



Integration of Substation IED Information into EMS Functionality

Final Project Report

Power Systems Engineering Research Center

*A National Science Foundation
Industry/University Cooperative Research Center
since 1996*





Power Systems Engineering Research Center

Integration of Substation IED Information into EMS Functionality

Final Project Report

Project Team

Mladen Kezunovic, Project Leader
Texas A&M University

Thomas. J. Overbye
University of Illinois at Urbana-Champaign

University of Illinois Research Team Graduate Students
Charles Davis
Joseph Tate

Texas A&M Research Team Graduate Students
Papiya Dutta
Yufan Guan
Chengzong Pang

PSERC Publication 08-25

November 2008

Information about this project

For information about this project contact:

Mladen Kezunovic, Ph.D.
Texas A&M University
Department of Electrical and Computer Engineering
College Station, TX 77843-3128
Tel: 979-845-7509
Fax: 979-845-9887
Email: kezunov@ece.tamu.edu

Power Systems Engineering Research Center

The Power Systems Engineering Research Center (PSERC) is a multi-university Center conducting research on challenges facing the electric power industry and educating the next generation of power engineers. More information about PSERC can be found at the Center's website: <http://www.pserc.org>.

For additional information, contact:

Power Systems Engineering Research Center
Arizona State University
577 Engineering Research Center
Tempe, Arizona 85287-5706
Phone: 480-965-1643
Fax: 480-965-0745

Notice Concerning Copyright Material

PSERC members are given permission to copy without fee all or part of this publication for internal use if appropriate attribution is given to this document as the source material. This report is available for downloading from the PSERC website.

**© 2008 Texas A&M University and University of Illinois at Urbana/Champaign.
All rights reserved.**

Acknowledgements

This is the final report for the Power Systems Engineering Research Center (PSERC) research project titled “Integration of Substation IED Information into EMS Functionality.” We express our appreciation for the support provided by PSERC’s industrial members and by the National Science Foundation’s Industry / University Cooperative Research Center program. Texas Engineering Experiment Station (TEES) and Concurrent Technologies Corporation provided match funding for Part I of this project.

Special thanks are due to the advisors that were actively involved in the project: Ron Allen (AEP), Joon Park (AEP), Jay Giri (AREVA), John Bee (Exelon), Ali A. Chowdhury (CAISO) and Jim Stoupis (ABB). Several Texas A&M students contributed to the project at various stages: Ozgur Gonen, Satish Natti, Maja Knezev. Special thanks are due to a colleague from Texas A&M University, Prof. Ergun Akleman who advised Mr. Ozgur Gonen on the activities related to this project. At the University of Illinois at Urbana/Champaign, two graduate students contributed extensively to the research and creation of this report: Matt Davis, and Zeb Tate.

Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of any of the sponsoring agencies or any other entity.

Executive Summary

Most electric power transmission and distribution substations are now equipped with intelligent electronic devices (IEDs) such as digital protective relays, digital fault recorders, phasor measurement units, circuit breaker monitors, power quality monitors, and remote terminal units. These devices collect operational and non-operational data, and perform their intended functions. Operational data are for monitoring and operating the power system. For example, remote terminal units are the data acquisition part of a supervisory control and data acquisition system, the main infrastructure for monitoring and operating the power system. Non-operational data from such IEDs as digital fault recorders and digital protective relays collect data only when a disturbance occurs so that the causes of disturbances can be studied. Phasor measurement units are unique devices in that they collect data continuously, but they have not yet been fully integrated with supervisory control and data acquisition system data. At this point in time, these new data are referred to as situational awareness data.

Operational data are used in energy management systems by power system operators to monitor and control the power system. This project's objective was to explore the potential benefits of integrating non-operational and situation awareness data into an energy management system. The project focused on three major tasks:

1. Identify properties of data collected by IEDs
2. Specify how an energy management system could use non-operational and situation awareness data and the benefits of doing so
3. Outline new design approaches for future energy management system functionality using fault location, topology processing, alarm processing, and operator displays as examples.

Research team members at the Texas Engineering Experiment Station (TEES) were solely responsible for the first task while researchers at TEES and at the University of Illinois at Urbana-Champaign shared responsibility for the remaining tasks. TEES staff also developed a demonstration of how new substation data may improve fault location. This supplemental work was funded by the U.S. Department of Energy through Concurrent Technologies Corporation. The findings are reported separately in PSERC Report 08-07 titled "Optimized Fault Location" published in April 2008.

Part I: Integration of Operational and Non-Operational Data for Improved Energy Management System Monitoring

Part I of the project mainly focused on:

- Understanding the properties of data from IEDs installed in substations
- Integrating the data and extracting information in an energy management system
- Designing energy management system functions that incorporate substation information

1. Understanding the properties of data from intelligent electronic devices installed in substations

The typical measurement chain in gathering and processing data includes:

- Sensing current and voltage levels by transforming power level analog signals to instrument level signals using current and voltage instrument transformers or transducers
- Filtering and digitizing instrument-level signals, and submitted for processing in IEDs
- Computing the intended information and making it available at the device's output.

Distortion is introduced in each stage. Our research focused on the characteristics of sensors, the devices that measure the data, and the performance accuracy of IEDs.

2. Integrating the data and extracting information in an energy management system

To construct a substation database, numerous challenges have to be addressed, such as:

- Interpreting the data obtained from IEDs of different vendors
- Exchanging data using standardized data format (e.g., COMTRADE)
- Adding the static system configuration data corresponding to the recorded data.

Approaches to addressing these challenges were identified in this project.

After integration, data must be processed to extract information to meet needs of various utility personnel or other automated applications. This is achieved by device-specific data analysis functions that use event data and configuration data to create analysis reports. Examples of device specific data analysis functions were created and tested in this project.

3. Designing energy management system functions that incorporate substation information

Three EMS functions that could incorporate non-operational and situational awareness data are:

- Optimized Fault Location
- Intelligent Alarm Processor
- Detection, Classification and Mitigation of Cascading Outages.

Each of these functions were designed and successfully tested.

Part II: Application of Phasor Measurement Unit Data in the Control Room

There were three research accomplishments in applying phasor measurement unit (PMU) data in the control room:

- Identifying the best filters for quickly processing raw PMU values to obtain useable measurements

- Designing an improved algorithm for using PMU angle measurements to determine power system device status changes
- Specifying the best locations for PMU placement for event detection.

1. Identifying the best filters for quickly processing raw PMU values to obtain useable measurements

Following a system disturbance, PMU measurements will not be in a steady-state condition, thus making it difficult to use them for power system state determination. Two types of digital filters, finite impulse response (FIR) and median, were analyzed for their ability to quickly determine the steady-state angle information. The sensitivity of the filter response to oscillation frequencies is much more prominent in median filtering than in FIR filtering. For systems in which the post-event oscillation frequencies are easily predicted and at specific frequencies, median filtering can provide much better edge preservation while maintaining low percent overshoot. On the other hand, for systems with broad ranges of oscillation frequencies, higher-order FIR filtering is more robust. Because the oscillations seen on a system are due to an array of factors, including the type of event and the event location, the research results suggest that FIR filtering should be used to maintain robustness.

2. An improved algorithm for using PMU angle measurements to determine power system device status changes

PMU-based angle measurements may be used to quickly determine which devices may have caused a discrete system disturbance. Disturbances considered in this part of the research project included single and multiple-line outages, and generator outages. A quick and robust algorithm was developed for detecting the device likely to have caused the system disturbance, and tested to determine the constraints that limit accurate determination of the initiating device. A key constraint is the location of the PMUs.

3. An investigation into the best locations for PMU placement for event detection

PMU location is critical in determining what events can be successfully traced to an initiating device. Two algorithms were created to solve this problem using three different objective functions: exhaustive search and simulated annealing. The objective functions were: 1) maximize the number of correctly ranked events; 2) maximize the number of correctly ranked events with no misranked events, and 3) maximize the number of correctly ranked events with no misranked events and no events with the same rank. The exhaustive search was shown to be time-consuming, but always gave an optimal solution. Simulated annealing could not guarantee an optimal solution, but was significantly faster.

Future Work

Results from both parts of the project are ready for field demonstration and deployment.

Part I

Integration of Operational and Non-Operational Data for Improved EMS Monitoring

Mladen Kezunovic
Texas A&M University

Texas A&M Research Team Graduate Students

Papiya Dutta
Yufan Guan
Chengzong Pang

Information about this project

For information about this project contact:

Mladen Kezunovic, Ph.D.
Texas A&M University
Department of Electrical and Computer Engineering
College Station, TX 77843-3128
Tel: 979-845-7509
Fax: 979-845-9887
Email: kezunov@ece.tamu.edu

Power Systems Engineering Research Center

The Power Systems Engineering Research Center (PSERC) is a multi-university Center conducting research on challenges facing the electric power industry and educating the next generation of power engineers. More information about PSERC can be found at the Center's website: <http://www.pserc.org>.

For additional information, contact:

Power Systems Engineering Research Center
Arizona State University
Department of Electrical Engineering
Ira A. Fulton School of Engineering
Phone: (480) 965-1879
Fax: (480) 965-0745

Notice Concerning Copyright Material

PSERC members are given permission to copy without fee all or part of this publication for internal use if appropriate attribution is given to this document as the source material. This report is available for downloading from the PSERC website.

Table of Contents

1. Introduction	1
1.1 Motivation and Goals.....	1
1.2 Background Information	2
1.3 Organization of Part I.....	4
2. Properties of Substation Data	5
2.1 Sources of Data	5
2.2 Accuracy of Data	6
2.2.1 Accuracy of the sensors under various dynamic conditions.....	7
2.2.2 Accuracy of internal processing of IEDs.....	12
3. Integration of Data and Information Exchange	16
3.1 Integration Concept	16
3.2 Formats of Data	17
3.3 Integration of Data	19
3.4 Extraction of Information.....	21
4. Examples of Integration of Data	24
4.1 Example Applications	24
4.2 Optimized Fault Location Algorithm [26]	24
4.2.1 Algorithms	25
4.2.2 Data requirements	26
4.2.3 Data handling.....	28
4.2.4 Implementation	30
4.2.5 Evaluation of the algorithm	31
4.3 Intelligent Alarm Processing [45]	31
4.3.1 FRPN diagnosis model	32
4.3.2 Data requirements	34
4.3.3 Implementation	34
4.3.4 Case study	36
4.3.5 Deployment strategy	38

Table of Contents (continued)

4.4 Detection, Classification and Mitigation of Cascading Outages [53] 40

4.4.1 Algorithm..... 41

4.4.2 Data requirements 42

4.4.3 Data handling..... 44

4.4.4 Implementation 47

5. Conclusion..... 49

6. References 50

List of Figures

Figure 1.1. Functional diagram for flow of substation data.....	3
Figure 2.1. Data processing in IED.....	13
Figure 2.2. Scanning and synchronous sampling of analog inputs.....	15
Figure 3.1. File repository.....	21
Figure 3.2. DFRA functional diagram	22
Figure 3.3. DPRA functional diagram	22
Figure 3.4. CBMA functional diagram	23
Figure 4.1. Flowchart of Optimized Fault Location Algorithm.....	26
Figure 4.2. Architecture of Optimized Fault Location Algorithm.....	31
Figure 4.3. Backward Reasoning concept for structuring transmission line diagnosis models....	33
Figure 4.4. Backward Reasoning concept for structuring busbar diagnosis models	33
Figure 4.5. 14-Bus power system model	35
Figure 4.6. A FRPN model for L1314 fault based on SCADA data	35
Figure 4.7. A FRPN model for L1314 fault based on SCADA and digital protective relay data.....	36
Figure 4.8. Overall Implementation Structure	39
Figure 4.9. Overall interactive monitoring and control scheme	41
Figure 4.10. Block diagram of the program for simulation fault and non-fault scenarios	47
Figure 4.11. Block diagram of the implementation containing local and system-wide analysis..	48

List of Tables

TABLE 2.1 Transformer Correction Factor for Metering Accuracy Classes (according to IEEE Standard)	8
TABLE 2.2 Ratio Error for Metering CT Accuracy Classes (according to IEC 60044-1 Standard)	9
TABLE 2.3 Ratio Error for Metering Inductive VT Accuracy Classes (according to IEC 60044-2 Standard)	10
TABLE 2.4 Limit of Errors for Relaying CT Accuracy Classes (according to IEC 60044-1 Standard)	11
TABLE 2.5 Ratio Error for Relaying Inductive VT Accuracy Classes (according to IEC 60044-2 Standard)	11
TABLE 2.6 Ratio Error for Relaying Capacitive VT Accuracy Classes (according to IEC 60044-5 Standard)	12
TABLE 2.7 Synchrophasor performance levels	15
TABLE 4.1 Data and related sources required for OFLA	27
TABLE 4.2 Data and related sources required for Intelligent Alarm Processor	34
TABLE 4.3 SCADA data	37
TABLE 4.4 Relay data.....	37
TABLE 4.5 Candidates for estimated fault sections based on SCADA data	38
TABLE 4.6 Candidates for estimated fault sections based on SCADA data and relay data	38
TABLE 4.7 Data and related source required for system level analysis	44
TABLE 4.8 Data and related source required for local level analysis.....	45

1. Introduction

1.1 Motivation and Goals

Traditionally in a substation, supervisory control and data acquisition system (SCADA) measurements are typically bus voltages, flows (amps, MW, MVAR), frequency, breaker status, transformer tap position, etc which are acquired using remote terminal units (RTUs) and sent to the energy management systems (EMS) in every two to ten seconds. With the rapid advancement of technology, intelligent electronic devices (IEDs) come into picture. These computer-based devices can record and store a huge amount of data (both operational and non-operational) with a periodicity depending upon the intended purpose of the device. For example, digital fault recorders (DFRs) only capture data during occurrence of a fault whereas phasor measurement units (PMUs) take continuous time-synchronized data with high sampling rates. Thus a great amount of data is recorded, which if used properly can become a great benefit for the EMS operators when trying to predict, monitor and post-mortem analyze power system events.

The project is intended to explore the potential benefits from the integration of the information from these additional IED data. The objectives were proposed in the proposal as:

- Better understanding how the substation IED data may be converted into information and incorporated into EMS functions
- Clear definition of the benefits that this approach may bring to the operators as they perform their everyday monitoring and control actions
- Outline of a new EMS design that incorporates information from IEDs with an indication what are the existing EMS functions that will be affected and what new functions may be created as a result

Thus the analysis presented in this report is focused on the understanding of the following issues:

- Which data to be used and when (for what application)
- Accuracy of such data (in the measurement process from the place of data capture to where it is used)
- Extraction of useful information from captured data
- Use of this information in EMS functions

To achieve these goals, we have done an extensive survey of performance of different IEDs in present day substations. The accuracy characteristics of the total measurement chain for a typical substation data is presented in this report. The issues related to extracting useful information from raw data and integrating that information to enhance the substation database are also covered. New design approaches for some of the EMS functions using this enhanced information and other non-substation data are illustrated using three new automated stand-alone applications: optimized fault location, intelligent alarm processing and, detection, classification and mitigation of cascading events.

1.2 Background Information

From late sixties SCADA systems are being used to provide information about power system states and thereby improve the performance of EMS functions. But the 2003 Blackout in North America demonstrated the SCADA data are inadequate in terms of prediction, mitigation and also post mortem analysis of cascading blackouts. On the other hand, digital protective relays, recorders and controllers collectively called IEDs (introduced in the early eighty's and used very successfully ever since) can perform a data acquisition function in addition to their intended functions and thus acquire a great amount of substation data. It has been observed in the blackout analysis report that if the data from the IEDs could have been used the analysis would have been much faster and more comprehensive [1].

Typically in a modern day integrated substation, various types of IEDs are employed for monitoring, control and protection purposes. Once the substation analog signals at high power level are measured they must be transformed to instrumentation level using current and voltage instrument transformers. The signals are then filtered and digitized, and submitted for the processing in IEDs. Finally, the measurement data is extracted and supplied as output of these devices. This is the typical measurement chain for the data acquisition.

The basic idea of integration of data is to collect all the IED data in a substation database and use it for extracting information automatically and then utilize the extracted information for several power system applications. The functional diagram for substation data flow is shown in Figure 1.1.

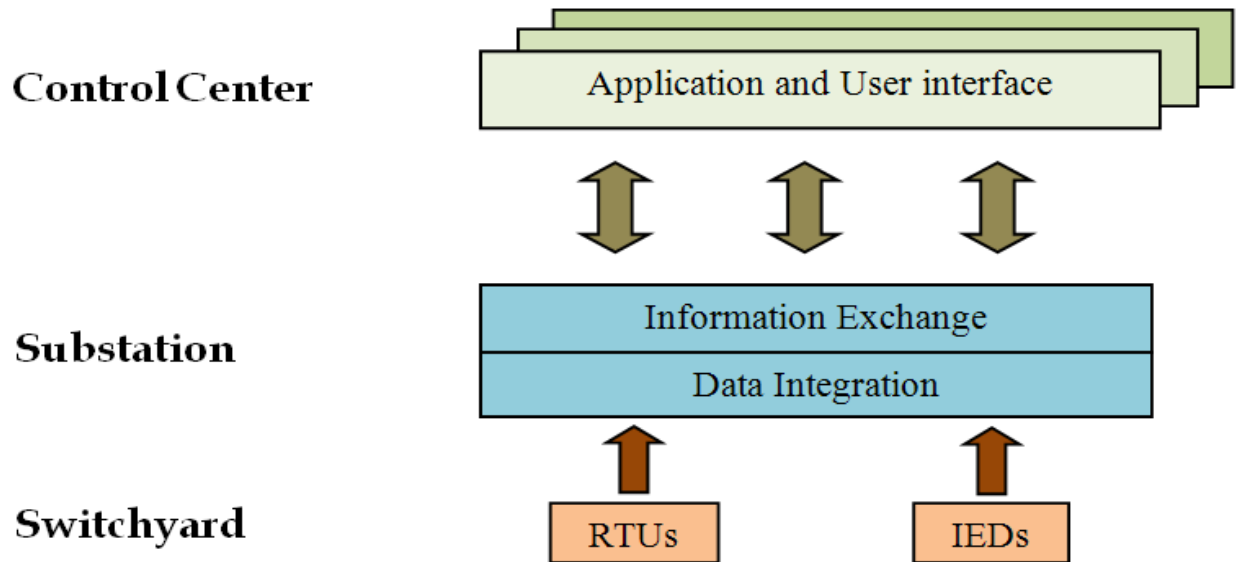


Figure 1.1. Functional diagram for flow of substation data

To import the IED data into the central repository requires means of data format conversion and communication among different IEDs. In addition to the automatically retrieved IED and RTU data, the database should contain several other data, such as:

- Static system data containing description of the system components and their connections (i.e. topology)
- Substation interpretation data that allows one to correlate the naming convention of recording devices and of the static system model
- SCADA EMS PI Historian data which may be used to tune the static system model with real time data.

Although, integrating a huge amount of data provides improved information by exploiting redundancy, the quality of data is also a major concern. Distortion in magnitude and phase angle of current and voltage signal is introduced in each stage of the measurement chain. Ideally the output waveform should be an exact replica of the input signal, but the error introduced in several data processing stages makes the output distorted. Thus the quality of data depends largely on the performance of devices used in the measurement chain.

1.3 Organization of Part I

Part I is organized into five chapters. Chapter one gives a brief introduction about the project and the background information. Chapter two discusses the properties of substation data considering performance of several devices in the entire data flow process and the means of extraction and integration of useful information. Chapter three presents the properties of data other than substation data which may be used in the EMS functions to improve their performance. Chapter four illustrates how the integration of information can be beneficial in some stand-alone EMS functions such as fault location, alarm processing and, detection, classification and mitigation of cascading outages. Chapter five gives the conclusion about the deliverables of the project and the future work.

2. Properties of Substation Data

2.1 Sources of Data

The different measurement devices (collectively termed as sensors) that are source of data in substations fall into the following categories:

- Transducer
- Relaying transformers
- Metering transformers
- Electronic (optical) transformers

Traditionally, the data processing is done by substation RTUs. Nowadays the different IEDs (microprocessor based devices intended to perform monitoring, control and protection functions) used to process data in a typical substation are [2]:

- Digital protective relay (DPR)
- Digital fault recorder (DFR)
- Phasor measurement unit (PMU)
- Power quality meter (PQM)
- Sequence of event recorder (SER)
- Programmable logic controller (PLC)
- Fault locator (FL)
- Circuit breaker monitor (CBM)
- Remote terminal unit (RTU)

The digital protective relays (DPRs) are capable of measuring and recording analog and/or status data, as well as communicating with a centralized location in addition to doing the traditional relaying tasks. These DPRs collect current and voltage signals from instrument transformers and digitize them by sampling and then converting the analog signal to digital domain. As relay should be very fast acting device that does not require very accurate measurements for relaying

purposes, to speed up the A/D conversion, lower sampling rates are generally used. Thus data obtained from DPRs are generally less accurate than from the other IEDs. Digital fault recorders (DFRs) record highly accurate waveforms related to faults. They can record huge amount of analog and status data for pre-fault, fault and post-fault conditions. The data captured by DFRs are not used in real time; they are stored as samples for further off-line processing and thus higher sampling rates can be used in these devices to obtain better accuracy in magnitude and higher frequency bandwidth. Generally these DFRs are installed in most important substations. Phasor measurement units (PMUs) compute synchronous phasors from sampled analog inputs based on Coordinated Universal Time (UTC) obtained from receivers tuned to the global positioning system (GPS) of satellites [3]. Generally PMUs are the most accurate IEDs. Power quality meters (PQMs) analyze power quality disturbances such as voltage sags, voltage swells, switching transients, impulses, notches, flickers, harmonics, etc. Sequence of event recorders (SERs) are complex recording devices implemented lately using programmable logic controllers (PLCs) capable of capturing changes in contacts (statuses) and analogous waveform through dedicated data acquisition subsystems. SERs record switching sequence of switching devices. In general the accuracy of the captured events is high due to the use of higher sampling rates. Fault locators (FL) are stand-alone devices for accurate determination of location of transmission line faults, and for this purpose they monitor current and voltage signals accurately. Circuit breaker monitors (CBM) analyze breaker operation and possible malfunctions. Remote terminal units (RTUs) are used to monitor substation quantities and communicate recorded data to SCADA database.

Thus, the IEDs are intended for specific function (sometimes for multiple functions) and depending upon power system events IEDs record analog and/or status data either in a specific location or for the entire substation.

2.2 Accuracy of Data

Distortion in magnitude and phase angle of current and voltage signal is introduced in each stage of the measurement chain. Ideally the output waveform should be an exact replica of the input signal, but the error introduced in several stages make the output distorted. This section is aimed at discussing the performance characteristics of the different stages in the measurement chain.

2.2.1 Accuracy of the sensors under various dynamic conditions

Sensors (instrument transformers and transducers) measure current and voltage (for either metering or protection purposes) from high voltage electric circuit (primary side) and supply the RTUs or other IEDs (secondary side) with these quantities proportional to those of the high power circuit but in substantially reduced value and also provide galvanic isolation for these devices from high voltage circuitry. Generally, conventional instrument transformers (current transformers-CTs, voltage transformers-VTs and capacitive coupled voltage transformers-CCVTs) are relatively inaccurate comparing to the microprocessor based IEDs. Thus the accuracy of data is highly dependent on the accuracy of instrument transformers, noise on control cables and impact of burdens (the external load connected to secondary transformer terminals, including IEDs for this discussion). With the increase of burdens, the accuracy class of instrument transformers decreases. However current and voltage transducers and optical instrument transformers are much more accurate devices compared to the traditional instrument transformers.

This section will focus on determination of inaccuracies of sensors under various conditions (steady state, transients, switching etc).

IEEE Standard C57.13 on requirements for instrument transformers [4] and IEC standard 60044 [5] defines instrument transformer types, parameters for insulation class and accuracy class, and standard burden types. With this knowledge it is possible to predict the inaccuracies in instrument transformers as well as understand how they reflect on the performance of IEDs. These standards emphasize mainly only the steady state performance of instrument transformers for fundamental frequency (50 or 60Hz) signals. Since power system is prone to many types of transients, the instrument transformers no longer operate on linear characteristics under these circumstances.

The instrument transformers can operate for either metering purposes (up to full load of the system) or protection purposes (for faulted values). Accordingly the preferred accuracy is also different.

For metering, the transformer correction factor (TCF, combined ratio and phase angle error) for the accuracy classes as defined in IEEE Standard C57.13 and C57.13.6 ([6] for high accuracy transformers) is shown in Table 2.1 (for 0.6-1.0 lagging power factor of metered load). For

current transformers, the error limits are mentioned for only 100% and 10% of the rated current (not for the entire range) for accuracy classes 0.3-2.4 and 100% and 5% of the rated current (again not for the entire range) for accuracy classes 0.15-0.15S. But when the current becomes much lower (much less than 5% of rated current), the CT characteristics enters the “ankle” region and due to non-linearity the errors become exponential. On the other hand, as the current (and flux density) increases above the rated current, the errors remain linear until the characteristics enter into saturation zone, and after that there will be an avalanche increase of ratio error and distortion in secondary current waveform. For voltage transformers, the accuracy class is defined within 90% to 110% of rated voltage. Saturation of VT occurs at not much above 110% of rated voltage as the operating flux densities are much higher than that of CTs.

The burdens used for metering purposes are 0.1, 0.2, 0.5, 0.9, 1.8 ohms which corresponds to 2.5, 5, 12.5, 22.5, 45 VA respectively (for CT accuracy classes 0.3-2.4) for 5A secondary current. The burdens used for metering purposes with CT accuracy classes 0.15 and 0.15S (S stands for special attention) are 0.2 and 0.04 ohms which correspond to 5 and 1 VA respectively for 5A secondary current.

TABLE 2.1

Transformer Correction Factor for Metering Accuracy Classes (according to IEEE Standard)

Accuracy Class	Transformer Correction Factor Range			
	CT			VT
	% rated current			
	100	10	5	
0.15S	0.9985-1.0015		0.9985-1.0015	N.A.
0.15	0.9985-1.0015		0.997-1.003	0.9985-1.0015
0.3	0.997-1.003	0.994-1.006		0.997-1.003
0.6	0.994-1.006	0.988-1.012		0.994-1.006
1.2	0.988-1.012	0.976-1.024		0.988-1.012
2.4	0.976-1.024	0.952-1.048		0.976-1.024

As per IEC standard for CTs (IEC-60044-1) the ratio error (current) for the accuracy classes for metering CT defined in that standard are shown in Table 2.2 (for 0.8 lagging power factor of metered load). The burden should be 25%-100% of rated burden for accuracy classes 0.1-1.0

(including 0.2S and 0.5S) and 50%-100% of rated burden for accuracy classes 3.0-5.0. The standard burdens used for metering purposes are 2.5, 5, 10, 15, 30, 40 VA respectively.

TABLE 2.2

Ratio Error for Metering CT Accuracy Classes (according to IEC 60044-1 Standard)

Accuracy Class	Ratio Error ($\pm$ %)					
	% rated current					
	120	100	50	20	5	1
0.1	0.1	0.1		0.2	0.4	
0.2S	0.2	0.2		0.2	0.35	0.75
0.2	0.2	0.2		0.35	0.75	
0.5S	0.5	0.5		0.5	0.75	1.5
0.5	0.5	0.5		0.75	1.5	
1.0	1.0	1.0		1.5	3.0	
3.0	3.0		3.0			
5.0	5.0		5.0			

As per IEC standard for inductive VTs (IEC-60044-2) the ratio error (voltage) for the accuracy classes for metering inductive VT defined in that standard are shown in Table 2.3 (for 0.8 lagging power factor of metered load). For voltage transformers, the accuracy class is defined within 80% to 120% of rated voltage. The burden should be 25%-100% of rated burden. The standard burdens used for metering purposes are 10, 15, 25, 30, 50, 75, 100, 150, 200, 300, 400, 500 VA respectively. For capacitive VTs (IEC- 60044-5), the accuracy classes are same as accuracy classes 0.2-3.0 of inductive VTs.

TABLE 2.3

Ratio Error for Metering Inductive VT Accuracy Classes (according to IEC 60044-2 Standard)

Accuracy Class	Ratio Error ($\pm$ %)
0.1	0.1
0.2	0.2
0.5	0.5
1.0	1.0
3.0	3.0

As per IEEE standard, for relaying purposes, TCF should not exceed 10% at 1-20 times rated secondary current at the standard burden or any lower burden.

Relaying CT is classified into two classes:

- Class C or K: Low leakage reactance type i.e. the leakage flux does not have considerable effect on the ratios. K types should also have a knee point voltage at least 70% of secondary voltage rating.
- Class T: High leakage reactance type i.e. the leakage flux has considerable effect on the ratios.

Secondary terminal voltage rating of relaying CT is the voltage (100, 200, 400 or 800 Volts) the transformer will deliver across a standard burden at 20 times rated secondary current without exceeding 10% ratio error. The available burdens are 0.1, 0.2, 1, 2, 4, 8 ohms which correspond to secondary terminal voltage of 10, 20, 50, 100, 200, 400V respectively.

The standard burdens for relaying VTs are W, X, M, Y, Z, ZZ which corresponds to VA of 12.5, 25, 35, 75, 200, 400 respectively.

As per IEC standard for current transformers (IEC-60044-1) the limit of errors for the accuracy classes for relaying CT (5P and 10P: P stands for protection) defined in that standard are shown in Table 2.4. The standard burdens are 2.5, 5, 7.5, 10, 15 and 30VA.

TABLE 2.4

Limit of Errors for Relaying CT Accuracy Classes (according to IEC 60044-1 Standard)

Accuracy Class	Current error at rated primary current (%)	Composite error at rated accuracy limit primary current (%)
5P	± 1	5
10P	± 3	10

Though IEEE standard does not define accuracy classes for relaying VT, there are two accuracy classes for relaying inductive VT in IEC standard 60044-2. The ratio error (voltage) for the accuracy classes for relaying VT defined in that standard are shown in Table 2.5 (for 0.8 lagging power factor of metered load). The accuracy class is defined within 5% of rated voltage to rated voltage multiplied by RVF, where rated voltage factor (RVF) is the multiplying factor to be applied to the primary accuracy rating voltage to determine the maximum voltage at which a transformer complies with the relevant thermal requirements for a specified time, and at which a relaying inductive VT complies with the relevant protective accuracy classes. The burden should be 25%-100% of rated burden.

TABLE 2.5

Ratio Error for Relaying Inductive VT Accuracy Classes (according to IEC 60044-2 Standard)

Accuracy Class	Ratio Error ($\pm$ %)
3P	3.0
6P	6.0

For relaying capacitive VTs (IEC- 60044-5), the ratio error (voltage) for the accuracy classes for relaying VT defined are shown in Table 2.6 at 2 % and 5 % rated voltage and rated voltage multiplied by the RVF (1,2, 1,5 or 1,9).

TABLE 2.6

Ratio Error for Relaying Capacitive VT Accuracy Classes (according to IEC 60044-5 Standard)

Accuracy Class	Ratio Error ($\pm$ %)			
	% rated voltage			
	2	5	100	RVF*100
3P	6.0	3.0	3.0	3.0
6P	12.0	6.0	6.0	6.0

The DC component of the fault current may force the CTs to enter into saturation by significantly increasing the flux. However, this increase in flux depends on time constant of the circuit and thus most of the high speed relay operates before CT enters into saturation zone.

Magneto-Optic current transformers (MOCT) and Electro-Optic Voltage Transducer (EOVT) that typically operate on well known optic principles (Faraday's effect and Pockel's effect respectively) are much more accurate than traditional electromagnetic instrument transformers as they can operate under linear operating region over wide dynamic range. These transformers are typically designed to meet IEC 0.2S accuracy class or IEEE 0.15S accuracy class.

Current and voltage transducers have low voltage outputs to be directly used with microprocessor based metering and relaying devices. These devices are much more accurate as there is no problem associated with saturation of core and can be simultaneously used for both metering and relaying purposes.

2.2.2 Accuracy of internal processing of IEDs

Usually the microprocessor based IEDs have very good accuracy. While the basic data processing stages for IEDs are similar (as shown in Figure 2.1), the recording performance may differ. In all of the IEDs, the basic data processing steps are:

- Processing of the analog input
 - o Galvanic isolation using auxiliary transformer
 - o Low pass (anti-aliasing) filtering of the analog input waveforms to eliminate the high frequency components
- Sampling of the analog input waveforms

- Analog to Digital (A/D) conversion
- Processing of the digital signal samples

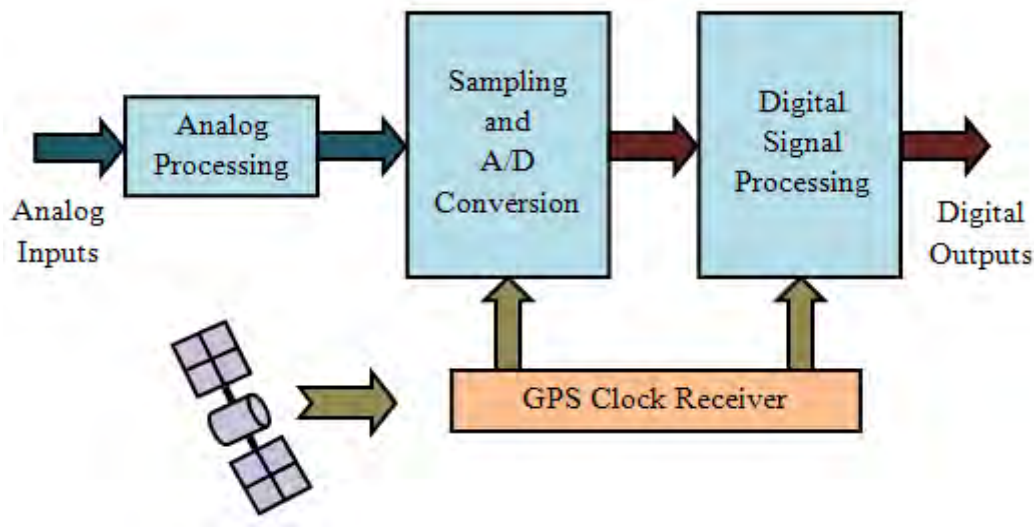


Figure 2.1. Data processing in IED

The overall accuracy of the IED analog signal input channel depends on several factors [7]:

- Impact of auxiliary transformers
- Mode of sampling
- Anti-aliasing filtering
- Sampling rate
- Resolution of A/D conversion

Some of the IEDs (DPR, DFR, FL etc) operate on synchronous sampling and some (RTU) operate on scanning of samples. As shown in Figure 2.2, in scanning, one analog input channel is sampled at a time and then converted to digital signal, whereas in synchronous sampling all the input channels are sampled at the same time and then they are converted to digital signal (there may be only one ADC serving all channels or each of the analog input channels has combined sample-and-hold circuit and ADC). Retrieving the basic properties of the signal as well as the actual phase difference between analog signals is easy in synchronous sampling as all signals are sampled at the same time. In case of synchronized sampling, the clock signal may be provided

locally or from a receiver for the Global Positioning System (GPS) of satellites [3]. Most of the IEDs of modern day are using GPS for sampling and time stamping.

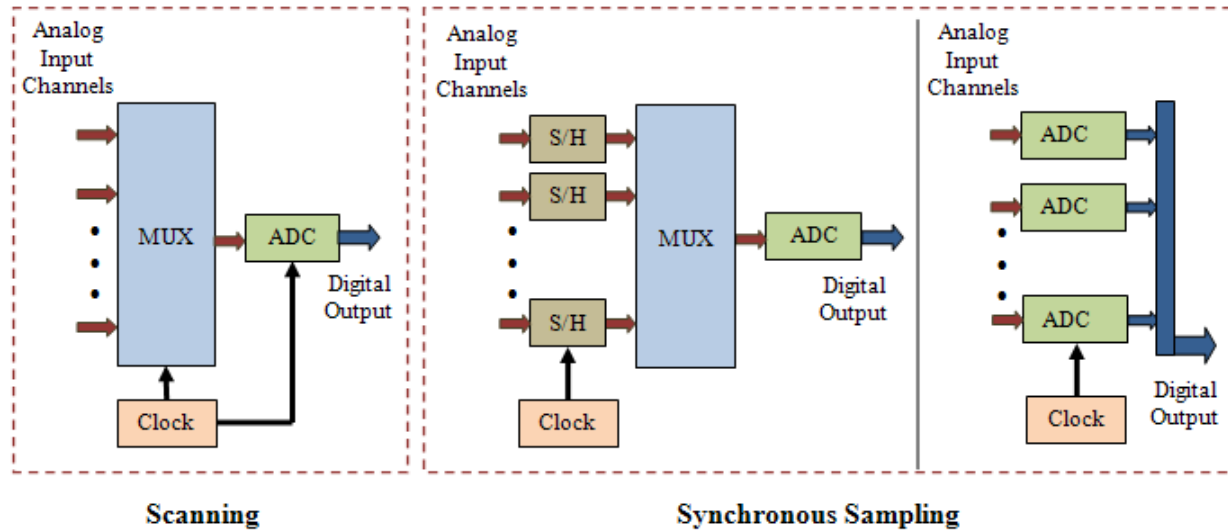


Figure 2.2. Scanning and synchronous sampling of analog inputs

The replication of signal largely depends on the “horizontal” resolution (sampling rate) and “vertical” resolution (number of bits for ADC). With higher sampling rate, the better signal representation is achieved. With more bits, better signal accuracy is achieved. Sampling rate also affects the choice of anti-aliasing filters. DPR generally is a device with low sampling rate (16 samples/cycle i.e. 960Hz for 60Hz system). Some of the DFRs of modern day achieved much higher sampling rates of 48 kHz. Again the higher value of “vertical” ADC resolution is desirable. Most of the IEDs use 16 bit ADC resolution. The selection of the resolution is driven by the ADC cost and dynamic behavior of the measured signal.

While most of the other IEDs allow access to samples of signal, phasor measurement unit (PMU) provides measurements of synchronous phasors of analog inputs. The GPS synchronized PMUs are the most accurate among all IEDs. Such GPS synchronized devices provide phasor data termed as synchrophasor. The IEEE C37.118 [8] (extension of IEEE Standard 1344) is used for accuracy definition of synchrophasors:

- Magnitude accuracy of 0.1% or better
- Time accuracy better than 1 μ s or phase angle accuracy of 0.02° at 60 Hz.

The North American Synchro Phasor Initiative (NASPI) is looking at dynamic accuracy beyond what is defined in C37.118 [9].

The total vector error for synchrophasors defined in this standard should be less than 1%. [8]

$$\text{Where, Total Vector Error (TVE)} = \frac{|X_n - X|}{|X|}$$

X = theoretical value and X_n = measured value

Two performance levels are also defined in the synchrophasor standard as shown in Table 2.7.

TABLE 2.7
Synchrophasor performance levels

Quantity	Performance Levels	
	Level 0	Level 1
Signal frequency	± 0.5 Hz	± 5 Hz
Signal magnitude	80%-120%	10%-120%
Phase angle	$\pm \pi$ radians	$\pm \pi$ radians
Harmonic distortion	1%	10%
Out-of-band frequency	1%	10%

3. Integration of Data and Information Exchange

3.1 Integration Concept

The basic approach is to collect all the IED data in a substation database and use it further for extracting information automatically. This extracted information then may be used for several power system applications [10-11]. To import the IED data into the central repository requires means of data format conversion and communication among different IEDs. In addition to the automatically retrieved RTU and other IED data, the database should contain several other data, such as:

- Static system data containing description of the system components and their connections (i.e. topology)
- Substation interpretation data to correlate the naming convention between the recording devices and static system model
- SCADA EMS PI Historian data to tune the static system model with real time data.

By integrating data from all the sources, quality of data is improved due to the redundancy of the great amount of data collected.

Though using IED data improves information, some of the problems related to user experiences with existing IED systems are:

- IED data records cannot be efficiently analyzed manually due to an overwhelming number of records captured in a moderately sized system.
- Use of different vendor specific programs for extracting data from IEDs increases personnel training costs due to distinctively different features, as well as the look and feel of different packages.
- Slow response (for example, manual analysis of IED data takes time) is an impediment if several records supplied by different IEDs for the same event must be uploaded and analyzed. Slow response can also be a result of the communication issues or the fact that different types of IEDs are used, or that the used devices are coming from different vendors.

- Lack of ability to integrate data coming from different IED types and models is evident when one attempts to integrate the different IED systems and services. Sometimes different IED subsystems require different communication channels and interfaces, different protocols, and almost on a regular bases use different data formats.
- Highly-skilled people devote a lot of time to routine tasks because most of the records may just confirm the proper operation of the equipment being monitored.
- Non-selectivity (for example, IED data records are not event prioritized) is an issue if the operator must sort out the records for the analysis purposes. This generally depends on the type of the IED and its functions, but when dealing with multiple devices it is a common problem.
- Inefficient data archival and retrieval due to rather primitive means of time-stamping, storing and retrieving the captured data. Some of the devices are being installed for their main functions and these problems occur when one wants to use them as recording devices. It is common that providing communication resources to connect these IEDs to store or retrieve the data is not adequately planned.

This chapter discusses the integration of different formats of data into substation database, extraction of information from the data in different formats collected in substation database and exchange of information within several utility groups such as operators, protection engineers, maintenance personnel, and asset managers. A general background of information extraction and exchange is discussed in this section briefly, which will be further illustrated using specific power system application examples in the next chapter.

3.2 Formats of Data

Generally a typical substation is equipped with several IEDs of different type and from different vendors which may supply data in different format. Typically IED data records are uploaded to the substation database from IED memory using vendor specific software. Getting required information from huge amount of redundant data obtained from different IEDs (having different data format) is an issue that requires particular attention.

To integrate data among different IEDs and IEDs from different vendors, standardized data format is necessary. After the introduction of COMTRADE data format [12], most of the

vendors are accepting this standardized data format, while some others are still keeping their own native data formats, developing new ones and just providing additional utility programs or commands for exporting data in COMTRADE. Unfortunately, this export-to-COMTRADE feature, in most cases, is not configurable for automated operation. Additional issue is that the COMTRADE format specification allows freedom, to some extent, on how to provide information inside the files. This leads to situations where different software packages supporting COMTRADE file format cannot exchange data among themselves due to the lack of standardized descriptions of the files and signals inside. In addition to the original COMTRADE standard specification [12], there are the latest IEEE revision [13] and IEC version [14]. Having three versions currently being used increases a possibility not to be able to exchange IED data among different types of software packages due to inconsistencies between different versions. On the other hand, the power quality meter data representation is standardized using Power Quality Data Interchange Format (PQDIF) [15].

One step further was the introduction of the standardized IEEE file naming convention for the time sequence data [16]. The proposed convention defines coded schema for naming the data files captured at different times. Such file names can enable easier handling of large volume of files as well as unique file identification since the file name should contain unique information about the event: date, time, station, company, duration, location etc. Benefits of using this standardized file naming schema should encourage IED and related software vendors to provide the standard support, which is not a common feature today.

There is still a lack of a standardized approach to defining universal data formats for IEDs and a convention for providing information on parameters that describe system objects (signals and associated equipment) monitored by IEDs in substations.

Having a standardized communication protocol [17] facilitates easier integration of IEDs into systems and enables interchangeability of IEDs, but does not solve the problem of inconsistent and/or incomplete data formats. Future IED systems may utilize standards like the one recently proposed by IEC [18] enhanced with further necessary details, but unfortunately, at present, we may still be far from that possibility. Currently, a common approach is to use software provided by vendors that needs to be configured to automatically collect IED data files and make them accessible through corporate network.

Substation data integration solution should provide the framework based on the following requirements [19-20]:

- Communication and automated data retrieval. The solution should use either vendor's software or custom built communication modules to provide for automated retrieving of newly recorded event files.
- Unifying event data file format by converting to widely accepted standard format. All IED event data files should be automatically converted into same file format (for example IEEE 1999 COMTRADE [13]) and all the files should utilize file naming convention.
- Making sure that content of the files is correct and all the fields are utilized properly (for example: substation information, time stamps, channel assignment, phase information, scaling, etc.).
- All the power system components descriptions should be available and easily mapped to the event data collected from substation IEDs.

3.3 Integration of Data

Most of the IEDs were designed and made with a very specific, often limited, data collection function in mind. With the technological progress, IEDs evolved into more and more sophisticated devices with new capabilities. The crucial improvements include: more memory, better communication interfaces, and higher quality of data recording. These improvements enable adding new functionality, primarily related to automated processing and analysis of IED-recorded data.

For substation data integration, each IED type requires communication and data format conversion modules in order to import data to the IED data file repository.

Automated collection of the IED data can be done utilizing IED vendor legacy software packages or by direct access to IED's communication resources. The first approach is less involved and it is recommended since the standard communication protocols have not been widely accepted and/or implemented yet. All the IED-recorded data is meant to be converted to an existing standard data format such as COMTRADE [12-13].

The file repository in the database should utilize standardized file naming convention [16]. Some needed modifications to the file naming recommendation include: 4-digit year information, use

of additional time stamp that corresponds to time when the event files are downloaded, time stamp features microseconds resolution (in order to achieve equal length of the file names).

It is most likely that the actual file repository integration will require combination of vendor-based and custom developed software modules in order to make sure that the records comply with the selected data format and naming standards.

Besides the IED data, the database has to contain system configuration data, which describe:

- System components and their relationship (i.e. lines, buses, circuit breakers, switches, relays, CTs, VTs, etc.)
- IED channel assignments and mapping/calibration to specific system components (line/bus voltages, line currents, status signals).

The system configuration data enables automated IED data conversion into standard formats and integration into the database thus making the data available for new functions (software applications).

The result of substation data integration is a database with file repository (Figure3.1) [19]. The file repository should contain:

- IED data in a standard format such as COMTRADE file format (IEEE Std. C37.111-1999) with file names conforming to the file naming convention guidelines.
- Configuration descriptions in a standard format such as Substation Configuration Language (SCL) [16]. The configuration should describe power system components (lines, buses, relays, breakers, transformers, etc.) and how are the components being monitored (IED channel assignments, scaling, etc.).
- Additionally, as the data is aimed for automated analysis, the repository should also contain analysis reports. In general, reports can be produced by different analysis applications that utilize integrated data.

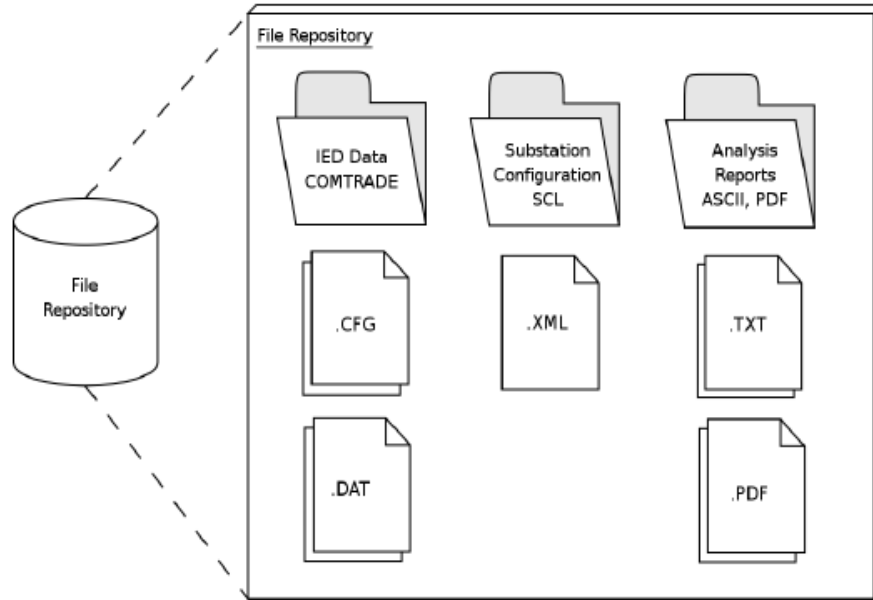


Figure 3.1. File repository

3.4 Extraction of Information

After integration of data from different IEDs in the substation database, the data should be further processed to extract information to meet needs of various utility personnel. This can be achieved by device specific data analysis functions (utilizing event data and configuration data to create analysis reports) [19-20]:

DFRA (Digital Fault Recorder Analysis) – is an application which provides automated analysis of fault records captured by DFR and disseminates event records and analysis reports [21]. The analysis looks at all the monitored circuits and identifies the one with the most significant disturbance. For that selected circuit, DFRA performs signal processing to identify pre and post-fault analogue values, statuses of the digital channels corresponding to relay trip, breaker auxiliary, communication signals, etc. The expert system determines fault type, faulted phases, checks and evaluates system protection performance. At the end, the analysis program calculates the fault location [22].

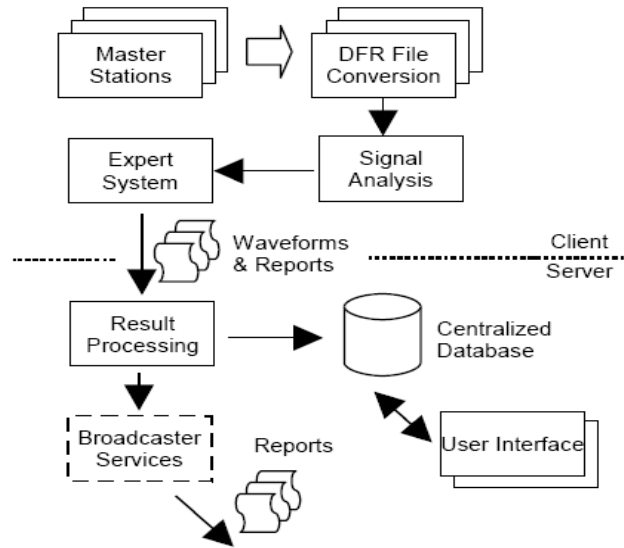


Figure 3.2. DFRA functional diagram

DPRA (Digital Protective Relay Analysis) - is an expert system based analysis application which automates validation and diagnosis of relay operation. Validation and diagnosis of relay operation is based on comparison of expected and actual relay behavior in terms of the status and timing of logic operands. If actual status and timing of an operand is consistent and as expected, the correctness of the status and timing of that operand is validated. If not, a failure or misoperation is identified and diagnosis is initiated to trace the reasons by the logic of cause-effect chain [23].

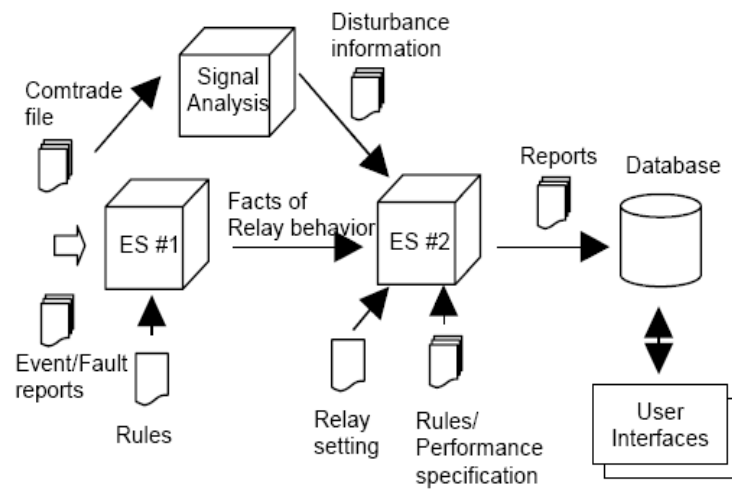


Figure 3.3. DPRA functional diagram

CBMA (Circuit Breaker Monitor Analysis) – is an application based on analysis of records of waveforms taken from the circuit breaker control circuit using a Circuit Breaker Monitor (CBM) device. It enables protection engineers, maintenance crews and operators to quickly and consistently evaluate circuit breaker performance identify deficiencies and trace possible reasons for malfunctioning. It can automatically analyze switching operations of large number of circuit breakers under complex switching conditions [24].

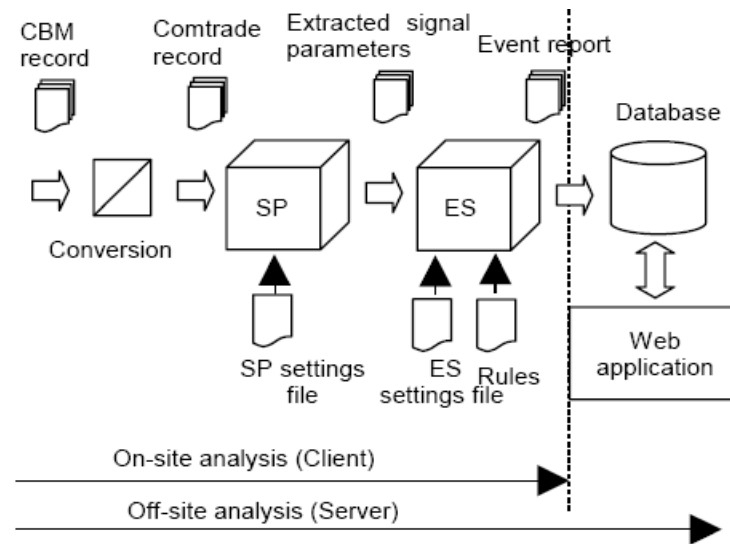


Figure 3.4. CBMA functional diagram

Power quality meter data analysis (PQMA) – is a powerful application which can be used for power quality analysis and simulation. It analyzes power quality events by converting, retrieving and managing PQM files. It can replay the recorded events or simulated signals online and implements events detection and classification [25].

The discussed approach to substation data integration enables multiple uses of the data. If all the event data is stored using same file format and file naming convention it is very easy to locate event files of interest, search through the database, and use the same viewing and analysis tools. The integrated solution needs to be fully open and allow for expansion with different viewing, processing, and analysis functions.

4. Examples of Integration of Data

4.1 Example Applications

This chapter focuses on how the use of data available from IEDs as well as the other data (SCADA data collected using RTUs, satellite data, and static system data) enhances some power system functions. The processing of data from the collection point to the applications will be discussed for three examples of power system applications

- Optimized Fault Location Algorithm
- Intelligent Alarm Processing
- Detection, Classification and Mitigation of Cascading Outages

The three examples illustrate which data can be used for those specific applications, how the useful information can be extracted and how such information can be integrated and used in the applications. Detailed description of the Optimized Fault Location application is contained in a separate PSERC report published recently [26]. This report is a result of co-funding of our project provided by Concurrent Technologies Corporation.

4.2 Optimized Fault Location Algorithm [26]

Transmission lines occasionally suffer from faults which are generally caused by several random and unpredictable reasons. Protective device (relay) senses these faults and isolates the faulty line as soon as possible. Distance relays used for transmission line protection contain algorithms that give some idea about the fault location, but they may over-reach or under-reach due to several reasons, such as prefault loading, fault resistance etc. To restore service, an accurate location of the fault should be known to help the maintenance crew find and repair the faulted line as soon as possible.

When fault appears different IEDs such as DPRs and DFRs sparsely located in substations will be automatically triggered by the change in monitored signals and will record corresponding current, voltage and status signals. These recorded quantities can be used along with the data collected by SCADA to predict the location of transmission line fault accurately.

Transmission line fault location approaches can be broadly classified into two categories [27]:

- Phasor based using fundamental frequency component of the signal and lumped parameter model of the line.
- Time-domain based using transient components of the signal and distributed parameter model of the line.

The phasor based methods can be subdivided into another two broad classes depending upon the availability of recorded data: single-end methods where data from only one terminal of the transmission line is available and double-end methods where data from both (or multiple) ends of the transmission line can be used.

Apart from these methods and their variants, artificial intelligence based methods are being widely used to locate transmission line faults [28-31].

4.2.1 Algorithms

Typical power system contains several thousands of transmission lines. Installation of recording devices at each transmission line is very expensive and it is not used in practice. Installing DFRs in critical substations is a common practice. Although protective relays exist on every transmission lines, most of them may still be electromechanical and they do not have capability to record measurements. As a result, in some cases it may happen that there are no recordings at all available line ends close to a fault. Thus depending on the availability of data, different fault location algorithms should be used to get the most accurate result.

Depending on the nature and location of the measurements, the following FL algorithms are used:

- Two or multiple-ended time-domain synchronized sampling [32]
- Two-ended phasor-based asynchronized sampling [33]
- Single-ended phasor-based [34]
- System-wide sparse measurement based [35]

The flowchart of the optimized fault location algorithm (OFLA) [36] is shown in Figure 4.1.

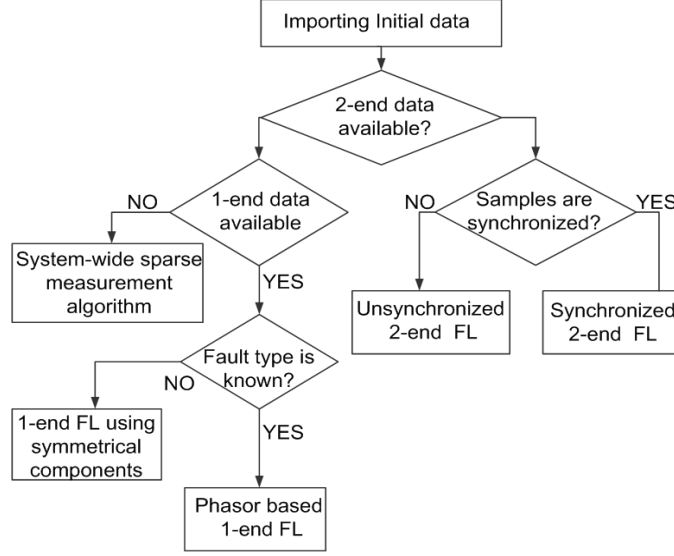


Figure 4.1. Flowchart of Optimized Fault Location Algorithm

4.2.2 Data requirements

The required data for this application can be broadly classified into

- System level data : these include power system model data and data reflecting real time changes in power system
- Field data : these include event data triggered by different IEDs after occurrence of any abnormality

The detailed description of system and field data needed for OFLA application is listed in Table 4.1.

Several commercial packages are used to implement this solution. The static power system is modeled using PSS/E [37]. To tune the power grid with pre-fault data, SCADA PI-Historian data is used [38]. The visualization of OFLA is implemented using the Power World Retriever software [39], which can retrieve information from central repository.

The field recorded data (DFR data) should follow the COMTRADE format. Using this format, the measured data, configuration, data and interpretation files should be described. The DFR data contains analog and digital sample values for all input channels for a specific substation. The configuration data contains information for interpreting the allocation of measured data to the equipment in substation. The interpretation file for each substation represents the correlation

between the nomenclature used in DFR files and those used in PSS/E file. These interpretation files should be modified frequently to reflect the DFR configuration or system model changes. The DFR recorded data supplied in native DFR format are converted to COMTRADE file using DFR Assistant software [22] which can generate an analysis report (containing the type of fault and a possible faulted line) in addition to generating the COMTRADE files.

TABLE 4.1
Data and related sources required for OFLA

Type of Data	Function Module	Detailed Data	Related Source	Description
System Level	Power Flow Analysis	Power Flow Raw Data	Input Data Files for PSS/E (*.raw)	This file contains power flow system specification data for the establishment of a static system model. This data is used by PSS/E to run the power flow analysis.
	Tuning of Static System Model	SCADA PI Historian Data		This file contains the latest load, branch and generator data to tune the static system data with the actual pre and post fault conditions
	Short Circuit Study	Sequence Impedance Data	Input Data Files for PSS/E (*.seq)	This file contains the negative and zero sequence impedance data needed for short circuit study. It is used by PSS/E to add the impedance data to the case of interest.
Local Level (Field Data)	Optimized fault selection algorithm	Recorded Data during Faults	DFR Configuration (*.cfg) and Data (*.dat)	These files are the fault events to be used by Optimized Fault Location algorithm.
		Substation Interpretation Data	Interpretation Data (*.int) for each Substation of interest	The “Interpretation File” contains information that relates the channel numbers to the monitored signals. It also represents the correspondence between the designations used in the DFR files and those used in the PSS/E file.

4.2.3 Data handling

The event data obtained from IEDs (in this case DFRs) should be pre-conditioned before using in the applications. This section will illustrate how the data captured by DFRs are converted into information and integrated with system level data to be used in OFLA.

4.2.3.1 *Extraction of phasors*

Once the disturbance events are obtained from the IEDs, two processing steps are taken to obtain phasors from the triggered event signals:

- Removal of high-frequency noises by low-pass filtering
- Use of an improved Fourier algorithm to effectively remove decaying dc-offset component and obtain the pre-fault and during-fault phasors of voltages and currents [40]

The pre-fault phasor can be calculated using first cycle of the recorded phasor. The during-fault phasor can be calculated using any fault cycle following the fault inception and prior to fault clearance. Therefore identification of the moment of fault inception is required before calculating the during-fault phasors. The fault cycle is determined from the waveforms recorded by DFR.

It is possible to select different fault cycle to calculate the during-fault phasors from the different DFR recordings. This may introduce fault location error, especially for the arcing faults during which the fault resistance is changing. Under this situation, selecting different fault cycles means experiencing different impacts of fault resistance. An alternative is that the user checks the waveforms manually and specifies the same matching fault cycles across all recordings.

4.2.3.2 *Synchronization of phasors*

It is well known that the phase angle of a selected signal in a three-phase system, calculated by a Fourier algorithm, may be constant or it may rotate $360/N$ (N is sample times per cycle) degrees with each new sample depending upon selection of the time reference. For a specific fault occurring in a power system, several DFRs may be triggered. Phasors calculated by using waveforms recorded in different substations may lack synchronization since the time reference in different substation may not be the same. This will introduce phase angle difference among phasors calculated using waveforms from the substation where DFRs are triggered. This factor will introduce an error in calculating the fault location using during-fault phasor. The synchronization of the phasors obtained from different DFR recordings is necessary.

The PSS/E load flow study based on the modified system model (actually the real-time power system model obtained from SCADA PI Historian), is carried out to obtain the pre-fault simulated phasors. The phasors calculated from each DFR recording are angle-synchronized by rotating them in reference to the phasors obtained by the load flow study. As a result, each recorded pre-fault phasor has the same phase angle as the corresponding one obtained through simulation. It is assumed that the angle difference between the pre- and during-fault phasor, for the corresponding recorded current or voltage, is fixed. Hence, the during-fault phasor should also be rotated for the same angle. All recorded pre- and post-fault phasors are synchronized using the same reference.

4.2.3.3 Retrieval of the Power Grid Switching Status [41]

Power grid topology (switching status) describes connectivity of various components in a power system. Beside event recordings, FL algorithms may require knowledge of the switching status in order to process retrieved fault event measurements. This knowledge is related to a specific position from which the waveforms and contacts are measured and the information how the measurement positions were interconnected at the time of the fault occurrence. Therefore, the system topology must be known. Some FL algorithms use information about transmission line length and impedance, while some use power flow information in addition. Beside the network connectivity it is necessary to obtain information about power system component characteristics at a specific moment in time.

Most utilities have power system model information in a format that is defined by the PSS/E or some other Short Circuit program. For example, using PSS/E program, status of the equipment can be updated according to the obtained real time status values. As said before, operator is able to track these changes in real time using Supervisory Control and Data Acquisition (SCADA) system. Through SCADA database, low-speed recording typically used to capture short term and long term disturbances is available. Captured data is typically scanned every second or two and it is usually phasor or RMS data, not sampled data [42].

It is important to notice that SCADA database is fed by Remote Terminal Units (RTUs). The modern RTUs can be very sophisticated recording instruments that may have a recording performance of a DFR, and at the same time may produce a variety of pre-calculated quantities. However, RTUs are primarily designed to interface EMS SCADA database using mostly

customized communication protocols and database formats [43]. Basically it is not possible to access recorded data before it is sent to a centralized location. The only way to retrieve those data beside SCADA system, which is available only in real time in a centralized location, is to use the archive. In order to retrieve pre fault power grid switching status during fault investigation EMS PI Historian data could be used. The PI Historian is capable of retrieving the load, branch and generator data scan in a period before and after the fault.

Finally, one of the most important inputs from the power system model needed for FL estimation is transmission line impedance. System model in PSS/E format contains these values. Methods used for calculation are [44]:

- Based on Carson's equations
- Making direct measurement of the open and short circuit voltages and currents
- Solving 2-port equations based on synchronous phasor measurements obtained from direct measurement from a digital relay or fault recorder.

By combining the above approaches, specially the first and third which do not require special equipment for testing, impedances can be easily obtained. It is important to notice that impedances depend on different factors such as temperature, current magnitude, frequency etc. This may cause the calculated impedances to deviate from real ones and this may influence accuracy of FL estimation.

4.2.4 Implementation

OFLA updates power system status with retrieved data, processes new event files, decides the most suitable FL algorithm and executes it. The output of the software is a fault report which can be used in a visualization module to generate Visual-Interactive-Distributed (VID) Spreadsheet, providing view into physical environment surrounding exact location of the fault and also views of involved equipment [33]. Further details are given in the PSerc report dedicated to Optimized Fault Location [26].

The architecture of the OFLA is shown in Figure 4.2. [26, 41]

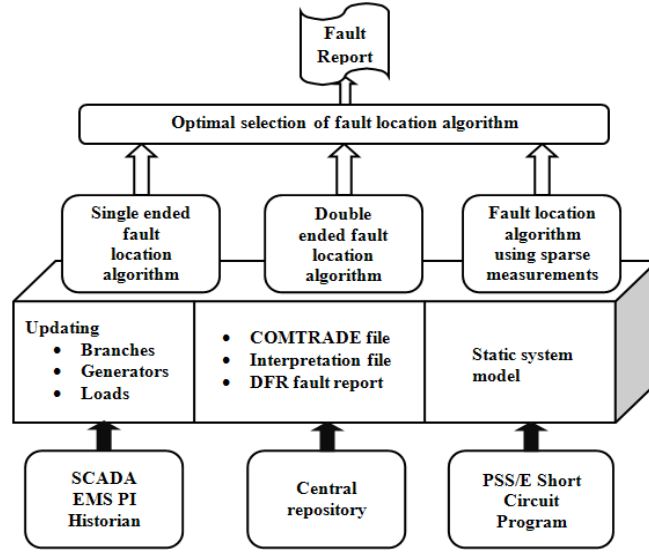


Figure 4.2. Architecture of Optimized Fault Location Algorithm

4.2.5 Evaluation of the algorithm

Implementation of OFLA is evaluated for the following issues: using varying number of DFR files, specifying the search region, using preprocessed fault location estimation, using different quantities for the match between measured and simulated data, evaluating differences in the accuracy when different input data are available and different assumption are satisfied etc. These different options may produce different results. Test activities are conducted on the data collected from a real life electric power system [41]. Again, further details are given in the PSerc report dedicated to Optimized Fault Location [26]

4.3 Intelligent Alarm Processing [45]

With the growth of power system complexity, operators are often overloaded with alarm messages generated by the events in the system. A major power system disturbance could trigger hundreds and sometimes thousands of individual alarms and events [46]. Obviously, this is beyond the capacity of any operators to handle. Thus, operators may not able to respond to the unfolding events in a timely manner, and even worse, the event interpretation by the operators may be either wrong or inconclusive affecting their ability to perform expected actions. The task of an intelligent alarm processor is to analyze thousands of alarm messages and extract the information that concisely explains the network events.

A lot of research has been done on the Fuzzy Reasoning Petri-nets (FRPN) [47-49]. FRPN takes

advantages of Expert System and Fuzzy Logic, as well as parallel information processing to solve the problem of fault section estimation. Reference [50] gives an optimal design of a structure of FRPN diagnosis model. This particular approach is used to illustrate how data from substation IEDs may affect the ways Intelligent Alarm Processor may be implemented. Further details of this approach are contained in a study recently completed for ERCOT [45]

4.3.1 FRPN diagnosis model

When one or more faults occur on certain sections of the power system, protection devices will reach certain statuses accordingly. The observed relay trip signals and circuit breaker status signals obtained from RTUs of SCADA systems are used as inputs for estimation of the faulted sections. The strategy is to build one FRPN diagnosis model for each section of the power system. Each model establishes reasoning from a set of SCADA data to the conclusion of fault occurrence on its associated section with certain truth degree value. In case of single fault, the conclusion with the highest truth degree value is the final conclusion. In case of multiple faults, the several conclusions with the highest truth degree values which are greater than a threshold are regarded as the final conclusions.

To build the FRPN diagnosis models, several issues should be carefully considered. First, a sound methodology for structure design needs be adopted to achieve good diagnosis performance while keeping the model size small. Second, to deal with failures of protective devices, backup protection operations need to be considered in the models. Third, to tackle false data problem introduced by defects of protection devices, measurement systems or communication systems, fuzzy logic concept needs to be effectively utilized.

We use backward reasoning concept to structure the FRPN diagnosis models and generalize the design for transmission lines and buses [51] [10]. Figure 4.3 and Figure 4.4 illustrate backward reasoning concept for structuring transmission line and bus diagnosis models respectively. The ‘AND-OR’ structure concisely represents all the possible combinations of main, primary backup and secondary backup protection operations for inferring a fault. Compared with the ‘OR-AND’ “enumeration” type of structure used in [52], our proposed structure effectively covers more scenarios with smaller number of rules, which will eventually achieve higher diagnosis accuracy with smaller size of Petri-nets matrix.

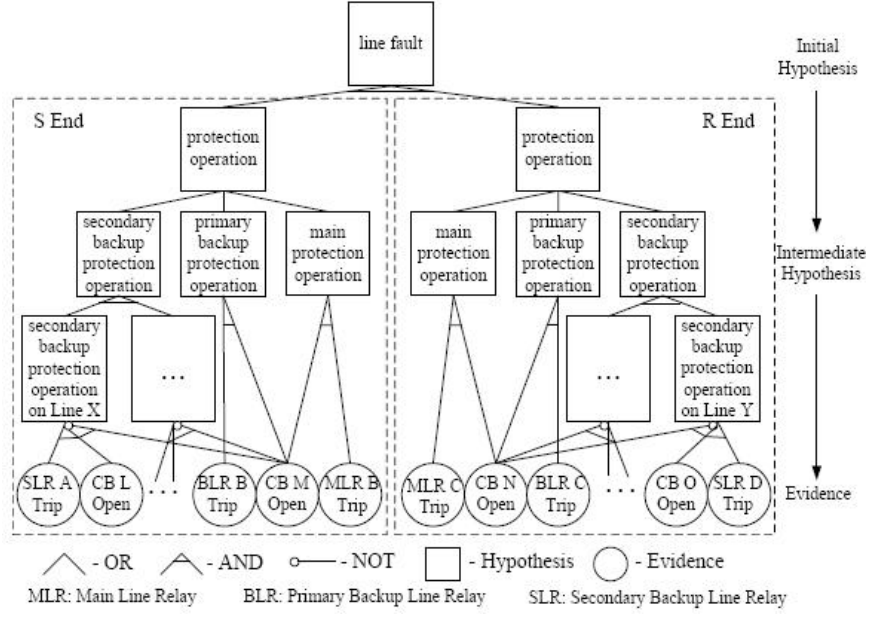


Figure 4.3. Backward Reasoning concept for structuring transmission line diagnosis models

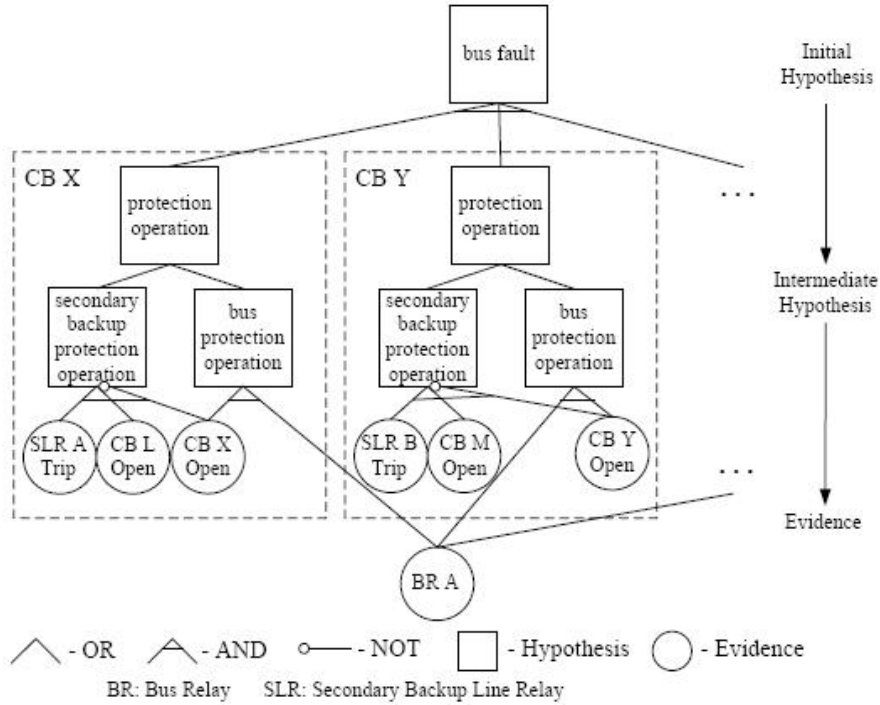


Figure 4.4. Backward Reasoning concept for structuring busbar diagnosis models

4.3.2 Data requirements

It has been proven that the logic operand data of digital protective relays can be used as additional inputs to enhance the alarm interpretation [10]. In a digital protective relay, the pickup and operation information of protection elements is usually in the form of logic operands [52]. The pickup and operation logic operands are more reliable than SCADA data because they are more redundant and have less uncertainty than relay trip signals and circuit breaker status signals.

In such a solution, input data such as relay trip signals and circuit breaker status signals are acquired by RTUs of the SCADA system. If such data are not available from SCADA, they can be brought from DPRs. Relay logic operand signals are defined in DPR data memories and retrieved from relays either directly or by the SCADA front-end computers in substations. The data are acquired from different substations and are transmitted to the control center through selected communication link such as microwave or optical fiber. In the control center, the SCADA master computer puts the input data into a real-time data base and keeps updating them at each scan time.

The detailed description of field data needed for this application is listed in Table 4.2.

TABLE 4.2
Data and related sources required for Intelligent Alarm Processor

Data from RTU of SCADA (Main data)	
1	CB status change alarms (Opening and Closing)
2	Trip signal of Main Transmission Line Relays
3	Trip signal of Primary Backup Transmission Line Relays
4	Trip signal of Secondary Backup Transmission Line Relays
5	Trip signal of Bus Relays
Data from Digital Protective Relays (*Additional data)	
1	Pickup & Operation signals of Main Transmission Line Relays
2	Pickup & Operation signals of Primary Backup Transmission Line Relays
3	Pickup & Operation signals of Secondary Backup Transmission Line Relays
4	Pickup & Operation signals of Bus Relays

4.3.3 Implementation

A 14-bus power system as shown in Figure 4.5 is used for the study of fault section estimation problem. The system consists of 34 sections, including 14 buses and 20 transmission lines. The buses are denoted as Bnn. The transmission lines are denoted as Lnnmm.

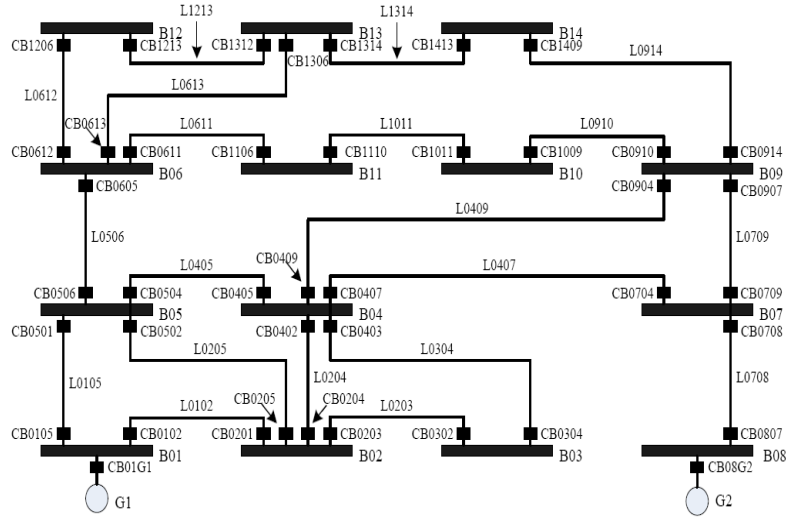


Figure 4.5. 14-Bus power system model

Based on the proposed structure, in Figure 4.3 and Figure 4.4 all the FRPN diagnosis models are developed. As an example, Figure 4.6 shows the FRPN models for the transmission line L1314.

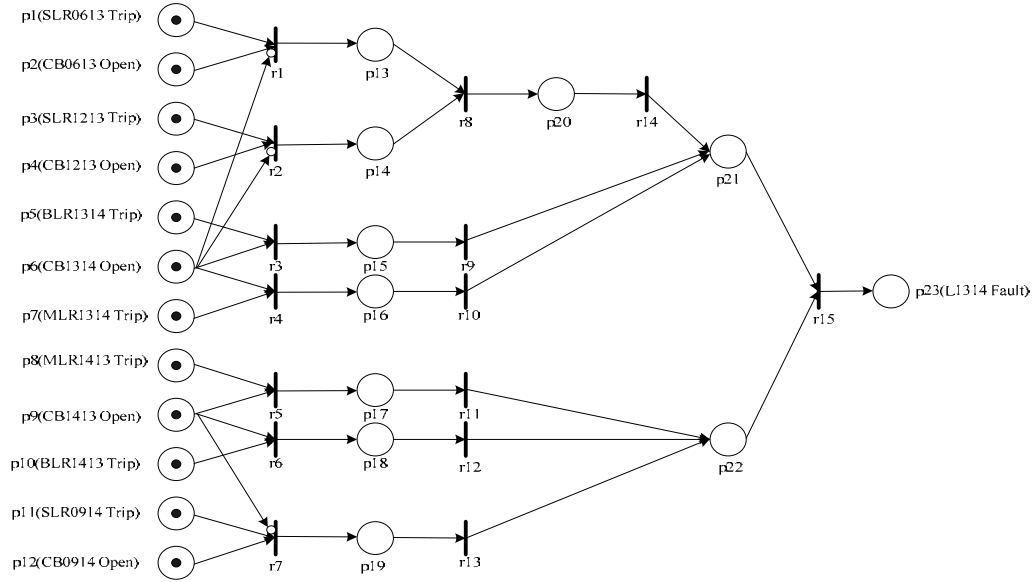


Figure 4.6. A FRPN model for L1314 fault based on SCADA data

The pickup and operation logic operands can be utilized to improve the accuracy of fault section estimation based on SCADA data, as shown in Figure 4.5. When a fault occurs on the

transmission line L1314, its associated protection system operated to respond to the fault. In addition to the observed SCADA data, the following relay signals are also observed: SLR0613 Pickup, SLR0613 Operation, SLR1213 Pickup, SLR1213 Operation, BLR1314 Pickup, BLR1314 Operation, MLR1314 Pickup, MLR1314 Operation, MLR1413 Pickup, MLR1413 Operation, BLR1413 Pickup, and SLR0914 Pickup. Since the relay data are more reliable than the SCADA data, they are given a larger truth value 0.98.

If MLR1413 Trip is missing in the SCADA data due to data transmission error while MLR1413 Pickup and MLR1413 Operation are observed, the conclusion will be that a fault occurs on the transmission line L1314 with a truth degree value 0.827.

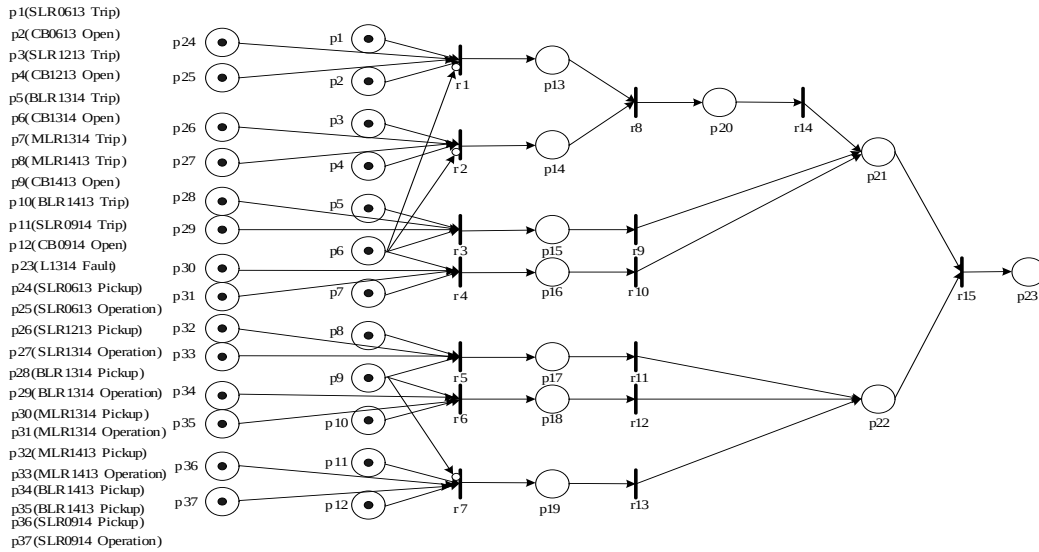


Figure 4.7. A FRPN model for L1314 fault based on SCADA and digital protective relay data

4.3.4 Case study

Based on the approach introduced in [10], a power system/protection system interactive simulation environment for the case study has been developed. The evaluation environment enables one to set up fault scenarios, insert user-defined errors, and generate SCADA data and relay data.

A permanent fault occurred on the bus B04 at 0.05 second. A second permanent fault occurred on the bus B09 at 0.09 second. All the protection devices operated correctly. No false data occur. The observed SCADA data are listed in Table 4.3. The observed relay data are listed in Table 4.4.

Based on the SCADA data in Table 4.3, the candidates for the fault section are estimated and results are listed in Table 4.5. Based on both the SCADA data in Table 4.3 and relay data in Table 4.4, the candidates for the fault section are estimated and the results are listed in Table 4.6.

As shown in Table 4.5 and Table 4.6, besides the bus B04 and the bus B09, on which faults actually occur, the transmission line L0409, which has no fault, is included in the candidate set. The transmission line L0409 has a far smaller truth degree value than the other two candidates, which indicates small possibility of fault occurrence. The truth degree values of the candidates based on both the relay data and SCADA data are higher than those based on only the SCADA data.

TABLE 4.3
SCADA data

Sequence No.	Time Stamp (Sec)	Observed Signal
1	0.1000	BR04 TRIP
2	0.2000	CB0402 OPEN
3	0.2000	CB0403 OPEN
4	0.2000	CB0405 OPEN
5	0.2000	CB0407 OPEN
6	0.2000	CB0409 OPEN
7	0.2000	BR09 TRIP
8	0.2000	CB0904 OPEN
9	0.2000	CB0907 OPEN
10	0.2000	CB0910 OPEN
11	0.2000	CB0914 OPEN

TABLE 4.4
Relay data

Sequence No.	Time Stamp (Sec)	Observed Signal
1	0.0662	SLR0409
2	0.0677	SLR0709
3	0.0693	BLR0910
4	0.0698	MLR0910
5	0.0703	MLR1009
6	0.0703	BLR1009
7	0.0703	SLR1110
8	0.0724	SLR1409
9	0.0740	MLR0910
10	0.0745	MLR1009

TABLE 4.5

Candidates for estimated fault sections based on SCADA data

Candidate No.	Fault Section	Truth Degree Value
1	B04	0.855
2	B09	0.855
3	L0409	0.513

TABLE 4.6

Candidates for estimated fault sections based on SCADA data and relay data

Candidate No.	Fault Section	Truth Degree Value
1	B04	0.882
2	B09	0.882
3	L0409	0.618

If the scenario is a single fault without protection device failure and false data, the faulted section can be accurately identified. The truth degree value of the result based on both the relay data and SCADA data are higher than that based on only the SCADA data, because the relay data are assigned higher truth degree values due to their higher reliability. However, multiple faults may occur at the same time, which is more complex than a single fault situation. As shown in Table 4.5 and Table 4.6, besides the bus B04 and the bus B09, on which faults actually occur, the transmission line L0409, which has no fault, is included in the candidate set. The transmission line L0409 has a far smaller truth degree value than the other two candidates, which indicates small possibility of fault occurrence. Similar to single fault situation, the truth degree values of the candidates based on both the relay data and SCADA data are higher than those based on only the SCADA data.

4.3.5 Deployment strategy

The fault section estimation application may be implemented in a control center to assist the system operator in rapidly identifying faulted sections for restoration process, as shown in Figure 4.8. [45]

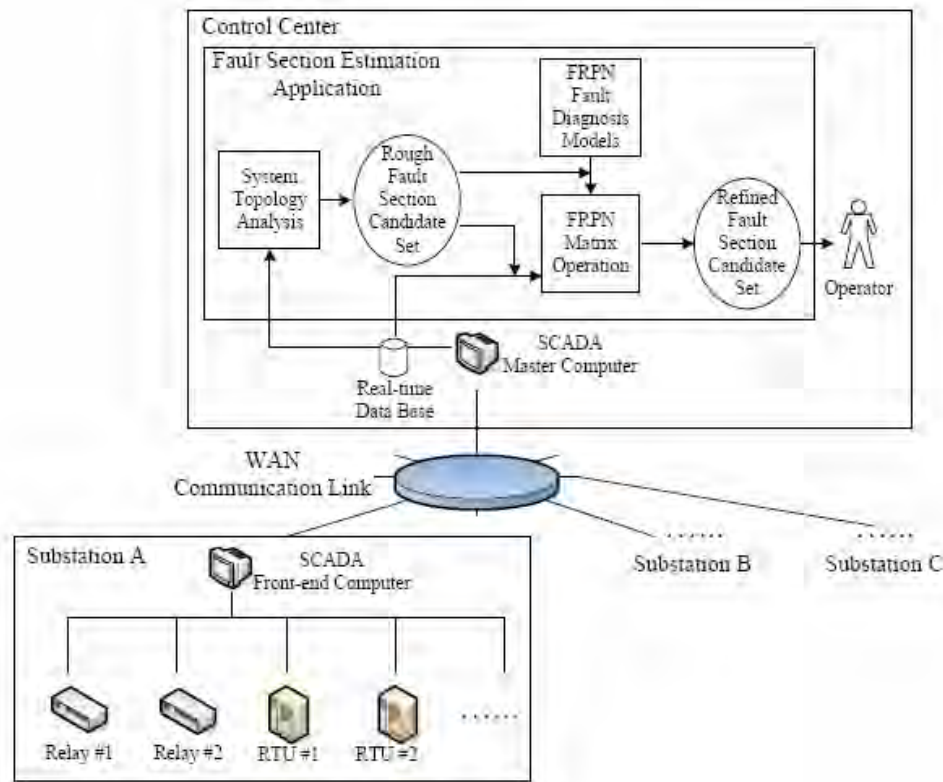


Figure 4.8. Overall Implementation Structure

The fault section estimation application includes two stage analysis:

- *First Stage*

The system's topology is analyzed based on circuit breaker status data from the real-time data base. The analysis includes all sections isolated by the opening of circuit breakers into a rough candidate set. The set is rough because it may include sections which are not faulted but are isolated due to backup relay operation.

- *Second Stage*

The Fuzzy Reasoning Petri-net diagnosis model as well as data in the real-time data base corresponding to each section in the rough candidate set is used and Fuzzy Reasoning Petri-net matrix operation is implemented. As a result, each section will be associated with a truth degree value. The section with a truth degree value greater than a certain threshold will be included in the refined candidate set. Such a refined candidate set is presented to the system operator for decision-making

In such a solution, the FRPN models which are represented by various matrices are separated from FRPN matrix operations. This is analogous to an expert system whose rule base is separated from its inference engine. The FRPN models can be built in advance based on power system and protection system configurations and stored in files. In such a way, the FRPN models can be easily modified according to the changes of input data as well as power system and protection system configuration.

4.4 Detection, Classification and Mitigation of Cascading Outages [53]

Power system cascading outage is quite often a complicated phenomenon, which may finally result in a large area blackout. It can cause great economic consequences and devastating impact on people's life. For example, Northeastern System Blackout in 2003 led to the load loss of 61.8GW, which influenced more than 50 million people [1].

There are many factors contributing to power system cascading outages, such as inadequate understanding of unfolding events, inadequate operational awareness, inadequate tree trimming, relaying problems, bad weather conditions, human errors, etc [1]. No single factor may be the cause of system cascading outage and large area blackout. From the technical side, relaying setting problems and inadequate understanding of unfolding events are two major contributing factors, although neither of them may be the triggering cause. Relaying problems were the contributing factor in almost 70% of the US disturbances from 1984 to 1991 [54]. Relay failure and misoperation during weakened system conditions contribute the most to the cascading outages. Another problem is that power system operators lack sufficient analysis and decision support to take quick corrective actions due to inadequate understanding of unfolding events. This was clearly demonstrated in the August 1996 US Western Coast System Blackout and August 2003 US Northeastern System Blackout [1, 55].

Many researchers and engineers have put lots of research efforts in this topic. The proposed solutions are aimed at understanding and finding ways to detect, prevent, and mitigate the cascading events, which includes dynamic and probabilistic study of the cascade model, dynamic decision-event tree analysis, wide area back-up protection expert system, relay hidden failure analysis, special protection scheme, controlled islanding, generalized line outage distribution factors calculation, etc. [56-60] There are no effective interactive monitoring and control tools developed so far to detect and mitigate the cascading outage.

Recently, a new interactive scheme of system/local monitoring and control tools for detection, prevention and mitigation of cascading events was recently introduced [61-65]. This scheme can identify the disturbance information and provide control means for preventing further unfolding of the cascading events and keeping the stability of the power system. The implementation of the proposed algorithm is highly depended on the availability of ready-for-use data. This application illustrates how substation data may be integrated with system-wide data to provide new solutions.

4.4.1 Algorithm

The proposed monitoring and control scheme for detection, prevention and mitigation of cascading events coordinates the system-wide and local analysis tools. The overall interactive solution is conceptually shown in Figure 4.9.

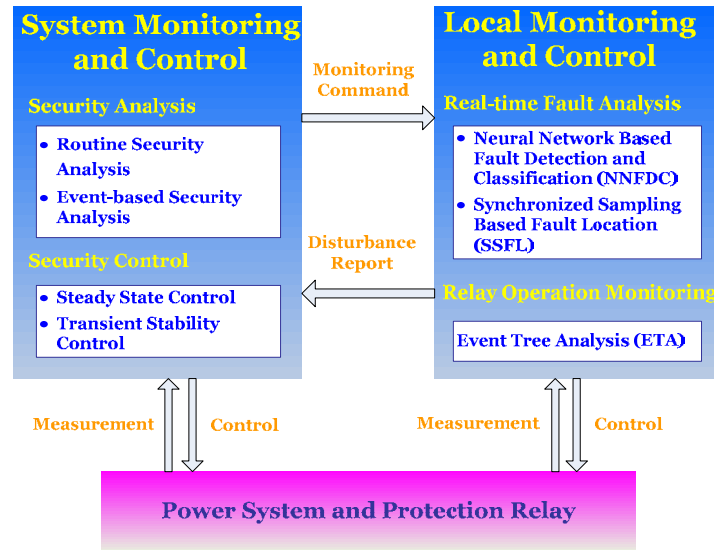


Figure 4.9 .Overall interactive monitoring and control scheme

The system tool consists of the routine and event-based security analyses based on the power flow method and topology processing method, along with the security control schemes for expected and unexpected events. Vulnerable elements are identified using calculated Vulnerability Index and Margin Index [61] and related relays are closely monitored. The event-based security analysis is triggered when an unexpected disturbance occurs. It indicates whether the emergency control is needed to mitigate the transient stability problem or not. Steady state control tools are based on Network Contribution Factor (NCF) [62], Generator Distribution

Factor (GDF) [66], Load Distribution Factor (LDF) [66], and Selected Minimum Load Shedding (SMLS) used for early detection and prevention of cascading outages as analyzed utilizing steady state method. Transient stability control tools are based on Potential Energy Boundary Surface (PEBS) method [67], Admittance-based Control (ABC) and Generator-input-based Control (GIBC).

The local monitoring and control tool intended for installation at local substations consists of an advanced real time fault analysis tool and a relay operation monitoring tool utilizing neural network based fault detection and classification algorithm (NNFDC), synchronized sampling based fault location algorithm (SSFL), and event tree analysis (ETA) [66-68]. The advanced on-line fault analysis tool will detect the disturbance by analyzing local measurements. Once the disturbance is detected and classified, event tree analysis process will be invoked to validate relay operations. The combination of the two algorithms performs a more accurate fault analysis than conventional relays do. This provides a reference for monitoring and verification of the distance relay operations. The substation-based solution can provide the system-wide tool with local disturbance information and diagnostic support so that the system-wide tool can have correct local information and take better control action to ensure the secure of operation.

4.4.2 Data requirements

Data availability is the prime factor to make the automatic monitoring and control feasible. In order to implement the proposed algorithms of system/local level, two issues needed to be cleared: what type of data is needed and how to extract the useful information from obtained data. To solve these issues, different types of data from various sources are needed, which are naturally divided into two classes: system level data and substation data.

4.4.2.1 System Level Data

The data required for system level analysis aims at getting the power system network topology and establishing the steady state status. Thus the power system model and real time network topology change information are needed. Different utilities may use different commercial programs for power system operations. However, the main functions of those programs are same. To meet the data requirement for system analysis and control tools, the following detailed data needs to be obtained:

- *Data for Power Flow Analysis*

This data contains the basic power system network topology. It is used to run the power flow analysis to establish the steady state status of whole system.

- *Data for Short Circuit Analysis*

This data contains the supplemental input data needed to run short circuit calculation for a given time when the fault has occurred. It is acquired whenever the system topology changes.

- *Data for Dynamic Analysis*

This data contains the dynamic models for generators, exciters, compensators, stabilizers, etc, which is used to perform dynamic analysis to check system dynamic stability under various disturbances.

4.4.2.2 Local Level Data

The data required for local level aims at getting detailed configuration for each substation, as well as measured data for any disturbances. To meet the data requirement for local fault analysis and control tools, the following detailed data needs to be obtained:

- *Relay Setting and Event Report Data*

Data for relay setting contains the relay configuration information. Data for relay event report contains the results of relay operations under disturbances. They can be used for rebuilding relay operation scenarios and simulating relay behavior to verify relay operation.

- *Recorded Fault Data*

Data recorded from fault contains analog and digital sample values for all input channels for a specific fault or disturbance. It can be used to for the training and testing sets of fault scenarios for Neural Network based Fault Detection and Classification (NNFDC) Algorithm.

- *Synchronized Samples of Fault Voltages and Currents*

Synchronized samples of fault voltages and currents contain time stamps from Global Positioning System (GPS) of satellites on all the pre- and during-fault sample signals [66]. It can be used to verify the results of the Synchronized Sampling based Fault Location (SSFL) Algorithm.

4.4.3 Data handling

This section will discuss how to obtain the desired data and pre-process those data to meet the requirements of system/local algorithms.

4.4.3.1 Obtaining System Data

As mentioned above, the system level data aims at getting the power system network topology and establishing the steady state status. Different utilities may adopt different programs for system analysis and operation. For this demonstration, we choose Siemens PTI Power System Simulator for Engineering (PSS/E) as the basic program to implement the system level tools. The static power system is modeled using PSS/E. Table 4.7 shows the detailed description of system data needed for system level applications.

TABLE 4.7
Data and related source required for system level analysis

Function Module	Detailed Data	Related Source	Description
Security Analysis	Power Flow Raw Data	Input Data Files for PSS/E (*.raw)	This file contains power flow system specification data for the establishment of an initial case. This data is used by PSS/E to get the initialization values which will be utilized for security analysis.
	Binary Data or Drawing Coordinate Data	Input Data Files for PSS/E (*.sld for V30 or *.drw for V2x)	This file is used to create and modify one-line diagrams and to display a variety of results.
Stability Analysis	Sequence Impedance Data	Input Data Files for PSS/E (*.seq)	This file contains the negative and zero sequence impedance data needed for unbalanced fault analysis. It is used by PSS/E to add the impedance data to the case of interest.
	Dynamics Data	Input Data Files for PSS/E (*.dyr)	This file is used by the PSS/E Dynamics program. It contains synchronous machines and other system components data for input to the PSS/E dynamic simulation working memory.

4.4.3.2 Obtaining Local Data

As mentioned above, the data required for local level analysis aims at getting detailed configuration for each substation, measured data for any disturbances. The data needed for local level analysis can be divided into two classes:

- Field data: event data from different IEDs and SCADA system of any disturbance
- Simulation data: fault or non-fault scenarios regarding each transmission line for the NNFDC algorithms.

Acquisition of Field Data:

The field data are acquired by different IEDs and SCADA. All the fault waveform will come from IEDs, since they have capability for capturing waveforms with relatively high sample rate. The status of system configuration will be extracted from SCADA data. The field recorded data should follow the COMTRADE format based on measured data, configuration, data and interpretation files. Table 4.8 shows the detailed description of Field data acquisition needed for local level applications.

TABLE 4.8
Data and related source required for local level analysis

Detailed Data	Related Source	Description
Relay Setting	Relay Database	This file is used to set the relay to be tested in the lab
Event File	Event File output from Relays	The relay event file is used to get the report of relay operations.
Recorded Data during Faults	DFR files	This fault data file is used to check the results for relay operations using Neural Network based Fault Detection and Classification (NNFDC)
Synchronized Samples of Voltage and Current Waveforms during Faults	GPS-Synchronized DFR files	This synchronized data file is used for evaluation of Synchronized Sampling based Fault Location (SSFL)
SCADA data	SCADA system	This data will offer the real time system configuration and system status.

After the COMTRADE format data files are obtained, the useful fault information needs to be extracted.

For NNFDC, the three phase voltages and currents of pre-fault and during-fault are needed. The fault type needs to be determined in order to perform training and testing of the neural network. Together with generated simulation of fault and non-fault scenarios, the training and testing sets of cases are built for each transmission lines. NNFDC is trained according to the parameters set for each transmission line. For SSFL, the synchronized samples are the voltage and current data from DFRs with GPS time stamp. The integration of NNFDC and SSFL will build the advanced fault analysis tools at the local level.

Generation of Simulated Data:

NNFDC algorithm needs waveforms from large number of fault and non-fault cases to complete the process of training and testing for neural network. Those training and testing cases are totally different for various transmission lines due to the different parameters and setting. The simulations are performed because it is not possible to acquire enough fault cases from the field. Figure 4.10 shows the block diagram of the program for generation of the fault and non-fault scenarios based on ATP/ATPDraw and Matlab [68]. The power system of interest is first modeled in ATP/ATPDraw. Then user can define the desired fault or non-fault cases by initializing the parameters set in Matlab. The measured three-phase voltage and current samples, which could also include time label, are extracted in the data format files defined by user.

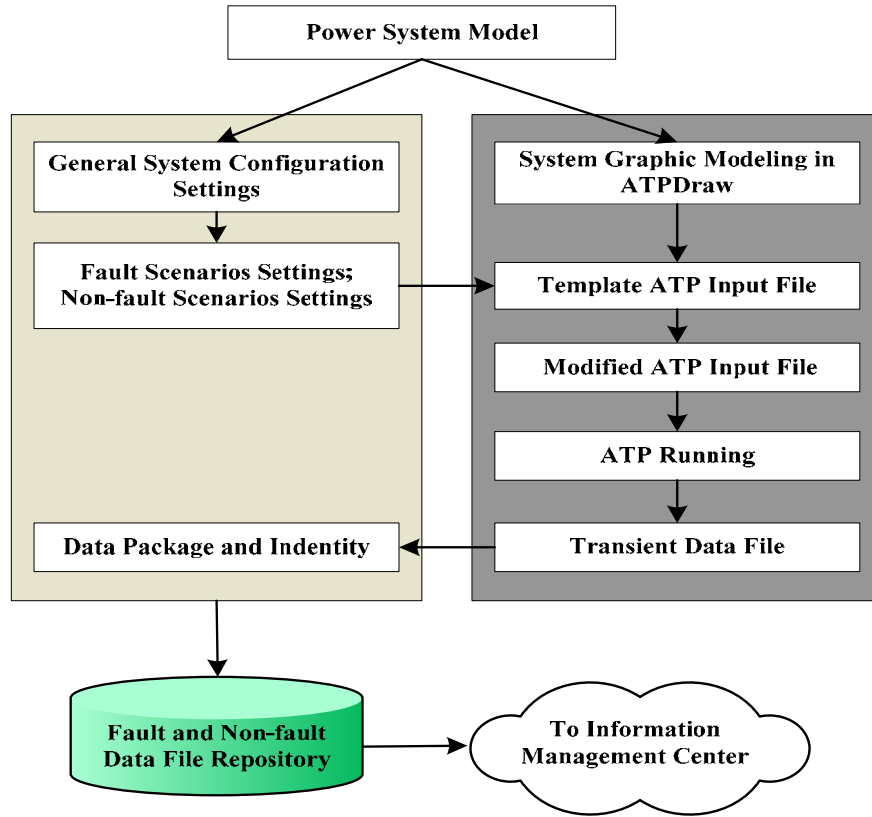


Figure 4.10. Block diagram of the program for simulation fault and non-fault scenarios

4.4.4 Implementation

System-wide tool has wide-area information and comprehensive view of the system security and vulnerability conditions. It can notify the local tool to engage in careful monitoring of vulnerable elements during abnormal conditions. The Local tool collects the exact real-time local disturbance information. It has the ability to detect, classify and locate the fault with high accuracy and provide good reference for evaluating relay operation. Both the system-wide and local tools work together to fulfill the major task: to help detect, and classify the cascading outages. The block diagram of the implementation is shown in Figure 4.11.

5. Conclusion

This chapter focuses on the accomplishments of the project. The report addressed the following issues that need to be understood to appreciate how local substation data may be used to enhance EMS applications:

- The nature of the data collected from substations by using newly implemented intelligent electronic devices:
 - The performance requirements of the sensors used for capturing the data
 - The performance requirements for the internal processing of the IEDs
 - Integration of the various types of data obtained from different types of IEDs from different vendors
 - Exchange of information obtained from the integrated data automatically from the substation data repository
- The use of the integrated data for some examples of EMS functions:
 - Optimal fault location algorithm: It has been explained how the data from DFRs located sparsely in the substation can be efficiently integrated with the real time power system data and utilized to estimate a correct location of transmission line faults.
 - Intelligent alarm processing: A fuzzy reasoning Petri net based alarm processing algorithm is implemented which can integrate relay data with SCADA data to analyze thousands of alarm messages and extract the information that concisely explains the network events.
 - Detection, classification and mitigation of cascading outages: The new approach of the detecting and preventing the cascading events by coordinating the system-wide and local tools could provide control means for preventing further unfolding of the cascade. The possible means to obtain the desired data and information are presented, which are based on the integration of data from SCADA and substation IEDs

6. References

- [1] U.S.-Canada Power System Outage Task Force, “*Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*”, April 5, 2004, Available: <http://www.nerc.com>
- [2] M. Kezunovic, A. Abur, A. Edris, D. Sobajic, “*Data Integration/Exchange Part I: Existing Technical and Business Opportunities*”, IEEE Power and Energy Magazine, pp. 14 - 19, March/April 2004.
- [3] A. Phadke, et. al., “*Synchronized Sampling and Phasor Measurements for Relaying and Control*”, IEEE Trans. Power Delivery, Vol. 9, No. 1, pp. 442-452, January 1994.
- [4] IEEE Inc., “*IEEE Standard Requirements for Instrument Transformers*”, IEEE Std C57.13-1993.
- [5] IEC Standard for Instrument Transformers- IEC Std 60044-X-1992
- [6] IEEE Inc., “*IEEE Standard for High-Accuracy Instrument Transformers*”, IEEE Std C57.13.6-2005.
- [7] M. Kezunovic, “*Data Integration and Information Exchanged for Enhanced Control and Protection of Power Systems*,” Hawaii International Conference on Systems Science, Waikoloa Village, Hawaii, January 2003.
- [8] IEEE Inc., “*IEEE Standard for Synchrophasors for Power Systems*”, IEEE Std C37.118-2005.
- [9] Interim Report: “*Synchrophasor Measurement Accuracy Characterization*”, North American Synchro Phasor Initiative Performance & Standards Task Team, August 26, 2007.
- [10] X. Luo, M. Kezunovic, “*Implementing fuzzy reasoning Petri nets for fault section estimation*”, IEEE Trans. Power Delivery, Vol.23, No.2, pp.676-685, April 2008.
- [11] S. Luo, M. Kezunovic, D.R. Sevcik, “*Locating Faults in the Transmission Network Using Sparse Field Measurements, Simulation Data and Genetic Algorithms*”, Electric Power Systems Research, Vol. 71, No. 2, pp.169-174, October 2004.
- [12] IEEE Inc., “*IEEE standard Common Format for Transient Data Exchange (COMTRADE) for power systems*”, IEEE Std C37.111-1991, 1991.
- [13] IEEE Inc., “*IEEE standard Common Format for Transient Data Exchange (COMTRADE) for power systems*”, IEEE Std C37.111-1999, 1999.
- [14] IEC Std. 60255-24, “*Common format for transient data exchange (COMTRADE) for power systems*”, First Edition 2001-05, International Electrotechnical Commission, 2001.
- [15] IEEE Inc., “*IEEE Recommended Practice for the Transfer of Power Quality Data (PQDIF)*”, IEEE Std 1159.3-2003.
- [16] IEEE Inc., “*IEEE Recommended Practice for Naming Time Sequence Data Files*”, IEEE Std C37.232-2007.
- [17] IEC Std. 61850, “*Communication Networks and Systems in Substations*”, work in progress, International Electrotechnical Commission, [Online]. Available: <http://www.iec.ch>.
- [18] IEC Std. 61968 (Draft), “*System Interfaces for Distribution Management*”, International Electrotechnical Commission, [Online]. Available: <http://www.iec.ch>.
- [19] M. Kezunovic, D. R. Sevcik, R. Lunsford, T. Popovic, “*Integration and Use of Substation Data*”, Fault and Disturbance Conference, Atlanta, Georgia, May 2007.

- [20] M. Kezunovic, A. Abur, D.Sobajic, “*Managing complexity through data integration and information exchange*”, IREP Symposium, Bulk Power System Dynamics and Control, Cortine D’Ampezzo, Italy, Aug. 2004.
- [21] M. Kezunovic, P. Spasojevic, C. Fromen, D. Sevcik, “*An Expert System for Transmission Substation Event Analysis*”, IEEE Trans. Power Delivery, Vol. 8, No. 4, pp. 1942-1949, October 1993.
- [22] Test Laboratories International, Inc.: “*DFR Assistant - Software for Automated Analysis and Archival of DFR records with Integrated Fault Location Calculation*” [Online]. Available: <http://www.tliinc.com>
- [23] X. Luo, M. Kezunovic, “*An Expert System for Diagnosis of Digital Relay Operation*”, 13th Conference on Intelligent Systems Application to Power Systems, Washington DC, USA, November 2005.
- [24] M. Kezunovic, Z. Ren, G. Latisko, D.R. Sevcik, J. Lucey, W. Cook, E. Koch, “*Automated Monitoring and Analysis of Circuit Breaker Operation*”, IEEE Trans. Power Delivery, Vol. 20, No. 3, pp. 1910-1918, July 2005.
- [25] M. Kezunovic, Y. Liao, X. Xu, A. Abur, “*Power Quality Assessment Using Advanced Modeling, Simulation and Data Processing Tools*”, Int’l. Power Quality Conference 2002, Singapore, October 2002.
- [26] M. Kezunovic, “*Optimized Fault Location*”, Final report for Concurrent Technologies Corporation, PSerc publication 08-07, April 2008.
- [27] M. Kezunovic, and B. Perunicic, “*Fault Location*”, Wiley Encyclopedia of Electrical and Electronics Terminology , Vol.7, pp. 276-285, John Wiley 1999.
- [28] J. C. S. de Souza, et. al, “*Fault Location in Electrical Power Systems Using Intelligent Systems Techniques*”, IEEE Trans. Power Delivery, Vol. 16, No. 1, pp. 59-67, January 2001.
- [29] K.K. Li, L.L. Lai, A.K.David, “*Application of artificial neural network in fault location technique*”, International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, pp.226 – 231, 4-7 April 2000.
- [30] H. Kanoh, K. Kanemaru, M. Kaneta, M. Nishiura, “*A study on practical fault location system for power transmission lines using neural networks*”, Proceedings of the Second International Forum on Applications of Neural Networks to Power Systems, pp.9 – 13, 1993.
- [31] C.K. Jung, K.H. Kim, J.B. Lee and Bernd Klöckl, “*Wavelet and neuro-fuzzy based fault location for combined transmission systems*”, International Journal of Electrical Power & Energy Systems, Vol. 29, No. 6, pp. 445-454, July 2007.
- [32] M. Kezunovic, B. Perunicic, and J. Mrkic, “*An Accurate Fault Location Algorithm Using Synchronized Sampling*”, Electric Power Systems Research Journal, Vol. 29, No. 3, pp. 161-169, May 1994.
- [33] D. Novosel, D.G. Hart, E. Udren, and J. Garitty, “*Unsynchronized two-terminal fault location estimation*”, IEEE Trans. Power Delivery, Vol. 11, No. 1, pp. 130 – 138, Jan. 1996.
- [34] T. Takagi, Y. Yamakoshi, M. Yamaura, R. Kondow, and T. Matsushima, “*Development of a New Type Fault Locator Using the One-Terminal Voltage and Current Data*”, IEEE Trans. on Power App. and System, Vol. 101, No. 8, pp 2892-2898, Aug. 1982.
- [35] M. Kezunovic, Y. Liao, “*Fault Location Estimation Based on Matching the Simulated and Recorded Waveforms Using Genetic Algorithms*”, Development in Power System Protection, Amsterdam, The Netherlands, April 2001.

- [36] M. Kezunovic, E. Akleman, M. Knezev, O. Gonen, S. Natti, “*Optimized Fault Location*”, IREP Symposium 2007, Charleston, South Carolina, Aug 2007.
- [37] PSSTME Load Flow ©Siemens Power Transmission & Distribution, Inc., PTI, Schenectady, NY, USA
- [38] <http://techsupport.osisoft.com/>
- [39] PowerWorld Corporation: “*PowerWorld Retriever*”, [Online] <http://www.powerworld.com/>
- [40] Yong Guo, M. Kezunovic, Deshu Chen, “*Simplified algorithms for removal of the effect of exponentially decaying dc-offset on the Fourier algorithm*”, IEEE Transaction on Power Delivery, Vol.18, No.3, July 2003.
- [41] M. Knezev, “*Optimal Fault Location*”, M.S. Thesis, Dept. of Electrical Engineering, Texas A & M University, 2007.
- [42] IEEE Inc., “*Considerations for use of disturbance recorders*”, A report to the System Protection Subcommittee of the Power System Relaying Committee of the IEEE Power Engineering Society, Dec 2006.
- [43] M. Kezunovic, C.C. Liu, J. McDonald, L.E. Smith, “*Automated fault analysis*”, IEEE Tutorial, IEEE PES, 2000.
- [44] IEEE Inc., “*IEEE Guide for determining fault location on AC transmission and distribution lines*”, IEEE Std. C37.114-2004, Jun 2005.
- [45] M. Kezunovic, Y. Guan, “*Intelligent Alarm Processor*”, Final Report, ERCOT, March, 2008
- [46] M. Kezunovic, J. Domaszewicz, V. Skendzic, M. Aganagic, J.K. Bladow, S.M. McKenna, D.M. Hamai, “*Design, implementation and validation of a real-time digital simulator for protection relay testing*”, IEEE Trans. Power Delivery, Vol. 11, No. 1, pp158-164, January 1996.
- [47] S. M. Chen, J. S. Ke, and J. F. Chang, “*Knowledge representation using fuzzy Petri nets,*” *IEEE Trans. Knowledge and Data Engineering*, Vol. 2, No. 3, pp. 311–319, September 1990.
- [48] T. V. Manoj, J. Leena, and R. B. Soney, “*Knowledge representation using fuzzy Petri nets revisited*”, *IEEE Trans. Knowledge and Data Engineering*, Vol. 4, No. 10, pp. 666–667, July 1998.
- [49] M. M. Gao, M. C. Zhou, X. G. Huang, and Z. M. Wu, “*Fuzzy reasoning Petri nets*”, *IEEE Trans. Systems, Man and Cybernetics, Part A*, Vol. 33, No. 3, pp. 314–324, May 2003.
- [50] J. Sun, S.Y. Qin and Y. H. Song, “*Fault diagnosis of electric power systems based on fuzzy Petri nets*”, *IEEE Trans. Power Systems*, Vol. 19, No. 4, pp. 2053-2059, Nov 2004.
- [51] Xu Luo, Doctoral Dissertation, Texas A&M University, 2007.
- [52] *Instruction Manual for D60 Line Distance Relay*, General Electric Company, Fairfield, CT, 2004.
- [53] M. Kezunovic, C. Pang, “*Detection, Prevention and Mitigation of Cascading Events; Part I*”, Final Report, PSerc, September, 2008.
- [54] NERC System Disturbances Reports, North American Electric Reliability Council, New Jersey, 1984-1991.
- [55] NERC Disturbance Analysis Working Group, “*Western Interconnection (WSCC) System Disturbance — August 10, 1996*”, NERC 1996 System Disturbances Report, Aug 2002, Available: [http:// www.nerc.com](http://www.nerc.com)
- [56] Q. Chen, K. Zhu, and J.D. McCalley, “*Dynamic decision-event trees for rapid response to unfolding events in bulk transmission systems*” *Proc. of IEEE 2001 Power Tech Proceedings*, Vol. 2 , Sept 2001.

- [57] B.A. Carreras, V.E. Lynch, and I. Dobson, “*Dynamical and probabilistic approaches to the study of blackout vulnerability of the power transmission grid*”, 2004 Proc. of the 37th Annual Hawaii International Conference on System Sciences, Jan. 2004, pp. 55 – 61.
- [58] J.C. Tan, P.A. Crossley, and P.G. McLaren, “*Application of a wide area backup protection expert system to prevent cascading outages*”, IEEE Trans. Power Delivery, Vol. 17, No. 2, pp. 375 – 380, April 2002.
- [59] D. C. Elizondo, J. de La Ree, A. G. Phadke, and S. Horowitz, “*Hidden failures in protection systems and their impact on wide-area disturbances*”, Proc. of IEEE 2001 PES Winter Meeting, Vol. 2, pp. 710 – 714, Jan/Feb 2001.
- [60] B. Yang, V. Vittal, and G. T. Heydt, “*Slow-coherency-based controlled islanding--a demonstration of the approach on the August 14, 2003 blackout scenario*”, IEEE Trans. Power Systems, Vol. 21, No. 4, pp. 1840-1847, Nov. 2006.
- [61] H. Song, and M. Kezunovic, “*Static Security Analysis based on Vulnerability Index (VI) and Network Contribution Factor (NCF) Method*”, 2005 IEEE PES T&D Asia Pacific, Dalian, China, August, 2005.
- [62] H. Song, and M. Kezunovic, “*A New Analysis Method for Early Detection and Prevention of Cascading Events*”, Electric Power Systems Research, Vol. 77, Issue 8, Pages 1132-1142, June 2007.
- [63] M. Kezunovic, and C. Pang, “*Improved Transmission Line Protection During Cascading Events*”, CIGRE B5 Colloquium, Madrid, Spain, October 2007.
- [64] N. Zhang, and M. Kezunovic, “*A Real Time Fault Analysis Tool for Monitoring Operation of Transmission Line Protective Relay*”, Electric Power Systems Research, Vol. 77, No. 3-4, pp. 361-370, March 2007.
- [65] Nan Zhang, and M. Kezunovic, “*Coordinating fuzzy ART neural networks to improve transmission line fault detection and classification*”, IEEE PES General Meeting, San Francisco, June 2005.
- [66] J. Bialek, “*Topological generation and load distribution factors for supplement charge allocation in transmission open access*”, IEEE Trans. Power Systems, Vol. 12, No. 3, pp. 1185–1193, Aug. 1997.
- [67] V. Vittal, E.Z. Zhou, C. Hwang, and A.-A. Fouad, “*Derivation of stability limits using analytical sensitivity of the transient energy margin*”, IEEE Trans. Power Systems, Vol. 4, No. 4, pp. 1363–1372, Nov. 1989.
- [68] M. Kezunovic, X. Luo, N. Zhang, and H. Song, “*Testing and Evaluating New Software Solutions for Automated Analysis of Protective Relay Operations*”, The 7th International Conference on Power Systems Transients, Lyon, France, June 2007.

Part II

Application of PMU Measurements in the Control Room

Thomas. J. Overbye
University of Illinois at Urbana-Champaign

Illinois Research Team Graduate Students
Charles (Matt) Davis
Joseph (Zeb) Tate (now with University of Toronto)

Information about this project

For information about this project contact:

Tom Overbye
University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering
1406 W. Green St.
Urbana, IL 61801
Phone: 217-333-4463
Fax: 217-333-1162
Email: overbye@ece.uiuc.edu

Power Systems Engineering Research Center

The Power Systems Engineering Research Center (PSERC) is a multi-university Center conducting research on challenges facing the electric power industry and educating the next generation of power engineers. More information about PSERC can be found at the Center's website: <http://www.pserc.org>.

For additional information, contact:

Power Systems Engineering Research Center
Arizona State University
577 Engineering Research Center
Tempe, Arizona 85287-5706
Phone: 480-965-1643
Fax: 480-965-0745

Notice Concerning Copyright Material

PSERC members are given permission to copy without fee all or part of this publication for internal use if appropriate attribution is given to this document as the source material. This report is available for downloading from the PSERC website.

© 2008 University of Illinois at Urbana/Champaign. All rights reserved.

Table of Contents

1	INTRODUCTION.....	1
1.1	Motivation.....	1
1.2	Literature Review	2
1.3	Report Overview	5
2	FILTERING OF PMU MEASUREMENTS TO EXTRACT STEADY STATE VALUES	8
2.1	FIR filtering of measurements.....	8
2.2	Median filtering of measurements.....	13
2.3	Comparing FIR and median filtering	15
2.4	Filter evaluation	19
2.4.1	Step change with no oscillations.....	20
2.4.2	Step change with 1 Hz oscillations	21
2.4.3	Step change with variable oscillations.....	23
2.4.4	Real PMU signal	26
2.5	Conclusions.....	29
3	DETECTION OF EVENTS USING PMU DATA.....	31
3.1	Event detection and determination of the angle change vector	31
3.1.1	Basic procedure.....	31
3.1.2	Parameter selection	34
3.2	Single line outage detection	39
3.2.1	Analytical basis for single line outage detection	39
3.2.2	Basic single-line outage event detection algorithm	43
3.2.3	Computational complexity.....	44
3.2.4	Heuristic modifications to the basic algorithm	44
3.3	Generator outage detection.....	44
3.3.1	Analytical basis for generator outage detection.....	44
3.3.2	Verification of droop-based participation factors	48
3.3.3	Basic generator outage detection algorithm.....	53
3.3.4	Computational complexity	53
3.3.5	Heuristic modifications to the basic algorithm	54
3.4	Double line outage detection	54
3.4.1	Analytical basis for double line outage detection	54
3.4.2	Basic double line outage event definition algorithm	59
3.4.3	Computational complexity.....	60

Table of Contents (continued)

3.4.4	Heuristic modifications to the basic algorithm	60
4	EVENT DETECTION EVALUATION.....	65
4.1	Single line outage examples.....	65
4.1.1	All 37 bus single line outages	65
4.1.2	TVA system line outage.....	82
4.2	Generator outage examples.....	86
4.2.1	All buses monitored	87
4.2.2	Eighteen buses monitored	92
4.3	Double line outage examples.....	95
4.3.1	All non-islanding double line outages	95
4.3.2	All non-islanding double line outages that share at least one terminal..	113
5	PHASOR MEASUREMENT UNIT PLACEMENT FOR EVENT DETECTION	117
5.1	Overview	117
5.2	Objective function definition	117
5.2.1	Objective 1: Maximize the number of correctly ranked events.....	117
5.2.2	Objective 2: Maximize the number of correctly ranked events with <i>no</i> misranked events.....	118
5.2.3	Objective 3: Maximize the number of correctly ranked events with <i>no</i> misranked events and <i>no</i> events with the same rank	119
5.3	Searching methods.....	120
5.3.1	Exhaustive search.....	120
5.3.2	Simulated annealing.....	120
5.4	Results for optimal PMU placement to detect single line outages.....	123
5.4.1	Three PMUs placed by exhaustive search and simulated annealing	123
5.4.2	Six PMUs placed by exhaustive search and simulated annealing	124
5.4.3	Higher numbers of PMUs placed by simulated annealing.....	126
6	GPU-BASED VISUALIZATION OF PMU DATA.....	127
6.1	Overview	127
6.2	Implementation of contouring on the GPU	127
6.2.1	Introduction to contouring on the GPU	127
6.2.2	Contouring background	128
6.2.3	GPU programming[83]	130

Table of Contents (continued)

6.2.4	Algorithm definition and implementation details	132
6.2.5	GPU-based contouring algorithm performance results.....	134
6.2.6	Advantages and disadvantages of GPU-based contouring versus CPU-based contouring	141
6.3	Benefits and applications of accelerated contouring	142
6.3.1	Improvements in usability.....	142
6.3.2	Visualizing faster data.....	144
7	CONCLUSIONS AND FUTURE WORK	147
	REFERENCES.....	149
	APPENDIX A 37 BUS SYSTEM DESCRIPTION	158

List of Figures

Figure 1.1: Determining the angle changes on the system resulting from an event	6
Figure 2.1: Responses of 60-order, low-pass, 0.1 Hz FIR filters on a PMU angle signal using rectangular, Hamming, and Blackman windows	10
Figure 2.2: Magnitude response of 0.1 Hz, 60-order, low-pass FIR filters with three different window types	11
Figure 2.3: Effect of a 0.5 Hz, Hamming-window, low-pass FIR filter on a PMU angle signal with 30, 60, and 90 filter orders	12
Figure 2.4: Illustration of the delay introduced by using a causal median filter with window length 21 for a step change input signal.....	14
Figure 2.5: Effect of a median filter on a PMU angle signal with window sizes of 31, 61, and 91.....	15
Figure 2.6: Values used to calculate percent overshoot (<i>PO</i>).....	17
Figure 2.7: Values used to calculate rise samples (<i>RS</i>)	18
Figure 2.8: Values used to calculate delay samples (<i>DS</i>)	19
Figure 2.9: <i>RS</i> and <i>DS</i> values for FIR and median filters with filter orders / window sizes in the range of 1-100 and 1-99 applied to a unit step change.	20
Figure 2.10: Unfiltered test signal used to test filter responses for a step change with 1 Hz oscillations after the step change.....	21
Figure 2.11: <i>RS</i> and <i>DS</i> values for FIR and median filters with filter orders / window sizes in the range of 1-100 and 1-99 applied to a unit step change with post-step oscillations of 1 Hz.....	22
Figure 2.12: <i>PO</i> values for FIR and median filters with filter orders / window sizes in the range of 1-100 and 1-99 applied to a unit step change with post-step oscillations of 1 Hz.	23
Figure 2.13: <i>RS</i> and <i>DS</i> values for FIR filters with filter orders in the range of 1-99 applied to a unit step change with post-step oscillations of 0.1-15 Hz.....	24
Figure 2.14: <i>RS</i> and <i>DS</i> values for median filters with window lengths in the range of 1-99 applied to a unit step change with post-step oscillations of 0.1-15 Hz.	24
Figure 2.15: <i>PO</i> values for FIR filters with filter orders in the range of 1-15 applied to a unit step change with post-step oscillations of 0.1-15 Hz.....	25
Figure 2.16: <i>PO</i> values for median filters with window lengths in the range of 1-15 applied to a unit step change with post-step oscillations of 0.1-15 Hz.....	25
Figure 2.17: $PODS_{x\%}^{FIR} - PODS_{x\%}^{median}$ based on unit step and oscillations signals with frequencies of 0.1-15 Hz	26

List of Figures (continued)

Figure 2.18: Real PMU angle signal used to test filters	27
Figure 2.19: RS and DS values for median and FIR filtering of a real PMU angle signal	28
Figure 2.20: PO values for median and FIR filtering of a real PMU angle signal	29
Figure 3.1: Definition of N_{trans} , which is used to generate candidate signals $\Delta\theta_{i,candidate}[n]$..	31
Figure 3.2: Peak detection and determination of the angle change vector	32
Figure 3.3: Illustration of the relationship between N_{trans} and $\Delta\theta_{observed}$	34
Figure 3.4: Simulated PMU angle signal used to evaluate edge detection parameters	36
Figure 3.5: $\Delta\theta_{miss}$ and n_{while} values using 61-order FIR and median filtering as N_{trans} values range from 1 to 500 samples	37
Figure 3.6: n_{while} values using a filter length of 147 FIR and median filtering as N_{trans} values range from 1 to 250 samples	38
Figure 3.7: Filtering signals obtained using FIR and median filtering of length 147	38
Figure 3.8: Normalized angle difference (NAD) metric	43
Figure 3.9: Illustration of droop characteristics	46
Figure 3.10: Evolution of frequency difference f_{diff} over time for each generator outage...	50
Figure 3.11: Evolution of PFD values over time for each generator outage	51
Figure 3.12: Double-line outage model with power injections	55
Figure 3.13: Modeling the outage of two parallel lines with a power transfer	58
Figure 3.14: Distance between two lines, as defined in (3.48)	62
Figure 3.15: Minimum distance between two lines, as defined in (3.49)	63
Figure 3.16: System used to test the impact of distance-based double outage filtering	64
Figure 3.17: Percentage of double outage events considered as a function of the distance threshold $d_{threshold}$ using d_{circ} and d_{min} for threshold ranges of 0-1500 miles (a) and 0-5 miles (b)	64
Figure 4.1: Original simulated angles (left side) and simulated PMU angle measurements (right side) for the outage of the line connecting buses 28 and 31, with all angles referenced to the SLACK345 bus angle	66

List of Figures (continued)

Figure 4.2: Minimum threshold values and error values for each line, ordered in ascending order by minimum threshold, using a median filter of length 31	67
Figure 4.3: Mean <i>NADOutaged</i> values for median filtering using a window length of 31 with the angle threshold varied from 0.01 to 0.57 degrees; the left plot shows the right plot zoomed into the region $\tau \in [0.01, 0.05]$	68
Figure 4.4: Calculated changes in angles at all system buses due to the outage of a line connecting buses 18 and 37 using the ac and dc power flow	69
Figure 4.5: Scatter plot of <i>NADOutaged</i> versus <i>RankOutaged</i> for 84,867 detected line outages using window lengths from 3 to 61 in increments of 2 and angle thresholds from 0.01 to 0.57 degrees in increments of 0.01 degrees.	70
Figure 4.6: Effect of median window length on the mean <i>NADOutaged</i> value using an angle threshold of 0.05 degrees	71
Figure 4.7: Elimination of an oscillation in the BLT138 bus angle signal for the outage of line 47-53 as the median window length is changed from 9 to 41	71
Figure 4.8: Minimum threshold values and error values for each line, ordered in ascending order by minimum threshold, using an FIR filter of order 31	72
Figure 4.9: Effect of FIR filter order on the mean <i>NADOutaged</i> value using an angle threshold of 0.05 degrees	73
Figure 4.10: Scatter plot of <i>NADOutaged</i> versus <i>RankOutaged</i> for 86,438 detected line outages using filter orders from 3 to 61 and angle thresholds from 0.01 to 0.57 degrees.	73
Figure 4.11: Scatter plot of <i>NADOutaged</i> versus <i>RankOutaged</i> for 82,289 detected line outages (FIR filtering) and 81,846 detected line outages (median filtering) using filter lengths from 3 to 61 and angle thresholds from 0.01 to 0.57 degrees.....	74
Figure 4.12: Detailed <i>RankOutaged</i> distribution for the outage of line 33-50 using FIR filter orders of 3-61 and angle thresholds of 0.01 to 0.57.....	75
Figure 4.13: Reducing the full system to only buses that are monitored with PMUs	76
Figure 4.14: The changes in the equivalent injections due to the event will sum to zero if there is no change in power injections within the unmonitored system.	77
Figure 4.15: Indistinguishable line outages using the 18 PMU configuration of the 37 bus case.....	80
Figure 4.16: Instance of a radial system (single boundary bus) from the 37 bus case	81
Figure 4.17: Location of PMU angle measurements on TVA system and location of the line outage.....	82

List of Figures (continued)

Figure 4.18: Unfiltered angle measurements from before and after the line outage	83
Figure 4.19: <i>RankOutaged</i> values for detection of the TVA outage event using filter lengths in the range of 3 to 401 and angle thresholds in the range of 0.1 to 0.5 degrees.....	84
Figure 4.20: Differences in noise attenuation for FIR and median filtering for a filter length of 379.....	84
Figure 4.21: <i>EarlyEdge</i> contour plot for FIR (left) and median (right) filtering using filter lengths in the range of 3 to 401 and angle thresholds in the range of 0.1 to 0.5 degrees.....	85
Figure 4.22: <i>RankOutaged</i> values with and without the usage of 8CUMBERL PMU angle measurements.....	86
Figure 4.23: Contour plot of <i>RankOutaged</i> for generator outages at WEBER69, JO345, SLACK345, and LAUF69 using FIR filtering orders of 3 to 301 and angle thresholds of 0.1 to 3.0 degrees	87
Figure 4.24: Contour plot of <i>RankOutaged</i> for generator outages at BOB69, ROGER69, BLT138, and BLT69 using FIR filtering orders of 3 to 301 and angle thresholds of 0.1 to 3.0 degrees.....	88
Figure 4.25: Prevailing oscillations after the outage of a JO345 generator using FIR filtering with filter order 151 and cutoff frequency of 0.1 Hz.	89
Figure 4.26: Effect of FIR low-pass cutoff frequency on the detection and rank of the JO345 generator outage.....	89
Figure 4.27: Contour plot of <i>RankOutaged</i> for generator outages at WEBER69, JO345, SLACK345, and LAUF69 using median filtering with window lengths of 3 to 301 and angle thresholds of 0.1 to 3.0 degrees	90
Figure 4.28: Contour plot of <i>RankOutaged</i> for generator outages at BOB69, ROGER69, BLT138, and BLT69 using median filtering with window lengths of 3 to 301 and angle thresholds of 0.1 to 3.0 degrees	91
Figure 4.29: A detailed look at ranking and detection of the JO345 generator outage using median filtering.....	92
Figure 4.30: Contour plots analogous to those in Figure 4.23 for FIR filtering with eighteen buses monitored instead of full bus monitoring. The WEBER69 outage is undetected and the LAUF69 outage has <i>RankOutaged</i> equal to one over the entire region, so these contours are not shown.	92
Figure 4.31: Contour plots analogous to those in Figure 4.24 for FIR filtering with eighteen buses monitored instead of full bus monitoring.....	93

List of Figures (continued)

Figure 4.32: Contour plots analogous to those in Figure 4.27 for median filtering with eighteen buses monitored instead of full bus monitoring. The WEBER69 outage is undetected and the LAUF69 outage has <i>RankOutaged</i> equal to one over the entire region, so these contours are not shown.	94
Figure 4.33: Contour plots analogous to those in Figure 4.28 for median filtering with eighteen buses monitored instead of full bus monitoring.....	95
Figure 4.34: Ranking of each line-line combination using FIR filter order 31, angle threshold of 0.02 degrees. Green squares indicates double line outages where <i>RankOutaged</i> = 1, red squares indicate <i>RankOutaged</i> > 1 (i.e., misclassification), and white squares indicate outages that were not tested.	97
Figure 4.35: Effect of angle threshold on event detection with 31-order FIR filtering; black squares represent outages that were undetected due to insufficient angle changes.....	98
Figure 4.36: A detailed look at the observed and event matching angle changes due to the outage of the parallel lines between buses 39 and 38.....	100
Figure 4.37: Unmonitored system connected to monitored system through three boundary buses.....	105
Figure 4.38: Portion of the 37 bus system that, when monitored with the 18 PMU set, has eleven indistinguishable double line outages	108
Figure 4.39: Program flow for the double line outage detection implementation, including cascading event classification	110
Figure 4.40: FIR filtered results, using a filter order of 31 and the set of eighteen PMUs defined in section 4.1.1.2	111
Figure 4.41: Median filtered results, using a window length of 31 and the set of eighteen PMUs defined in section 4.1.1.2	112
Figure 4.42: Effect of reducing the PMU set on <i>SharedOutaged</i> with 31-order FIR filtering and τ set to the minimum value needed to detect the maximum number of events.....	113
Figure 4.43: <i>RankOutaged</i> results for FIR filtering with angle threshold of 0.02 degrees, for the full double outage set (left side) and the double outages which share a terminal bus (right side). Red squares indicate where <i>RankOutaged</i> is greater than one and green squares indicate where <i>RankOutaged</i> equals 1. White squares indicate events which were not considered.....	115
Figure 4.44: Effect of reducing the PMU set on <i>SharedOutaged</i> with 31-order FIR and τ set to the minimum value needed to detect all events, with the event set restricted to those lines that share a terminal bus	116

List of Figures (continued)

Figure 6.1: The GPU rendering pipeline.....	129
Figure 6.2: Oneline diagram of contour area	134
Figure 6.3: Contour color map relating per unit voltage to displayed color.....	135
Figure 6.4: Contour rendering with resolution set to 1024x768, radius of influence set to 100136	
Figure 6.5: Effect of contour resolution on contour rendering times	137
Figure 6.6: Effect of fragment culling on radius of influence effects.....	138
Figure 6.7: Effect of influence radius on contour rendering time with contour resolution set to 1024x768.....	140
Figure 6.8: Contour rendering with resolution set to 1024x768, radius of influence set to 25141	
Figure 6.9: Contour of angle changes on the TVA system before and after the line outage.	145
Figure 6.10: Colormap used for angle contours in Figure 6.9	146

List of Tables

Table 3.1: Accuracy of droop-based participation factors using model parameters.....	51
Table 3.2: Droop means and standard deviations determined from simulation	52
Table 3.3: Accuracy of droop-based participation factors using Table 3.2 droop values	52
Table 3.4: Droop means and standard deviations determined from simulation, with WEBER69 droop set to 10% instead of 5%	52
Table 3.5: : Accuracy of droop-based participation factors using Table 3.4 droop values, with WEBER69 droop set to 10% instead of 5%	53
Table 4.1: Algorithm results for lines shown in Figure 4.15 using order 61 FIR filtering and angle threshold of 0.01 degrees.....	80
Table 4.2: Misrankings that do not involve 18-37 or 15-54 using FIR filtering with a filter order of 31 and angle threshold of 0.02 degrees	99
Table 4.3: Summary of algorithm results for FIR (median) filtered angle measurements with complete bus monitoring.	103
Table 4.4: Indistinguishable double line outages for the portion of the 37 bus system illustrated in Figure 4.38.....	109
Table 4.5: Summary of algorithm results for FIR (median) filtered angle measurements with complete bus monitoring considering only double line outages where a terminal bus is shared	114
Table 5.1: Placement of three PMUs using exhaustive search and SA	124
Table 5.2: Placement of six PMUs using exhaustive search and SA.....	125
Table 5.3: Placement of 9, 18, and 27 PMUs by simulated annealing	126
Table 6.1: Algorithm implementation on the CPU and the GPU	132
Table 6.2: Timings for seven contour resolutions with the influence radius set to 100	136
Table 6.3: Timing results for several values of influence radius with a constant contour resolution of 1024x768.....	139

1 Introductionⁱ

1.1 Motivation

With the increasing loading of the power system, along with the massive inter-area transfers enabled by the deregulation of the 80's and 90's, there is a clear need to have reliable information about both the local system and external systems. Tellingly, four of the six major North American blackouts were due in part to a lack of situational awareness [1]. Although there is a clear need for sharing of information, there is limited real-time sharing of SCADA or state estimator information in the United States [2]. However, as phasor measurement units (PMUs) [3] have been deployed throughout the North American power grid, there have been significant efforts to ensure that PMU data is shared between all interested parties [4]. Because PMU data is more widely available in near real-time than other power system measurement data, it can provide unique insights into the global operation of the grid. Therefore the focus of the University of Illinois portion of this project was to investigate the use of this key set of substation-based information.

Extensive research in applying PMU information to improve situational awareness has been conducted since their introduction, including applications in state estimation [5-7], dynamic security assessment [8-10], and visualization [11-13]. Yet another key aspect of situational awareness in the power grid is the knowledge of transmission line, transformer, and generator statuses. Beyond incorporation of PMUs into traditional state estimation, which can include topology estimation [14], there has been little research into how PMUs can be used to enhance topology information, particularly outside of the local area. Current network topology processors focus on obtaining local system topology information through the use of predominantly local area measurements. However, connectivity information over the wide area is very important for system operations and is a major reason for the existence of the NERC System Data Exchange (SDX) [1]. Because there is significant value in improving system operators' knowledge of external system line outages, a method which utilizes PMU data to effect this improvement has been developed.

The method discussed in this work makes use of SDX information, as it is currently the best source of system-wide line status information. One of the most telling indications of its usefulness is its application in processing Transmission Loading Reliefs (TLRs) on the North American power grid. However, despite increased awareness of the importance of inter-area

ⁱ This report is based substantially on the Ph.D. Thesis authored by project team member Joseph (Zeb) Tate titled, "Event Detection and Visualization Based on Phasor Measurement Units for Improved Situational Awareness," University of Illinois at Urbana-Champaign, July 2008. Dr. Tate is now an Assistant Professor at University of Toronto.

information exchange since the August 2003 blackout, updates to the SDX are still only required on an hourly basis [15]. Because much can happen on the power grid within an hour, there is a need for tools which are capable of providing more current information on external system outages. This work looks at ways to improve upon the status information from local topology processors and the NERC SDX by incorporating PMU data which, like SDX information, is also available over the entire interconnect through phasor data concentrators (PDCs) [4]. Although the methods presented here would work with any source of synchronized angle data, PMUs are the only devices which are deployed over the entire power grid and are capable of providing geographically dispersed, synchronized, and accurate phasor angle measurements.

The processing needed to extract steady-state angle information from PMUs is the first topic of discussion, and the advantages and disadvantages of two types of digital filters—finite impulse response and median—are investigated. The usefulness of each filtering method is demonstrated with a broad spectrum of signals, both real and simulated, which are meant to be representative of typical power system signals. This process is necessary to convert the raw angle measurements into a form which can be used with traditional steady-state analysis tools such as the power flow. The intended application of this filtering, system event identification is then presented in a general form, with specific algorithms defined for detecting single line, double line, and generator outages. Test results for each of these event classes are provided to demonstrate the efficacy of the proposed event detection algorithms using both real and simulated data. Several optimization problems are also defined to guide placement of PMUs for event detection, and results are presented based on usage of exhaustive search and simulated annealing to perform the placement optimization.

In addition to the new methods presented for the processing of PMU data, techniques for visualization of PMU data in its processed and unprocessed forms are described. A new technique based on graphical processing units (GPUs), developed to allow rendering speeds to match the relatively fast data rate of PMUs, is presented, along with results demonstrating the marked increase in visualization speed. Some of the benefits of speeding up contouring are discussed, including how PMU derivative information can be visualized with GPU-based rendering.

1.2 Literature Review

The basic operations needed for PMUs to work, including relevant filtering and signal processing techniques, were first described in 1983 [16] and the first prototype PMU was developed in 1988 [17]. Shortly after these new measurement devices were described, PMU applications related to situational awareness began to be researched. Initial applications using PMUs were primarily focused on improvements to state estimation based on the ability to directly measure system states [7]. A key aspect of this initial research into PMU usage was the need for traditional SCADA measurements (such as breaker status, voltage magnitude, and current magnitude) to accompany PMU measurements in order to use the new data. The

problem with this dependence on SCADA measurements is that, outside of the local control area, SCADA measurements are usually unavailable. One way in which researchers have attempted to get around this dependence on SCADA measurements has been to determine how complete coverage of the power system could be obtained using only PMU measurements. The key paper on this topic demonstrates that complete observability could be obtained with as few as one fourth to one third of the system buses having PMUs [18]. Unfortunately, current coverage of PMUs is still well below this number (e.g., for the Eastern Interconnect, only 60 PMUs are currently online [19]). Additional work in limited placement of PMUs for incomplete observability fails to provide applications in which this partial observability would be useful beyond incorporation into traditional state estimators [20]. Besides applications in state estimation, additional research has been performed in the usage of PMUs for system monitoring. Monitoring of power system harmonics based on PMU data is discussed in [21] and [22], although there is limited discussion of how PMU location and filtering can affect the applicability of this monitoring task for improving operator awareness. Fault detection and classification based on highly localized PMU measurements has also been researched. Reference [23] provides an overview of the different methods proposed in this area and notes that all of the currently developed methods for fault detection require PMUs to be located at one or more of the terminal buses of the faulted transmission line. Finally, direct visualizations of phasor angles for system monitoring have been developed [10-12], although there is little guidance in how the visualized angles should be interpreted by system operators.

The notion that PMU measurements can be applied in more problem domains, particularly event detection and classification, is supported by the literature. The close correspondence between PMU angle measurements and correct steady-state phasor angles on the system has been demonstrated in numerous studies [24-26]. In addition, the IEEE standard governing PMU accuracy requires that the total vectorized error (TVE), defined as the normalized Euclidean distance between the true and measured steady-state phasors, be within 1% [27]. Although it cannot be assumed that SCADA and state estimator data is available over the entire interconnect, some additional wide area information, such as that available through the NERC System Data Exchange (SDX) [28], can also be used for interpretation of PMU data due to the system-wide availability of this information. Usage of PMU measurements for comprehensive dynamic information is less promising, mainly due to the lack of a unifying standard on PMU behavior during transient conditions.

Event detection and classification can be divided into three key components: detecting when an event has occurred, extracting salient information about the event from the raw data, and classifying the event based on this information. For the first two tasks, detection and information extraction, the largest body of research is in the area of image processing due to the need for accurate edge detectors in many applications. Edge detectors are essentially step change detectors and can be used to return not only step change locations but also the amplitude of step changes. An excellent survey of classic edge detection techniques is provided in [29]. A widely used “hill climbing” method used to accurately detect edges which involve ramp rather

than step changes is detailed in [30] and serves as the basis for the edge detection techniques described later in the dissertation. Moving beyond linear filters, nonlinear filtering for edge detection, in particular median filtering, has been shown to outperform linear filtering for certain image processing [31] and power electronics applications [32]. However, there are two key issues which make nonlinear filters less appealing—the potentially higher computational burden relative to linear filters, and the inability to determine the filter’s performance a priori. One commonality amongst all edge detectors surveyed is the domain specific length of the filter used. In all cases, analysis of representative data must be conducted to determine the most appropriate filter length for a given application [30]. In addition to the image processing literature, there has also been extensive research into event detection, feature extraction, and event classification in power systems, primarily centered on fault detection and classification. Modern methods of solving this problem rely on techniques from the field of artificial intelligence (AI), such as neural networks [33] and expert systems [34]. There has also been significant work in detecting power quality disturbances using various data processing techniques, including wavelet analysis, expert systems, neural networks, and genetic algorithms [35]. Although these methods could be used for detecting changes and extracting information about the changes in system states, adapting these techniques to wide area event classification is not feasible due to the unavailability of quality training data. In the area of large-scale, steady-state event detection, the primary tools currently used are topology processing and state estimation, relying almost exclusively on traditional power system data [36]. One notable exception is [37], which provides a method for direct determination of external system outages based on expected correlations between changes in internal system states and changes in boundary flows.

Power system visualization was first discussed in the context of visualizing load flow solutions using the IBM System/360 mainframe computer [38]. Additional work in visualization techniques continued throughout the 1970’s, 80’s, and 90’s; reference [39] provides a concise overview of state-of-the-art visualization techniques developed up to the mid-1990’s. There has since been substantial research and deployment of system visualizations throughout control centers with a focus on increasing situational awareness of grid operators [40, 41]. In addition, formal evaluation has shown the potential benefits of several visualization techniques such as voltage contouring and network flow animation [42, 43]. One key similarity amongst all currently used visualization methods is the reliance on the CPU to perform any necessary calculations. This has been sufficient in the past because rendering times have only had to keep up with the relatively slow rates of SCADA and state estimator data, typically on the order of seconds or minutes. Because PMU measurements provide data at rates which are an order of magnitude faster, CPU-based techniques will be difficult to use if real-time visualization is desired. Taking as one example voltage contouring, one of the fastest techniques currently used for CPU-based voltage contouring requires 0.63 seconds to render a contour of a medium-sized system [44]; however, real-time visualization of PMU data, which has a data rate of 30 frames per second, would require rendering times to be shorter than 0.033 (1/30) seconds. It has been shown in numerous applications that one way of achieving significant visualization speedups is

through the usage of programmable GPUs. The promise of improved graphical performance through the utilization of GPUs is based on the massive computation power of modern GPUs—for instance, some of the latest GPUs such as the GeForce GTX 280 have theoretical processing speeds in excess of 900 billion floating point operations per second (Gflops) which is much higher than modern CPUs which are only capable of around 30 Gflops [45]. References [46] and [47] provide a broad survey of applications in which GPUs have provided significant acceleration of visualization techniques. The use of GPUs to speed up scattered data interpolation, the more formal name for the contouring techniques used in power systems, of three- and four-dimensional data is presented in [48]; one of the key results is that interpolating data on a 128^3 grid, with 350,000 data points, requires only 0.328 seconds. In addition, it was found that using GPU-based data interpolation can facilitate real-time updating of data-driven displays at frames rates of approximately 10 frames per second (fps). Using power system visualizations to illustrate data trends, as suggested in [12], would also require significant reductions in rendering times to obtain frame rates which are still perceived as responsive [49].

1.3 Report Overview

PMUs provide detailed angle information that is sampled simultaneously at each measured bus. This capability allows for extremely fast detection of changes in system angles, which in turn can provide information about events occurring on the system. Power system events can be modeled as a change in power injection at the system buses, which can in turn be related to changes in angles through the power flow equations. Figure 1.1 illustrates how an event on the system is mapped into a change in angles through a combination of event modeling and application of the power flow equations.

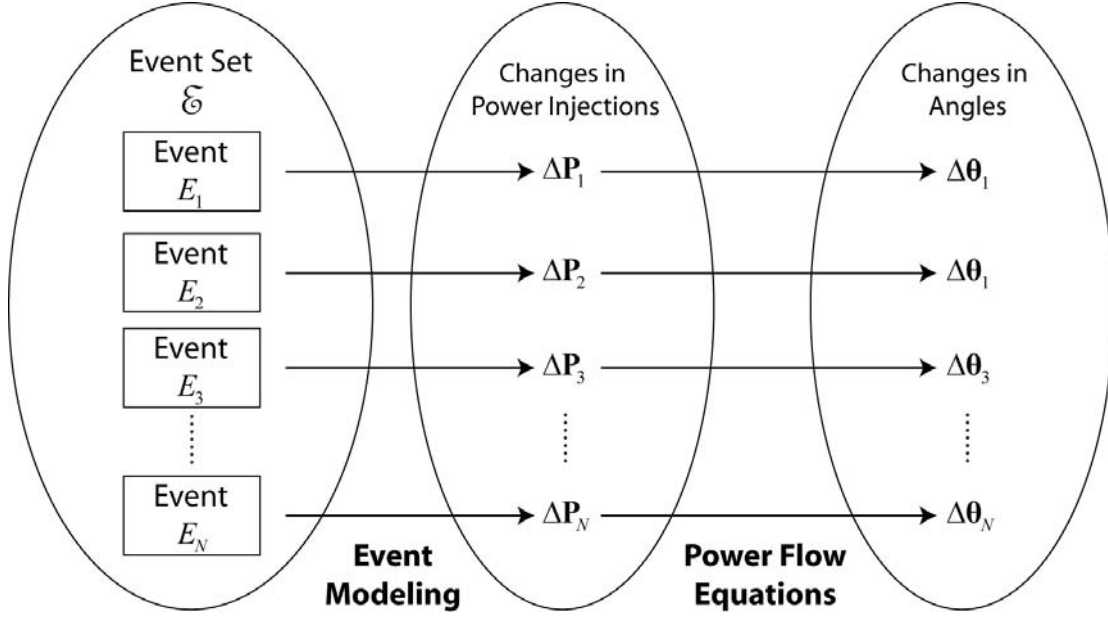


Figure 1.1: Determining the angle changes on the system resulting from an event

For each event E from the event set $\mathcal{E}$, a vector $\Delta \mathbf{P}$ is constructed which models the event as a set of change in power injections on the system. The construction of this $\Delta \mathbf{P}$ vector varies based on the type of event being considered and is discussed in more detail below within the sections dealing with each event type. Once the $\Delta \mathbf{P}$ vector is determined, the power flow equations are then used to determine the associated changes in angles on the system, $\Delta \boldsymbol{\theta}$. Either the ac or dc power flow equations can be used to perform the mapping from $\Delta \mathbf{P}$ to $\Delta \boldsymbol{\theta}$, and the choice between the two is generally a tradeoff between accuracy and computation time [50]. From a performance standpoint, the ac power flow solution is slower than the dc power flow solution but provides higher solution accuracy. In addition, the ac power flow equations require the full state of the system to be known. On the other hand, utilization of the dc power flow equations only requires knowledge of the system topology and line parameters. Because complete state information is not available over the entire interconnect, the dc power flow equations are used in this work. One form of the dc power flow equations is [50]:

$$\Delta \boldsymbol{\theta} = \mathbf{B}^{-1} \Delta \mathbf{P}$$

$$B_{ik} = -\frac{1}{X_{ik}}, k \neq i \quad (1.1)$$

$$B_{ii} = \sum_{i \omega k} \frac{1}{X_{ik}}$$

where “ $i \omega k$ ” indicates the existence of a branch between the two buses i and k , and X_{ik} is the reactance of the branch connecting buses i and k .

Event detection can be considered the inverse of the procedure shown in Figure 1.1, i.e., determination of the event which resulted in an observed set of angle changes on the system. A formal definition of the problem is:

$$\begin{aligned}
E^* &= \arg \min_{E \in \mathcal{E}} \|\Delta \boldsymbol{\theta}_{\text{observed}} - \mathbf{f}(E)\| \\
\Delta \boldsymbol{\theta}_{\text{observed}} &\in \mathbb{R}^K \\
\mathbf{f} : \mathcal{E} &\rightarrow \mathbb{R}^K
\end{aligned} \tag{1.2}$$

where K is the number of PMU-observable buses on the system. The mapping $\mathbf{f}$ is represented in Figure 1.1 as the transformation from an event E to a change in angles $\Delta \theta$.

Because $\|\Delta \boldsymbol{\theta}_{\text{observed}} - \mathbf{f}(E)\|$ in (1.2) is calculated independently for each event, it is possible to separate the event set into C different classes of events and minimize over each individual class of events:

$$\begin{aligned}
E_1^* &= \arg \min_{E \in \mathcal{E}_1} \|\Delta \boldsymbol{\theta}_{\text{observed}} - \mathbf{f}(E)\| \\
E_2^* &= \arg \min_{E \in \mathcal{E}_2} \|\Delta \boldsymbol{\theta}_{\text{observed}} - \mathbf{f}(E)\| \\
&\vdots \\
E_C^* &= \arg \min_{E \in \mathcal{E}_C} \|\Delta \boldsymbol{\theta}_{\text{observed}} - \mathbf{f}(E)\|
\end{aligned} \tag{1.3}$$

The event detection algorithm of (1.2) can then be broken down into the following steps:

1. Determine $\Delta \boldsymbol{\theta}_{\text{observed}}$
2. For each class of events c
 - a. Find the minimizing event E_c^* according to (1.3)

Chapter 2 and section 3.1 provide details on how step 1 is performed. The majority of chapter 3 is devoted to describing in detail how step 2.a is performed for single line outage, generator outage, and double line outage event classes. Chapter 4 provides results and analysis for outages on a 37-bus study system (using simulated data) and the Tennessee Valley Authority system (using real PMU measurements). Chapter 5 looks at several different objective functions and optimization methods which can be used to place PMUs for event detection. Chapter 6 describes a new GPU-based contouring method along with the advantages in terms of usability and applications. Finally, Chapter 7 presents conclusions along with several potential avenues of future research.

2 Filtering of PMU Measurements to Extract Steady State Values

In order to evaluate the possibility of an event having occurred on the system, it is first necessary to determine the quasi-steady state changes in measured phasor angles, $\Delta\theta_{observed}$. The phasor angle measurements at bus i are referred to as $\theta_i[n]$, where n is the n th sample of the phasor angle. Because only the quasi-steady state angle values are of interest, rather than the total dynamic response, filtering must be applied to the signal. The key objectives for the filtering process are:

1. Eliminate noise and oscillations
2. Maintain step changes due to switching of devices
3. Minimize the amount of delay between an event occurring and the extraction of the angle changes on the system

Two filtering methods are evaluated for the purposes of event detection—finite impulse response (FIR) linear filtering and median filtering. Each of these has its own advantages and disadvantages which are discussed in detail below. In addition, a method of quantitatively evaluating filter performance is presented, and the different filtering methods are evaluating for a range of real and simulated PMU signals.

2.1 FIR filtering of measurements

FIR filtering is one of the most common types of digital signal processing used, due primarily to its stability and linear phase characteristics. In addition, this type of filter has shown promise in other applications related to PMU signal processing [51-54]. A general N order FIR filter takes the following form:

$$x_{filt}^{FIR}[n] = \sum_{i=0}^N \alpha_i x[n-i] \quad (2.1)$$

FIR filtering was chosen rather than infinite impulse response (IIR) filtering, the other common linear filtering method, for several reasons. First, FIR filters are guaranteed to be BIBO stable because there is no feedback in the system. This can be an important feature for this application, particularly if measurement noise cannot be bounded ahead of time. For example, impulse noise has been observed in real PMU data, and BIBO stability is necessary to ensure impulse noise does not have a disastrous effect on the output signal. A second desirable quality of FIR filters is their linear phase characteristic. Maintaining edge information is crucial in detecting where angle changes have occurred, along with the magnitude of the angle changes. Considering the Fourier transform of a unit step function [55]:

$$\frac{1}{1 - e^{-j\omega}} + \sum_{k=-\infty}^{\infty} \pi \delta(\omega + 2\pi k) \quad (2.2)$$

non-uniform displacement of different frequency components could lead to difficulties in detecting angle changes and properly quantifying the magnitude of the changes.

FIR filters do suffer from some shortcomings relative to IIR filters. FIR filters typically require higher orders in order to obtain similar frequency response to an IIR filter. This can lead to increased delay between the occurrence of an event and the detection of the event. In addition, the lower order of IIR filters results in fewer computations, although the typical PMU sampling rate of 30 samples per second is orders of magnitude slower than modern processors which operate in the MHz to GHz range. Design of FIR filters is also more complex than IIR filter design, primarily due to the lack of closed-form design equations for FIR filters [55].

Because closed form design equations do not exist for FIR filters, several different design techniques are typically employed depending on the application. The two methods most commonly used in FIR filter design are window-based methods and optimization-based methods. Window-based FIR filter design is conceptually and computationally simple, with the following three basic design stages [55]:

1. Specify the ideal response within the frequency domain.
2. Determine the corresponding impulse response via the inverse Fourier transform.
3. Apply a window to determine the causal FIR coefficients which approximate the ideal impulse response.

This application requires low-pass filtering in order to attenuate undesired oscillations, leading to three design parameters for a window-based FIR filter design: the window type, the low-pass cutoff frequency, and the filter order.

In the literature, both Blackman [52] and Hamming [54] windows have been used to process PMU data. The Blackman window is characterized by a wide transition band with decaying sidelobes, while the Hamming window has a sharper transition band with constant sidelobe amplitudes [55]. These two windows, along with the simpler rectangular window (which has the narrowest transition band but the largest sidelobe amplitudes), were tested with a simulated PMU angle signal sampled at 30 samples per second for a FIR low-pass filter design having a cutoff frequency of 0.1 Hz. Figure 2.1 shows the results of applying each of these windows to sample PMU data; clearly, the Hamming window provides the best combination of oscillation attenuation and delay minimization. Figure 2.2 provides the magnitude response of each of the different filters used in Figure 2.1; the filter based on the Hamming window provides a good compromise between the responses of the rectangular window-based filter, which has poorly attenuated sidebands, and the Blackman window-based filter, which has a very wide transition band.

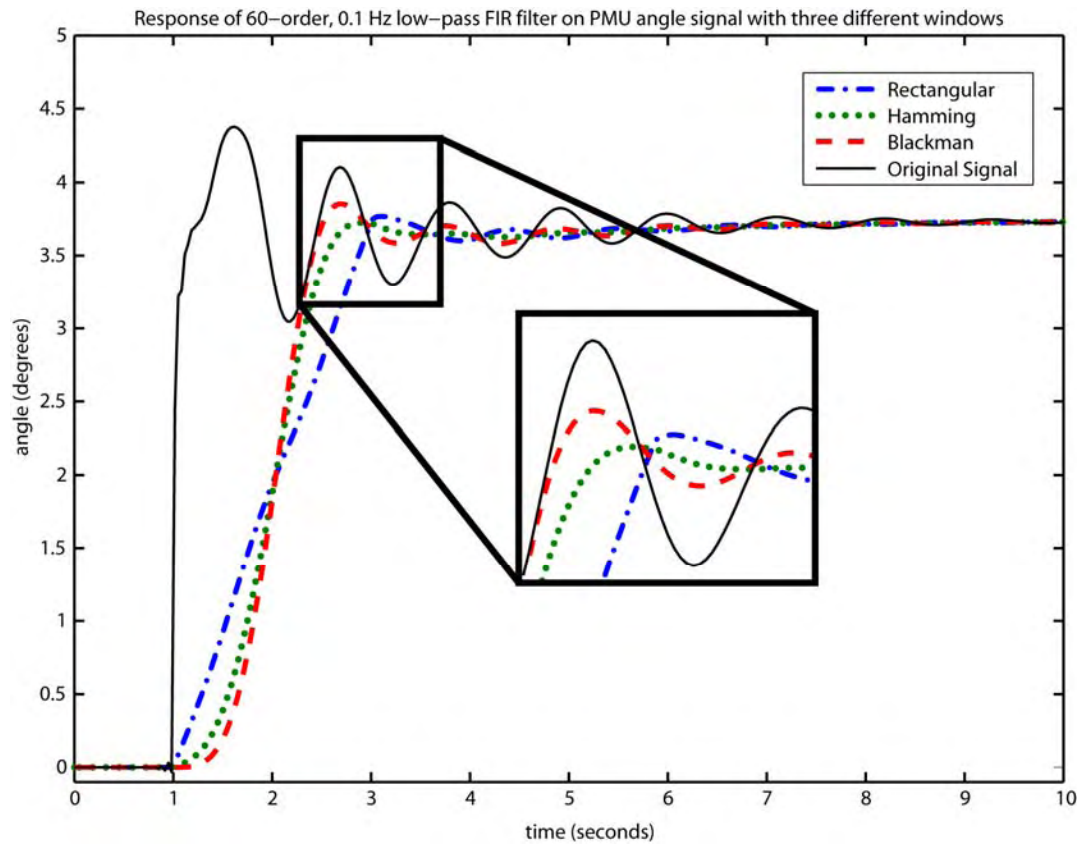


Figure 2.1: Responses of 60-order, low-pass, 0.1 Hz FIR filters on a PMU angle signal using rectangular, Hamming, and Blackman windows

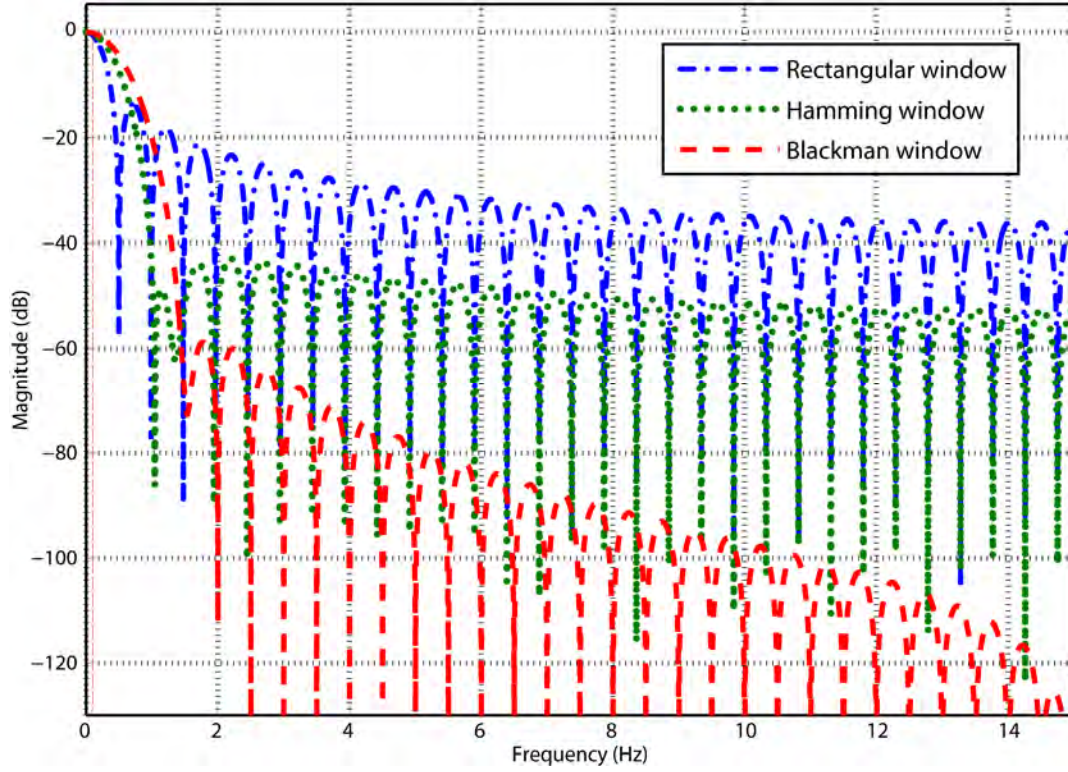


Figure 2.2: Magnitude response of 0.1 Hz, 60-order, low-pass FIR filters with three different window types

The choice of cutoff frequency is based on the lowest frequency anticipated in the angle oscillations after a system event occurs. Typical power system oscillations at the 15 Hz frequency (the Nyquist frequency for the a PMU data rate of 30 samples per second) and below can be divided into two categories: local modes in the frequency range of 0.7 to 2.0 Hz and inter-area modes in the frequency range of 0.1 to 0.8 Hz [56]. The frequencies of the inter-area modes on a power system are a function of many factors, including generator excitation systems, load characteristics, and typical system operating conditions [57]. To eliminate as much of these low-frequency oscillations as possible, the low-pass cutoff frequency was set to 0.1 Hz in the FIR filter design. The cutoff frequency could also be tailored to reflect the expected minimum frequency of electromechanical oscillations on a given system, e.g., 0.5 Hz for the Eastern Interconnect [57] and 0.1 Hz for the Western Interconnect [58].

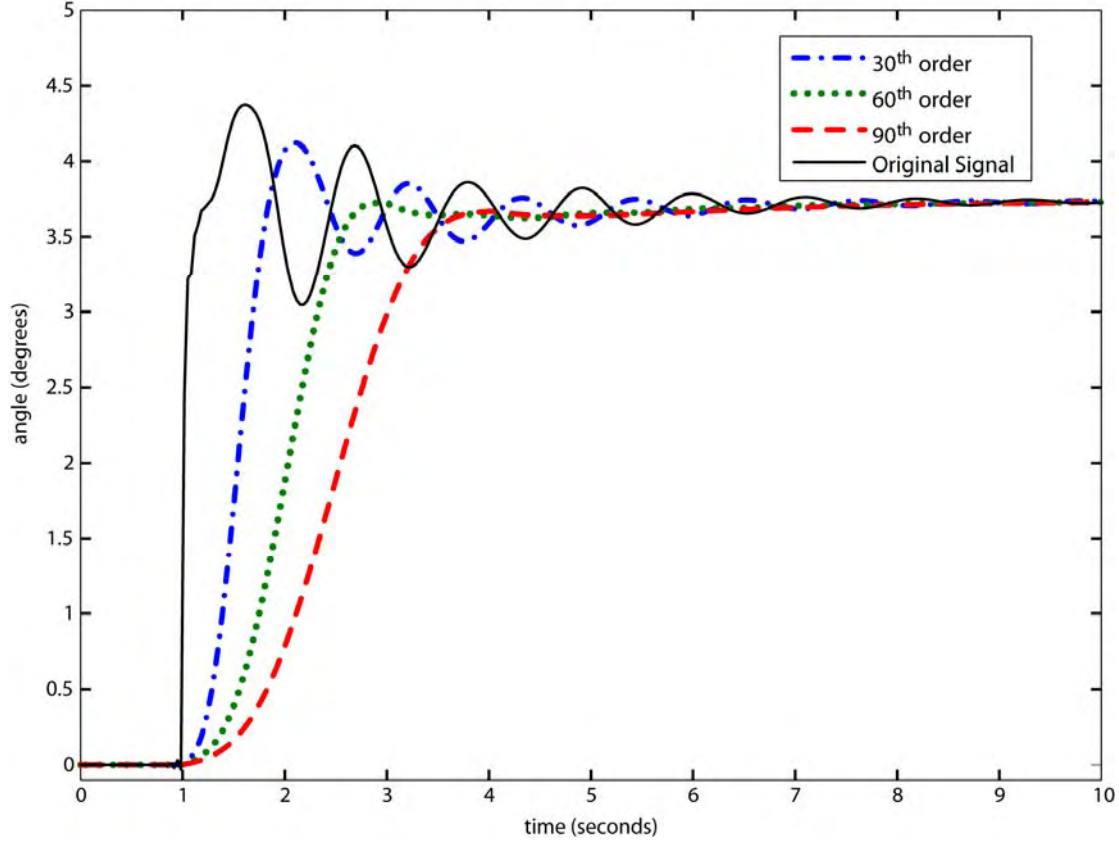


Figure 2.3: Effect of a 0.5 Hz, Hamming-window, low-pass FIR filter on a PMU angle signal with 30, 60, and 90 filter orders

The last design parameter, filter order, serves as a tradeoff between delay in detection of the event (which, for a symmetric FIR filter, is equal to the filter order minus one divided by two) and attenuation of undesired frequency components. In Figure 2.3, the same signal from Figure 2.1 was filtered with three FIR, Hamming-window filters using filter orders of 30, 60, and 90. The 60-order filter provides better attenuation of the initial overshoot compared to the 30-order filter and has roughly the same attenuation capabilities as the 90-order filter but with reduced delay. Analysis of real and simulated data has shown that using a filter of order 61 is a good tradeoff between delay (1 second, for a 30 samples per second signal) and attenuation. In addition, chapter 4 provides a detailed look at how a broad range of FIR filter orders affect the ability to detect events.

Because the goal is to provide a real-time application for use in control centers, the computational complexity of any filtering operation must be taken into account. This provides another bound on the filter order, although, as shown above, the delay and attenuation characteristics are the primary factors used in choosing the desired filter order. For an N order FIR filter, the calculation of each element of the output signal x_{filt}^{FIR} requires N multiplications and N adds (also known as a multiply and accumulate (MAC) sequence [59]), according to (2.1).

Therefore, the order of the FIR filter implementation, using “big-O” notation, is $O(N)$. In terms of storage requirements, only the last N data points must be stored in order to compute the corresponding output point; therefore, storage is also $O(N)$.

2.2 Median filtering of measurements

Because much is known about the typical frequency characteristics of power system signals, and FIR linear filters are designed using explicit frequency response characteristics, FIR filters provide a good way to attenuate unwanted oscillations (criterion 1). However, linear filters in general do a poor job of maintaining step changes in signals (criterion 2). Instead of maintaining the sharp transition, linear filters tend to convert the step change into a ramp [32], and this effect can cause a delay in the determination of the angle change vector (criterion 3). To maintain step changes and reduce delay, median filtering provides a possible alternative to FIR filtering. In addition, the usage of median filtering to smooth out signals in power electronics control applications [32], FNET monitoring [60], and PMU calculations [61] indicates that this filter type can be a viable alternative to linear filtering.

The basic idea of a median filter is to run a sliding window over the data samples, then use the median value from each window as the output value. The typical form used to describe the median filter is [32]:

$$x_{filt}^{Med}[n'] = \text{median of } \{x[n' - k], \dots, x[n'], \dots, x[n' + k]\} \quad (2.3)$$

for a median filter of length $N = 2k + 1$. However, this form of the filter is noncausal because the output samples are dependent on later samples of the input data. To convert this into a causal filter, the substitution $n = n' + k$ is made, resulting in the following causal median filter:

$$x_{filt}^{Med}[n] = \text{median of } \{x[n - N + 1], x[n - N + 2], \dots, x[n]\} \quad (2.4)$$

By converting from the noncausal form of (2.3) to the causal form (2.4), a delay of $\frac{N-1}{2}$ samples is introduced into the output signal. This delay effect is illustrated in Figure 2.4 for a step change input signal, with the step change occurring at sample 50. The dependence of the delay on the filter length indicates that lowering the filter length improves the ability of the median filter to satisfy criterion 3.

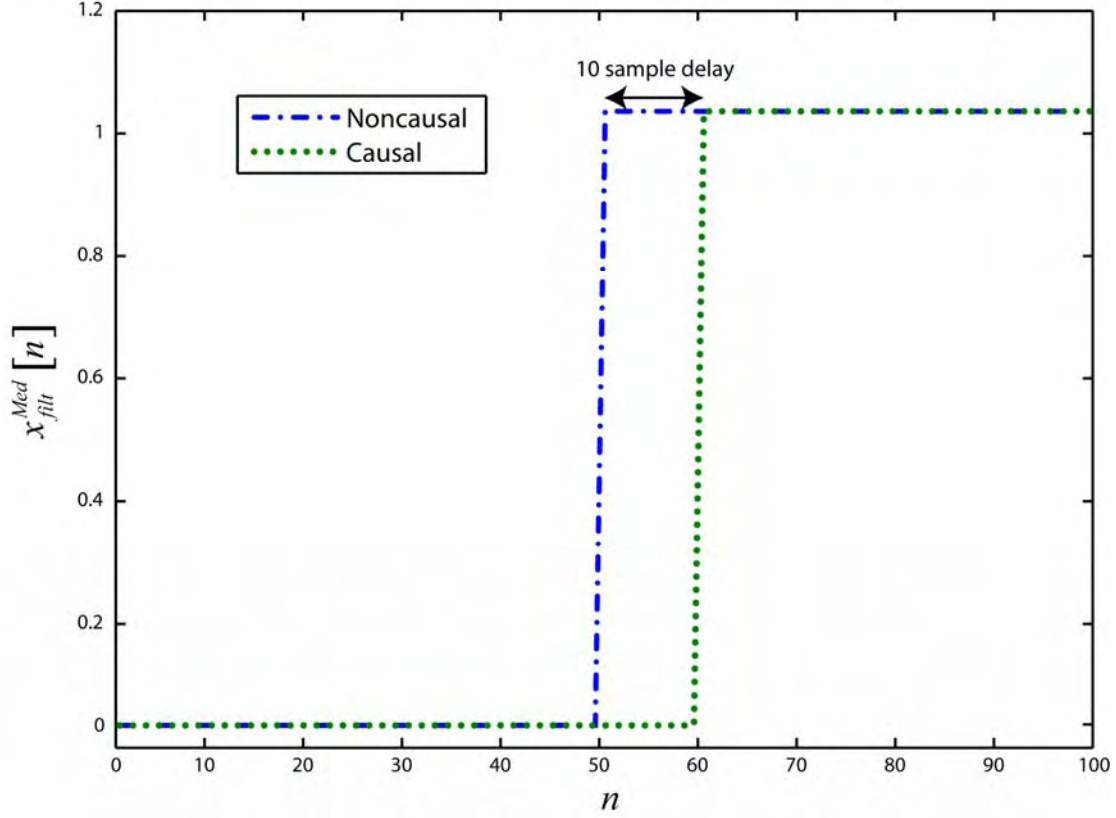


Figure 2.4: Illustration of the delay introduced by using a causal median filter with window length 21 for a step change input signal

As shown in Figure 2.4, the median filter does not corrupt an unmodified step change, except for the known delay introduced by the causal median filter. Unfortunately, power system signals do not typically follow this ideal step change pattern and instead contain step changes combined with oscillations (as mentioned in the previous section). Therefore, the ability of the median filter to handle low-frequency oscillations must be demonstrated in order to ensure it is a viable alternative to FIR filtering. As presented in (2.4), the median filter only has one design parameter— N , the size of the window. To ascertain the performance of the median filter for different values of N , the same PMU signal used in Figure 2.1 and Figure 2.3 was tested with several different window sizes. The outputs of these filters are shown in Figure 2.5; as with FIR filtering, it is clear that the window length is closely tied to the ability to attenuate unwanted oscillations; in addition, an increased window size results in increased delay as expected.

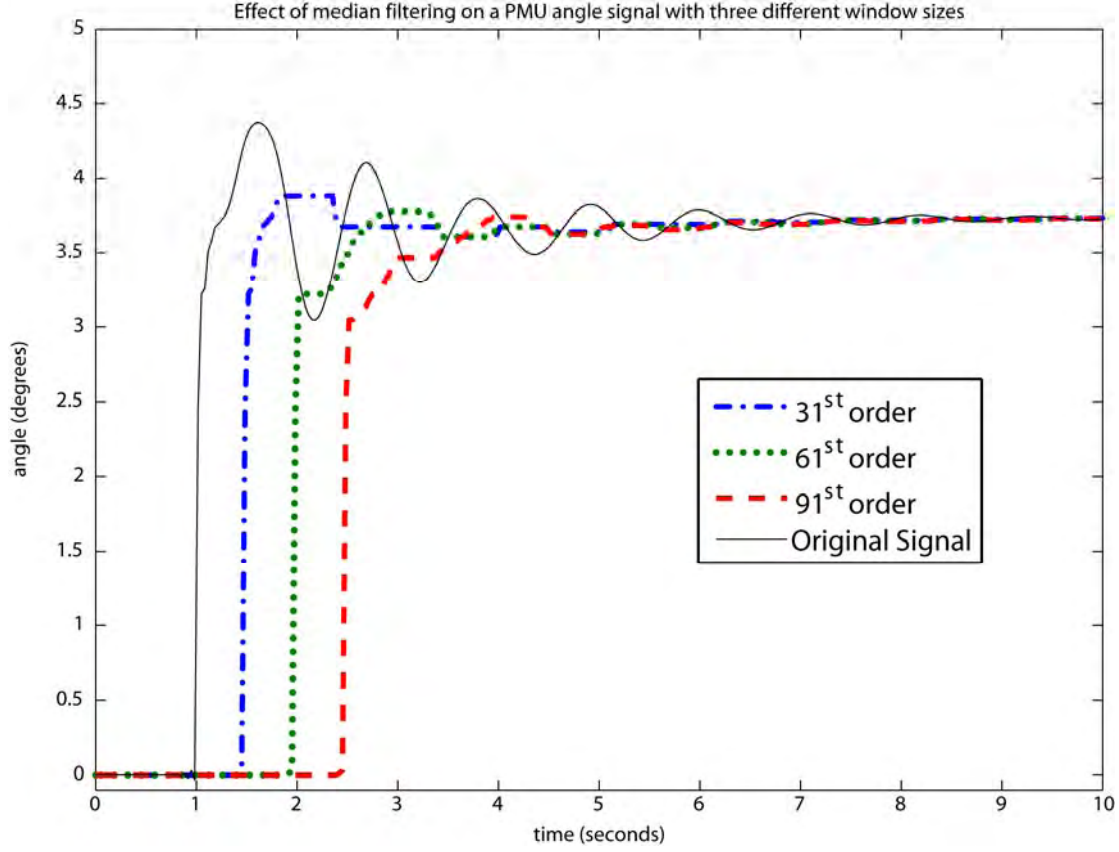


Figure 2.5: Effect of a median filter on a PMU angle signal with window sizes of 31, 61, and 91

Computational and storage requirements of median filtering are based on the need to sort incoming data points to determine the median value over each window. The fact that the window is moved one sample at a time, so that each new median is taken with just one sample changed, allows for several optimized implementation of the median filtering algorithm [62], including one method which is capable of calculating each new output value in $O(\log N)$ time with $O(N)$ storage requirements. Therefore, the computational and storage burden of median filtering is very close to that of FIR filtering. Practically speaking, median filtering data at 4 kHz with a window length of 5 was achievable as early as 1992 using a 20 MHz digital signal processor [32], and the computational requirements of median filtering are not expected to be a determining factor in the filter design.

2.3 Comparing FIR and median filtering

In order to compare the performance of FIR and median filtering, several metrics are adapted from [63] and address the filter criteria given as 1-3:

- Percent overshoot (PO):

$$PO = \left| \frac{x_{filt}[n_{max}] - x_{ss}}{x_{ss}} \right| \times 100\% \quad (2.5)$$

$$\arg \max_{n \in [n_{begin}, \infty)} |x_{filt}[n]| = n_{max}$$

- Rise samples (RS):

$$RS = n_{95\%} - n_{5\%} \quad (2.6)$$

$$\forall n \geq n_{p\%}, |x_{filt}[n]| \geq \frac{p}{100\%} \times |x_{ss}|$$

- Delay samples (DS):

$$DS = n_{50\%} - n_{begin} \quad (2.7)$$

The metrics as defined in (2.5)-(2.7) assume the signal has the steady-state value before the onset of the event subtracted from all signal values so that the steady-state value before the onset of the event is adjusted to 0. The percent difference metric is used to quantify how well the post-event oscillations are damped. The sample value n_{begin} is the last sample before the event occurs (i.e., the last sample in which x maintains its pre-event steady-state value) in the original signal $x[n]$. The quantity x_{ss} is the steady-state value associated with the post-event system conditions. In Figure 2.6, the quantities used to calculate percent overshoot are labeled for a given filter output. As $x_{filt}[n_{max}]$ becomes closer to x_{ss} , the percent overshoot approaches zero and the criterion 1, removal of unwanted noise and oscillations, is better satisfied. Therefore, the best filter would be one which has a percent overshoot of zero, and the closeness of a given filter to zero is a quantitative measure of fitness with respect to criterion 1.

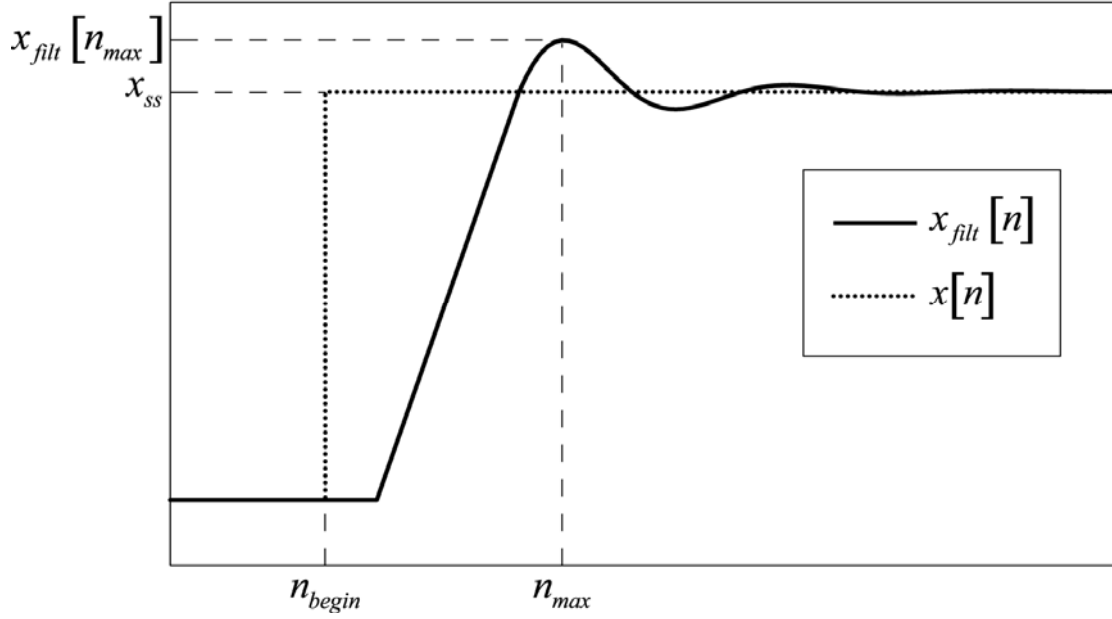


Figure 2.6: Values used to calculate percent overshoot (PO)

The second metric given above, rise samples (RS), provides a measure of how many samples it takes for the output of the filter to reach the new steady state. Delay is eliminated from this calculation by referencing the rise interval from the sample at which 5% the new steady state is attained, rather than n_{begin} . This separates the delay introduced by the filter from the spreading of the edge information, i.e., it separates the filter performance with respect to criteria 2 and 3. Figure 2.7 illustrates the values needed to determine the RS value for a given filter's output. For perfect edge preservation, $n_{95\%}$ and $n_{5\%}$ would be the same sample, leading to an RS value of 0 samples. At the other extreme, for a filter which results in converting the step change into a very slowly rising signal, RS will be a high number. Therefore, RS provides a quantitative measure of the fitness of the filter with respect to criterion 2.

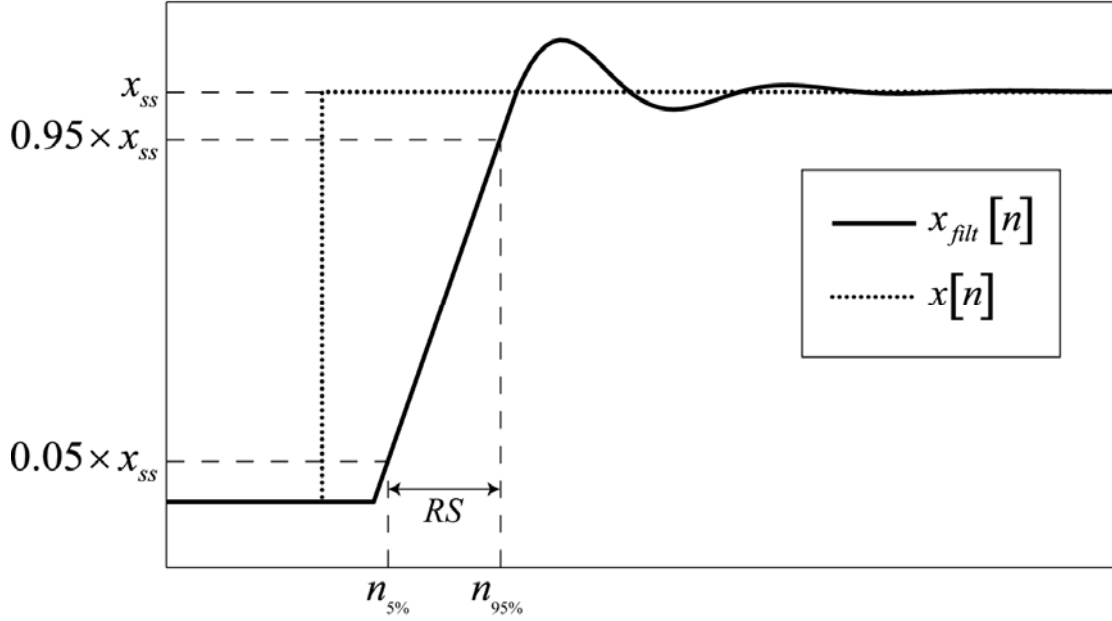


Figure 2.7: Values used to calculate rise samples (RS)

The last of the metrics defined above, delay samples (DS), is a measure of how long after an event occurs before the change in state can be observed. The delay amount is based on the number of samples it takes for the filtered signal to reach 50% of the final steady-state value x_{ss} . This is an important metric in terms of situational awareness, because any delay introduced by the filtering operation can reduce the ability of system operators to react to changing system conditions. Figure 2.8 illustrates how the DS value is calculated for a given filter output. By reducing the amount of delay introduced by the filter, the DS value can be brought to zero; however, as discussed in the previous sections on FIR and median filtering, this is generally not possible due to the usage of causal filtering. However, minimizing DS is a key concern in order to improve the capability of operators to react to changing network conditions and serves as a quantitative measurement of criterion 3.

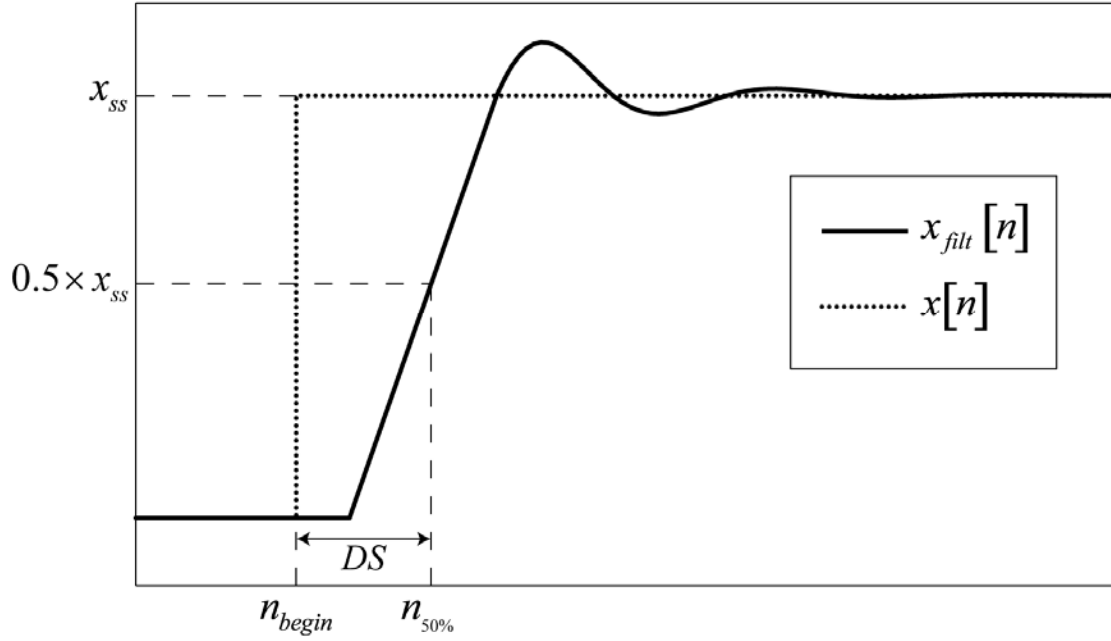


Figure 2.8: Values used to calculate delay samples (DS)

2.4 Filter evaluation

Using the PO , RS , and DS definitions, it is possible to provide a quantitative description of the different filtering options available for usage in PMU-based event detection. A series of tests was run in order to evaluate the capability of various FIR and median filter designs for this application. The results of these tests are presented below.

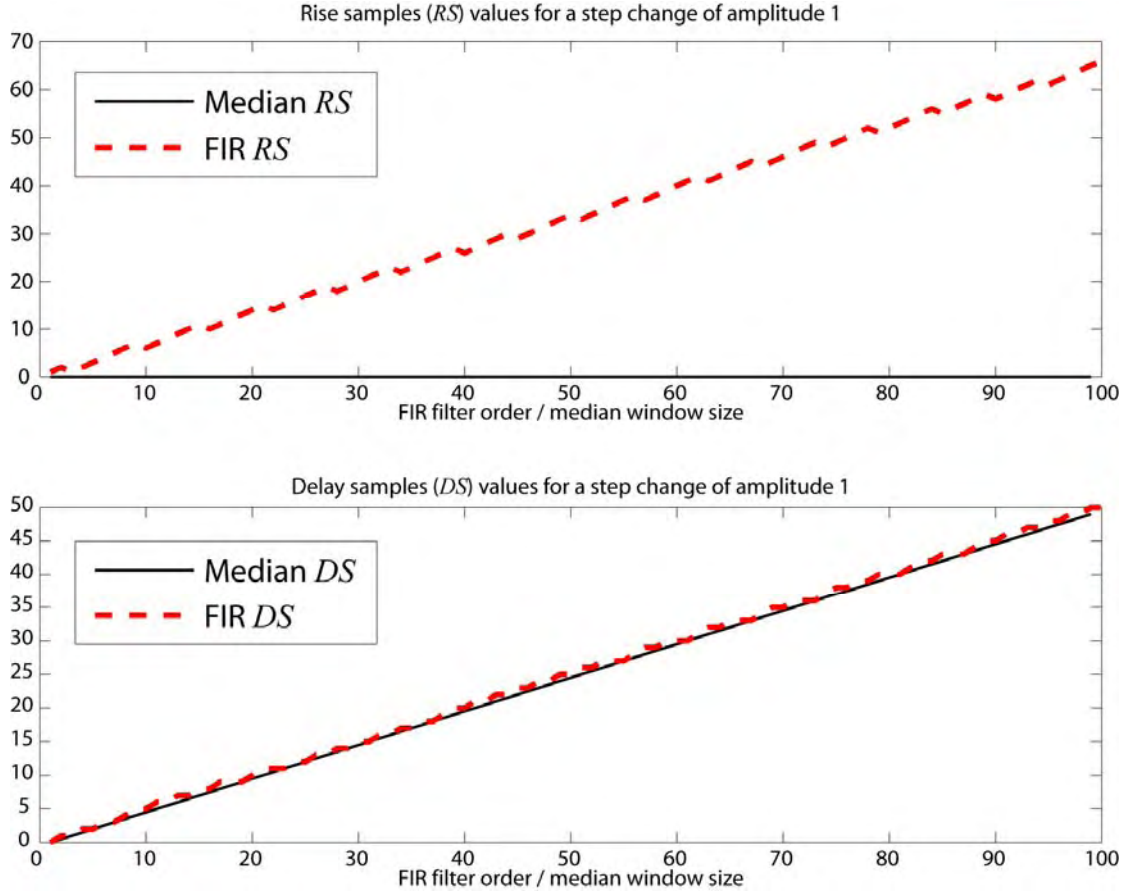


Figure 2.9: *RS* and *DS* values for FIR and median filters with filter orders / window sizes in the range of 1-100 and 1-99 applied to a unit step change.

2.4.1 Step change with no oscillations

The first test performed looks at the response of each filter to a step change in the input signal with no oscillations after the step change. This would correspond to a system which has complete damping of all modes after an event occurs. The signal was constructed to have a sampling time of 1/30 seconds, corresponding to typical PMU signals. The onset of the step change was set to occur at sample 60 (i.e., $n_{begin} = 60$), and the amplitude of the step change was set to 1.0. Figure 2.9 illustrates the performance of the FIR and median filters for filter orders and window lengths ranging from 1 to 99. The *PO* values are not displayed, as the values were zero to within machine precision for all of the filters tested with the basic step change. The uniformly zero value of *RS* for the median filters indicates that the median filter does not corrupt the edge in any way; in contrast, the FIR filters convert the step change into a ramp and result in significantly higher *RS* values. In terms of delay, the linear relationship shown in the second half of Figure 2.9 shows that the FIR and median filters have equivalent delays for a given filter

order or window size (i.e., using an FIR filter of order N results in approximately the same delay as using a median filter with window size N).

Although these results show extremely linear behavior for RS and DS in terms of the filter order, the likelihood of observing a true step change on the power system is very small; therefore, more realistic signals must also be tested.

2.4.2 Step change with 1 Hz oscillations

The second type of signal which the filters were tested with is a unit step change with oscillations added after the step change. The oscillation was set to a frequency of 1 Hz, with a 0.2 damping ratio and magnitude of 0.2, giving rise to the following signal:

$$x[n] = \begin{cases} 0 & n < 60 \\ 1 + 0.2e^{-2\frac{(n-60)}{30}} \sin\left(2\pi \frac{(n-60)}{30}\right) & n \geq 60 \end{cases} \quad (2.8)$$

The unfiltered signal is shown in Figure 2.10.

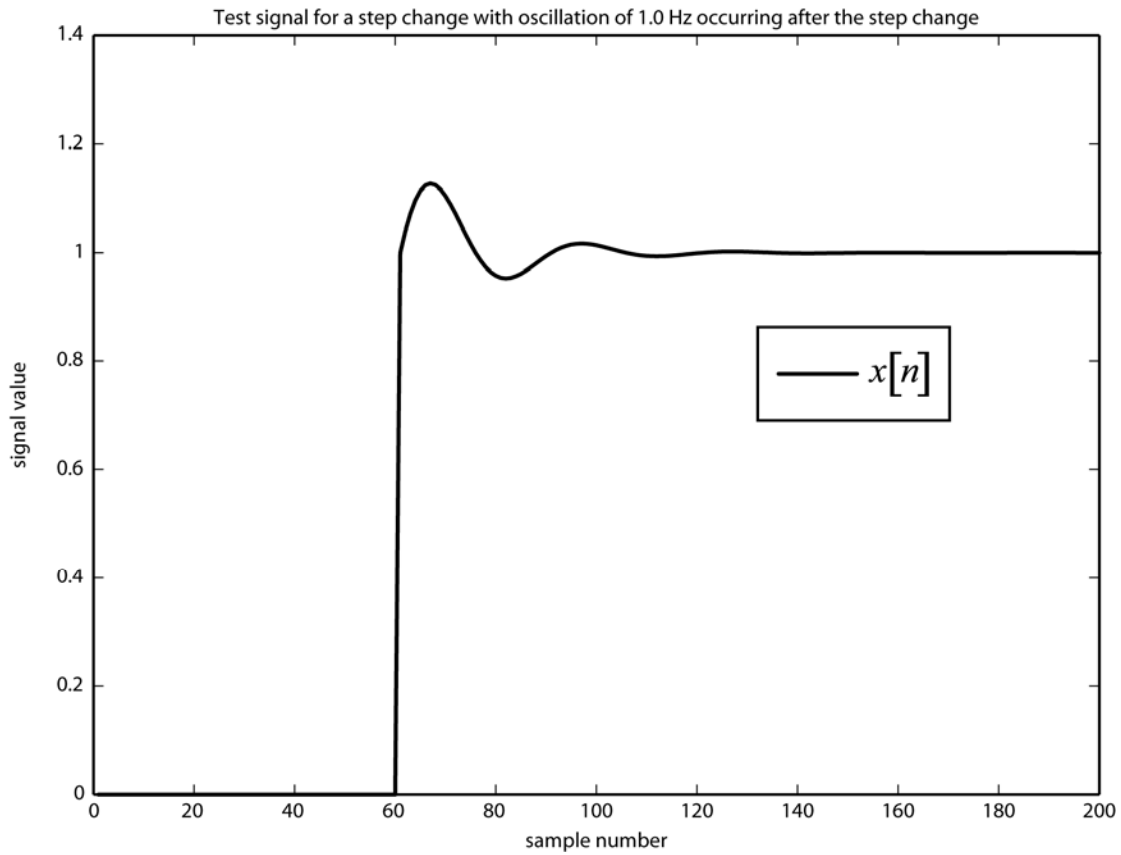


Figure 2.10: Unfiltered test signal used to test filter responses for a step change with 1 Hz oscillations after the step change

As with the step change test outlined in the previous section, the RS and DS values were calculated for FIR filter orders of 1-100 and median window lengths of 1-99. The results obtained are shown in Figure 2.11. By comparing this figure to Figure 2.9, it is clear that the RS and DS values associated with a given FIR filter order or median window length are due primarily to the length of the filter rather than the characteristics of the original signal.

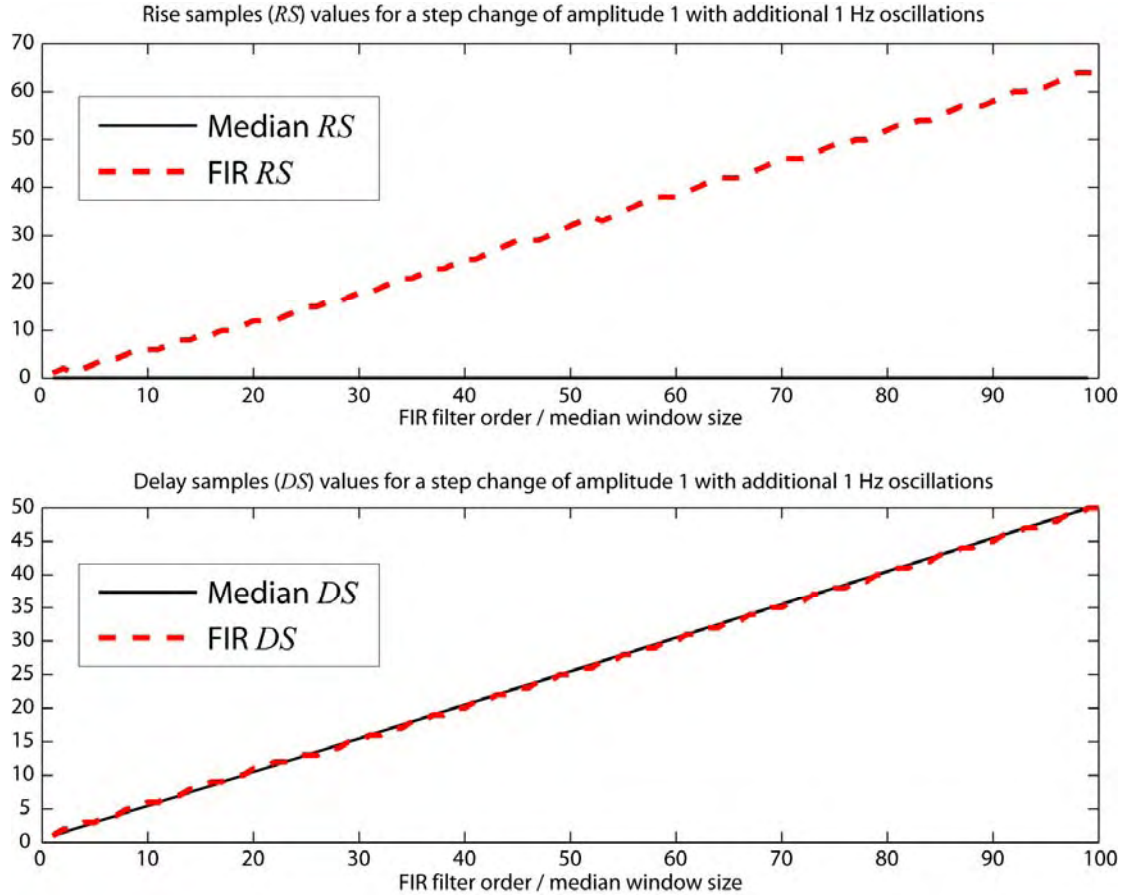


Figure 2.11: RS and DS values for FIR and median filters with filter orders / window sizes in the range of 1-100 and 1-99 applied to a unit step change with post-step oscillations of 1 Hz.

The percent overshoot values associated with the different FIR and median filter lengths are shown in Figure 2.12. For low filter lengths (less than 30), the median filters perform slightly better than the FIR filters; for filter lengths between 35 and 55, the FIR filters perform slightly better than the median filters; and for filter lengths beyond 55, the two filters are essentially equivalent in performance. The key result shown in this figure is that, although the median filters have consistently lower RS values compared to the FIR filters, there are very few differences between the two filter types in terms of DS and PO for a given filter length. Therefore, usage of median filtering results in significant gains with respect to edge corruption with little to no compromise in delay or percent overshoot.

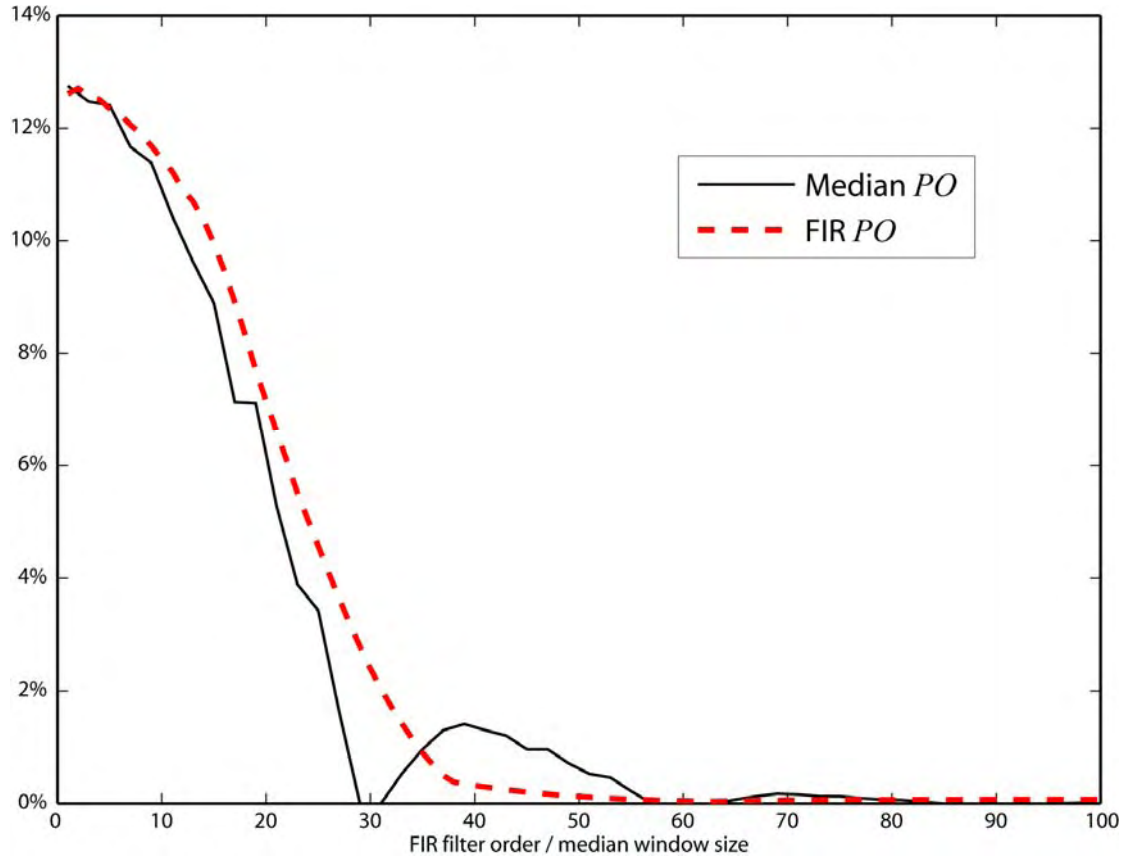


Figure 2.12: PO values for FIR and median filters with filter orders / window sizes in the range of 1-100 and 1-99 applied to a unit step change with post-step oscillations of 1 Hz.

2.4.3 Step change with variable oscillations

Because the data captured by PMUs connected to a real power system can exhibit oscillatory modes over the range of 0.1 to 15 Hz, it is important to understand how the filters behave in the presence of these frequencies. To achieve this goal, RS , DS , and PO values were calculated for signals identical to Figure 2.10 (i.e., with amplitude of 0.2 and damping of 2), with the post-step oscillation frequency varied in 0.1 Hz increments from 0.1 to 15 Hz. The FIR filters tests were conducted with filter orders ranging from 1 to 99 in increments of 2, and the median filter tests were conducted with window lengths ranging from 1 to 99 in increments of 2.

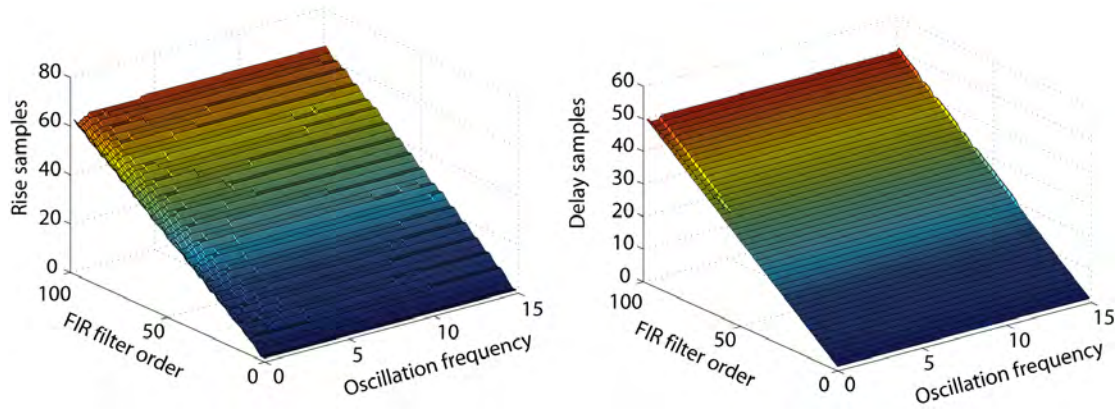


Figure 2.13: *RS* and *DS* values for FIR filters with filter orders in the range of 1-99 applied to a unit step change with post-step oscillations of 0.1-15 Hz.

Figure 2.13 shows that the *RS* and *DS* values associated with a particular FIR filter order are independent of the oscillation frequency. The *DS* value is independent of frequency because the delay of an FIR filter is a function of its filter order, and *RS* is independent because this metric quantifies the effect of the filter on the edge, not the oscillations. The *RS* and *DS* results for the median filters are shown in Figure 2.14; as with the signal tests in sections 2.4.1 and 2.4.2, median filters have similar *DS* values to the FIR filters of the same length. The non-zero *RS* values are due to the interference of the post-step oscillations with the median determination around the step boundary (known as “edge jitter” [64]); however, the highest *RS* value obtained is 9 samples, much lower than the corresponding FIR filters with similar delays.

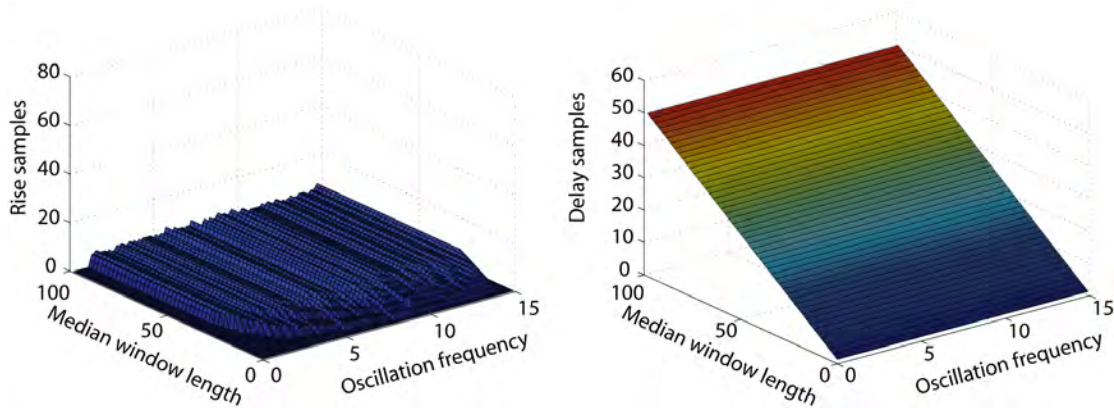


Figure 2.14: *RS* and *DS* values for median filters with window lengths in the range of 1-99 applied to a unit step change with post-step oscillations of 0.1-15 Hz.

The most significant differences with respect to the two filter types are exhibited in the *PO* values. The *PO* values for the FIR and median filters are provided in Figure 2.15 and Figure

2.16, respectively. The *PO* value for each oscillation frequency consistently decreases as the FIR filter order is increased, as expected from the frequency response shown in Figure 2.2. On the other hand, the median filters perform about the same at both low and high frequency extremes.

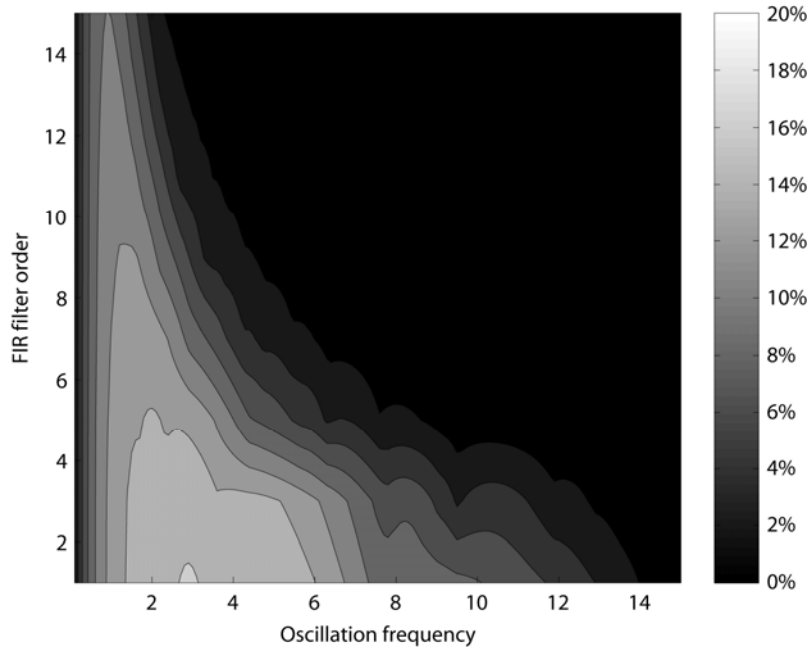


Figure 2.15: *PO* values for FIR filters with filter orders in the range of 1-15 applied to a unit step change with post-step oscillations of 0.1-15 Hz.

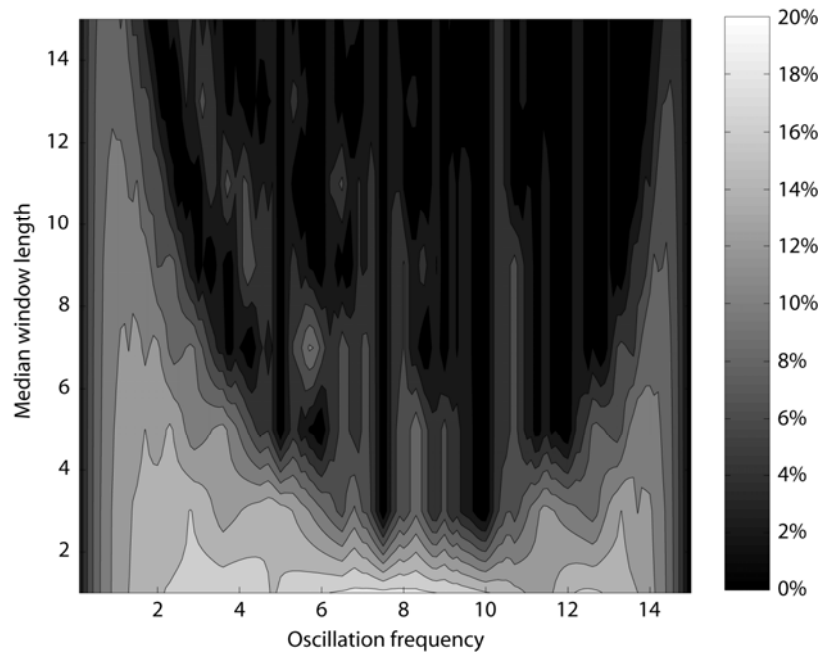


Figure 2.16: *PO* values for median filters with window lengths in the range of 1-15 applied to a unit step change with post-step oscillations of 0.1-15 Hz.

Although the RS values are significantly higher with FIR filters regardless of the filter length or oscillation frequency, there is a tradeoff between DS and PO values for both filter types. To investigate this relationship, a new quantity is defined, $PODS_{x\%}^{\text{filter type}}$, which is the minimum DS value such that for all oscillation frequencies in the range of 0.1-15 Hz, the PO value is less than or equal to $x\%$ for the filter type. The difference in this metric for the two filter types, $PODS_{x\%}^{\text{FIR}} - PODS_{x\%}^{\text{median}}$, can indicate which filter to chose for a desired minimum PO . Figure 2.17 shows that for small percent overshoot requirements ($x < 14$), median filtering can achieve the desired percent overshoot with less delay than FIR filtering.

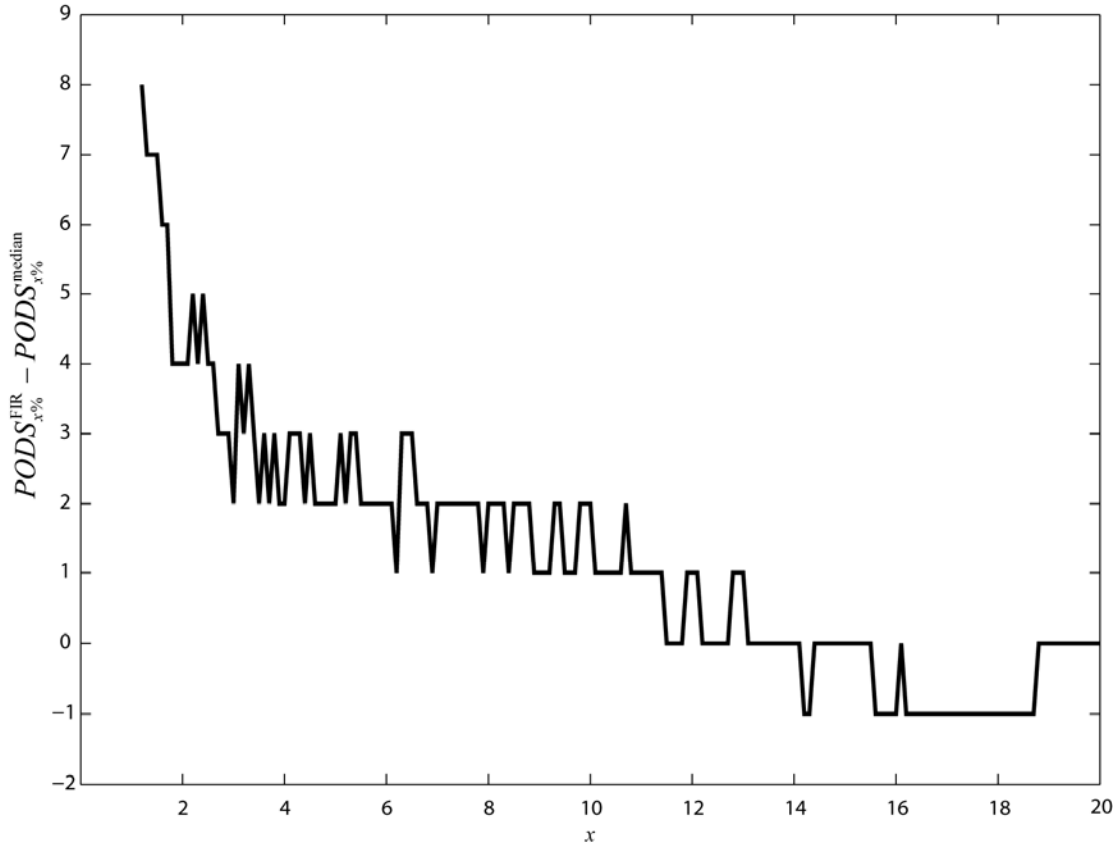


Figure 2.17: $PODS_{x\%}^{\text{FIR}} - PODS_{x\%}^{\text{median}}$ based on unit step and oscillations signals with frequencies of 0.1-15 Hz

2.4.4 Real PMU signal

To test the filters' performance in extracting steady-state angles from PMU signals, a test was also conducted using a PMU angle signal obtained from a North American power company. The particular signal used for this test, along with the steady state estimate, is shown in Figure 2.18. The value for n_{begin} was determined by inspection to be 401. The steady state value

after the event occurs was estimated to be 4.28, obtained by taking the mean of the signal values after n_{begin} .

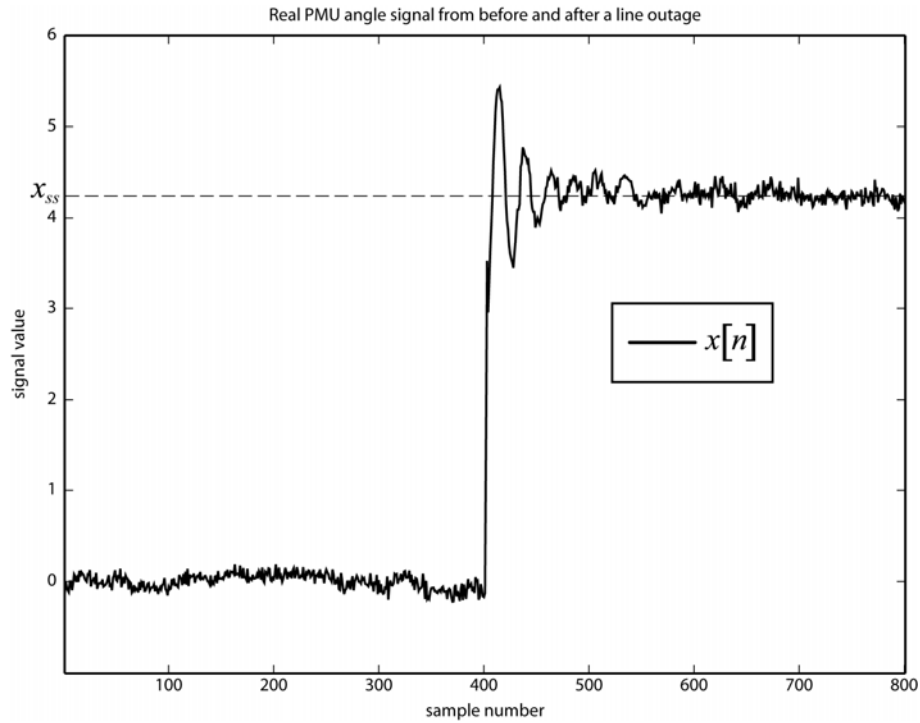


Figure 2.18: Real PMU angle signal used to test filters

FIR and median filters were tested with filter lengths in the range of 1-99. The RS and DS values obtained are shown in Figure 2.19. As in previous tests, the DS

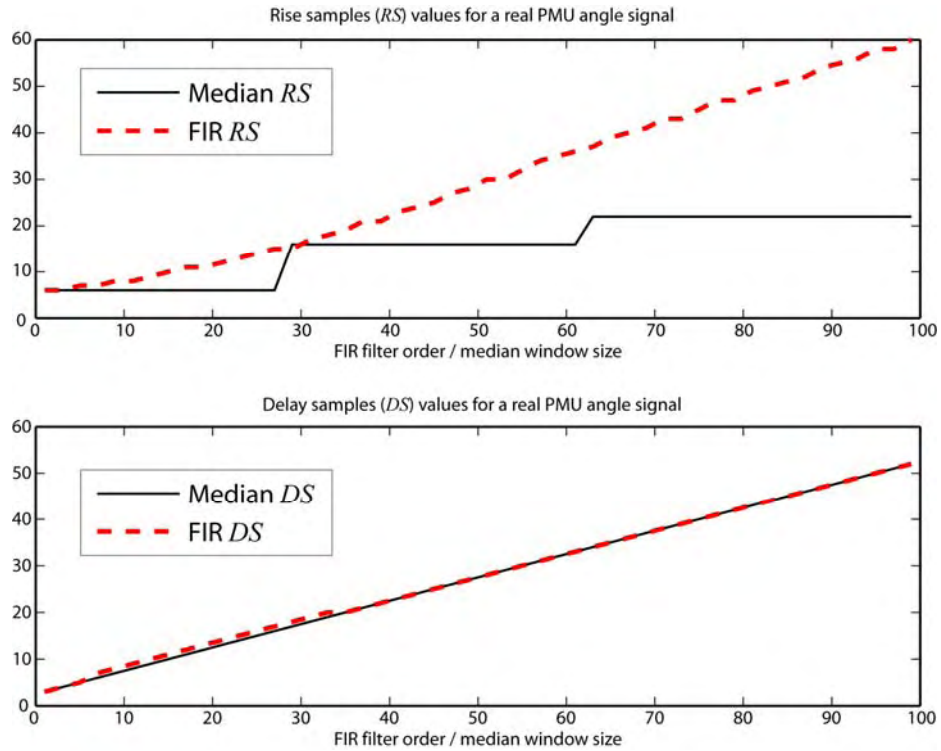


Figure 2.19: *RS* and *DS* values for median and FIR filtering of a real PMU angle signal

values are the same for a given filter order or median length, and the median filter has a lower *RS* value than the FIR filter for all filter lengths except one (the filter length of 29). As in the previous tests, the median filters are able to preserve the edge shape (lower *RS* value) for the same amount of delay (*DS* value).

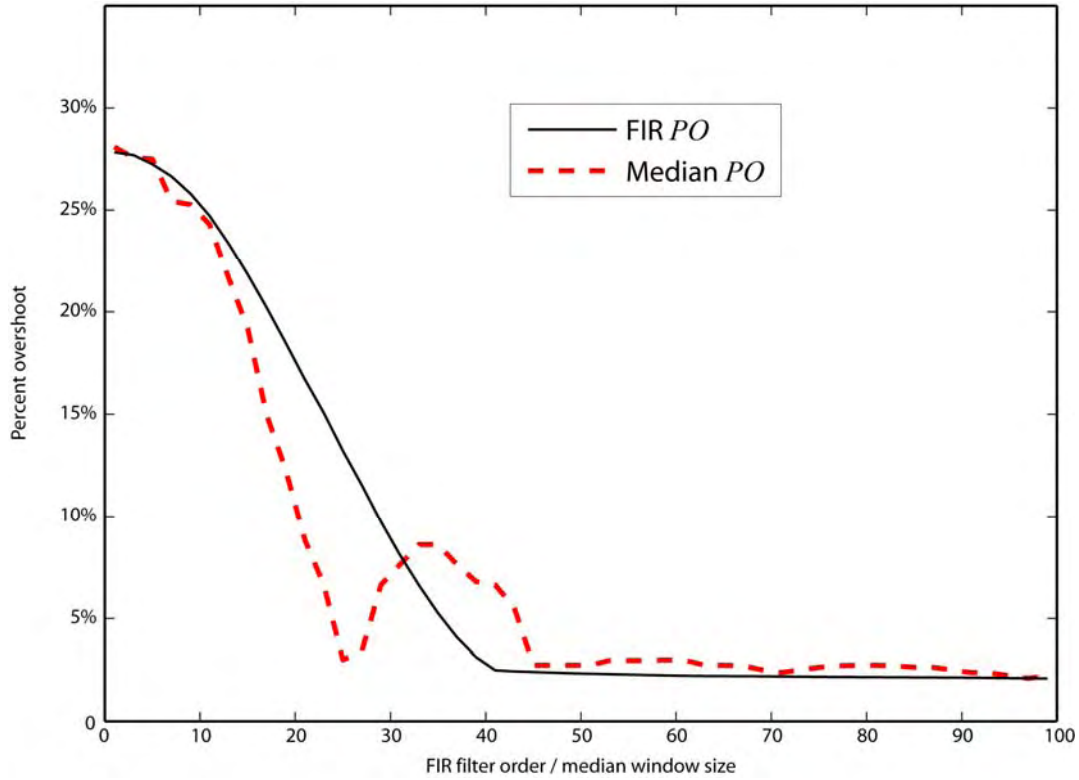


Figure 2.20: *PO* values for median and FIR filtering of a real PMU angle signal

The final metric, percent overshoot, is displayed in Figure 2.20. Based on the results shown in Figure 2.15, the expected monotonic decrease in *PO* values for increasing FIR filter orders is clearly seen. The characteristics of median filtering are more difficult to characterize; for median window lengths which closely match the oscillation period of the original signal, the median filter does an excellent job of eliminating oscillations and results in very low overshoot values. On the other hand, as the median window length increases, the *PO* value increases. This is not desirable, as it indicates an increased amount of delay and processing time is actually reducing the effectiveness of the filter. This type of behavior is also seen in Figure 2.16 and shows that the median filter is not as robust as FIR filtering, particularly if the oscillation frequencies are either unknown or widely varying.

2.5 Conclusions

The sensitivity of the filter response to oscillation frequencies is much more prominent in median filtering than in FIR filtering. For systems in which the post-event oscillation frequencies are easily predicted and at specific frequencies, median filtering can provide much better edge preservation while maintaining low percent overshoot. On the other hand, for systems with broad ranges of oscillation frequencies, higher-order FIR filtering is more robust. Because the oscillations seen on a system are due to an array of factors, including the type of

event and the event location, it is recommended that FIR filtering be used to maintain robustness. The main downside of FIR filtering, edge distortion, is accounted for in the determination of $\Delta\theta_{observed}$, as discussed in the following chapter.

3 Detection of events using PMU data

3.1 Event detection and determination of the angle change vector

3.1.1 Basic procedure

The raw angle measurements are first filtered using one of the methods presented in the previous chapter, with the output of the filter named $\theta_{i, \text{filt}}[n]$ for the filtered phasor angle measurements from bus i . Once the angles have been filtered, a candidate angle change signal $\Delta\theta_{i, \text{candidate}}[n]$ is constructed for each bus i :

$$\Delta\theta_{i, \text{candidate}}[n] = \theta_{i, \text{filt}}[n] - \theta_{i, \text{filt}}[n - N_{\text{trans}}] \quad (3.1)$$

where N_{trans} is the number of samples over which the difference in angles is calculated (see Figure 3.1).

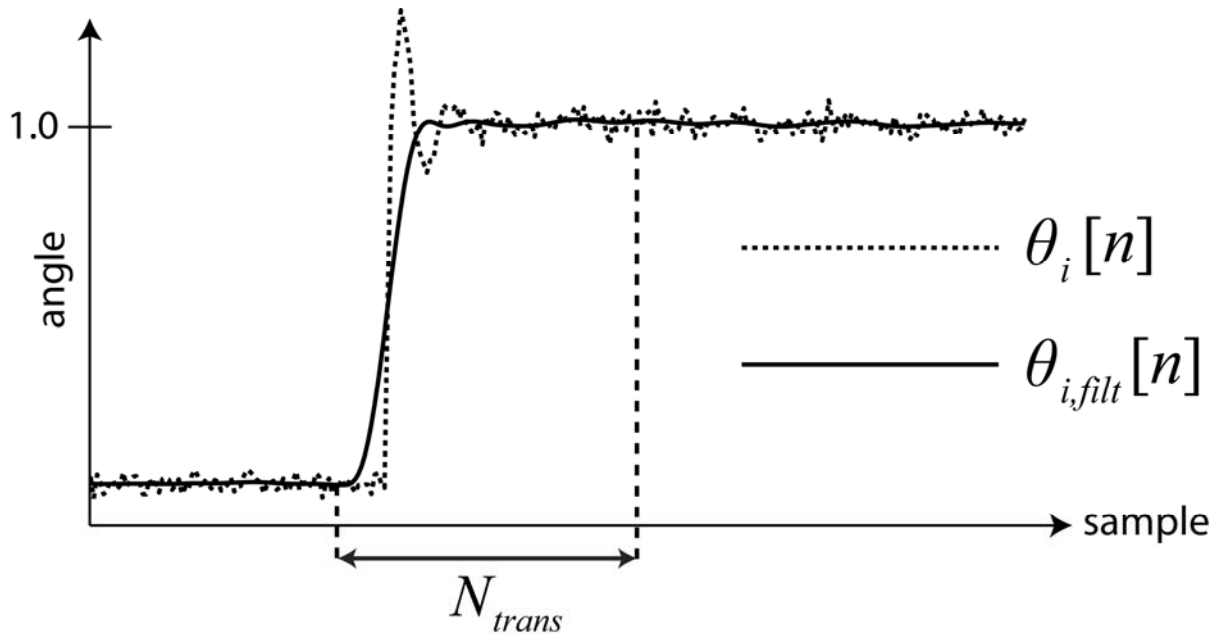


Figure 3.1: Definition of N_{trans} , which is used to generate candidate signals $\Delta\theta_{i, \text{candidate}}[n]$

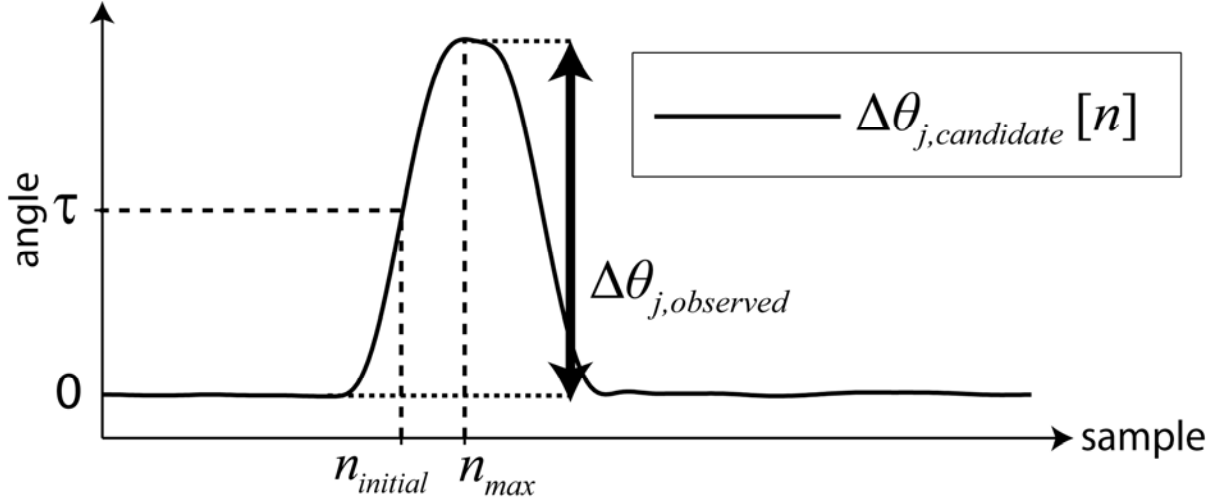


Figure 3.2: Peak detection and determination of the angle change vector

To detect whether an event has occurred, a method commonly used in edge detection [30] was adapted for our purposes. The first step in the process is to continuously compare each of the candidate signals $|\Delta\theta_{i,candidate}[n]|$ against a threshold value τ . If the candidate signal at bus j exceeds the threshold value at sample $n_{initial}$, candidate signal $|\Delta\theta_{i,candidate}[n]|$ is then tracked for $n > n_{initial}$ until it begins to decrease. A decrease implies that the maximum of $|\Delta\theta_{i,candidate}[n]|$ has been reached, and the $\Delta\theta_{observed}$ vector is then constructed using the angle information from all of the buses. The pseudo-code for this “hill climbing” procedure, visualized in Figure 3.2, is:

$$\begin{aligned}
 &\text{if } |\Delta\theta_{j,candidate}[n_{initial}]| > \tau \text{ for a bus } j, \\
 &\quad n_{max} \leftarrow n_{initial} \\
 &\quad n_{test} \leftarrow n_{initial} \\
 &\quad \Delta^2\theta_{j,candidate} \leftarrow +\infty \\
 &\quad \Lambda \leftarrow \text{sign}(\Delta\theta_{j,candidate}[n_{initial}]) \\
 &\quad \text{while } (\Delta^2\theta_{j,candidate} \geq 0) \\
 &\quad \quad \Delta^2\theta_{j,candidate} \leftarrow \Lambda \times (\Delta\theta_{j,candidate}[n_{test}+1] - \Delta\theta_{j,candidate}[n_{test}]) \\
 &\quad \quad n_{test} \leftarrow n_{test} + 1 \\
 &\quad \quad \text{if } (\Delta^2\theta_{j,candidate} > 0) \text{ then } n_{max} \leftarrow n_{test}
 \end{aligned} \tag{3.2}$$

After n_{max} has been determined according to (3.2) using the angle information from bus j , the observed angle change vector is then constructed using the n_{max} samples from all of the candidate signals:

$$\Delta\theta_{observed} = \begin{bmatrix} \Delta\theta_{1,candidate} [n_{max}] \\ \Delta\theta_{2,candidate} [n_{max}] \\ \vdots \\ \Delta\theta_{K,candidate} [n_{max}] \end{bmatrix} \quad (3.3)$$

A key assumption is needed for (3.3) to truly represent the change in steady state angles at all buses—namely, that the n_{max} value obtained from the measurements at bus j corresponds to the steady state changes at the other buses. This assumption will be correct if there is a constant delay in the transition from the old to new steady-state angles at each bus in the system (i.e., the DS and RS values are the same for the angle signal at each bus). For coherent systems, such as regional networks, this is likely to be the case. For large interconnected systems, different buses may reach their steady-state value at a later or earlier time than the first signal to cross the threshold τ . If this behavior is expected, (3.2) can be run independently for each measurement as it passes the threshold, in which case the n_{max} value used in (3.3) would potentially be different for each bus.

3.1.2 Parameter selection

3.1.2.1 Transition samples (N_{trans})

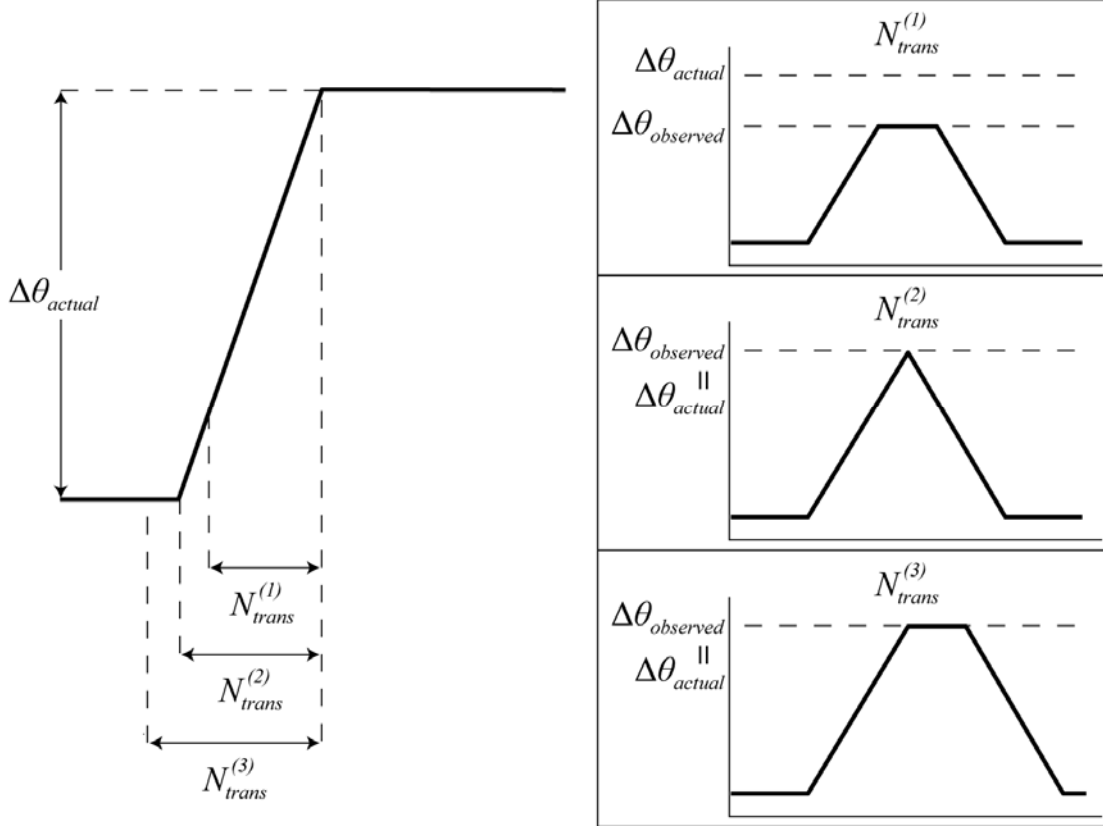


Figure 3.3: Illustration of the relationship between N_{trans} and $\Delta\theta_{observed}$

There are two key parameters involved in both the event detection and angle change vector determination as formulated in the previous section: N_{trans} and τ . The first parameter, N_{trans} , is the number of samples over which the difference in angle measurements is taken. Figure 3.3 illustrates how different values of N_{trans} impact the accuracy and delay in determination of $\Delta\theta_{observed}$. The value $N_{trans}^{(1)}$ represents the case where N_{trans} is lower than the number of samples over which the transition occurs. The resulting $\Delta\theta_{observed}$ underestimates the true change in angles, $\Delta\theta_{actual}$. The second value shown, $N_{trans}^{(2)}$, represents the case where N_{trans} is optimal and $\Delta\theta_{observed}$ is equal to $\Delta\theta_{actual}$. The third value, $N_{trans}^{(3)}$, represents the case where N_{trans} is larger than the number of samples needed to transition between states; in this case, it takes longer for the $\Delta\theta_{candidate}$ signal to decrease below $\Delta\theta_{actual}$. The result of this delay is that the while loop condition in (3.2) will take longer to become false, thereby delaying the detection of the event.

Based on the results shown in Figure 3.3, upper and lower bounds on N_{trans} can be described. The lower bound on N_{trans} is based on the need for the entire transition region to fit inside N_{trans} samples (otherwise the resulting $\Delta\theta_{observed}$ signal will underestimate the true difference in steady state angle values as with $N_{trans}^{(1)}$). A new quantity is defined to measure how well a given value of N_{trans} contains the transition band :

$$\Delta\theta_{miss} = |\Delta\theta_{observed} - \Delta\theta_{actual}| \quad (3.4)$$

Because RS provides a measure of the number of samples in the transition from the pre- to post-event angles, RS can be considered a lower bound for N_{trans} when using a particular filter. An upper bound on N_{trans} is more difficult to quantify; however, choosing too large of a value for N_{trans} could result in misidentification of the pre-event steady state angle and will delay the number of samples before the while loop in (3.2) is exited. To measure this last effect, a new quantity is defined, n_{while} , representing the number of samples spent inside of the while loop of (3.2). The longer the algorithm spends in the while loop, the longer it takes for the event to be detected; as a result, minimizing n_{while} is essential for obtaining timely event information. The optimal N_{trans} can then be described as the value which minimizes $\Delta\theta_{miss}$ and n_{while} . Referring back to Figure 3.3, $N_{trans}^{(1)}$ is a poor choice because it results in large $\Delta\theta_{miss}$ and n_{while} values, $N_{trans}^{(3)}$ is a poor choice because it results in a large n_{while} value, and $N_{trans}^{(2)}$ is the best choice because it minimizes both $\Delta\theta_{miss}$ and n_{while} .

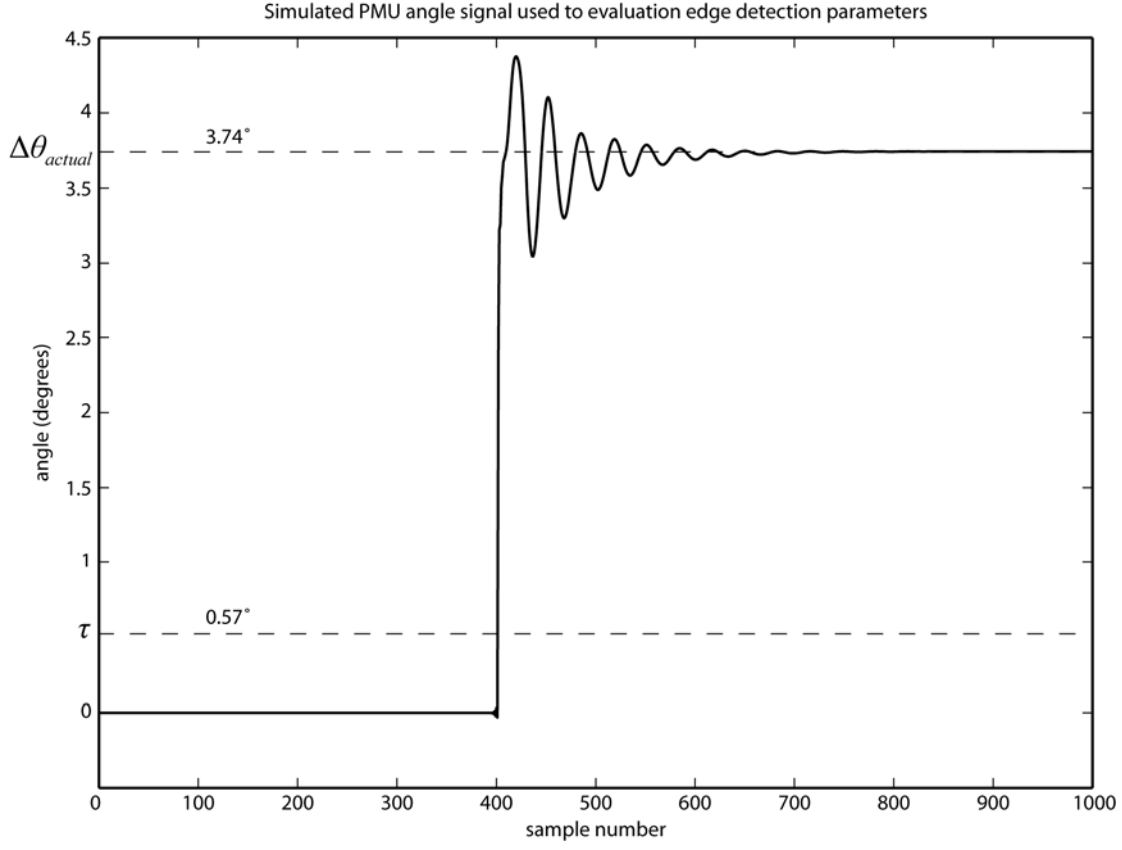


Figure 3.4: Simulated PMU angle signal used to evaluate edge detection parameters

To examine the effects of N_{trans} on a PMU angle signal, the algorithm defined in (3.2) was run on the simulated PMU angle signal shown in Figure 3.4, with the threshold τ set to 0.57 degrees (the maximum error bound from the IEEE PMU standard [27]). Because of the different behavior of FIR and median filtering, particularly with respect to RS , both filter types were evaluated, with filter lengths ranging from 1 to 99 samples.

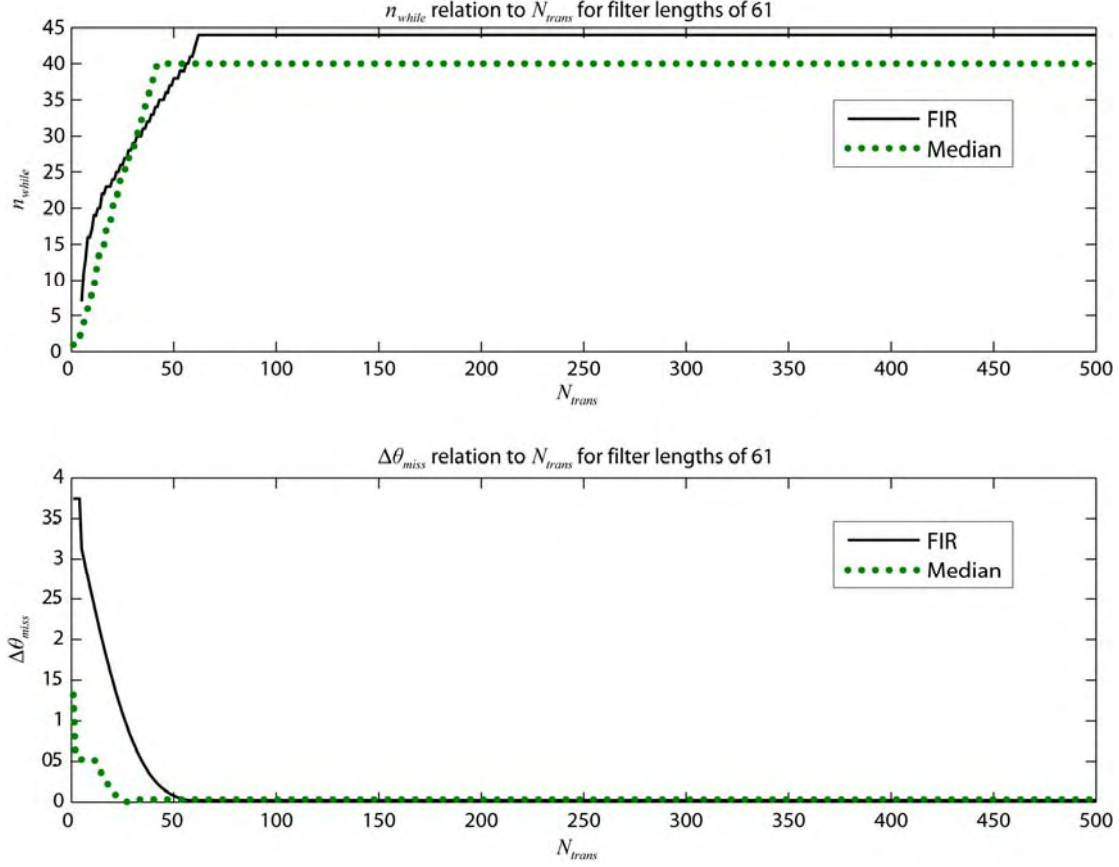


Figure 3.5: $\Delta\theta_{miss}$ and n_{while} values using 61-order FIR and median filtering as N_{trans} values range from 1 to 500 samples

Figure 3.5 shows the $\Delta\theta_{miss}$ and n_{while} values using FIR and median filters of length 61 as N_{trans} values range from 1 to 500 samples. The median filter is able to minimize $\Delta\theta_{miss}$ with shorter values of n_{while} ; this is due primarily to conservation of the step change after the median filter is applied and is directly related to the lower RS values associated with median filtering as described in the previous chapter. On the other hand, the FIR filter results in a better $\Delta\theta_{miss}$ value as N_{trans} gets very large (0.0237 degrees vs. 0.0336 degrees); this indicates that FIR filtering, for a given filter length, is slightly better at filtering out the oscillations present in the original signal. This result corresponds to the reduced PO values associated with FIR filtering, as discussed in the previous chapter. Therefore, the choice of using an FIR or median filter is a tradeoff between minimizing $\Delta\theta_{miss}$ (for which FIR filtering is best) and n_{while} (for which median filtering is best). Similarly, the choice of N_{trans} is also a tradeoff between accuracy and delay—a larger value of N_{trans} tends to increase the value of n_{while} while reducing the value of $\Delta\theta_{miss}$.

In Figure 3.5, the n_{while} values level off after N_{trans} exceeds a certain value for filter lengths of 61. This is because the filters are not able to completely eliminate the overshoot in

the original signal; as a result, the problem illustrated with $N_{trans}^{(3)}$ in Figure 3.3 does not occur. In contrast, Figure 3.6 shows that using a much higher order filter (in this case, with a length of 147) results in enough attenuation that the case illustrated by $N_{trans}^{(3)}$ does occur for FIR filtering. This phenomenon does not occur using median filtering because of the discontinuities present in the final signal as shown in Figure 3.7.

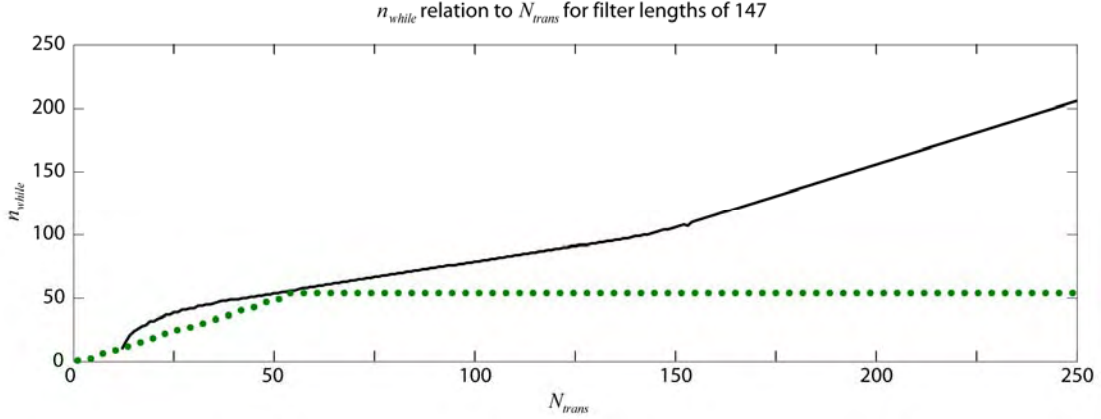


Figure 3.6: n_{while} values using a filter length of 147 FIR and median filtering as N_{trans} values range from 1 to 250 samples

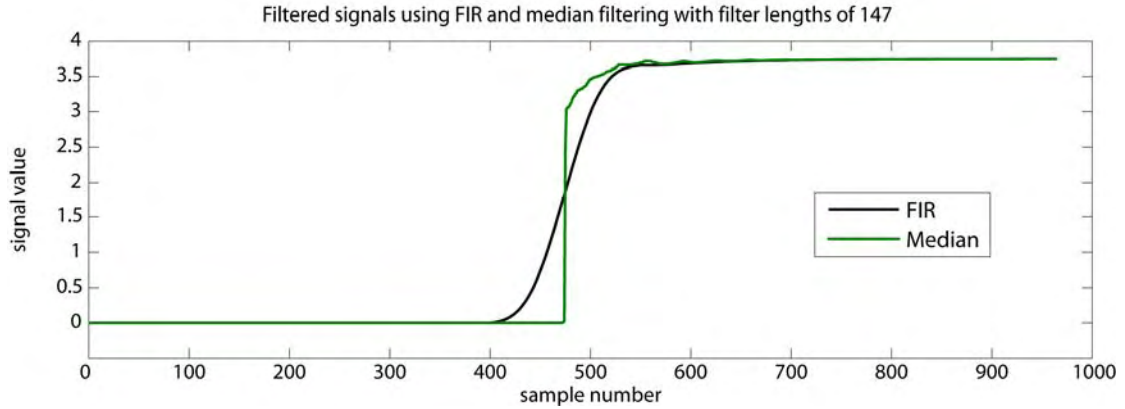


Figure 3.7: Filtering signals obtained using FIR and median filtering of length 147.

3.1.2.2 Difference threshold (τ)

The threshold value τ must also be chosen with care, because setting the threshold value too high might result in missing events that only result in small angle changes (e.g., outages of lines with low pre-outage flow), whereas choosing a threshold value which is too low could result in misclassification of noise as an event. Because the IEEE standard governing PMU behavior [27] requires angle measurements to be accurate within 0.57 degrees of the true angle value, this can serve as a useful upper bound on the threshold value. A useful lower bound on

τ is more difficult to specify, primarily due to the lack of documentation detailing the performance of PMUs from different manufacturers. One lower bound is the accuracy of the GPS timing signals, approximately 0.5 μ s, which translates to a 0.01 degree phase error [27]. In addition, a study of PMUs manufactured by Schweitzer Engineering Laboratories indicates that they are accurate to within 0.01 degrees [63] for a nominal system signal. The ultimate choice of τ must balance the occurrence of false positives (setting τ too low and incorrectly triggering on non-events) with false negatives (setting τ too high and neglecting to trigger on events). In addition, usage of filtering to attenuate noise allows for a lower τ without increases in false positives; therefore, there is a tradeoff between the lower bound on τ and the filter length. This can also be considered as a tradeoff between delay and event misclassification, where increasing delay results in less misclassification.

3.2 Single line outage detection

3.2.1 Analytical basis for single line outage detection

When $\mathcal{E}$ is restricted to a set of single line outages on the system, then the problem defined in (1.3) becomes:

$$\begin{aligned} \text{line outaged } l^* = \\ \arg \min_{l \in \{1, 2, \dots, L\}} \left(\min_{P_l} \left\| \Delta \theta_{\text{observed}} - \text{deltaAngles}_l(P_l) \right\| \right) \end{aligned} \quad (3.5)$$

where L is the number of lines in service before the event is detected and $\text{deltaAngles}_l(P_l)$ is a function which returns the estimated change in angles for the outage of line l with a pre-outage flow of P_l . Because P_l is unknown a priori, it is allowed to vary in order to achieve the best match in observed and calculated angles. Therefore, a unique solution of (3.5) requires that each line outage be distinguishable from the outage of other lines regardless of the pre-outage flow on each line. Solution of (3.5) requires the ability to relate the pre-outage flow on a line l to the observed angle changes on that line if it were to be outaged (represented by $\text{deltaAngles}_l(P_l)$). A simple expression for $\text{deltaAngles}_l(P_l)$ is obtained if the dc power flow equations (1.1) are used. When the dc power flow equations are used, the effect of the outage of a line l can be approximated by a power transfer between the line's "from" bus l_{from} and its "to" bus l_{to} [65]. The transfer amount $\tilde{P}_l$ can be determined from the following equation:

$$\tilde{P}_l = \frac{P_l}{1 - \text{PTDF}_{l, l_{\text{from}} - l_{\text{to}}}} \quad (3.6)$$

where P_l is the pre-outage flow on line l defined as positive if flowing from l_{from} to l_{to} . The value $\text{PTDF}_{l, l_{\text{from}} - l_{\text{to}}}$ is the power transfer distribution factor (PTDF) relating the change in flow

on line l due to a transfer from bus l_{from} to bus l_{to} and can be calculated using only topology and impedance information if the dc power flow assumptions are used [65]. If the denominator is zero (i.e., $PTDF_{l,l_{from}-l_{to}} = 1$), this indicates that the line constitutes a radial connection between two otherwise disconnected systems and the outage cannot be represented as a transfer across the line. In this case, the changes in generator dispatch in the disconnected systems must be modeled to capture the line outage. Section 3.3 below proposes methods for modeling generation redispatch.

If the power transfer $\tilde{P}_l$ is imposed on the system, then a change in angles occurs at all buses. To distinguish the observable angles from the complete set of angles at all buses, a K by N matrix $\mathbf{K}$ is introduced:

$$\mathbf{K} = \begin{bmatrix} \mathbf{I}_{K \times K} & \mathbf{0}_{K \times (N-K)} \end{bmatrix} \quad (3.7)$$

where K is the number of phasor angles observable from the PMUs, N is the total number of system buses, $\mathbf{I}_{K \times K}$ is the $K \times K$ identity matrix, and $\mathbf{0}_{K \times (N-K)}$ is a $K \times (N-K)$ matrix of zeros. The set of angle changes at the observable buses, which is denoted as $\Delta\boldsymbol{\theta}_{calc,l}^{\tilde{P}_l}$, is then found by applying equation (1.1):

$$\begin{aligned} \Delta\boldsymbol{\theta}_{calc,l}^{\tilde{P}_l} &= \mathbf{K}\mathbf{B}^{-1} \begin{bmatrix} 0 \\ \tilde{P}_l \\ -\tilde{P}_l \\ 0 \end{bmatrix} \begin{matrix} \leftarrow l_{from} \\ \leftarrow l_{to} \end{matrix} \\ &= \tilde{P}_l \mathbf{K}\mathbf{B}^{-1} \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} \begin{matrix} \leftarrow l_{from} \\ \leftarrow l_{to} \end{matrix} \\ &= \tilde{P}_l \Delta\tilde{\boldsymbol{\theta}}_{calc,l} \end{aligned} \quad (3.8)$$

As shown in (3.8), the changes in angles are linear with respect to $\tilde{P}_l$; therefore, the calculated changes in angles for a particular pre-outage flow on line l can be written as a scalar $\tilde{P}_l$ multiplying a constant vector $\Delta\tilde{\boldsymbol{\theta}}_{calc,l}$. In turn, (3.5) can be rewritten with $\text{deltaAngles}_l(P_l)$ replaced by the appropriate scalar-vector product:

$$\begin{aligned} \text{line outaged } l^* &= \\ \arg \min_{l \in \{1,2,\dots,L\}} \left(\min_{\tilde{P}_l} \left\| \Delta\boldsymbol{\theta}_{observed} - \tilde{P}_l \Delta\tilde{\boldsymbol{\theta}}_{calc,l} \right\| \right) \end{aligned} \quad (3.9)$$

The optimization given in (3.9) can be performed very quickly using dot products. To see why this is the case, first consider two arbitrary vectors $\mathbf{a}$ and $\mathbf{b}$. From linear algebra, it is

known that the projection of $\mathbf{b}$ onto $\mathbf{a}$, $proj_{\mathbf{a}}\mathbf{b}$, is the vector that minimizes $\mathbf{b} - k\mathbf{a}$, where k is allowed to take on any value [66]. The formula for calculating $proj_{\mathbf{a}}\mathbf{b}$ is:

$$\begin{aligned} k^* &= \frac{\mathbf{a} \cdot \mathbf{b}}{\mathbf{a} \cdot \mathbf{a}} = \arg \min_k \|\mathbf{b} - k\mathbf{a}\| \\ proj_{\mathbf{a}}\mathbf{b} &= k^* \mathbf{a} \end{aligned} \quad (3.10)$$

Comparing equations (3.9) and (3.10), the inner minimization of (3.9) can be rewritten as:

$$\begin{aligned} \tilde{P}_l^* &= \frac{\Delta\boldsymbol{\theta}_{observed} \cdot \tilde{\Delta\boldsymbol{\theta}}_{calc,l}}{\tilde{\Delta\boldsymbol{\theta}}_{calc,l} \cdot \tilde{\Delta\boldsymbol{\theta}}_{calc,l}} \\ \min_{\tilde{P}_l} \|\Delta\boldsymbol{\theta}_{observed} - \tilde{P}_l \tilde{\Delta\boldsymbol{\theta}}_{calc,l}\| &= \|\Delta\boldsymbol{\theta}_{observed} - \tilde{P}_l^* \tilde{\Delta\boldsymbol{\theta}}_{calc,l}\| \end{aligned} \quad (3.11)$$

The inner minimization can then be eliminated and the complete minimization rewritten using dot products:

$$\begin{aligned} \text{line outaged } l^* &= \\ \arg \min_{l \in \{1,2,\dots,L\}} &\left(\left\| \Delta\boldsymbol{\theta}_{observed} - \left(\frac{\Delta\boldsymbol{\theta}_{observed} \cdot \tilde{\Delta\boldsymbol{\theta}}_{calc,l}}{\tilde{\Delta\boldsymbol{\theta}}_{calc,l} \cdot \tilde{\Delta\boldsymbol{\theta}}_{calc,l}} \right) \tilde{\Delta\boldsymbol{\theta}}_{calc,l} \right\| \right) \end{aligned} \quad (3.12)$$

Some manipulation of (3.12) can provide additional intuition into the nature of the minimization process. Consider first the expansion of the norm into dot products:

$$\begin{aligned}
& \arg \min_{l \in \{1,2,\dots,L\}} \left(\left\| \Delta \theta_{observed} - \left(\frac{\Delta \theta_{observed} \cdot \Delta \tilde{\theta}_{calc,l}}{\Delta \tilde{\theta}_{calc,l} \cdot \Delta \tilde{\theta}_{calc,l}} \right) \Delta \tilde{\theta}_{calc,l} \right\| \right) = \\
& \arg \min_{l \in \{1,2,\dots,L\}} \left(\begin{aligned} & (\Delta \theta_{observed} \cdot \Delta \theta_{observed}) \\ & -2(\Delta \theta_{observed} \cdot \Delta \tilde{\theta}_{calc,l}) \left(\frac{\Delta \theta_{observed} \cdot \Delta \tilde{\theta}_{calc,l}}{\Delta \tilde{\theta}_{calc,l} \cdot \Delta \tilde{\theta}_{calc,l}} \right) \\ & + (\Delta \tilde{\theta}_{calc,l} \cdot \Delta \tilde{\theta}_{calc,l}) \left(\frac{\Delta \theta_{observed} \cdot \Delta \tilde{\theta}_{calc,l}}{\Delta \tilde{\theta}_{calc,l} \cdot \Delta \tilde{\theta}_{calc,l}} \right)^2 \end{aligned} \right) = \\
& \arg \min_{l \in \{1,2,\dots,L\}} \left(\begin{aligned} & \frac{(\Delta \theta_{observed} \cdot \Delta \tilde{\theta}_{calc,l})^2}{\Delta \tilde{\theta}_{calc,l} \cdot \Delta \tilde{\theta}_{calc,l}} \\ & -2 \frac{(\Delta \theta_{observed} \cdot \Delta \tilde{\theta}_{calc,l})^2}{\Delta \tilde{\theta}_{calc,l} \cdot \Delta \tilde{\theta}_{calc,l}} \end{aligned} \right) = \\
& \arg \min_{l \in \{1,2,\dots,L\}} \left(\frac{-(\Delta \theta_{observed} \cdot \Delta \tilde{\theta}_{calc,l})^2}{\Delta \tilde{\theta}_{calc,l} \cdot \Delta \tilde{\theta}_{calc,l}} \right) = \\
& \arg \max_{l \in \{1,2,\dots,L\}} \left(\left(\frac{\Delta \theta_{observed}}{\Delta \theta_{observed} \cdot \Delta \theta_{observed}} \right) \cdot \left(\frac{\Delta \tilde{\theta}_{calc,l}}{\Delta \tilde{\theta}_{calc,l} \cdot \Delta \tilde{\theta}_{calc,l}} \right) \right) \tag{3.13}
\end{aligned}$$

The last expression indicates that the minimization of (3.12) is equivalent to the maximization of the dot product between the normalized $\Delta \theta_{observed}$ and $\Delta \tilde{\theta}_{calc,l}$ vectors. Furthermore, maximization of the dot product is equivalent to minimization of the inverse cosine:

$$\begin{aligned}
& \arg \max_{l \in \{1,2,\dots,L\}} \left(\left(\frac{\Delta \theta_{observed}}{\Delta \theta_{observed} \cdot \Delta \theta_{observed}} \right) \cdot \left(\frac{\Delta \tilde{\theta}_{calc,l}}{\Delta \tilde{\theta}_{calc,l} \cdot \Delta \tilde{\theta}_{calc,l}} \right) \right) = \\
& \arg \min_{l \in \{1,2,\dots,L\}} \left(\cos^{-1} \left(\left(\frac{\Delta \theta_{observed}}{\Delta \theta_{observed} \cdot \Delta \theta_{observed}} \right) \cdot \left(\frac{\Delta \tilde{\theta}_{calc,l}}{\Delta \tilde{\theta}_{calc,l} \cdot \Delta \tilde{\theta}_{calc,l}} \right) \right) \right) \tag{3.14}
\end{aligned}$$

This last expression indicates that the line outage which best matches the observed angle changes is the one which has a $\Delta \tilde{\theta}_{calc,l}$ vector that matches the direction of the $\Delta \theta_{observed}$ vector. A normalized angle distance (NAD) metric is defined to quantify the difference in the direction between the observed and expected angle changes for each line:

$$\theta_l = \frac{\cos^{-1} \left(\left(\frac{\Delta \theta_{observed}}{\|\Delta \theta_{observed}\|} \right) \cdot \left(\frac{\Delta \tilde{\theta}_{calc,l}}{\|\Delta \tilde{\theta}_{calc,l}\|} \right) \right)}{2} \quad (3.15)$$

$$NAD_l = 2 \sin \theta_l$$

As shown in Figure 3.8, an NAD value of 0 would correspond to a perfect match between the expected and observed angle changes and the maximum NAD value of $\sqrt{2}$ would correspond to the worst possible match (where $\Delta \theta_{observed}$ and $\Delta \tilde{\theta}_{calc,l}$ are perpendicular).

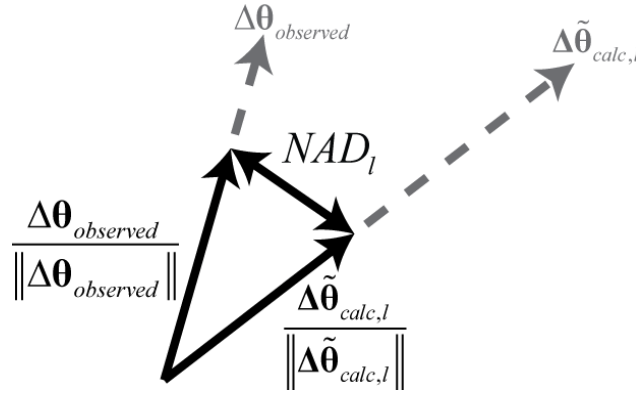


Figure 3.8: Normalized angle difference (NAD) metric

3.2.2 Basic single-line outage event detection algorithm

In order to detect a line outage, identify the outaged line, and determine the pre-outage flow on that line, the following basic algorithm is used:

1. For each line l :
 - a. Calculate $\Delta \tilde{\theta}_{calc,l}$ using (3.8).
 - b. Calculate $\tilde{P}_l^*$ using (3.11).
 - c. Calculate the NAD value for line l using (3.15), then store the calculated error value in the indexed array $NADVals$:

$$NADVals_l = NAD_l \quad (3.16)$$

2. Determine the line l^* that was outaged by sorting $NADVals$:

$$l^* = \arg \min_l NADVals_l \quad (3.17)$$

3. Determine the pre-outage flow on the line which best fits the observed angle, $P_{l^*}^*$, using (3.6):

$$P_{l^*}^* = \tilde{P}_{l^*}^* \left(1 - PTDF_{l^*, l_{from}^* - l_{to}^*} \right) \quad (3.18)$$

3.2.3 Computational complexity

To calculate the angle change due to the outage of a line l using (3.8), the $\mathbf{B}$ matrix must be factored using LU decomposition [67]. This is the most expensive operation in the algorithm, but it only needs to be performed once per change in topology. Once $\mathbf{B}$ is factored, $\Delta\tilde{\theta}_{calc, l}$ and the necessary PTDF values can be computed using forward and backward substitution. In addition, because only certain elements of the angle vector are needed, fast-forward and fast-backward solution can be used to reduce computation time. Outside of step 1.a, the algorithm requires only addition and dot products with vectors of dimension K and a sort operation. These last operations are highly parallelizable and would see performance gains on the order of the number of computing cores available.

3.2.4 Heuristic modifications to the basic algorithm

Additional information is available about the power system, and this information can be exploited to reduce the search space and improve the performance of the line outage detection algorithm. One useful modification is to reject potential outages requiring excessive pre-outage flow on the line in question. For the studies presented below, this was set to either 150% of the line rating in MVA or 5 GW in cases where the rating was unavailable. In addition, system experts may be capable of evaluating potential outages for their feasibility; to facilitate evaluation by operators, the algorithm can easily output the top outage candidates, along with estimated pre-outage flows.

3.3 Generator outage detection

3.3.1 Analytical basis for generator outage detection

As with single-line outages, the basic event detection formulation from (1.3) is first restricted to the set of generator outages:

$$\begin{aligned} &\text{generator outaged } g^* = \\ &\arg \min_{g \in \{1, 2, \dots, G\}} \left(\min_{P_g} \left\| \Delta\theta_{observed} - \text{deltaAngles}_g(P_g) \right\| \right) \end{aligned} \quad (3.19)$$

In (3.19), $\text{deltaAngles}_g(P_g)$ represents the expected changes in observed angles due to the outage of generator g which, before the outage, is generating P_g . As with the single-line outage detection, the dc power flow equations are used to derive an expression for $\text{deltaAngles}_g(P_g)$.

Unlike in the line outage case, there are non-zero changes to power injections on the system upon the outage of a generator. In typical power flow studies, the pickup of the remaining generators is modeled in a variety of ways. One common method is through the designation of a slack bus which, upon the outage of a generator on the system, will pick up the “slack” left by the outaged generator. In this case, the power injection vector would have only one non-zero entry at the row corresponding to the outaged generator. This method is the easiest way to account for changes in generation, but it does not factor in the characteristics of the system’s generators.

In real system operations, a loss in generation is usually picked up by a combination of the remaining generators rather than just one. Usage of participation factors attempts to capture this behavior. In this method, the entry for each generator bus in the power injection vector is determined based on the participation factor (pf) of the corresponding generating unit, i.e.,:

$$\Delta P = \begin{bmatrix} pf_1 \times P_g \\ pf_2 \times P_g \\ \vdots \\ -P_g \\ \vdots \\ pf_G \times P_g \end{bmatrix} \quad \begin{matrix} \leftarrow \text{row corresponding to} \\ \text{outaged generator} \end{matrix} \quad (3.20)$$

$$\sum_{i \in \{1,2,\dots,G\} \setminus g} pf_i = 1$$

The summation in (3.20) ensures that all generation lost by the outaged unit g is picked up by the remaining units. The assumption implicit in having the participation factors sum to one is that changes in losses on the system are negligible; if this is not the case, then the right hand side of the equality can be modified to reflect the expected changes in losses.

While the formulation in (3.20) is much more flexible than the slack bus approach, the validity of (3.20) is dependent on the accuracy of the participation factors used. These factors are commonly used in contingency analyses and are, as a result, available on an area-wide basis for many control areas; however, a system-wide database of participation factors is unavailable. Nonetheless, participation factors can be estimated based on the characteristics of each generator [68] which are available from system-wide dynamic models. For short term redispatch of generation, one of the key determining factors is each generator’s droop and machine base. Droop, otherwise known as speed regulation, is defined as follows [68]:

$$\begin{aligned}
R &= \frac{\left(\frac{\text{decrease in frequency}}{\text{nominal system frequency}} \right)}{\left(\frac{\text{increase in power output}}{\text{maximum power output}} \right)} \times 100\% \\
&= \frac{\left(\frac{f_{\text{no load}} - f_{\text{full load}}}{f_{\text{nominal}}} \right)}{\left(\frac{P^{\text{max}} - 0}{P^{\text{max}}} \right)} \times 100\% = \left(\frac{f_{\text{no load}} - f_{\text{full load}}}{f_{\text{nominal}}} \right) \times 100\%
\end{aligned} \tag{3.21}$$

The subscripted f quantities refer to electrical frequency at various operating conditions (no load, full load, and nominal system frequency), and P^{max} refers to the maximum power output of the generator, which is assumed to be the same as the machine base. For example, if the droop of a generator is 1%, then a 1% deviation in frequency (normalized to system frequency) results in a 100% change in power output of the unit. Figure 3.9 provides an illustration of droop for a single generator.

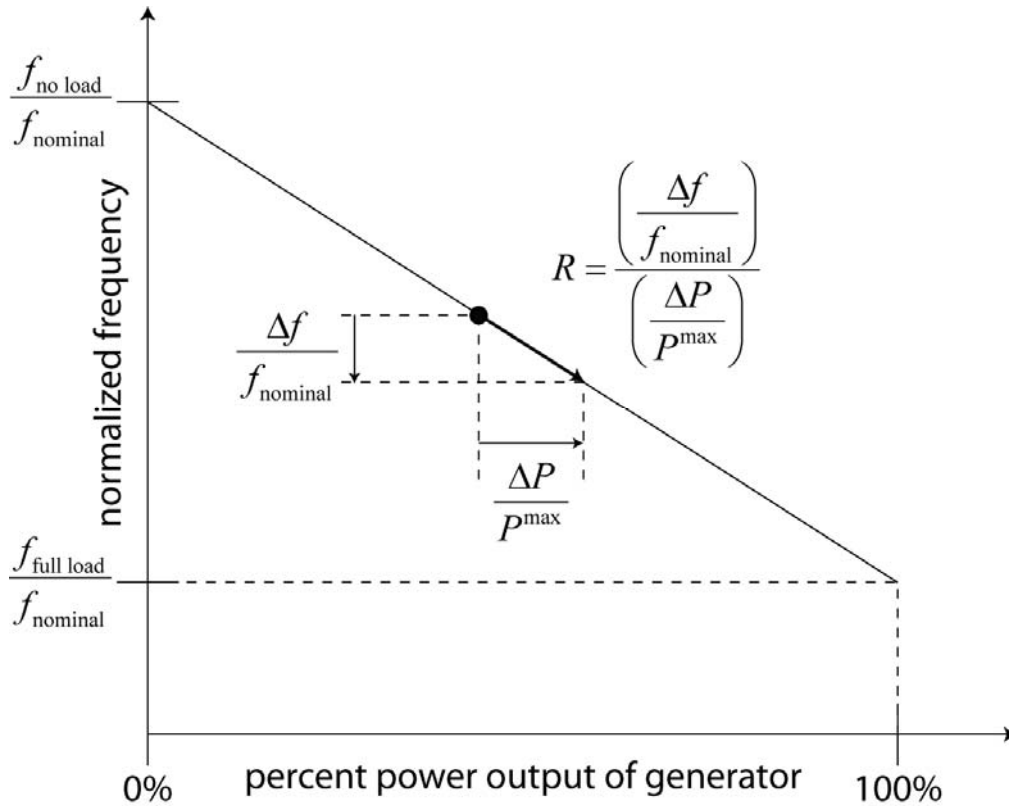


Figure 3.9: Illustration of droop characteristics

Whenever there is a frequency change on the system, this frequency change will propagate throughout the electrical network and affect the output of each generator which is still connected

to the system based on droop settings. If each generator i that is connected to the system has a droop value of R_i and maximum power output of $P_i^{\max}$, the following relations must hold:

$$R_1 \frac{\Delta P_1}{P_1^{\max}} = R_2 \frac{\Delta P_2}{P_2^{\max}} = \dots = R_G \frac{\Delta P_G}{P_G^{\max}} = \frac{\Delta f}{f_{\text{nominal}}} \quad (3.22)$$

after the new system frequency has been reached. The change in frequencies, Δf , is measured at each PMU and is provided in the PMU data set; however, a priori knowledge of which generators are connected to the system is required to determine ΔP_i via (3.22). To calculate ΔP_i if generator g has lost connectivity, (3.20) and (3.22) can be combined:

$$\begin{aligned} R'_i &= \frac{R_i}{P_i^{\max}} \\ R'_1 \times pf_1 \times P_g &= \dots = R'_G \times pf_G \times P_g \\ R'_1 \times pf_1 &= \dots = R'_{g-1} \times pf_{g-1} = R'_{g+1} \times pf_{g+1} = \dots = R'_G \times pf_G \\ \sum_{i \in \{1, 2, \dots, G\} \setminus g} pf_i &= 1 \end{aligned} \quad (3.23)$$

This set of equations is solvable for the participation factors using the following set of linear equations:

$$\begin{bmatrix} R'_1 & -R'_2 & 0 & \dots & \dots & 0 \\ 0 & R'_2 & -R'_3 & 0 & \dots & 0 \\ 0 & 0 & \ddots & \ddots & 0 & \vdots \\ \vdots & \dots & 0 & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & R'_{G-1} & -R'_G \\ 1 & \dots & \dots & \dots & 1 & 1 \end{bmatrix} \begin{bmatrix} pf_1 \\ pf_2 \\ \vdots \\ \vdots \\ pf_{G-1} \\ pf_G \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (3.24)$$

where $\Delta P_i = pf_i \times P_g$

The participation factors obtained via (3.24) can then be used with (1.1), (3.7), and (3.20) to estimate the changes in angles for the outage of generator g providing pre-outage power P_g :

$$\begin{aligned}
\Delta \mathbf{\theta}_{calc,g}^{P_g} &= \mathbf{KB}^{-1} \begin{bmatrix} pf_1 \times P_g \\ \vdots \\ pf_{g-1} \times P_g \\ -P_g \\ pf_{g+1} \times P_g \\ \vdots \\ pf_G \times P_g \end{bmatrix} \\
&= P_g \mathbf{KB}^{-1} \begin{bmatrix} pf_1 \\ \vdots \\ pf_{g-1} \\ -1 \\ pf_{g+1} \\ \vdots \\ pf_G \end{bmatrix} \\
&= P_g \Delta \mathbf{\theta}_{calc,g}
\end{aligned} \tag{3.25}$$

As with the single-line outage events, the expected changes in angles due to a generator outage can be represented as a scalar-vector multiplication. As a result, similar analysis provides the optimizing P_g value for a given observed change in angles:

$$\begin{aligned}
P_g^* &= \frac{\Delta \mathbf{\theta}_{observed} \cdot \Delta \mathbf{\theta}_{calc,g}}{\Delta \mathbf{\theta}_{calc,g} \cdot \Delta \mathbf{\theta}_{calc,g}} \\
\min_{P_g} \|\Delta \mathbf{\theta}_{observed} - P_g \Delta \mathbf{\theta}_{calc,g}\| &= \|\Delta \mathbf{\theta}_{observed} - P_g^* \Delta \mathbf{\theta}_{calc,g}\|
\end{aligned} \tag{3.26}$$

In addition, an *NAD* value can also be defined for generator outages:

$$\begin{aligned}
\theta_g &= \frac{\cos^{-1} \left(\left(\frac{\Delta \mathbf{\theta}_{observed}}{\Delta \mathbf{\theta}_{observed} \cdot \Delta \mathbf{\theta}_{observed}} \right) \cdot \left(\frac{\Delta \mathbf{\theta}_{calc,g}}{\Delta \mathbf{\theta}_{calc,g} \cdot \Delta \mathbf{\theta}_{calc,g}} \right) \right)}{2} \\
NAD_g &= 2 \sin \theta_g
\end{aligned} \tag{3.27}$$

3.3.2 Verification of droop-based participation factors

To verify the usefulness of droop-based participation factors, generator outages were run for each of the nine generators within the system described in Appendix A. Each generator uses an IEEE G1 model [69] with K_G set to 20 (i.e., droop set to 5%). Using these values, the differences between the participation factors estimated from equation (3.24) and calculated from

dynamic simulation were calculated for each outage at each sample time t , where $t = 0$ corresponds to the time of the outage event:

$$\begin{aligned}
\Omega_i &= \begin{cases} 1 & , i \neq \text{outaged gen} \\ 0 & , i = \text{outaged gen} \end{cases} \\
\Delta P_i^{\text{simulated}}(t) &= \Omega_i \times \Delta P_i(t) \text{ simulated} \\
\Delta P_{\text{total}}^{\text{simulated}}(t) &= \sum_{i=\{1,\dots,G\}} \Delta P_i^{\text{simulated}}(t) \\
\mathbf{pf}_{\text{simulated}}(t) &= \begin{bmatrix} \frac{\Delta P_1^{\text{simulated}}(t)}{\Delta P_{\text{total}}^{\text{simulated}}(t)} & \dots & \frac{\Delta P_G^{\text{simulated}}(t)}{\Delta P_{\text{total}}^{\text{simulated}}(t)} \end{bmatrix}^T \\
pf_i^{\text{estimate}} &= \Omega_i \times pf_i, \text{ estimated participation} \\
&\quad \text{factor using equation 4.21} \\
\mathbf{pf}_{\text{estimate}} &= \begin{bmatrix} pf_1^{\text{estimate}} & \dots & pf_G^{\text{estimate}} \end{bmatrix}^T \\
PFD(t) &= \frac{\|\mathbf{pf}_{\text{simulated}}(t) - \mathbf{pf}_{\text{estimate}}\|}{\|\mathbf{pf}_{\text{simulated}}(t)\|} \times 100\%
\end{aligned} \tag{3.28}$$

The droop-based participation factors require the change in frequency seen by each generator to be the same. To quantify how well the frequencies on the system match this condition, a frequency deviation quantity is defined:

$$\begin{aligned}
f_{\text{avg}}(t) &= \frac{\sum_{g \in \{1,\dots,G\}} \Omega_g \Delta f_g(t)}{\sum_{g \in \{1,\dots,G\}} \Omega_g} \\
f_{\text{dev},g}(t) &= \Omega_g (\Delta f_g(t) - f_{\text{avg}}(t)) \\
f_{\text{diff}}(t) &= \left\| \begin{bmatrix} f_{\text{dev},1}(t) \\ \vdots \\ f_{\text{dev},G}(t) \end{bmatrix} \right\|
\end{aligned} \tag{3.29}$$

where Δf_g is the change in frequency from the pre-outage frequency at the bus generator g is connected to. Figure 3.10 shows that the frequency difference between the generators takes between 10 and 20 seconds to reach zero, which implies that the condition given in (3.22) will only be reached 10 to 20 seconds after the outage occurs.

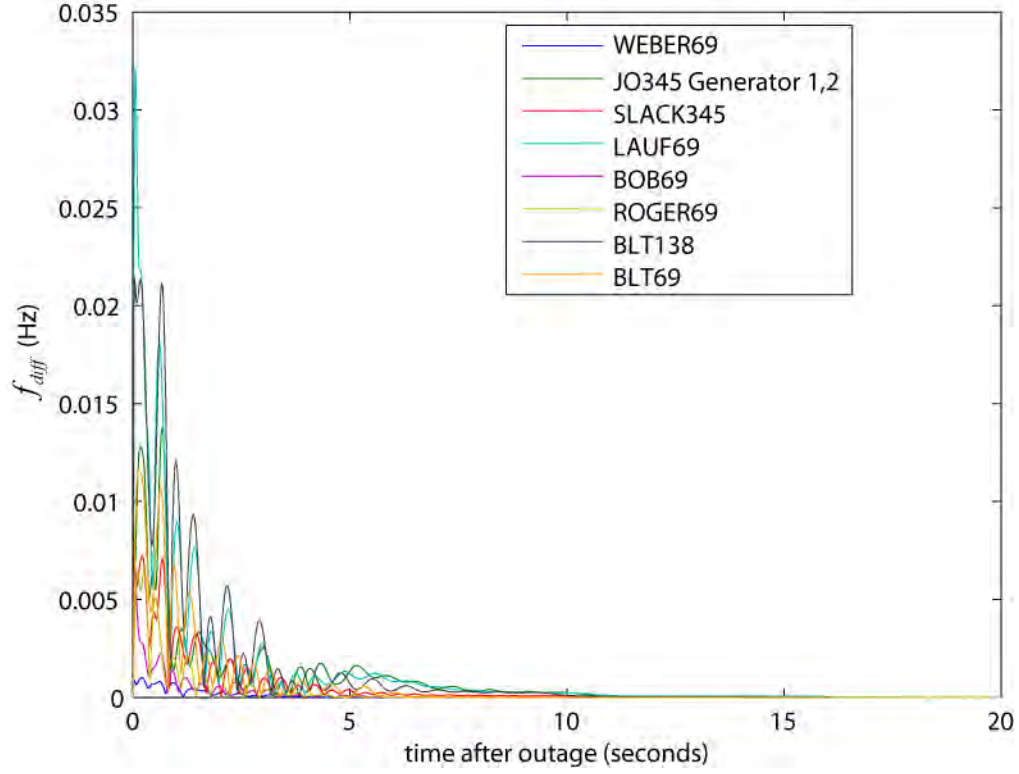


Figure 3.10: Evolution of frequency difference f_{diff} over time for each generator outage

The *PFD* values, which reflect the deviation between the participation factors based on droop and the true participation factors, reach zero several seconds after the frequency deviations reach zero. If the droop response were instantaneous, then *PFD* would decay along with f_{diff} ; however, because there are delays in regulation due to machine inertia and governor delay, there is a lag of several seconds between the frequency deviations going to zero and the *PFD* value going to zero. Comparing Figure 3.10 and Figure 3.11, it is clear that regulation delay results in the *PFD* values for each outage reaching a minimum several seconds after the f_{diff} value reaches zero.

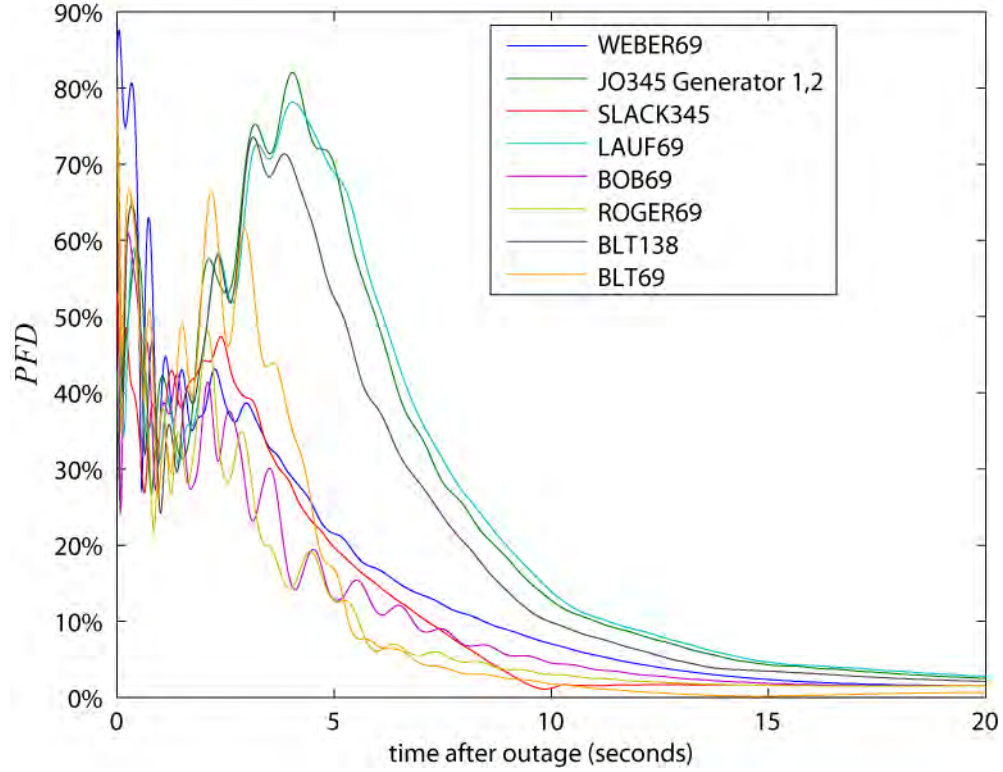


Figure 3.11: Evolution of PFD values over time for each generator outage

The PFD values for each generator outage are provided in Table 3.1 using frequency and power output values 20 seconds after the outage event (i.e., at the rightmost point in Figure 3.11). The results shown in Table 3.1 are promising in that they indicate droop-based participation factors can provide a very close approximation (<2% error) to the true participation factors.

Table 3.1: Accuracy of droop-based participation factors using model parameters

Generator Outaged	WEBER69	JO345	SLACK345	LAUF69	BOB69	ROGER69	BLT138	BLT69
PFD	1.60%	1.67%	1.48%	1.72%	1.60%	1.59%	1.56%	1.00%

For each generator outage, the actual droop can be determined using (3.21) based on the change in frequency and power output at each generator. The mean and standard deviation (taken over the set of nine generator outages) in the simulated droop values for each of the generators is provided in Table 3.2. Although the values are different than the 5% value specified in the generator governor models, the low standard deviation indicates that determining droop values through offline simulation is a viable means of correctly assessing the droop.

Using the droop values in Table 3.2 instead of the 5% value specified in the governor models, the *PFD* values after 20 seconds were evaluated again.

Table 3.2: Droop means and standard deviations determined from simulation

Generator:	WEBER69	JO345	SLACK345	LAUF69	BOB69	ROGER69	BLT138	BLT69
Droop mean	4.93%	4.81%	4.91%	4.84%	4.94%	4.91%	4.81%	4.64%
Droop st. dev.	0.02%	0.02%	0.01%	0.01%	0.01%	0.02%	0.01%	0.01%

The *PFD* determined using the droop values of Table 3.2 are provided in Table 3.3. The usage of the droop values from Table 3.2 provides a more accurate representation of the generator response, as evidenced by the decrease in the *PFD* values for each of the generator outages.

Table 3.3: Accuracy of droop-based participation factors using Table 3.2 droop values

Generator Outaged	WEBER69	JO345	SLACK345	LAUF69	BOB69	ROGER69	BLT138	BLT69
<i>PFD</i>	0.36%	0.06%	0.08%	0.07%	0.13%	0.06%	0.13%	0.10%

Additional tests were run with the droop value for the generator at bus WEBER69 set to 10% rather than 5% to test the sensitivity of this method with respect to individual generator droop settings. Table 3.4 provides the droop means and standard deviations determined from simulation. The droop values for all buses except for WEBER69 stay the same as in Table 3.2, while the WEBER69 droop is approximately 10% as specified in the model of the governor at this generator.

Table 3.4: Droop means and standard deviations determined from simulation, with WEBER69 droop set to 10% instead of 5%

Generator:	WEBER69	JO345	SLACK345	LAUF69	BOB69	ROGER69	BLT138	BLT69
Droop mean	9.74%	4.81%	4.91%	4.84%	4.94%	4.91%	4.81%	4.64%
Droop st. dev.	0.06%	0.02%	0.01%	0.01%	0.01%	0.02%	0.02%	0.01%

Using the droop values in Table 3.4, participation factors were calculated using (3.24) and the PFD values given in Table 3.5 were determined 20 seconds after the outage. These low error values demonstrate that droop-based participation factors are effective in modeling generator response independently of the droop value, as long as sufficient time is allowed for the generators to reach the operating points specified by their respective droop settings.

Table 3.5: : Accuracy of droop-based participation factors using Table 3.4 droop values, with WEBER69 droop set to 10% instead of 5%

Generator Outaged	WEBER69	JO345	SLACK345	LAUF69	BOB69	ROGER69	BLT138	BLT69
PFD	0.39%	0.07%	0.13%	0.05%	0.14%	0.05%	0.14%	0.11%

3.3.3 Basic generator outage detection algorithm

Because the expected angle changes for a single-line outage and a generator outage can both be expressed as scalar-vector multiplications, the algorithms are very similar:

1. For each generator g :
 - a. Calculate $\Delta\theta_{calc,g}$ using (3.25).
 - b. Calculate P_g^* using (3.26).
 - c. Calculate the NAD value for generator g using (3.27), then store the calculated error value in the indexed array $NADVals$:

$$NADVals_g = NAD_g \quad (3.30)$$

2. Determine the generator g^* that was outaged by sorting $NADVals$:

$$g^* = \arg \min_g NADVals_g \quad (3.31)$$

3. Report back the estimated pre-outage generation P_g^* calculated in step 1.b.

3.3.4 Computational complexity

Although the methods used to determine the most likely single line outage and generator outage events are very similar, there are two key differences. The first difference is in the density of the ΔP vector used to calculate $\Delta\theta_{calc,g}$. In the single-line outage case, the ΔP vector has only two non-zero entries corresponding to the from and to buses of the outaged line. With generator outages, the ΔP vector will have nonzero entries at each bus with an online

generator attached. This is typically a small subset of the total number of system buses, but may result in increased computation time.

The second key difference is that there are fewer generators than lines within a power system, so the number of events is drastically reduced. This reduction in events causes fewer calculations of $\Delta\theta_{calc}$ and NAD values and a faster sort operation in determining the most likely generator outage.

3.3.5 Heuristic modifications to the basic algorithm

By taking advantage of known generator ratings, it is possible to further restrict the event set. The way this is handled in the algorithm is to remove generators from consideration if the estimated pre-outage generation is greater than 1.5 times the rating of the generation. The extra 50% leeway allows for possible errors introduced by the dc power flow assumptions, participation factor selection, and errors in determining $\Delta\theta_{observed}$.

3.4 Double line outage detection

3.4.1 Analytical basis for double line outage detection

When $\mathcal{E}$ is restricted to the set of double line outages on the system, then the problem defined in (1.3) becomes:

$$\begin{aligned} \text{lines outaged } \{l_1^*, l_2^*\} = \\ \arg \min_{\{l_1, l_2\} \in \{1, 2, \dots, L\} \times \{1, 2, \dots, L\}} \left(\min_{P_{l,1}, P_{l,2}} \left\| \Delta\theta_{observed} - \text{deltaAngles}_{l_1, l_2}(P_{l,1}, P_{l,2}) \right\| \right) \end{aligned} \quad (3.32)$$

where L is the number of lines in service before the event is detected and $\text{deltaAngles}_{l_1, l_2}(P_{l,1}, P_{l,2})$ is a function which returns the estimated change in angles for the outages of lines l_1 and l_2 with pre-outage flows of $P_{l,1}$ and $P_{l,2}$, respectively. Because the pre-outage flows are unknown a priori, each is allowed to vary in order to achieve the best match in observed and calculated angles.

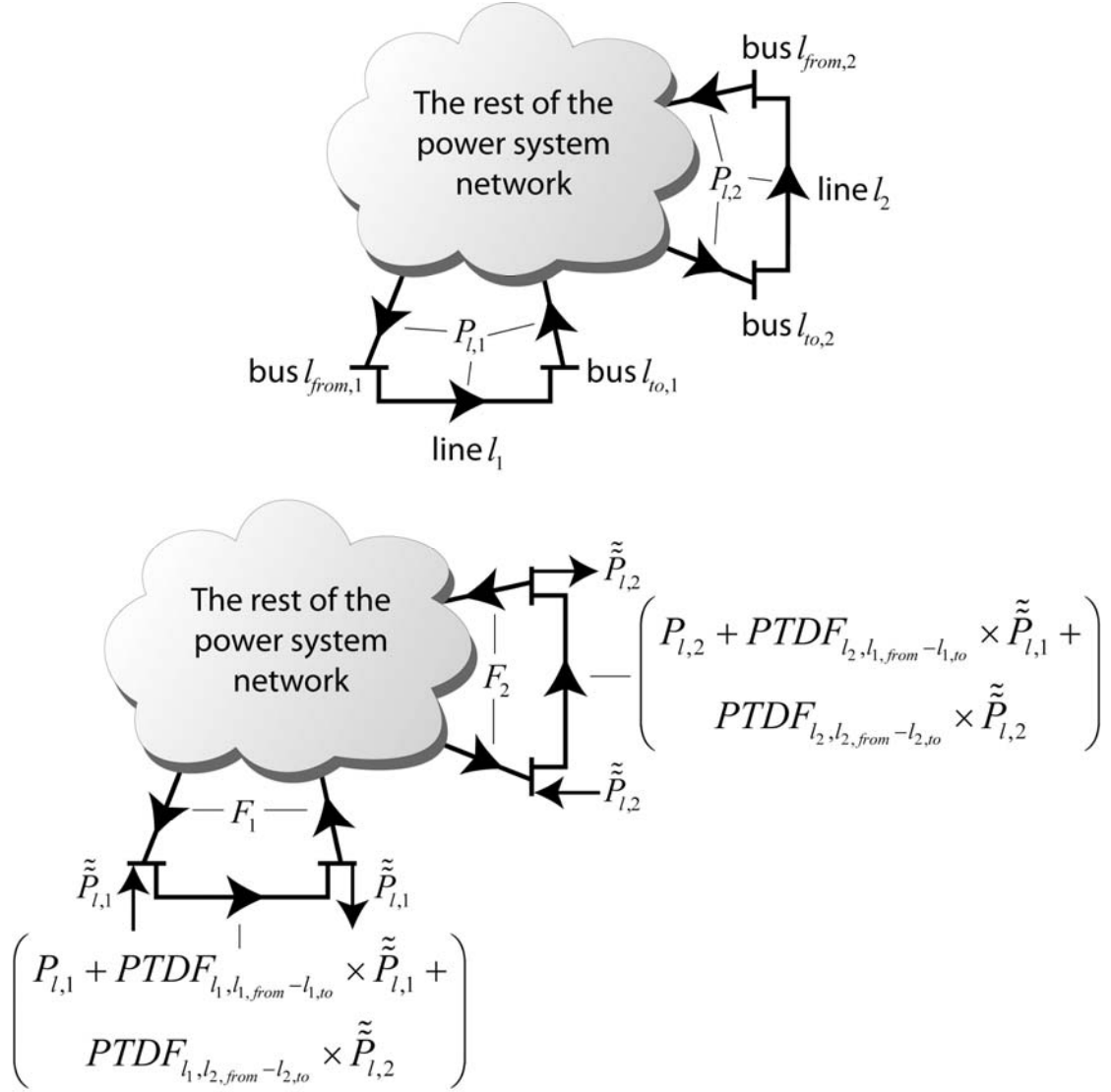


Figure 3.12: Double-line outage model with power injections

To characterize the function $\text{deltaAngles}_{l_1, l_2}(P_{l,1}, P_{l,2})$ under the dc power flow assumptions, the outages are modeled by power injections at the terminal buses of the outaged lines. Figure 3.12 illustrates the relevant quantities used to model a double line outage. Based on the flows and injections shown in Figure 3.12, the following two equalities must hold for the flow from the rest of the system to the outaged lines, F_1 and F_2 , to be zero:

$$\begin{bmatrix} \tilde{\tilde{P}}_{l,1} \\ \tilde{\tilde{P}}_{l,2} \end{bmatrix} = \begin{bmatrix} \left(P_{l,1} + PTDF_{l_1,l_1,from-l_{1,to}} \times \tilde{\tilde{P}}_{l,1} + \right. \\ \left. PTDF_{l_1,l_2,from-l_{2,to}} \times \tilde{\tilde{P}}_{l,2} \right) \\ \left(P_{l,2} + PTDF_{l_2,l_1,from-l_{1,to}} \times \tilde{\tilde{P}}_{l,1} + \right. \\ \left. PTDF_{l_2,l_2,from-l_{2,to}} \times \tilde{\tilde{P}}_{l,2} \right) \end{bmatrix} \quad (3.33)$$

Rewriting this in matrix form, an expression for $\tilde{\tilde{P}}_{l,1}$ and $\tilde{\tilde{P}}_{l,2}$ in terms of PTDFs and pre-outage line flows can be obtained:

$$\begin{bmatrix} \tilde{\tilde{P}}_{l,1} \\ \tilde{\tilde{P}}_{l,2} \end{bmatrix} = \begin{bmatrix} 1 - PTDF_{l_1,l_1,from-l_{1,to}} & -PTDF_{l_1,l_2,from-l_{2,to}} \\ -PTDF_{l_2,l_1,from-l_{1,to}} & 1 - PTDF_{l_2,l_2,from-l_{2,to}} \end{bmatrix}^{-1} \begin{bmatrix} P_{l,1} \\ P_{l,2} \end{bmatrix} \quad (3.34)$$

The inverse matrix in (3.34) exists only if the outage of the two lines does not result in islanding [70]. If the lines do result in islanding of the system, then the outages cannot be modeled as transfers across the lines and must instead be modeled as changes in power injection at the boundary buses of the original system, taking into account any necessary redispatch of generation as discussed in section 3.3.1.

Using the formula for the inverse of a 2 x 2 matrix, the solution in (3.34) can be rewritten:

$$\begin{aligned} \alpha_{1,2} &= \left(1 - PTDF_{l_1,l_1,from-l_{1,to}}\right) \left(1 - PTDF_{l_2,l_2,from-l_{2,to}}\right) - \\ &\quad \left(PTDF_{l_2,l_1,from-l_{1,to}}\right) \left(PTDF_{l_1,l_2,from-l_{2,to}}\right) \\ \begin{bmatrix} \tilde{\tilde{P}}_{l,1} \\ \tilde{\tilde{P}}_{l,2} \end{bmatrix} &= \frac{1}{\alpha_{1,2}} \begin{bmatrix} 1 - PTDF_{l_2,l_2,from-l_{2,to}} & PTDF_{l_1,l_2,from-l_{2,to}} \\ PTDF_{l_2,l_1,from-l_{1,to}} & 1 - PTDF_{l_1,l_1,from-l_{1,to}} \end{bmatrix} \begin{bmatrix} P_{l,1} \\ P_{l,2} \end{bmatrix} \\ \begin{bmatrix} \tilde{\tilde{P}}_{l,1} \\ \tilde{\tilde{P}}_{l,2} \end{bmatrix} &= \frac{P_{l,1}}{\alpha_{1,2}} \begin{bmatrix} 1 - PTDF_{l_2,l_2,from-l_{2,to}} \\ PTDF_{l_2,l_1,from-l_{1,to}} \end{bmatrix} + \frac{P_{l,2}}{\alpha_{1,2}} \begin{bmatrix} PTDF_{l_1,l_2,from-l_{2,to}} \\ 1 - PTDF_{l_1,l_1,from-l_{1,to}} \end{bmatrix} \end{aligned} \quad (3.35)$$

Once the double outage model injection vectors, $\tilde{\tilde{P}}_{l,1}$ and $\tilde{\tilde{P}}_{l,2}$, are obtained, the resulting change in angles can then be determined:

$$\begin{aligned}
\Delta \theta_{\tilde{\tilde{P}}_{l_1, l_2}}^{\tilde{\tilde{P}}_{l_1, l_2}} &= \mathbf{KB}^{-1} \left(\begin{bmatrix} 0 \\ \tilde{\tilde{P}}_{l_1} \\ -\tilde{\tilde{P}}_{l_1} \\ 0 \end{bmatrix} \leftarrow \begin{matrix} l_{from,1} \\ l_{to,1} \end{matrix} + \begin{bmatrix} 0 \\ \tilde{\tilde{P}}_{l_2} \\ -\tilde{\tilde{P}}_{l_2} \\ 0 \end{bmatrix} \leftarrow \begin{matrix} l_{from,2} \\ l_{to,2} \end{matrix} \right) \\
&= \tilde{\tilde{P}}_{l_1} \left(\mathbf{KB}^{-1} \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} \leftarrow \begin{matrix} l_{from,1} \\ l_{to,1} \end{matrix} \right) + \\
&\quad \tilde{\tilde{P}}_{l_2} \left(\mathbf{KB}^{-1} \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} \leftarrow \begin{matrix} l_{from,2} \\ l_{to,2} \end{matrix} \right) \\
&= \tilde{\tilde{P}}_{l_1} \Delta \tilde{\tilde{\theta}}_{calc, l_1} + \tilde{\tilde{P}}_{l_2} \Delta \tilde{\tilde{\theta}}_{calc, l_2}
\end{aligned} \tag{3.36}$$

Note that the $\Delta \tilde{\tilde{\theta}}_{calc, l}$ vectors in (3.36) are the same vectors defined in (3.8) for single line outage modeling. The optimization of (3.32) can then be rewritten in terms of the vectors defined in (3.36):

$$\begin{aligned}
&\text{lines outaged } \{l_1^*, l_2^*\} = \\
&\arg \min_{\{l_1, l_2\} \in \{1, 2, \dots, L\} \times \{1, 2, \dots, L\}} \left(\min_{\tilde{\tilde{P}}_{l_1}, \tilde{\tilde{P}}_{l_2}} \left\| \Delta \theta_{observed} - \left(\tilde{\tilde{P}}_{l_1} \Delta \tilde{\tilde{\theta}}_{calc, l_1} + \tilde{\tilde{P}}_{l_2} \Delta \tilde{\tilde{\theta}}_{calc, l_2} \right) \right\| \right) \tag{3.37}
\end{aligned}$$

If $\Delta \tilde{\tilde{\theta}}_{calc, l_1} = \Delta \tilde{\tilde{\theta}}_{calc, l_2}$, which occurs if line l_1 and l_2 are in parallel, then it is possible to easily determine the values of $\tilde{\tilde{P}}_{l_1}$ and $\tilde{\tilde{P}}_{l_2}$ which result in the best match with $\Delta \theta_{observed}$. In this case, the following one dimensional minimization is performed:

$$\begin{aligned}
&\Delta \tilde{\tilde{\theta}}_{calc, l_{1,2}} = \Delta \tilde{\tilde{\theta}}_{calc, l_1} = \Delta \tilde{\tilde{\theta}}_{calc, l_2} \\
&\min_{\tilde{\tilde{P}}_{l_1, 2}} \left\| \Delta \theta_{observed} - \tilde{\tilde{P}}_{l_1, 2} \Delta \tilde{\tilde{\theta}}_{calc, l_{1,2}} \right\| = \left\| \Delta \theta_{observed} - \tilde{\tilde{P}}_{l_1, 2}^* \Delta \tilde{\tilde{\theta}}_{calc, l_{1,2}} \right\| \tag{3.38} \\
&\tilde{\tilde{P}}_{l_1, 2}^* = \frac{\Delta \tilde{\tilde{\theta}}_{calc, l_{1,2}} \cdot \Delta \theta_{observed}}{\Delta \tilde{\tilde{\theta}}_{calc, l_{1,2}} \cdot \Delta \tilde{\tilde{\theta}}_{calc, l_{1,2}}}
\end{aligned}$$

Using the $\tilde{\tilde{P}}_{l_1, 2}^*$ scaling factor, it is possible to determine the pre-outage flows on each of the parallel lines:

$$\begin{aligned}
\tilde{P}_{l,1,2}^* &= P_{l_1} + PTDF_{l_1,l_1,from-l_{1,to}} \times \tilde{P}_{l,1,2}^* + \\
&\quad P_{l_2} + PTDF_{l_2,l_2,from-l_{2,to}} \times \tilde{P}_{l,1,2}^* \\
P_{l_1} + P_{l_2} &= \tilde{P}_{l,1,2}^* \left(1 - PTDF_{l_1,l_1,from-l_{1,to}} - PTDF_{l_2,l_2,from-l_{2,to}} \right) \quad (3.39) \\
\frac{P_{l_1}}{P_{l_2}} &= \frac{B_{l_1}}{B_{l_2}} = \frac{PTDF_{l_1,l_1,from-l_{1,to}}}{PTDF_{l_2,l_2,from-l_{2,to}}} \\
\begin{bmatrix} P_{l_1} \\ P_{l_2} \end{bmatrix} &= \tilde{P}_{l,1,2}^* \frac{\left(1 - PTDF_{l_1,l_1,from-l_{1,to}} - PTDF_{l_2,l_2,from-l_{2,to}} \right)}{PTDF_{l_1,l_1,from-l_{1,to}} + PTDF_{l_2,l_2,from-l_{2,to}}} \begin{bmatrix} PTDF_{l_1,l_1,from-l_{1,to}} \\ PTDF_{l_2,l_2,from-l_{2,to}} \end{bmatrix}
\end{aligned}$$

The last expression is based on the fact that the same angle difference exists across both lines, so the flows must be in proportion to their reactances. Figure 3.13 provides a graphical depiction of the relevant quantities used to derive the equations in (3.39) which model a parallel line outage with power injections.

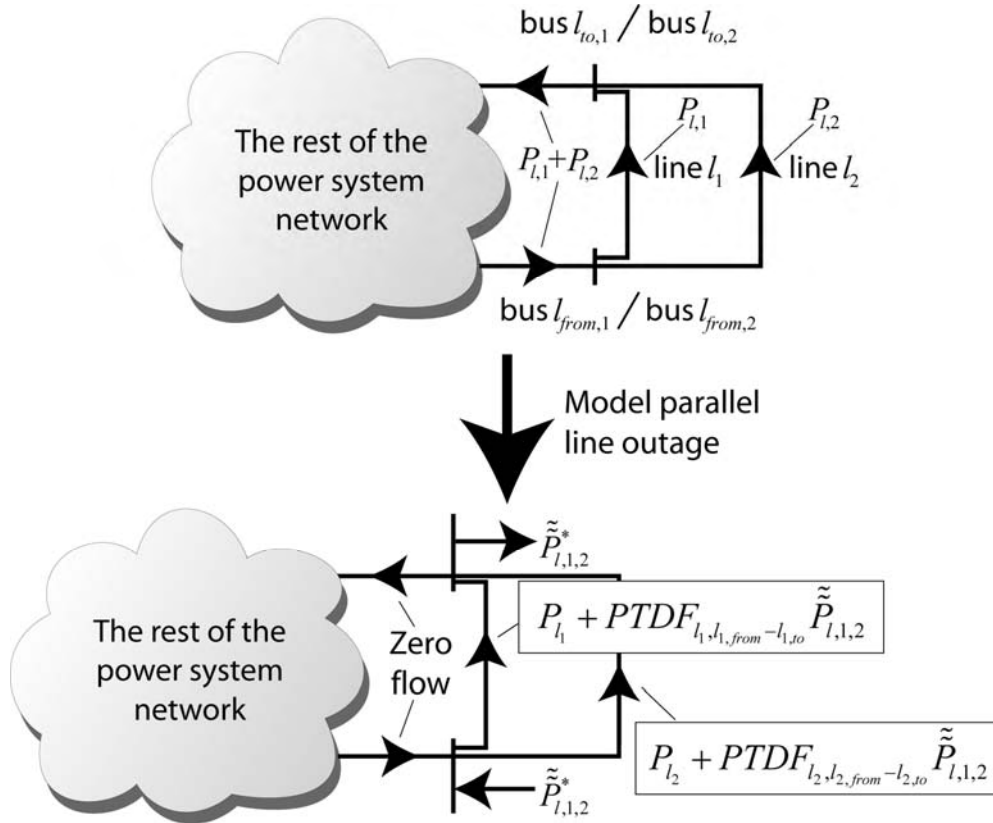


Figure 3.13: Modeling the outage of two parallel lines with a power transfer

If the outage of the parallel lines results in island formation, then $\left(1 - PTDF_{l_1,l_1,from-l_{1,to}} - PTDF_{l_2,l_2,from-l_{2,to}} \right)$ will be equal to zero and modeling of the double line outage must account for changes in generation due to islanding.

For cases where the lines are not in parallel, the minimization in (3.37) can be solved as a least squares minimization:

$$\begin{aligned}
\begin{bmatrix} \tilde{P}_{l,1}^* \\ \tilde{P}_{l,2}^* \end{bmatrix} &= \arg \min_{\tilde{P}_{l,1}, \tilde{P}_{l,2}} \left\| \Delta \mathbf{\theta}_{observed} - \begin{bmatrix} \Delta \tilde{\mathbf{\theta}}_{calc,l_1} & \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \end{bmatrix} \begin{bmatrix} \tilde{P}_{l,1} \\ \tilde{P}_{l,2} \end{bmatrix} \right\| \\
\begin{bmatrix} \tilde{P}_{l,1}^* \\ \tilde{P}_{l,2}^* \end{bmatrix} &= \left(\begin{bmatrix} (\Delta \tilde{\mathbf{\theta}}_{calc,l_1})^T \\ (\Delta \tilde{\mathbf{\theta}}_{calc,l_2})^T \end{bmatrix} \begin{bmatrix} \Delta \tilde{\mathbf{\theta}}_{calc,l_1} & \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \end{bmatrix} \right)^{-1} \begin{bmatrix} \Delta \tilde{\mathbf{\theta}}_{calc,l_1} \cdot \Delta \mathbf{\theta}_{observed} \\ \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \cdot \Delta \mathbf{\theta}_{observed} \end{bmatrix} \\
&= \begin{bmatrix} \Delta \tilde{\mathbf{\theta}}_{calc,l_1} \cdot \Delta \tilde{\mathbf{\theta}}_{calc,l_1} & \Delta \tilde{\mathbf{\theta}}_{calc,l_1} \cdot \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \\ \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \cdot \Delta \tilde{\mathbf{\theta}}_{calc,l_1} & \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \cdot \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \end{bmatrix}^{-1} \begin{bmatrix} \Delta \tilde{\mathbf{\theta}}_{calc,l_1} \cdot \Delta \mathbf{\theta}_{observed} \\ \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \cdot \Delta \mathbf{\theta}_{observed} \end{bmatrix} \quad (3.40) \\
&= \frac{1}{\left\| \Delta \tilde{\mathbf{\theta}}_{calc,l_1} \right\|^2 \left\| \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \right\|^2 - \left(\Delta \tilde{\mathbf{\theta}}_{calc,l_1} \cdot \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \right)^2} \times \\
&\quad \begin{bmatrix} \left\| \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \right\|^2 & -\Delta \tilde{\mathbf{\theta}}_{calc,l_1} \cdot \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \\ -\Delta \tilde{\mathbf{\theta}}_{calc,l_2} \cdot \Delta \tilde{\mathbf{\theta}}_{calc,l_1} & \left\| \Delta \tilde{\mathbf{\theta}}_{calc,l_1} \right\|^2 \end{bmatrix} \begin{bmatrix} \Delta \tilde{\mathbf{\theta}}_{calc,l_1} \cdot \Delta \mathbf{\theta}_{observed} \\ \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \cdot \Delta \mathbf{\theta}_{observed} \end{bmatrix}
\end{aligned}$$

Using the optimal injections defined in (3.40), the optimization of (3.37) can be simplified further:

$$\begin{aligned}
&\text{lines outaged } \{l_1^*, l_2^*\} = \\
&\arg \min_{\{l_1, l_2\} \in \{1, 2, \dots, L\} \times \{1, 2, \dots, L\}} \left(\left\| \Delta \mathbf{\theta}_{observed} - \left(\tilde{P}_{l,1}^* \Delta \tilde{\mathbf{\theta}}_{calc,l_1} + \tilde{P}_{l,2}^* \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \right) \right\| \right) \quad (3.41)
\end{aligned}$$

The normalized angle distance (*NAD*) metric can also be extended to double line outage events as follows:

$$\begin{aligned}
\theta_{l_1, l_2} &= \cos^{-1} \left(\frac{\Delta \mathbf{\theta}_{observed}}{\left\| \Delta \mathbf{\theta}_{observed} \right\|} \cdot \frac{\left(\tilde{P}_{l,1}^* \Delta \tilde{\mathbf{\theta}}_{calc,l_1} + \tilde{P}_{l,2}^* \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \right)}{\left\| \left(\tilde{P}_{l,1}^* \Delta \tilde{\mathbf{\theta}}_{calc,l_1} + \tilde{P}_{l,2}^* \Delta \tilde{\mathbf{\theta}}_{calc,l_2} \right) \right\|} \right) \quad (3.42) \\
NAD_{l_1, l_2} &= 2 \sin \theta_{l_1, l_2}
\end{aligned}$$

With the error metric properly defined, the algorithm can be defined for double line outage event detection which closely aligns with those for single line and generator outage events.

3.4.2 Basic double line outage event definition algorithm

In order to detect double line outage events, identify the outaged lines, and determine the pre-outage flows on the lines, the following basic algorithm is used:

1. For each line $l_1 \in [1, L]$
 - a. Calculate $\Delta \tilde{\theta}_{calc, l_1}$ using (3.36).
 - b. For each line $l_2 \in [l_1 + 1, L]$
 - i. Calculate $\Delta \tilde{\theta}_{calc, l_2}$ using (3.36).
 - ii. Calculate $\tilde{P}_{l_1}^*$ and $\tilde{P}_{l_2}^*$ using (3.40) for non-parallel lines or $\tilde{P}_{l_1, 2}^*$ using (3.38) for parallel lines
 - iii. Calculate the *NAD* value for the double outage $\{l_1, l_2\}$ using (3.42), then store the calculated error value in the indexed array *NADVals*:

$$NADVals_{\{l_1, l_2\}} = NAD_{l_1, l_2} \quad (3.43)$$

2. Determine the lines $\{l_1^*, l_2^*\}$ that were outaged by sorting *NADVals*:

$$\{l_1^*, l_2^*\} = \arg \min_{l_1, l_2} NADVals_{\{l_1, l_2\}} \quad (3.44)$$

3. Determine the pre-outage flows on the lines which correspond to the optimal double outage model injections using (or (3.39) for parallel lines):

$$\begin{bmatrix} P_{l_1, 1}^* \\ P_{l_1, 2}^* \end{bmatrix} = \begin{bmatrix} 1 - PTDF_{l_1, l_1, from -l_1, to} & -PTDF_{l_1, l_2, from -l_2, to} \\ -PTDF_{l_2, l_1, from -l_1, to} & 1 - PTDF_{l_2, l_2, from -l_2, to} \end{bmatrix} \begin{bmatrix} \tilde{P}_{l_1, 1}^* \\ \tilde{P}_{l_1, 2}^* \end{bmatrix} \quad (3.45)$$

3.4.3 Computational complexity

The extension of events from single to double line outages can greatly increase both storage and computational demands. The key difference between evaluating single and double line outages is the number of potential events. For single line outage events, only L events need to be considered, if L is the number of lines which are currently in service. On the other hand, for double line outage, the number of considered events is $\binom{L}{2}$, or $[L(L-1)]/2$. For a typical system, with lines numbering in the tens of thousands, this can represent a tremendous increase in the amount of work needed to find the minimizing event.

3.4.4 Heuristic modifications to the basic algorithm

There are several ways to reduce the dimensionality of the search space, with a tradeoff between consideration of improbable events and reduction in storage and computational burden. One method is to filter the event set to only include outages with probability above a certain threshold $Prob_{threshold}$:

$$\text{lines outaged } \{l_1^*, l_2^*\} = \arg \min_{\{l_1, l_2\} \in \{1, 2, \dots, L\} \times \{1, 2, \dots, L\}, Prob(\{l_1, l_2\}) > Prob_{threshold}} (\dots) \quad (3.46)$$

The probability data needed to execute the filtering in (3.46) should be based on detailed modeling or operational data. Unfortunately, this data is currently unavailable over the wide area; however, the need for this data to improve event identification could serve as a motivating factor to improve the collection and distribution of outage statistics.

A second way of filtering the event set is to account for the geographic proximity of two lines on the system. Simultaneous line outages have a tendency to occur in shared rights of way [71], and the wide availability of line locations makes this option more feasible than (3.46). The basic formulation of this filtering method is:

$$\text{lines outaged } \{l_1^*, l_2^*\} = \arg \min_{\{l_1, l_2\} \in \{1, 2, \dots, L\} \times \{1, 2, \dots, L\}, d(l_1, l_2) < d_{\text{threshold}}} (\dots) \quad (3.47)$$

where a distance function is defined between any two lines, $d(l_1, l_2)$, and this distance must be below a threshold value of $d_{\text{threshold}}$ for the two lines to be considered together. Because line locations rarely change at the transmission level, the determination of $d(l_1, l_2)$ for each pair-wise combination of lines can be stored and used for long periods of time.

One simple but conservative distance metric that can be used for $d(l_1, l_2)$ is:

$$\begin{aligned} (x_i, y_i) &= \arg \min_{(\bar{x}_i, \bar{y}_i)} \left\{ R_i(\bar{x}_i, \bar{y}_i) = \max_{\text{all points } (x, y) \in \text{line } i} \|(\bar{x}_i, \bar{y}_i) - (x, y)\| \right\} \\ d_{\text{circ}}(l_1, l_2) &= \max \left\{ \left\| (x_{l_1}, y_{l_1}) - (x_{l_2}, y_{l_2}) \right\| - R_1(x_{l_1}, y_{l_1}) - R_2(x_{l_2}, y_{l_2}), \right. \\ &\quad \left. 0 \right\} \end{aligned} \quad (3.48)$$

A visualization of this distance function is provided in Figure 3.14. This formulation provides a lower bound on the point-wise distance between any two points in either line. The first equation in (3.48) represents the solution of the minimum enclosing circle (MEC) problem, with the line vertices as the points to be enclosed. This is a well-studied problem, and algorithms exist which can calculate (x_i, y_i) and the associated radius in $O(n_i)$ time, where n_i is the number of vertex points for line i [72]. Pairwise evaluation of these distances can be done very quickly once the (x_i, y_i) and R_i values are known, with each distance calculation requiring only three add and two multiply operations.

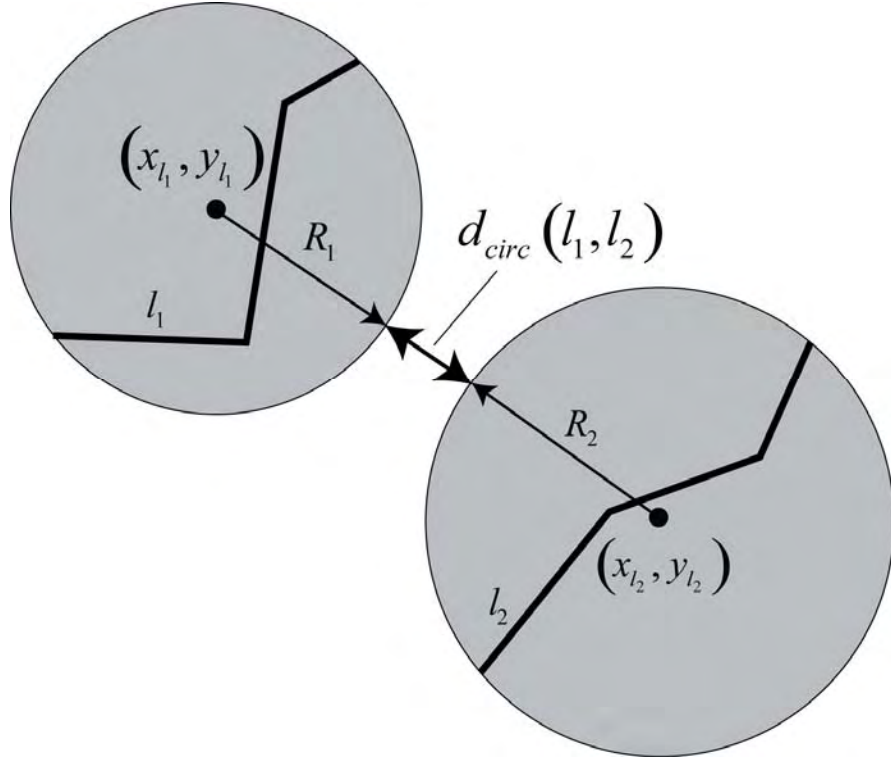


Figure 3.14: Distance between two lines, as defined in (3.48)

An alternative method of defining the distance between two lines is to use the minimum distance between any two points on the lines:

$$d_{min}(l_1, l_2) = \min_{\substack{\text{all points } (x_1, y_1) \in \text{line 1,} \\ \text{all points } (x_2, y_2) \in \text{line 2}}} \|(x_1, y_1) - (x_2, y_2)\| \quad (3.49)$$

This distance function is illustrated in Figure 3.15. The relationship between d_{min} and d_{circ} is that d_{circ} is a lower bound on $d_{min}(l_1, l_2)$, since any point on a line must by definition be within the MEC of the line. The relative magnitude of d_{min} and d_{circ} depends on the line configuration, but d_{min} should be substantially higher since the MEC tends to greatly enlarge the influence region of a line, particularly for long lines.

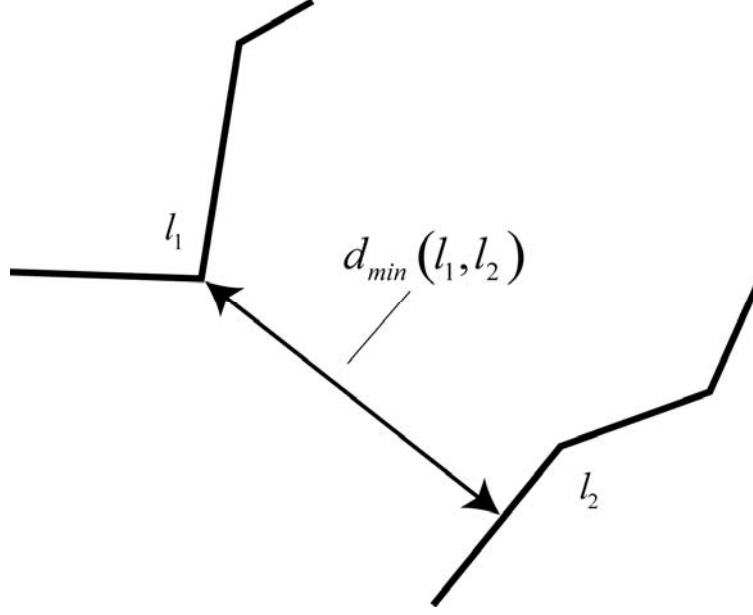


Figure 3.15: Minimum distance between two lines, as defined in (3.49)

These distances can be calculated offline and stored in a table for later retrieval. Determining what the set of considered double line outages is for a given threshold is then a matter of looking for all entries in the table below a threshold, which can be greatly accelerated by utilizing sorted lists for storage. As a result, the calculations involved in the filtering operation should be much lower than the search of the complete double outage event space and result in a net reduction in computation time and storage.

To evaluate the usefulness of the filtering method defined in (3.47) and (3.48), a large system with 4609 lines, illustrated in Figure 3.16, was used. The minimum distance between lines is zero, resulting from the case where two lines' have the same terminal bus. The maximum distance between any two lines is 1422 miles using d_{circ} and 1429 miles using d_{min} , and the mean distance is 342 miles using d_{circ} and 347 miles using d_{min} .

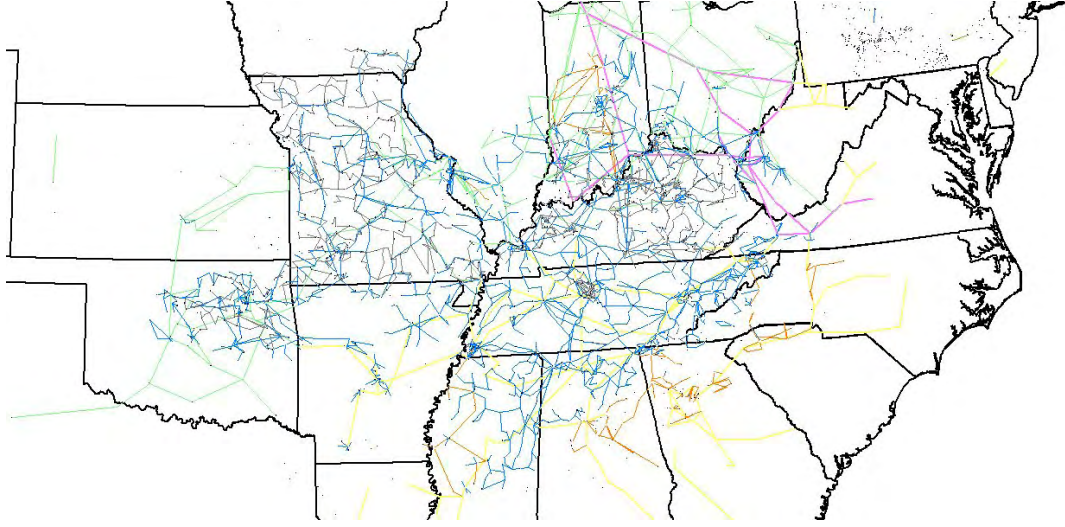


Figure 3.16: System used to test the impact of distance-based double outage filtering

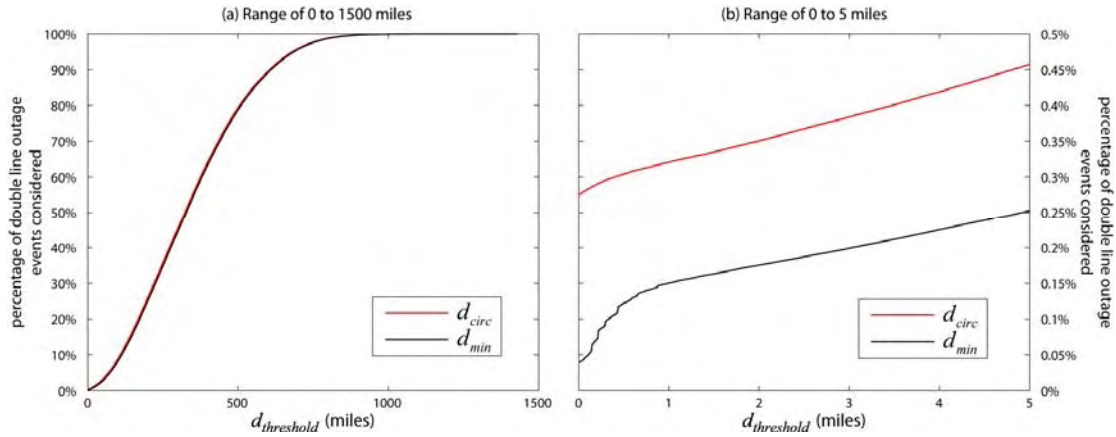


Figure 3.17: Percentage of double outage events considered as a function of the distance threshold

$d_{threshold}$ using d_{circ} and d_{min} for threshold ranges of 0-1500 miles (a) and 0-5 miles (b)

Figure 3.17 illustrates how the percentage of double line outage events filtered out by the distance function varies with the distance cutoff values ranging from 0 to 1500 miles on the left side and 0 to 5 miles on the right side. These results indicate that even a conservative threshold of 5 miles removes over 99.75% of the double outage events from consideration if the larger distance d_{min} is used. For real-time implementation, Figure 3.17 could be used to find the distance threshold corresponding to a defined number of events (e.g., if a maximum search time is specified) or the number of events corresponding to a defined distance threshold (e.g., if a maximum inter-line distance is specified).

The other heuristic applied to double line outage detection is the rejection of potential outages when the estimated pre-outage flow is above 150% of the line rating or 5 GW. This is the same heuristic used in single line outage detection and described above in section 3.2.4.

4 Event Detection Evaluation

4.1 Single line outage examples

To test the single line outage detection algorithm, two tests were performed. The first set of tests, using the system defined in 0, examines the algorithm's performance for each of the 56 single line outages on the system. The second set examines the algorithm's performance using real data from the Tennessee Valley Authority system recorded during the outage of a major 500 kV transmission line.

To evaluate single line outage detection, several performance metrics are defined. The first is an indicator function to signal whether or not a particular outage is detected by the algorithm for a given set of detection parameters:

$$AboveThreshold(l_{outaged}) = \begin{cases} 1 & , \text{outage of line } l_{outaged} \text{ detected} \\ 0 & , \text{otherwise} \end{cases} \quad (4.1)$$

where $l_{outaged}$ is the outaged line. Next, a metric is defined to represent the *NAD* value of the outaged line:

$$NADOutaged(l_{outaged}) = NADVals_{l_{outaged}} \quad (4.2)$$

This metric indicates how well the expected and observed angles match up for the line that was outaged. The rank of the outaged line in the sorted *NADVals* list is also important:

$$RankOutaged(l_{outaged}) = \left| \left\{ l : NADVals_l < NADVals_{l_{outaged}} \right\} \right| + 1 \quad (4.3)$$

Ideally, *RankOutaged* would be one, indicating that the algorithm has correctly matched the outaged line with the observed angle changes.

Taken together, these metrics define whether a line outage is detected (*AboveThreshold*), how well the observed and expected angle changes match (*NADOutaged*), and how well the expected angles from the outaged line compare with those from other lines (*RankOutaged*).

4.1.1 All 37 bus single line outages

The first set of tests was conducted with the system described in Appendix A. The 56 in-service lines were each outaged and a transient simulation was run for 5 seconds after the outage. For the transient simulation, the default simulation parameters were used from PSS/E version 31, including the default simulation time step of 1/120 seconds (i.e., a sampling rate of 120 Hz). To simulate the data that would be obtained from a phasor measurement unit, the raw

simulated data was downsampled to a sampling rate of 30 Hz. A low-pass anti-aliasing filter with a cutoff of 15 Hz and delay compensation was used to ensure that no out-of-bandwidth components were present in the simulated measurements. An example of the simulated angle measurements at each bus on the system for the outage of the line connecting buses 28 and 31 is shown in Figure 4.1.

The results of the transient simulation were then filtered using one of the methods described in chapter 2, and the observed angle changes were determined using the method described in chapter 3.1. These results were then processed using the algorithm defined in sections 3.2.2 and 3.2.4 to detect the outaged line.

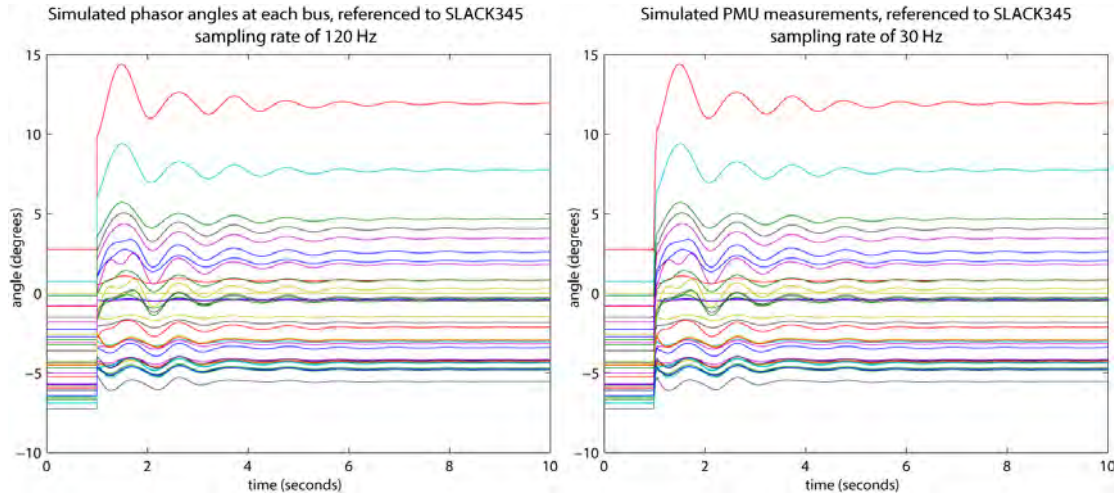


Figure 4.1: Original simulated angles (left side) and simulated PMU angle measurements (right side) for the outage of the line connecting buses 28 and 31, with all angles referenced to the SLACK345 bus angle

For each study, one or more of the following parameters were varied:

- *FilterLength*: the length of the digital signal filter, equal to N from (2.1) (for FIR filtering) or (2.4) (for median filtering)
- τ : the threshold used to determine whether an event has occurred, defined in section 3.1.1
- *PMUSet*: the set of bus numbers where measured angles are available

In addition, N_{trans} was set to *FilterLength* for FIR filtering and $\left\lfloor \frac{FilterLength}{2} \right\rfloor$ for median filtering based on the results shown in Figure 3.5.

4.1.1.1 All buses monitored

The first set of tests examines the effect of changing the threshold and filtering methods with full PMU coverage (i.e., $PMUSet = \{\text{all system buses}\}$). Because all buses are monitored, this test is designed to provide best case results for specific filter parameters and angle detection thresholds.

4.1.1.1.1 Median filtered angles

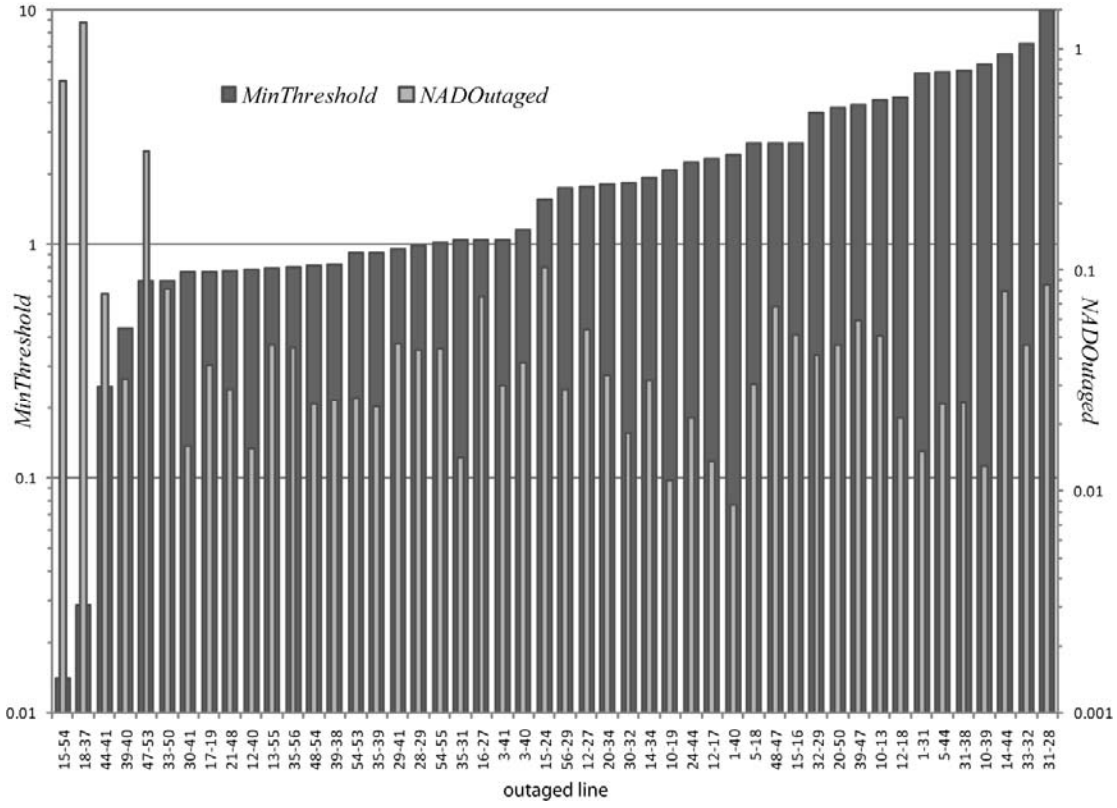


Figure 4.2: Minimum threshold values and error values for each line, ordered in ascending order by minimum threshold, using a median filter of length 31

One of the two filtering methods discussed in chapter 2 is median filtering. The first parameter considered is τ which determines how much of an angle change needs to occur before it is recognized as an event. Graphed in Figure 4.2 are the *NADOutaged* and *MinThreshold* (the minimum value of τ needed to detect the event) for each line outage on the system, sorted in ascending order by the *MinThreshold* and using a window length of 31. For parallel lines, only one of the lines is provided in this graph. A log scale is used for both y-axes due to the widely varying values of *MinThreshold* and *NADOutaged* among the set of all outaged lines. Several useful insights can be obtained by analyzing this figure. First of all, the minimum threshold value needed to detect every line outage is 0.01 degrees; however, those events which require extremely low threshold values also have the highest *NADOutaged* values, indicating that choosing the minimal threshold value will result in misclassification of these lines. To illustrate this behavior, Figure 4.3 shows the average *NADOutaged* value, taken over all line outages, for varying values of the angle threshold. The reduction in *NADOutaged* as the angle threshold is brought past 0.03 degrees, in essence ignoring the troublesome outages between buses 15-54 and 18-37, can clearly be seen in this plot and illustrates how the threshold value can impact the *NADOutaged* results.

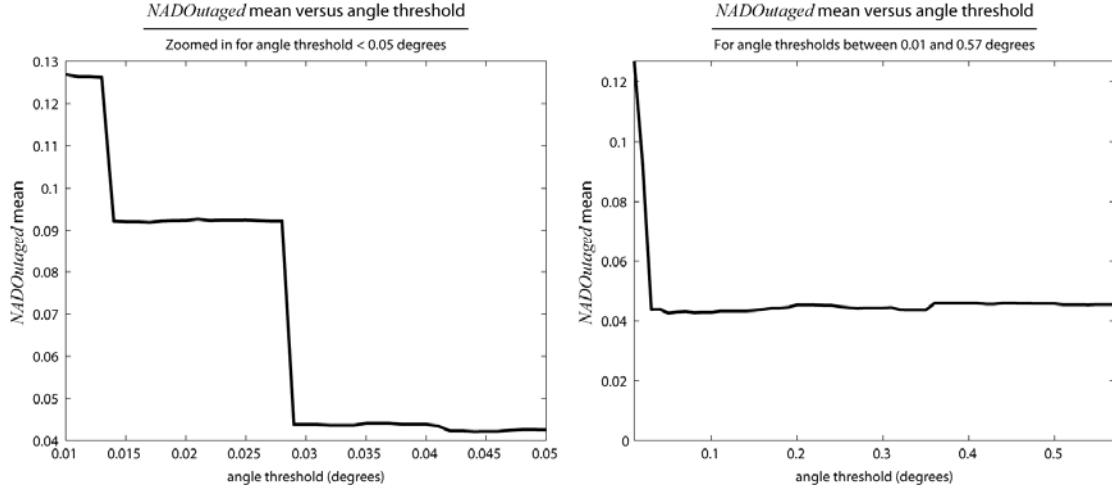


Figure 4.3: Mean *NADOutaged* values for median filtering using a window length of 31 with the angle threshold varied from 0.01 to 0.57 degrees; the left plot shows the right plot zoomed into the region $\tau \in [0.01, 0.05]$

To understand why the *NADOutaged* values are so high for the outages of lines 15-54 and 18-37, Figure 4.4 provides a detailed look at the angle changes for all system buses due to the outage of the line with the highest *NADOutaged* value, 18-37, using both ac and dc power flow solutions. Because the dc power flow assumptions are used in constructing the expected angle changes, a failure of these assumptions to hold for the outage of this line is responsible for the large *NADOutaged* value. The norm of the differences in angle changes between the ac and dc power flow solutions is 0.045 degrees for this outage, which is comparable in magnitude to the norm of the estimated angle changes under the dc power flow assumptions (0.04 degrees). One possible reason for the poor performance of the dc power flow for the outage of the 18-37 or 15-34 lines is that these lines have the highest R/X ratios on the system, with $R/X = 1.82$, and lines with high R/X ratios are known to cause errors in the accuracy of the dc power flow.

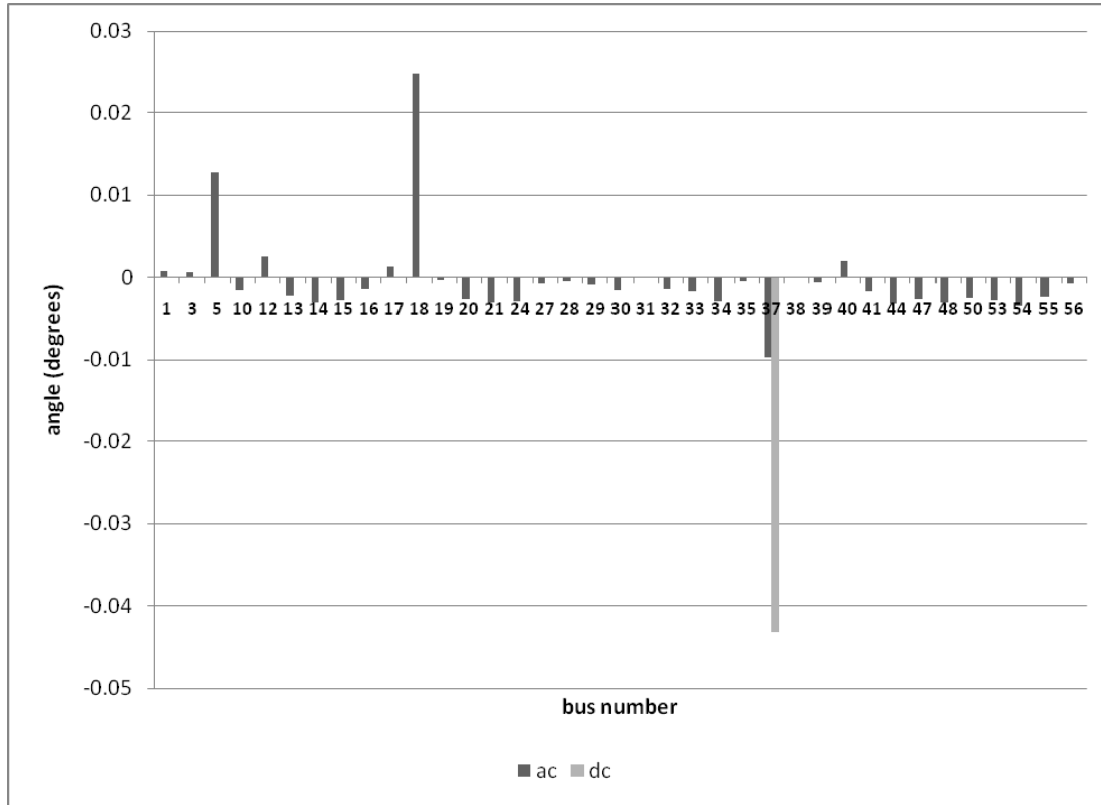


Figure 4.4: Calculated changes in angles at all system buses due to the outage of a line connecting buses 18 and 37 using the ac and dc power flow

A high *NADOutaged* in and of itself does not guarantee that the algorithm is failing to properly identify the outaged line; what really matters is where a particular line outage ranks in the set of ordered *NAD* relative to all the other possible outages. Figure 4.5 shows that the rank of the line in the *NADVals* list, *RankOutaged*, is indeed closely tied to the *NADOutaged* value of the outaged line. This figure is a scatter plot of *NADOutaged* versus *RankOutaged* for a wide range of filter lengths (from 3 to 61 in increments of 2) and angle thresholds (from 0.01 to 0.57 degrees in increments of 0.01 degrees). As shown in the plot, for any detected event with an *NADOutaged* value less than 0.8, the algorithm correctly ranks the outaged line. As the true system response becomes less similar to the response predicted by the dc power flow, *NADOutaged* gets larger and the ranking of the outaged line becomes poorer, in some cases resulting in rankings as high as 43 out of 56 (for the outage of the two lines between buses 18 and 37 using window lengths of 29 or 31 with angle thresholds of 0.01 or 0.02 degrees).

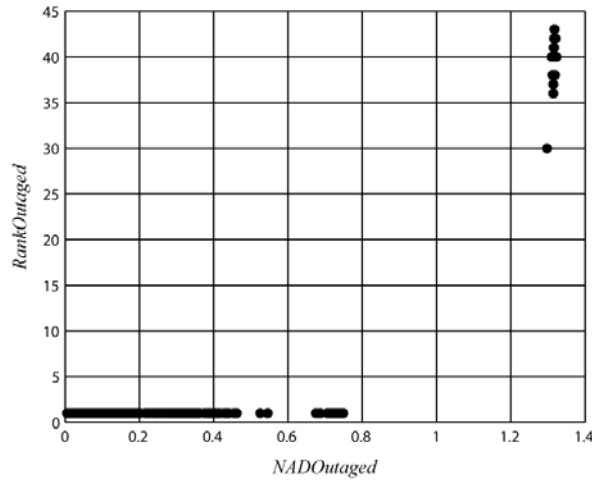


Figure 4.5: Scatter plot of *NADOutaged* versus *RankOutaged* for 84,867 detected line outages using window lengths from 3 to 61 in increments of 2 and angle thresholds from 0.01 to 0.57 degrees in increments of 0.01 degrees.

In addition to the effect of the angle threshold on successful line outage detection, the window length can also have an effect on the value of *NADOutaged* by changing the attenuation of oscillations and the estimation of the steady state angle changes. Figure 4.6 shows the relationship between the median window length and the average *NADOutaged* value using a constant angle threshold value of 0.05 degrees. There is a sharp decrease in the *NADOutaged* mean as the filter length is increased from 3 to 19, then a more gradual decrease as the filter length is increased further. This corresponds to the results shown in Figure 3.5, where $\Delta\theta_{miss}$ shows a rapid decay for small window lengths and a more gradual decay for larger window lengths.

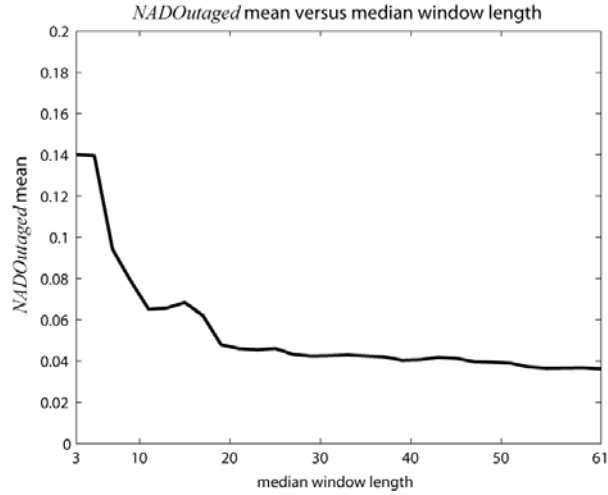


Figure 4.6: Effect of median window length on the mean *NADOutaged* value using an angle threshold of 0.05 degrees

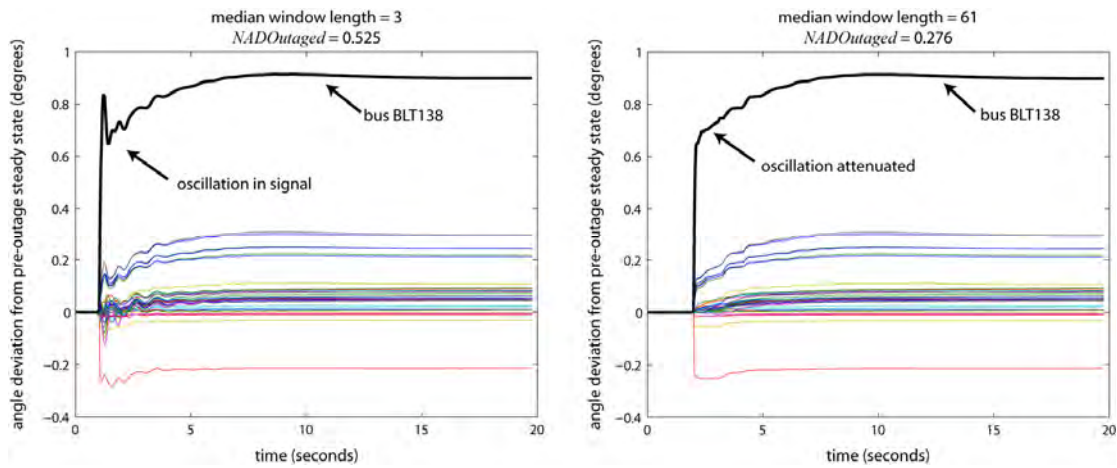


Figure 4.7: Elimination of an oscillation in the BLT138 bus angle signal for the outage of line 47-53 as the median window length is changed from 9 to 41

The decrease in *NADOutaged* as the window length is increased is due to improved attenuation. As in illustrative example, Figure 4.7 shows an oscillation in the angle signal of bus BLT138 that is eliminated as the median window length changes from 9 to 41.

4.1.1.1.2 FIR filtered angles

The same outage cases were also analyzed using FIR filtering to determine the angle change vectors. A graph analogous to that of Figure 4.2 showing the *MinThreshold* and *NADOutaged* values for each of the lines in the system is provided in Figure 4.8. The same problem at the low end of the graph can be seen—lines which require very low threshold values have high *NADOutaged* values. This shows that the disparity between the ac and dc power flow solutions

(e.g., as shown in Figure 4.4) cannot be compensated for by changing the filtering method and is a fundamental limitation due to the usage of the dc power flow equations to detect the outage event.

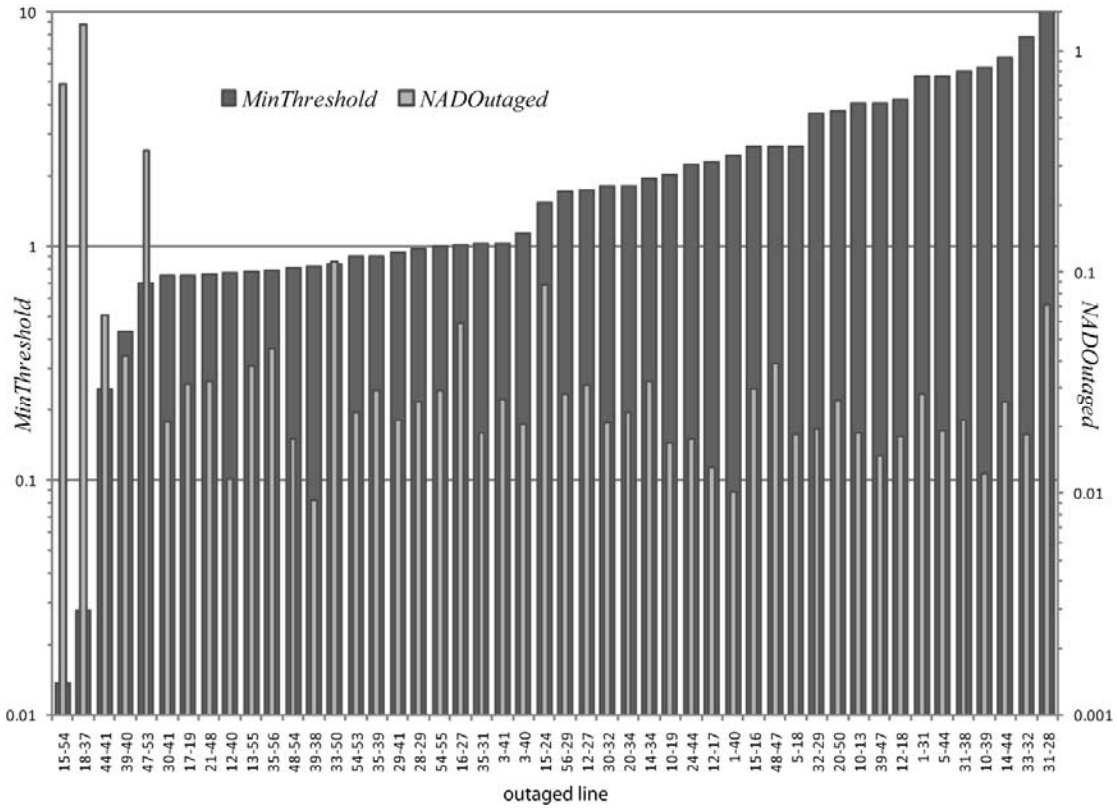


Figure 4.8: Minimum threshold values and error values for each line, ordered in ascending order by minimum threshold, using an FIR filter of order 31

The usage of higher order FIR filters also has a minimizing effect on the *NADOutaged* mean, as illustrated in Figure 4.9. Of particular note is that there are no portions of the curve in which the *NADOutaged* mean increases as the FIR filter order increases. Comparing this to Figure 4.6, the inverse relationship between filter length and *NADOutaged* mean is more consistent for FIR filtering than median filtering. One reason for this behavior is that FIR filtering attenuates better at all oscillation frequencies above the cutoff frequency as the filter order is increased (see Figure 2.15), whereas this is not necessarily the case for median filtering (see Figure 2.16).

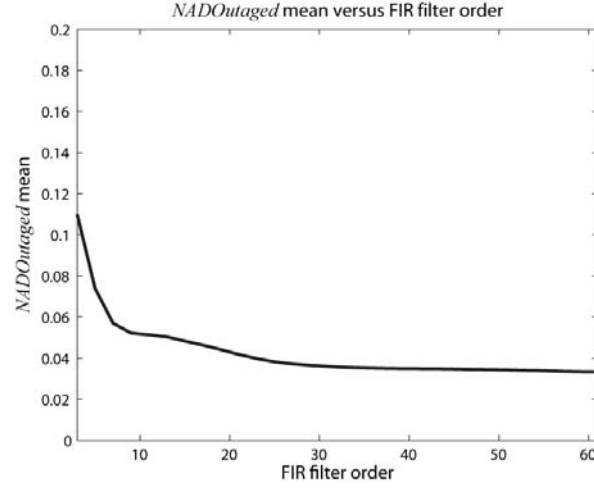


Figure 4.9: Effect of FIR filter order on the mean *NADOutaged* value using an angle threshold of 0.05 degrees

The scatter plot of *NADOutaged* versus *RankOutaged* in Figure 4.10 shows that the relationship between these two values obtained from median filtering also holds if FIR filtering is used, indicating that the algorithm is more sensitive to the accuracy of the dc power flow assumptions than the filtering method for the case where all bus angles are monitored.

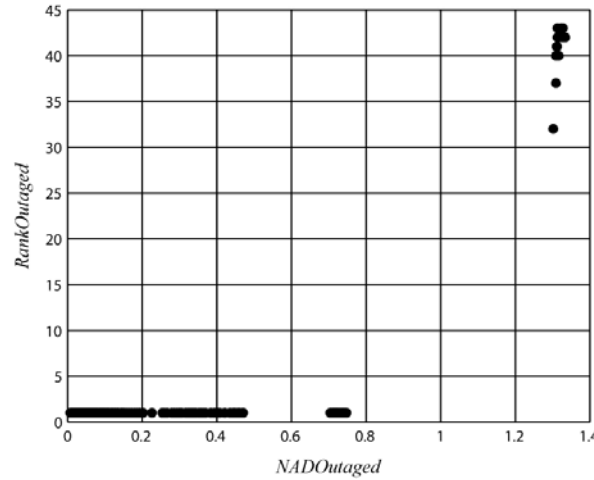


Figure 4.10: Scatter plot of *NADOutaged* versus *RankOutaged* for 86,438 detected line outages using filter orders from 3 to 61 and angle thresholds from 0.01 to 0.57 degrees.

4.1.1.2 Eighteen buses monitored

To examine how the algorithm performs with lower PMU deployment, additional tests were run with only half the buses monitored, using:

$$PMUSet = \{3, 10, 13, 15, 17, 19, 21, 27, 29, 31, 33, 35, 38, 40, 44, 48, 53, 55\}.$$

Figure 4.11 shows the *NADOutaged* versus *RankOutaged* scatter plot for FIR and median filtering using the same parameter range used to construct Figure 4.5 and Figure 4.10. The outliers from the scatter plot using full PMU placement are no longer on the plot because the reduced PMU set does not result in detection of the problematic line outages. Of all the detected events, only 166 have rank not equal to one, and all 166 of these misrankings are with the outage of line 33-50.

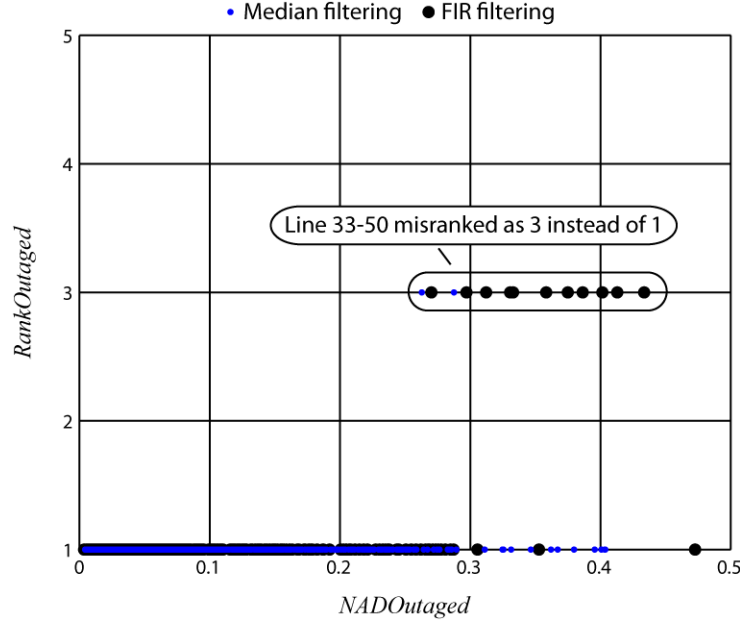


Figure 4.11: Scatter plot of *NADOutaged* versus *RankOutaged* for 82,289 detected line outages (FIR filtering) and 81,846 detected line outages (median filtering) using filter lengths from 3 to 61 and angle thresholds from 0.01 to 0.57 degrees.

Figure 4.12 shows the *RankOutaged* values for all instances in which the outage of line 33-50 is detected with FIR and median filtering. Each blue dot represents a combination of filter order and threshold such that the ranking of the 33-50 line in the *NADVals* array is 3, and each green dot represents the combination such that the ranking is 1. The areas without dots represent the cases where the event was not detected because no candidate signals exceeded the threshold angle τ . This figure demonstrates that setting the angle threshold to a low value can result in misclassification of lines by the algorithm. In this case, the very small amount of flow on line 33-50 before the outage, 2.3 MW, results in small observed angle changes which are difficult to classify.

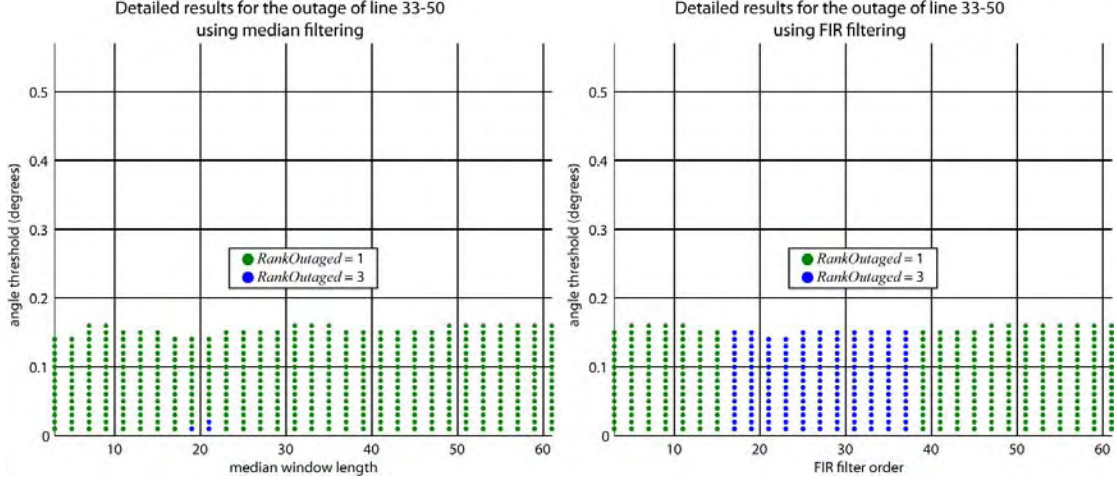


Figure 4.12: Detailed *RankOutaged* distribution for the outage of line 33-50 using FIR filter orders of 3-61 and angle thresholds of 0.01 to 0.57.

One additional result of changing from having a PMU at every bus to having a PMU at 18 buses is how low of an *NADOutaged* is needed to guarantee correct ranking of the line, referred to as *NADOutagedMin*. A higher value of *NADOutagedMin* means that more confidence can be placed in the detection algorithm results if the *NADOutaged* value for the detected line is much lower than *NADOutagedMin*. For median filtering, the *NADOutagedMin* is 0.867 for complete PMU coverage but drops to 0.252 in the case where only 18 PMUs are placed on the system. This is expected, since having more information about the system should allow for more confidence to be placed in the results based on that information. Using FIR filtering gives similar results—*NADOutagedMin* is 0.254 in the case where only 18 PMUs are placed on the system and 0.8403 for a full PMU placement.

The biggest difference between the performance of the algorithm using full PMU coverage and half PMU coverage is in the number of lines which are distinguishable with respect to the algorithm. Two line outages are indistinguishable with respect to the algorithm if the $\Delta\tilde{\theta}_{calc,l}$ values calculated using (3.8) are scalar multiples of each other, i.e.,

$$\exists \alpha \in \mathbb{R}, \Delta\tilde{\theta}_{calc,l_1} = \alpha \Delta\tilde{\theta}_{calc,l_2} \quad (4.4)$$

One way this can occur is if two lines are in parallel. For the 37-bus case, an example where this occurs is with the two parallel lines connecting buses 21 (WOLEN69) and 48 (BOB69). The outage of either of these lines is modeled as an injection and withdrawal at bus 21 and 48, respectively, as discussed in section 3.2.1 above. As a result, the $\Delta\tilde{\theta}_{calc,l}$ values calculated for the outage of either line will be the same, regardless of the PMU deployment on the system. For full PMU deployment, as considered in section 4.1.1.1, this is the only situation in which two lines will satisfy (4.4). This is due to the fact that the $\mathbf{K}$ matrix used in (3.8) is the $N \times N$ identity matrix when all buses are monitored.

As the number of PMUs on the system is reduced, there are situations where non-parallel lines still satisfy (4.4). In particular, this can occur when there is an unmonitored portion of the system connected to only one or two PMU-monitored buses.

The first case considered is the where there are two boundary buses. In Figure 4.13, the “Buses not monitored with PMUs” region is eliminated and replaced by loads at the boundary buses equal to the flows out of the boundary bus. The systems on the left and right of Figure 4.13 result in the same angles at the monitored buses.

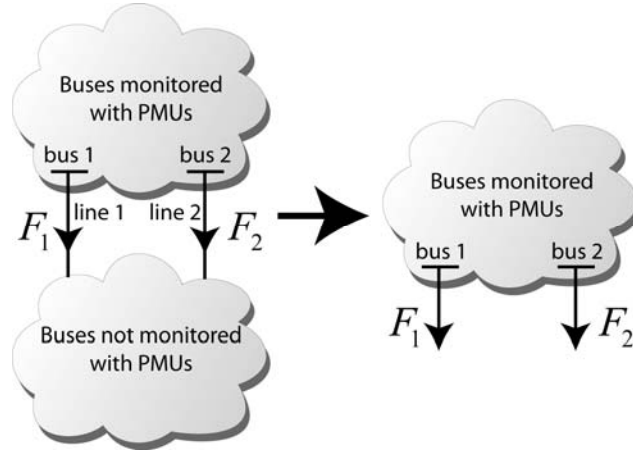


Figure 4.13: Reducing the full system to only buses that are monitored with PMUs

To see why this is true, first the power flow equations for the left system are given:

$$\begin{aligned}
 Buses_m &= \{\text{monitored buses}\}, Buses_u = \{\text{unmonitored buses}\} \\
 \forall \text{ buses } i \in Buses_m, \quad &\sum_{\substack{j \omega i, \\ j \in Buses_m \\ j \neq i}} F_{i,j}(\theta_m) + \sum_{\substack{j \omega i, \\ j \in Buses_u \\ j \neq i}} F_{i,j}(\theta_m, \theta_u) = P_{inj,i} \\
 \forall \text{ buses } k \in Buses_u, \quad &\sum_{\substack{j \omega k, \\ j \in Buses_u \\ j \neq k}} F_{k,j}(\theta_u) + \sum_{\substack{j \omega k, \\ j \in Buses_m \\ j \neq k}} F_{k,j}(\theta_m, \theta_u) = P_{inj,k}
 \end{aligned} \tag{4.5}$$

where $x \omega y$ means there is a line connecting buses x and y , $F_{i,j}$ is the power flow from bus i to bus j , θ_m is the set of angles at the monitored buses, θ_u is the set of angles at the unmonitored buses, and $P_{inj,i}$ is the power injected into bus i (positive for generators and negative for loads). Referring to Figure 4.13, the power flow equations at the monitored buses can be rearranged so that θ_u is eliminated and replaced by the boundary flow:

$$\begin{aligned}
& \forall \text{ buses } i \in \{Buses_m \setminus \{1, 2\}\}, \sum_{\substack{j \in Buses_m \\ j \neq i}} F_{i,j}(\theta_m) = P_{inj,i} \\
& \sum_{\substack{j \in Buses_m \\ j \neq 1}} F_{1,j}(\theta_m) + \sum_{\substack{j \in Buses_u \\ j \neq 1}} F_{1,j}(\theta_m, \theta_u) = \sum_{\substack{j \in Buses_m \\ j \neq 1}} F_{1,j}(\theta_m) + F_1 = P_{inj,1} \quad (4.6) \\
& \sum_{\substack{j \in Buses_m \\ j \neq 2}} F_{2,j}(\theta_m) + \sum_{\substack{j \in Buses_u \\ j \neq 2}} F_{2,j}(\theta_m, \theta_u) = \sum_{\substack{j \in Buses_m \\ j \neq 2}} F_{2,j}(\theta_m) + F_2 = P_{inj,2}
\end{aligned}$$

Because equation (4.6) constitutes the power flow equations for the right-side system in Figure 4.13, the power flow solution θ_m is the same for both the left and right systems.

If an event occurs on the unmonitored system which results in changes to the boundary flows, the external system can still be removed and replaced with injections using the new line flows between the two systems, as shown in Figure 4.14. Barring any net changes in power injections within the unmonitored system (e.g., if the event is a line outage or a transfer between two buses within the unmonitored system), then the sum of the power entering the unmonitored system must stay constant [70]. As a result, the sum of the changes in flows and, consequently, the sum of the changes in equivalent injections must equal zero (i.e., $\Delta F_1 = -\Delta F_2$).

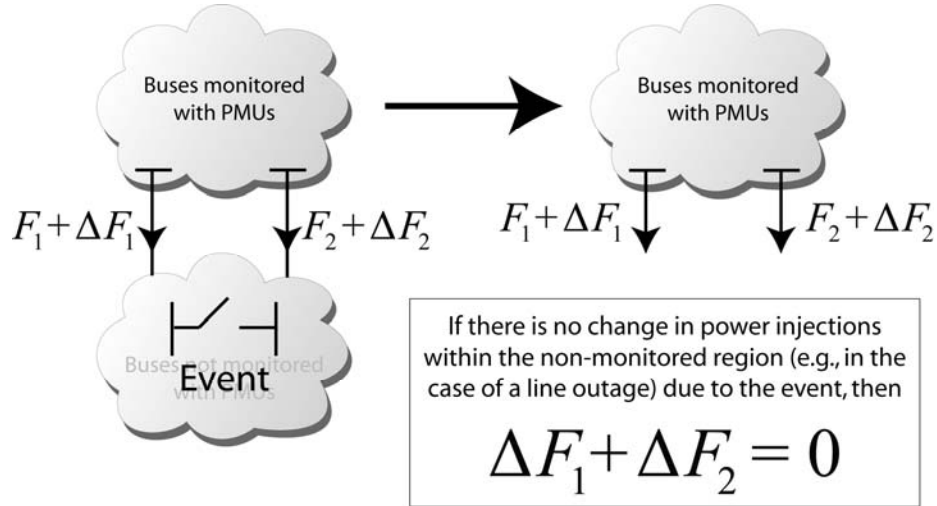


Figure 4.14: The changes in the equivalent injections due to the event will sum to zero if there is no change in power injections within the unmonitored system.

One consequence of this fact is that any two line outages within the unmonitored system will result in angle changes that are scalar multiples of one another (i.e., $\exists \alpha \in \mathbb{R}, \Delta \tilde{\theta}_{calc,l_1} = \alpha \Delta \tilde{\theta}_{calc,l_2}, l_1, l_2 \in \text{unmonitored system}$). To see why this is the case, first let $\Delta F_{1,l_1}$ be the change in flow on one of the boundary lines due to the outage of line l_1 , and let $\Delta F_{1,l_2}$ be the change in flow on one of the boundary lines due to the outage of line l_2 . Referring to Figure 4.14, the outages

can be represented by injections of $\{\Delta F_{1,l_1}, -\Delta F_{1,l_1}\}$ and $\{\Delta F_{1,l_2}, -\Delta F_{1,l_2}\}$ at the two boundary buses. The estimated change in angles within the monitored system for the outage of line l_1 can then be expressed as:

$$\begin{aligned}
\Delta \theta_{calc, l_1} &= \mathbf{KB}^{-1} \begin{bmatrix} 0 \\ \tilde{P}_{l,1} \\ -\tilde{P}_{l,1} \\ 0 \end{bmatrix} \begin{matrix} \leftarrow l_{1,from} \\ \leftarrow l_{1,to} \end{matrix} \\
&= \tilde{P}_{l,1} \Delta \tilde{\theta}_{calc, l_1} \\
&= \mathbf{KB}^{-1} \begin{bmatrix} 0 \\ \Delta F_{1,l_1} \\ -\Delta F_{1,l_1} \\ 0 \end{bmatrix} \begin{matrix} \leftarrow \text{boundary bus 1} \\ \leftarrow \text{boundary bus 2} \end{matrix} \\
&= \Delta F_{1,l_1} \mathbf{KB}^{-1} \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} \begin{matrix} \leftarrow \text{boundary bus 1} \\ \leftarrow \text{boundary bus 2} \end{matrix} \quad (4.7) \\
&= \Delta F_{1,l_1} \Delta \theta_{boundary} \\
\Delta \tilde{\theta}_{calc, l_1} &= \frac{\Delta F_{1,l_1}}{\tilde{P}_{l,1}} \Delta \theta_{boundary}
\end{aligned}$$

Similarly, the change in angles due to the outage of line l_2 can be expressed as:

$$\begin{aligned}
\Delta \theta_{calc, l_2} &= \mathbf{KB}^{-1} \begin{bmatrix} 0 \\ \tilde{\tilde{P}}_{l,2} \\ -\tilde{\tilde{P}}_{l,2} \\ 0 \end{bmatrix} \begin{matrix} \leftarrow l_{2, from} \\ \leftarrow l_{2, to} \end{matrix} \\
&= \tilde{\tilde{P}}_{l,2} \Delta \tilde{\theta}_{calc, l_2} \\
&= \mathbf{KB}^{-1} \begin{bmatrix} 0 \\ \Delta F_{1, l_2} \\ -\Delta F_{1, l_2} \\ 0 \end{bmatrix} \begin{matrix} \leftarrow \text{boundary bus 1} \\ \leftarrow \text{boundary bus 2} \end{matrix} \\
&= \Delta F_{1, l_2} \mathbf{KB}^{-1} \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} \begin{matrix} \leftarrow \text{boundary bus 1} \\ \leftarrow \text{boundary bus 2} \end{matrix} \\
&= \Delta F_{1, l_2} \Delta \theta_{boundary}
\end{aligned} \tag{4.8}$$

$$\Delta \tilde{\theta}_{calc, l_2} = \frac{\Delta F_{1, l_2}}{\tilde{\tilde{P}}_{l,2}} \Delta \theta_{boundary}$$

Combining the results from (4.7) and (4.8), the condition in (4.4) holds:

$$\begin{aligned}
\frac{\tilde{\tilde{P}}_{l,1}}{\Delta F_{1, l_1}} \Delta \tilde{\theta}_{calc, l_1} &= \Delta \theta_{boundary} = \frac{\tilde{\tilde{P}}_{l,2}}{\Delta F_{1, l_2}} \Delta \tilde{\theta}_{calc, l_2} \\
\Delta \tilde{\theta}_{calc, l_1} &= \alpha \Delta \tilde{\theta}_{calc, l_2}, \alpha = \frac{\Delta F_{1, l_1}}{\tilde{\tilde{P}}_{l,1}} \frac{\tilde{\tilde{P}}_{l,2}}{\Delta F_{1, l_2}}
\end{aligned} \tag{4.9}$$

An example of where this occurs for the study case with 18 PMUs is shown in Figure 4.15. Relating this diagram to the discussion in the previous paragraph, buses 14, 34, 20, and 50 constitute the unmonitored system, buses 33 and 44 are the boundary buses, and lines 14-44 and 33-50 are the boundary lines.

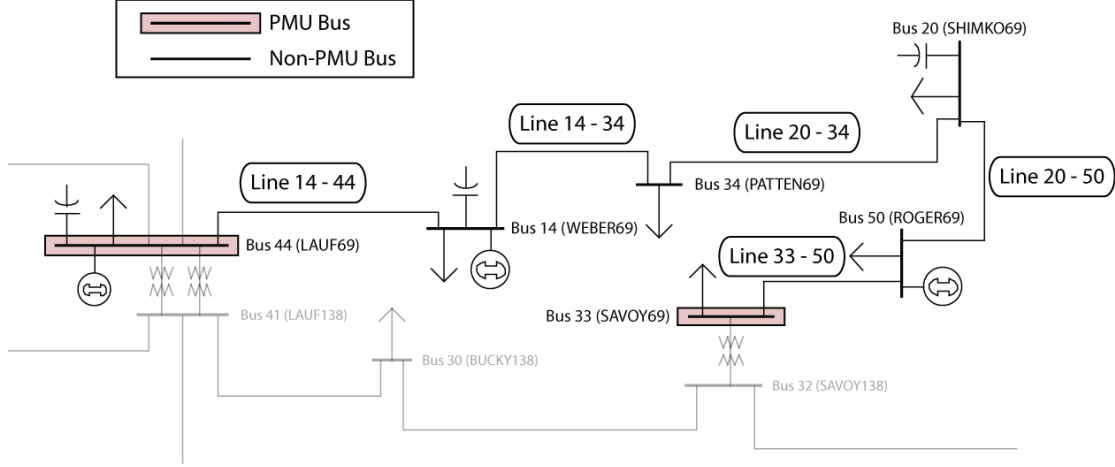


Figure 4.15: Indistinguishable line outages using the 18 PMU configuration of the 37 bus case

Table 4.1: Algorithm results for lines shown in Figure 4.15 using order 61 FIR filtering and angle threshold of 0.01 degrees

From bus	To bus	$NAD_{Outaged}$	$Rank_{Outaged}$	$SharedRank$	$FlowError$ (MW)
14	34	0.0138	1	4	0.6641
14	44	0.0636	1	4	0.8682
20	34	0.0394	1	4	0.0002
20	50	0.0349	1	4	0.6832
33	50	0.1966	1	4	0.6573

Using FIR filtering of order 61 and an angle threshold of 0.01 degrees, the algorithm is able to correctly identify each line when it goes out (i.e., $Rank_{Outaged} = 1$ for all outages) and predict the pre-outage flow accurately. Table 4.1 provides details on the algorithm's performance for the outage of each of the lines in the unmonitored system with two additional performance metrics:

$$\begin{aligned}
 Shared_{Outaged}(l_{outaged}) &= \left| \left\{ l : NADVals_l = NADVals_{l_{outaged}} \right\} \right| - 1 \\
 FlowError(l_{outaged}) &= P_{l_{outaged}}^{(estimated \text{ by algorithm})} - P_{l_{outaged}}^{(from case)}
 \end{aligned} \tag{4.10}$$

Ideally, *SharedOutaged* would be zero (indicating that the outaged line is distinguished from all other lines) and *FlowError* would also be zero (indicating that the algorithm predicted the pre-outage flow perfectly). The *FlowError* values in Table 4.1 are very low, and the *SharedOutaged* confirms that the topology results in each line being indistinguishable from the other with respect to the algorithm. One way to mitigate the impact of having several lines that are indistinguishable is to present the results by highlighting all lines which have *RankOutaged* equal to one on the one-line diagram. This would still provide very useful information; namely, the subset of the lines on the system that contains the outage line.

For the case where there is only one boundary bus between the monitored and unmonitored system (i.e., the unmonitored system is radially connected to the monitored system), the change in flow on the boundary lines will be zero if the net power injections within the unmonitored system do not change (e.g., for a line outage within the unmonitored system). Therefore, the change in equivalent injection representing the unmonitored system is zero and the angles at the PMU-monitored buses do not change. One instance of this topology for the 37 bus case is shown in Figure 4.16. Because bus 37 is only connected to the rest of the system through one bus, unless there is a PMU measurement from bus 37 there will be no estimated change in the estimated system angles ($\Delta\tilde{\theta}_{calc,l} = 0$). This is also shown in Figure 4.4 above, where the angle changes obtained by running the dc power flow solution after the outage of one of the lines connecting buses 18 and 37 are shown to be zero for all buses besides bus 37.

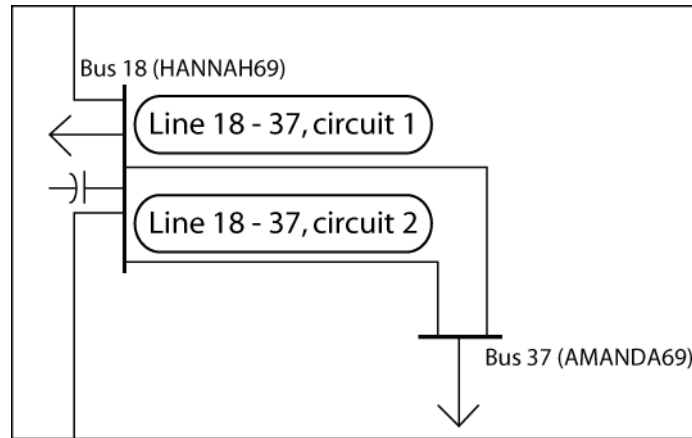


Figure 4.16: Instance of a radial system (single boundary bus) from the 37 bus case

The inability of the algorithm to detect radial lines without monitoring of the radial buses can be significant. For instance, if one of the lines between buses 18 and 37 goes out, it would be very helpful for the operator to know that there is only one line left to serve the load at bus 37. However, to allow the algorithm to detect one of the lines going out, a PMU must be placed at bus 37. This fact can help guide system planners in determining where to place new PMUs on the system, particularly if there is a known issue with reliability of one of the lines.

4.1.2 TVA system line outage

The second event tested is based on real measurements from the Tennessee Valley Authority (TVA) system after the outage of a 500 kV transmission line carrying approximately 1000 MW of power. The buses where PMU angle signals are shown in Figure 4.17. The unfiltered PMU signals obtained from these PMUs are shown in Figure 4.18, with the 8CORDOVA bus angle taken as the reference.

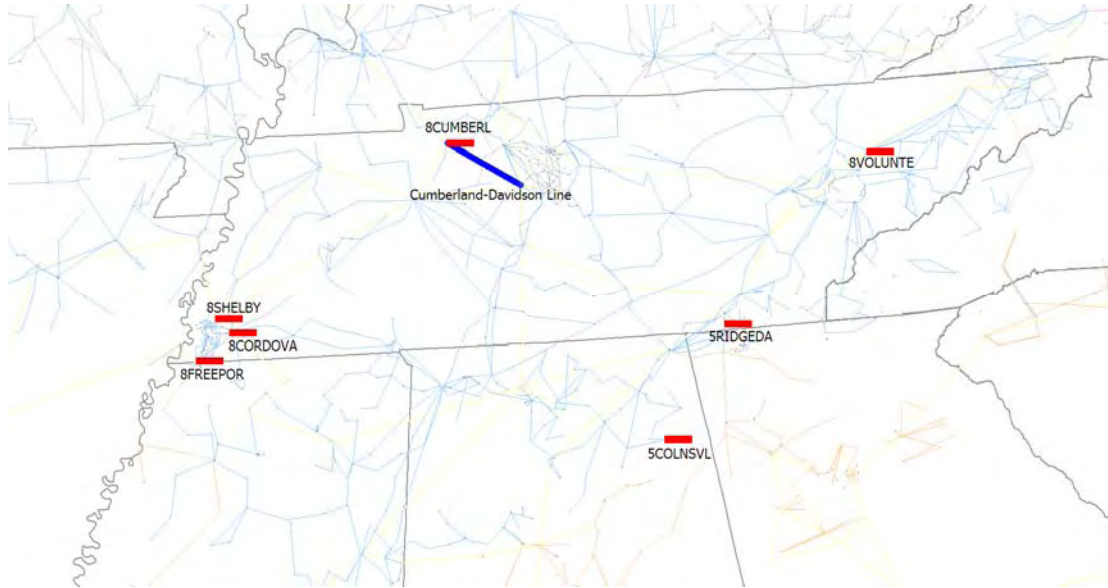


Figure 4.17: Location of PMU angle measurements on TVA system and location of the line outage

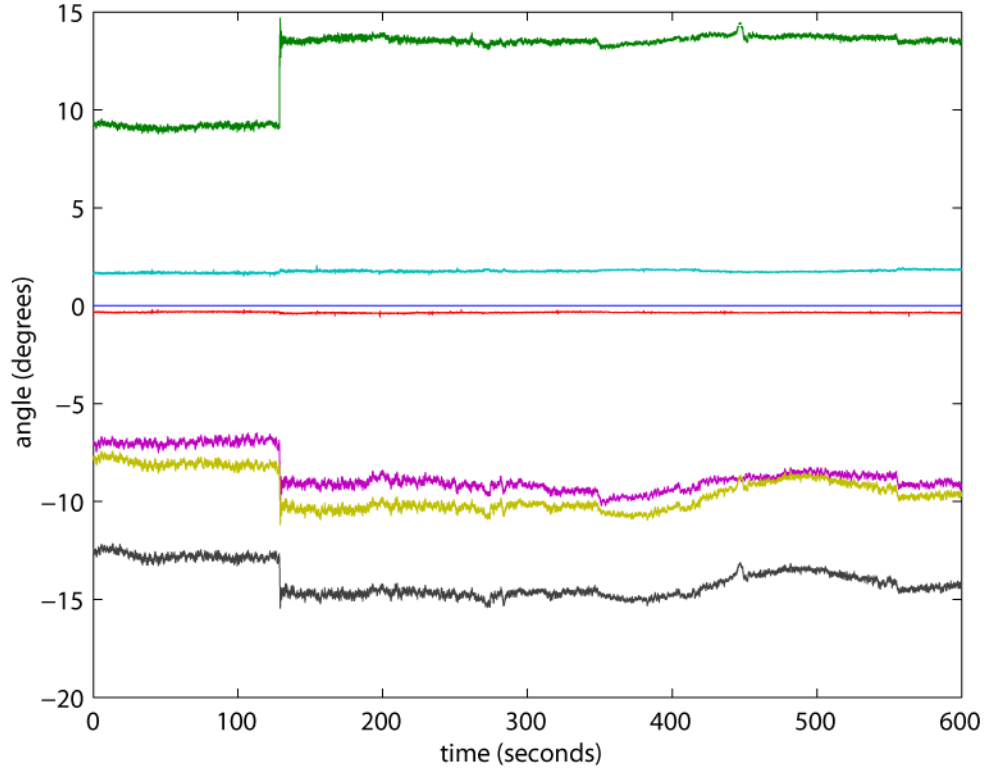


Figure 4.18: Unfiltered angle measurements from before and after the line outage

As with the 37 bus outages, an extensive sweep of filter type, filter length, and angle threshold was conducted to test the algorithm with this event. Figure 4.19 shows the *RankOutaged* values for FIR and median filtering using filter lengths in the range of 3 to 401 and τ values between 0.1 and 0.5 degrees. Using a larger value of τ results in an improved ranking for a given filter length due to the reduced interference of noise in determining the sample n_{max} which is in turn used to determine $\Delta\theta_{observed}$. The median filter also performs much poorer than FIR filtering for these signals, primarily due to the lack of noise attenuation (as shown in Figure 4.20).

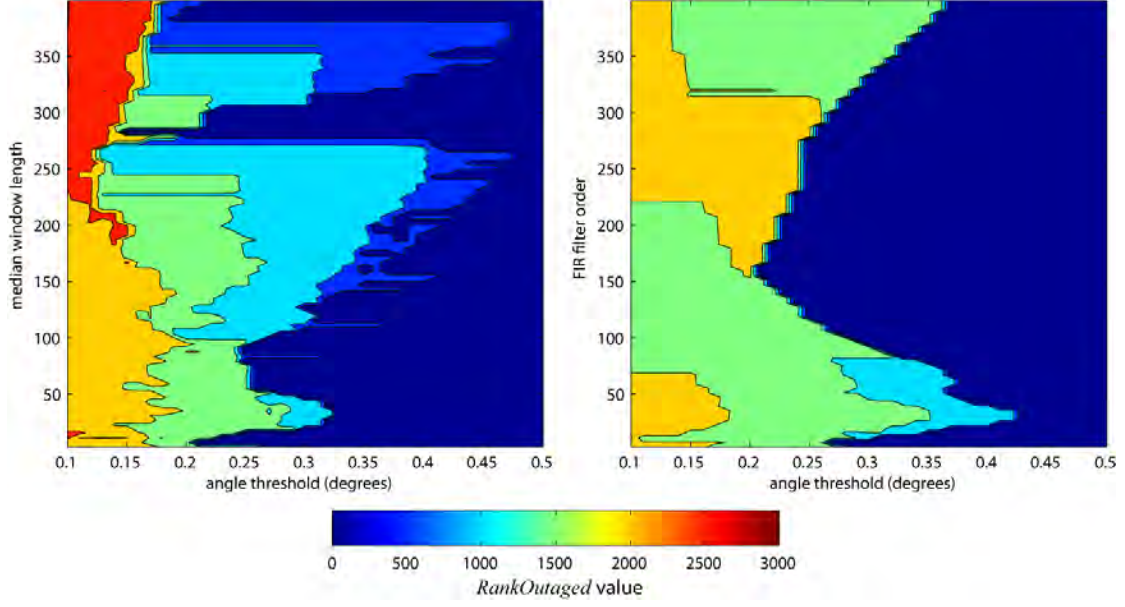


Figure 4.19: *RankOutaged* values for detection of the TVA outage event using filter lengths in the range of 3 to 401 and angle thresholds in the range of 0.1 to 0.5 degrees

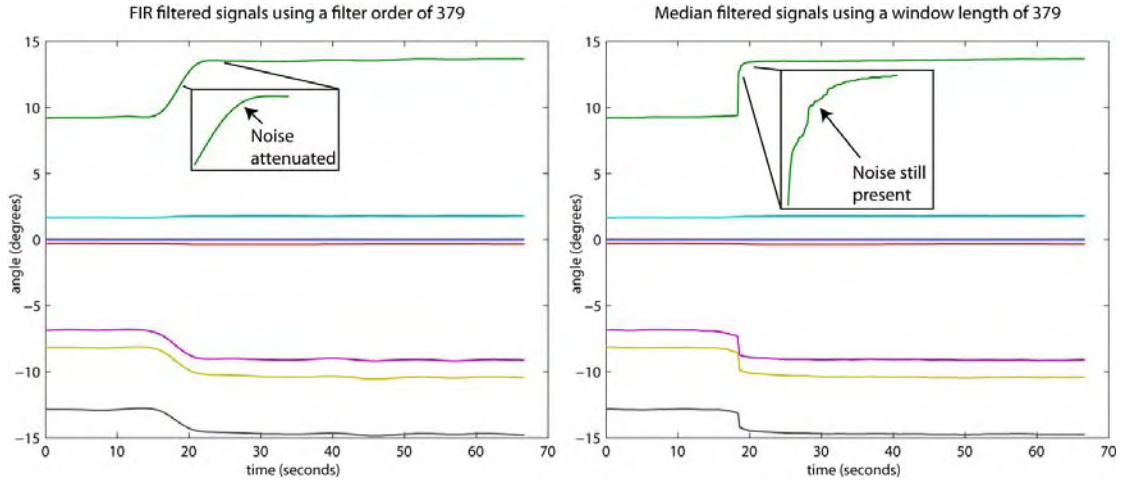


Figure 4.20: Differences in noise attenuation for FIR and median filtering for a filter length of 379

The presence of noise in the TVA PMU signals is the key difference between these values and those simulated for the 37 bus case. Because noise is present, the lower bound on τ must be raised to avoid any misclassification of noise as an event. To show how a poor choice of τ and filter length can cause false positives, the following metric is defined:

$$EarlyEdge(FilterLength, \tau) = \begin{cases} 1 & , n_{max} < n_{line\ switch} \\ 0 & , \text{otherwise} \end{cases} \quad (4.11)$$

Where $n_{line\ switch}$ is the sample number corresponding to the outage of the Cumberland-Davidson line. Figure 4.21 shows the contour plots of *EarlyEdge* for FIR and median filtering, calculated with filter lengths between 3 and 401, and angle thresholds between 0.1 and 0.5 degrees. The red areas indicate which combinations of filter length and angle threshold result in misclassification of noise as an event. For any filter length, increasing the threshold results in lower misclassification; increasing the filter length can occasionally result in additional false positives due to the random noise present in the signals. These results show that choosing a higher value of τ will tend to give better results, although the downside of using a large value of τ is the potential increase in false negatives.

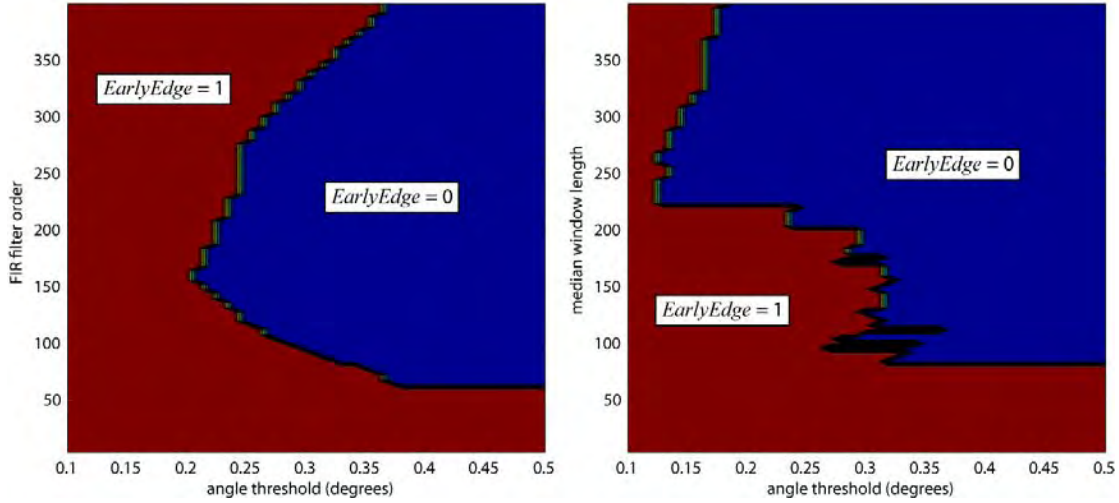


Figure 4.21: *EarlyEdge* contour plot for FIR (left) and median (right) filtering using filter lengths in the range of 3 to 401 and angle thresholds in the range of 0.1 to 0.5 degrees

Because the outaged line is connected to the PMU at 8CUMBERL, it is interesting to see how the algorithm performs when the angle measurement at 8CUMBERL is removed. Figure 4.22 shows that the algorithm performs just as well for certain combinations of filter length and angle threshold, but for other settings there is significant performance degradation. One cause of the poorer performance without the 8CUMBERL angle is that the oscillations are easier to attenuate for the other buses due to the lower amplitudes, and as a result it takes longer for the algorithm to settle on the n_{max} value. As n_{max} gets larger, more of the post-outage drift in the steady state angle is incorporated into $\Delta\theta_{observed}$ and results in poorer algorithm performance. The correct detection of the line outage for lower filter orders is promising, and indicates that the line outage can be detected with relatively low delay. These results also provide strong evidence that a basic median filter is likely to perform worse than a similarly sized FIR filter with noisy measurements due to poorer attenuation of noise.

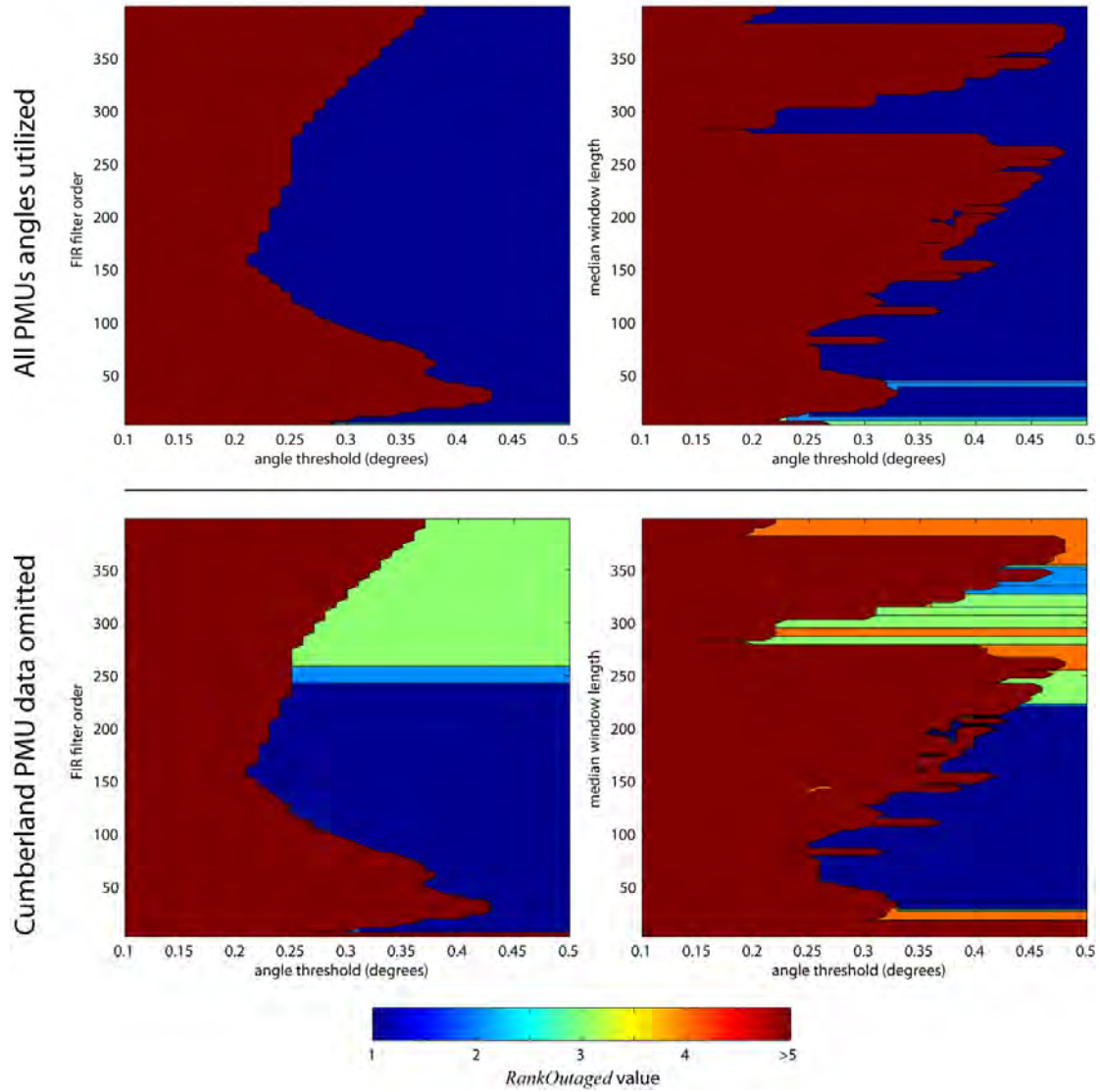


Figure 4.22: *RankOutaged* values with and without the usage of 8CUMBERL PMU angle measurements

4.2 Generator outage examples

The next set of examples looks at how well the algorithm can detect generator outages on the system. The 37 bus case used to test single line outage detection was also used to simulate PMU signals due to generator outages. Each of the nine generator outages was simulated using PSS/E, and the data was then processed in the manner discussed at the beginning of section 4.1.1.

4.2.1 All buses monitored

As with the single line outages, the first case examined is the case where there are PMUs located at all buses on the system. The two sets of contours in Figure 4.23 and Figure 4.24 show how well the algorithm ranks the outaged generator based on the angle measurements using FIR filtering with filter orders between 3 and 301 and angle thresholds between 0.1 and 3.0 degrees. The blank spaces in the contours show where the event was not detected (i.e., no angle exceeded the threshold).

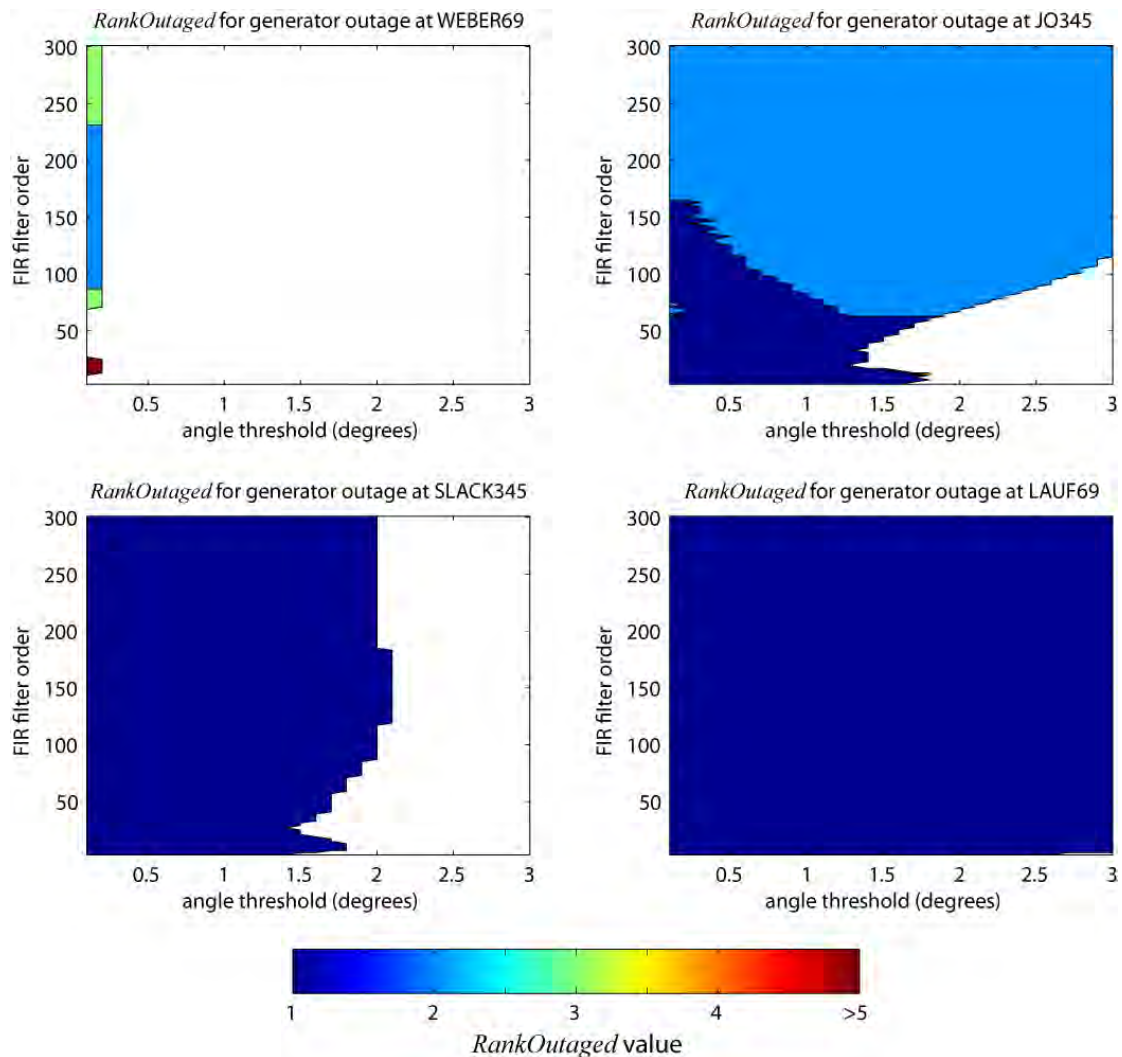


Figure 4.23: Contour plot of *RankOutaged* for generator outages at WEBER69, JO345, SLACK345, and LAUF69 using FIR filtering orders of 3 to 301 and angle thresholds of 0.1 to 3.0 degrees

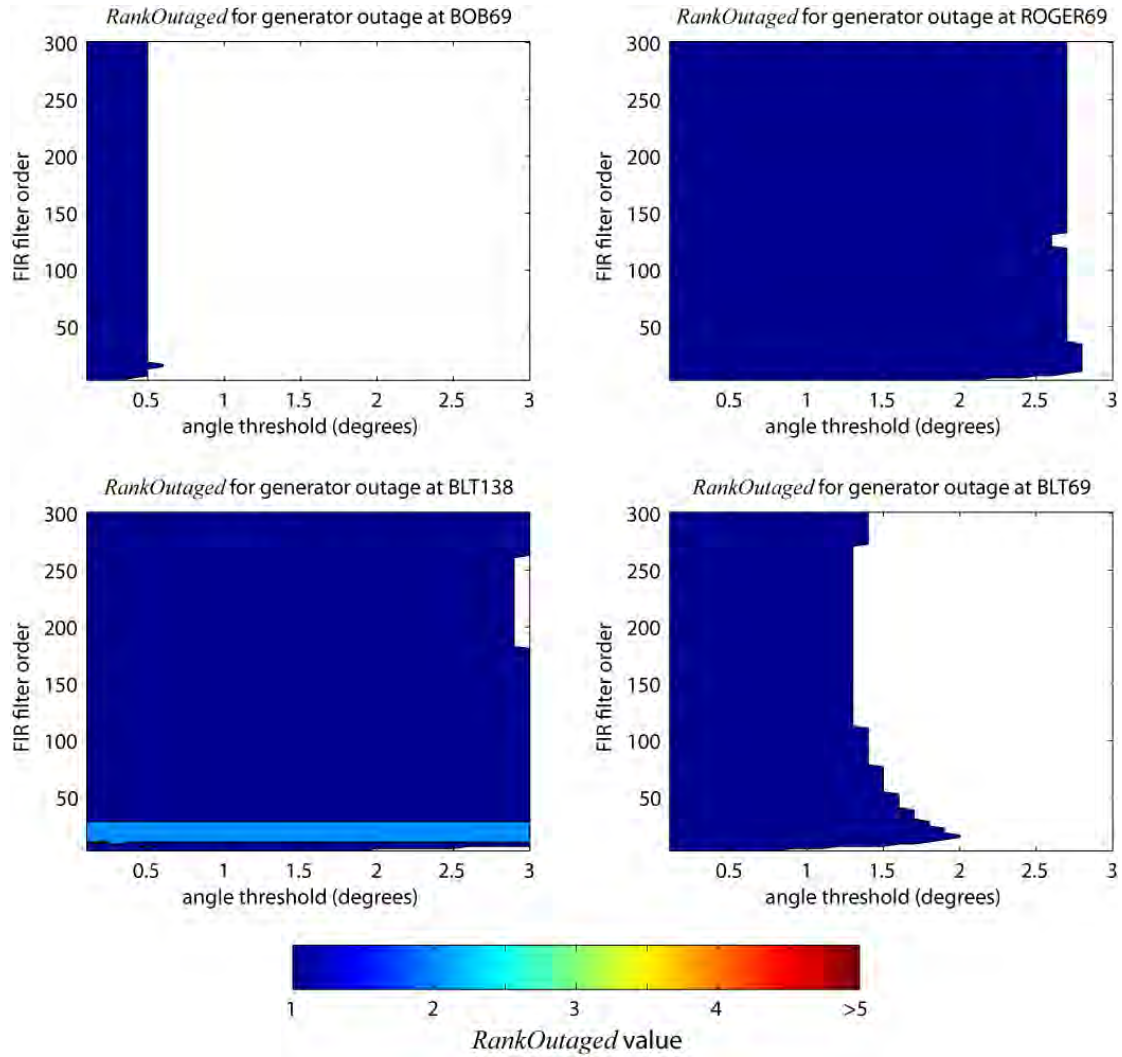


Figure 4.24: Contour plot of *RankOutaged* for generator outages at BOB69, ROGER69, BLT138, and BLT69 using FIR filtering orders of 3 to 301 and angle thresholds of 0.1 to 3.0 degrees

The only generator which is grossly misranked is WEBER69. As with the single line outages that resulted in small angle changes, the small deviation in angles due to the outage of this generator make it very difficult to match the correct outage to the observed angle changes. This is further evidence that selecting the correct value of τ is necessary to ensure the algorithm detects events properly.

The reason that the JO345 outages have *RankOutaged* equal to two for large portions of the contour is the inability of the algorithm to attenuate the oscillations with a filter cutoff frequency of 0.1 Hz. Figure 4.25 shows that an FIR filter of order 151 with a cutoff frequency of 0.1 Hz is incapable of properly damping the oscillations that are caused by this outage. Lowering the cutoff frequency to 0.01 Hz can improve results, as shown in Figure 4.26, but only for filter orders above 400.

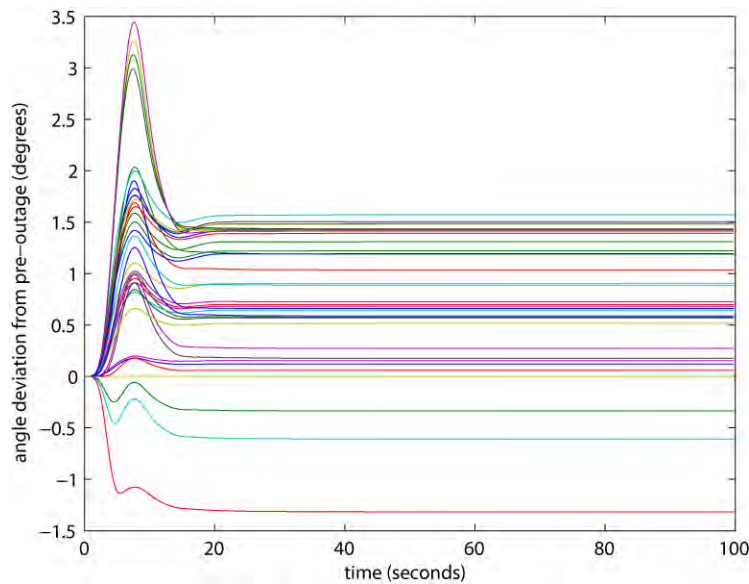


Figure 4.25: Prevailing oscillations after the outage of a JO345 generator using FIR filtering with filter order 151 and cutoff frequency of 0.1 Hz.

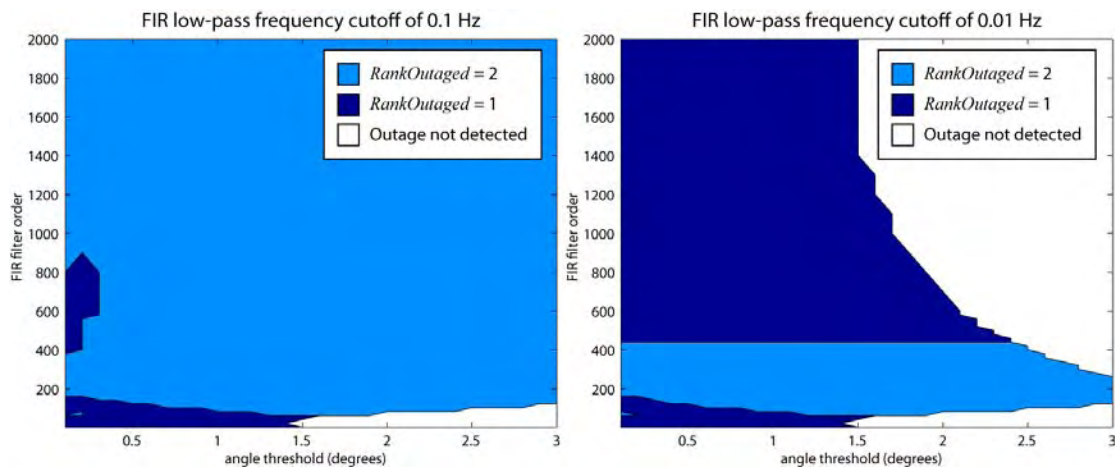


Figure 4.26: Effect of FIR low-pass cutoff frequency on the detection and rank of the JO345 generator outage

The same tests were run using median filtering instead of FIR filtering, and the resulting contour plots are shown in Figure 4.27 and Figure 4.28. The performance of the algorithm is similar for both filtering methods, although median filtering is able to correctly rank the JO345 outage slightly more often than FIR filtering. The other key difference is that the band in Figure 4.24 where *RankOutaged* is equal to two for the BLT138 outage is absent in the same contour of Figure 4.28.

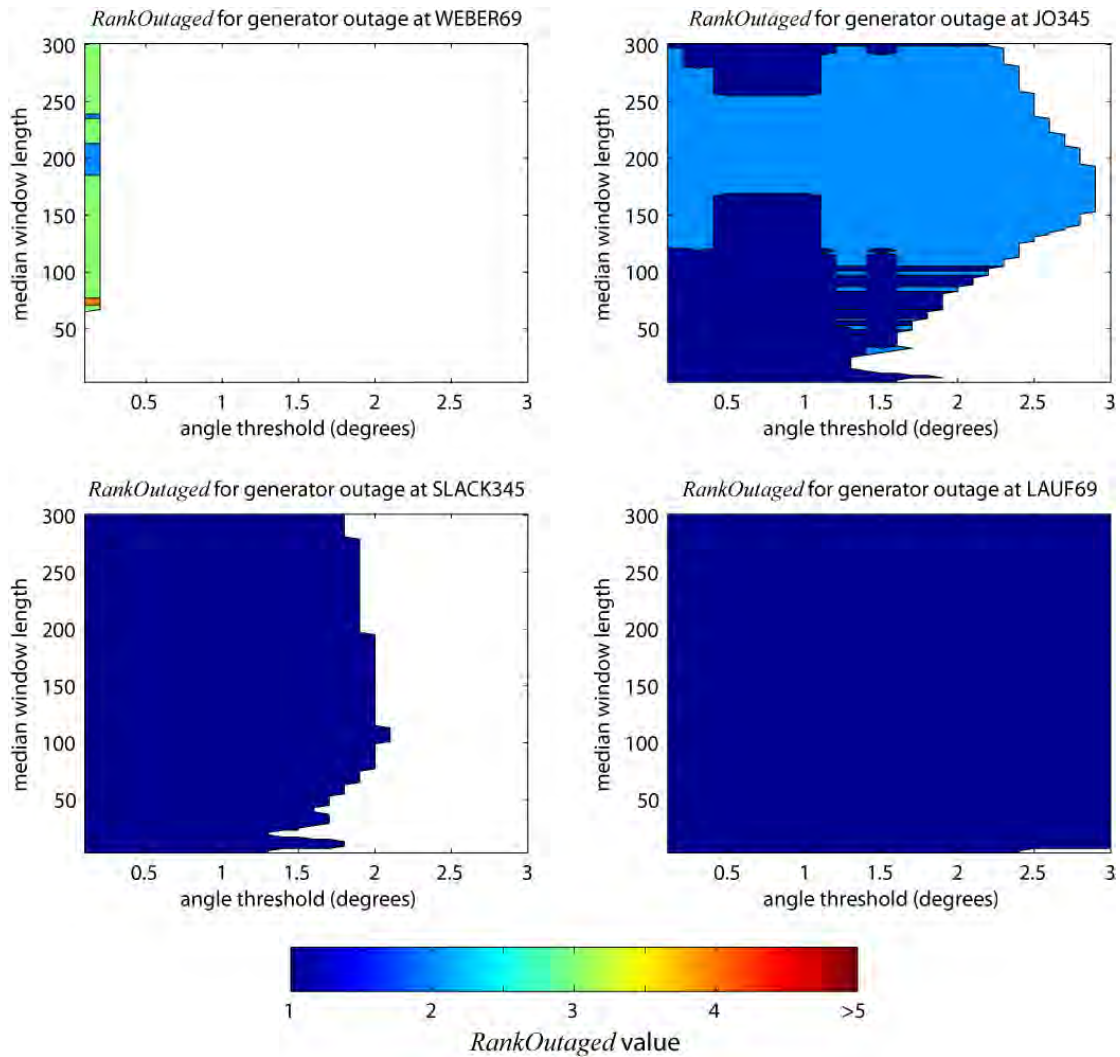


Figure 4.27: Contour plot of *RankOutaged* for generator outages at WEBER69, JO345, SLACK345, and LAUF69 using median filtering with window lengths of 3 to 301 and angle thresholds of 0.1 to 3.0 degrees

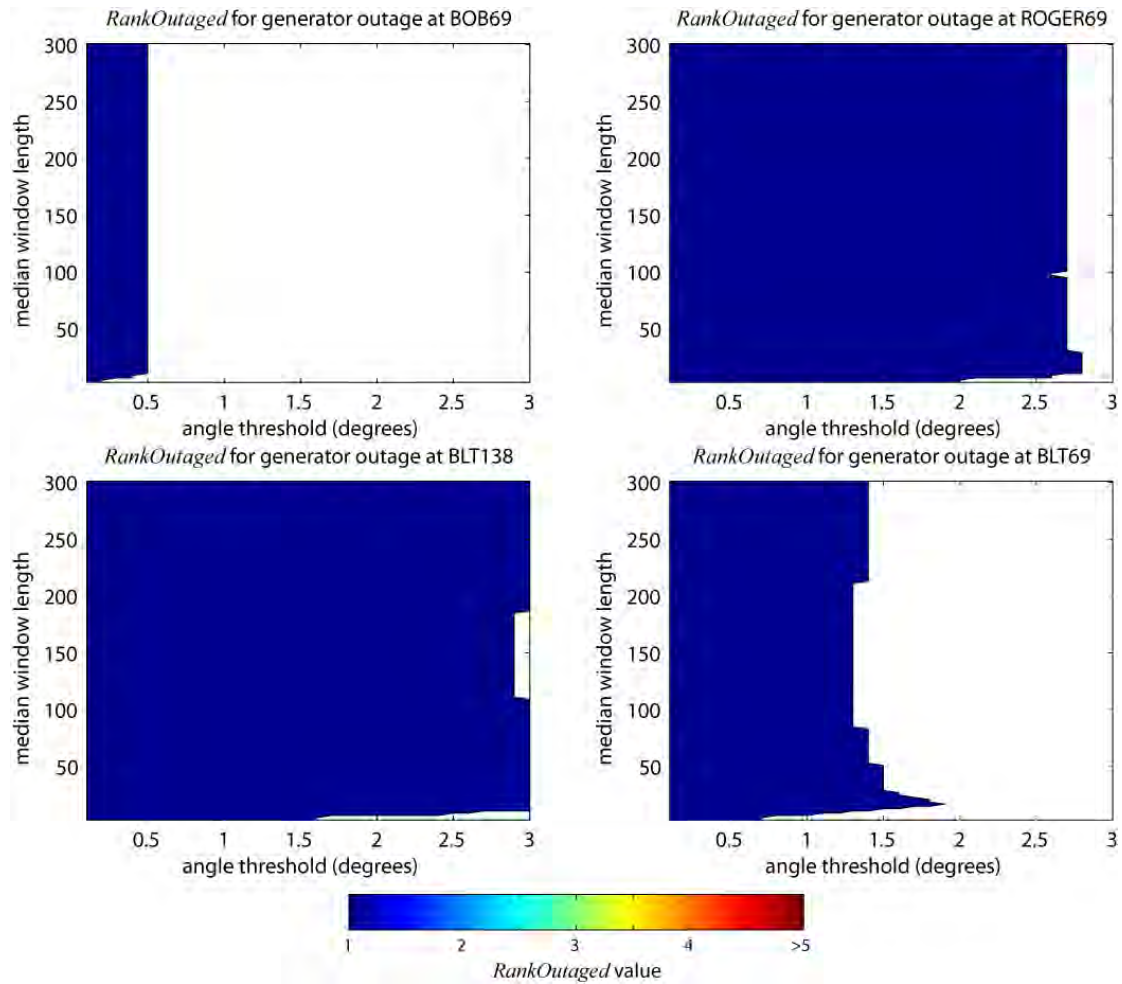


Figure 4.28: Contour plot of *RankOutaged* for generator outages at BOB69, ROGER69, BLT138, and BLT69 using median filtering with window lengths of 3 to 301 and angle thresholds of 0.1 to 3.0 degrees

Because the median filter has only one parameter, the window length, there is no need to tune any additional parameters such as the cutoff frequency. Figure 4.29 is presented to contrast the performance when median and FIR filtering are used. Comparing Figure 4.26 and Figure 4.29, median filtering results in a *RankOutaged* value of one significantly more often than FIR filtering with 0.1 or 0.01 Hz cutoff frequencies.

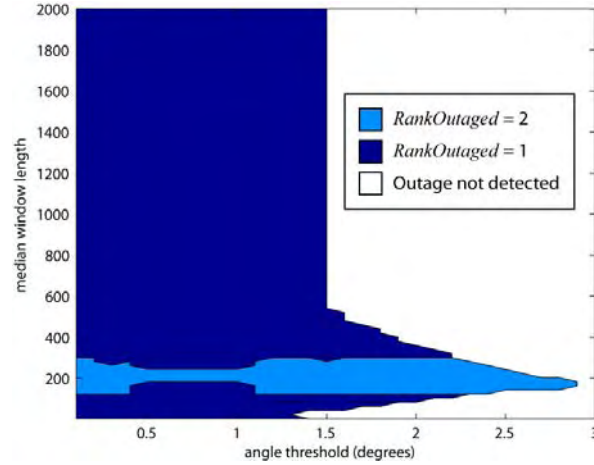


Figure 4.29: A detailed look at ranking and detection of the JO345 generator outage using median filtering

4.2.2 Eighteen buses monitored

The same placement of eighteen PMUs used in the single outage study was used to see how well the generator outage detection algorithm performs with sparser PMU deployment. Figure 4.30 and Figure 4.31 show the *RankOutaged* plots for the generator outages using this reduced PMU set with FIR filtering.

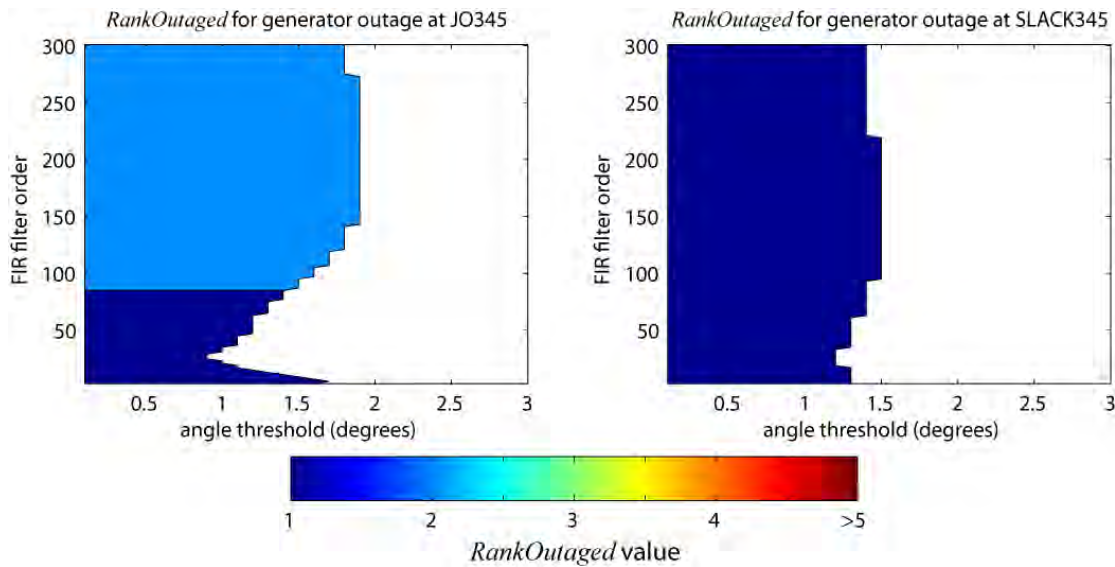


Figure 4.30: Contour plots analogous to those in Figure 4.23 for FIR filtering with eighteen buses monitored instead of full bus monitoring. The WEBER69 outage is undetected and the LAUF69 outage has *RankOutaged* equal to one over the entire region, so these contours are not shown.

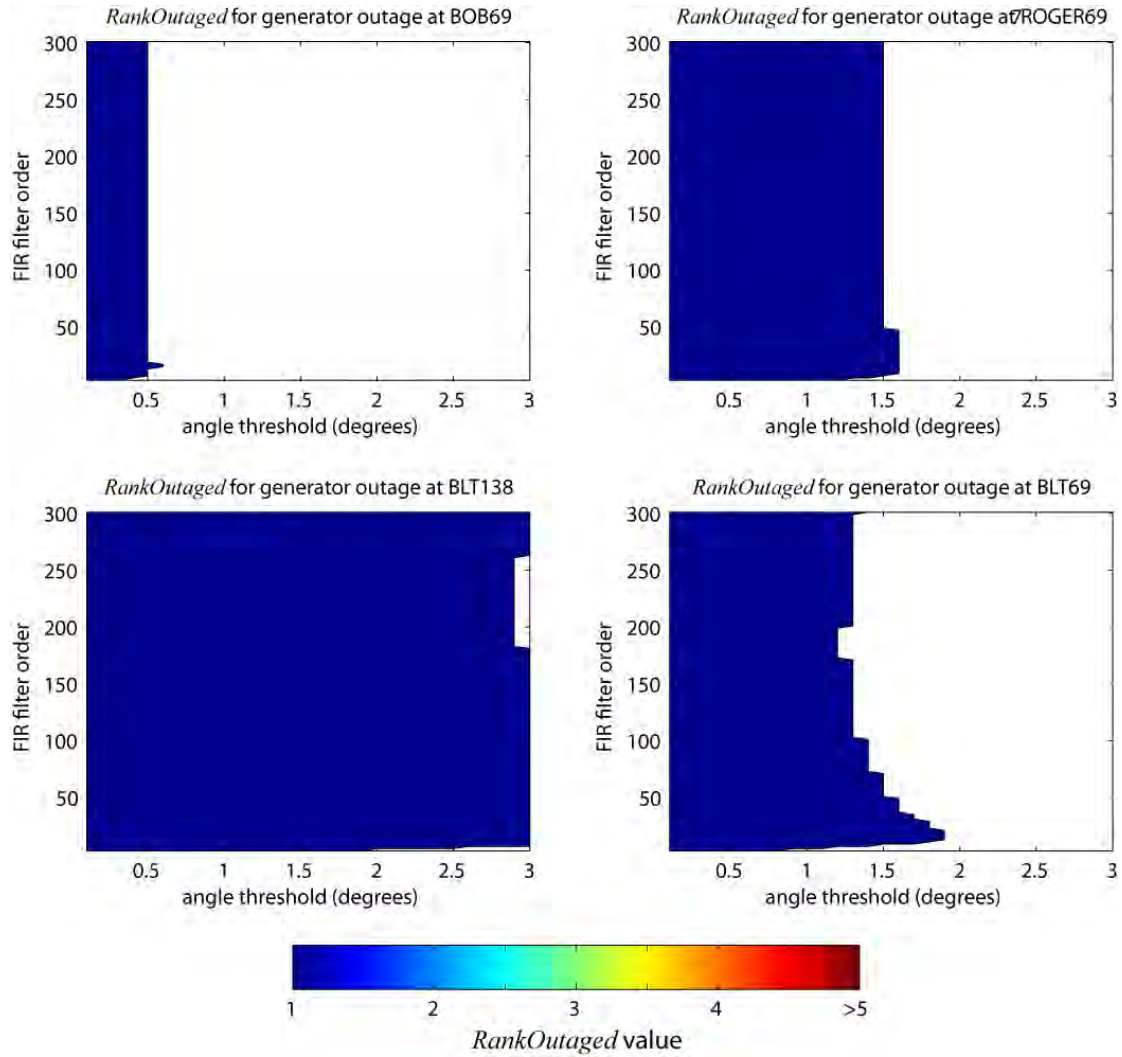


Figure 4.31: Contour plots analogous to those in Figure 4.24 for FIR filtering with eighteen buses monitored instead of full bus monitoring

As the number of monitored buses decreases, the set of angles which can signal the occurrence of an event is reduced. This is why there are larger empty regions in the contours with only 18 buses monitored—some angle measurements needed to detect the line with higher values of τ are no longer available. On real systems, this knowledge of how a change in the set of monitored buses impacts the number of false negatives can help in deciding where to place PMUs on the system. Finally, Figure 4.32 and Figure 4.33 show that median filtering results in a similar reduction in the ability to detect the outaged generators.

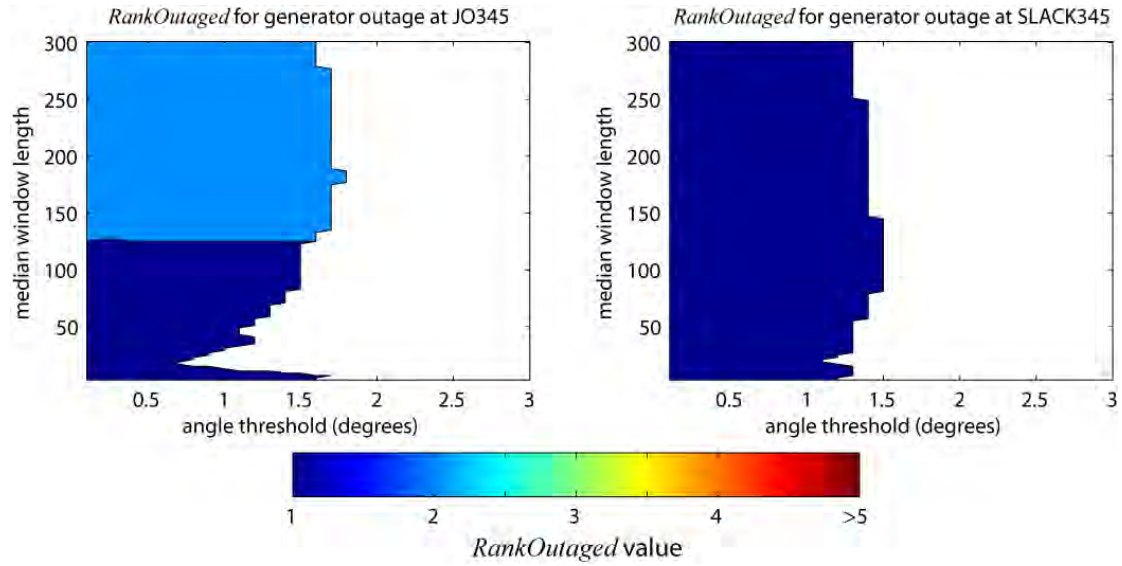


Figure 4.32: Contour plots analogous to those in Figure 4.27 for median filtering with eighteen buses monitored instead of full bus monitoring. The WEBER69 outage is undetected and the LAUF69 outage has *RankOutaged* equal to one over the entire region, so these contours are not shown.

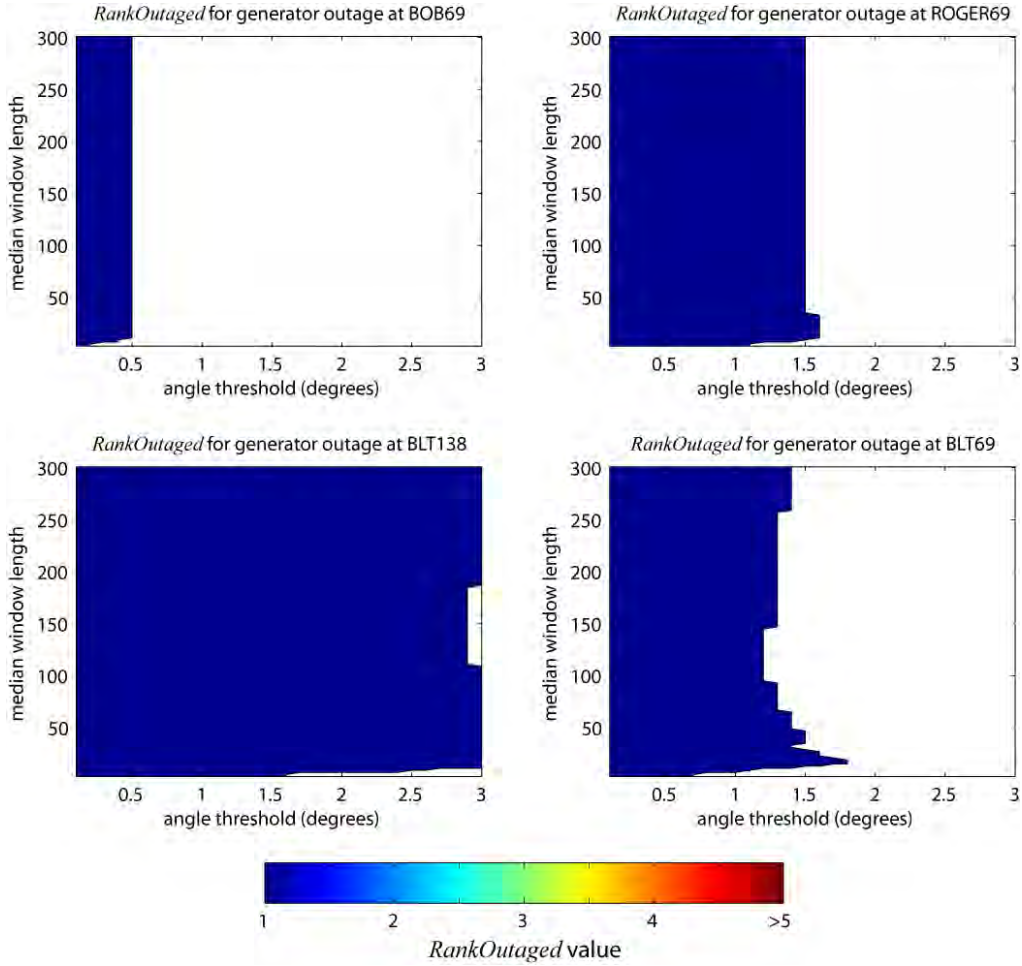


Figure 4.33: Contour plots analogous to those in Figure 4.28 for median filtering with eighteen buses monitored instead of full bus monitoring

4.3 Double line outage examples

4.3.1 All non-islanding double line outages

The same 37 bus system was used to test the performance of double line outage detection. A time domain simulation was run for each of the 1504 non-islanding double line outages to see how well the algorithm detects double line outages using the methods described in section 3.4.

4.3.1.1 All buses monitored

The first PMU configuration considered is with PMUs monitoring every system bus. The algorithm was run using filter lengths of 3, 9, 31, and 61 with FIR and median filtering using τ values of 0.02, 1.83, 2.5, and 5.16 degrees. These τ values were chosen so that, for signals

filtered with an FIR filter of order 31, 100%, 75%, 50%, and 25% (respectively) of the outages were detected by at least one angle exceeding the threshold.

Figure 4.34 shows in matrix form the *RankOutaged* value for each double line outage on the system, with each row/column representing one of the 56 in-service lines. The color of each square is used to indicate how the algorithm performed; red indicates a misranking (i.e., *RankOutaged* > 1) and green indicates correct ranking (i.e., *RankOutaged* = 1). The blank squares represent single line outages and double line outages that were not simulated due to islanding. Of the 1504 total double line outages considered, 1231 are ranked correctly and 273 are misranked; of these 273, 255 involve the two lines between buses 18 and 37 and the three lines between buses 15 and 54. These lines correspond to the rows with high red density at the top of Figure 4.34. Given the high R/X ratios of these lines, it is unsurprising that the algorithm, which relies on the dc power flow equations, is unable to properly rank these outages. Also, these were the same lines that caused problems for the single line outage detection (see Figure 4.2). The only way to solve this problem would be to use the full ac power flow equations rather than the dc power flow equations when attempting to match outages with observed measurements; however, this would lead to significantly higher computational costs and require much more state information than is currently available in real power systems over a suitably wide area.

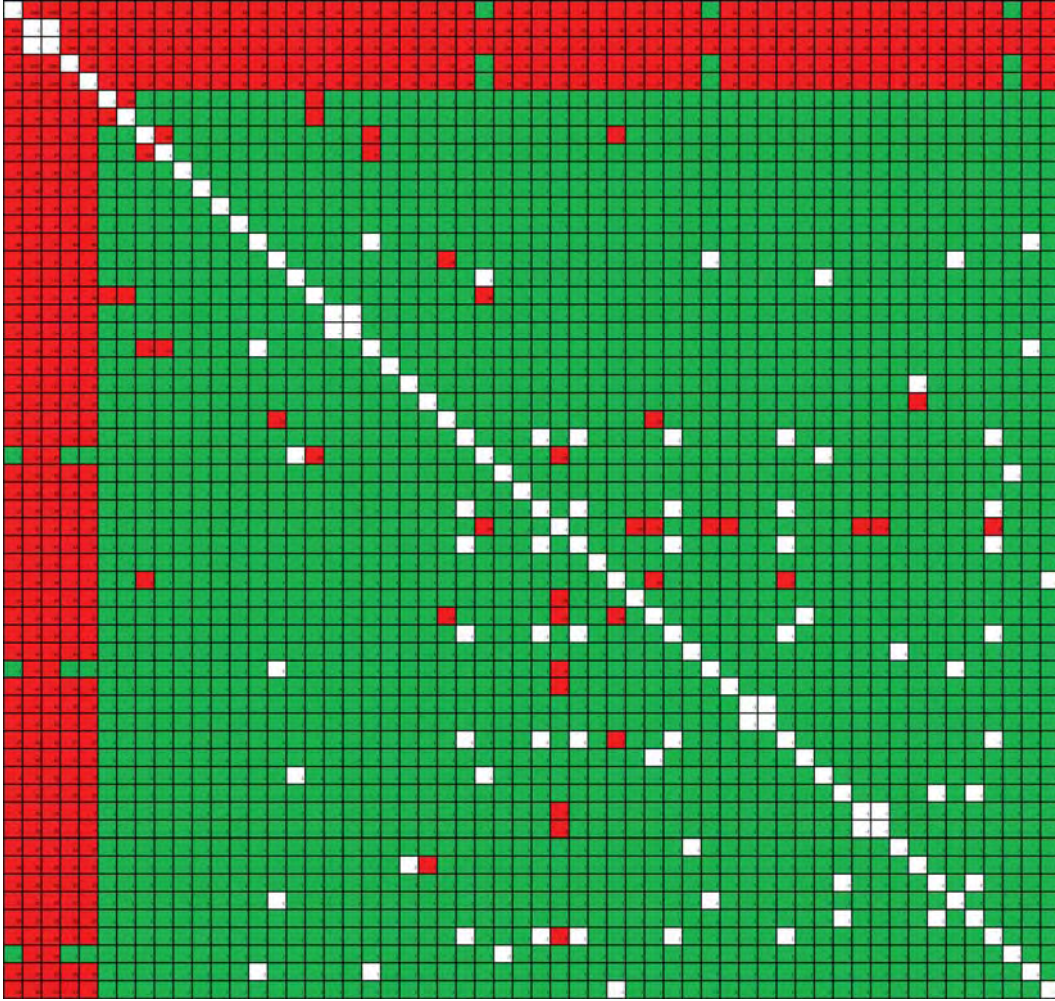


Figure 4.34: Ranking of each line-line combination using FIR filter order 31, angle threshold of 0.02 degrees. Green squares indicates double line outages where $RankOutaged = 1$, red squares indicate $RankOutaged > 1$ (i.e., misclassification), and white squares indicate outages that were not tested.

Fortunately, due to the small angle changes associated with the outages of the high R/X lines, many of the double outages involving lines 18-37 and 15-54 are undetected if τ is set higher. Figure 4.35 shows the $RankOutaged$ matrix for the four values of τ used to test the algorithm, where black squares are used to indicate events that were not detected. With the threshold set to 1.18 degrees, the number of misranked events involving the 18-37 and 15-54 lines drops from 255 to 100. The undesirable side effect is that several outages which were ranked properly in the $\tau = 0.02^\circ$ runs are no longer detected.

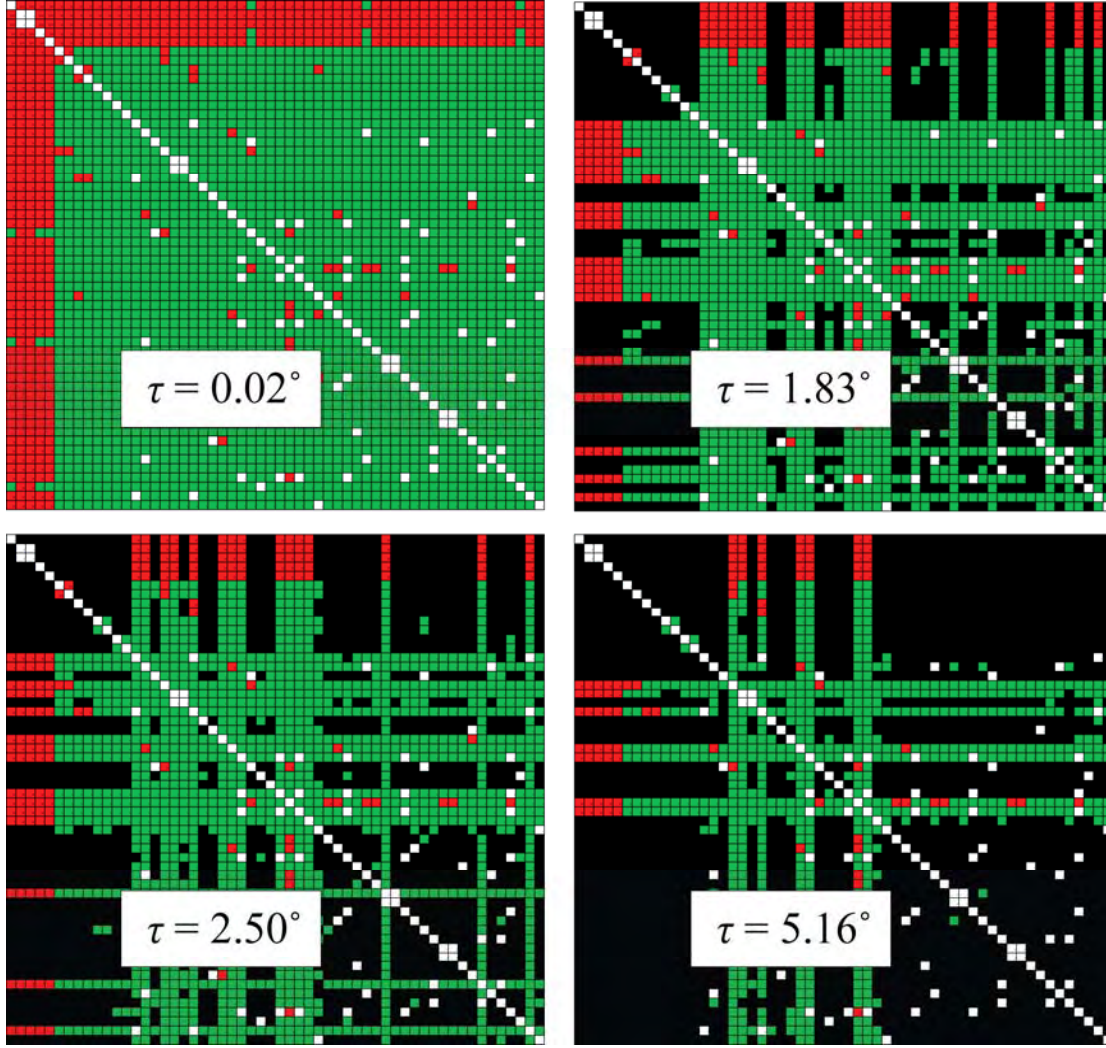


Figure 4.35: Effect of angle threshold on event detection with 31-order FIR filtering; black squares represent outages that were undetected due to insufficient angle changes.

Outside of the problems due to the 18-37 and 15-54 lines, there are still several instances where a double line outage is incorrectly identified. Using the $\tau = 0.02^\circ$ results, there are 21 additional misranked outages. Detailed information about these misranked outages is provided in Table 4.2, with the entries sorted by *RankOutaged*.

Table 4.2: Misrankings that do not involve 18-37 or 15-54 using FIR filtering with a filter order of 31 and angle threshold of 0.02 degrees

From bus (line 1)	To bus (line 1)	Circuit ID (line 1)	From bus (line 2)	To Bus (line 2)	Circuit ID (line 2)	<i>RankOutaged</i>	<i>NADOutaged</i>
39	38	1	39	38	2	105	0.0165
44	41	1	44	41	2	105	0.0137
24	44	1	47	53	1	44	0.0151
17	19	1	31	28	1	33	0.0549
31	38	1	17	19	1	30	0.0270
31	28	1	39	40	1	26	0.0205
10	13	1	10	39	1	23	0.0193
10	39	1	47	53	1	23	0.0212
31	28	1	47	53	1	21	0.0173
44	41	2	5	44	1	21	0.0129
44	41	1	5	44	1	19	0.0208
31	28	1	21	48	1	9	0.0062
31	28	1	21	48	2	9	0.0353
31	28	1	14	34	1	8	0.0162
31	28	1	13	55	1	6	0.0197
32	29	1	3	40	1	5	0.0571
31	28	1	35	31	1	4	0.0201
24	44	1	33	50	1	2	0.0209
39	38	1	31	38	1	2	0.0336
39	38	2	31	38	1	2	0.0200
44	41	1	24	44	1	2	0.0227

The two entries with the highest *RankOutaged* values in

Table 4.2 correspond to the outage of parallel lines between buses 39 and 38 and between buses 44 and 41. To better understand what happens in the case of the outage of the two lines between buses 39 and 38, Figure 4.36 shows the observed angle changes due to the outage along with the best match in angle changes due to this event (i.e., $\tilde{P}_{39-38,1}^* \Delta \tilde{\theta}_{calc,39-38,1} + \tilde{P}_{39-38,2}^* \Delta \tilde{\theta}_{39-38,2}$ as defined in (3.41)) and the best match in angle changes due to the simultaneous outage of line 39-38, circuit 1, and line 39-47, circuit 1 ($\tilde{P}_{39-38,1}^* \Delta \tilde{\theta}_{calc,39-38,1} + \tilde{P}_{39-47,1}^* \Delta \tilde{\theta}_{39-47,1}$).

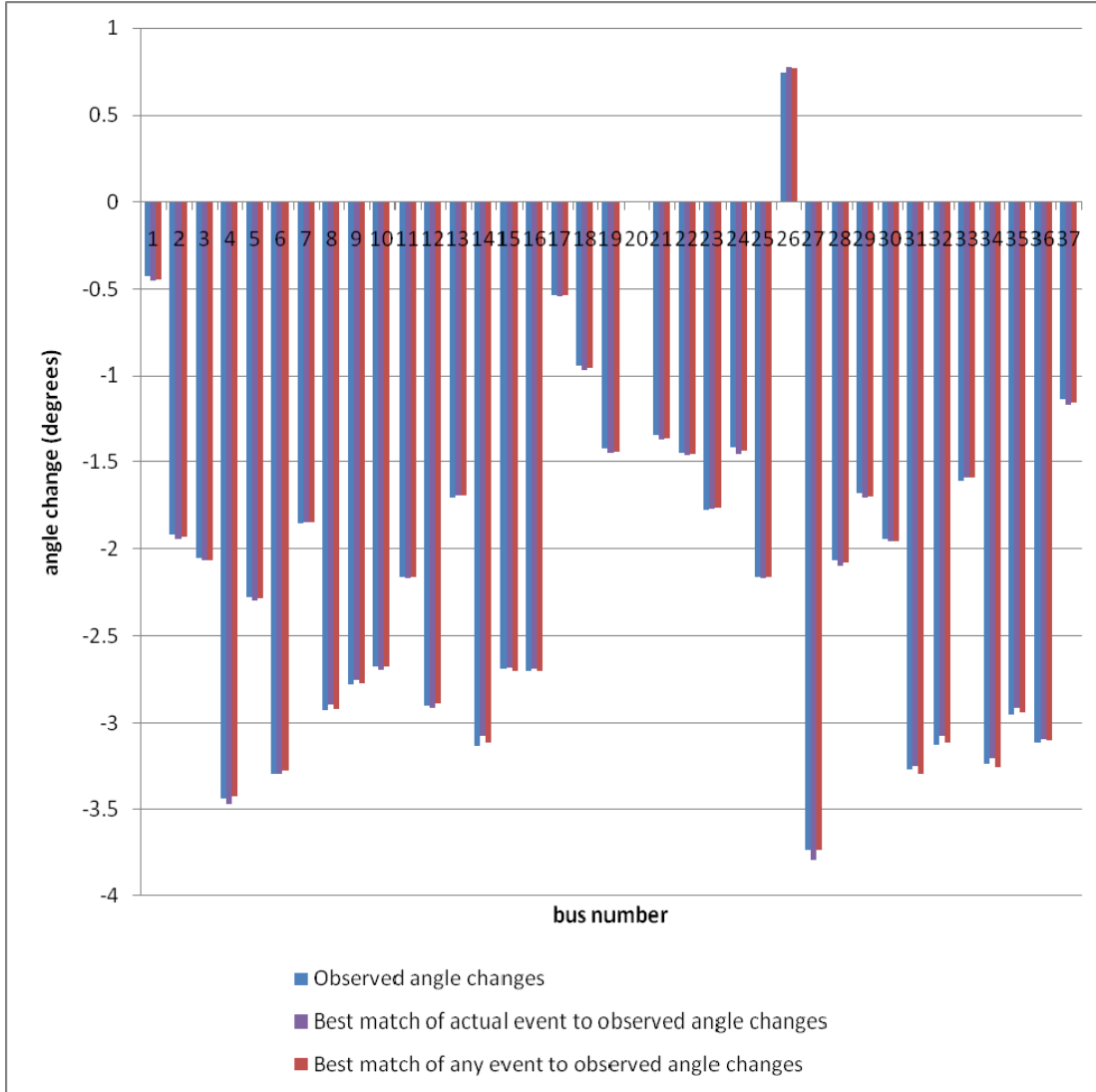


Figure 4.36: A detailed look at the observed and event matching angle changes due to the outage of the parallel lines between buses 39 and 38

The predicted angle changes due to both the true outaged event and the top-ranked event are very close, as shown in the figure. In addition, the top-ranked event includes one of the parallel

lines; in fact, in each of the 104 double line outages which have lower *NAD* values than the true line outage, either line 39-38, circuit 1 or line 39-38, circuit 2 is one of the two outages. Because only one parameter is used to fit parallel lines to observed angle changes (as discussed in section 3.4.1), the extra degree of freedom when other lines are considered as possibilities allows the algorithm to more closely match the observed angles. Restriction of the double outage event set to highly probable outages should help to mitigate this effect. In addition, if the top 10 results were presented to an operator, the fact that one of the 39-38 lines shows up in each entry would suggest that something has happened to one of those lines.

The highest *RankOutaged* value for a non-parallel set of lines is the outage of 24-44, circuit 1 and 47-53, circuit 1. In this case, all of the double line outages with *NAD* values less than the true event *NAD* value include line 24-44, circuit 1. *RankOutaged* does not decrease as the filter order is increased to 61, indicating that the inability of the algorithm to recognize line 47-53, circuit 1 as the second outaged line is due to the mismatch between the dc power flow solution and the true system response to the outages.

The double line outages do not exhibit any of the problematic high amplitude, low-frequency oscillations which make filtering such a crucial component of generator outage detection. Accordingly, the differences in results based on FIR and median filtering are minimal.

Table 4.3 contains a summary of the detection results for each combination of filter length and angle threshold mentioned at the beginning of this section. The statistics based on FIR filtering are outside of parentheses, and the statistics based on median filtering are inside parentheses. The similarity in the results for the two filtering methods, particularly as the filter length increases, is further indication that failure in the dc power flow assumptions is the primary cause of incorrect ranking of events.

In summary, double line outage detection is substantially more likely to generate misrankings due to the extra degree of freedom in choosing two, rather than one, pre-outage flows to match expected angles to observed angles. As a result, careful attention must be paid to how the double outage event set is defined, not only to reduce the computational effort involved in searching through the events, but also to reduce the chances of misranking the true outage event. Also, the choice of filtering method and filter length has only a slight impact on performance, as indicated in Table 4.3.

Table 4.3: Summary of algorithm results for FIR (median) filtered angle measurements with complete bus monitoring.

Events where $RankOutaged = 1$					
Filter length	Angle threshold (degrees)				
		0.02	1.83	2.5	5.16
	3	1180 (1179)	660 (604)	493 (413)	84 (61)
	9	1216 (1215)	805 (777)	645 (625)	304 (280)
	31	1228 (1231)	815 (814)	660 (651)	327 (318)
	61	1238 (1242)	826 (829)	663 (665)	330 (328)

Events where $RankOutaged > 1$					
Filter length	Angle threshold (degrees)				
		0.02	1.83	2.5	5.16
	3	318 (319)	136 (167)	103 (114)	19 (24)
	9	288 (289)	125 (118)	102 (93)	46 (39)
	31	276 (273)	120 (119)	92 (98)	50 (58)
	61	266 (262)	110 (113)	89 (89)	47 (48)

Undetected Events					
Filter length	Angle threshold (degrees)				
		0.02	1.83	2.5	5.16
	3	6 (6)	708 (733)	908 (977)	1401 (1419)
	9	0 (0)	574 (609)	757 (786)	1154 (1185)
	31	0 (0)	569 (571)	752 (755)	1127 (1128)
	61	0 (0)	568 (562)	752 (750)	1127 (1128)

4.3.1.2 Eighteen buses monitored

In addition to testing with PMUs at all buses, tests of the double line detection algorithm were run with the eighteen bus set defined in section 4.1.1.2. As with single line outage detection, there are some special cases of double line outages which deserve special consideration if some buses are unmonitored.

The first case considered is where both outaged lines are in the unmonitored system and are connected to the remaining system through two boundary buses. Let l_1 and l_2 denote the lines which are outaged. Then, because the vectors used to minimize the difference in observed and expected angles are the same in the single and double line outage detection (see (3.8) and (3.36)), the change in angles on the monitored system must be colinear (i.e., $\exists \alpha, \Delta \tilde{\theta}_{calc, l_1} = \alpha \Delta \tilde{\theta}_{calc, l_2}$). As a result, the minimization of the difference between the observed and expected angles, (3.37), collapses to a one-dimensional minimization,

$$\begin{aligned}
& \min_{\tilde{P}_{l,1}, \tilde{P}_{l,2}} \left\| \Delta \theta_{observed} - \left(\tilde{P}_{l,1} \Delta \tilde{\theta}_{calc,l_1} + \tilde{P}_{l,2} \Delta \tilde{\theta}_{calc,l_2} \right) \right\| = \\
& \min_{\tilde{P}_{l,1}, \tilde{P}_{l,2}} \left\| \Delta \theta_{observed} - \left(\tilde{P}_{l,1} \Delta \tilde{\theta}_{calc,l_1} + \tilde{P}_{l,2} \alpha \Delta \tilde{\theta}_{calc,l_1} \right) \right\| = \\
& \min_{\tilde{P}_{l,1}, \tilde{P}_{l,2}} \left\| \Delta \theta_{observed} - \Delta \tilde{\theta}_{calc,l_1} \left(\tilde{P}_{l,1} + \tilde{P}_{l,2} \alpha \right) \right\|
\end{aligned} \tag{4.12}$$

which has the following solution:

$$\begin{aligned}
& \left(\tilde{P}_{l,1} + \tilde{P}_{l,2} \alpha \right)^* = \frac{\Delta \tilde{\theta}_{calc,l_1} \cdot \Delta \theta_{observed}}{\Delta \tilde{\theta}_{calc,l_1} \cdot \Delta \tilde{\theta}_{calc,l_1}} \\
& \min_{\tilde{P}_{l,1}, \tilde{P}_{l,2}} \left\| \Delta \theta_{observed} - \Delta \tilde{\theta}_{calc,l_1} \left(\tilde{P}_{l,1} + \tilde{P}_{l,2} \alpha \right) \right\| = \\
& \left\| \Delta \theta_{observed} - \Delta \tilde{\theta}_{calc,l_1} \left(\tilde{P}_{l,1} + \tilde{P}_{l,2} \alpha \right)^* \right\|
\end{aligned} \tag{4.13}$$

An additional equation would allow for determination of $\tilde{P}_{l,1}$ and $\tilde{P}_{l,2}$, but the monitored system only “sees” the composite effect of the two outages and it is therefore not possible to determine $\tilde{P}_{l,1}$ and $\tilde{P}_{l,2}$ uniquely. On the other hand, the outage event can still be ranked by comparing the expected and observed angle changes.

Another interesting case is when one of the outaged lines connects an unmonitored portion of the system to a single bus within the monitored system. As discussed in section 4.1.1.2, the outage of a radial line connected to one boundary bus results in no angle changes on the monitored system (i.e., $\Delta \tilde{\theta}_{calc,radial\ line} = \mathbf{0}$). Referring to the boundary line as l_2 , the following formulation of the angle matching minimization is obtained:

$$\begin{aligned}
& \min_{\tilde{P}_{l,1}, \tilde{P}_{l,2}} \left\| \Delta \theta_{observed} - \left(\tilde{P}_{l,1} \Delta \tilde{\theta}_{calc,l_1} + \tilde{P}_{l,2} \Delta \tilde{\theta}_{calc,l_2} \right) \right\| = \\
& \min_{\tilde{P}_{l,1}, \tilde{P}_{l,2}} \left\| \Delta \theta_{observed} - \left(\tilde{P}_{l,1} \Delta \tilde{\theta}_{calc,l_1} + \tilde{P}_{l,2} \mathbf{0} \right) \right\| = \\
& \min_{\tilde{P}_{l,1}} \left\| \Delta \theta_{observed} - \tilde{P}_{l,1} \Delta \tilde{\theta}_{calc,l_1} \right\|
\end{aligned} \tag{4.14}$$

Because there are no angle changes associated with the boundary line l_2 , the minimization reduces to a single line outage detection problem for the other line. If both lines are connected through only one bus to the monitored system, then both $\Delta \tilde{\theta}_{calc}$ vectors are zero and there is no way to identify the outage with the algorithm.

Finally, there are instances in which two or more double line outages are indistinguishable from one another even though the individual $\Delta \tilde{\theta}_{calc}$ of each line are not collinear. Just as in the single line outage case indistinguishable lines resulted from there being only two buses

connecting the monitored and unmonitored system, indistinguishable double line outages occur when there are only three buses connecting the monitored and unmonitored system (see Figure 4.37).

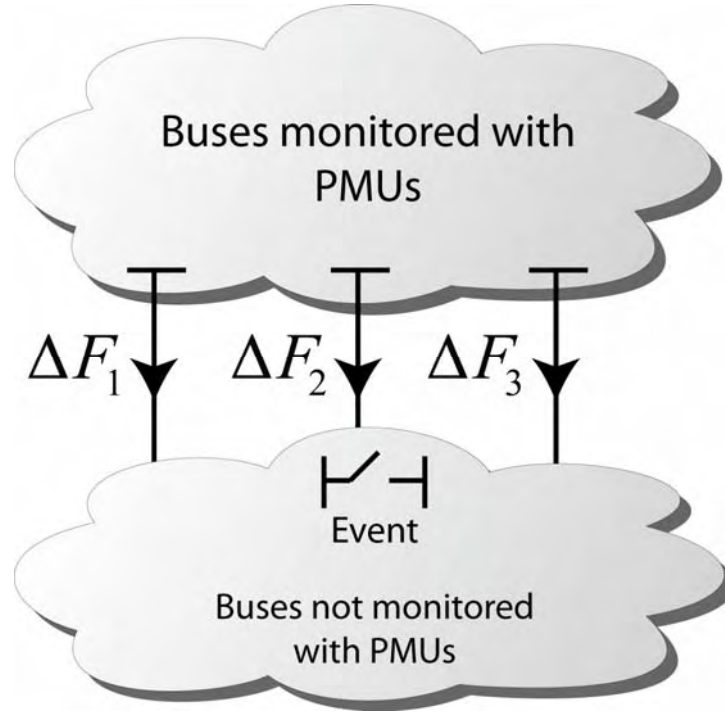


Figure 4.37: Unmonitored system connected to monitored system through three boundary buses

Two double line outages are indistinguishable if the following condition is true:

$$\forall a, b \in \square; \exists c, d \in \square; \quad (4.15)$$

$$a\Delta\theta_{calc, l_{1,EV1}} + b\Delta\theta_{calc, l_{2,EV1}} = c\Delta\theta_{calc, l_{1,EV2}} + d\Delta\theta_{calc, l_{2,EV2}}$$

In (4.15), one double line outage in the unmonitored system is referred to as $EV1$, and the other double line outage is referred to as $EV2$. The lines constituting the outage $EV1$ are $l_{1,EV1}$ and $l_{2,EV1}$; similarly, $l_{1,EV2}$ and $l_{2,EV2}$ refer to the lines in the double outage event $EV2$. If the relation in (4.15) is true, then when matching either event against the observed angle changes the same minimum difference will be obtained.

To determine the cases under which (4.15) holds, first let ΔF_1^{EV1} be the change in flow on boundary line 1 due to $EV1$, with similar notation to designate the other boundary flows ($\Delta F_2^{EV1}, \Delta F_3^{EV1}, \Delta F_1^{EV2}, \Delta F_2^{EV2}$ and ΔF_3^{EV2}) depicted in Figure 4.37. If no islanding results from the outages in the unmonitored system, then the bus power injections in the unmonitored system will stay the same and the boundary flows must sum to zero for each outage:

$$\Delta F_1 + \Delta F_2 + \Delta F_3 = 0 \quad (4.16)$$

As was done in 4.1.1.2, it can be shown that the $\Delta\tilde{\theta}_{calc,l}$ vector for the outage of any single line l in the unmonitored system can be determined by representing the changes in boundary flows as changes in injections at the boundary buses:

$$\Delta\tilde{\theta}_{calc,l} = \mathbf{KB}^{-1} \begin{pmatrix} \Delta F_1^l \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \leftarrow \text{boundary bus 1 row} + \\ \Delta F_2^l \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \leftarrow \text{boundary bus 2 row} + \\ \Delta F_3^l \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \leftarrow \text{boundary bus 3 row} \end{pmatrix} \quad (4.17)$$

Combining (4.16) and (4.17), $\Delta\tilde{\theta}_{calc,l}$ can be expressed in terms of ΔF_1^l and ΔF_2^l by eliminating ΔF_3^l :

$$\Delta F_3^l = -(\Delta F_1^l + \Delta F_2^l)$$

$$\Delta\tilde{\theta}_{calc,l} = \mathbf{KB}^{-1} \begin{pmatrix} \Delta F_1^l \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} \leftarrow \begin{matrix} \text{boundary bus 1 row} \\ \text{boundary bus 3 row} \end{matrix} + \\ \Delta F_2^l \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} \leftarrow \begin{matrix} \text{boundary bus 2 row} \\ \text{boundary bus 3 row} \end{matrix} \end{pmatrix} \quad (4.18)$$

Equation (4.18) can then be simplified further to give $\Delta\tilde{\theta}_{calc,l}$ as the sum of two scalar-vector products:

$$\begin{aligned}
\Delta \tilde{\boldsymbol{\theta}}_{calc,l} &= \Delta F_1^l \mathbf{KB}^{-1} \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} \begin{matrix} \leftarrow \text{boundary bus 1 row} \\ \leftarrow \text{boundary bus 3 row} \end{matrix} + \\
&\Delta F_2^l \mathbf{KB}^{-1} \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} \begin{matrix} \leftarrow \text{boundary bus 2 row} \\ \leftarrow \text{boundary bus 3 row} \end{matrix} \\
\Delta \tilde{\boldsymbol{\theta}}_{calc,l} &= \Delta F_1^l \Delta \boldsymbol{\theta}_1 + \Delta F_2^l \Delta \boldsymbol{\theta}_2
\end{aligned} \tag{4.19}$$

Combining (4.19) with (4.15), the condition under which two double line outages are indistinguishable can be restated in terms of the $\Delta \boldsymbol{\theta}_1$ and $\Delta \boldsymbol{\theta}_2$ vectors from (4.19):

$$\begin{aligned}
&\forall a, b \in \mathbb{R}, \exists c, d \in \mathbb{R}, \\
&a \left(\Delta F_1^{l_{1,EV1}} \Delta \boldsymbol{\theta}_1 + \Delta F_2^{l_{1,EV1}} \Delta \boldsymbol{\theta}_2 \right) + b \left(\Delta F_1^{l_{2,EV1}} \Delta \boldsymbol{\theta}_1 + \Delta F_2^{l_{2,EV1}} \Delta \boldsymbol{\theta}_2 \right) = \\
&c \left(\Delta F_1^{l_{1,EV2}} \Delta \boldsymbol{\theta}_1 + \Delta F_2^{l_{1,EV2}} \Delta \boldsymbol{\theta}_2 \right) + d \left(\Delta F_1^{l_{2,EV2}} \Delta \boldsymbol{\theta}_1 + \Delta F_2^{l_{2,EV2}} \Delta \boldsymbol{\theta}_2 \right) \tag{4.20} \\
&\left(a \Delta F_1^{l_{1,EV1}} + b \Delta F_1^{l_{2,EV1}} \right) \Delta \boldsymbol{\theta}_1 + \left(a \Delta F_2^{l_{1,EV1}} + b \Delta F_2^{l_{2,EV1}} \right) \Delta \boldsymbol{\theta}_2 = \\
&\left(c \Delta F_1^{l_{1,EV2}} + d \Delta F_1^{l_{2,EV2}} \right) \Delta \boldsymbol{\theta}_1 + \left(c \Delta F_2^{l_{1,EV2}} + d \Delta F_2^{l_{2,EV2}} \right) \Delta \boldsymbol{\theta}_2
\end{aligned}$$

One condition under which (4.20) is true is if the coefficients on the $\Delta \boldsymbol{\theta}_1$ and $\Delta \boldsymbol{\theta}_2$ vectors match on both sides of the equation:

$$\begin{aligned}
c \Delta F_1^{l_{1,EV2}} + d \Delta F_1^{l_{2,EV2}} &= a \Delta F_1^{l_{1,EV1}} + b \Delta F_1^{l_{2,EV1}} \\
c \Delta F_2^{l_{1,EV2}} + d \Delta F_2^{l_{2,EV2}} &= a \Delta F_2^{l_{1,EV1}} + b \Delta F_2^{l_{2,EV1}} \\
\begin{bmatrix} \Delta F_1^{l_{1,EV2}} & \Delta F_1^{l_{2,EV2}} \\ \Delta F_2^{l_{1,EV2}} & \Delta F_2^{l_{2,EV2}} \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} &= \begin{bmatrix} a \Delta F_1^{l_{1,EV1}} + b \Delta F_1^{l_{2,EV1}} \\ a \Delta F_2^{l_{1,EV1}} + b \Delta F_2^{l_{2,EV1}} \end{bmatrix} \tag{4.21} \\
\begin{bmatrix} c \\ d \end{bmatrix} &= \begin{bmatrix} \Delta F_1^{l_{1,EV2}} & \Delta F_1^{l_{2,EV2}} \\ \Delta F_2^{l_{1,EV2}} & \Delta F_2^{l_{2,EV2}} \end{bmatrix}^{-1} \begin{bmatrix} a \Delta F_1^{l_{1,EV1}} + b \Delta F_1^{l_{2,EV1}} \\ a \Delta F_2^{l_{1,EV1}} + b \Delta F_2^{l_{2,EV1}} \end{bmatrix}
\end{aligned}$$

The inverse matrix in (4.21) exists if and only if the determinant is non-zero, i.e.,

$$\begin{aligned}
&\Delta F_1^{l_{1,EV2}} \Delta F_2^{l_{2,EV2}} - \Delta F_1^{l_{2,EV2}} \Delta F_2^{l_{1,EV2}} \neq 0 \\
&\frac{\Delta F_1^{l_{1,EV2}}}{\Delta F_1^{l_{2,EV2}}} \neq \frac{\Delta F_2^{l_{1,EV2}}}{\Delta F_2^{l_{2,EV2}}} \tag{4.22}
\end{aligned}$$

The condition in (4.22) will be true if the line outages are distinguishable with respect to the single line outage detection algorithm (i.e., the angle change vectors associated with each line are not scalar multiples of one another). If (4.22) is false, then the double outages $EV1$ and $EV2$ may still be indistinguishable, but for other reasons (e.g., if there are three lines in parallel). If (4.22) is true, then (4.21) gives the unique values of c and d for each value of a and b that are needed to satisfy the criteria for indistinguishable double outages, (4.15). Therefore, except for those double outage events where condition (4.22) is violated, the double line outages in the unmonitored system connected by three buses to the monitored system will not be distinguishable.

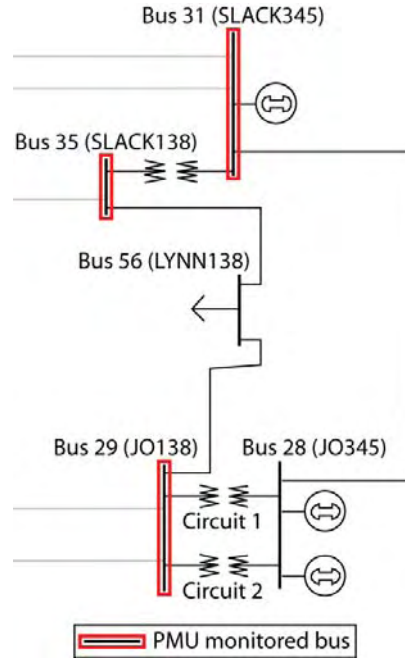


Figure 4.38: Portion of the 37 bus system that, when monitored with the 18 PMU set, has eleven indistinguishable double line outages

Figure 4.38 shows a portion of the 37 bus system that, when monitored at the 18 buses defined above, has eleven indistinguishable double line outages. The set of double line outages that are indistinguishable is provided in Table 4.4. There are six lines within the system unmonitored system, which are, using the notation (from bus number, to bus number, circuit ID), defined as: (35, 31, 1), (35, 56, 1), (56, 29, 1), (28, 29, 1), (28, 29, 2), and (31,28,1). The indistinguishable double outage set does not include four of the line combinations: $\{(29, 28, 1), (29, 28, 2)\}$, $\{(29, 28, 1), (31, 28, 1)\}$, $\{(29, 28, 2), (31, 28, 1)\}$, and $\{(35, 56, 1), (56,29,1)\}$. What distinguishes these double outage sets from the rest of the double outage sets is that these sets involve two lines which have collinear $\Delta\tilde{\theta}_{calc,l}$ vectors. As a result, the condition in (4.22) is not true and the condition for two double line outages to be indistinguishable is not satisfied.

Table 4.4: Indistinguishable double line outages for the portion of the 37 bus system illustrated in Figure 4.38

Line 1 From bus	Line 1 To bus	Line 1 Circuit ID	Line 2 From bus	Line 2 To bus	Line 2 Circuit ID
35	31	1	35	56	1
28	29	1	35	31	1
28	29	1	35	36	1
28	29	1	56	29	1
28	29	2	35	31	1
28	29	2	35	36	1
28	29	2	56	29	1
31	28	1	35	31	1
31	28	1	35	36	1
31	28	1	56	29	1
35	31	1	56	29	1

Due to the complicating factors associated with having unmonitored buses, the algorithm uses several steps to ensure each possible double outage is properly handled when attempting to match the event with the observed angle changes. Figure 4.39 shows the process that is undertaken for each event that is detected. The set of diamonds on the right hand side of the figure constitute a cascaded classification system to determine how the double line outage event should be modeled and fit to the observed angle changes.

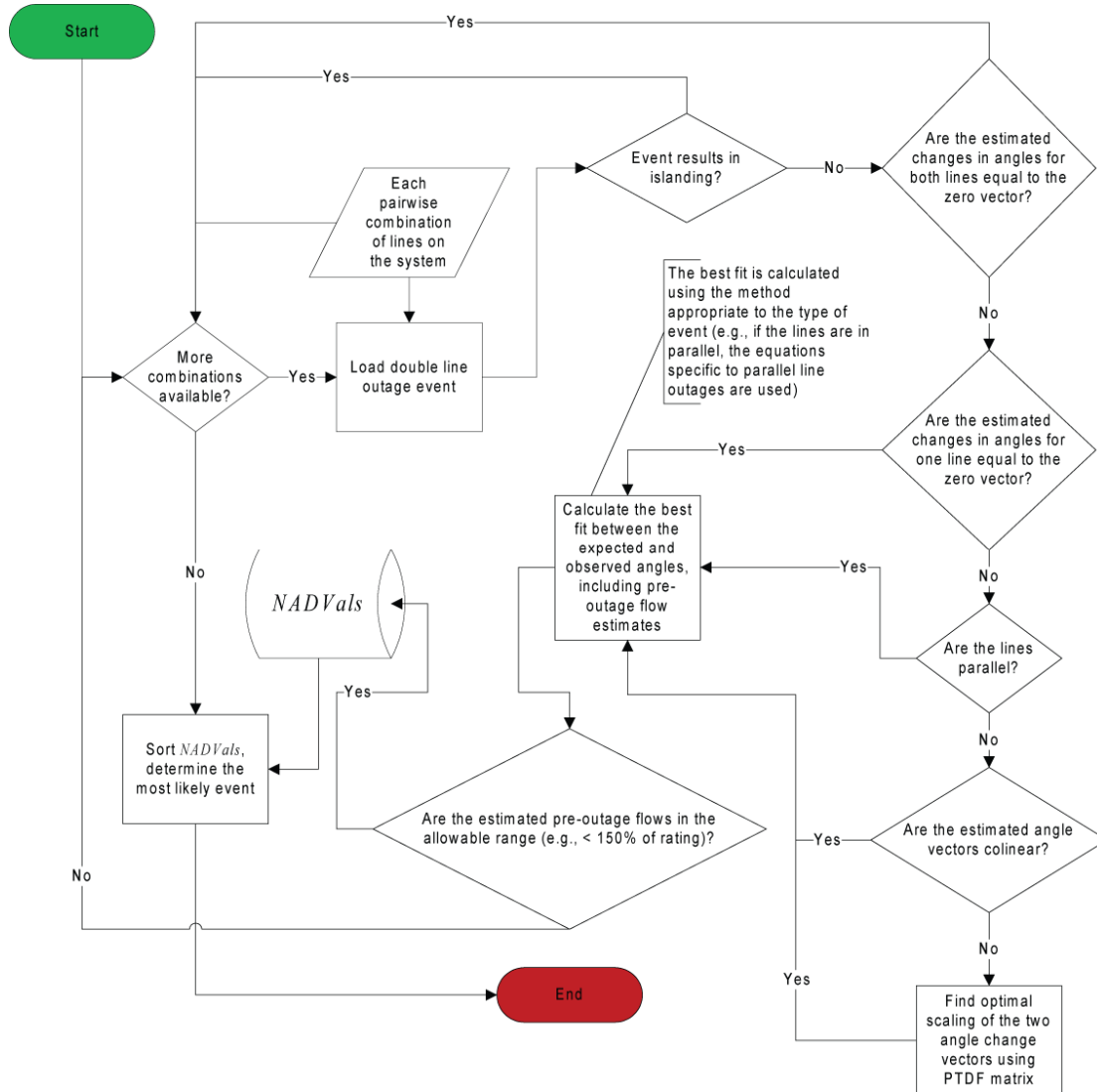


Figure 4.39: Program flow for the double line outage detection implementation, including cascading event classification

To test the performance of the algorithm with the lower number of PMUs, filter lengths of 3, 9, 31, and 61 were used along with angle thresholds of 0.006, 0.71, 1.04, and 2.31 degrees. These angle thresholds were chosen based on detecting 100%, 75%, 50%, and 25% of the events using FIR filtering of length 31.

The results based on FIR filtering are very similar to those results obtained with full PMU placement. Figure 4.40 provides the *RankOutaged* matrix for FIR filtering of order 31 using the reduced PMU set. Comparing this with Figure 4.35, the performance of the algorithm with the reduced PMU set is very similar to using the full PMU set. Also, the negative impact of the lines between buses 18 and 37 and the lines between buses 15 and 54 is clearly not mitigated by

reducing the number of PMUs. Figure 4.41 shows that the *RankOutaged* values also change very little when median filtering is used on the reduced PMU measurements.

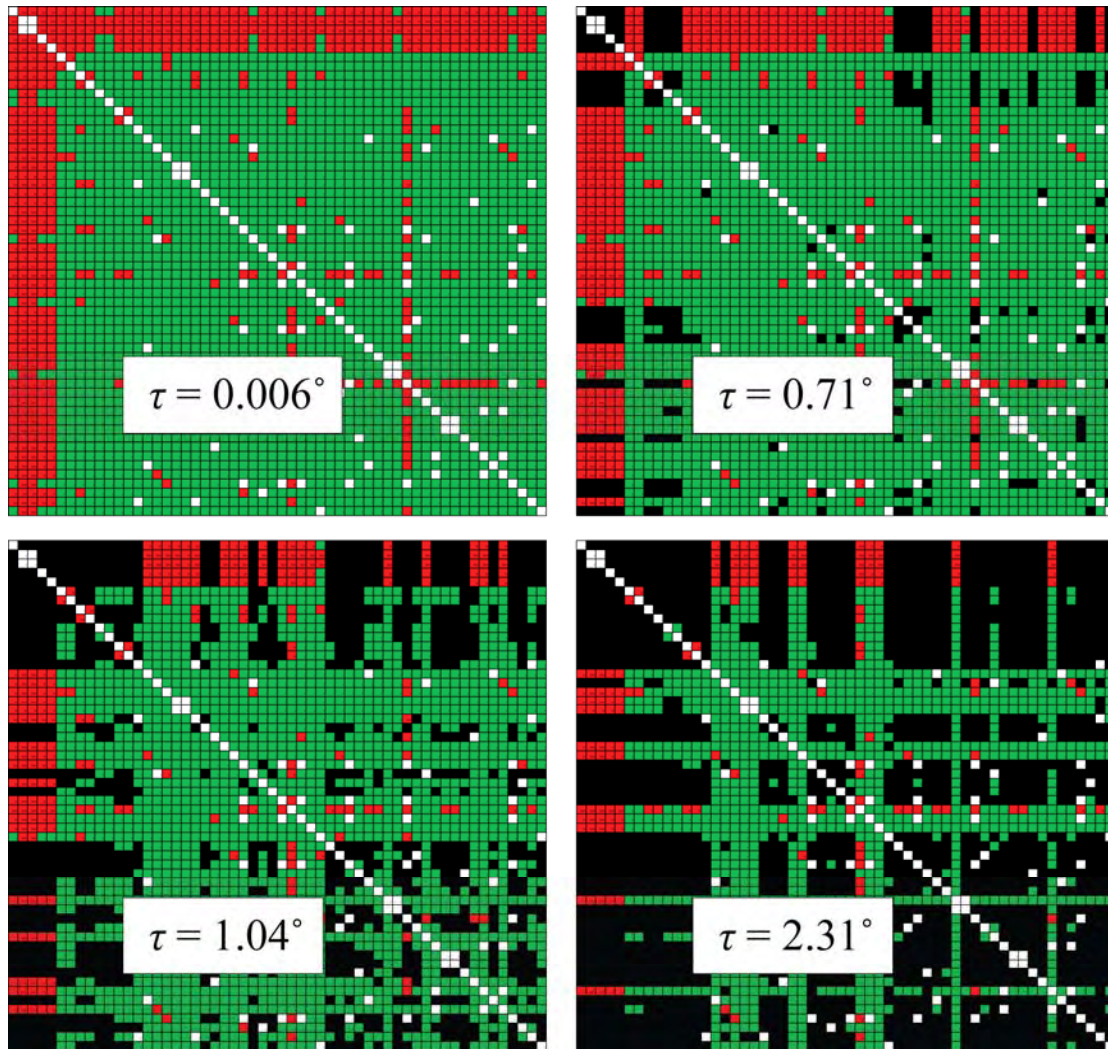


Figure 4.40: FIR filtered results, using a filter order of 31 and the set of eighteen PMUs defined in section 4.1.1.2

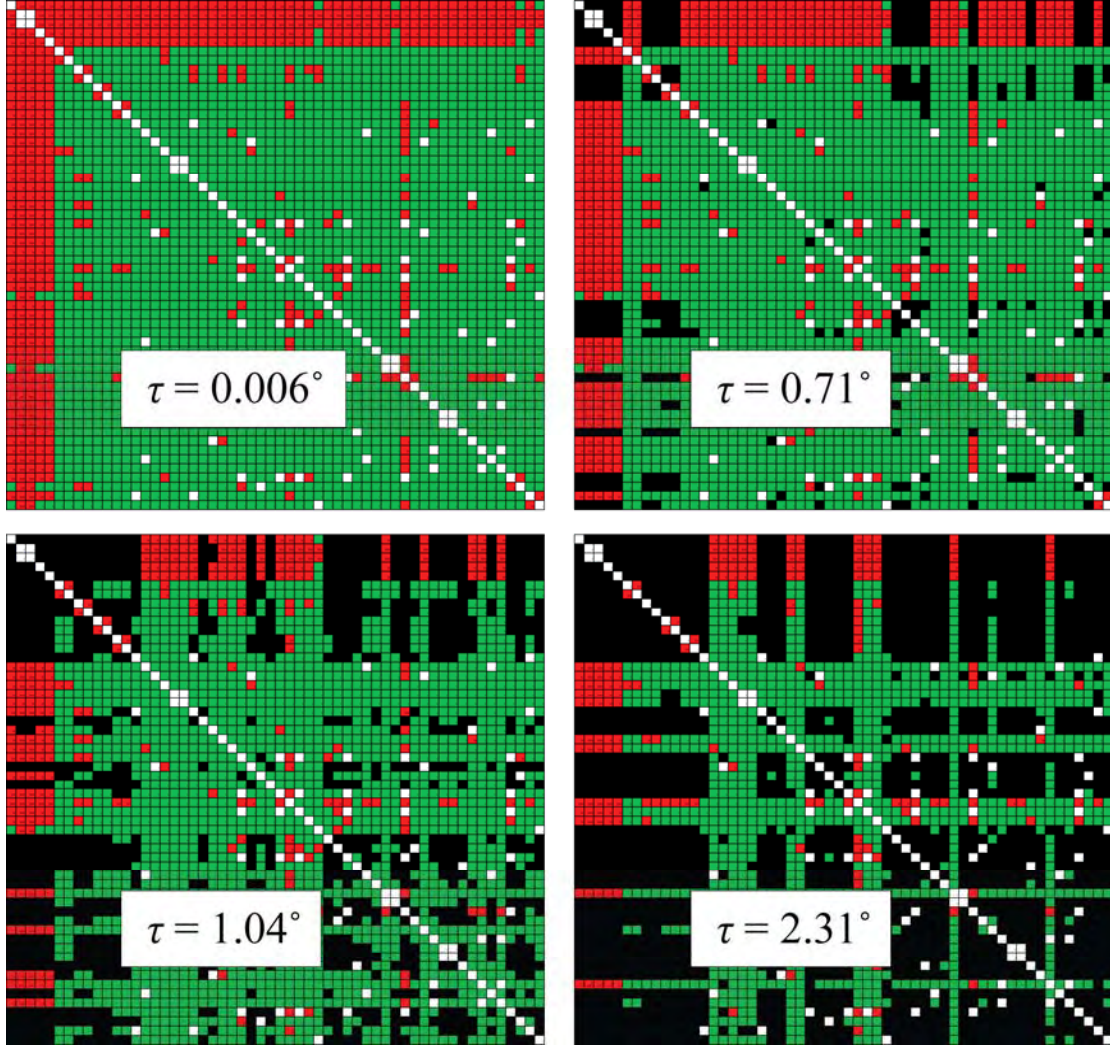


Figure 4.41: Median filtered results, using a window length of 31 and the set of eighteen PMUs defined in section 4.1.1.2

As with the single line outage tests, the most significant difference between having a full PMU deployment and 18 PMUs is the number of events which are indistinguishable. To quantify this effect, the *SharedOutaged* quantity defined in (4.10) is examined for the two different PMU deployments with 31-order FIR filtering and τ set to the minimum needed to detect all events using the full PMU deployment. Figure 4.42 shows that the ability of the algorithm to differentiate between the true outage event and other events on the system is significantly impaired by a decrease in PMU deployment. From an operational standpoint, higher *SharedOutaged* values require listing *SharedOutaged* + 1 events rather than just listing one event, but this can still be useful information for operators if the results are visualized.

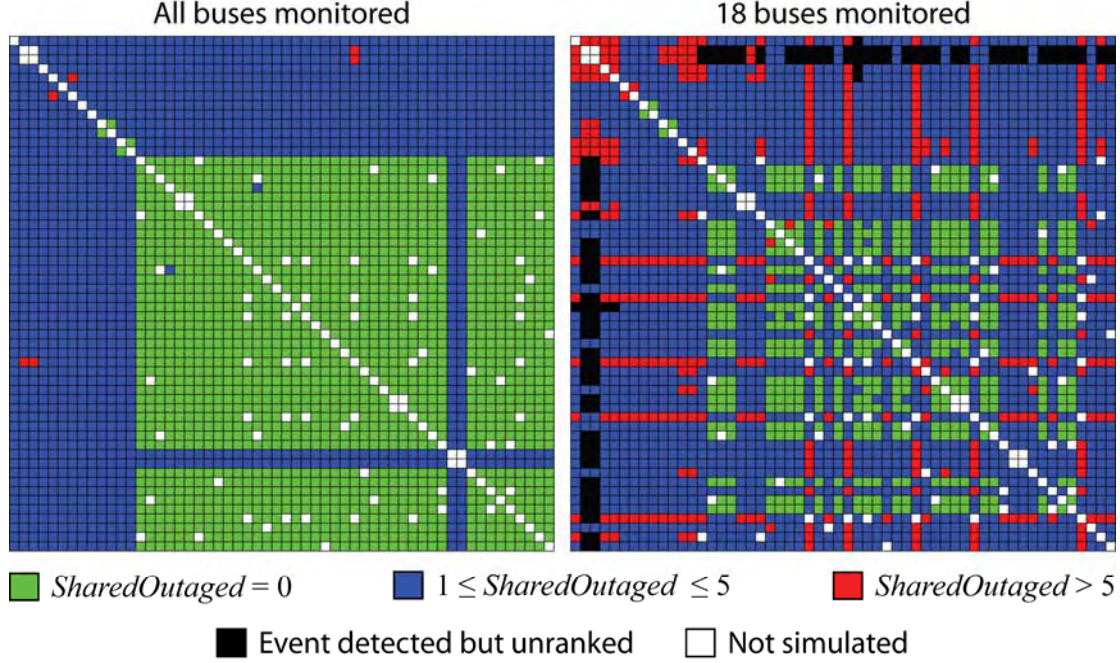


Figure 4.42: Effect of reducing the PMU set on *SharedOutaged* with 31-order FIR filtering and τ set to the minimum value needed to detect the maximum number of events

The black squares in Figure 4.42 designate double line outages which were detected but not ranked. This can only occur when $\Delta\tilde{\theta}_{calc}$ is equal to $\mathbf{0}$ for one or both of the lines outaged. If only one of the lines has $\Delta\tilde{\theta}_{calc} = \mathbf{0}$, then (4.14) is applied and, if the necessary pre-outage flow on the other line is above 1.5 times the line rating, then the event is removed from consideration (as discussed at the end of section 3.4.4) and not ranked. If both $\Delta\tilde{\theta}_{calc}$ vectors are $\mathbf{0}$, then it is not possible to perform the minimization in (3.37) and the event is not ranked.

4.3.2 All non-islanding double line outages that share at least one terminal

The results presented in section 4.3.1 are based on the simultaneous, unforced outage of each pairwise combination of lines on the system. In real power systems, two lines are unlikely to go out simultaneously unless they are close to one another, which is the basis for the filtering mechanism described in section 3.2.4. To ascertain the performance of the algorithm for more realistic double line outage events, additional tests were run using only those outages where at least one terminal bus is shared between the two lines. The set of events $\mathcal{E}$ tested to match with the observed angle changes was similarly restricted. This restriction reduces the number of possible double line outage events from 1504 to 115.

The algorithm was first tested with full PMU deployment using the same filter lengths and angle thresholds from section 4.3.1.1. A summary of the algorithm performance is provided in Table 4.5. Comparing these results with those from Table 4.3, there are significantly fewer misranked events. Figure 4.43 shows the *RankOutaged* values for the full double outage event

set (left side) and the set of double outages where a terminal bus is shared (right side) using the same row/column ordering and color scheme as the figures in section 4.3.1. The misranked double line outages which share a terminal bus are still concentrated in the first few rows—11 of the 16 misranked outages include lines between buses 18-37 and 15-54. As in the full event case, there are still problems with parallel line outages, although the *RankOutaged* value is reduced due to the smaller size of the event detection set $\mathcal{E}$.

Table 4.5: Summary of algorithm results for FIR (median) filtered angle measurements with complete bus monitoring considering only double line outages where a terminal bus is shared

Events where <i>RankOutaged</i> = 1					
Filter length	Angle threshold (degrees)				
	0.02	1.83	2.5	5.16	
	3	93 (93)	54 (53)	41 (34)	13 (8)
	9	98 (97)	59 (61)	51 (55)	24 (27)
	31	99 (101)	62 (65)	52 (57)	28 (27)
	61	99 (99)	62 (62)	52 (52)	28 (26)

Events where <i>RankOutaged</i> > 1					
Filter length	Angle threshold (degrees)				
	0.02	1.83	2.5	5.16	
	3	22 (22)	10 (11)	7 (6)	0 (1)
	9	17 (18)	10 (8)	10 (6)	7 (1)
	31	16 (14)	9 (6)	9 (4)	3 (4)
	61	16 (16)	9 (9)	9 (9)	3 (5)

Undetected Events					
Filter length	Angle threshold (degrees)				
	0.02	1.83	2.5	5.16	
	3	0 (0)	51 (51)	67 (75)	102 (106)
	9	0 (0)	46 (46)	54 (54)	84 (87)
	31	0 (0)	44 (44)	54 (54)	84 (84)
	61	0 (0)	44 (44)	54 (54)	84 (84)

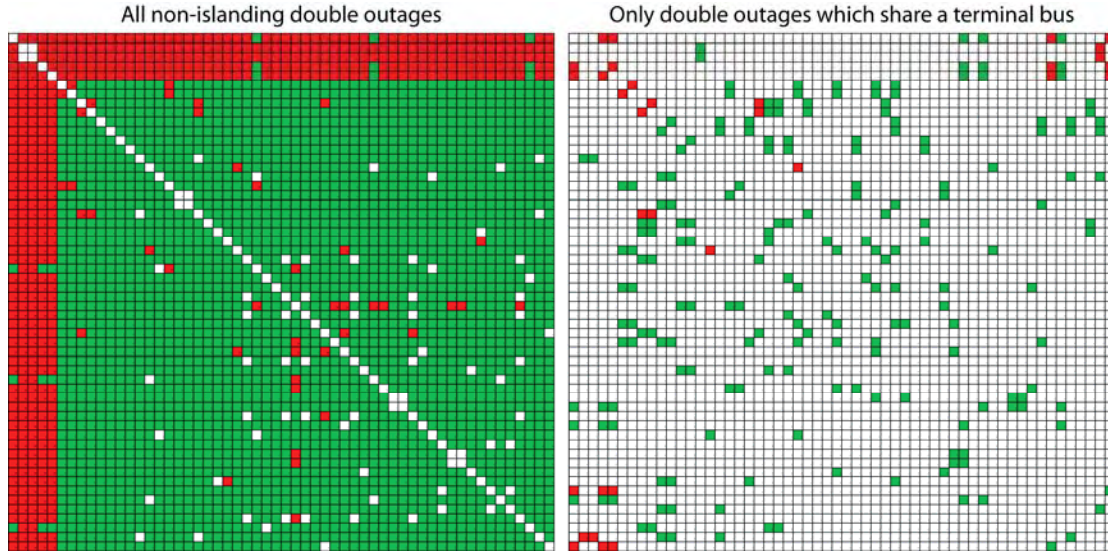


Figure 4.43: *RankOutaged* results for FIR filtering with angle threshold of 0.02 degrees, for the full double outage set (left side) and the double outages which share a terminal bus (right side). Red squares indicate where *RankOutaged* is greater than one and green squares indicate where *RankOutaged* equals 1. White squares indicate events which were not considered.

If the PMU coverage is reduced to 18 PMUs, the ranking of the events is not significantly impacted. In addition, reducing the event set to only those double line outages that share a terminal bus reduces the increase in *SharedOutaged*. Figure 4.44 shows the change in *SharedOutaged* for each of the double line outages as the PMU set is reduced to 18 PMUs. Although there is a small increase in *SharedOutaged* values as the number of PMUs is reduced, the maximum *SharedOutaged* value is 4. This is much lower than the maximum *SharedOutaged* value of 14 which is obtained if all events are considered and 18 PMUs are used (shown in Figure 4.42).

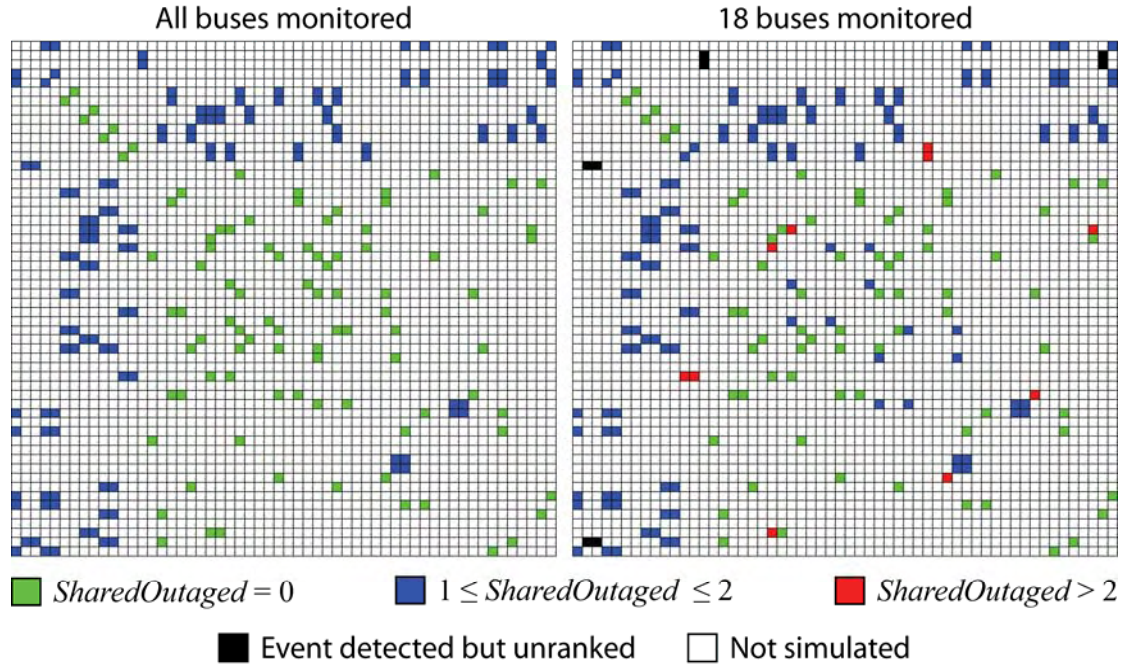


Figure 4.44: Effect of reducing the PMU set on *SharedOutaged* with 31-order FIR and τ set to the minimum value needed to detect all events, with the event set restricted to those lines that share a terminal bus

5 Phasor measurement unit placement for event detection

5.1 Overview

If the goal of deploying additional PMUs is to improve event detection, then there are some key characteristics of a given set which must be considered:

- Detection of events of interest
- Proper ranking of detected events
- Differentiation between events

Optimized placement of PMUs requires two components: an objective function to maximize (or minimize) and a method of searching the space of PMU sets so that the objective function is maximized (or minimized). The first section below deals with defining an objective function to meet certain criteria, while the second section provides details on simulated annealing (SA), the method used to search the set space. Finally, the third section presents some results showing how exhaustive search and SA perform in placing PMUs based on the objective functions.

5.2 Objective function definition

To evaluate each PMU placement set, an objective function must be defined which provides a numerical fitness value for each PMU set. For the purposes of PMU placement, it is assumed that filtering is applied to the raw measurements such that the observed angle change vector is equal to the difference in steady state angles (i.e., all oscillations around the new steady state angles are completely attenuated). To determine the observed angle change vector corresponding to each event, the post-event angle after 20 seconds was subtracted from the original angle, with the measurements filtered using a 61-order FIR filter. On the other hand, the angle threshold, which determines which events are detected by the algorithm, is allowed to vary so that the optimization function is maximized. The objective functions below include τ in their definition and the impact of τ on the objective function is discussed in each individual section.

5.2.1 Objective 1: Maximize the number of correctly ranked events

The simplest objective function is to evaluate the PMU set by maximizing the number of events that are detected and ranked correctly. An optimization problem which would meet this objective is:

$$\begin{aligned}
X_{Rank1}(e, \tau, PMUSet) &= \begin{cases} 1 & , RankOutaged(e, \tau, PMUSet) = 1 \\ 0 & , \text{event undetected} \end{cases} \\
ObjFunc_{Rank1}(PMUSet) &= \max_{\tau} \sum_{e \in \mathcal{E}} X_{Rank1}(e, \tau, PMUSet) \quad (5.1) \\
PMUSet^* &= \arg \max_{PMUSet} ObjFunc_{Rank1}(PMUSet)
\end{aligned}$$

where $\mathcal{E}$ is the set of events used to evaluate the PMU placement. Note that the set $PMUSet^*$ can have several members, corresponding to several PMU placements which result in the same maximum value of the objective function. In (5.1), the dependence of $RankOutaged$ on the angle threshold is made explicit in determining $RankOutaged$ and X_{Rank1} . Notice that if an event is undetected, it does not impact the objective function; as a result, this particular objective function does not penalize a particular PMU placement for resulting in misranked events. Therefore, the inner maximization over τ should always choose τ equal to the minimum angle change over all angles for each event, since choosing a different τ would reduce the number of possible events where $RankOutaged = 1$. The primary shortcomings of this objective function are the lack of a penalty for misranking of events and the lack of a penalty for having events which share the same rank.

5.2.2 Objective 2: Maximize the number of correctly ranked events with *no* misranked events

Building upon the first objective function, it is possible to define an objective function which penalizes a PMU placement if it results in misranked events. One such optimization definition is:

$$\begin{aligned}
X_{NoMisrank}(e, \tau, PMUSet) &= \begin{cases} 1 & , RankOutaged(e, \tau, PMUSet) = 1 \\ -\infty & , RankOutaged(e, \tau, PMUSet) > 1 \\ 0 & , e \text{ undetected given } \tau \text{ and } PMUSet \end{cases} \quad (5.2) \\
ObjFunc_{NoMisrank}(PMUSet) &= \max_{\tau} \sum_{e \in \mathcal{E}} X_{NoMisrank}(e, \tau, PMUSet) \\
PMUSet_{NoMisrank}^* &= \max_{PMUSet} ObjFunc_{NoMisrank}(PMUSet)
\end{aligned}$$

Once again, the dependence of $RankOutaged$ on the angle threshold is made explicit in determining $RankOutaged$ and $X_{NoMisrank}$. Unlike with the previous objective function, the optimal value of τ for the inner maximization is not necessarily the minimum angle change. Consider the case where there is an event (such as the outage of one of the 18-27 lines) that is misranked no matter how many angle measurements are taken due to failure of the dc power flow assumptions. In this case, the only way to improve the objective function would be to stop

detecting this event, which will be the case if τ is set sufficiently high. Therefore, the inner maximization must be explicitly calculated for this optimization problem. Fortunately, this does not pose much of a computation burden due to the finite number of τ values which impact the results; in the worst case, the number of evaluations needed to exhaustively solve the inner maximization is equal to the number of events.

This objective function is a significant improvement over the first, in that it explicitly accounts for the problem of misranking by removing from consideration any combination of PMU placement and τ that would result in a misranked event. The number of events which share the same rank is still not accounted for, but this optimization provides a much better match with the objectives mentioned at the beginning of the chapter.

5.2.3 Objective 3: Maximize the number of correctly ranked events with *no* misranked events and *no* events with the same rank

Taking the optimization problem from the previous section one step further, it is also possible to remove any PMU set and threshold combination if it results in events which share the same rank. The optimization to be solved in this case is:

$$\begin{aligned}
 X_{NoShared}(e, \tau, PMUSet) = & \begin{cases} 1 & , RankOutaged(e, \tau, PMUSet) = 1 \text{ and} \\ & SharedOutaged(e, \tau, PMUSet) = 0 \\ -\infty & , RankOutaged(e, \tau, PMUSet) > 1 \text{ or} \\ & SharedOutaged(e, \tau, PMUSet) > 0 \\ 0 & , e \text{ undetected given } \tau \text{ and} \\ & PMUSet \end{cases} \quad (5.3) \\
 ObjFunc_{NoShared}(PMUSet) = & \max_{\tau} \sum_{e \in \mathcal{E}} X_{NoShared}(e, \tau, PMUSet) \\
 PMUSet_{NoShared}^* = & \arg \max_{PMUSet} ObjFunc_{NoShared}(PMUSet)
 \end{aligned}$$

This is a much more restrictive objective function, in that only events which are ranked one and have no other events sharing that rank count towards the objective function value for a particular PMU placement. While this last objective function meets each of the objectives stated in 5.1, not all instances of *SharedOutaged* greater than zero are problematic if visualization of the event detection is used to present the results (e.g., in the case of parallel lines). As a result, this objective function may result in PMU placements which are poorer than other placements from a practical sense.

5.3 Searching methods

5.3.1 Exhaustive search

One method of determining the optimal placement of PMUs with respect to these objective functions is to perform an exhaustive search. This can become computationally intractable, depending on how *PMUSets* is constructed. If, for instance, *PMUSets* consists of all possible PMU combinations on an N bus system, then the number of possible PMU configurations to evaluate is:

$$|PMUSets| = \sum_{n=1}^N \binom{N}{n} = \sum_{n=1}^N \frac{N!}{n!(N-n)!} \quad (5.4)$$

Even if the number of PMUs to be placed is fixed, the size of *PMUSets* is likely to make exhaustive search computationally intractable. For instance, placing 4 PMUs on a 37 bus system requires the search of a space that is of size 66,045. If each evaluation of the objective function is carried out in 0.5 seconds, then it would still take 9 hours to process the whole set of possible PMU combinations. In the results presented below, exhaustive search was only used for the placement of up to 6 PMUs due to the amount of time needed to perform exhaustive search. The good thing about exhaustive search is that it is guaranteed to find the global optimal value of the objective function and all PMU placements which attain this optimal value, unlike the simulated annealing method described in the next section.

5.3.2 Simulated annealing

An alternative method which has been used to performing large-scale combinatorial optimization in other PMU placement applications [18, 20] is simulated annealing (SA) [73]. This method is based on the properties of material annealing (heating and then gradual cooling), where atoms within a material will tend to reach a globally minimum energy state if the material is cooled slowly enough. Leveraging this idea to perform optimization, the following algorithm is defined:

1. Initialize the algorithm temperature and configuration:

$$\begin{aligned} T &\leftarrow T_{initial} \\ PMUSet &\leftarrow PMUSet_{Initial} \\ f_{max} &\leftarrow ObjFunc(PMUSet_{Initial}) \\ PMUSet_{max} &\leftarrow PMUSet_{Initial} \\ StallIterations &\leftarrow 0 \\ IterationsAtTemperature &\leftarrow 0 \end{aligned} \quad (5.5)$$

2. Until exit condition is reached:

a. While $IterationsAtTemperature < NumIterationsAtTemperature$ and exit condition has not been met

i. Evaluate a new PMU placement:

$$\begin{aligned}
 StallIterations &\leftarrow StallIterations + 1 \\
 PMUSet' &\leftarrow Neighbor(PMUSet) \\
 \Delta &\leftarrow ObjFunc(PMUSet') - ObjFunc(PMUSet) \\
 \text{if } \Delta > 0 \\
 PMUSet &\leftarrow PMUSet' \\
 \text{if } ObjFunc(PMUSet') > f_{max} & \quad (5.6) \\
 f_{max} &\leftarrow ObjFunc(PMUSet') \\
 PMUSet_{max} &\leftarrow PMUSet' \\
 StallIterations &\leftarrow 0 \\
 \text{elseif } e^{\left(\frac{\Delta}{T}\right)} > Uniform([0,1]) \\
 PMUSet &\leftarrow PMUSet'
 \end{aligned}$$

ii. Check for exit condition

$$\begin{aligned}
 \text{if } StallIterations = StallIterationsMax \\
 \text{exit algorithm} & \quad (5.7)
 \end{aligned}$$

iii. If exit condition has not been met, increment $IterationsAtTemperature$:

$$IterationsAtTemperature \leftarrow IterationsAtTemperature + 1 \quad (5.8)$$

b. Update the temperature and reset $IterationsAtTemperature$

$$\begin{aligned}
 T &\leftarrow 0.99 \times T \\
 IterationsAtTemperature &\leftarrow 0 & (5.9)
 \end{aligned}$$

The algorithm parameters which have not been previously defined are as follows:

- $T, T_{initial}$: T is the current “temperature” in the optimization. High temperatures correspond to hot materials in which the atoms or molecules can move around freely, whereas low temperatures correspond to cold materials where movement is restricted to those movements which will directly lower the energy of the system. The key feature of simulated annealing is that it allows for suboptimal changes in the current $PMUSet$ so that the algorithm does not get stuck in local maxima, as would be the case using a pure

hill-climbing algorithm such as steepest ascent. The mathematical model of this behavior is represented by the conditional acceptance of poorer PMU placements in (5.6) based on the Boltzmann distribution. $T_{initial}$, the initial value of the temperature, is chosen so that essentially any change in the PMU placement is accepted at the start of the algorithm. For the results presented below, this was set to ten times the minimum of P , the number of PMUs in the placement set, and $N - P$, the number of buses without PMUs.

- *IterationsAtTemperature*, *NumIterationsAtTemperature*: The *IterationsAtTemperature* counter keeps track of how many *PMUSet*'s are tested at the current temperature. After *NumIterationsAtTemperature* *PMUSet*'s have been tested at a given temperature, the temperature is adjusted. Several iterations need to be run at each temperature to allow the current state of the system to settle into the probability distribution associated with a given temperature [73]. For the results presented below, *NumIterationsAtTemperature* was set to the minimum of P , the number of PMUs in the placement set, and $N - P$, the number of buses without PMUs.
- *PMUSet_{initial}*: An initial guess at what the optimal PMU placement should be; for a finite number of PMUs, the only restriction is that the reference bus is within this set and that the cardinality of the set is the number of PMUs to be placed. For the results presented below, this was set to include PMUs at the referenced bus and $P-1$ other buses chosen at random, where P is the number of PMUs to be placed.
- *ObjFunc*: One of the three objective functions defined in section 5.2
- *StallIterations*: Keeps track of the number of iterations since the last increase in the objective function due to a new optimal PMU placement set
- *StallIterationsMax*: The maximum number of iterations allowed after the last increase in the objective function before the algorithm exits. In generating the results presented below, this was set to 1000 times the minimum of P , the number of PMUs in the placement set, and $N - P$, the number of buses without PMUs. The main purpose of setting this to a high number is to ensure the local maximum is attained as the temperature reaches low values.
- *Neighbor(PMUSet)*: A function which takes a PMU placement set and returns a new PMU placement set which is a variation on the previous set. For the results presented below, this function randomly selects a PMU from a non-reference bus and moves it to an adjacent bus without a PMU, where two buses are adjacent if there is a line connecting the buses.
- *PMUSet_{max}*, *f_{max}*: The best PMU placement set found at any point during the run of the algorithm, and the optimal objective function value associated with the best PMU placement set.
- *Uniform([0,1])*: A sample from the uniform distribution with a lower bound of zero and upper bound of one.

Using these parameters, SA produces a PMU placement set $PMUSet_{max}$ which would ideally be equal to the true optimal $PMUSet^*$. There is no guarantee that the SA algorithm will reach the true optimal, but as shown in the results below the algorithm does a good job in finding an objective function maximum that is close to the true optimal while drastically reducing the amount of computation needed relative to exhaustive search.

5.4 Results for optimal PMU placement to detect single line outages

As mentioned in the beginning of section 5.2, the impact of filtering can be neglected in the placement problem by first determining $\Delta\theta_{observed}$ for each outage using a high order filter of the simulated angles and calculating the changes in angles several seconds after the outage occurs. Using these $\Delta\theta_{observed}$ values, $RankOutaged$, $AboveThreshold$, and $SharedOutages$ are determined for a given PMU placement set $PMUSet$ and angle threshold τ . These quantities were evaluated using the methods described in section 3.2 and then used to evaluate the objective functions defined in section 5.2. Exhaustive search was used for the placement of three and six PMUs, and SA was used for placement of three, six, nine, eighteen, and twenty-seven PMUs. SA was run three times for each PMU placement problem, and the best result is reported below. The computation time given in the results tables for SA includes all three runs of SA.

5.4.1 Three PMUs placed by exhaustive search and simulated annealing

The first set of results looks at the optimal placement of three PMUs on the system with each objective function using both exhaustive search and simulated annealing. The total number of possible PMU placements for three PMUs is 630.

As seen in Table 5.1, for the $ObjFunc_{Rank1}$ objective function both SA and exhaustive search determine the global optimal of {18, 31, 37} which results in 40 outages being ranked correctly. The time taken to perform the optimization is much lower for exhaustive search in this case due to the small number of possible configurations for a 37 bus case. The results for the other two objective functions are also shown in Table 5.1; the SA results are slightly sub-optimal for one of the objective functions, $ObjFunc_{NoMisrank}$, and the time to perform the optimization is significantly higher, indicating that for very small PMU deployments on this system simulated annealing is not as efficient as exhaustive search. Also noteworthy are the large number of PMU placements which optimize the $ObjFunc_{NoShared}$ objective function. Because so many of the PMU sets optimize this objective, SA can easily find optimal PMU placements and this is reflected in the fact that all three runs of SA result in optimal PMU placements. The $ObjFunc_{NoShared}$ results also illustrate a key shortcoming of SA—namely, that it does not provide an exhaustive list of optimizing placement sets, whereas exhaustive search does.

Table 5.1: Placement of three PMUs using exhaustive search and SA

Objective function	Search method	Highest objective function value	Corresponding $PMUSet(s)$	Time taken to perform optimization (seconds)
$ObjFunc_{Rank1}$	Exhaustive	40	{18, 31, 37}	1.31
$ObjFunc_{Rank1}$	SA	40	{18, 31, 37}	42.71
$ObjFunc_{NoMisrank}$	Exhaustive	15	{1, 9, 31}	1.29
$ObjFunc_{NoMisrank}$	SA	14	{12, 31, 32}	42.03
$ObjFunc_{NoShared}$	Exhaustive	3	{1, 13, 31} {1, 28, 31} {1, 38, 31} {5, 13, 31} {13, 18, 31} {13, 21, 31} {13, 28, 31} {13, 37, 31} {13, 38, 31} {13, 48, 31} {17, 38, 31} {19, 38, 31} {21, 38, 31} {28, 30, 31} {28, 33, 31} {38, 48, 31}	1.58
$ObjFunc_{NoShared}$	SA	3	{13, 31, 37} {13, 31, 48} {17, 31, 38}	43.31

5.4.2 Six PMUs placed by exhaustive search and simulated annealing

Next, the placement of six PMUs on the system is examined using both exhaustive search and simulated annealing. Placing six PMUs pushes the bounds of how many PMUs can be placed in a reasonable amount of time with exhaustive search; doing the same search for seven PMUs would take approximately four times longer, resulting in search times on the order of

hours rather than minutes. Going beyond seven PMUs would require days of computation if exhaustive search were used.

Table 5.2: Placement of six PMUs using exhaustive search and SA

Objective function	Search method	Highest objective function value	Corresponding $PMUSet(s)$	Time taken to perform optimization (seconds)
$ObjFunc_{Rank1}$	Exhaustive	49	{1,3,5,21,55,31}	589.95
$ObjFunc_{Rank1}$	SA	46	{3,5,10,21,31,54} {13,21,24,31,34,37}	98.04
$ObjFunc_{NoMisrank}$	Exhaustive	31	{10,15,21,48,56,31}	620.55
$ObjFunc_{NoMisrank}$	SA	21	{12,24,28,30,31,40}	90.03
$ObjFunc_{NoShared}$	Exhaustive	9	{1,21,24,28,38,31} {1,24,28,38,48,31} {1,24,28,38,53,31}	710.72
$ObjFunc_{NoShared}$	SA	8	{19,28,30,31,38,48}	118.15

The results shown in Table 5.2 illustrate the difference in computation time for SA relative to exhaustive search—there is a roughly fivefold decrease in computation time. This shows that SA scales much better with the number of PMUs to be placed than exhaustive search does. In addition, SA has many more parameters which can be tweaked, particularly the exit condition *StallIterationsMax*, which can be modified to reduce this computational time.

Table 5.2 also showcases the other main issue in using an approximate optimizer such as simulated annealing—the SA result is sub-optimal for all three objective functions, and is significantly suboptimal for the $ObjFunc_{NoMisrank}$ objective function. One possible reason for this behavior is that the objective function has infinitely steep derivatives in the presence of misranked events. This can prevent the SA algorithm from transitioning between configurations, particularly as the temperature gets low and the algorithm stops allowing suboptimal transitions. One solution would be to reformulate the objective function so that the penalty for having misranked events is less than infinity, although the choice of penalty would have to be tailored to each system and event set in order to continue avoiding misranking. Another option would be to try additional combinatorial optimization methods (e.g., genetic algorithms [74], particle swarm [75], or a hybrid of several methods as in [76]). Ultimately, there is no way to avoid the fact that with an extremely large search space tradeoffs must be made between computational effort and optimality.

5.4.3 Higher numbers of PMUs placed by simulated annealing

The final set of results illustrate the performance of SA for the placement of PMUs in which exhaustive search is computationally infeasible. SA was run once for each combination of objective function and number of PMUs P . The results of these runs are given in Table 5.3. The steady increase in optimal values of the three objective functions indicate that SA is finding better PMU placements as the number of PMUs increases. This fits well with the results from section 4.1.1.2, where the reduction in PMUs from 37 to 18 led to slightly poorer rankings and more events with non-zero *SharedOutaged* values. Moreover, these results show that if 27 PMUs are placed on this system, there exists a PMU placement and angle threshold such that 51 of the 56 events are correctly detected without any misrankings. This confirms the usefulness of the single outage detection algorithm for reduced PMU placements, particularly if misranking is to be minimized. Also, although these results are based on the dynamic simulations with the initial conditions given in Appendix A, running SA with different system configurations (e.g., summer peak, winter peak, midday, midnight, etc.) would be feasible due to the small amount of time needed to perform the search. On the other hand, testing a wide variety of system configurations would be prohibitively expensive from a computation standpoint if a more exhaustive search approach is used.

Table 5.3: Placement of 9, 18, and 27 PMUs by simulated annealing

Objective function	Number of PMUs (P)	Highest objective function value	Time taken to perform optimization (seconds)
$ObjFunc_{Rank1}$	9	47	65.33
$ObjFunc_{NoMisrank}$	9	26	56.41
$ObjFunc_{NoShared}$	9	11	55.74
$ObjFunc_{Rank1}$	18	52	148.76
$ObjFunc_{NoMisrank}$	18	38	121.96
$ObjFunc_{NoShared}$	18	24	172.83
$ObjFunc_{Rank1}$	27	54	76.52
$ObjFunc_{NoMisrank}$	27	51	62.87
$ObjFunc_{NoShared}$	27	37	84.99

6 GPU-Based Visualization of PMU Data

6.1 Overview

The high data rate of phasor measurement units requires sufficiently fast rendering if real-time visualization is to be achieved. Graphical processing units (GPUs) have shown remarkable progress in the past decade, providing unprecedented computational power and programmability for advanced graphics applications. The use of GPUs to render PMU data is a natural fit between the high data rate of PMUs and the high rendering rate of GPU-based visualization algorithms. Accordingly, two applications have been developed and are presented in the remainder of this chapter. The first section considers how one of the more common visualization techniques, bus data contouring, can be significantly accelerated by using GPU rendering. The second section discusses some additional processing of the data that can make the data visualization more content-rich. Analysis of these techniques shows how combining cutting-edge visualization techniques with cutting-edge measurement techniques can provide significant improvements in situational awareness.

6.2 Implementation of contouring on the GPU

6.2.1 Introduction to contouring on the GPU

Based on the recommendations of the August 2003 blackout report [1], a heavy emphasis has been placed on improving situational awareness in control centers around the world. One visualization technique which has been used to increase situational awareness has been contouring [40]. This technique, also known as scattered data interpolation (see [77] for a survey of the state-of-the-art), aims to provide a weather map-like visualization of the power grid, showing various bus- or substation-based data throughout an area of interest. This type of visualization has been used in many different fields, e.g., meteorology [78] and medical imagery [79], and its usefulness has been demonstrated for power system applications through human factors testing [42].

One key similarity between the methods used to generate power system contours in the past has been a reliance on the CPU to do the necessary contour calculations. CPUs, which are optimized for general purpose computing, are not well-suited for the parallel calculations that are inherent in some contouring algorithms. Several incremental improvements have been made, but CPU-based contouring is fundamentally constrained by the serial nature of CPU program execution. On the other hand, graphics processing units (GPUs), which are present in most modern computer systems, are ideally suited to the computations needed to generate contours because of the parallel nature of GPU program execution. Furthermore, some work has already

been done in demonstrating the effectiveness of using GPUs to perform real-time interpolation of three- and four-dimensional data [48]. This work suggests that there is a potentially large benefit to using GPUs in two-dimensional power system contours.

In order to investigate the possible performance gains of using GPU- rather than CPU-based contouring for power system visualizations, a prototype GPU-based contouring algorithm has been developed. The remainder of this chapter discusses this work, and is divided into several sections. Section 6.2.2 provides a brief overview of the contouring problem. Section 6.2.3 discusses some of the fundamental aspects of GPU programming, while section 6.2.4 provides the algorithm and implementation details. Section 6.2.5 provides some performance results for contouring a real power system area as parameters such as influence radius and contour resolution are varied. Section 6.3 discusses some of the key advantages and disadvantages of moving contouring to the GPU, and section 6.3 details several benefits and potential applications of GPU-based contouring.

6.2.2 Contouring background

Formally, the purpose of contouring (or scattered data interpolation) in the power system context is to find a function $F(x, y)$ such that for each bus k , located at position (x_k, y_k) :

$$F(x_k, y_k) = f_k \quad (6.1)$$

where f_k is the value to be contoured (e.g., voltage), known to be a specific value at each bus k . This value is typically obtained from a state estimator or directly from a measurement devices such as a PMU. There are an infinite number of possible functions that can satisfy this constraint, so additional restrictions are useful in reducing the set of possible functions. One additional constraint imposed on F in this application is that F must be “smooth”, i.e., continuous and once differentiable [80].

While there are many different methods which can be used to determine the contouring function F (see [77]), this work focuses on inverse distance weighted methods due to their intuitive nature and ease of parallelization. The intuition behind inverse distance weighted methods is that the function value at any point should be the weighted average of all points with known values. In this method of contouring, the function F is defined as follows:

$$F(x, y) = \frac{\sum_{k=1}^N w_k(x, y) f_k}{\sum_{k=1}^N w_k(x, y)} \quad (6.2)$$

, where $w_k(x, y)$ is known as the weighting function. The weighting function first proposed for this scheme, which gives rise to the contouring algorithm known as Shepherd’s Method [81] is:

$$d_k = \sqrt{(x - x_k)^2 + (y - y_k)^2}$$

$$w_{k,SM}(x, y) = \frac{1}{d_k^2} \quad (6.3)$$

This weighting function gives a nonzero value of $w_k(x, y)$ for all values of x and y . Because each bus k has a non-zero contribution to every point on the screen, using the weighting function of equation (6.3) is characterized as a global contouring algorithm.

Global methods such as Shepherd's Method are computationally burdensome due to the need to consider each bus' contribution at each pixel on the screen. As a result, several local methods have been proposed in the literature [80]. For the implementation of contouring on the GPU, the local weighting function developed by Franke and Little is used:

$$w_{k,FL}(x, y) = \begin{cases} \frac{(R - d_k)^2}{R d_k} & , d_k < R \\ 0 & , d_k \geq R \end{cases} \quad (6.4)$$

, where d_k is defined in (6.3). For this local weighting function, only pixels within a distance R of bus k will be affected by the value f_k . This weighting function also maintains continuity and first-order differentiability of the interpolating function F [82] and reduces to the weighting function of (6.3) as $R \rightarrow \infty$.

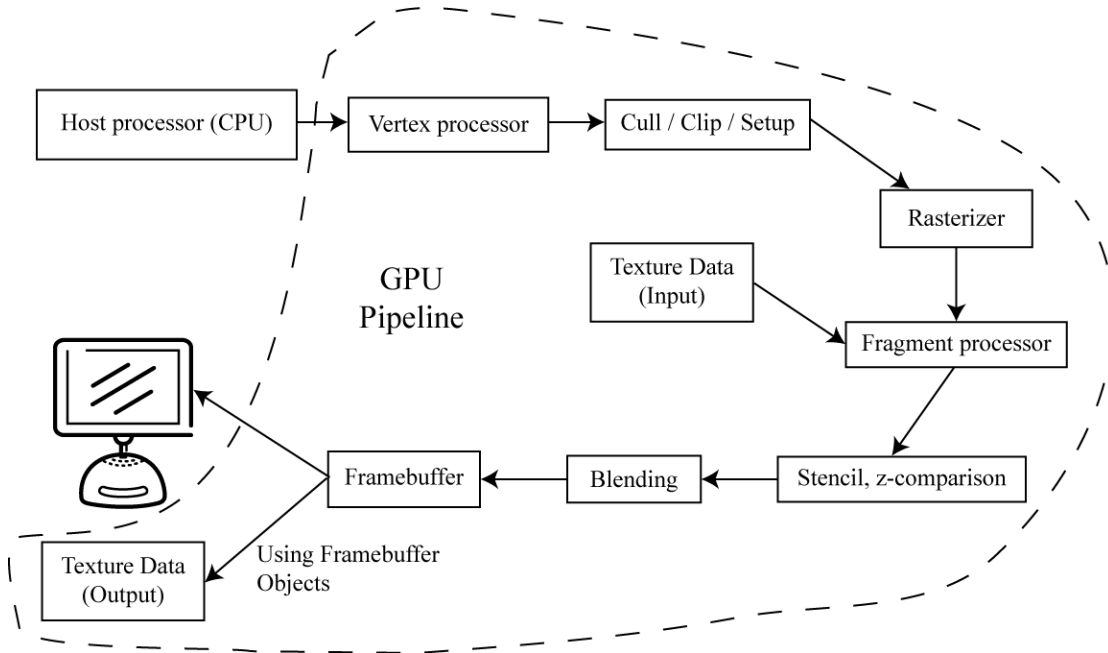


Figure 6.1: The GPU rendering pipeline

6.2.3 GPU programming[83]

6.2.3.1 The rendering pipeline

GPU programming is significantly different from CPU programming, and this must be taken into account when adapting any algorithm for GPU implementation. The basic GPU pipeline, illustrated in Figure 6.1, was originally constructed to efficiently render polygons, lines, and points to a graphical display. The basic steps in the GPU rendering pipeline are [83]:

1. Polygon vertex coordinates, viewable area definitions, color values, texture coordinates, and textures are sent to the GPU via OpenGL calls from the CPU.
2. The vertex processor transforms coordinates defined via function calls from the CPU in step 1 into coordinates relative to the display. Also, vertices are grouped into primitives (points, lines, and triangles) for the later stages of the pipeline.
3. The Cull / Clip / Setup stage takes the primitives that are output from the vertex processor and removes unviewable objects. Also, any primitives that are only partially inside the viewable area are clipped.
4. The rasterization stage determines which fragments are covered by each primitive. Fragments are the rasterized pieces of the polygon and correspond to a single pixel of the final output. Also, per-vertex values such as colors and texture coordinates are linearly interpolated across each primitive and assigned to each enclosed fragment.
5. The fragment processor runs on each fragment, performing texture lookups and determining the final color that the fragment will be. For this contour implementation, two non-default fragment processors are used to change the behavior of this step.
6. Once fragment colors have been assigned by the fragment processor, several tests (e.g., stencil and z-compare) are run on the fragments to see if they should be discarded. If the pixel passes all tests, then it is blended with the current framebuffer entry at its location and the final pixel value is written to the framebuffer. This write to the framebuffer is the final stage of the OpenGL pipeline. For traditional graphics applications, the framebuffer represents the pixels on the current video display. Framebuffer objects (FBOs) make it possible to draw into a texture in video memory rather than directly to the video display [84]. This texture can then be read back on a subsequent rendering pass. This technique is used in the GPU-based contouring implementation, as discussed below.

6.2.3.2 Important fragment processor characteristics

Two of the programmable units on the GPU are the vertex processor and the fragment processor, which allow for custom processing of vertices and fragments in stages 2 and 5 of the graphics pipeline. The default vertex processor is used in the contouring implementation, so it is not discussed any further. On the other hand, two custom fragment processors are used for contour generation.

One important aspect of fragment processor programming is that there can be fragments at different stages of the pipeline at any given time. As a result, the fragment processor is not allowed to write to any locations in video memory except for the framebuffer at the end of stage 5. Otherwise, each fragment might have to wait for another fragment to complete its processing before it can be sent out to the framebuffer. In other words, if the framebuffer is conceptualized as an array of bytes, then the fragment processor can only output data to one particular array index in the framebuffer, and this array index cannot be changed from within the fragment processor. Although it cannot write directly to texture memory, the fragment processor is capable of reading directly from texture memory; in fact, this is how traditional polygonal texturing is performed for graphics applications. Because of these features, it is helpful to think of a fragment processor as having a large set of memory addresses it can read from, and only one memory address it can write to, with no overlap between these address spaces. This is fundamentally different from CPU programs, where programs routinely read and write to the same memory address.

Although fragment processors must be programmed with a restrictive set of operations, particularly when it comes to memory reads and writes, their sheer processing power makes them ideal for speeding up computations that can be performed in parallel. For instance, modern GPUs such as the NVIDIA GeForce 7-series used in these experiments are capable of performing over 165 billion floating point operations per second (gigaflops or Gflops), whereas a CPU of the same generation, such as a dual-core Pentium 4, is only capable of around 24.6 Gflops [45]. More advanced chips, such as the recently released GeForce 8800 GTX, have theoretical processing speeds in excess of 500 Gflops. For cases where the GPU can be run near its processing limits, this difference in processing power allows GPUs to easily outperform CPUs.

There are two key issues involved in fully taking advantage of the GPU's computational power. First, memory reads should be performed as sequentially as possible in order to take advantage of hardware caches built into the GPU. The GPU contouring implementation is designed to ensure coherent texture reads by making all reads from rectangular textures, where each texel is accessed in a sequential fashion from the top-left to the bottom-right. The other key issue in programming for the GPU is to use algorithms which are computationally limited (as opposed to I/O limited). Contouring algorithms clearly fall into this category, as the bulk of the time spent in creating a contour is based on the need to evaluate (6.4) for each pixel within the influence radius influence region of each bus. Also, because the processing elements of the GPU work in parallel, defining the algorithm using for...next loops allows for simple implementation as a fragment processor program.

6.2.4 Algorithm definition and implementation details

6.2.4.1 Definition of the contouring algorithm

Based on equations (6.2) and (6.4), one definition of the contouring algorithm which is well-suited for GPU implementation is:

1. For each point (x, y)
 - 1.1. $N(x, y) \leftarrow 0$
 - 1.2. $D(x, y) \leftarrow 0$
 - 1.3. $F(x, y) \leftarrow 0$
2. For each bus k
 - 2.1. For each point (x, y) such that $w_{k,FL}(x, y) > 0$
 - 2.1.1. $N(x, y) \leftarrow N(x, y) + w_{k,FL}(x, y) f_k$
 - 2.1.2. $D(x, y) \leftarrow D(x, y) + w_{k,FL}(x, y)$
3. For each point (x, y) such that $D(x, y) > 0$
 - 3.1. $F(x, y) = \frac{N(x, y)}{D(x, y)}$
4. For display to the screen, each value of F is mapped to a color as discussed in [40] and stored as the color value for point (x, y)

Table 6.1: Algorithm implementation on the CPU and the GPU

Algorithm Steps	CPU Implementation	GPU Implementation
1	Allocate space for N and D in main system memory.	Allocate space for <i>AccumTexture</i> and <i>ContourTexture</i> on the GPU.
2	For each bus, calculate and store the numerator and denominator values within each bus' influence region.	Draw a circle of radius R around each bus with the video output redirected to <i>AccumTexture</i> . Use a custom fragment shader to calculate and store the N values in the red channel and the D values in the green channel. Use additive blending to perform accumulation as each bus is drawn.
3	Iterate through each element in the N and D arrays and divide the numerator by the	Draw a rectangle of size Res with its texture set to <i>AccumTexture</i> and the video output redirected to <i>ContourTexture</i> . Using a custom fragment shader, divide the red

	denominator. Look up the corresponding color in the color map and write this color to a bitmap.	channel values of <i>AccumTexture</i> by the green channel values. For each pixel, look up the corresponding color in the color map and write this color to <i>ContourTexture</i> .
--	---	---

Each of the algorithm steps given above can be translated into a set of instructions on the CPU or GPU, as shown in Table 6.1. For the GPU implementation, the C programming language was used with the GLUT library [85] for OpenGL programming and the Cg language was used for fragment processor programming [86]. The implementation of the algorithm was split into three stages, explained in detail below.

6.2.4.2 Implementation step one—allocation of textures and framebuffer objects

To begin, texture memory is allocated for two textures: *AccumTexture* and *ContourTexture*. *AccumTexture* is created as a GL_RGBA16F_ARB formatted texture, which allocates four components (red, green, blue, and alpha) for each texel. Each component stores an IEEE-formatted 16-bit floating point value. *ContourTexture* is allocated as a standard 32-bit texture containing 8-bit red, green, blue, and alpha components for each texel. *AccumTexture* stores $N(x, y)$ in its red channel and $D(x, y)$ in its green channel. *ContourTexture* stores the finished, color-mapped contour as a texture which can then be rendered to the display. Both textures are created with the same dimensions, denoted as *Res*. This size is the resolution of the contour to be created (e.g., if a contour is to be created which exactly fits into a 640x480 window, then the two textures will have dimensions 640x480). Because the algorithm requires rendering to these textures rather than simply reading from them, a framebuffer object is created which allows the GPU to write to the textures [84]. The framebuffer objects used to write to each texture are named *AccumFBO* and *ContourFBO*. The allocation of texture memory occurs only when the desired contour resolution changes (typically when a window is resized) to reduce the which reduces the amount of time spent allocating memory on the GPU.

6.2.4.3 Implementation step two—accumulation of the numerator and denominator values for each pixel

Once the textures are allocated, *AccumFBO* is set as the output framebuffer for the GPU. By setting *AccumFBO* as the output location, all pixels drawn by the GPU are sent to *AccumTexture* rather than the video display. After *AccumFBO* is attached, *glClear()* is called, which performs steps 1.1 and 1.2 of the algorithm by clearing the red and green channels of *AccumTexture*. Next, a custom fragment processor is bound to the GPU that takes as arguments the position of the bus that is currently being drawn, (x_k, y_k) , and writes out the values $w_{k,FL}(x, y) f_k$ to the red channel and $w_{k,FL}(x, y)$ to the green channel. The final setup step is to enable additive blending on the GPU which performs the assignment and sum

operations needed in steps 2.1.1 and 2.1.2 of the algorithm. Once all the setup operations are completed, for each bus k , a circle of radius R is drawn around the point (x_k, y_k) . After this has been done for each bus in the system, the red channel of *AccumTexture* contains $N(x, y)$ and the green channel contains $D(x, y)$ for all points in the contour.

6.2.4.4 Implementation step three—evaluation of the interpolating function and color mapping

For the next step in the contouring process, *ContourFBO* is attached as the render target for the GPU. A call to `glClear()` is then made to execute step 1.3 of the algorithm. Next, *AccumTexture* is set as the current texture to be read within the fragment processor of the GPU. A custom fragment program is then bound to the GPU which performs steps 3.1 and 3.2. Finally, a rectangle of size *Res* is drawn with texture coordinates assigned so that each point in *AccumTexture* corresponds to one output pixel. Once this draw operation has completed, *ContourTexture* contains the color-mapped contour. This texture can then be drawn to the screen by binding *ContourTexture* as the current texture and rendering an appropriately sized rectangle using the default fragment processor.

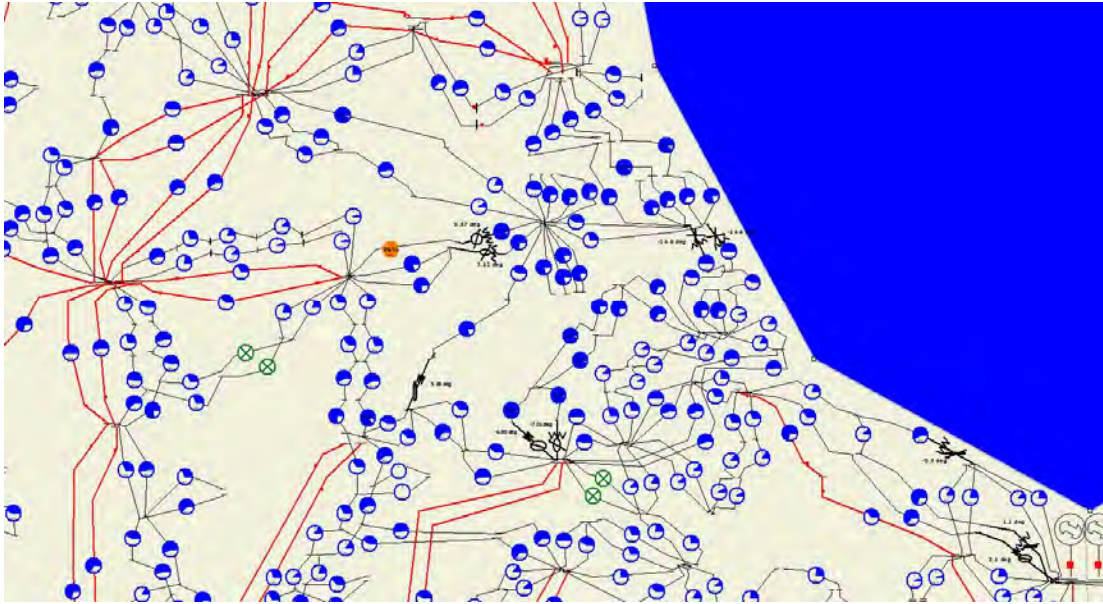


Figure 6.2: Oneline diagram of contour area

6.2.5 GPU-based contouring algorithm performance results

For the results given below, the test system used consisted of an AMD Athlon 64 X2 Dual Core 3800+ CPU with an NVIDIA GeForce 7600 GT GPU. Timings for the GPU algorithm were determined using the high-resolution Windows performance counter [87], with an average taken over 100 contour renderings.

The power system one-line diagram used as the basis for the results and figures shown below is provided in Figure 6.2. The number of buses that influence the contour area ranges from 199 (for an influence radius of 1) to 818 (for an influence radius of 400). The data that is contoured is the per unit voltage level at each bus. The color map used to convert voltage levels to colors is shown in Figure 6.3.



Figure 6.3: Contour color map relating per unit voltage to displayed color

6.2.5.1 The effect of changes in Res

There are two key parameters which affect the performance of the contouring algorithm. The first of these parameters, Res , controls how large of a contour image to create. For the highest-quality contour, Res should be set to the same size as the screen area the contour covers upon final rendering to the video display. However, reducing Res to a smaller value results in a decrease in rendering time; therefore, a tradeoff must sometimes be made between contour resolution and rendering speed. To explore the effects of varying Res , the system was contoured with several different resolution levels. In these tests, R , the radius of influence, was held at a constant value of 100. Figure 6.4 shows an example contour rendering with Res set to 1024 x 768.

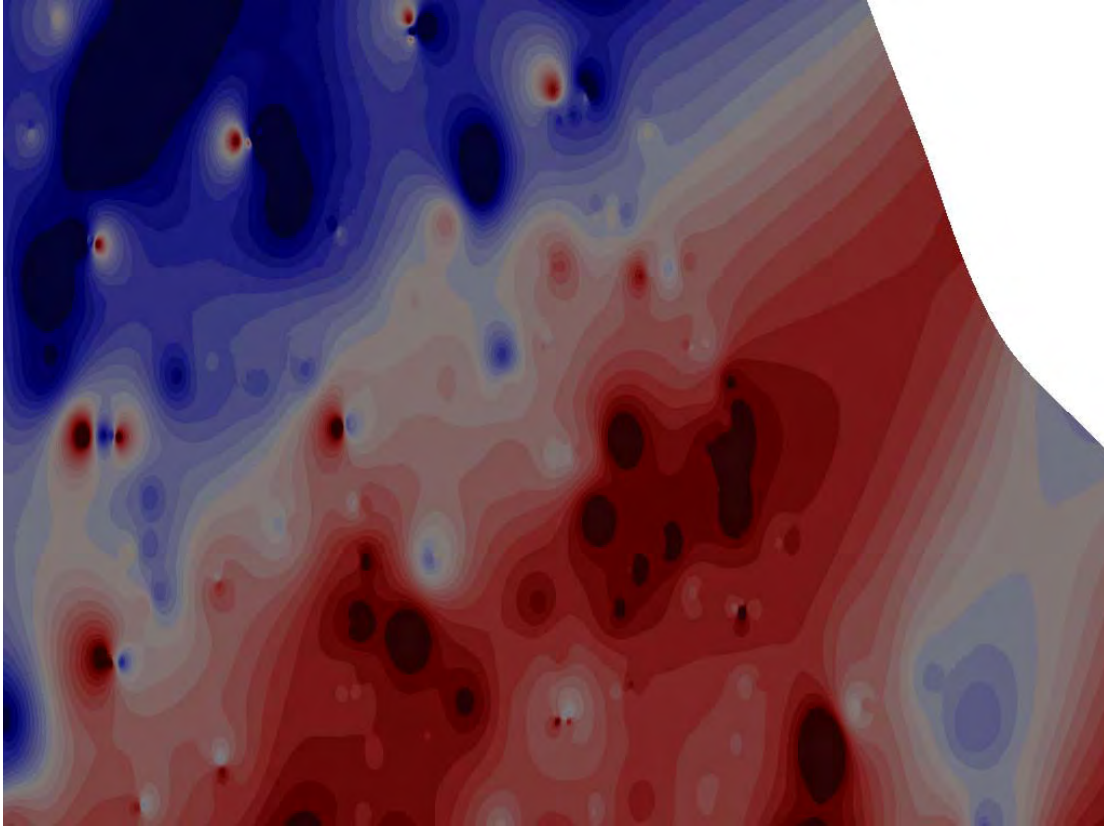


Figure 6.4: Contour rendering with resolution set to 1024x768, radius of influence set to 100

Table 6.2 shows the timing results obtained for various contour resolutions. In order to gauge the performance of the algorithm in a typical control room setting, resolutions were chosen which are commonly used in both projection and conventional video displays.

Table 6.2: Timings for seven contour resolutions with the influence radius set to 100

<i>Res</i>			
Width	Height	Common Name	Time to render a contour (seconds)
3840	2400	WQUXGA	0.938
1920	1200	WUXGA	0.249
1600	1200	UXGA	0.208
1280	1024	XGA+	0.150
1024	768	XGA	0.100
800	600	SVGA	0.066
640	480	VGA	0.050

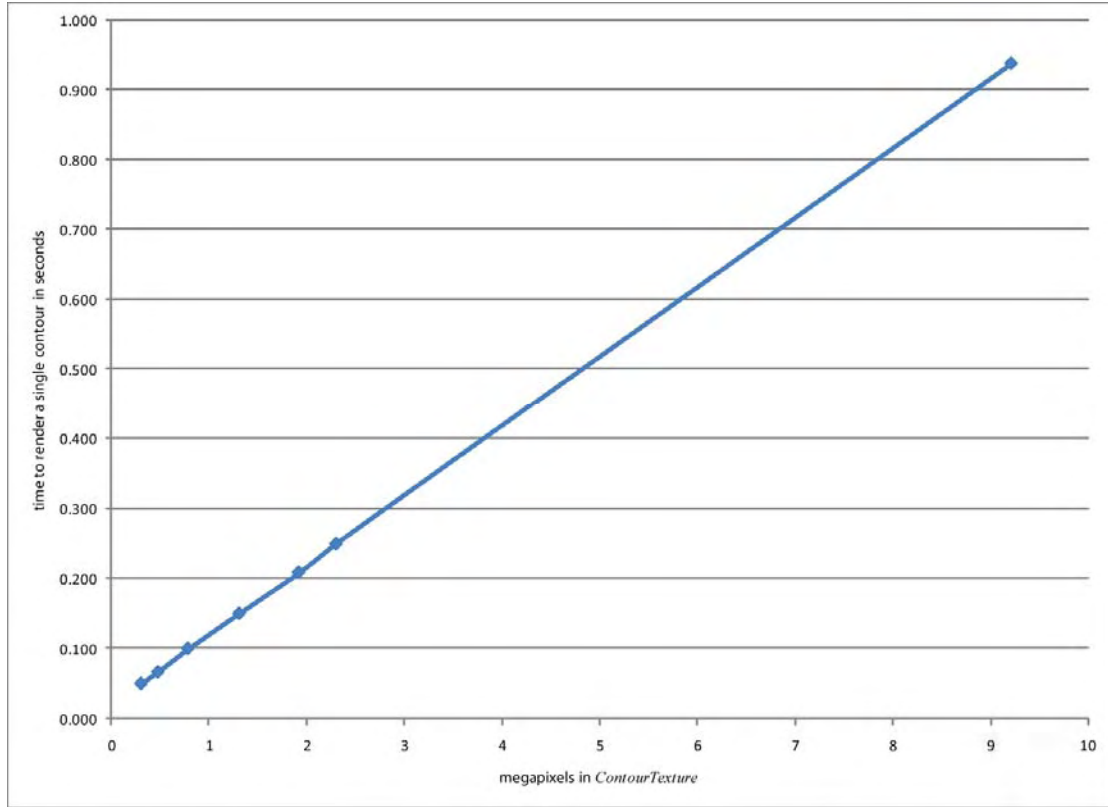


Figure 6.5: Effect of contour resolution on contour rendering times

Figure 6.5 illustrates the effect on rendering time as the contour resolution is varied. The x-axis is the number of megapixels associated with each tested resolution, and the y-axis is the amount of time it takes to render one contour. The linear relationship between resolution and rendering time indicates that the time required to render a particular contour is on the order of the number of pixels in the contour. The fact that this linear relationship holds for resolutions ranging from 640x480 all the way up to 3840x2400 indicates that the algorithm does not suffer from scalability issues.

It is also worthwhile to consider the visual impact of changing the contour resolution. For a given screen size Res_{Screen} , the highest quality contour would be obtained by setting $Res = Res_{Screen}$, assuming the contour takes up the entire screen display. However, by setting $Res < Res_{Screen}$, the speed at which the contour is rendered can be greatly accelerated. In addition, rendering the contour at a lower resolution is not necessary noticeable. Finally, the usage of advanced texture filtering methods can also help to improve the rendered quality of a lower resolution contour.

6.2.5.2 The effect of changes in R

The other key parameter which affects contour accuracy and rendering speed is the radius of influence, R . Because a circle of radius R is drawn around each bus (i.e., each bus affects a

locus of points of radius R around its location), the contour rendering time should scale in proportion to the area of the circle drawn around each bus, i.e., quadratically in R . However, for a contour with a finite area, this relationship does not always hold.

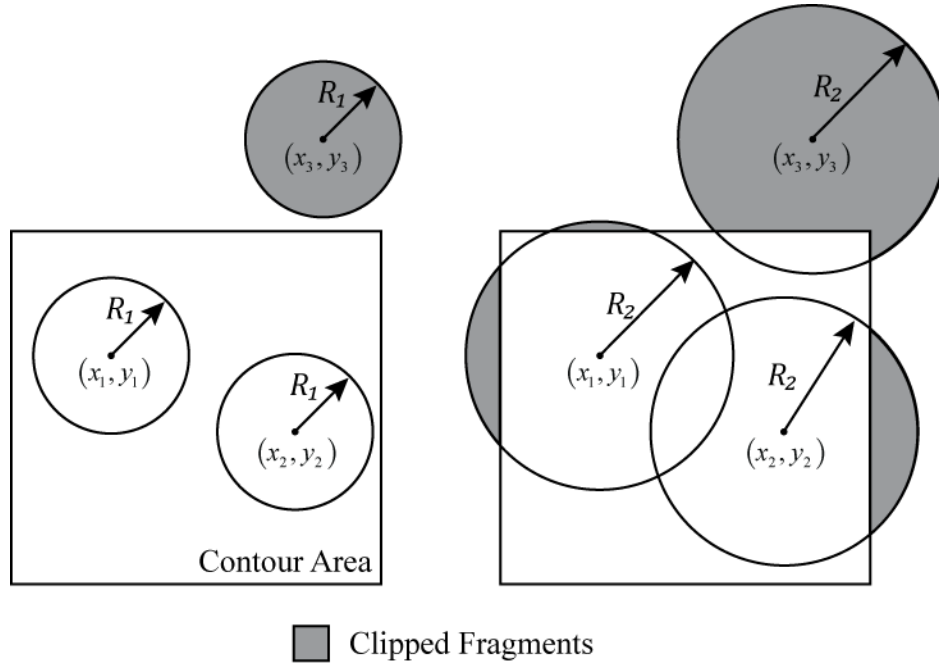


Figure 6.6: Effect of fragment culling on radius of influence effects

Figure 6.6 illustrates a simple three-bus contouring example where the relationship between R and the number of processed fragments is not quadratic. The left side of the figure shows the case where the circles of radius R_1 drawn around buses 1 and 2 fits entirely within the contour area, whereas the circle around bus 3 is entirely outside the contour area. Because the circle drawn around bus 3 does not intersect the contour area, there is no need to draw the circle around bus 3. On the right side of the figure, as the radius is increased to R_2 , the circles surrounding all three buses are partially inside of the contour area. In this case, a circle is drawn around each bus, but the GPU clips away the portions of each circle which lie outside the contour area before the fragment processor stage [83]. As a result, steps 2.1.1 and 2.1.2 of the algorithm are skipped for the darkened areas of each circle. Based on Figure 6.6, it is clear that linear increase in the influence radius will cause a quadratic increase in computation time only if the set of buses influencing the contour area does not change and all bus influence regions are entirely enclosed in the contour area. Otherwise, the increase in computation time is dependent on the characteristics of the area being contoured and the distribution of the influencing buses.

To examine the effect of changing the influence radius, several timing measurements were taken with a fixed contour resolution of 1024x768 while R was varied from 1 to 400. The

results of these tests are given in Table 6.3. In addition, Figure 6.7 illustrates the complex relationship between the radius of influence and the time to construct a single contour.

Table 6.3: Timing results for several values of influence radius
with a constant contour resolution of 1024x768

R	Number of buses influencing the contour area	Time to render a contour (seconds)
1	199	0.0225
5	206	0.0227
10	221	0.0229
25	274	0.0249
50	367	0.0361
75	438	0.0665
100	485	0.1001
125	514	0.1331
150	580	0.1838
200	653	0.2669
300	753	0.4251
400	818	0.4994

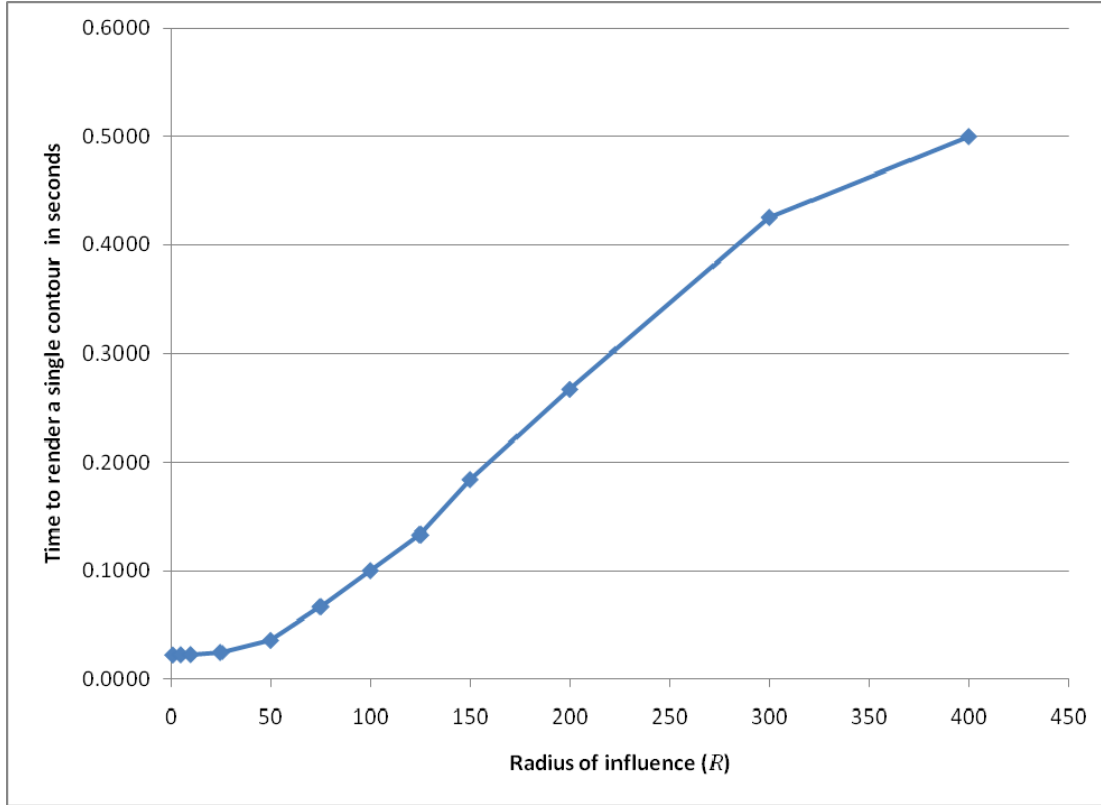


Figure 6.7: Effect of influence radius on contour rendering time with contour resolution set to 1024x768

For the first few values of influence radius—1, 5, and 10—the time to render a contour is basically constant. This indicates that for very low influence radii, the time spent rendering a contour is based more on other factors (such as the texture lookups, OpenGL calls, etc.) than on the processing time needed to construct the contour. As the influence radius increases to values between 10 and 150, a quadratic relationship is evident. For radii above 150, a much more complicated relationship is demonstrated that confirms the behavior discussed above and illustrated in Figure 6.6.

As with the other tunable parameter Res , there is a tradeoff between how large of an influence radius is used and how fast the contours are rendered. In general, the proper choice of R will be dependent on many factors, including the density of buses within the power system under study. In fact, this is one justification for the usage of dynamic influence regions when contouring power system data, as discussed in [44].

To illustrate the visual impact of changing the influence radius, Figure 6.8 shows how reducing the influence radius from 100 to 25 affects the contour shown in Figure 6.4. In Figure 6.8, it is very easy to distinguish the circles that surround each bus, and the silhouette of the contour is a series of arc segments, typical for contours with relatively low influence radii. On the other hand, in Figure 6.4, a much smoother contour is shown with no obvious circles present.

Ultimately, the individual using the contour must decide on which value of R provides a good representation of the underlying data without sacrificing too much in terms of rendering speed.

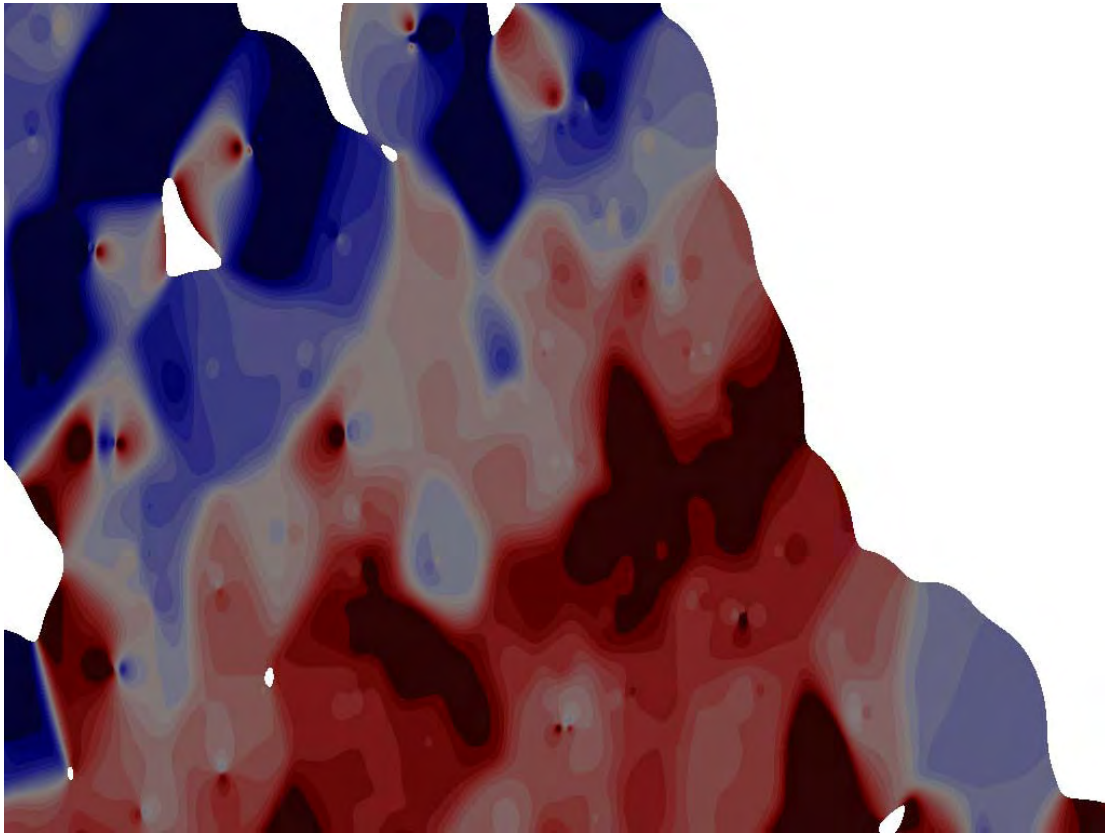


Figure 6.8: Contour rendering with resolution set to 1024x768, radius of influence set to 25

6.2.6 Advantages and disadvantages of GPU-based contouring versus CPU-based contouring

Because implementation of a GPU-based contouring algorithm can require significant expenditure of time and resources, it must be justified as an alternative to current CPU-based contouring methods. The greatest benefit of moving contouring to the GPU is the decrease in rendering time for contours. For example, contouring the power system area in Figure 6.2 using a resolution of 640x640, with a radius of influence of 150, the GPU-based contour takes 0.102 seconds to render. Rendering the same contour using the CPU (with the method outlined in [44]) takes 5.10 seconds.

The rendering speeds of CPU-based contouring methods are significantly slower than GPU-based methods for several reasons. First, CPUs are designed to process instructions in a

serial fashion, rather than for parallel operations. GPUs, on the other hand, have always been built to process multiple vertices and fragments in parallel to quickly render polygons to the screen. For instance, the NVIDIA GeForce 7600 GT GPU used in the above tests has 12 fragment processors, each capable of performing steps 2.1.1 and 2.1.2 of the algorithm independently. On a CPU, each of these operations must be performed serially. Secondly, modern CPUs typically have a 6.4 GB/sec memory bandwidth, whereas modern GPUs have memory bandwidths of 32 GB/sec and higher [83]. As a result, reading and writing to memory locations can occur at a much faster rate on the GPU. Finally, the clipping operations built into GPUs are able to efficiently exclude calculations which do not influence the contour plot (e.g., calculations on the shaded pixels in Figure 6.6). On the other hand, when using a CPU to perform contouring, the effects of clipping at the contour boundaries must be explicitly calculated. Another key benefit of using GPUs for contouring is that the CPU is then freed up to perform tasks better-suited for CPU computation. For instance, while the CPU is used to calculate load flows or perform filtering operations, the GPU can independently construct and render a contour with limited CPU interaction.

GPU-based contouring also has several disadvantages when compared to CPU-based contouring. The greatest potential disadvantage to using GPU-based contouring is the potential loss in accuracy due to floating point errors. Although many of the latest GPUs support 32-bit operations within the fragment and vertex processors, using 32-bit floating point incurs a performance penalty relative to 16-bit operations. Another potential difficulty with using 32-bit floating point textures is that blending of 32-bit floating point values, needed to implement steps 2.1.1 and 2.1.2 of the algorithm, is only possible on the very latest (and most expensive) video cards such as the NVIDIA GeForce 8- and 9-series. Usage of older GPUs with 16-bit precision can lead to numerical instability, a problem which does not occur with modern CPUs that are capable of handling arbitrarily high precision floating point values.

Another potential problem with using GPU-based contouring is the wide variety of capabilities of GPUs. Although there are some standards for GPU programmability as defined by the OpenGL Architecture Review Board, it is nontrivial to write a GPU contouring program which takes advantage of all GPUs equally. For instance, writing textures with an NVIDIA GPU is optimized for a different set of OpenGL instructions than an ATI GPU. CPUs do not typically suffer as much from this difficulty due to the existence of well-established instruction sets.

6.3 Benefits and applications of accelerated contouring

6.3.1 Improvements in usability

The ability to render faster contours is not inherently useful; the usefulness comes from the improvements in usability and new applications which are enabled by the faster rendering rates. Two key criteria used to evaluate a system's usability are the efficiency of the interface, typically

measured by the amount of time needed to complete a specific task, and user satisfaction in using the interface, which can be measured by both subjective and objective means [88]. The improvements in contour rendering times facilitated by using the GPU impact both of these criteria.

6.3.1.1 Improvements to user efficiency

There are two common tasks which are typically used to interact with a power system contour—navigation and changing of contour parameters. The first of these tasks, navigation, consists primarily of zooming and panning the contour area. For example, if an operator wants to see a more detailed voltage contour within a particular area, this requires the operator to first select the area of detail and then wait for the contour to be rendered again. As mentioned above, CPU-based contouring can take up to five times as long as GPU-based contouring of the same area. In terms of user efficiency, this means that after the user has signaled to the software that the contour area needs to change, it takes a significantly longer time for CPU contouring to catch up with the user's request. As a result, using GPU-based contouring can significantly improve the efficiency of users' interactions. Furthermore, any attempt by the user to fine tune either zooming or panning is better facilitated by the rapid contouring speeds enabled by the method presented in section 6.2.

Changing contouring parameters such as influence radius, resolution, and color mapping are also ways in which users interact with contours to better highlight certain aspects of the underlying data. For example, if a color mapping is initially defined for a range of 0.95 to 1.05 per unit voltage, it may be necessary to change the range to highlight areas where the voltage is lower than 0.95 or higher than 1.05. The speed at which the contour can be regenerated directly impacts the amount of time it takes for any such changes to take effect. Therefore, this is another type of task in which improvements to contour rendering can significantly increase the efficiency of using contours to understand power system behavior.

6.3.1.2 Improvements to user satisfaction

It has been shown in many previous studies that frame rate is directly related to the both performance and satisfaction of users when interacting with computer systems (see [49] for a survey of more than 50 such studies). Figure 3 in [49], which summarizes the satisfaction findings from 16 different studies, shows that the lowest acceptable refresh rate over a wide variety of tasks is 5 Hz. Meeting this target would require contour rendering that takes 0.2 seconds or less, a goal that is not easily achievable with CPU-based contouring. One way to get these speeds using CPU contouring is to reduce the contour resolution, but evidence suggests that lowering resolution can have as much of an impact on user satisfaction as having a low frame rate [89]. Because GPUs allow contours to be rendered an order of magnitude faster than CPU contours, there should be marked increase in user satisfaction if contouring is performed on the GPU instead. In addition, one study suggests that increasing the frame rate can actually

reduce the stress of operators using the visualization [90], which can be particularly important if the system operators are already in a state of heightened stress due to poor system conditions.

6.3.2 Visualizing faster data

Because the rendering times for contours using GPUs are very short, it should be possible to display data that arrives at a much faster rate than if CPU-based contouring is used. Using the aforementioned rendering times of 0.102 seconds and 5.10 seconds for GPU- and CPU-based creation of a single contour, the CPU-based method could only keep up with a data rate below 0.2 Hz, whereas the GPU-based contouring method could keep with a data rate of 10 Hz. CPU-based methods have been adequate in the past for state estimator visualization, because the data coming out of the state estimator changes around once per minute, or 0.02 Hz. On the other hand, CPU-based contouring of SCADA data, which is sampled every two to three seconds, can require a significant reduction in both influence radius and resolution to keep up with the faster data rate. Moving forward, the realtime visualization of PMU data, which arrives at rates as high as 30 Hz, is impractical for CPU-based contouring and is a clear case where GPU-based contouring should be used instead.

By upgrading the video card, improvements on the 0.102 second rendering time can be attained. For instance, using a GeForce 8800 GT card, the average rendering time drops to 0.033 seconds which corresponds to a refresh rate of 30 Hz. Therefore, if this video card is used, contours can be updated in real-time based on PMU data coming in at 30 samples per second. One of the other advanced metering devices deployed on the grid, FNET sensors, return data to the central server at 10 samples per second. In addition, contouring has been used by the research team developing FNET in order to interpret the raw frequency data [91]. Although the videos hosted at [91] were created using offline processing tools in MATLAB, real-time contouring of the frequency data could easily be handled by either the 7600 or 8800 GT hardware.

One final application investigated is the visualization of angle difference information at the PMU-monitored buses. The data visualized are the $\Delta\theta_{candidate}$ signals defined in (3.1), determined for each bus using an N_{trans} value corresponding to 60 seconds. This is a much bigger value of N_{trans} than is used for event detection, and the reason a longer N_{trans} is used is that it keeps any changes in the angles visible to the operator for an extended period of time as illustrated by the $N_{trans}^{(3)}$ case in Figure 3.3. Showing these angle contours to an operator would be a useful complement to the event detection work, in that an operator, upon noticing a substantial shift in phasor angles, could then query the event detection algorithms to help determine what causes these angle changes. Figure 6.9 show how this visualization would look during the TVA line outage analyzed in section 4.1.2. The color mapping used is shown in Figure 6.10. The change in the contour before and after the event highlights the change in angles on the system and would draw the attention of system operators for further investigation. Also, the use of an N_{trans} value of 60 seconds lets the change stay visible for 60 seconds after

the event occurs; this is why the same contour is shown at 49 seconds and 103 seconds. The deadband between -0.75 and 0.75 degrees in the colormap corresponds to having an angle threshold of 0.75 degrees in event detection, since the angle change must exceed 0.75 degrees before being highlighted.

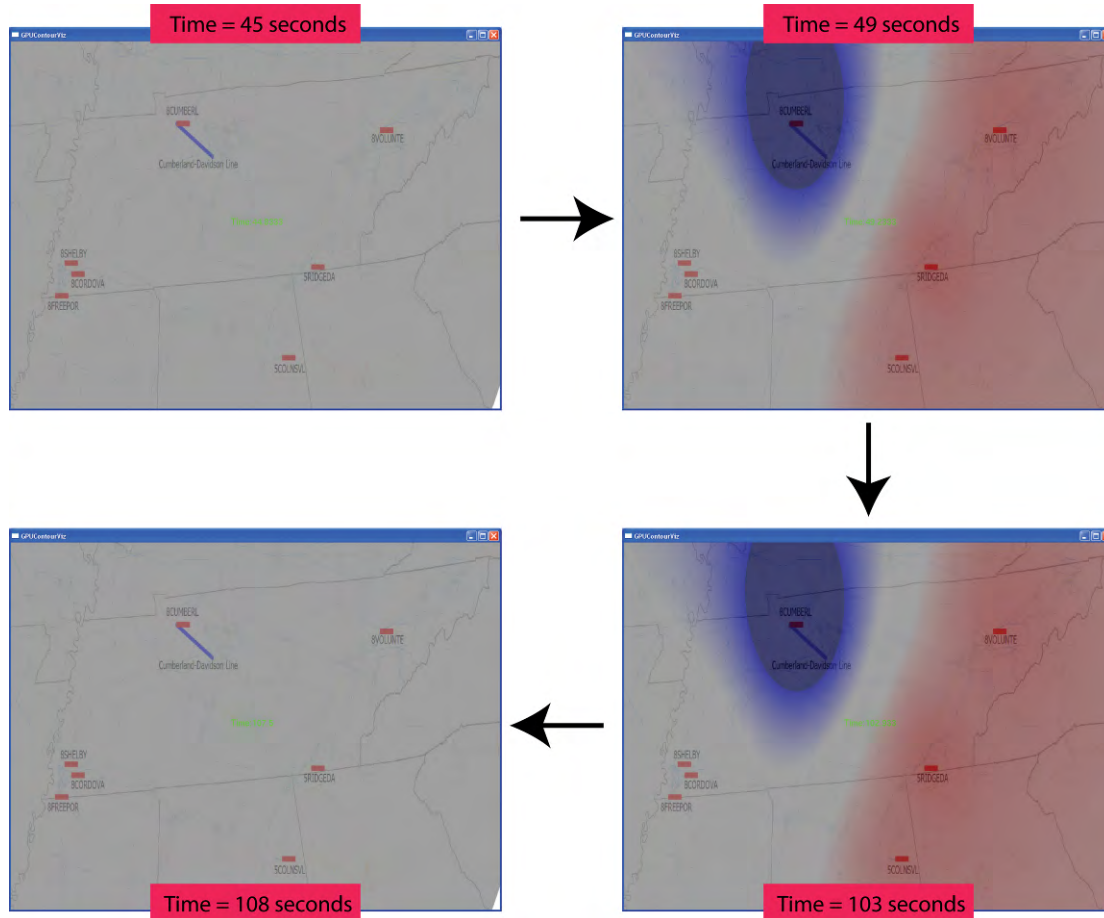


Figure 6.9: Contour of angle changes on the TVA system before and after the line outage

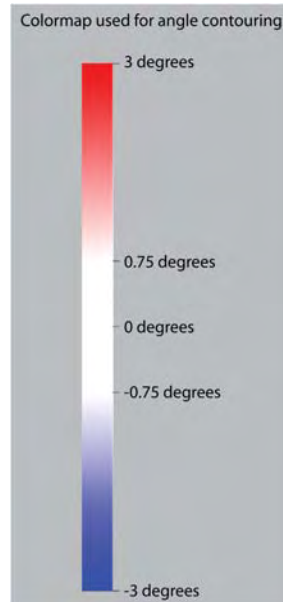


Figure 6.10: Colormap used for angle contours in Figure 6.9

Alternative methods of processing the incoming data could be used, or alternative data such as voltage or frequency could be visualized without the extensive processing used to obtain useful angle signals. Regardless, using GPUs provides a great deal more flexibility in terms of PMU data visualization, and this technique should significantly improve the potential of having real-time data visualization within control centers.

7 Conclusions and Future Work

Chapter 2 deals with the processing of phasor measurement data in order to detect system events. The results presented for FIR and median filtering highlight the key differences in these two filtering methods—FIR filtering provides more consistent attenuation of noise and unwanted frequency components, whereas median filtering preserves step changes in the system angles. When FIR filtering is used in event detection, the main downside is the increased delay due to the corruption of the step change and the need to capture a longer transition region. Also, as seen with the filtering of the generator outage data, the choice of the cutoff frequency can have a significant impact on the performance of FIR filtering. On the other hand, with median filtering, performance is difficult to predict due to the widely varying response depending on filter length. Also, the median filtering results based on the real data obtained from TVA shows that median filtering is less effective at removing noise from the angle measurement signals. Ultimately, the choice of using FIR or median filtering boils down to a tradeoff between minimizing delay in the detection of the event (for which median filtering is superior) and minimizing errors in the determination of $\Delta\theta_{\text{observed}}$ (for which FIR filtering is superior). There are several useful avenues of research for future work in the area of PMU data filtering. One area of research, mentioned briefly in the main text, is to determine whether or not hybrid FIR-median filters, which have been shown to work in image processing applications, have similar benefits when applied to PMU measurements. In addition, there are entire classes of filtering methods which have not been considered, including infinite impulse response (IIR) filtering and other types of nonlinear filters.

Chapter 3 discusses three algorithms which can be used to detect three different types of events—single line outages, double line outages, and generator outages. The way single and double line outages are modeled is typical when using the dc power flow approximations, and it is unlikely that any improvements can be made to the line outage modeling without adding additional state information. The usage of geographically based filtering of the double outage set is shown to significantly reduce the search space for double line outage detection. Future work examining whether or not the filtered event set captures real power system events would be useful. The generator outage modeling makes some significant assumptions in order to predict the post-event angles on the system, including frequency uniformity across the grid. This assumption, which results in accurate droop-based participation factors, leads to a significant delay between event occurrence and detection. One possible area of future research would be to look at different ways to model the changes in the system due to generator outages in order to detect and classify these outages more rapidly.

The fourth chapter examines the performance of the three algorithms from chapter 3 in classifying events. All of the results show that in the majority of the events the algorithms can properly classify events based on the observed angle changes due to the events. Reducing the number of PMUs on the system is shown with both single and double line outages to result in a

significant increase in the number of events which cannot be differentiated. This is one justification for increasing the PMU deployment on the power system, which currently has a very sparse deployment of these advanced sensors. In addition, the failure of the dc power flow assumptions for several of the single and double line outages due to high R/X ratios is clearly seen, showing that the algorithms are ultimately dependent on how well the real system meets the dc power flow assumptions. For the generator outage studies, the algorithm performs well in ranking each generator outage, although lag between event occurrence and classification should be reduced in order to improve situational awareness. One way to extend this research is to obtain and test the algorithms with more real-world data and larger test systems.

The PMU placement problem is discussed in chapter 5, beginning with definition of several possible objective functions and moving on to description of two means of optimizing based on these objectives—exhaustive search and simulated annealing. The overwhelming computational burden of exhaustive search and the shortcomings of using an approximate optimizer like simulated annealing are both discussed. Results from the simulated annealing and exhaustive search optimizations for the placement of three and six PMUs show that simulated annealing does provide useful results, although it is suboptimal. Testing other methods of large scale combinatorial optimization such as genetic algorithms would be useful, particularly if the results are compared to simulated annealing and exhaustive search. The results obtained by using simulated annealing for the placement of 9, 18, and 27 PMUs show that deploying more PMUs will definitely increase the performance of the event detection algorithms and also show that simulated annealing scales much better than exhaustive search.

Finally, chapter 6 presents a new method of contouring power system data which leverages the abundant computational power of modern graphics cards. The results indicate that moving contouring from the CPU to the GPU can result in significant improvements to usability and provide opportunities for development of applications which render newer measurement data in real-time. In continuing this work, GPU-based visualization techniques from other domains such as medical imagery and scientific visualization can also be evaluated to see whether or not they provide a better understanding of the system conditions to operators. Formal evaluations of how GPU-based techniques impact user performance and satisfaction would also be useful to obtain.

As new measurement devices are developed and deployed on the grid, applications to improve situational awareness should be one of the top priorities due to its importance in power system operations. This report, which focuses on both the processing and presentation of the new data available from PMUs, should help in improving the awareness of grid conditions and, by doing so, improve the reliability of the power grid.

References

- [1] "Final Report on the August 14, 2003 Blackout in the United States and Canada," US-Canada Power System Outage Task Force, April 2004.
- [2] "Steps to establish a real-time transmission monitoring system for transmission owners and operators within the Eastern and Western Interconnections," U.S. Department of Energy and Federal Energy Regulatory Commission, February 2006.
- [3] A. G. Phadke, "Synchronized phasor measurements in power systems," *Computer Applications in Power, IEEE*, vol. 6, no. 2, pp. 10-15, 1993.
- [4] M. Donnelly, M. Ingram, and J. R. Carroll, "Eastern Interconnection Phasor Project," in *System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on*, 2006, pp. 245a-245a.
- [5] J. Weiqing, V. Vittal, and G. T. Heydt, "A Distributed State Estimator Utilizing Synchronized Phasor Measurements," *Power Systems, IEEE Transactions on*, vol. 22, no. 2, pp. 563-571, 2007.
- [6] M. Zhou, V. A. Centeno, J. S. Thorp, and A. G. Phadke, "An Alternative for Including Phasor Measurements in State Estimators," *Power Systems, IEEE Transactions on*, vol. 21, no. 4, pp. 1930-1937, 2006.
- [7] J. S. Thorp, A. G. Phadke, and K. J. Karimi, "Real Time Voltage-Phasor Measurement For Static State Estimation," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-104, no. 11, pp. 3098-3106, 1985.
- [8] A. P. S. Meliopoulos, et al., "PMU data characterization and application to stability monitoring," in *Power Engineering Society General Meeting, 2006. IEEE*, 2006, pp. 8 pp.
- [9] K. Sun, S. Likhate, V. Vittal, V. S. Kolluri, and S. Mandal, "An Online Dynamic Security Assessment Scheme Using Phasor Measurements and Decision Trees," *Power Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 1935-1943, 2007.
- [10] A. R. Khatib, R. F. Nuqui, M. R. Ingram, and A. G. Phadke, "Real-time estimation of security from voltage collapse using synchronized phasor measurements," in *Power Engineering Society General Meeting, 2004. IEEE*, 2004, pp. 582-588 Vol.1.
- [11] G. Zhang, P. Hirsch, and S. Lee, "Wide Area Frequency Visualization using Smart Client Technology," in *Power Engineering Society General Meeting, 2007. IEEE*, 2007, pp. 1-8.
- [12] R. Klump, R. E. Wilson, and K. E. Martin, "Visualizing Real-Time Security Threats Using Hybrid SCADA / PMU Measurement Displays," in *System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on*, 2005, pp. 55c-55c.

- [13] S. Chun-Lien and J. Bo-Yuan, "Visualization of Large-Scale Power System Operations Using Phasor Measurements," in *Power System Technology, 2006. PowerCon 2006. International Conference on*, 2006, pp. 1-6.
- [14] A. Monticelli, "Modeling circuit breakers in weighted least squares state estimation," *Power Systems, IEEE Transactions on*, vol. 8, no. 3, pp. 1143-1149, 1993.
- [15] "Final Report on the Implementation of the Task Force Recommendations," US-Canada Power System Outage Task Force, September 2006.
- [16] A. G. Phadke, J. S. Thorp, and M. G. Adamiak, "A New Measurement Technique for Tracking Voltage Phasors, Local System Frequency, and Rate of Change of Frequency," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-102, no. 5, pp. 1025-1038, 1983.
- [17] A. G. Phadke, "Synchronized phasor measurements-a historical overview," in *Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES*, 2002, pp. 476-479 vol.1.
- [18] T. L. Baldwin, L. Mili, M. B. Boisen, Jr., and R. Adapa, "Power system observability with minimal phasor measurement placement," *Power Systems, IEEE Transactions on*, vol. 8, no. 2, pp. 707-715, 1993.
- [19] S. E. Widergren, Z. Huang, and J. E. Dagle, "Electric System-wide Measurements: North American Directions," in *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, 2007, pp. 120-120.
- [20] R. F. Nuqui and A. G. Phadke, "Phasor measurement unit placement techniques for complete and incomplete observability," *Power Delivery, IEEE Transactions on*, vol. 20, no. 4, pp. 2381-2388, 2005.
- [21] B. Fardanesh, S. Zelingher, A. P. Sakis Meliopoulos, G. Cokkinides, and J. Ingleson, "Multifunctional synchronized measurement network [power systems]," *Computer Applications in Power, IEEE*, vol. 11, no. 1, pp. 26-30, 1998.
- [22] A. R. Messina, V. Vittal, D. Ruiz-Vega, and G. Enriquez-Harper, "Interpretation and Visualization of Wide-Area PMU Measurements Using Hilbert Analysis," *Power Systems, IEEE Transactions on*, vol. 21, no. 4, pp. 1763-1771, 2006.
- [23] S. M. Brahma and A. A. Girgis, "Fault location on a transmission line using synchronized Voltage measurements," *Power Delivery, IEEE Transactions on*, vol. 19, no. 4, pp. 1619-1622, 2004.
- [24] R. O. Burnett, Jr., et al., "Synchronized phasor measurements of a power system event," *Power Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 1643-1650, 1994.
- [25] J. F. Hauer, "Validation of phasor calculations in the macrodyne PMU for California-Oregon Transmission Project tests of March 1993," *Power Delivery, IEEE Transactions on*, vol. 11, no. 3, pp. 1224-1231, 1996.

- [26] J. F. Hauer, N. B. Bhatt, K. Shah, and S. Kolluri, "Performance of "WAMS East" in providing dynamic information for the North East blackout of August 14, 2003," in *Power Engineering Society General Meeting, 2004. IEEE*, 2004, pp. 1685-1690 Vol.2.
- [27] "IEEE Standard for Synchrophasors for Power Systems," *IEEE Std C37.118-2005 (Revision of IEEE Std 1344-1995)*, 2006.
- [28] "Reliability Coordinator Reference Document," *March 25, 2004*: North American Electric Reliability Corporation, 2004.
- [29] L. S. Davis, "A Survey of Edge Detection Techniques," *Computer Graphics Image Processing*, vol. 4, no. 3, pp. 248-270, 1975.
- [30] D. Forsyth and J. Ponce, in *Computer Vision: A Modern Approach*. New Jersey: Prentice Hall, 2003, pp. 176-179.
- [31] A. C. Bovik and J. D. C. Munson, "Edge detection using median comparisons," *Computer Vision, Graphics, and Image Processing*, vol. 33, no. 3, pp. 377-389, March 1986 1986.
- [32] W. C. Karl, S. B. Leeb, L. A. Jones, J. L. Kirtley, and G. C. Verghese, "Applications of rank-based median filters in power electronics," *Power Electronics, IEEE Transactions on*, vol. 7, no. 3, pp. 437-443, 1992.
- [33] M. Kezunovic, I. Rikalo, and D. J. Sobajic, "Real-time and off-line transmission line fault classification using neural networks," *International Journal of Engineering Intelligent Systems*, vol. 4, no. 1, pp. 57-63, March 1996 1996.
- [34] C. J. Kim and B. d. Russell, "Classification of faults and switching events by inductive reasoning and expert system methodology," *Power Delivery, IEEE Transactions on*, vol. 4, no. 3, pp. 1631-1637, 1989.
- [35] W. R. Anis Ibrahim and M. M. Morcos, "Artificial intelligence and advanced mathematical tools for power quality applications: a survey," *Power Delivery, IEEE Transactions on*, vol. 17, no. 2, pp. 668-673, 2002.
- [36] H. Singh and F. L. Alvarado, "Network topology determination using least absolute value state estimation," *Power Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 1159-1165, 1995.
- [37] F. L. Alvarado, "Determination of External System Topology Errors," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. 11, pp. 4553-4561, 1981.
- [38] F. Schlaepfer, T. C. Kelly, and A. G. Dewey, "An Interactive Load Flow Program," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-91, no. 1, pp. 78-84, 1972.
- [39] G. Pires de Azevedo, C. Sieckenius de Souza, and B. Feijo, "Enhancing the human-computer interface of power system applications," *Power Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 646-653, 1996.
- [40] J. D. Weber and T. J. Overbye, "Voltage contours for power system visualization," *Power Systems, IEEE Transactions on*, vol. 15, no. 1, pp. 404-409, 2000.

- [41] Y. Sun and T. J. Overbye, "Visualizations for power system contingency analysis data," *Power Systems, IEEE Transactions on*, vol. 19, no. 4, pp. 1859-1866, 2004.
- [42] T. J. Overbye, D. A. Wiegmann, A. M. Rich, and Y. Sun, "Human factors aspects of power system voltage contour visualizations," *Power Systems, IEEE Transactions on*, vol. 18, no. 1, pp. 76-82, 2003.
- [43] D. A. Wiegmann, G. R. Essenberg, T. J. Overbye, and Y. Sun, "Human Factor Aspects of Power System Flow Animation," *Power Systems, IEEE Transactions on*, vol. 20, no. 3, pp. 1233-1240, 2005.
- [44] D. Savageau and T. J. Overbye, "Adaptive Influence Distance Algorithm for Contouring Bus-Based Power System Data," in *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, 2007, pp. 117-117.
- [45] D. Geer, "Taking the graphics processor beyond graphics," *Computer*, vol. 38, no. 9, pp. 14-16, 2005.
- [46] M. Pharr, *GPU Gems 2*: Addison-Wesley, 2005.
- [47] H. Nguyen, *GPU Gems 3*: Addison-Wesley, 2007.
- [48] S. W. Park, L. Linsen, O. Kreylos, J. D. Owens, and B. Hamann, "A Framework for Real-time Volume Visualization of Streaming Scattered Data," in *10th International Fall Workshop on Vision, Modeling, and Visualization 2005*, 2005, pp. 225-232.
- [49] J. Y. C. Chen and J. E. Thropp, "Review of Low Frame Rate Effects on Human Performance," *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, vol. 37, no. 6, pp. 1063-1076, 2007.
- [50] B. Stott, "Review of load-flow calculation methods," *Proceedings of the IEEE*, vol. 62, no. 7, pp. 916-929, 1974.
- [51] Y. Ota, T. Hashiguchi, H. Ukai, M. Sonoda, Y. Miwa, and A. Takeuchi, "Monitoring of Interconnected Power System Parameters using PMU based WAMS," in *Power Tech 2007 Conference*. Lausanne, Switzerland, 2007, pp. 5.
- [52] Y. Jun-Zhe, L. Chih-Wen, and W. Wen-Giang, "A Hybrid Method for the Estimation of Power System Low-Frequency Oscillation Parameters," *Power Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 2115-2123, 2007.
- [53] K. Mei, S. M. Rovnyak, and O. Chee-Mun, "Dynamic event detection using wavelet analysis," in *Power Engineering Society General Meeting, 2006. IEEE*, 2006, pp. 7.
- [54] I. Kamwa, J. Beland, and D. McNabb, "PMU-Based Vulnerability Assessment Using Wide- Area Severity Indices and Tracking Modal Analysis," in *Power Systems Conference and Exposition, 2006. PSCE '06. 2006 IEEE PES*, 2006, pp. 139-149.
- [55] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing, Second Edition*. Upper Saddle River, New Jersey: Prentice Hall, 1999.
- [56] M. Klein, G. J. Rogers, and P. Kundur, "A fundamental study of inter-area oscillations in power systems," *Power Systems, IEEE Transactions on*, vol. 6, no. 3, pp. 914-921, 1991.

- [57] P. Kundur and W. Lei, "Small signal stability analysis: experiences, achievements, and challenges," in *Power System Technology, 2002. Proceedings. PowerCon 2002. International Conference on*, 2002, pp. 6-12 vol.1.
- [58] R. L. Cresap and J. F. Hauer, "Emergence of a New Swing Mode in the Western Power System," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. 4, pp. 2037-2045, 1981.
- [59] N. Sankarayya, K. Roy, and D. Bhattacharya, "Optimizing computations in a transposed direct form realization of Floating-Point LTI-FIR systems," in *Computer-Aided Design, 1997. Digest of Technical Papers., 1997 IEEE/ACM International Conference on*, 1997, pp. 120-125.
- [60] J. K. Wang, R. M. Gardner, and L. Yilu, "Analysis of system oscillations using wide-area measurements," in *Power Engineering Society General Meeting, 2006. IEEE*, 2006, pp. 6 pp.
- [61] E. Chen, H. S. Timorabadi, and F. P. Dawson, "Real-time phasor measurement method including a GPS common time-stamp for distributed power system monitoring and control," in *Electrical and Computer Engineering, 2005. Canadian Conference on*, 2005, pp. 441-444.
- [62] M. Juhola, J. Katajainen, and T. Raita, "Comparison of algorithms for standard median filtering," *Signal Processing, IEEE Transactions on*, vol. 39, no. 1, pp. 204-208, 1991.
- [63] Z. Huang, T. Faris, K. Martin, J. Hauer, C. Bonebrake, and J. Shaw, "Laboratory Performance Evaluation Report of SEL 421 Phasor Measurement Unit," Pacific Northwest National Laboratory, November 2007.
- [64] J. Astola and P. Kuosmann, *Fundamentals of Nonlinear Digital Filtering*. Boca Raton: CRC Press, 1997.
- [65] A. Wood and B. Wollenberg, *Power Generation, Operation, and Control*. New York: Wiley, 1984.
- [66] H. Anton, *Elementary Linear Algebra, 8th Edition*. New York: Wiley, 2000.
- [67] M. Crow, *Computational Methods for Electric Power Systems*. Boca Raton: CRC Press, 2003.
- [68] P. Kundur, *Power System Stability and Control*. New York: McGraw-Hill, 1994.
- [69] "Dynamic Models for Steam and Hydro Turbines in Power System Studies," IEEE Task Force on Overall Plant Response 1973.
- [70] T. Guler and G. Gross, "Detection of Island Formation and Identification of Causal Factors Under Multiple Line Outages," *Power Systems, IEEE Transactions on*, vol. 22, no. 2, pp. 505-513, 2007.
- [71] B. L. Silverstein and D. M. Porter, "Contingency ranking for bulk system reliability criteria," *Power Systems, IEEE Transactions on*, vol. 7, no. 3, pp. 956-964, 1992.
- [72] N. Megiddo, "Linear-Time Algorithms for Linear Programming in $\mathbb{R}^3$ and Related Problems," *SIAM Journal on Computing*, vol. 12, no. 4, pp. 759-776, 1983.

- [73] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671-680, May 13, 1983 1983.
- [74] K. A. De Jong, "Analysis of the behavior of a class of genetic adaptive systems," University of Michigan, August 1975, available at <http://hdl.handle.net/2027.42/4507>.
- [75] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, 1995, pp. 1942-1948 vol.4.
- [76] K. P. Wong and Y. W. Wong, "Combined genetic algorithm/simulated annealing/fuzzy set approach to short-term generation scheduling with take-or-pay fuel contract," *Power Systems, IEEE Transactions on*, vol. 11, no. 1, pp. 128-136, 1996.
- [77] I. Amidror, "Scattered data interpolation methods for electronic imaging systems: a survey," *Journal of Electronic Imaging*, vol. 11, no. 2, pp. 157-176, 2002.
- [78] W. A. Nuss and D. W. Titley, "Use of Multiquadric Interpolation for Meteorological Objective Analysis," *Monthly Weather Review*, vol. 122, no. 7, pp. 1611-1631, July 1994 1994.
- [79] T. M. Lehmann, C. Gonner, and K. Spitzer, "Survey: interpolation methods in medical image processing," *Medical Imaging, IEEE Transactions on*, vol. 18, no. 11, pp. 1049-1075, 1999.
- [80] R. J. Renka, "Multivariate interpolation of large sets of scattered data," *ACM Trans. Math. Softw.*, vol. 14, no. 2, pp. 139-148, 1988.
- [81] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the 1968 23rd ACM national conference*: ACM, 1968.
- [82] R. E. Barnhill, "Representation and approximation of surfaces," in *Mathematical Software III*, J. R. Rice, Ed. New York: Academic Press, 1977, pp. 517-523.
- [83] E. Kilgariff and R. Fernando, "The GeForce 6 Series GPU Architecture," in *GPU Gems 2*, M. Pharr, Ed. Upper Saddle River, NJ: Addison-Wesley, 2005, pp. pp. 471-491.
- [84] S. Green, "The OpenGL Framebuffer Object Extension," in *Game Developers Conference 2005*, 2005.
- [85] M. Kilgard, *The OpenGL Utility Toolkit (GLUT) Programming Interface API Version 3*. [Online]. Available: <http://www.opengl.org/documentation/specs/glut/spec3/spec3.html>.
- [86] W. R. Mark, R. S. Glanville, K. Akeley, and M. J. Kilgard, "Cg: a system for programming graphics hardware in a C-like language," in *ACM SIGGRAPH 2003 Papers*. San Diego, California: ACM, 2003.
- [87] J. D. Meier, S. Vasireddy, A. Babbar, and A. Mackman, "How To: Time Managed Code Using QueryPerformanceCounter and QueryPerformanceFrequency," Microsoft Developer Network, May 2004, available at <http://msdn2.microsoft.com/en-us/library/ms979201.aspx>.
- [88] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human-Computer Interaction*, 3rd ed. New York: Pearson, 2004.

- [89] J. D. McCarthy, M. A. Sasse, and D. Miras, "Sharp or smooth?: comparing the effects of quantization vs. frame rate for streamed video," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. Vienna, Austria: ACM, 2004.
- [90] G. M. Wilson, "Psychophysiological indicators of the impact of media quality on users," in *CHI '01 extended abstracts on Human factors in computing systems*. Seattle, Washington: ACM, 2001.
- [91] "Frequency wave propagation," vol. 2008: Virginia Tech Power IT Lab, 2008.
- [92] J. D. Glover, M. S. Sarma, and T. Overbye, *Power System Analysis and Design, 4th Edition*, 4 ed: Thomson Engineering, 2008.
- [93] F. P. de Mello and L. H. Hannett, "Validation of Synchronous Machine Models and Derivation of Model Parameters from Tests," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. 2, pp. 662-672, 1981.
- [94] "Computer Representation of Excitation Systems," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-87, no. 6, pp. 1460-1464, 1968.

Appendix A 37 Bus System Description

The basis for this system is the 37-bus system from Example 13.9 in [92]. The machine and exciter models were changed to the GENROU [93] and IEEE1 [94] models to provide a more detailed model over the original case, and the parameters for these models were taken from similarly sized units in the Eastern Interconnect. In addition, an IEEEG1 governor [69] was added to each machine using parameters from similarly sized units in the Eastern Interconnect. A full description of the system is provided in the following tables.

Table A.1: Bus information

Number	Name	Nom kV	PU Volt	Angle (Deg)	Number	Name	Nom kV	PU Volt	Angle (Deg)
1	TIM345	345	1.02621	-0.75766	30	BUCKY138	138	1.01008	-1.820895
3	MORO138	138	1.00773	-3.2384245	31	SLACK345	345	1.03	0
5	HOMER69	69	1.00817	-5.252666	32	SAVOY138	138	1.01242	-1.5171459
10	RAY69	69	1.02293	-4.2961769	33	SAVOY69	69	1.01303	-2.7431788
12	TIM69	69	1.01852	-5.0273795	34	PATTEN69	69	1.01277	-4.7015119
13	FERNA69	69	1.00666	-5.786334	35	SLACK138	138	1.02684	-0.8142443
14	WEBER69	69	1.02	-4.3555994	37	AMANDA69	69	1.00408	-6.945878
15	UIUC69	69	1.00707	-6.1080384	38	RAY345	345	1.02706	-0.7578956
16	PETE69	69	1.00424	-6.6900287	39	RAY138	138	1.02132	-2.5541735
17	PAI69	69	1.01186	-5.9505262	40	TIM138	138	1.01074	-3.6131575
18	HANNAH69	69	1.00546	-6.904346	41	LAUF138	138	1.00651	-2.2658136
19	GROSS69	69	1.01225	-5.7058263	44	LAUF69	69	1.02	-2.9113064
20	SHIMKO69	69	1.01331	-4.3103828	47	BOB138	138	1.01703	-4.51683
21	WOLEN69	69	0.98743	-7.2428098	48	BOB69	69	1	-6.4654179
24	HALE69	69	1.00509	-5.7531943	50	ROGER69	69	1.02	-3.0960746
27	HISKY69	69	1.00605	-6.5725245	53	BLT138	138	1.02	-4.4752307
28	JO345	345	1.03	2.7282064	54	BLT69	69	1.01	-6.0853167
29	JO138	138	1.02513	0.7128861	55	DEMAR69	69	1.00124	-6.4195685
					56	LYNN138	138	1.02457	-0.151776

Table A.2: Generator information

Name of Bus	ID	Status	MVA Base	Gen MW	Gen Mvar	Set Volt	Min MW	Max MW	Min Mvar	Max Mvar
WEBER69	1	Closed	41	10	24.12	1.02	0	41	0	5
JO345	1	Closed	200	150	-2.85	1.03	0	200	-60	60
JO345	2	Closed	200	150	-2.85	1.03	0	200	-60	60
SLACK345	1	Closed	295	88.96	23.3	1.03	0	295	-90	120
LAUF69	1	Closed	221.4	150	-30.66	1.02	0	221.4	-20	40
BOB69	1	Closed	97.58	16	-21.21	1	0	97.58	-14	26
ROGER69	1	Closed	101.8	38	5.63	1.02	0	101.8	-18	33
BLT138	1	Closed	176.5	140	76.54	1.02	0	176.5	0	45
BLT69	1	Closed	147.06	75.23	34.75	1.01	0	147.06	-20	60

Table A.3: Line information

From Number	To Number	Circuit	Status	R	X	B	Limit (MVA)	From Number	To Number	Circuit	Status	R	X	B	Limit (MVA)
1	31	1	Closed	0.00117	0.017	0.182	597	20	50	1	Closed	0.0424	0.07509	0.0008	82
1	40	1	Closed	0.001	0.0623	0	250	21	48	1	Closed	0.01824	0.04236	0.0075	72
3	40	1	Closed	0.0048	0.0368	0.0182	233	21	48	2	Closed	0.01829	0.04246	0.0078	72
3	41	1	Closed	0.00902	0.0571	0.0135	233	24	44	1	Closed	0.03993	0.09965	0.002	82
5	18	1	Closed	0.03133	0.07675	0.0015	102	28	29	1	Closed	0.00087	0.051	0	220
5	44	1	Closed	0.03463	0.08253	0.0016	102	28	29	2	Closed	0.00087	0.051	0	220
10	13	1	Closed	0.03375	0.0789	0.0012	82	31	28	1	Closed	0.00224	0.03268	0.35	600
10	19	1	Closed	0.0405	0.0953	0.0021	72	32	29	1	Closed	0.0103	0.05681	0.0164	191
10	39	1	Closed	0.00106	0.03949	-0.0041	186.7	29	41	1	Closed	0.01868	0.1259	0.0356	191
12	17	1	Closed	0.02172	0.06935	0.0019	82	56	29	1	Closed	0.00771	0.0544	0.0138	133
12	18	1	Closed	0.02747	0.08909	0.0019	106	30	32	1	Closed	0.00225	0.0134	0.004	191
12	27	1	Closed	0.0142	0.07557	0.0695	112	30	41	1	Closed	0.00735	0.04395	0.0123	191
12	40	1	Closed	0.00107	0.04039	-0.0042	187	35	31	1	Closed	0.00087	0.051	0	220
12	40	2	Closed	0.00107	0.04039	-0.0042	187	31	38	1	Closed	0.00075	0.01092	0.117	597
13	55	1	Closed	0.02216	0.0904	0.0017	90	33	32	1	Closed	0.0025	0.0723	0	101
14	34	1	Closed	0.02625	0.06429	0.0012	72	33	50	1	Closed	0.0502	0.101	0.0025	82
14	44	1	Closed	0.04211	0.08545	0.0474	81	35	39	1	Closed	0.01028	0.07253	0.0176	288
15	16	1	Closed	0.00753	0.02582	0.0206	93	35	56	1	Closed	0.01243	0.07847	0.0192	100
15	24	1	Closed	0.01835	0.02845	0.0221	74	39	38	1	Closed	0.00094	0.05107	-0.0127	224
15	54	1	Closed	0.00846	0.00465	0.021	77	39	38	2	Closed	0.00095	0.05116	-0.0124	224
15	54	2	Closed	0.00855	0.0047	0.0213	77	39	40	1	Closed	0.01161	0.08085	0.0225	233
15	54	3	Closed	0.00859	0.00472	0.0214	77	39	47	1	Closed	0.00775	0.05244	0.0154	233
16	27	1	Closed	0.00366	0.01312	0.046	112	44	41	1	Closed	0.0025	0.07144	-0.0045	101
17	19	1	Closed	0.02703	0.03613	0.002	50	44	41	2	Closed	0.00244	0.06829	-0.006	101
18	37	1	Closed	0.01017	0.00559	0.0253	68	48	47	1	Closed	0.00101	0.03925	-0.0035	187
18	37	2	Closed	0.01017	0.00559	0.0253	68	47	53	1	Closed	0.00204	0.00692	0.1601	185
20	34	1	Closed	0.02423	0.05862	0.0011	72	48	54	1	Closed	0.01473	0.036	0.0744	105
20	48	1	Open	0.02133	0.0521	0.0009	82	54	53	1	Closed	0.00134	0.04988	-0.0004	187
								54	55	1	Closed	0.06246	0.08246	0.0003	50

Table A.4: Load information

Number of Bus	ID	Status	MW	Mvar	Number of Bus	ID	Status	MW	Mvar
3	1	Closed	12.3	5	24	1	Closed	36.3	10.4
5	1	Closed	14	3.2	27	1	Closed	20	6
10	1	Closed	16.8	2.5	30	1	Closed	23.4	6.2
12	1	Closed	22.9	6.5	33	1	Closed	28	6
13	1	Closed	23	7	34	1	Closed	22.81	3
14	1	Closed	22.2	15.2	37	1	Closed	27	3
15	1	Closed	58.2	36.3	44	1	Closed	59.8	12.3
16	1	Closed	57.8	40.4	48	1	Closed	55.8	12.5
17	1	Closed	32.8	12.9	50	1	Closed	14.1	3
18	1	Closed	45	12	53	1	Closed	59.5	27.8
19	1	Closed	18.3	5	54	1	Closed	12.43	5.73
20	1	Closed	15.3	5	55	1	Closed	22.65	6.15
21	1	Closed	74.4	26.8	56	1	Closed	14	3.7

Table A.5: Switched shunt information

Number of Bus	ID	Reg Bus Num	Status	Control Mode	Regulates	Actual Mvar	Volt High	Volt Low	Reg Volt	Nominal Mvar	Max Mvar	Min Mvar
13	1	13	Closed	Discrete	Volt	4.86	1.025	0.998	1.0067	4.8	4.8	0
14	1	14	Closed	Discrete	Volt	7.49	1.025	0.998	1.02	7.2	7.2	0
15	1	15	Closed	Discrete	Volt	12.78	1.025	0.998	1.0071	12.6	12.6	0
16	1	16	Closed	Discrete	Volt	29.04	1.025	0.998	1.0042	28.8	28.8	0
17	1	17	Closed	Discrete	Volt	15.97	1.025	0.998	1.0119	15.6	15.6	0
18	1	18	Closed	Fixed	Volt	18.2	1.025	0.998	1.0055	18	14.4	0
20	1	20	Closed	Discrete	Volt	7.39	1.025	0.998	1.0133	7.2	7.2	0
44	1	44	Closed	Discrete	Volt	0	1.025	0.998	1.02	0	20	0

Table A.6: Transformer control information

From Number	To Number	Circuit	Type	Status	Tap Ratio	Phase (Deg)	XF Auto	Reg Bus	Reg Value	Reg Error	Reg Min	Reg Max	Tap Min	Tap Max	Step Size
1	40	1	LTC	Closed	1	0	No	1	1.02621	0.02621	0.99	1	0.9	1.1	0.00625
10	39	1	LTC	Closed	1.00625	0	Yes	10	1.02293	0	1.015	1.025	0.9	1.1	0.00625
12	40	1	LTC	Closed	1.0125	0	Yes	12	1.01852	0	1.015	1.025	0.9	1.1	0.00625
12	40	2	LTC	Closed	1.0125	0	Yes	12	1.01852	0	1.015	1.025	0.9	1.1	0.00625
28	29	1	LTC	Closed	1	0	No	28	1.03	0.03	0.99	1	0.9	1.1	0.00625
28	29	2	LTC	Closed	1	0	No	28	1.03	0.03	0.99	1	0.9	1.1	0.00625
35	31	1	LTC	Closed	1	0	No	35	1.02684	0.02684	0.99	1	0.9	1.1	0.00625
33	32	1	LTC	Closed	1	0	No	33	1.01303	0.01303	0.99	1	0.9	1.1	0.00625
39	38	1	Fixed	Closed	1	0	No	0	0	-0.51	0.51	1.5	0.51	1.5	0.00625
39	38	2	Fixed	Closed	1	0	No	0	0	-0.51	0.51	1.5	0.51	1.5	0.00625
44	41	1	LTC	Closed	1.02125	0	No	44	1.02	0	1	1.02	0.99	1.1	0.00625
44	41	2	LTC	Closed	1.02125	0	No	44	1.02	0	1	1.02	0.99	1.1	0.00625
48	47	1	LTC	Closed	1	0	No	48	1	-0.015	1.015	1.025	0.9	1.1	0.00625
54	53	1	LTC	Closed	1	0	No	0	0	-1	1	1.02	0.99	1.1	0.00625

Table A.7: Line shunt information

From Number	To Number	Circuit	Bus Number Located At	ID	G	B	Status
12	40	1	40	1	0.0006	0	Closed
12	40	2	40	1	0.0006	0	Closed
39	38	1	38	1	0.0014	0	Closed
39	38	2	38	1	0.0014	0	Closed
44	41	1	41	1	0.0007	0	Closed
44	41	2	41	1	0.0007	0	Closed
48	47	1	47	1	0.0006	0	Closed
54	53	1	53	1	0.0004	0	Closed

Table A.8: Machine model information (all non-specified parameters are zero)

Bus Name	Gen ID	Type	MVA Base	Device Status	H	Xd	Xq	Xdp	Xqp	Xdpp	Xl	Tdop	Tqop	Tdopp	Tqopp	S(1.0)	S(1.2)
WEBER69	1	GENROU	41	Active	2.35	1.77	1.7	0.255	0.454	0.2	0.14	4	0.514	0.032	0.062	0.083	0.39
JO345	1	GENROU	200	Active	3.021	1.695	1.616	0.239	0.389	0.1795	0.1	5.3	0.49	0.034	0.081	0.108	0.455
JO345	2	GENROU	200	Active	3.021	1.695	1.616	0.239	0.389	0.1795	0.1	5.3	0.49	0.034	0.081	0.108	0.455
SLACK345	1	GENROU	295	Active	5.93	1.65	1.6	0.216	0.895	0.12	0.1	7.6	1.5	0.055	0.265	0.13	0.456
LAUF69	1	GENROU	221.4	Active	2.08	2.1	2.02	0.295	0.505	0.225	0.16	4	0.55	0.032	0.06	0.094	0.421
BOB69	1	GENROU	97.58	Active	4.321	1.37	1.32	0.175	0.29	0.135	0.075	5.6	0.54	0.036	0.083	0.1214	0.5897
ROGER69	1	GENROU	101.8	Active	4.51	1.22	1.18	0.126	0.22	0.085	0.08	5.6	0.48	0.032	0.078	0.1	0.48
BLT138	1	GENROU	176.5	Active	2.27	1.69	1.61	0.23	0.4225	0.17	0.0845	9.6	0.3	0.05	0.05	0.0963	0.555
BLT69	1	GENROU	147.059	Active	2.7779	1.7604	1.7503	0.383	0.7689	0.3076	0.2322	4.54	0.5	0.05	0.06	0.11	0.48

Table A.9: Exciter model information

Bus Name	Gen ID	Type	MVA Base	Device Status	Tr	Ka	Ta	Vrmax	Vrmin	Ke	Te	Kf	Tf	Switch	E1	SE(E1)	E2	SE(E2)
WEBER69	1	IEEE1	41	Active	0	24.86	0.02	1	-1	0	0.588	0.024	1	0	2.1	0.08	2.81	0.314
JO345	1	IEEE1	200	Active	0.06	25	0.2	1	-1	0	0.5	0.0805	0.35	0	3.375	0.0684	4.5	0.2667
JO345	2	IEEE1	200	Active	0.06	25	0.2	1	-1	0	0.5	0.0805	0.35	0	3.375	0.0684	4.5	0.2667
SLACK345	1	IEEE1	295	Active	0	40	0.15	1	-1.5	0	0.5	0.08	1	0	2.5	0.1	3	0.4
LAUF69	1	IEEE1	221.4	Active	0	50	0.02	1	-1	-0.057	0.541	0.086	1.02	0	3.08	0.038	4.11	0.3
BOB69	1	IEEE1	97.58	Active	0	400	0.02	7.3	-6.6	1	0.8	0.03	1	0	3	0.5	4	0.86
ROGER69	1	IEEE1	101.8	Active	0.06	25	0.2	1	-1	-0.045	0.05	0.07	0.35	0	2.25	0.066	3	0.267
BLT138	1	IEEE1	176.5	Active	0.06	25	0.2	1	-1	0	0.5	0.1	1.351	0	2.98	0.077	3.968	0.302
BLT69	1	IEEE1	147.059	Active	0.06	40	0.1	1	-1	0	0.67	0.11	1	0	3	0.09	4	0.368

Table A.10: Governor model information (all non-specified parameters are zero)

Bus Name	Gen ID	Type	MVA Base	Device Status	K	T1	T2	T3	Uo	Uc	Pmax	Pmin	T4	K1	K2	T5	K3	K4	T6	K5
WEBER69	1	IEEEG1	41.0000	Active	20	0	0	0.05	999	-999	1	0	0.065	0	0	0.036	1	0	0	0
JO345	1	IEEEG1	200.0000	Active	20	0.083	0.05	0.2	0.27	-0.25	1	0	0.05	0.3	0	7.85	0.25	0	0.4	0.45
JO345	2	IEEEG1	200.0000	Active	20	0.083	0.05	0.2	0.27	-0.25	1	0	0.05	0.3	0	7.85	0.25	0	0.4	0.45
SLACK345	1	IEEEG1	295.0000	Active	20	0	0	0.15	0.012	-0.012	1	0	0.275	0.224	0	0.1	0.395	0	0.3	0.381
LAUF69	1	IEEEG1	221.4000	Active	20	0	0	0.1	0.1	-1	1	0	0.25	0.3	0	7.5	0.4	0	0.4	0.3
BOB69	1	IEEEG1	97.5800	Active	20	0	0	0.1	0.1	-1	0.9	0	0.4	1	0	0	0	0	0	0
ROGER69	1	IEEEG1	101.8000	Active	20	0	0	0.1	0.1	-1	0.9	0	0.4	1	0	0	0	0	0	0
BLT138	1	IEEEG1	176.5000	Active	20	0.083	0	0.2	0.1	-0.1	1	0	0.218	0.218	0	7.77	0.782	0	0	0
BLT69	1	IEEEG1	147.0590	Active	20	0	0	0.1	0.007	-0.007	1	0	0.01	0	0	0	0	0	0	1

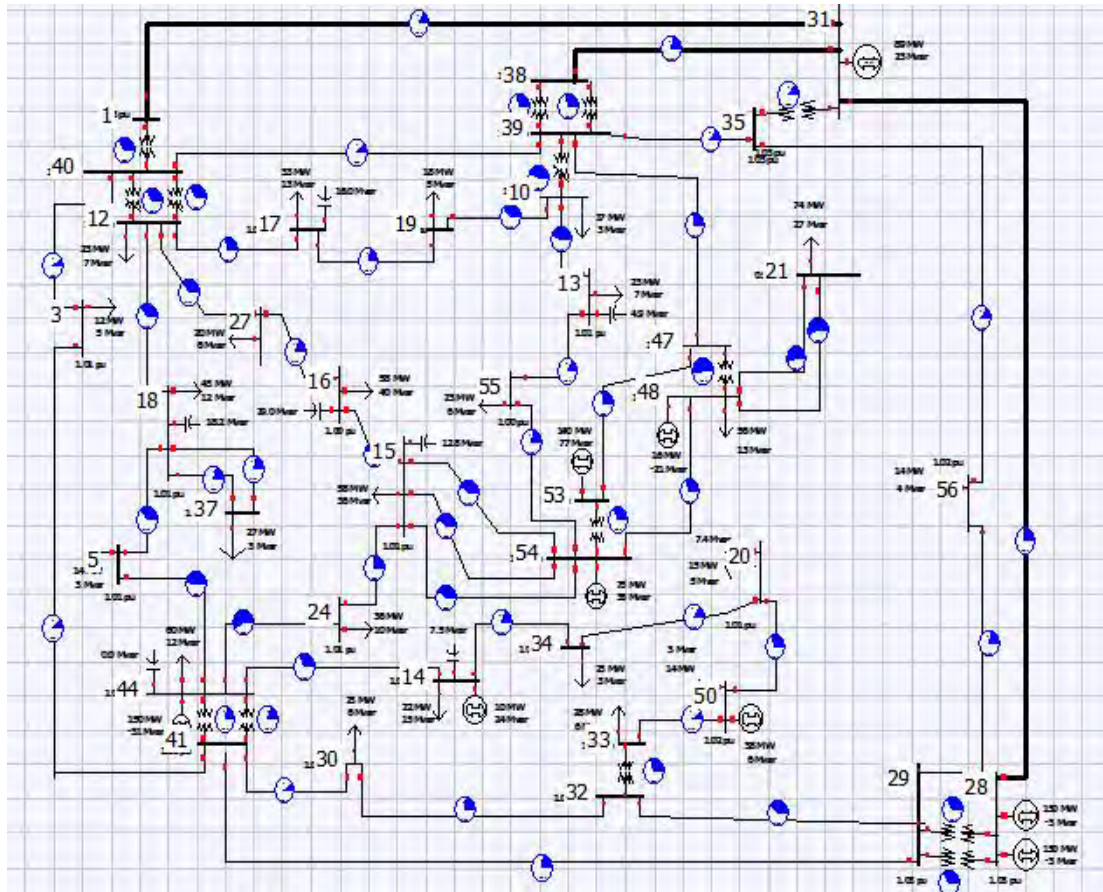


Figure A.1: Oneline diagram of the 37 bus system

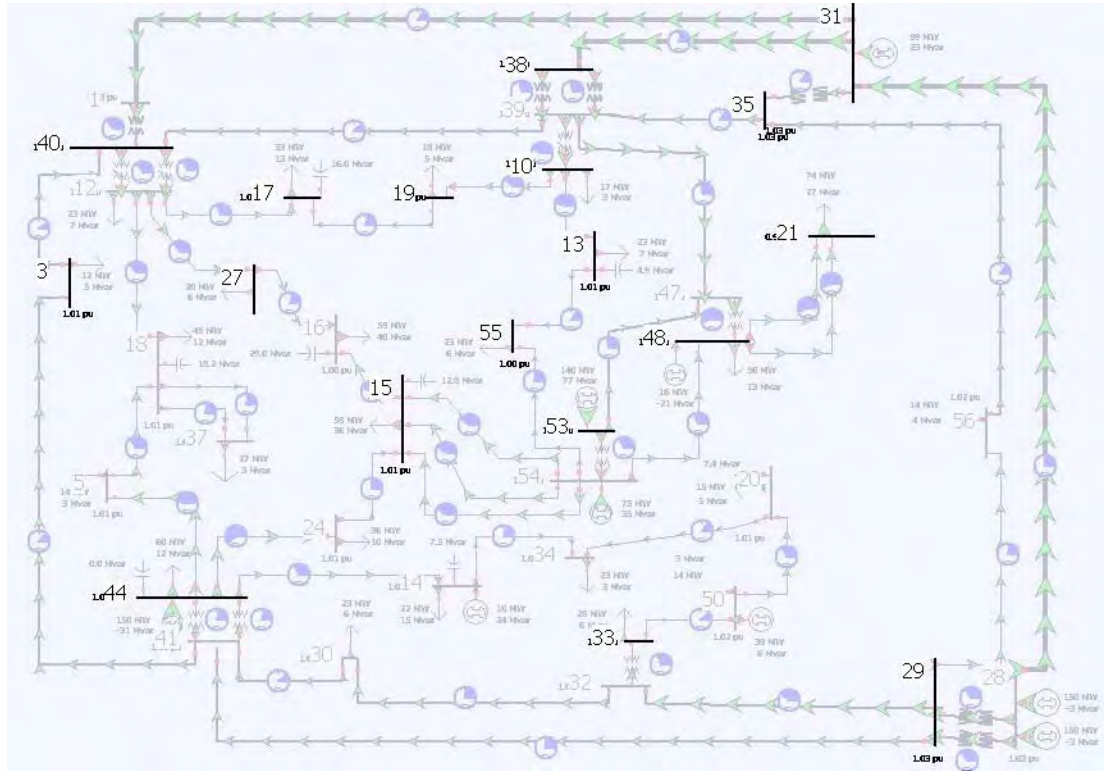


Figure A.2: Oneline diagram with 18 PMU buses as defined in section 4.1.1.2 highlighted