rule is typically implemented in order to obtain the best estimate for those states that are shared between subsystems. The estimates are then shared across those subsystems to achieve consensus in the estimates for the shared states.

4.3 Distributed, Reduced-order Particle Filter

In this section, the algorithm for the distributed, reduced-order particle filter is presented. This knowledge builds on the centralized particle filter which was derived in Section 3.3. As discussed previously, the particle filter approximates the posterior state estimate using particles drawn by random sampling from a weighted distribution and denoted by $\{\{\mathbb{X}_i(k)\}_{i=1}^{N_s}, W_i(k)\}_{i=1}^{N_s}\}$. In the reduced-order implementation, a local particle filter is assigned to each subsystem which evaluates the marginalized filtering distribution for its local state vector. Each subsystem, therefore, has its own particles and local weights denoted by $\{\mathbb{X}_i^{(l)}(k), W_i^{(l)}(k)\}$. The following algorithm describes the reduced-order, distributed particle filter at iteration k in terms of the state and observation models, Eqs. (4.1) and (4.2), for the system partitioned in subsystems $S^{(l)}$. The initial state of the system is assumed to be known.

1. **Compute Forcing Terms:** At the beginning of a new iteration, the forcing terms required for the local state vector are computed from the previous iteration. This is

done by taking the weighted mean of the particles for the particular state variable as follows

$$\boldsymbol{d}^{(l)}(k) = \frac{\sum_{i=1}^{N_s} W_i^{(l)}(k-1)\mathbb{X}_i^{(l)}(k-1)}{\sum_{i=1}^{N_s} W_i^{(l)}(k-1)}. \tag{4.4}$$

2. **Run Local Particle Filter:** The particles are propagated through the reduced-order state model and are used to update the observation model, with the final step to calculate the weights of the local particles.

$$S^{(l)} : \mathbb{X}^{(l)}(k) = \boldsymbol{F}^{(l)}(\mathbb{X}^{(l)}(k-1), \boldsymbol{d}^{(l)}(k-1)) + \boldsymbol{\xi}^{(l)}(k) \tag{4.5}$$

$$S^{(l)} : \boldsymbol{z}^{(l)}(k) = \boldsymbol{G}^{(l)}(\mathbb{X}^{(l)}(k)) + \boldsymbol{\zeta}^{(l)}(k) \tag{4.6}$$

$$S^{(l)} : W_i^{(l)}(k) \propto W_i^{(l)}(k-1)P(\boldsymbol{z}_i(k)^{(l)} \mid \mathbb{X}_i^{(l)}(k)) \times \frac{P(\mathbb{X}_i^{(l)}(k) \mid \mathbb{X}_i^{(l)}(k-1))}{q(\mathbb{X}_i^{(l)}(k) \mid \mathbb{X}_i^{(l)}(k-1))}, \tag{4.7}$$

where the symbol q represents the proposal distribution. This is obtained from the transition pdf, $p(\mathbb{X}_i^{(l)}(k) \mid \mathbb{X}_i^{(l)}(k-1))$, where the weights are pointwise evaluation of the likelihood function at the particle values [46].

3. **Implement Fusion Rule:** Each subsystem will have their own estimate for the shared states between them. In order to achieve a consensus of the estimate, the estimates of the shared states $X_n^{(\text{fuse})}$ are fused together using their mean ($\mu_n^{(l)}$) and covariance ($P_n^{(l)}$) as computed by the following data fusion rule [55]

$$X_n^{(\text{fuse})}(k). = \Big(\sum_{l \in \mathcal{G}_n} [P_n^{(l)}(k)]^{-1}\Big)^{-1} \Big(\sum_{l \in \mathcal{G}_n} [P_n^{(l)}(k)]^{-1} \mu_n^{(l)}(k)\Big), \tag{4.8}$$

where $\mathcal{G}_n$ represents the sensor nodes in the neighbourhood of system $S^{(l)}$.

4. **Resample:** Once the state vectors of each subsystem are updated with the fused value of the shared states, each local particle filter is resampled according to its local weights. This completes one full iteration of the distributed particle filter.

The pressing question when comparing centralized and distributed estimation architectures revolves around accuracy. Centralized SE has a number of practical issues, including high computational cost, increased latency, and a single point of failure, but provides the most accurate result. Can a reduced-order, distributed implementation reach the level of accuracy of a centralized scheme at a much lower computational cost and much higher efficiency? Chapters 5 and 6 attempt to answer this question by applying both centralized and reduced-order, distributed particle filters to highly non-linear EPGs and comparing their results.

Chapter 5 - State Estimation in Modern Power Grids

Chapters 2, 3, and 4 explained basic power systems, linear and non-linear state estimation techniques, as well as the centralized and distributed implementations of the particle filter. In this chapter, the distributed, reduced-order particle filter is applied to a modern, deregulated transmission power network. In Section 5.1, the system model for such networks is explored. Section 5.2 derives the system model for a sample test network (IEEE 5 bus) along with a reduced-order model of the system as an example. The centralized and reduced-order, distributed particle filters are then applied to an expanded test network (IEEE 14 bus) in Sections 5.3 and 5.4, respectively, followed by simulation results in Section 5.5. Section 5.6 is dedicated to discuss the situations in which the state estimation system can diverge and ultimately fail.

5.1 System Modeling

As discussed earlier, the electric power grid (EPG) can be thought of as a system which generates, transmits, and distributes electricity. Fig. 5.1 shows a schematic diagram of a generalized EPG, which comprises of generators, transmission lines, and loads.

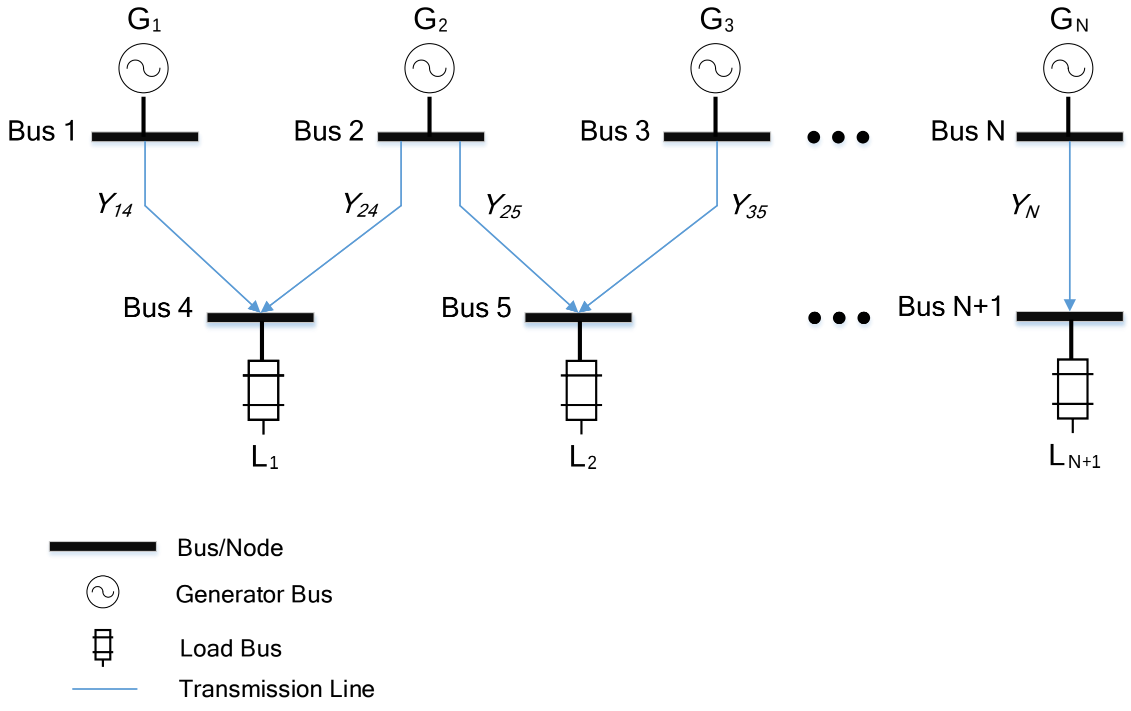


Figure 5.1: A Sample Power Network.

Each bus is a connection point on the grid and is either a generator bus or a load bus. Power is generated at the generator bus and flows through the transmission lines until it is consumed by the load bus. The entire system can be modeled by a set of ordinary, non-linear differential equations for the state variables representing the power grid. Typically, the state variables of interest for the generator busses are the voltage (V_{G_i}), phase angle (δ_{G_i}), and speed (ω_{G_i}) of the generator, while the state variables for the load busses are voltage V_{L_i} and phase angle δ_{L_i}.

5.1.1 State Model

The non-linear state equation for both types of busses are given below [58].

Generator Bus:

$$\dot{V_{G_i}}(t) = \frac{E_{f_i} - V_i}{T_{do_i}} + \frac{X_{d_i} - X^{'}_{d_i}}{T_{do_i}} \sum_{j \in \aleph_i} \Big\{ V_j \big(G_{ij} \sin(\theta_{ij}) - B_{ij} \cos(\theta_{ij}) \big) \Big\} \tag{5.1}$$

$$\dot{\theta_{G_i}}(t) = \omega_i(t) \tag{5.2}$$

$$\dot{\omega_{G_i}}(t) = \frac{-D_i \omega_i + P_{m_i}}{J_i} - \frac{1}{J_i} \sum_{j \in \aleph_i} \Big\{ V_j V_i \big(B_{ij} \sin(\theta_{ij}) - G_{ij} \cos(\theta_{ij}) \big) \Big\} \tag{5.3}$$

The notation θ_{ij} represents θ_i - θ_j, where the first subscript i represents the index of the *from* bus and the second subscript j represents the index of the *to* bus. The generator constants are as follows: T_{do} is the direct-axis transient time constant, X_d and $X_d^{'}$ are the direct and transient axis reactances, respectively, D is the damping factor and J is the rotor inertia. The control inputs are E_f and P_m which represent, respectively, the electromagnetic field used for excitation and mechanical input power. The symbols G_{ij} and B_{ij} represent the real and imaginary components of the nodal admittance matrix Y which will be discussed shortly. The equations for the load bus are presented below [59].

Load Bus:

$$\dot{V_{L_i}}(t) = -\sum_{j \in \aleph_i} \Big\{ V_i V_j \big((G_{ij} \sin(\theta_{ij}) - B_{ij} \cos(\theta_{ij}) \big) \Big\} \tag{5.4}$$

$$\dot{\theta_{L_i}}(t) = -\sum_{j \in \aleph_i} \Big\{ V_i V_j \big((G_{ij} \cos(\theta_{ij}) - B_{ij} \sin(\theta_{ij}) \big) \Big\} \tag{5.5}$$

The symbol $\aleph_i$ represents the *neighbourhood* of the bus at index i. The overall state vector can thus be defined as a collection of state variables based on the type of bus as follows

$$\dot{\boldsymbol{X}}(t) = \{V_{G_i}(t), \theta_{G_i}(t), \omega_{G_i}(t), V_{L_i}(t), \theta_{L_i}(t) ...\}, \tag{5.6}$$

where the complete state model in Eq. (5.6) can then be represented as a set of non-linear ordinary differential equations

$$\frac{d\boldsymbol{X}(t)}{dt} = \boldsymbol{F}(\boldsymbol{X}(t) + \boldsymbol{\xi}(t)). \tag{5.7}$$

For computational purposes, Eq. (5.7) is often discretized using a finite difference or finite element scheme as follows

$$\boldsymbol{X}(k+1) = \boldsymbol{X}(k) + \boldsymbol{F}(\boldsymbol{X}(k)) \times \Delta T, \tag{5.8}$$

where ΔT is a constant time-step.

It must be noted that the dynamics of the power flow between busses is more dependent on the *difference* of the phase angles as opposed to the phase angles themselves [12]. For this reason, one of the busses is nominated as the 'slack' or reference bus and its phase angle is set to zero. This is done so that the phase differences from this bus and other busses in the network can be found when performing load-flow analysis. By convention, the phase angle of the first bus is set to zero, and this practice will be followed in the thesis as well. Since the state variable corresponding to the reference phase angle is always zero, it will not be estimated.

5.1.2 Y-Bus

The connectivity of any power network can be described by constructing a Y-Bus Y, which is a symmetrical, two-dimensional matrix representing the bus to bus connectivity of the network. Each entry of the matrix (*i,j*) is the accumulative admittance of all power lines joining bus i to bus j, while each entry (*i,i*) represents the accumulative admittance y_{ij} of all power lines at bus i. If any entry in this matrix is zero, no transmission path exists for power to flow from bus i to j. The general form for a Y-Bus is presented in Eq. (5.9)

$$Y_{ij} = \begin{cases} y_{ii} + \sum_{k \neq i} y_{ij} & \text{if } i = j \quad (5.9a) \\ -y_{ij} & \text{if } i \neq j \quad (5.9b) \end{cases}$$

The entry Y_{ij} is complex valued and represented by

$$Y_{ij} = G_{ij} + jB_{ij} \tag{5.10}$$

where G is the conductance, B is the susceptance and $\hat{j} = \sqrt{-1}$ is the unit imaginary number. The conductance and susceptance of each power line is readily found in the supplementary data accompanying any IEEE test network and is common referred to as the line data.

5.1.3 Observation Model

Typical observations for an EPG are the real/reactive power flows and injections, branch current magnitudes, and bus voltage magnitudes. In this thesis, real power flows and real power injections are considered as the observations. However, the techniques presented for state estimation are general and can be applied with other kinds of observations as well. Since the estimation is being applied to a *modern* transmission network, these observations are assumed to come from PMUs. The equations used in the observation model are presented below [60].

Real Power Injection:

$$P_{ii}(t) = V_i \sum_{j \in \aleph_i} \Big\{ V_j \big(G_{ij} \cos(\theta_{ij}) + B_{ij} \sin(\theta_{ij}) \big) \Big\} \tag{5.11}$$

Real Power Flow:

$$P_{ij}(t) = V_i^2 (G_{ij}) - V_i V_j \big(G_{ij} \cos(\theta_{ij}) + B_{ij} \sin(\theta_{ij}) \big) \tag{5.12}$$

The overall observation vector can be defined as

$$\boldsymbol{z}(t) = \{P_{ii}(t), P_{ij}(t) ...\}, \tag{5.13}$$

where the overall observation model for observations $\boldsymbol{z}(t)$ can then be represented as

$$\boldsymbol{z}(t) = \boldsymbol{G}(\boldsymbol{X}(t)) + \boldsymbol{\zeta}, \tag{5.14}$$

where Eq. (5.14) can also be discretized using a finite difference or finite element scheme to

$$\boldsymbol{z}(k+1) = \boldsymbol{G}(\boldsymbol{x}(k+1)) + \boldsymbol{\zeta}(k+1). \tag{5.15}$$

5.2 Complete Modeling for the IEEE 5 Bus

Fig. 5.2 shows a simple test network based on the IEEE 5 Bus, which is often used by researchers to simulate the behaviour of a typical power system.

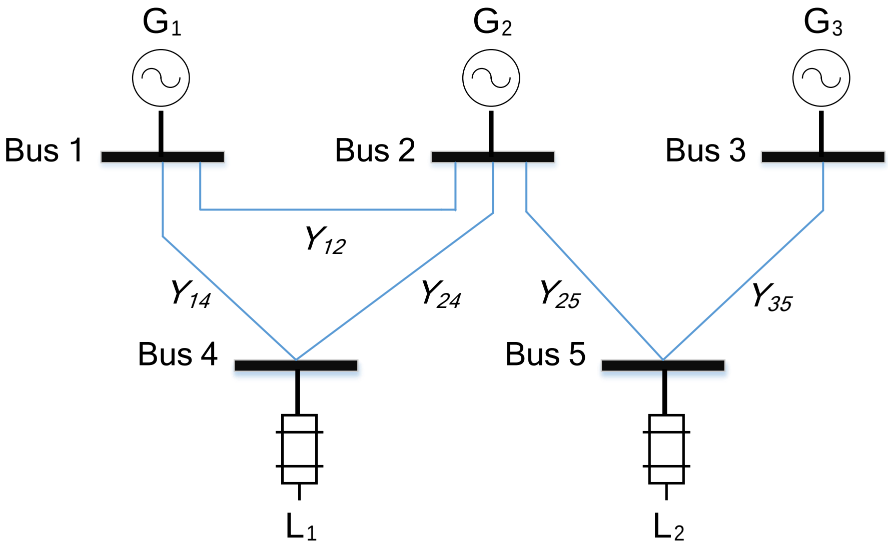


Figure 5.2: IEEE 5 Bus Power Network.

The Y-Bus of the IEEE 5 bus is first constructed using Eq. (5.9) as follows.

$$Y = \begin{bmatrix} y_{11} + y_{12} + y_{14} & -y_{12} & 0 & -y_{14} & 0 \\ -y_{21} & y_{21} + y_{22} + y_{24} + y_{25} & 0 & -y_{24} & -y_{25} \\ 0 & 0 & y_{33} + y_{35} & 0 & -y_{35} \\ -y_{41} & -y_{42} & 0 & y_{41} + y_{42} + y_{44} & 0 \\ 0 & -y_{52} & -y_{53} & 0 & y_{52} + y_{53} + y_{55} \end{bmatrix} \tag{5.16}$$

Knowing the Y-Bus and voltage V at each bus, the power (or current) injections that every bus contributes to the overall network is computed as

$$I = YV. \tag{5.17}$$

As such, the current injections in the IEEE 5 bus network can be represented by Eq. (5.18). The admittance values are taken from the line data for the IEEE 5 bus, which is included in Appendix B at the end of the thesis.

$$\begin{bmatrix} I_{11} \\ I_{22} \\ I_{33} \\ I_{44} \\ I_{55} \end{bmatrix} = \begin{bmatrix} 3-j7.9 & -2+j4 & 0 & -1+j4 & 0 \\ -2+j4 & 6-j12.7 & 0 & -2+j5 & -2+j4 \\ 0 & 0 & 2-j2.6 & 0 & -2+j3 \\ -1+j4 & -2+j5 & 0 & 3-j8.8 & 0 \\ 0 & -2+j4 & -2+j3 & 0 & 4-j6.7 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix} \tag{5.18}$$

Knowing the full form of the Y-Bus for the IEEE 5 bus, the complete state and observation models can be derived. In this example, the set of observations are the power injections at each bus. Note that in the following derivations, the demarcation of time, t, and the subscript appended to the end of the generator constants and control inputs, i, have been omitted to save on space.

State Model for Bus 1:

$$\dot{V}_1 = \frac{E_f - V_1}{T_{do}} + \frac{X_d - X_d^{'}}{T_{do}} \Big\{ V_2 \big(-2\sin(-\theta_2) - 4\cos(-\theta_2) \big) + V_4 \big(-\sin(-\theta_4) - 4\cos(-\theta_4) \big) \Big\} + \xi_1 \tag{5.19}$$

$$\dot{\theta}_1 = 0 \tag{5.20}$$

$$\dot{\omega}_1 = \frac{-D\omega_1 + P_m}{J} - \frac{1}{J} - \Big\{ V_1 V_2 \big(4\sin(-\theta_2) + 2\cos(-\theta_2) \big) + V_1 V_4 \big(4\sin(-\theta_4) + \cos(-\theta_4) \big) \Big\} + \xi_2 \tag{5.21}$$

State Model for Bus 2:

$$\dot{V}_2 = \frac{E_f - V_2}{T_{do}} + \frac{X_d - X_d^{'}}{T_{do}}\Big\{V_1\big(-2\sin(\theta_2) - 4\cos(\theta_2)\big) + V_4\big(-2\sin(\theta_2 - \theta_4)$$

$$- 5\cos(\theta_2 - \theta_4)\big) + V_5\big(-2\sin(\theta_2 - \theta_5) - 4\cos(\theta_2 - \theta_5)\big)\Big\} + \xi_3 \tag{5.22}$$

$$\dot{\theta}_2 = \omega_2 + \xi_4 \tag{5.23}$$

$$\dot{\omega}_2 = \frac{-D\omega_2 + P_m}{J} - \frac{1}{J} - \Big\{V_1V_2\big(4\sin(\theta_2) + 2\cos(\theta_2)\big) + V_2V_4\big(5\sin(\theta_2 - \theta_4)$$

$$+ 2\cos(\theta_2 - \theta_4)\big) + V_2V_5\big(4\sin(\theta_2 - \theta_5) + 2\cos(\theta_2 - \theta_5)\big)\Big\} + \xi_5 \tag{5.24}$$

State Model for Bus 3:

$$\dot{V}_3 = \frac{E_f - V_3}{T_{do}} + \frac{X_d - X_d^{'}}{T_{do}}\Big\{V_5\big(-2\sin(\theta_3 - \theta_5) - 3\cos(\theta_3 - \theta_5)\big)\Big\} + \xi_6 \tag{5.25}$$

$$\dot{\theta}_3 = \omega_3 + \xi_7 \tag{5.26}$$

$$\dot{\omega}_3 = \frac{-D\omega_3 + P_m}{J} - \frac{1}{J} - \Big\{V_5V_3\big(3\sin(\theta_3 - \theta_5) + 2\cos(\theta_3 - \theta_5)\big)\Big\} + \xi_8 \tag{5.27}$$

State Model for Bus 4:

$$\dot{V}_4 = -1\Big\{V_4V_1\big(-\sin(\theta_4) - 4\cos(\theta_4)\big) + V_4V_2\big(-2\sin(\theta_4 - \theta_2)$$

$$- 5\cos(\theta_4 - \theta_2)\big)\Big\} + \xi_9 \tag{5.28}$$

$$\dot{\theta}_4 = -1\Big\{V_4V_1\big(-\cos(\theta_4) - 4\sin(\theta_4)\big) + V_4V_2\big(-2\cos(\theta_4 - \theta_2)$$

$$- 5\sin(\theta_4 - \theta_2)\big)\Big\} + \xi_{10} \tag{5.29}$$

State Model for Bus 5:

$$\begin{aligned}\dot{V}_5 = -1\Big\{&V_5V_2\big(-2\sin(\theta_5-\theta_2)-4\cos(\theta_5-\theta_2)\big)+V_5V_3\big(-2\sin(\theta_5-\theta_3)\\&-3\cos(\theta_5-\theta_3)\big)\Big\}+\xi_{11}\end{aligned}\tag{5.30}$$

$$\begin{aligned}\dot{\theta}_5 = -1\Big\{&V_5V_2\big(-2\cos(\theta_5-\theta_2)-4\sin(\theta_5-\theta_2)\big)+V_5V_3\big(-2\cos(\theta_5-\theta_3)\\&-3\sin(\theta_5-\theta_3)\big)\Big\}+\xi_{12}\end{aligned}\tag{5.31}$$

Observations - Power Injections from Bus 1 - 5:

$$P_{11} = V_1\Big\{V_2\big(-2\cos(-\theta_2)+4\sin(-\theta_2)\big)+V_4\big(-\cos(-\theta_4)+4\sin(-\theta_4)\big)\Big\}+\zeta_1\tag{5.32}$$

$$\begin{aligned}P_{22} = V_2\Big\{&V_1\big(-2\cos(\theta_2)+4\sin(\theta_2)\big)+V_4\big(-2\cos(\theta_2-\theta_4)+5\sin(\theta_2-\theta_4)\big)\\&+V_5\big(-2\cos(\theta_2-\theta_5)+4\sin(\theta_2-\theta_5)\big)\Big\}+\zeta_2\end{aligned}\tag{5.33}$$

$$P_{33} = V_3\Big\{V_5\big(-2\cos(\theta_3-\theta_5)+3\sin(\theta_3-\theta_5)\big)\Big\}+\zeta_3\tag{5.34}$$

$$\begin{aligned}P_{44} = V_4\Big\{&V_1\big(-\cos(\theta_4)+4\sin(\theta_4)\big)+V_2\big(-2\cos(\theta_4-\theta_2)\\&+5\sin(\theta_4-\theta_2)\big)\Big\}+\zeta_4\end{aligned}\tag{5.35}$$

$$\begin{aligned}P_{55} = V_5\Big\{&V_2\big(-2\cos(\theta_5-\theta_2)+4\sin(\theta_5-\theta_2)\big)+V_3\big(-2\cos(\theta_5-\theta_3)\\&+3\sin(\theta_5-\theta_3)\big)\Big\}+\zeta_5\end{aligned}\tag{5.36}$$

5.2.1 Simulation Results of System Model

As illustrated in Fig. 5.3, implementing the state model can be done recursively by creating a function which contains the discretized set of equations from Eqs. (5.19) - (5.31). First, initial conditions are applied to the state vector for the very first time index. The evolution of the state model is then achieved by recursively executing the state model function. As is the norm with recursive functions, the output produced by the function is used as the input to the function in the subsequent iteration. This is typically achieved in any programming language using a sequential loop which executes until a specified time index. The simulation results for the state variables of the IEEE 5 bus are presented in Figs. 5.4 - 5.6.

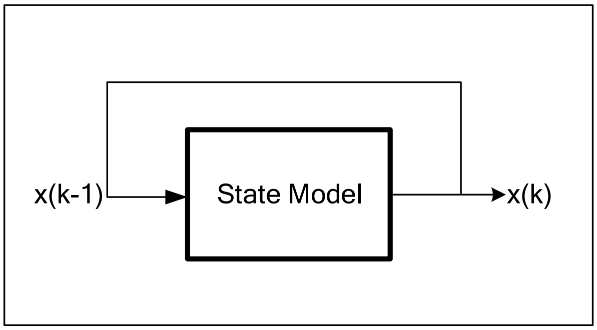


Figure 5.3: Recursive Implementation of the State Model.

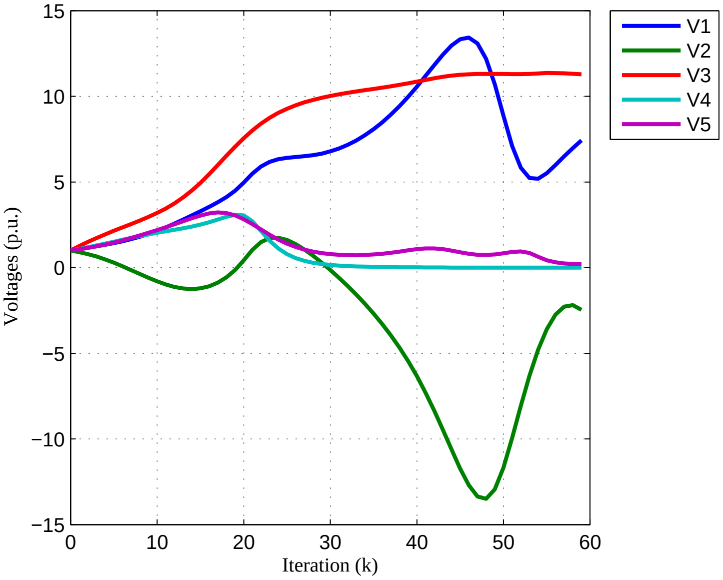


Figure 5.4: IEEE 5 Bus - Voltages.

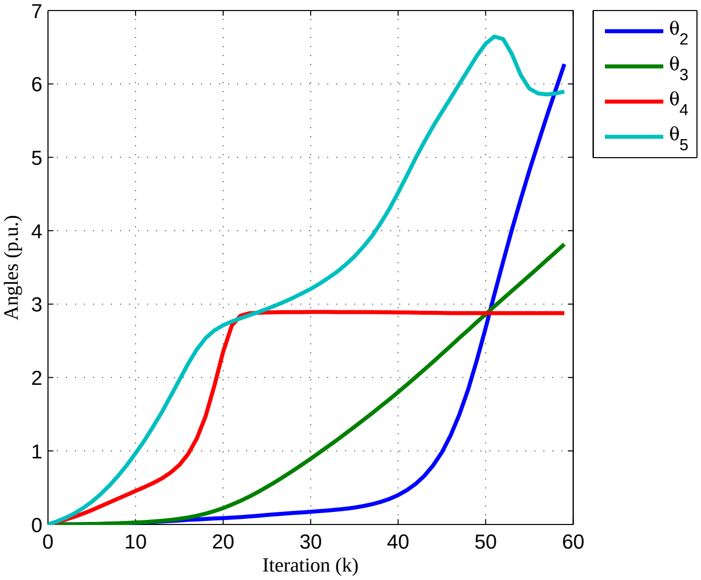


Figure 5.5: IEEE 5 Bus - Angles.

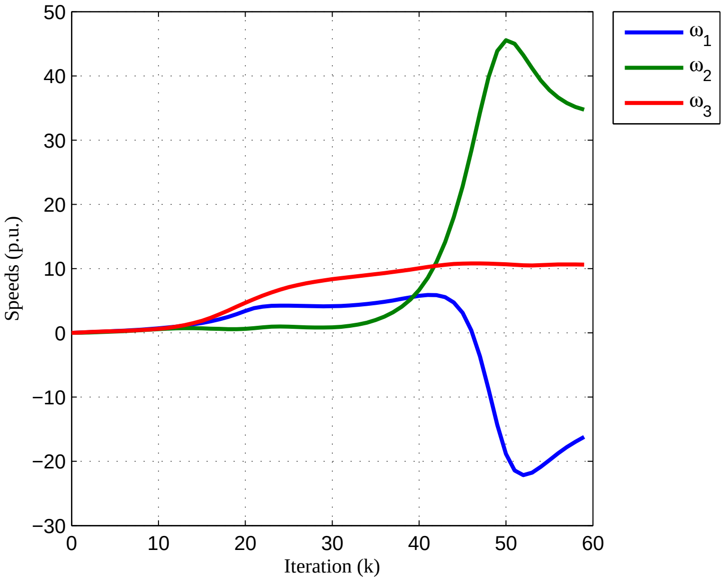


Figure 5.6: IEEE 5 Bus - Speeds.

As part of the initial conditions for the state vector in this simulation, the voltage for each bus is set to 1 while all other values are set to zero. Alternatively, a power flow analysis can be done within the network to compute the initial conditions as well. Additionally, the simulations run for a total of 60 time steps, with the duration of one time step being 0.01s. It is important to note that in the figures of the voltages, angles, and speeds of the IEEE 5 bus, the evolution of the state variables over time is without oscillation. Power system stability is of utmost importance, and any oscillation of the state variable at any time indicates instability within the network.

5.3 Centralized Particle Filter Implementation

With the system and observation modeling complete, the implementation of the Centralized Particle Filter (CPF) is now introduced in this section. The objective of the CPF is to estimate the state values of the EPG using measurements related to a subset of state variables. The observations are corrupted with noise to account for instrumentation error and system uncertainty. To simulate the corruptness of these measurements and the system in general, separate Gaussian noise vectors are included in the state model as well as in the observation model ($\boldsymbol{\xi}(t)$ and $\boldsymbol{\zeta}(t)$). These noise vectors are controlled by a specified signal-to-noise ratio (SNR) level measured in decibels (dB). The lower the SNR is, the higher the randomness of the noise added to the system. Low SNR values are used to test the resiliency and robustness of the state estimation algorithm.

The purpose of the experiment is to run the state model of the power network and establish a *ground truth* for its state values. These values are computed on a recursive basis and then used to generate observations using the observation model. Both sets of values are corrupted with noise having a predefined SNR. The same process is repeated with the particle filter where each particle's observations are compared with that of the generated observations in the previous step. In the next step, the multivariate probability density (likelihood) of the particles is calculated.

Resampling is used to discard those particles with the least weight. This completes one iteration of the CPF. The particle filter, given enough resources (particles), should provide an estimate that converges close to the ground truth [21].

Fig. 5.7 shows a flowchart for the implementation of the CPF. Major functions are represented by dashed rectangles, while their corresponding subfunctions are represented by normal rectangles. It must be noted that the process of establishing the ground truth (running the state model and using these values to update the observation model) does not necessarily need to be run in its entirety before the filtering process begins. The ground truth can be established iteratively in the same loop as the filter, provided that it is established *before* the filter processes the new observation.

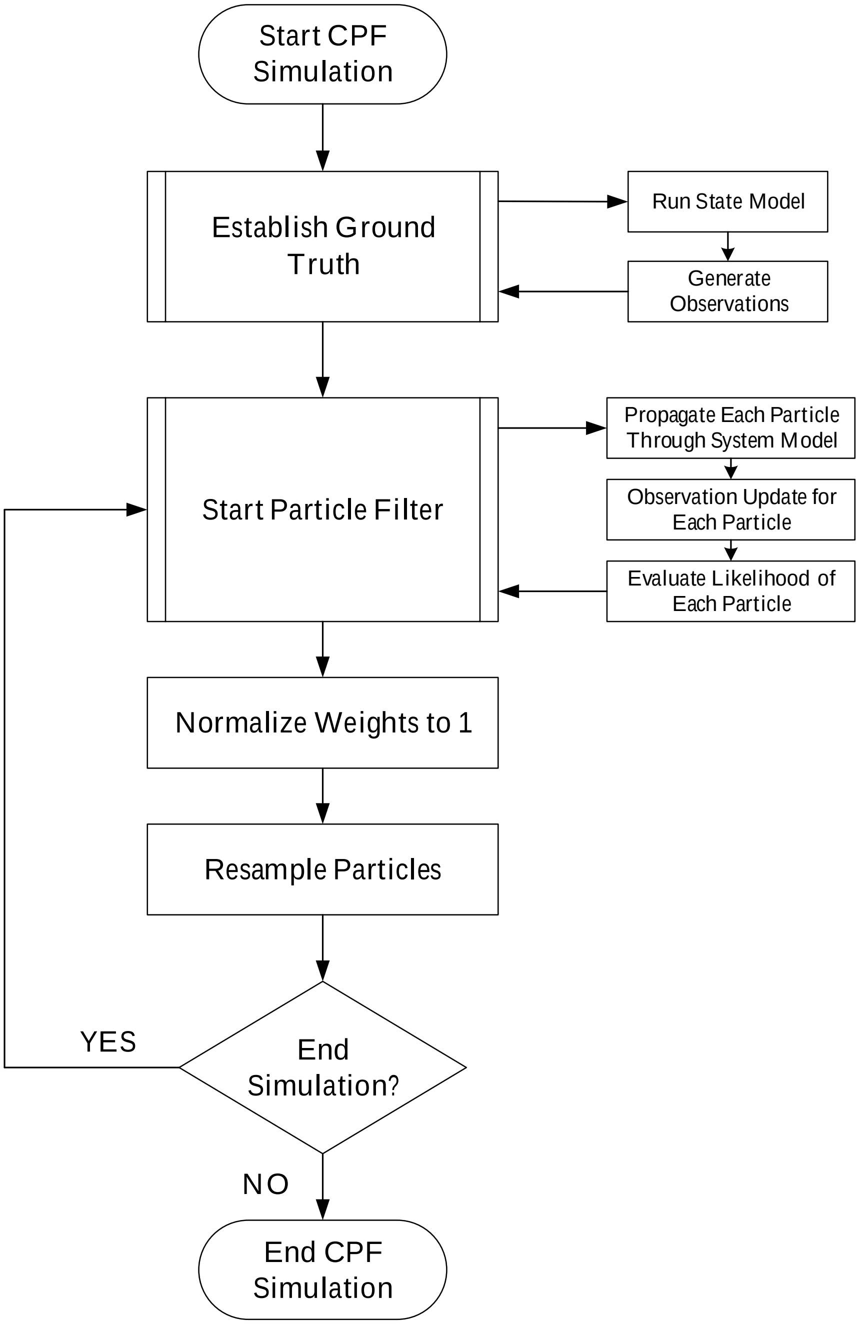


Figure 5.7: The Centralized Particle Filter.

5.4 Distributed Particle Filter Configuration and Implementation

5.4.1 Implementation

In this section, a reduced-order model of the IEEE 5 bus is presented, which partitions the overall subsystem into several subsystems. A distributed implementation of the particle filter is then implemented in order to estimate the state values from a limited set of observations. Each subsystem directly observes *only* the state variables included in its state model. Two important terms were defined in Chapter 4, which are related to partitioning the overall system into n_{sub} subsystems. A forcing term, $\boldsymbol{d}$, is defined as a state variable that is not *directly* observed by a subsystem but is still part of the subsystem's state model and is needed to solve the equations in the subsystem. This forcing term can be borrowed from a neighbouring subsystem. Secondly, a shared state is defined as a state variable which is observed by more than one subsystem. As each subsystem will compute a potentially different estimate of the state, a consensus step is required to combine these estimates together to maintain consistency of estimates across subsystems. Fig. 5.8 presents a flowchart which illustrates the implementation of the reduced-order, distributed particle filter (DPF).

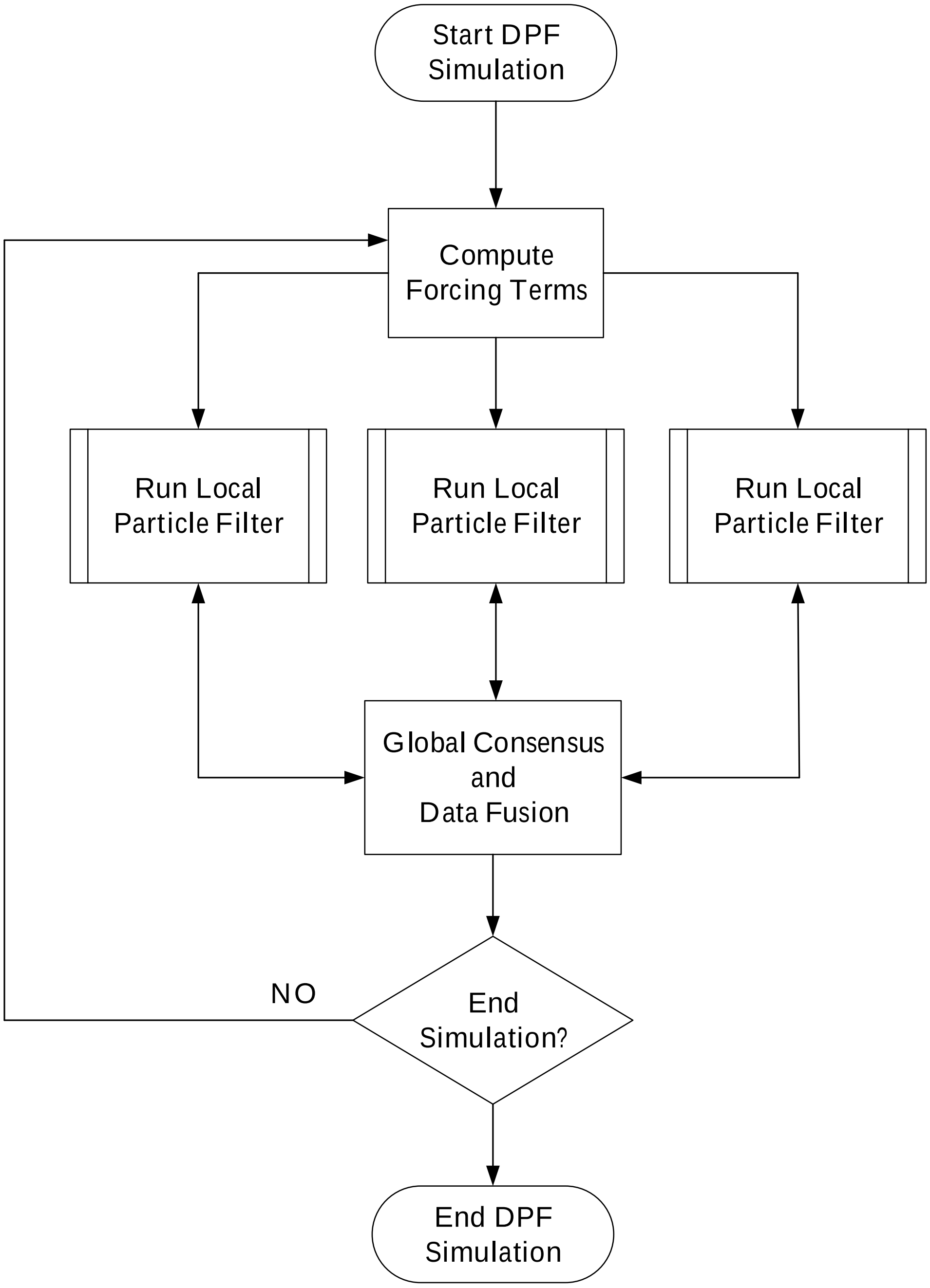


Figure 5.8: The Reduced-order Distributed Particle Filter.

5.4.2 Reduced-order Configuration

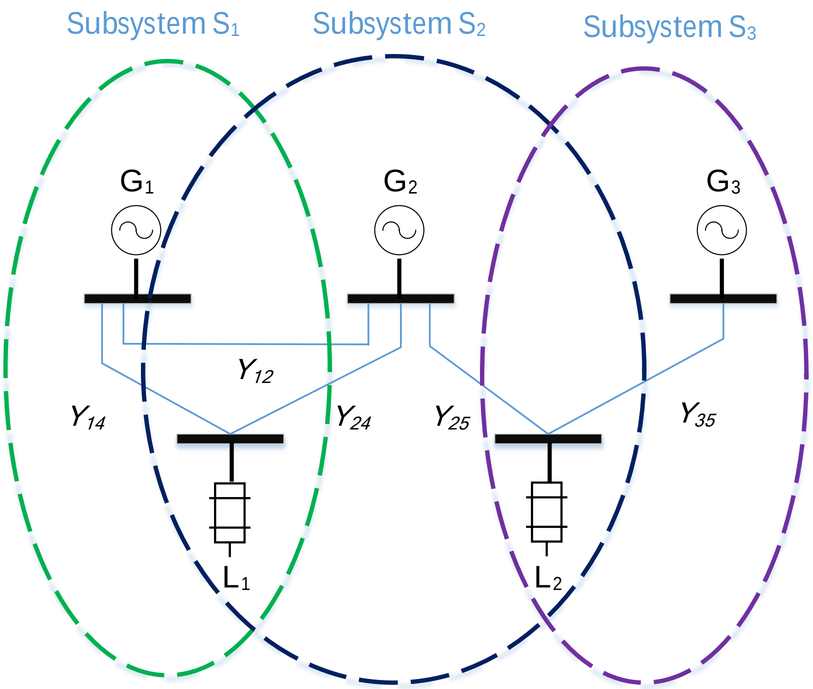


Figure 5.9: IEEE 5 Bus Partitioned Into 3 Subsystems.

Fig. 5.9 decomposes the IEEE 5 Bus system into 3 subsystems. It can be seen that bus 4 (corresponding to Load L_1) is shared between subsystem 1 and 2, while bus 5 (corresponding to Load L_2) is shared between subsystem 2 and 3. Both busses are load busses which have two state variables each, leading to a total of 4 shared states for this particular configuration. To find the forcing terms, the state models for subsystems S_1, S_2, and S_3 are presented below.

State Model Subsystem 1 - Bus 1 and 4:

$$\dot{V}_1 = \frac{E_f - V_1}{T_{do}} + \frac{X_d - X_d^{'}}{T_{do}}\Big\{\underbrace{V_2}_{}\big(-2\mathrm{sin}(-\underbrace{\theta_2}_{}) - 4\cos(-\underbrace{\theta_2)}_{}\big)$$

$$+ V_4\big(-\sin(-\theta_4) - 4\cos(-\theta_4)\big)\Big\} + \xi_1 \tag{5.37}$$

$$\dot{\theta}_1 = 0 \tag{5.38}$$

$$\dot{\omega}_1 = \frac{-D\omega_1 + P_m}{J} - \frac{1}{J} - \Big\{V_1\underbrace{V_2}_{}\big(4\sin(-\underbrace{\theta_2}_{}) + 2\cos(-\underbrace{\theta_2)}_{}\big)$$

$$+ V_1V_4\big(4\sin(-\theta_4) + \cos(-\theta_4)\big)\Big\} + \xi_2 \tag{5.39}$$

$$\dot{V}_4 = -1\Big\{V_4V_1\big(-\sin(\theta_4) - 4\cos(\theta_4)\big) + V_4\underbrace{V_2}_{}\big(-2\sin(\theta_4 - \underbrace{\theta_2}_{})$$

$$- 5\cos(\theta_4 - \underbrace{\theta_2)}_{}\big)\Big\} + \xi_3 \tag{5.40}$$

$$\dot{\theta}_4 = -1\Big\{V_4V_1\big(-\cos(\theta_4) - 4\sin(\theta_4)\big) + V_4\underbrace{V_2}_{}\big(-2\cos(\theta_4 - \underbrace{\theta_2)}_{}$$

$$- 5\sin(\theta_4 - \underbrace{\theta_2)}_{}\big)\Big\} + \xi_4 \tag{5.41}$$

Forcing Terms for Subsystem 1:

$\boldsymbol{d}^{(1)} = [V_2,\ \theta_2]$

State Model Subsystem 2 - Busses 2, 4 and 5:

$$\dot{V}_2 = \frac{E_f - V_2}{T_{do}} + \frac{X_d - X_d^{'}}{T_{do}}\Big\{\underbrace{V_1}_{}\big(-2\sin(\theta_2) - 4\cos(\theta_2)\big) + V_4\big(-2\sin (\theta_2 - \theta_4) - 5\cos(\theta_2 - \theta_4)\big) + V_5\big(-2\sin(\theta_2 - \theta_5) - 4\cos(\theta_2 - \theta_5)\big)\Big\} + \xi_1 \tag{5.42}$$

$$\dot{\theta}_2 = \omega_2 + \xi_2 \tag{5.43}$$

$$\dot{\omega}_2 = \frac{-D\omega_2 + P_m}{J} - \frac{1}{J} - \Big\{\underbrace{V_1}_{} V_2\big(4\sin(\theta_2) + 2\cos(\theta_2)\big) + V_2V_4\big(5\sin (\theta_2 - \theta_4) + 2\cos(\theta_2 - \theta_4)\big) + V_2V_5\big(4\sin(\theta_2 - \theta_5) + 2\cos(\theta_2 - \theta_5)\big)\Big\} + \xi_3 \tag{5.44}$$

$$\dot{V}_4 = -1\Big\{V_4\underbrace{V_1}_{}\big(-\sin(\theta_4) - 4\cos(\theta_4)\big) + V_4V_2\big(-2\sin(\theta_4 - \theta_2) - 5\cos(\theta_4 - \theta_2)\big)\Big\} + \xi_4 \tag{5.45}$$

$$\dot{\theta}_4 = -1\Big\{V_4\underbrace{V_1}_{}\big(-\cos(\theta_4) - 4\sin(\theta_4)\big) + V_4V_2\big(-2\cos(\theta_4 - \theta_2) - 5\sin(\theta_4 - \theta_2)\big)\Big\} + \xi_5 \tag{5.46}$$

$$\dot{V}_5 = -1\Big\{V_5V_2\big(-2\sin(\theta_5 - \theta_2) - 4\cos(\theta_5 - \theta_2)\big) + V_5\underbrace{V_3}_{}\big(-2\sin (\theta_5 - \underbrace{\theta_3)}_{} -3\cos(\theta_5 - \underbrace{\theta_3)}_{}))\Big\} + \xi_6 \tag{5.47}$$

$$\dot{\theta}_5 = -1\Big\{V_5V_2\big(-2\cos(\theta_5 - \theta_2) - 4\sin(\theta_5 - \theta_2)\big) + V_5\underbrace{V_3}_{}\big(-2\cos (\theta_5 - \underbrace{\theta_3)}_{} -3\sin(\theta_5 - \underbrace{\theta_3)}_{}))\Big\} + \xi_7 \tag{5.48}$$

Forcing Terms for Subsystem 2:

$\boldsymbol{d}^{(2)} = [V_1, V_3,\ \theta_3]$

State Model Subsystem 3 - Bus 3 and 5:

$$\dot{V}_3 = \frac{E_f - V_3}{T_{do}} + \frac{X_d - X_d^{'}}{T_{do}} \Big\{ V_5\big(-2\sin(\theta_3 - \theta_5) - 3\cos(\theta_3 - \theta_5)\big)\Big\} + \xi_1 \tag{5.49}$$

$$\dot{\theta}_3 = \omega_3 + \xi_2 \tag{5.50}$$

$$\dot{\omega_3} = \frac{-D\omega_3 + P_m}{J} - \frac{1}{J} - \Big\{ V_5 V_3 \big(3\sin(\theta_3 - \theta_5) + 2\cos(\theta_3 - \theta_5)\big)\Big\} + \xi_3 \tag{5.51}$$

$$\dot{V}_5 = -1\Big\{ V_5 \underbrace{V_2}_{} \big(-2\sin(\theta_5 - \underbrace{\theta_2}_{}) - 4\cos(\theta_5 - \theta_2)\big) + V_5 V_3 \big(-2\sin (\theta_5 - \theta_3) - 3\cos(\theta_5 - \theta_3)\big)\Big\} + \xi_3 \tag{5.52}$$

$$\dot{\theta}_5 = -1\Big\{ V_5 \underbrace{V_2}_{} \big(-2\cos(\theta_5 - \underbrace{\theta_2}_{}) - 4\sin(\theta_5 - \theta_2)\big) + V_5 V_3 \big(-2\cos (\theta_5 - \theta_3) - 3\sin(\theta_5 - \theta_3)\big)\Big\} + \xi_4 \tag{5.53}$$

Forcing Terms for Subsystem 3:

$\boldsymbol{d}^{(3)} = [V_2,\, \theta_2]$

5.5 Simulation Results - IEEE 14 Bus Network

The theory and code implementation of the previous chapter is now expanded to the IEEE 14 bus system, which is a test power network that represents a portion of the American Midwest power grid. Fig. 5.10 shows the IEEE 14 bus, while Fig. 5.11 shows a proposed reduced-order configuration. The IEEE 14 bus consists of 5 generators and 9 load busses for a total of 32 state variables. The observations for this simulation consists of power injections at all the generator busses and 15 additional power flow measurements between busses for a total of 20 observations.

The distributed configuration partitions the overall model into four subsystems with a total of 32 forcing terms and 8 shared terms. The centralized particle filter uses 500 particles to estimate the posterior density of the state, meaning $32 \times 500 = 16000$ particles are used in total. In the distributed implementation, shared states are additionally estimated, and as such $16000/(32 + 8) = 400$ particles are used for each subsystem to ensure a fair experiment.

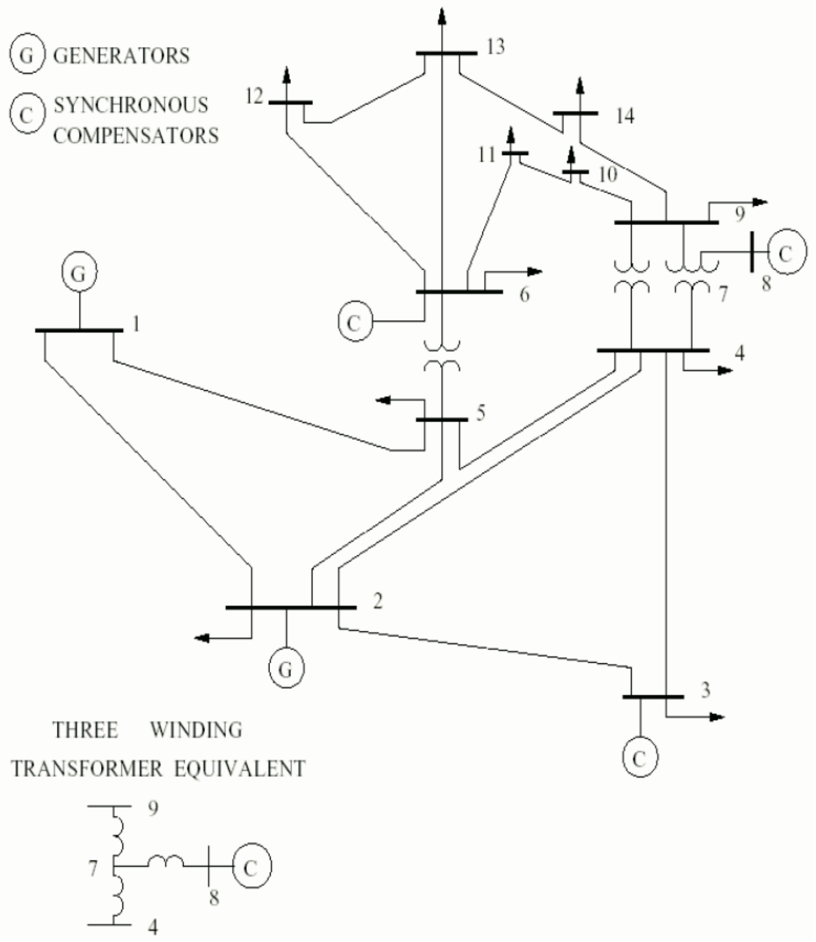

G GENERATORS
C SYNCHRONOUS COMPENSATORS
13
12
14
11
10
9
C
8
7
6
C
G
1
4
5
2
G
3
C
THREE WINDING TRANSFORMER EQUIVALENT
9
7
C
8
4

Figure 5.10: IEEE 14 Bus Network.

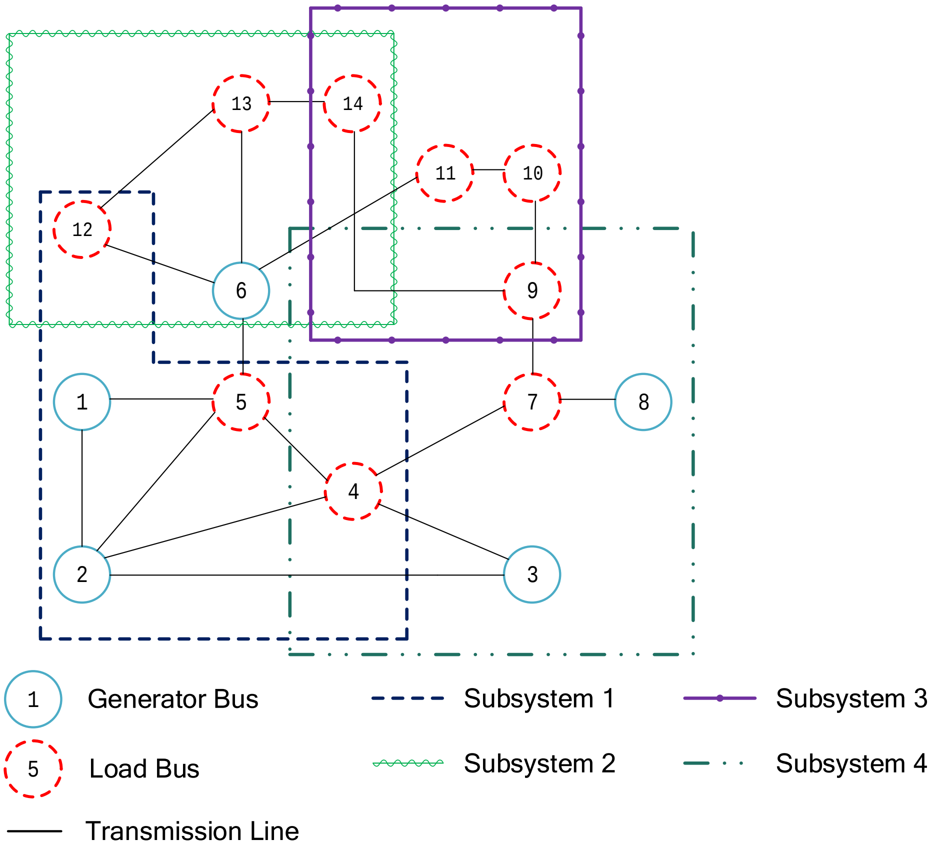


Figure 5.11: The IEEE 14 Bus Network Partitioned Into 4 Subsystems.

Both filters use a SNR of 30 dB for the process and measurement noise vectors to provide a degree of uncertainty in the experiment. The results obtained for the simulations are generated over 100 Monte Carlo runs.

The main focus of the experiment is to ensure that both the centralized particle filter and the reduced-order distributed particle filter track the noisy state values accurately. Even though the centralized particle filter will provide the optimal result, the goal is to ensure that the reduced-order distributed particle filter provides a reasonable approximation to the centralized implementation at a much lower computational cost.

In order to illustrate the impact of the noisy state values, Fig. 5.12 shows the true, clean evolution of the state variable ω_2, which represents the speed at generator 2. A corrupted version of the same state variable (at a SNR of 30 dB) is plotted alongside the ground truth. It can be noted that the evolution of the true state value is smooth, while the noisy signal is quite coarse and unstable. Performing state estimation using noisy state and observation values pose significant challenges for both the centralized and distributed particle filters.

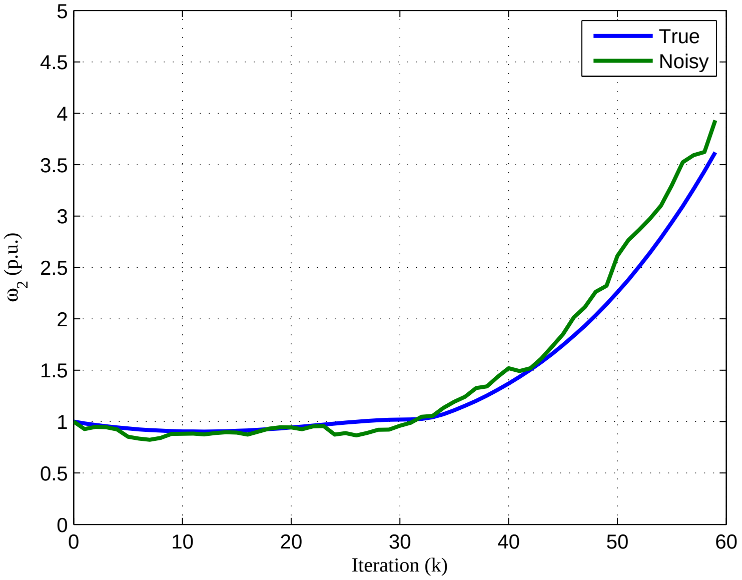
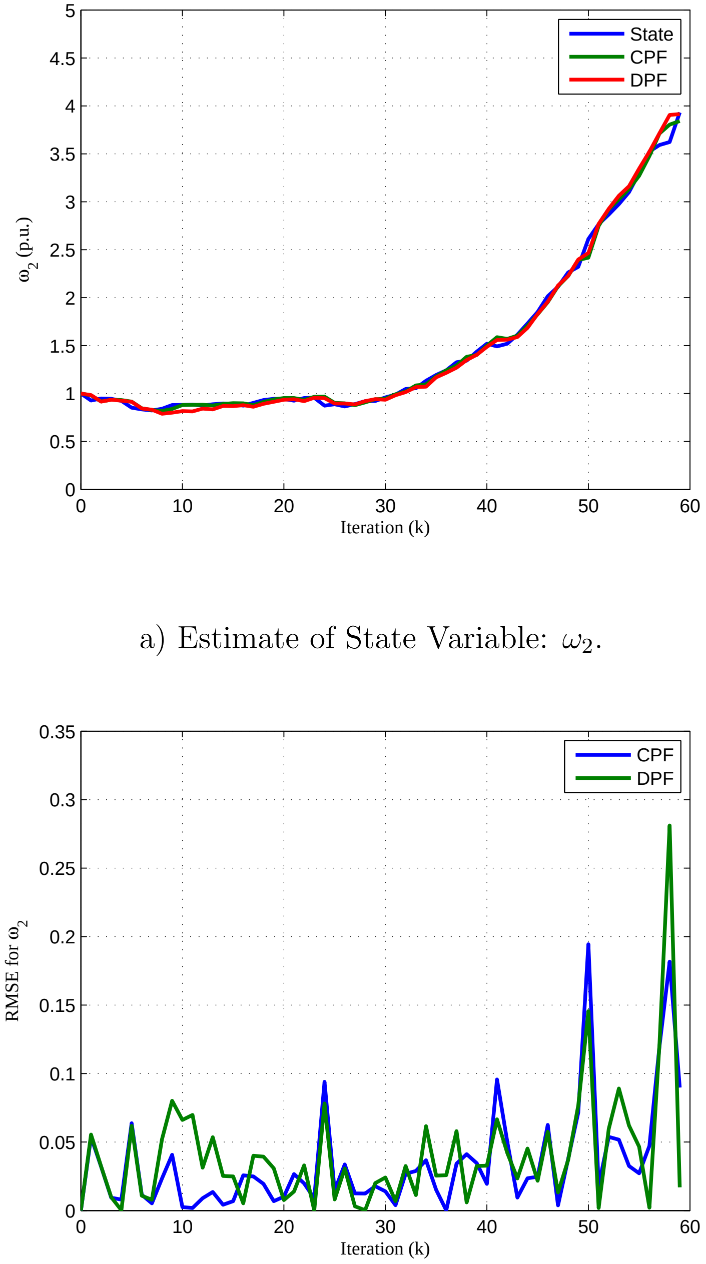


Figure 5.12: Corrupted State Value of ω_2.

Fig. 5.13 shows the noisy state value of Fig. 5.12 alongside the tracking results of the CPF and DPF. In spite of the perturbations in the evolution of the state variable, both the CPF and DPF provide accurate estimates. This is corroborated by the accompanying root mean square error (RMSE) plot for both filters, where it can be seen that the deviation of both filters compared to the ground truth is minimal. Quantitatively, the prediction error can be calculated by using the following formula

$$\text{Prediction Error} = \left\{ \frac{|\text{ Actual - Prediction }|}{\text{Actual}} \right\} \times 100\%. \tag{5.54}$$

Using the above formula, the maximum prediction error can be computed by taking the actual and estimated value of both filters at the point of maximum deviation. Using this approach, the maximum prediction error for the CPF and DPF are 7.45% and 7.76%. These two factors not only indicate a high degree of accuracy, but also, that the DPF operates at an accuracy that is very close to its centralized counterpart. This indicates that the reduced-order, distributed particle filter is indeed an acceptable alternative to the CPF at a much lower computational cost.



a) Estimate of State Variable: ω_2.

b) RMSE of State Estimate for ω_2.

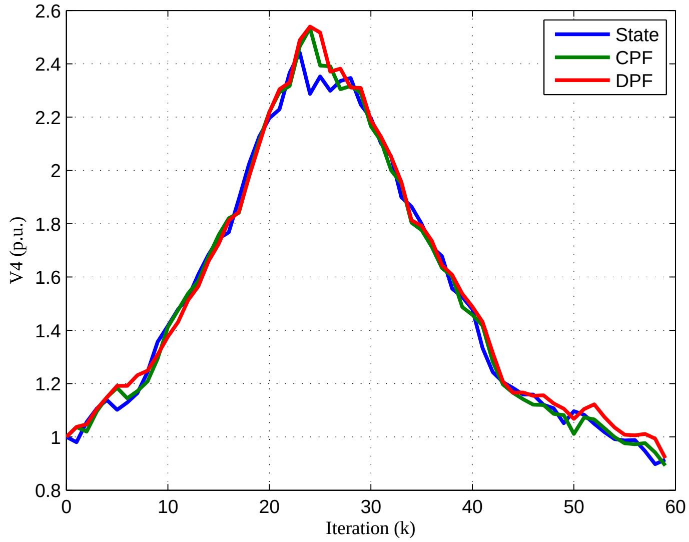
Figure 5.13: CPF vs DPF for Estimating ω_2.

Fig. 5.14 shows the voltage profile of the load bus at node 4, with both filters tracking the state value with a high degree of accuracy. Fig. 5.14(a) in particular shows that the CPF almost converges to the ground truth value at the end of the simulation. This is confirmed in the RMSE plot in Fig. 5.14(b), which shows that both filters improve their accuracy as time evolves. The CPF is proven to be more accurate according to the RMSE plots, but with a maximum prediction error of 0.94% compared to 1.92% of the DPF, it can be seen that the reduced-order DPF approaches the high levels of accuracy of the CPF as well.

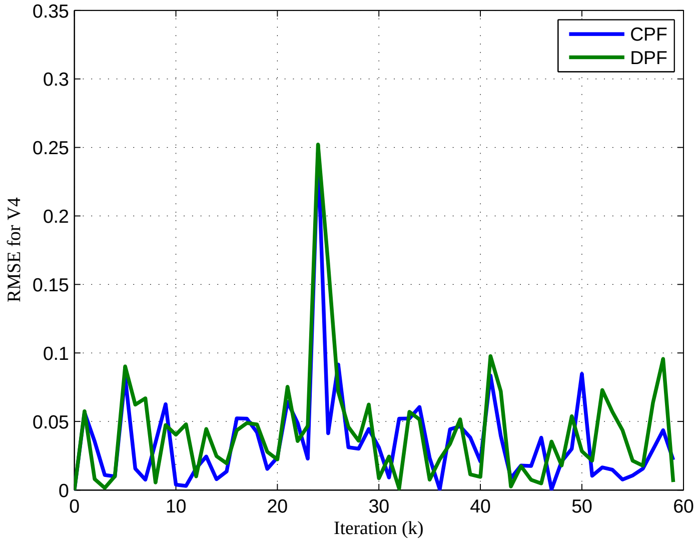
5.6 Estimation Divergence

It is important to take note of the situations in which the state estimation system provides incorrect results that diverge from the ground truth. The first situation is when there is an extremely high amount of noise that is present in the system. In the simulations, this is represented as the low SNR scenario. Low SNRs for both the process and observation vectors will affect the state estimates and the error in the estimation will increase for both centralized and distributed implementations. Although the error will diverge from the ground truth in both centralized and distributed implementations, the filters have the capacity to recover over time as more observations are incorporated.

The second cause of divergence is specific to the distributed implementation of the particle filter. In some cases, the consensus step for the distributed particle filter may not converge within two consecutive observations. This may happen in networks with intermittent connectivity, where loss in connectivity during the consensus step affects communication between local subsystems. If the estimator does not achieve consensus for the local estimates within two successive observations, the performance of the distributed estimator is severely degraded and the estimator may not only diverge but become unstable and not recover at all. Reference [55] deals with such a scenario, where a new implementation of the distributed particle filter is proposed to deal with networks with intermittent connectivity. Using the methodology presented in this thesis, the distributed approach suggested in [55] is applicable to the electric power grid and microgrids.



a) Estimate of State Variable: V_4.



b) RMSE of State Estimate for V_4.

Figure 5.14: CPF vs DPF for Estimating V_4.

Chapter 6 - State Estimation in Islanded Microgrids

The state estimation implementation techniques covered in Chapter 5 will now be applied to microgrids operating in the islanded mode of operation. In Section 6.1, a short review of microgrids will be followed by the detailed modeling of microgrid system components. In Section 6.2, a centralized state model will be derived for a simple 3 bus microgrid, followed by a reduced-order derivation in Section 6.3. These techniques will then be applied to a larger, 8 Bus microgrid network with accompanying simulation results in Section 6.4.

6.1 System Modeling

As discussed earlier, microgrids are a miniature model of a complete EPG in that they generate, transmit, and distribute power to local loads. However, unlike traditional large scale EPGs, microgrids differ in its philosophy of operation. Microgrids seek to amass distributed generation of renewable energy sources as its primary source of energy. In addition, microgrids are designed to service a much lower demand for power. As such, microgrids can be connected to the main power grid and used to inject power into the network when needed, or they can operate in the islanded mode where they satisfy local power demands.

When the microgrid is operating in grid connected mode, the overall system dynamics are dominated by the main grid itself [13]. When operating in the islanded mode, the microgrid system dynamics are a function of its internal components which comprise of distributed generators (DGs), Lines, and Loads.

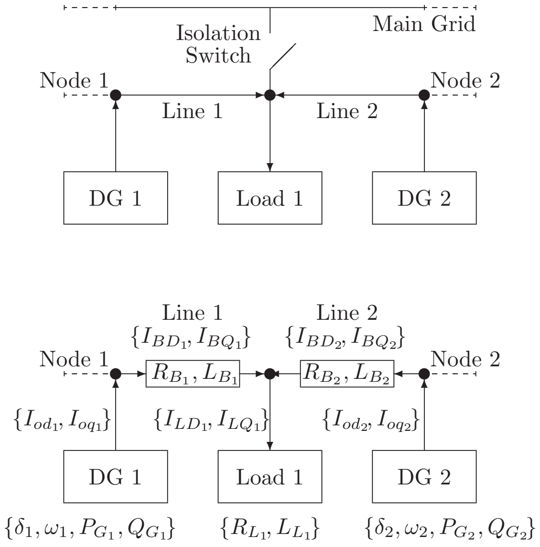


Figure 6.1: The State Variables of a Microgrid.

Fig. 6.1 is an example of a microgrid comprising of two DGs and a load. Each DG n supplies its output current $\{I_{oD_n}, I_{oQ_n}\}$ to a node that connects the DG to a transmission line. Modeled as a series resistor-inductor circuit, current $\{I_{BD_l}, I_{BQ_l}\}$ flowing on line l is the accumulative difference of DG source currents $\{I_{oD_n}, I_{oQ_n}\}$ and currents $\{I_{LD_m}, I_{LQ_m}\}$ consumed by the loads connected to line l.

6.1.1 DQ Reference Frame

The voltages and currents in any AC power network have three phases in a stationary phase coordinate system (commonly referred to as the ABC frame). Since network analysis in the ABC frame is complex, it is transformed to another framework with two phases (direct and quad) rotating about an axis. The transformed frame is called the DQ frame and is symbolized as (d_n, q_n). In a microgrid, each DG rotates at its own angular frequency ω_n leading to several individual DQ frames as shown in Fig. 6.2.

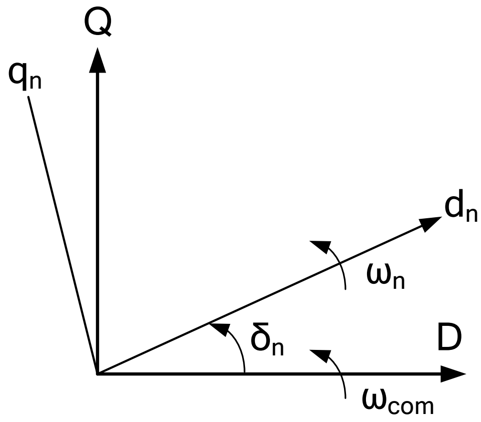


Figure 6.2: DQ-dq Reference Frame.

In order to analyze the overall system, all state variables in the microgrid are further transformed from their individual reference frames (d_n, q_n) to a reference DQ frame (typically the one associated with DG 1) by defining an angle δ_n, which represents the phase difference between the individual (d_n, q_n) frame and the reference DQ frame. Since the reference DG is already aligned to the DQ axis, its angle is effectively zero. Using this fact, the rotational frequency can be calculated and is referred to as ω_{com}, which in turn can be used to compute the phase angles δ_n associated with every other DG in the network. Once the angle for each DG has been obtained, the state variables (say f_d, f_q) of the individual DGs are mapped to (f_D, f_Q) in the reference DQ frame using the following transformation:

$$\begin{bmatrix} f_D \\ f_Q \end{bmatrix} = \begin{bmatrix} \cos(\delta_n) & -\sin(\delta_n) \\ \sin(\delta_n) & \cos(\delta_n) \end{bmatrix} \begin{bmatrix} f_d \\ f_q \end{bmatrix} \tag{6.1}$$

6.1.2 DG Modeling

The DG is responsible for producing power for the network to use and uses a voltage source inverter to convert the DC power provided by renewable energy sources into AC power. As such, the DG is modeled by a set of non-linear ordinary differential equations that couple the DG and inverter together.

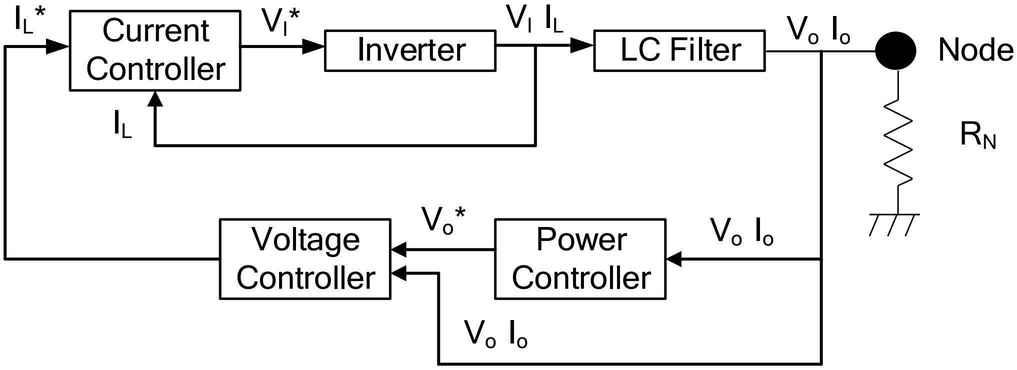


Figure 6.3: Block Diagram of the DG-Inverter.

Fig. 6.3 shows a block diagram of the internal control loops of the DG. The most important controller is the Power Controller, whose primary objective is to efficiently share the load of the network among the DGs. This is achieved by using a droop control strategy among the DGs in the network. A system parameter known as the droop gain is configured for each DG and allows the DGs to share power appropriately in the network. If the droop gain for all the DGs in the network is set equal, each DG contributes an equal amount of power towards servicing the load. The power controller receives the output voltage and current of the LC Filter (V_o and I_o) and sets the output magnitude and phase of the voltage ($V_o{}^*$).

The Voltage and Current Controllers, which are designed to reject high frequency disturbances and provide adequate damping for the output LC Filter, are then used to compute the final DC Voltage ($V_I{}^*$). The Inverter then converts the

DC Voltage to AC and this voltage is filtered by the LC Filter in order to attenuate the frequency ripple of the inverter [61]. The output current of the LC filter is what flows to the node (bus) and the rest of the network. Additionally, the addition of virtual resistor (R_N) at the node is to ensure that the numerical solution to the system is well defined. Since the Power Controller and Output LC Filter dominate system dynamics for the DG, the voltage and current controllers can be omitted to simplify the DG model [62].

In the simplified DG-inverter coupled model, each DG unit is modeled by a set of 5 state variables: δ_n is the angle associated with DG n; $\{P_{G_n}, Q_{G_n}\}$ the active and reactive power generated by the DG; I_{od_n} the output current in the d^{th} dimension, and; I_{oq_n} the output current in the q^{th} dimension. The non-linear state model for DG n consists of the following equations

$$\dot{\delta}_n(t) = \omega_n^* - M_{p_n} P_{G_n} - \omega_{\text{com}} \tag{6.2}$$

$$\dot{P}_{G_n}(t) = 1.5\omega_{cn}[V_{o_n}^* I_{od_n} - N_{q_n} Q_{G_n} I_{od_n}] - \omega_{cn} P_{G_n} \tag{6.3}$$

$$\dot{Q}_{G_n}(t) = 1.5\omega_{cn}[V_{o_n}^* I_{oq_n} - N_{q_n} Q_{G_n} I_{oq_n}] - \omega_{cn} Q_{G_n} \tag{6.4}$$

$$L_{c_n} \dot{I}_{od_n}(t) = -R_{c_n} I_{od_n} + \omega_{\text{com}} I_{oq_n} L_{c_n} + V_{od_n} - Vb_{D_n} \tag{6.5}$$

$$L_{c_n} \dot{I}_{oq_n}(t) = -R_{c_n} I_{oq_n} - \omega_{\text{com}} I_{od_n} L_{c_n} + V_{oq_n} - Vb_{Q_n} \tag{6.6}$$

Note that: $\mathrm{V}_{od_n} = \mathrm{V}_{o_n}^*$ - $\mathrm{N}_{q_n}\mathrm{Q}_{G_n}$

where ω_n^* is the DG output voltage angular frequency; $\{M_{p_n}, N_{q_n}\}$ the active and reactive droop gain; ω_{c_n} the cutoff frequency of the output filter; $V_{o^*_n}$ the nominal output voltage magnitude set point; R_{c_n} the resistance of the output filter; and L_{c_n} the inductance of the output filter. These parameters are all constants. Originally expressed in the local dq frame, the DG parameters in (6.2)-(6.6) are transformed to the reference DQ frame using (6.1).

Parameters $\{V_{bD_p}, V_{bQ_p}\}$ are the DQ-components of the nodal voltage at node p. To ensure the solution of the microgrid network is well grounded, a virtual resistor R_p of a high value (e.g. $1M\Omega$) is placed at each node [13]. Using Kirchhoff's current law, the nodal voltage is a function of the load and line currents (these state variables are to be discussed in the next subsection) entering and leaving the node.

$$V_{bD_p}{=}R_p\Big(\Sigma I_{oD_{p(\mathrm{in})}}+\Sigma I_{BD_{p(\mathrm{in})}}-\Sigma I_{BD_{p(\mathrm{out})}}-\Sigma I_{LD_{p(\mathrm{out})}}\Big) \tag{6.7}$$

$$V_{bQ_p}{=}R_p\Big(\Sigma I_{oQ_{p(\mathrm{in})}}+\Sigma I_{BQ_{p(\mathrm{in})}}-\Sigma I_{BQ_{p(\mathrm{out})}}-\Sigma I_{LQ_{p(\mathrm{out})}}\Big) \tag{6.8}$$

where subscript '(in)' denotes current entering the node and '(out)' current leaving the node. For example, notation $\{\Sigma I_{oD_{p(\mathrm{in})}}, \Sigma I_{oQ_{p(\mathrm{in})}}\}$ represent the DQ components of the accumulative current generated by DGs that enters node p. Likewise, $\{\Sigma I_{LD_{p(\mathrm{out})}}, \Sigma I_{LQ_{p(\mathrm{out})}}\}$ represent the DQ components of the accumulative current consumed by loads that leaves node p. Other variables use similar notations.

6.1.3 Line Modeling

The physical transmission lines connecting nodes are modeled as a series resistor-inductor circuit. For line l, the state variables are the line currents $\{I_{BD_l}, I_{BQ_l}\}$ given by

$$L_{B_l}\dot{I}_{BD_l}(t){=}{-}R_{B_l}I_{BD_l}{+}\omega_{\text{com}}L_{B_l}I_{BQ_l}{+}V_{BD_{l(\text{to})}}{-}V_{BD_{l(\text{from})}} \tag{6.9}$$

$$L_{B_l}\dot{I}_{BQ_l}(t){=}{-}R_{B_l}I_{BQ_l}{-}\omega_{\text{com}}L_{B_l}I_{BD_l}{+}V_{BQ_{l(\text{to})}}{-}V_{BQ_{l(\text{from})}} \tag{6.10}$$

where R_{B_l} is the resistance of line l and L_{B_l} is the inductance of line l. Also, $V_{BD_{l(\text{to})}}$ denotes the D-component of the voltage of the *to* bus and $V_{BD_{l(\text{from})}}$ the D-component of the voltage of the *from* bus.

6.1.4 Load Modeling

The loads are modeled by their admittances. The states for load m are its current $\{I_{LD_m}, I_{LQ_m}\}$ in the DQ frame

$$L_{L_m}\dot{I}_{LD_m}(t){=}{-}R_{L_m}I_{LD_m} + \omega_{\text{com}}L_{L_m}I_{LQ_m} + V_{BD_m} \tag{6.11}$$

$$L_{L_m}\dot{I}_{LQ_m}(t){=}{-}R_{L_m}I_{LQ_m} - \omega_{\text{com}}L_{L_m}I_{LD_m} + V_{BQ_m} \tag{6.12}$$

where $\{R_{L_m}, L_{L_m}\}$ are the resistance and inductance of load m, $\{V_{BD_m}, V_{BQ_m}\}$ the DQ-components of the bus voltage connected to load m and ω_{com} the rotational frequency of the reference DG.

6.1.5 Overall State Model and Observation Model

The overall state vector is formed by stacking all state variables corresponding to the DGs, lines, and loads in a vector

$$\begin{aligned} \boldsymbol{X}(t)=&\{\delta_n(t), P_{G_n}(t), Q_{G_n}(t), I_{oD_n}(t), I_{oQ_n}(t), \\ & I_{BD_l}, I_{BQ_l}(t), I_{LD_m}(t), I_{LQ_m}(t)\}_{n,l,m} \end{aligned} \tag{6.13}$$

which leads to the a set of non-linear ordinary differential equations as covered in Eq. (5.7).

As discussed previously, typical observations in a power network are a subset of active/reactive power flows, active/reactive power injections, and voltage/current magnitudes. In this case, PMUs are assumed to be installed at the nodes of the power network and the node voltages $\{V_{bD_p}, V_{bQ_p}\}$, in Eqs.(6.7)-(6.8) are considered as measurements.

6.1.6 Implementation of System Model

The system model can be described by the set of non-linear ordinary differential equations (ODEs) in Eq. (6.13). In the case of the microgrid model, the system of ODEs are extremely dynamic. When the terms of a differential equation cause rapid variation in the solution, the differential equation is classified as a stiff differential equation [63]. MATLAB specializes in solving stiff, non-linear differential equations and provides a set of special solvers for this purpose. The ODE solver *ode15s* is a variable order solver which uses two methods to integrate a system of differential equations: Numerical Differentiation Formulas (NDFs) or Backward Differentiation Formulas (BDFs) [64]. The ode15s solver is also a variable step solver, and will attempt to decrease the step size when rapid variations occur to capture dynamics as accurately as possible.

In Chapter 5, solving the state model for the transmission network was achieved by discretizing the system and solving recursively at a constant time step. For the highly dynamic microgrid system, this approach is not feasible because of the rapid variations of the state variables. In order to implement the system model recursively as demonstrated previously, the ODE solver must know at which specific time steps it is required to solve the system equations. As such, the system is first solved by the ode15s method in MATLAB in order to retrieve the time steps at which the method solves the system. The system model can then be solved in a recursive manner by solving for the system one time step at a time using the time steps determined above. Example MATLAB code and simulation results are shown below for the microgrid network illustrated in Fig. 6.1.

```
% Begin Simulation

% Set Simulation timings

t0 = 0;

tf = 3.5

tspan=[0,3.5];

% Number of States for microgrid network

numState = 15;

% Initialize initial conditions for first step of ODE solver

y0=zeros(1,numState);

% Run Microgrid Model to get solver points - in time

[T,YExp]=ode15s(@(t,y) centralStateModel(t,y),tspan,y0);

% Run State Model

time = T;

X = zeros(numState,length(time));

    % stateModel not only recieves previous outputs

    % but the time step to be solved for

    for t = 2:length(time)

        X(:,t) = stateModel(X(:,t-1),[time(t-1) time(t)]);

    end

% Active power of DGs are index 2 and 7

activePowerIndexes =[2,7];

% Generate figure

plot

figure(time, X(activePowerIndexes,:))
```

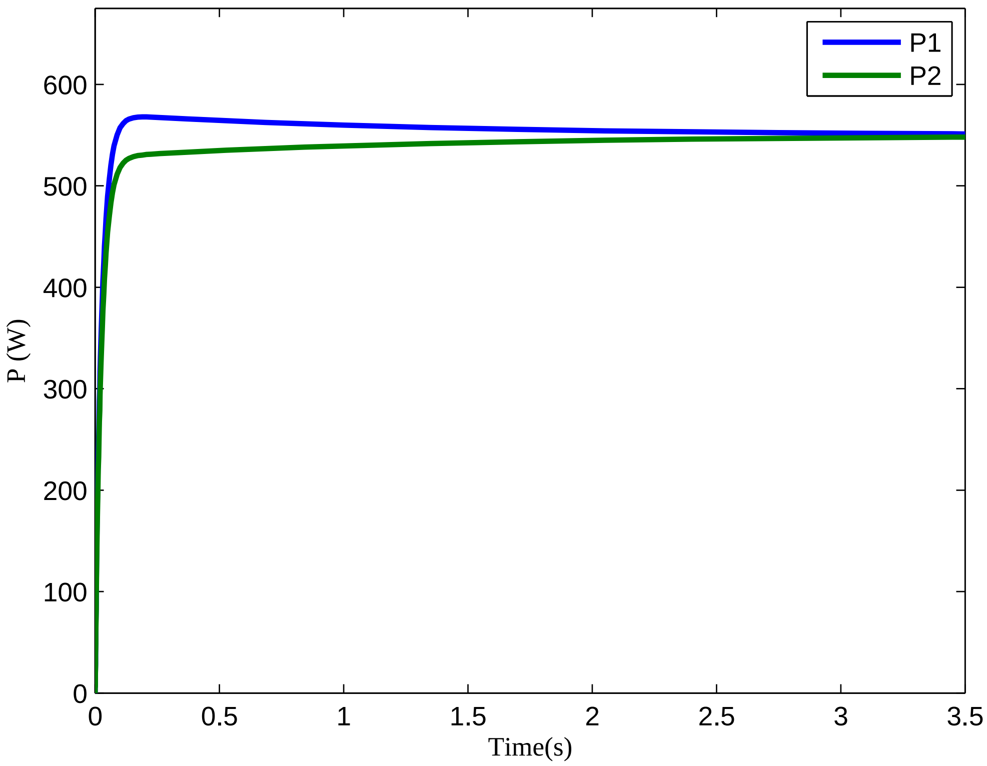


Figure 6.4: Active Power Generated by Both DGs.

In this simulation, the droop gain for both DGs are set equal to demonstrate the power sharing of the DGs. In Fig. 6.4, it can indeed be seen that the active power generated by both DGs in steady state is equal, with the power produced by both DGs converging to approximately 550 (W).

6.2 Centralized State Model

In this section, the centralized state model for an example 3 bus microgrid is derived. Fig. 6.5 shows the test microgrid which is configured of 2 DGs and a single load. This microgrid is operating in the islanded mode. The demarcation of time, t, has been omitted to save on space, and the values of the constants are included in Appendix C. The centralized state model of this microgrid is presented below.

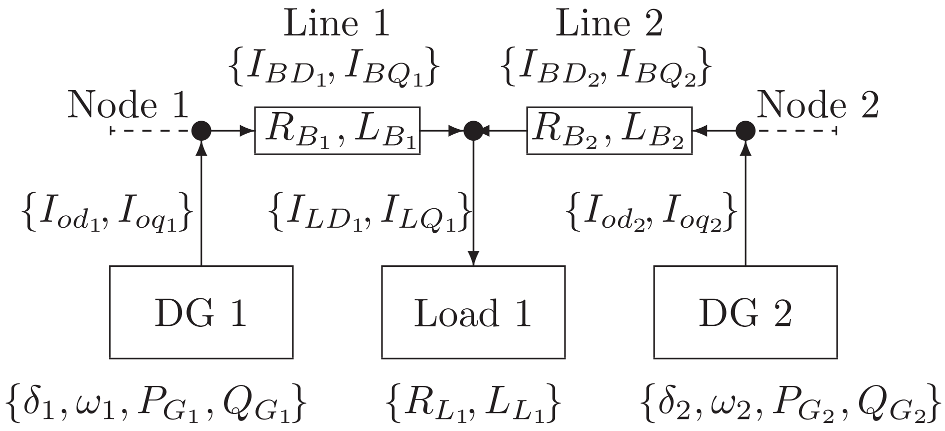


Figure 6.5: State Variables of the 3 Bus Microgrid.

State Model of DG 1

$$\dot{\delta}_1 = 0 \tag{6.14}$$

$$\dot{P}_{G_1} = 1.5\omega_{c1}[V^*_{o_1} I_{od_1} - N_{q_1} Q_{G_1} I_{od_1}] - \omega_{c1} P_{G_1} + \xi_1 \tag{6.15}$$

$$\dot{Q}_{G_1} = 1.5\omega_{c1}[V^*_{o_1} I_{oq_1} - N_{q_1} Q_{G_1} I_{oq_1}] - \omega_{c1} Q_{G_1} \quad + \xi_2 \tag{6.16}$$

$$L_{c_1} \dot{I}_{od_1} = -R_{c_1} I_{od_1} + \omega_{\text{com}} I_{oq_1} L_{c_1} + V_{od_1} - Vb_{D_1} + \xi_3 \tag{6.17}$$

$$L_{c_1} \dot{I}_{oq_1} = -R_{c_1} I_{oq_1} - \omega_{\text{com}} I_{od_1} L_{c_1} + V_{oq_1} - Vb_{Q_1} + \xi_4 \tag{6.18}$$

State Model of DG 2

$$\dot{\delta}_2 = \omega^*_2 - M_{p_2} P_{G_2} - \omega_{\text{com}} + \xi_5 \tag{6.19}$$

$$\begin{aligned} \dot{P}_{G_2} &= 1.5\omega_{c2}[V^*_{o_2}\big(I_{oq_2} * \sin(\delta_2) + I_{od_2} * \cos(\delta_2)\big) \\ &\quad - N_{q_2} Q_{G_1}\big(I_{oq_2} * \sin(\delta_2) + I_{od_2} * \cos(\delta_2)\big)] - \omega_{c2} P_{G_2} + \xi_6 \end{aligned} \tag{6.20}$$

$$\begin{aligned} \dot{Q}_{G_2} &= -1.5\omega_{c2}[V^*_{o_2}\big(I_{oq_2} * \cos(\delta_2) - I_{od_2} * \sin(\delta_2)\big) \\ &\quad - N_{q_2} Q_{G_1}\big(I_{oq_2} * \cos(\delta_2) + I_{od_2} * \sin(\delta_2)\big)] - \omega_{c2} Q_{G_2} + \xi_7 \end{aligned} \tag{6.21}$$

$$L_{c_2} \dot{I}_{od_2} = -R_{c_2} I_{od_2} + \omega_{\text{com}} I_{oq_2} L_{c_2} + V_{od_2} * \cos(\delta_2) - Vb_{D_2} + \xi_8 \tag{6.22}$$

$$L_{c_2} \dot{I}_{oq_2} = -R_{c_2} I_{oq_2} - \omega_{\text{com}} I_{od_2} L_{c_2} + V_{oq_1} * \sin(\delta_2) - Vb_{Q_1} + \xi_9 \tag{6.23}$$

State Model of Line 1

$$L_{B_1} \dot{I}_{BD_1} = -R_{B_1} I_{BD_1} + \omega_{\text{com}} L_{B_1} I_{BQ_1} + V_{BD_{3(\text{to})}} - V_{BD_{1(\text{from})}} + \xi_{10} \tag{6.24}$$

$$L_{B_1} \dot{I}_{BQ_1} = -R_{B_1} I_{BQ_1} - \omega_{\text{com}} L_{B_1} I_{BD_1} + V_{BQ_{3(\text{to})}} - V_{BQ_{1(\text{from})}} + \xi_{11} \tag{6.25}$$

State Model of Line 2

$$L_{B_2}\dot{I}_{BD_2} = -R_{B_2}I_{BD_2} + \omega_{\text{com}}L_{B_2}I_{BQ_2} + V_{BD_{3(\text{to})}} - V_{BD_{2(\text{from})}} + \xi_{12} \tag{6.26}$$

$$L_{B_2}\dot{I}_{BQ_2} = -R_{B_2}I_{BQ_2} - \omega_{\text{com}}L_{B_2}I_{BD_2} + V_{BQ_{3(\text{to})}} - V_{BQ_{2(\text{from})}} + \xi_{13} \tag{6.27}$$

State Model of Load 1

$$L_{L_1}\dot{I}_{LD_1} = -R_{L_1}I_{LD_1} + \omega_{\text{com}}L_{L_1}I_{LQ_1} + V_{BD_1} + \xi_{14} \tag{6.28}$$

$$L_{L_1}\dot{I}_{LQ_1} = -R_{L_1}I_{LQ_1} - \omega_{\text{com}}L_{L_1}I_{LD_1} + V_{BQ_1} + \xi_{15} \tag{6.29}$$

Observation at Node 1

$$V_{BD_1} = R_n * \{I_{od_1} - I_{BD_1}\} + \zeta_1 \tag{6.30}$$

$$V_{BQ_1} = R_n * \{I_{oq_1} - I_{BQ_1}\} + \zeta_2 \tag{6.31}$$

Observation at Node 2

$$V_{BD_2} = R_n * \{I_{od_2} - I_{BD_2}\} + \zeta_3 \tag{6.32}$$

$$V_{BQ_2} = R_n * \{I_{oq_2} - I_{BD_2}\} + \zeta_4 \tag{6.33}$$

Observation at Node 3

$$V_{BD_3} = R_n * \{I_{BD_1} + I_{BD_2} - I_{LD_1}\} + \zeta_5 \tag{6.34}$$

$$V_{BQ_3} = R_n * \{I_{BQ_1} + I_{BQ_2} - I_{LQ_1}\} + \zeta_6 \tag{6.35}$$

6.3 Reduced-order State Model

As can be seen in Fig. 6.6, the 3 bus microgrid is partitioned into 2 subsystems. The reduced-order state model for each subsystem is derived in this section. Subsystem 1 contains DG 1, Line 1 (connecting nodes 1 and 3), as well as Load 1 for a total of 8 state variables (the angle of the first DG is taken as zero and therefore not counted as a state variable). Subsystem 2 mirrors subsystem 1 and contains DG 2, Line 2, and Load 1 for a total of 9 state variables. The state variables of the load are the shared variables of the network.

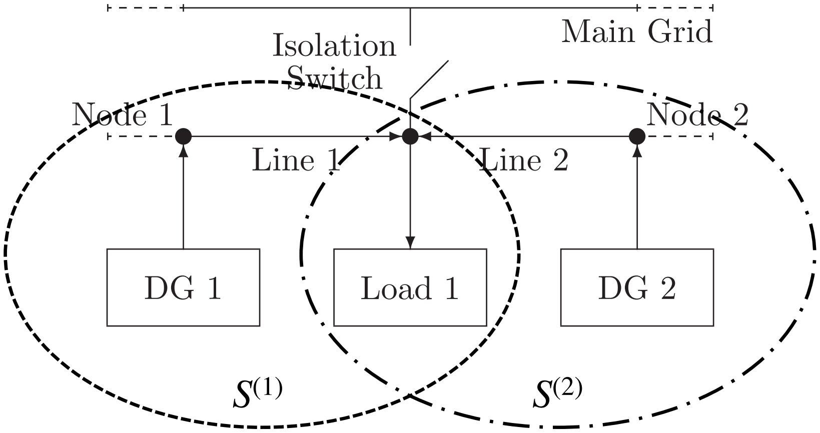


Figure 6.6: Microgrid Partitioned Into 2 Subsystems.

To find the forcing terms for each subsystem, the state modeling for the subsystems is presented below.

State Model Subsystem 1 - Nodes 1 and 3:

$$\dot{\delta}_1 = 0 \tag{6.36}$$

$$\dot{P}_{G_1} = 1.5\omega_{c1}[V^*_{o_1} I_{od_1} - N_{q_1} Q_{G_1} I_{od_1}] - \omega_{c1} P_{G_1} + \xi_1 \tag{6.37}$$

$$\dot{Q}_{G_1} = 1.5\omega_{c1}[V^*_{o_1} I_{oq_1} - N_{q_1} Q_{G_1} I_{oq_1}] - \omega_{c1} Q_{G_1} \quad + \xi_2 \tag{6.38}$$

$$L_{c_1} \dot{I}_{od_1} = -R_{c_1} I_{od_1} + \omega_{\text{com}} I_{oq_1} L_{c_1} + V_{od_1} - V b_{D_1} + \xi_3 \tag{6.39}$$

$$L_{c_1} \dot{I}_{oq_1} = -R_{c_1} I_{oq_1} - \omega_{\text{com}} I_{od_1} L_{c_1} + V_{oq_1} - V b_{Q_1} + \xi_4 \tag{6.40}$$

$$L_{B_1} \dot{I}_{BD_1} = -R_{B_1} I_{BD_1} + \omega_{\text{com}} L_{B_1} I_{BQ_1} + \underbrace{V_{BD_{3(\text{to})}}} - V_{BD_{1(\text{from})}} + \xi_5 \tag{6.41}$$

$$L_{B_1} \dot{I}_{BQ_1} = -R_{B_1} I_{BQ_1} - \omega_{\text{com}} L_{B_1} I_{BD_1} + \underbrace{V_{BQ_{3(\text{to})}}} - V_{BQ_{1(\text{from})}} + \xi_6 \tag{6.42}$$

$$L_{L_1} \dot{I}_{LD_1} = -R_{L_1} I_{LD_1} + \omega_{\text{com}} L_{L_1} I_{LQ_1} + V_{BD_1} + \xi_7 \tag{6.43}$$

$$L_{L_1} \dot{I}_{LQ_1} = -R_{L_1} I_{LQ_1} - \omega_{\text{com}} L_{L_1} I_{LD_1} + V_{BQ_1} + \xi_8 \tag{6.44}$$

Recall that $V_{BD_3} = R_N$ * {I_{BD_1} + $\underline{\mathrm{I}_{BD_2}}$ - I_{LD_1}} and that $V_{BQ_1} = R_N$ * {I_{BQ_1} + $\underline{\mathrm{I}_{BQ_2}}$ - I_{LQ_1}}. The state variables {I_{BD_2},I_{BQ_2}} are not directly observed by Subsystem 1, and thus are denoted forcing terms of Subsystem 1.

Forcing Terms for Subsystem 1:

$\boldsymbol{d}^{(1)}$= [I_{BD_2},I_{BQ_2}]

State Model Subsystem 2 - Nodes 2 and 3:

$$\dot{\delta}_2 = \omega_2^* - M_{p_2} P_{G_2} - \omega_{\text{com}} + \xi_1 \tag{6.45}$$

$$\begin{aligned}\dot{P}_{G_2} &= 1.5\omega_{c2}[V_{o_2}^* \big(I_{oq_2} * \sin(\delta_2) + I_{od_2} * \cos(\delta_2)\big) \\ &\quad - N_{q_2} Q_{G_1} \big(I_{oq_2} * \sin(\delta_2) + I_{od_2} * \cos(\delta_2)\big)] - \omega_{c2} P_{G_2} + \xi_2\end{aligned} \tag{6.46}$$

$$\begin{aligned}\dot{Q}_{G_2} &= -1.5\omega_{c2}[V_{o_2}^* \big(I_{oq_2} * \cos(\delta_2) - I_{od_2} * \sin(\delta_2)\big) \\ &\quad - N_{q_2} Q_{G_1} \big(I_{oq_2} * \cos(\delta_2) + I_{od_2} * \sin(\delta_2)\big)] - \omega_{c2} Q_{G_2} + \xi_3\end{aligned} \tag{6.47}$$

$$L_{c_2} \dot{I}_{od_2} = -R_{c_2} I_{od_2} + \omega_{\text{com}} I_{oq_2} L_{c_2} + V_{od_2} * \cos(\delta_2) - Vb_{D_2} + \xi_4 \tag{6.48}$$

$$L_{c_2} \dot{I}_{oq_2} = -R_{c_2} I_{oq_2} - \omega_{\text{com}} I_{od_2} L_{c_2} + V_{oq_1} * \sin(\delta_2) - Vb_{Q_1} + \xi_5 \tag{6.49}$$

$$L_{B_2} \dot{I}_{BD_2} = -R_{B_2} I_{BD_2} + \omega_{\text{com}} L_{B_2} I_{BQ_2} + \underbrace{V_{BD_{3(\text{to})}}} - V_{BD_{2(\text{from})}} + \xi_6 \tag{6.50}$$

$$L_{B_2} \dot{I}_{BQ_2} = -R_{B_2} I_{BQ_2} - \omega_{\text{com}} L_{B_2} I_{BD_2} + \underbrace{V_{BQ_{3(\text{to})}}} - V_{BQ_{2(\text{from})}} + \xi_7 \tag{6.51}$$

$$L_{L_1} \dot{I}_{LD_1} = -R_{L_1} I_{LD_1} + \omega_{\text{com}} L_{L_1} I_{LQ_1} + V_{BD_1} + \xi_8 \tag{6.52}$$

$$L_{L_1} \dot{I}_{LQ_1} = -R_{L_1} I_{LQ_1} - \omega_{\text{com}} L_{L_1} I_{LD_1} + V_{BQ_1} + \xi_9 \tag{6.53}$$

Similar to the previous subsystem, the nodal voltage V_{BD_3} contains the terms $\{\mathrm{I}_{BD_1},\mathrm{I}_{BQ_1}\}$ that are not directly observed by Subsystem 2.

Forcing Terms for Subsystem 2:

$\boldsymbol{d}^{(2)} = [\mathrm{I}_{BD_1},\mathrm{I}_{BQ_1}]$

6.4 Simulation Results - 8 Bus Network

Similar to the experiment in the previous chapter, the centralized particle filter and reduced-order, distributed particle filter are used to estimate the state vector for a 8 bus, 5 DG Microgrid operating at 110V and 60Hz frequency shown in Fig. 6.7. Both the state and observation vectors are corrupted with a SNR of 30 dB. The microgrid is configured for a blank start, meaning the initial conditions for all state variables are zero.

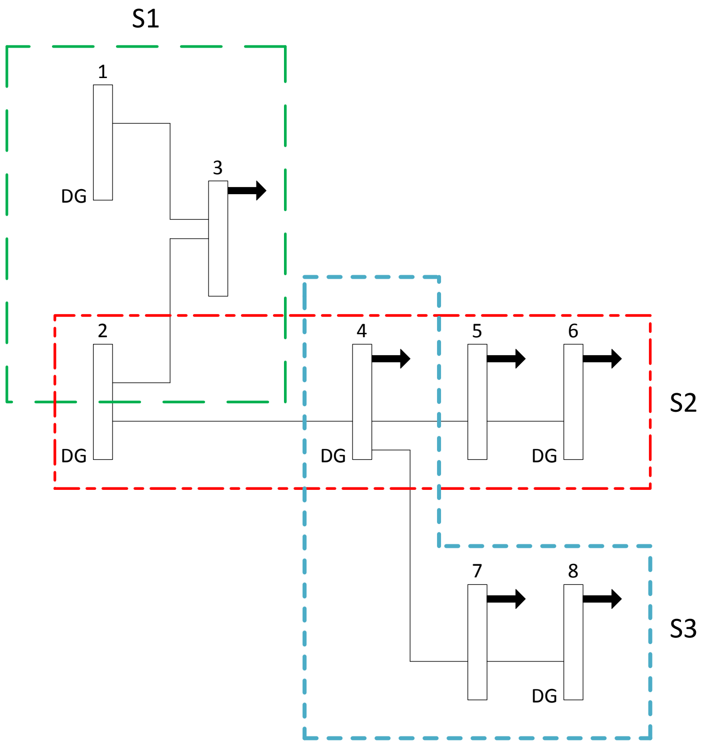


Figure 6.7: 8 Bus Microgrid.

The microgrid is spatially decomposed into 3 subsystems $\{S^{(1)}, S^{(2)}, S^{(3)}\}$, with the observations assumed to come from PMUs which measure the voltage at each node. In the centralized particle filter implementation, a total of 20 particles are used for each state variable in order to represent its posterior density, and with 51 state variables in total, this leads to 1020 total particles being used for the experiment. In order to maintain the same number of particles as in the centralized implementation, a total of $1020/(51 + 12) \approx 16$ particles/state are used in each subsystem in the distributed implementation. As in the centralized case, all states are initialized to zero for a blank start. With regards to the droop setting of the DGs, all DGs are assigned the same droop value in order to equally share the power demand of the loads.

Fig. 6.8 shows the output current of the DG at node 4 (I_{oD_4}) and the estimates provided by the CPF and DPF between 0 and 0.03 seconds. Compared to the simulations of the modern transmission networks in Chapter 5, microgrid state variables are extremely dynamic and have a very fast rise time. The time steps used by the MATLAB differential equation solver for this network range from milliseconds to microseconds. To exemplify this fact, the output current in Fig. 6.8 reaches its maximum point of 3.7463 amperes (A) in 4.8ms, partly due to the fact that since the microgrid is configured for a blank start, the DGs react to the power demand of the loads instantly before settling to steady state. Both filters do not perform to an exceedingly high level of accuracy in this part of the simulation, with maximum prediction errors being 11.31% and 15.13% for the CPF and DPF, respectively. However, as Fig. 6.9 shows, both filters do improve very quickly thereafter. Fig. 6.9(a) shows the performance of both filters for the duration of the simulation, while Fig. 6.9(b) shows the corresponding RMSE plots. The RMSE plots in particular show that after an inaccurate start, both filters continue to improve their performance until the end of the simulation. The prediction error at the end of the simulation for the CPF is 5.87%, while for the DPF it is 5.98%.

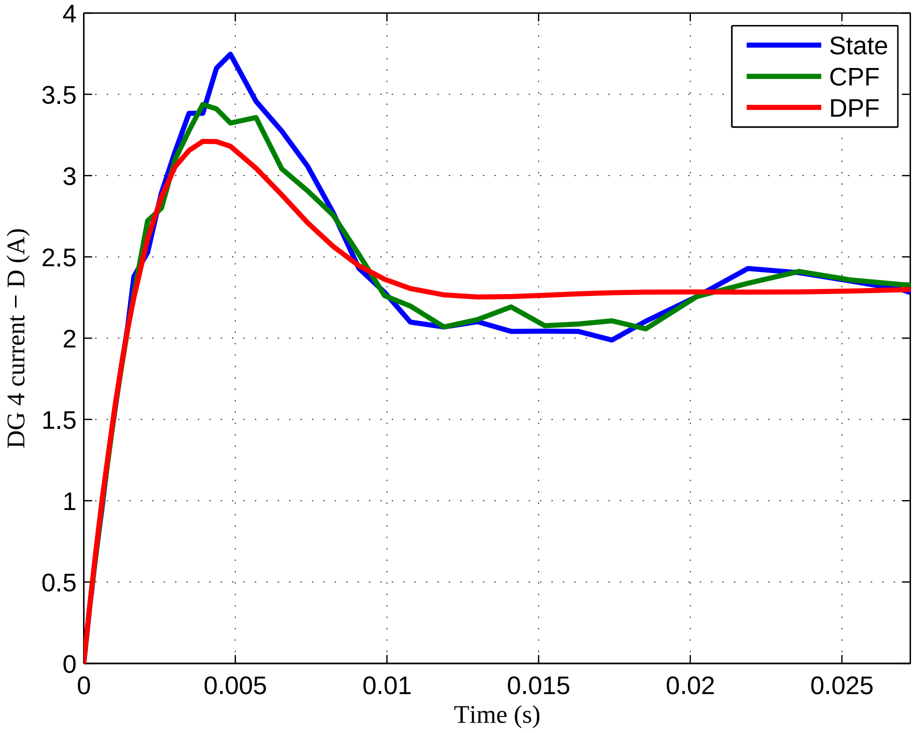
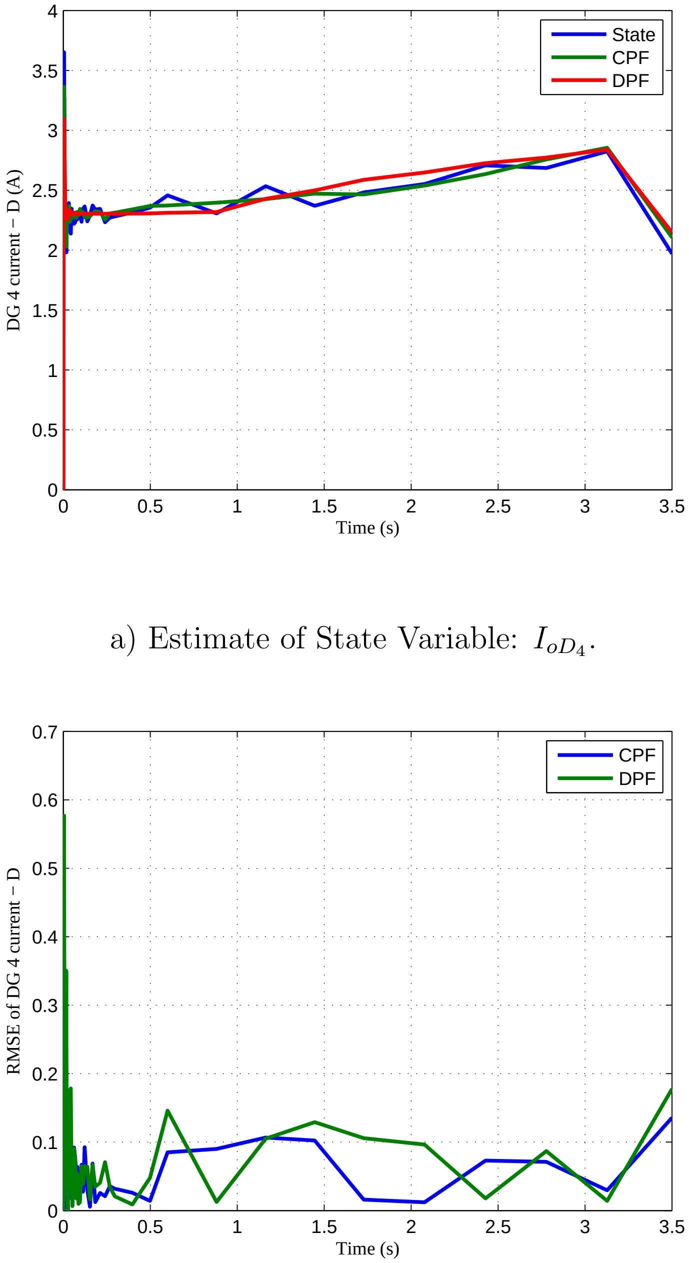


Figure 6.8: State Estimate of CPF and DPF During Rise Time For I_{oD_4}.



a) Estimate of State Variable: I_{oD_4}.

b) RMSE of State Estimate for I_{oD_4}.

Figure 6.9: CPF vs DPF for Estimating I_{oD_4}.

Fig. 6.10 shows the output current of the load node 8 (I_{Load_8}) in the D frame, and the estimates provided by the CPF and DPF between 0 and 0.07 seconds. Again, it can be seen that the rise time of the current is extremely fast, and both filters lag slightly behind the evolution of the state variable. The maximum error prediction for both filters in this period is 11.16% for the CPF and 16.67% for the DPF. However, as shown in Fig. 6.11(a), the CPF in particular recovers very quickly and begins to catch up to the state variable by the 0.01 second mark. The RMSE plots in Fig. 6.11(b) confirm that after a relatively inaccurate start, both filters converge to a prediction error of almost zero at the end of the simulation. The prediction errors at this point are 0.15% for the CPF, and 0.58% for the DPF.

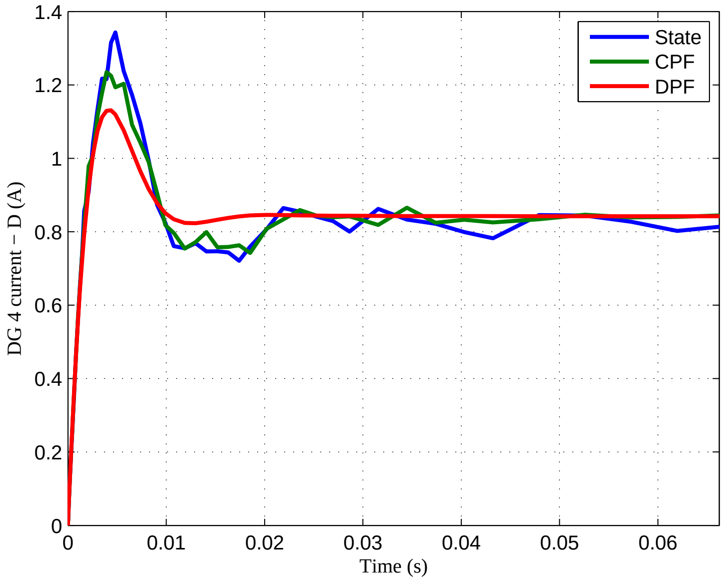
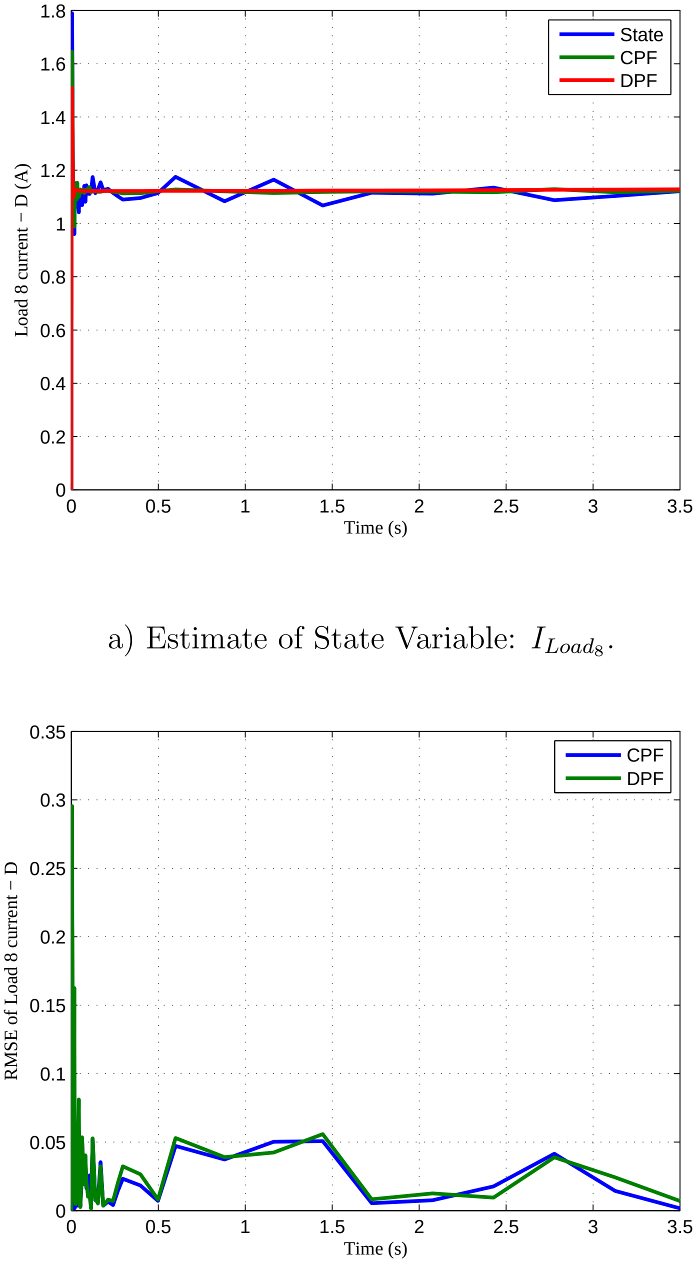


Figure 6.10: State Estimate of CPF and DPF During Rise Time for I_{Load_8}.



a) Estimate of State Variable: I_{Load_8}.

b) RMSE of State Estimate for I_{Load_8}.

Figure 6.11: CPF vs DPF for Estimating I_{Load_8}.

Chapter 7 - Conclusion and Future Work

In the final chapter of the thesis, Section 7.1 reviews the motivation of the research, addresses the challenges of state estimation, and summarizes the conclusive evidence found in the experiments. Section 7.2 provides the key contributions of the thesis, while Section 7.3 highlights some future work and possible directions that this research can follow.

7.1 Summary

This thesis explores the application of non-linear state estimation techniques to modern, smart electric power grids and addresses some of the key challenges faced in such applications. These challenges include: a high degree of computational complexity, high-order non-linear system dynamics, as well as the presence of highly corrupt measurements with non-Gaussian distributions. The reduced-order, distributed particle filter is proposed in order to tackle these aforementioned problems. This technique partitions the network into subsystems which are responsible for estimating the state of only the local nodes within the subsystem. This reduces the computational complexity of the estimation algorithm. The reduced-order, distributed implementation of the particle filter is well equipped to handle highly non-linear system dynamics and does not impose any restriction on the process or observation noise distributions. The main question this thesis seeks to answer is: Can the reduced-order distributed implementation of the particle filter approach the levels of accuracy achieved by its centralized counterpart?

To answer this question, the implementation of the reduced-order, distributed particle filter is tested on two types of power systems. In Chapter 5, the system and observation model for deregulated, power transmission is derived. Monte Carlo simulations are run on this system with a SNR of 30 dB, and plots are shown which compare both the centralized and distributed estimates versus the original, noisy signal. Additionally, the RMSE plots of the centralized and distributed estimate are also generated. Both plots illustrate that the reduced-order, distributed implementation provides approximately the same level of accuracy as the centralized implementation at a much lower computational cost. These experiments are repeated and confirmed in Chapter 6 for microgrids operating in the islanded mode.

7.2 Contributions

The main and original contribution of this thesis is the work done on state estimation (SE) in both modern, deregulated power transmission networks as well as in islanded microgrids. Currently, all SE techniques used for transmission networks are centralized. This thesis implements a distributed, reduced-order particle filter on a modern transmission network, which tracks the state values at a high degree of accuracy and increased amount of efficiency compared to the centralized particle filter. This technique is also much less vulnerable then the centralized estimation techniques used in practice. In the simulations for the smart power transmission networks, the usage of high frequency PMU's are employed as the sensor agents. The penetration of PMU's into the transmission network is a certainty in the future due to their highly accurate and synchronized measurement taking ability.

Microgrids are an extremely important part of the future smart grid setup and will rely on very quick, accurate, and decisive decision making since a large part of its operation will be autonomous. The role of state estimation in both nowcasting and forecasting of the microgrid state is therefore essential. The previous work done in this area, to the best of the author's knowledge, is done on microgrid system models which are linearized about an operating point [44]-[45]. This approach does not capture the true system dynamics. This thesis explores a microgrid model which preserves its non-linearity. Furthermore, the reduced-order implementation at a lower complexity is also an important contribution to this field.

A secondary contribution which is made by this thesis is the creation of a software toolbox which supports the rapid prototyping and simulation of custom microgrids. This toolbox provides a drag and drop interface where a user can build their own microgrid and simulate its behaviour. Microgrid modeling can be done in Simulink and PSCAD, however, modeling inverter based DGs in particular is a time consuming task. This toolbox uses a simplified model of the DG (which is still non-linear) to ease this process. The toolbox also supports the saving and loading of custom systems for archiving purposes. Additionally, the user can input a custom file (.csv) which provides the configuration details of the DGs, transmission lines, and loads. In this case, the user does not need to manually draw the network, which for large networks, can be a tedious process.

Work related to all three areas of this thesis (state estimation in transmission networks, toolbox for microgrid simulation, state estimation in islanded microgrids) has resulted in three conference papers [48]-[50]. This thesis is the amalgamation of the work done in these papers.

7.3 Future Work

The proposed future work is enumerated below:

1. The simulation results in this thesis have been expanded to a realistic portion of the American Midwest power grid (IEEE-14 Bus), however, larger test networks do exist (IEEE-118 and IEEE-300 bus for transmission networks, and the IEEE-69 bus for the microgrid). The expansion of these algorithms to truly large networks is needed to assess the practicality of implementing the estimation technique in the real world.

2. There exist many types of generators that can be described by a variety of state variables in addition to speed, voltage, and phase. The same goes for load modeling. A more complete realization of a real world EPG with higher order models would be useful to test out the estimation techniques. For microgrids, performing state estimation when they are in grid connected mode is an important experiment.

3. Incorporating system disturbances and outages could be an interesting experiment to assess how well the state estimator copes. An example of this is to simulate a generator fault at a particular time step and disrupt the system dynamics. Another example is to assume a local estimating subsystem has shut down and has stopped providing estimates.

4. Exploring the relationship between the number of partitions in an overall network and estimation accuracy is important. More subsystems typically lead

to a larger number of shared states which increases computational complexity. However, is there a way to determine the ideal number of subsystems based on the configuration of a network?

5. Running an experiment involving the Kalman filter, which uses a linearized state model of the power network. Quantifying the need for non-linear estimation by capturing system dynamics is extremely important since many industrial approaches today use linear estimation techniques.

6. Incorporating dynamic network topology, historical measurements, and bad data processing into the state estimator to make it more robust and resilient. In particular for microgrids, the step size of solving the differential equations are sometimes in the range of microseconds. A PMU cannot supply data this quickly, and so finding a way to leverage historical and virtual measurements for the microgrid is essential in evaluating the practicality of the proposed method.

Bibliography

[1] G. Andersson and P. Donalek, “Causes of the 2003 major grid blackouts in North America and Europe, and recommended means to improve system dynamic performance,” *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 1922-1928, 2005.

[2] M. Arulampalam, S. Maskell, N. Gordon, and T.Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, 2005.

[3] M. Bolic, P. Djuric, and H. Sangjin, “Resampling algorithms and architectures for distributed particle filters,” *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2442-2450, 2002.

[4] A. Mohammadi and A. Asif, “Distributed state estimation for large-scale nonlinear systems: A reduced order particle filter implementation,” *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 249-252, 2012.

[5] S. Xu, R. de Lemare, and H. Poor, “Dynamic topology adaptation for distributed estimation in smart grids,” *2013 IEEE Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 420-423, 2013.

[6] S. Bolognani, N. Bof, D. Michelotti, R. Muraro, and L. Schenato, “Identification of power distribution network topology via voltage correlation analysis,” *2013 IEEE Conference Decision and Control (CDC)*, pp. 1659-1664, 2013.

[7] V. Kekatos, G. Giannakis, R. Baldick, “Grid Topology Identification using Electricity Prices,” *2014 IEEE PES General Meeting — Conference & Exposition*, pp. 1-5, 2014.

[8] Y.F. Huang, S. Werner, J. Huang, N. Kashyap, and V. Gupta, “State Estimation in Electric Power Grids: Meeting new challenges presented by the requirements of the future grid,” *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 34-43, 2012.

[9] T. Klitgaard and R. Reddy, “Lowering Electricity Prices through Deregulation,” *Current Issues in Economics and Finance*, vol. 6, no. 14, pp. 1-6, 2000.

[10] J. Campillo, F. Wallin, I. Vassileva and E. Dahlquist, “Electricity demand impact from increased use of ground sourced heat pumps,” *2012 3rd IEEE PES International Conference and Exhibition*, pp. 1-7, 2012.

[11] *IEEE Guide for Design, Operation, and Integration of Distributed Resource Island Systems With Electric Power Systems*, IEEE Std. 1547.4, 2011.

[12] E. Blood, “From Static to Dynamic Electric Power Network State Estimation: The Role of Bus Component Dynamics” (2011). *Dissertations.* Paper 57.

[13] N. Pogaku, M. Prodanovic and T. Green, “Modeling, Analysis, and Testing of Autonomous Operation of an Inverter-Based Microgrid,” *IEEE Transactions on Power Electronics*, vol. 22, no.2, pp. 613-625, 2007.

[14] M. Barnes, J. Kondoh, H. Asano and J. Oyarzabal, “MicroGrids,” *IEEE International Conference on System of Systems Engineering*, pp. 1-8, 2007.

[15] R. Lasseter, “MicroGrids,” *Power Engineering Society Winter Meeting, 2002. IEEE*, pp. 305-308, 2002.

[16] A. Keyhani and M. Marwali, “Smart Power Grids 2011,” *Springer*, 2011.

[17] A. Smith and A. Gelfand, “Bayesian statistics without tears: a sampling–resampling perspective,” *The American Statistician*, vol. 46, no. 2, pp. 84-88, 1992.

[18] G. Taylor and L. Kleeman, “Visual Perception and Robotic Manipulation 3D Object Recognition, Tracking and Hand-Eye Coordination,” *Springer*, 2006.

[19] S. Julier and J. Uhlmann, “A new extension of the Kalman filter to nonlinear systems,” *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, 1997.

[20] N. Gordon, D. Salmond, and A. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proceedings for Radar and Signal Processing*, vol. 140, no. 2, pp. 107-113, 1993.

[21] D. Simon, “Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches,” *Wiley*, 2006.

[22] F. Schweppe and J. Wildes, “Power System Static-State Estimation, Part I: Exact Model,” *IEEE Transactions on Power Apparatus and Systems*, vol. 89, no. 1, pp. 120-125, 1970.

[23] A. Debs and R. Larson, “A Dynamic Estimator for Tracking the State of a Power System,” *IEEE Transactions on Power Apparatus and Systems*, vol. 89, no. 7, pp. 1670-1678, 1970.

[24] F. Wu, “Power System State Estimation: A survey,” *International Journal of Electrical Power & Energy Systems*, vol. 12, no.2, pp. 80-87, 1990.

[25] A. Leite de Silva, M. Do Coutto Filho and J. de Queiroz, “State forecasting in electric power systems,” *IEE Proceedings on Generation, Transmission, Distribution*, vol. 130, no. 5, pp. 237-244, 1983.

[26] N. Bretas, “An iterative dynamic state estimation and bad data processing,” *Electrical Power & Energy System*, vol. 11, no. 7, pp. 70-74, 1989.

[27] J. Mandal, A. Sinha and L. Roy,“Incorporating nonlinearities of measurement function in power system dynamic state estimation,” *IEE Proceedings on Generation, Transmission, Distribution*, vol. 142, no. 3, pp. 289-296, 1995.

[28] E. Ghahremani and I. Kamwa, “Dynamic State Estimation in Power System by Applying the Extended Kalman Filter With Unknown Inputs to Phasor Measurements,” *IEEE Transactions on Power Systems*, vol. 126, no. 4, pp. 2556-2566, 2011.

[29] E. Ghahremani and I. Kamwa, “Online State Estimation of a Synchronous Generator Using Unscented Kalman Filter From Phasor Measurement Units,” *IEEE Transactions on Energy Conversion*, vol. 26, no. 4, pp. 1099-1108, 2011.

[30] G. Valverde and V. Terzija, “Unscented kalman filter for power system dynamic state estimation,” *IET Generation, Transmission & Distribution*, vol. 5, no. 1, pp. 29-37, 2011.

[31] S. Wang, W. Gao and A. Meliopoulos, “An Alternative Method for Power System Dynamic State Estimation Based on Unscented Transform,” *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 942-950, 2012.

[32] Y. Li, Z. Huang, Z. Zhou, B. Lee, R. Diao and P. Du, “Application of Ensemble Kalman Filter in Power System State Tracking and Sensitivity Analysis,” *2012 IEEE PES Transmission and Distribution Conference and Exposition (T&D)*, pp. 1-8, 2012.

[33] R. Ebrahimian and R. Baldick, "State estimation distributed processing for power systems," *IEEE Transactions on Power Systems*, vol. 15, no.4, pp. 1240-1246, 2000.

[34] M. Zhao and A. Abur, "Multi area state estimation using synchronized phasor measurements," *IEEE Transactions on Power Systems*, vol. 20, no.2, pp. 611-617, 2005.

[35] W. Jiang, V. Vittal, and G. Heydt, "A Distributed State Estimation Utilizing Synchronized Phasor Measurements," *IEEE Transactions on Power Systems*, vol. 22, no.2, pp. 563-571, 2007.

[36] V. Kekatos and G. Giannakis, "Distributed Robust Power System State Estimation," *IEEE Transactions on Power Systems*, vol. 28, no.2, pp. 1617-1626, 2013.

[37] G. D'Antona, A. Monti, F Ponci, "A decentralized state estimator for nonlinear electric power systems," *2007 1st Annual IEEE Systems Conference*, pp. 247-252, 2007.

[38] G. Rigatos and P. Siano, "Distributed state estimation for condition monitoring of nonlinear electric power systems," *2011 IEEE International Symposium on Industrial Electronics (ISIE)*, pp. 1703-1708, 2011.

[39] H. Zhu and G. Giannakis, "Power System Nonlinear State Estimation Using Distributed Semidefinite Programming," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no.6, pp. 1932-1953, 2014.

[40] N. Zhou, D. Meng, Z. Huang and G. Welch, "Dynamic State Estimation of a Synchronous Machine using PMU Data: A Comparative Study," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1, 2014.

[41] O. Hlinka, F. Hlawatsch, and P. Djuric, "Distributed particle filtering in agent networks: A survey, classification, and comparison," *IEEE 30.1 Signal Processing Magazine*pp. 61-81, 2013.

[42] S. Lee and M. West, "Performance comparison of the distributed extended kalman filter and markov chain distributed particle filter (mcdpf)," *Proceedings of the 2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys 10)*, 2010.

[43] A. Mohammadi and A. Asif, "Consensus-based Particle Filter Implementations for Distributed Nonlinear Systems, Chapter 9, Nonlinear Estimation and Applications to Industrial Systems Control," *Nova Science Publishers*, 2012.

[44] G. Korres, N. Hatziargyriou and P. Katsikas, "State estimation in Multi-Microgrids," *European Transactions on Electrical Power*, vol. 21, no.2, pp. 1178-1199, 2011.

[45] Y. Wang, Y. Tian, X. Wang, Z. Chen and Y. Tan, "Kalman-Filter-Based state estimation for system information exchange in a multi-bus islanded microgrid," *7th IET International Conference on Power Electronics, Machines, and Drives (PEMD 2014)*, pp. 1-6, 2014.

[46] A. Mohammadi and A. Asif, "A Consensus/Fusion based Distributed Implementation of the Particle Filter," *2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 285-288, 2011.

[47] O. Hilinka, O. Sluciak, F. Hlawatsch, P. Djuric and M. Rupp, "Likelihood consensus: Principles and application to distributed particle filtering," *IEEE Asilomar Conference*, pp. 349-353, 2010.

[48] A. Asif, A. Mohammadi, and S. Saxena, "Reduced order distributed particle filter for electric power grids," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7609–7613, 2014.

[49] S. Saxena, H. Farag, and A. Asif, "A toolbox for the modelling and simulation of islanded microgrids," *IEEE International Conference on Smart Energy Grid Engineering*, pp. 1-17, 2014.

[50] S. Saxena, A. Asif, and H. Farag, "Non-linear, reduced order, distributed state estimation in microgrids," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[51] A. Mohammadi, A. Asif, "Consensus-based distributed unscented particle filter," *2011 IEEE Statistical Signal Processing Workshop*, pp. 237-240, 2011.

[52] S. Julier, J. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, no.3, pp. 401-422, 2004.

[53] A. Mohammadi and A. Asif, "Distributed particle filtering for large scale dynamical systems," *IEEE 13th International Multitopic Conference, INMIC 2009*, pp. 1-5, 2009.

[54] U.A. Khan and J.M.F. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. on Signal Processing*, vol. 56, no.10, pp. 4919–4935. 2008.

[55] A. Mohammadi and A.Asif, “Distributed Particle Filter Implementation with Intermittent/Irregular Consensus Convergence,” *IEEE Trans. on Sig. Proc.*, vol. 61, no. 10, pp. 2572-2587, May15, 2013.

[56] R. Karlsson, T. Schon, and F. Gustafsson, “Complexity Analysis of the Marginalized Particle Filter,” *IEEE Trans. Signal Processing*, vol. 53, no. 11, pp. 4408-4411. 2005.

[57] R. Van der Merwe, A. Doucet, N. de Freitas, and E. Wan, “The unscented particle filter,” Tech. Rep. CUED/F-INFENG/TR,380, Cambridge University, 2000.

[58] M. Dehgani, and S. Nikravesh, “State-Space Model Parameter Identification in Large-Scale Power Systems,” *IEEE Transactions on Power Systems*, vol. 23, no.3, pp. 1449-1457, 2008.

[59] M. Hong, C. Liu, and M. Gibescu, “Complete Controllability of an N-Bus Dynamic Power System Model,” *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, vol. 46, no.6, pp. 700-713, 1999.

[60] A. Chakrabarti and S. Halder, “Power System Analysis Operation and Control”, *Prentice-Hall of India*, 2006.

[61] A. Micallef, “Secondary control for reactive power sharing in droop-control islanded microgrid,” *IEEE International Symposium on Industrial Electronics*, pp. 1627-1633, 2012.

[62] H. Farag, M. Abdelaziz, and E. El-Saadany, “Voltage and reactive power impacts on successful operation of islanded microgrids,” *IEEE Transactions on Power Systems*, vol. 28, no.2, pp. 1716-1727, 2013.

[63] S. Schadenko, “Solution methods review of the stiff differential equations systems,” *Proceedings of the 9th International Scientific and Practical Conference of Students, Post-graduates and Young Scientists*, pp. 66-68, 2003.

[64] L. Shampine and M. Reichelt, “The MATLAB ODE Suite,” *SIAM Journal on Scientific Computing*, vol. 18, pp. 1-22, 1997.

[65] F. Milano, “An Open Source Power System Analysis Toolbox,” *IEEE Transactions on Power Systems*, vol. 20, no.3, pp. 1199-1206, 2005.

[66] M. Larsson, “ObjectStab an educational tool for power system stability,” *IEEE Transactions on Power Systems*, vol. 19, no.1, pp. 56-63, 2004.

[67] N. Bottrell, M. Prodanovic, T. Green, “Dynamic Stability of a Microgrid with an Active Load,” *IEEE Transactions on Power Electronics*, vol. 28, no.11, pp. 5107-5119, 2013.

Appendix A - Toolbox For Microgrid Simulation

A significant part of the thesis involves the modeling and simulation of islanded microgrids. This chapter discusses the creation of a software toolbox which allows for rapid prototyping of custom microgrid networks. Using the microgrid equations discussed in Chapter 6, a generic, graphical, and user-friendly software is created in order to model microgrid behaviour. This toolbox is significant for state estimation in microgrids since the state model produced by the toolbox can directly be used when implementing the estimation techniques.

A.1 Background

In order to test advanced algorithms involving power systems operation and control, interactive software toolboxes are used to simulate power system behaviour. Such toolboxes have been well established for conventional power networks [65]-[66]. However, there exists a lack of a realistic simulation platform used to model large, custom microgrids. Simulink and PSCAD are the most popular tools in which to model microgrids, however, creating large networks using these tools can be a complex and time consuming process. The modeling of the distributed generator (DG) is particularly difficult as all the internal components and controllers must be modeled. This toolbox uses a simplified model for the DG as a set of five differential equations in which the constant parameters are enabled to be customized by the user.

Using an intuitive user interface built in LabVIEW, a user can drag and drop a customized DG and model a microgrid with transmission lines and loads which are similarly customizable. The LabVIEW User Interface is integrated with a MATLAB ODE solver which is then used to solve the custom network modeled as a system of non-linear differential equations. The traditional approach in modeling power networks and microgrids is to linearize the non-linear differential equations which define the system [13],[67]. However, electrical power networks are highly non-linear in nature, and this is not an optimal strategy for capturing system dynamics. This toolbox preserves the non-linearity of the ODEs and allows a user to define a custom microgrid for the purposes of simulation.

Fig. A.1 illustrates the framework and features of the toolbox. The user interface (UI) is written in LabVIEW and allows the user to build a custom microgrid in two ways. For smaller networks, the user can elect to use a drag and drop technique in order to assemble the network from microgrid components that include DGs, transmission lines, and loads. For larger networks, this may prove to be a tedious task. In this case, an Excel spreadsheet (or .csv file) that includes all component details and configurations can be loaded into the toolbox. The toolbox will assess the network for any faults, and provide an illustration of the network according to the configuration provided. This allows the user to visually inspect the microgrid for any error in the configuration file. All networks can be saved to an Excel spreadsheet and maintained in a database for later use. The MATLAB ODE solver is used to solve the network created by the user, and returns the data back to the LabVIEW UI so the user will be able to plot the results.

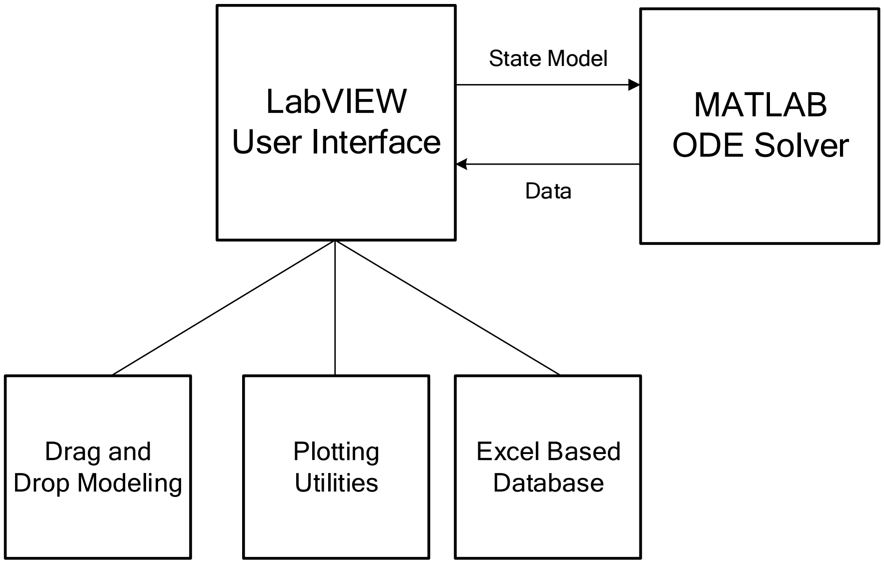


Figure A.1: Software Framework of the Toolbox.

A.1.1 Software Design of Toolbox

The objective of the toolbox is to provide an easy to use interface in which the user can solve an n node, n DG, n Line, and n Load microgrid network operating in the islanded mode. The software consists of two platforms: LabVIEW and MATLAB. The general framework is written in LabVIEW, which includes the user interface, main state machine, and main data storage objects. MATLAB is then called from within the LabVIEW framework to solve the set of differential equations generated by the custom network. LabVIEW is a graphical programming language and system design platform which is widely used in both academic and research institutions. LabVIEW is chosen for writing the general framework because it is naturally inclined towards a multithreaded style of programming. Multiple threads can be statically or dynamically created and inter-process communication is easily facilitated through built in queues and notifiers. LabVIEW also supports object oriented programming. Since the toolbox is modeled as a system of objects, and in the future has the potential for the aggregation of more network components, object oriented programming is an important consideration when it comes to the maintenance and expansion of the code in the future. Most importantly, LabVIEW offers an excellent interface to facilitate communication with MATLAB.

A.1.2 Object Model and System Architecture

Central to the idea of the toolbox is the idea of a network. A single network is custom created by the user and represents all the components that make up the network: DGs, Lines, Loads, and Nodes. An UML object diagram of the software model is shown in Fig. A.2. As can be seen in the figure, it follows that a network is *composed* of many nodes, while the nodes themselves are composed of many DGs, Lines, and/or Loads. The composition relationship between objects is represented by the black diamond, while the one-to-many relationship is represented by the black star.

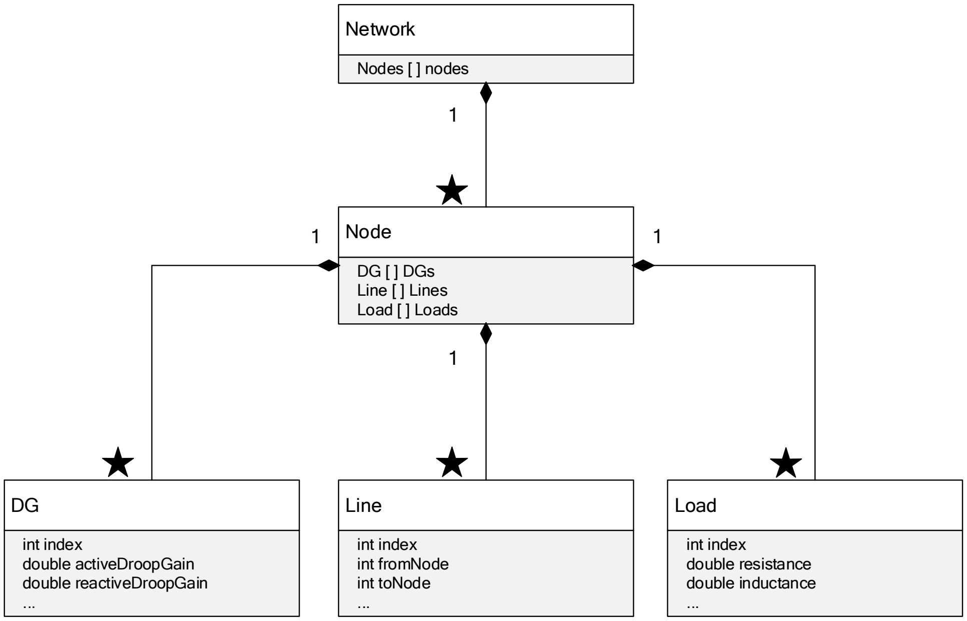


Figure A.2: Object Model of the Toolbox.

Fig. A.3 shows the system architecture of the proposed toolbox. As shown in the figure, the system architecture is divided into three separate threads: the User Interface Thread, the Main State Machine Thread, and the MATLAB ODE thread. The User Interface Thread is responsible for capturing all user events and forwarding relevant data to the main state machine. The user uses the user interface to add/modify/remove nodes, DGs, Lines, and Loads. The User Interface thread captures the specific option, services the user request, and forwards the updated data to the main state machine where the network data is stored.

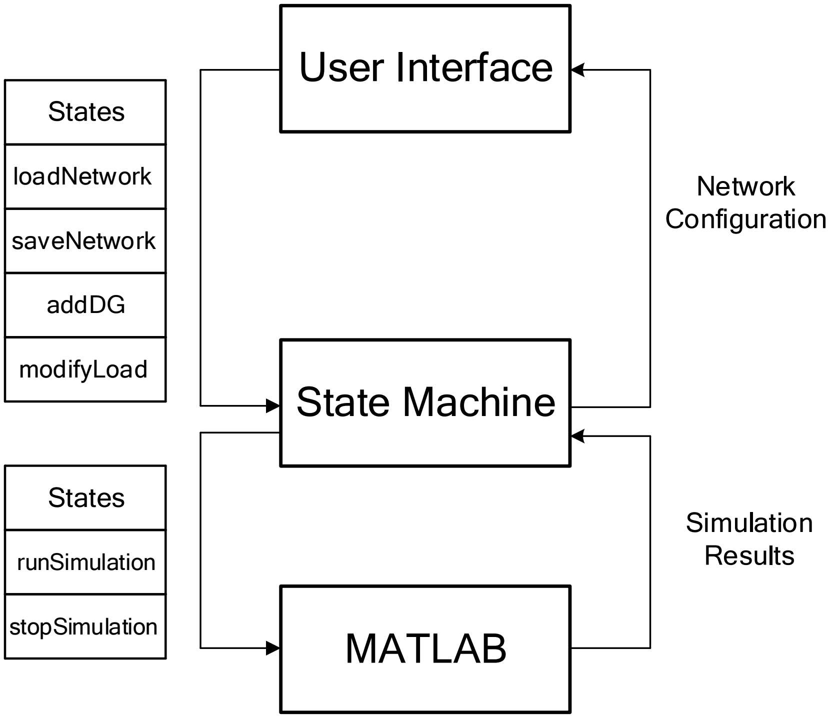
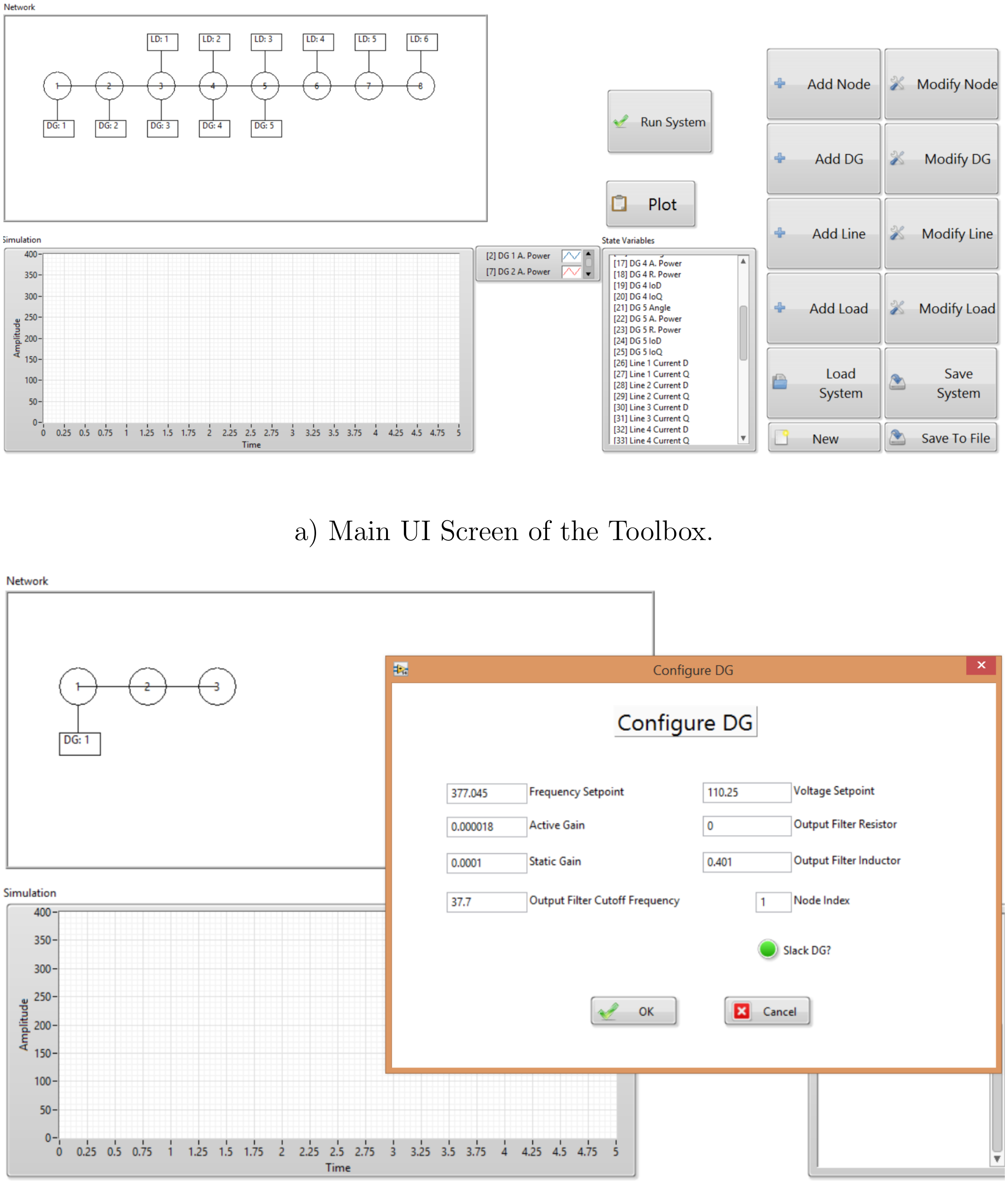


Figure A.3: System Architecture.

The Main State Machine Thread facilitates the program state and holds the latest copy of the Network data. It receives state change instructions from the User Interface Thread (state could go from idle, to running a simulation, to exiting) and also updates the network according to the users' changes. It also sends instructions and data to the MATLAB threadoop when the simulation is ready to be run and receives the data back when the simulation is complete. It is implemented as a queued state machine whose default state is simply idle. The UI thread interrupts the main state machine whenever the user engages with the program to make a request, while the MATLAB ODE thread is used to execute the time domain simulation of the network and return the results back to the Main State Machine

Inter-process communication is managed through the use of single element queues. Once instantiated, a reference (pointer to the memory location) to the queue can be called from anywhere in the program to either enqueue or dequeue an element in the queue. Two queues are used for each communication link between thread (UI Thread to State Machine, State Machine to MATLAB) to facilitate a send/receive interface. Fig. A.4 shows screenshots of the user interface. Fig. A.(4a) illustrates the main screen of the toolbox, which includes options such as: adding or modifying various microgrid components, saving/loading the system, as well as running a simulation on the loaded model. Fig. A.(4b) illustrates how a user can configure a custom DG.



[2] DG 1 A. Power
[7] DG 2 A. Power

[17] DG 4 A. Power
[18] DG 4 R. Power
[19] DG 4 IoD
[20] DG 4 IoQ
[21] DG 5 Angle
[22] DG 5 A. Power
[23] DG 5 R. Power
[24] DG 5 IoD
[25] DG 5 IoQ
[26] Line 1 Current D
[27] Line 1 Current Q
[28] Line 2 Current D
[29] Line 2 Current Q
[30] Line 3 Current D
[31] Line 3 Current Q
[32] Line 4 Current D
[33] Line 4 Current Q

a) Main UI Screen of the Toolbox.

Configure DG

377.045	Frequency Setpoint	110.25	Voltage Setpoint
0.000018	Active Gain	0	Output Filter Resistor
0.0001	Static Gain	0.401	Output Filter Inductor
37.7	Output Filter Cutoff Frequency	1	Node Index

● Slack DG?

b) Configuration of DG.

Figure A.4: Screenshots of the Toolbox.

A.2 Simulation Results

The toolbox is tested on the earlier illustrated 8 bus, 5 DG microgrid. In this experiment, the droop gains of each DG are set to the same value so that the real power output can be shared equally. Due to line impedance mismatches, the voltage at the nodes of the bus are not equal, and as such, equal reactive power sharing cannot be achieved. This is due to the fact that the bus voltages are affected by the droop gains settling at different values [61]. This does not affect equal real power sharing, however, since the frequency of the microgrid is set constant by the operator. The microgrid is also configured to be tested from a blank start, meaning that all state variables are set to zero.

As can be seen from Fig. A.5, the power generated by all the DGs converge to the same value, which indicates that the DGs are indeed sharing the responsibility of the 5 loads equally. In the second trial, the droop gains for the first two DGs are set to 25% more capacity than the second three DGs. This scenario ensures that the first two DGs, which are presumed closer to the substation, will shoulder the responsibility of the load more than the other DGs. As can be seen in Fig. A.6, the power supplied by the first 2 DGs are equal, and are significantly higher than the power supplied by the last 3 DGs.

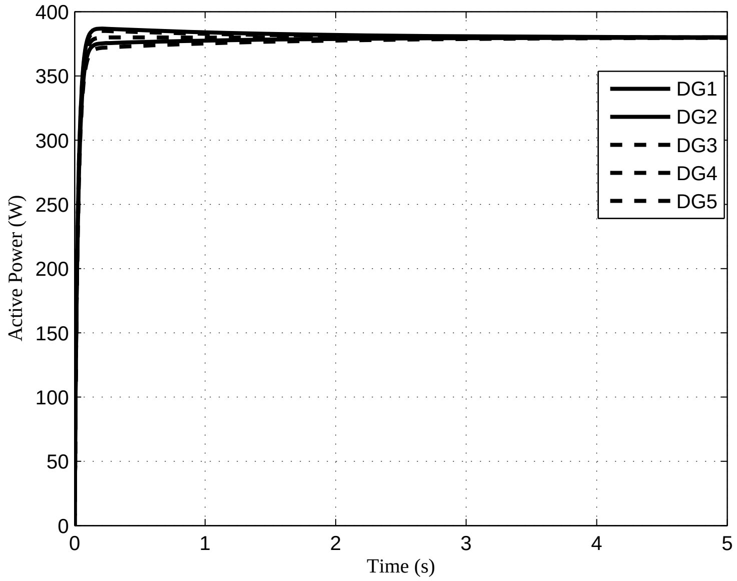


Figure A.5: Equal Power Sharing in the Network.

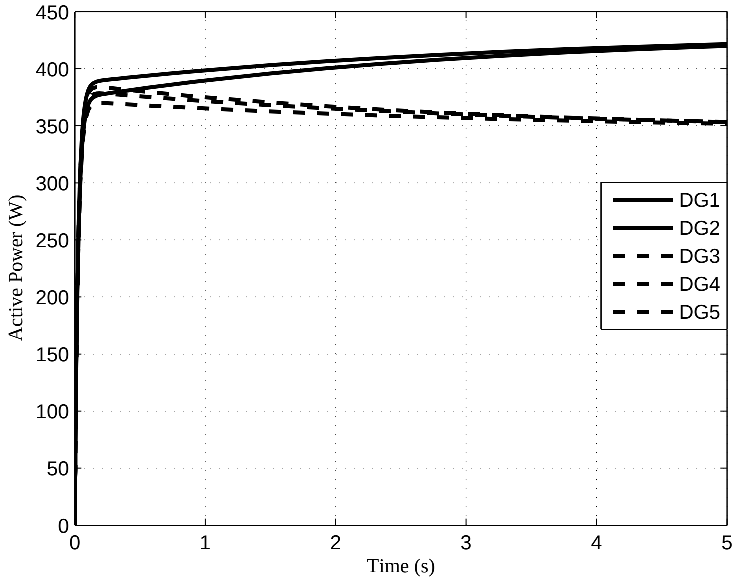


Figure A.6: An Alternate Power Sharing Strategy.

To test the accuracy and stability of the system, active and reactive power losses can calculated at each node by using the following formulas:

$$A_{Loss} = I^2 R \tag{A.1}$$

$$R_{Loss} = I^2 X \tag{A.2}$$

Where I represents the summated current at the node, R represents the resistance, while X represents the reactance.

Considering the negligible resistance of the lines and high virtual resistor value at the nodes, the power loss of the network is expected to be close to zero. To verify this, Kirchhoff's current law can be applied to any node in the network to determine the total current available at the node. The balance of current flowing into the node and current flowing out of the node should balance to zero, and as such both active and reactive power losses will also be zero.

Appendix B - Test Data for IEEE 5 and IEEE 4 Bus

B.1 IEEE 5 Bus - Line Data

From	To	R(p.u.)	X(p.u.)	Ys	Tap
1	2	0.02	0.06	0.03	1
1	3	0.08	0.24	0.025	1
2	3	0.06	0.18	0.02	1
2	4	0.06	0.18	0.02	1
2	5	0.04	0.12	0.015	1
3	4	0.01	0.03	0.01	1
4	5	0.08	0.24	0.025	1

Where From is the From Bus, To is the To Bus, R is the Resistance of the line, X is the Reactance of the line, Ys is the Ground admittance, and Tap is the Tap setting.

B.2 IEEE 5 Bus - Bus Data

Bus	Type	PGi	QGi	PLi	QLi	Vsp	Qmin	Qmax
1	1	0	0	0	0	1.06	0	0
2	2	0.4	0.3	0.2	0.1	1	-0.4	0.5
3	2	0	0	0.45	0.15	1	-0.06	0.24
4	3	0	0	0.4	0.05	1	0	0.4
5	3	0	0	0.6	0.1	1	-0.06	0.24

Where Bus is the bus number, type is the bus type (1 for slack, 2 for generator, 3 for load), PGi is the real power generated, QGi is the reactive power generated, PLi is the real power consumed, PQi is the reactive power consumed, Vsp is the initial voltage, Qmin is the minimum reactive power, and Qmax is the max reactive power.

B.3 IEEE 5 Bus - Generator Constants

Bus	Tdo	Xd	Xd'	J	D	Ef	Pm
1	0.25	1.05	0.185	1.26	2	10	10
2	0.25	1.05	0.185	1.26	2	10	10
3	0.25	1.05	0.185	1.26	2	10	10

B.4 IEEE 14 Bus - Line Data

From	To	R(p.u.)	X(p.u.)	Ys	Tap
1	2	0.001938	0.05917	0.0264	1
1	5	0.05403	0.22304	0.0246	1
2	3	0.04699	0.19797	0.0219	1
2	4	0.05811	0.17632	0.0170	1
2	5	0.05695	0.17388	0.0173	1
3	4	0.06701	0.17103	0.0064	1
4	5	0.01335	0.04211	0.0	1
4	7	0.0	0.20912	0.0	0.978
4	9	0.0	0.55618	0.0	0.969
5	6	0.0	0.25202	0.0	0.932
6	11	0.09498	0.19890	0.0	1
6	12	0.12291	0.25581	0.0	1
6	13	0.06615	0.13027	0.0	1
7	8	0.0	0.17615	0.0	1
7	9	0.0	0.11001	0.0	1
9	10	0.03181	0.08450	0.0	1
9	14	0.12711	0.27038	0.0	1

SB	EB	R(p.u.)	X(p.u.)	Ys	Tap
10	11	0.08205	0.19207	0.0	1
12	13	0.22092	0.19988	0.0	1
13	14	0.17093	0.34802	0.0	1

B.5 IEEE 14 Bus - Bus Data

Bus	Type	PGi	QGi	PLi	QLi	Vsp	Qmin	Qmax
1	1	0	0	0	0	1.06	0	0
2	2	40	42.4	21.7	12.7	1.045	-40	50
3	2	0	23.4	94.2	19.0	1.010	0	40
4	3	0	0	47.8	-3.9	1.0	0	0
5	3	0	0	7.6	1.6	1.0	0	0
6	2	0	12.2	11.2	7.5	1.070	-6	24
7	3	0	0	0.0	0.0	1.0	0	0
8	2	0	17.4	0.0	0.0	1.090	-6	24
9	3	0	0	29.5	16.6	1.0	0	0
10	3	0	0	9.0	5.8	1.0	0	0
11	3	0	0	3.5	1.8	1.0	0	0
12	3	0	0	6.1	1.6	1.0	0	0
13	3	0	0	13.5	5.8	1.0	0	0
14	3	0	0	14.9	5.0	1.0	0	0

B.6 IEEE 14 Bus - Generator Constants

Bus	Tdo	Xd	Xd'	J	D	Ef	Pm
1	0.25	1.05	0.185	1.26	2	10	10
2	0.25	1.05	0.185	1.26	2	10	10
3	0.25	1.05	0.185	1.26	2	10	10
6	0.25	1.05	0.185	1.26	2	10	10
8	0.25	1.05	0.185	1.26	2	10	10

Appendix C - Test Data for Microgrids

C.1 3 Bus Microgrid

DG Data

DG	Bus	Wo	Mp	Nq	Wc	Vo (V)	Rc(Ω)	Lc(H)
1	1	377.045	0.000018	0.0001	37.7	110.25	0	0.401
2	2	377.045	0.000018	0.0001	37.7	110.25	0	0.423

Line Data

Line	R(Ω)	L(H)	From	To
1	0	0.0226	1	3
2	0	0.0339	2	3

Load Data

Load	Node	R(Ω)	L(H)
1	3	6.552	7.88

C.2 8 Bus Microgrid

DG Data

DG	Bus	Wo	Mp	Nq	Wc	Vo (V)	Rc(Ω)	Lc(H)
1	1	377.045	0.000018	0.0001	37.7	110.25	0	0.401
2	2	377.045	0.000018	0.0001	37.7	110.25	0	0.423
3	4	377.045	0.000018	0.0001	37.7	110.25	0	0.423
4	6	377.045	0.000018	0.0001	37.7	110.25	0	0.423
5	8	377.045	0.000018	0.0001	37.7	110.25	0	0.423

Line Data

Line	R(Ω)	L(H)	From	To
1	0	0.0226	1	3
2	0	0.0339	2	3
3	0	0.0226	2	4
4	0	0.0226	4	5
5	0	0.0226	5	6
6	0	0.0226	4	7
7	0	0.0226	7	8

Load Data

Load	Node	R(Ω)	L(H)
1	3	13.104	15.76
2	4	13.104	15.76
3	5	26.208	31.52
4	6	39.312	47.28
5	7	39.312	47.28
6	8	52.416	63.04