1-2011

IMPROVED LINE OUTAGE DETECTION USING SYNCHROPHASOR MEASUREMENTS

Nick Mahoney
Clemson University, nmahone@g.clemson.edu

IMPROVED LINE OUTAGE DETECTION USING SYNCHROPHASOR MEASUREMENTS

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Electrical Engineering

---

by
Nicholas John Mahoney
August 2011

Accepted by:
Dr. Elham Makram, Committee Chair
Dr. Richard Groff
Dr. John Gowdy

ABSTRACT

The recent advent of synchronized phasor measurements has allowed a power system to be more readily observable. In fact, when multiple buses are observed, applications that were never before possible become a reality. One such application is the detection of line outages in remote or unobserved parts of the system. Two such methods of line outage detection are examined. First, principal component analysis is used to show that highly accurate line outage detection is possible. Using concepts similar to principal component analysis, a novel line outage detection algorithm is developed. Lastly, the efficacy of the novel line outage detection algorithm is examined using both steady-state and dynamic simulations.

DEDICATION

This thesis is dedicated to all those who have helped me along the way. Thank you to my family, my friends and my wife for being so understanding and supportive.

ACKNOWLEDGMENTS

Thank you to all of my professors for their guidance and words of encouragement. A very special thank you to my advisor Dr. Elham Makram for working so hard on my behalf during an especially difficult year.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER ONE

INTRODUCTION AND LITERATURE REVIEW

1.1 Introduction

The modern power system is one of the largest, most complex systems in existence. As such, it requires complex algorithms to both operate and control. Therefore, it is difficult to discern the exact state of a power system at a given moment in time. Many quantities of interest are not directly measured and those that are may not be time correlated [1]. Phasor measurement units (PMUs) were designed to alleviate this problem. PMUs measure phasor quantities like bus voltage magnitudes and angles tagged with their time of measurement. Each PMU utilizes a common time source so that many PMUs may be synchronized together. For this reason, the phasor measured by a PMU is also known as a synchrophasor or synchronized phasor.

Phasor measurement technology has its origins in the 1970's with the development of the Symmetrical Component Distance Relay (SCDR) [2]. Research on the SCDR subsequently led to the Symmetrical Component Discrete Fourier Transform or (SCDFT). The advent of the SCDFT allowed the calculation of positive sequence voltages and currents to be performed more quickly and more accurately than ever before. Many researchers realized that such precise measurements could be used in applications other than protective relaying. The possibility of using multiple PMUs at multiple different locations was promising. However, since no common time source was available, the measurements could never be directly compared. Even very small time differences meant that each measurement might be taken during entirely different

operating conditions. Synchronization of phasor measurements became possible when the Global Positioning System (GPS) came online in 1978 [2]. GPS enabled phasor measurements to be related to a common and highly accurate time reference. Thus, measurements taken relative to the GPS clock could be aggregated at a common location called a phasor data concentrator and aligned so that the absolute time reference was coincident between all measurements. Figure 1.1 shows how a GPS time source can be used to provide an absolute time reference.

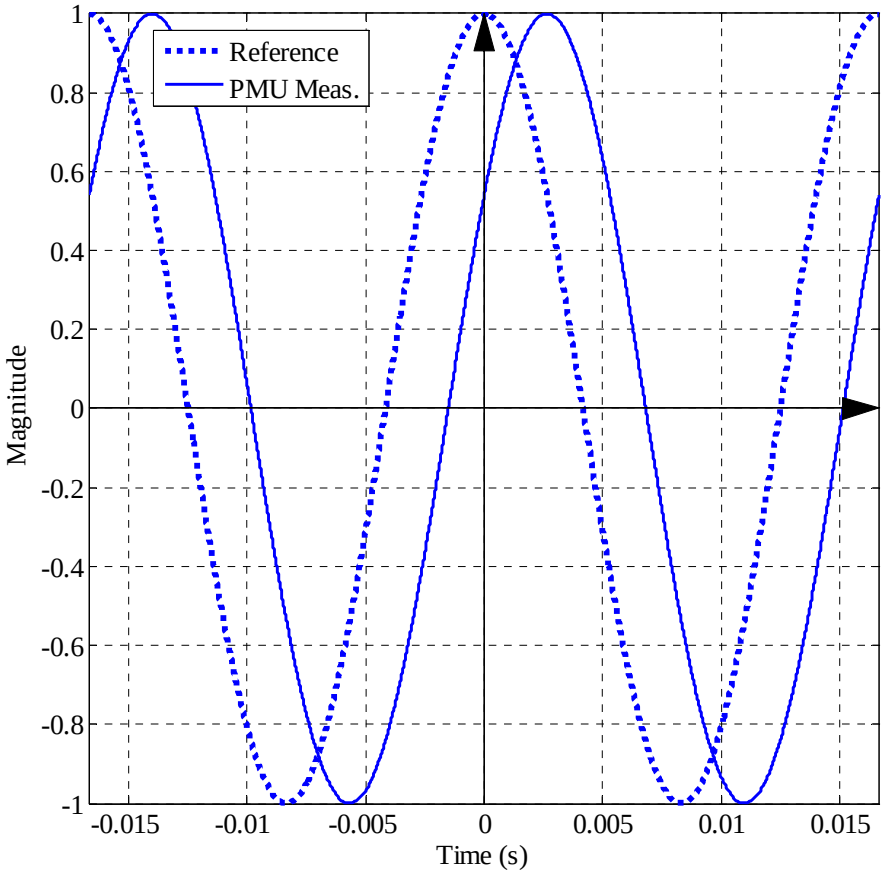


Figure 1.1 – Time referenced phasor measurement.

The current standard phase reference is a cosine function operating at nominal system frequency whose peak occurs on the second rollover [3]. Researchers at Virginia

Tech utilized the GPS time source to create the first PMU [4]. Many new applications were then developed to take advantage of the time aligned phasor data.

One of the most natural applications of synchrophasors was in the area of state estimation (SE). In the 1960's, as power systems grew, it became increasingly important to estimate the state of a power system for economic and security reasons [4]. Bus voltage magnitudes and angles were estimated at first using active and reactive line flows and subsequently using bus voltage and injection measurements. One disadvantage of these original systems was their possible slow time to convergence. Depending on the application and the size of the system in question, the results of the state estimator could be obsolete by the time the estimate converged. Phadke [4] was able to show a marked increase in SE performance if the algorithm utilized accurate bus voltage magnitude and angle measurements. The addition of these measurements eliminated the need to measure many of the line flows as required by traditional state estimation. With fewer measurements, the estimate converged faster. In fact, given a magnitude and angle measurement at every bus, the algorithm would converge in a single iteration. The inclusion of synchronized phasors into state estimation spurred many others to investigate the usefulness of this new tool.

Current synchrophasor technology has advanced far beyond state estimation and the system first developed at Virginia Tech. While the original PMUs were stand-alone devices, current synchrophasor technology exists mainly as an added feature in microprocessor based relays. Schweitzer et. al. [1] discuss some current applications of synchrophasors using their synchrophasor enabled microprocessor relays. Verifying that

a substation relay is correctly wired is easier using synchrophasors since the angle of voltage in each phase is measured relative to GPS time as opposed to a reference phase. Synchrophasors can also help to verify Supervisory Control and Data Acquisition (SCADA) systems. Since PMUs are able to measure phasors up to 30 times per second, synchrophasor measurements can measure waveforms with a greater resolution than a traditional SCADA system. Additionally, synchronized phasor measurements are useful for monitoring and control of large, geographically dispersed areas. These measurements allow engineers to capture voltage and current waveforms during wide area disturbances and can be used to perform corrective action such as tripping distributed generation. Monitoring of wide area disturbances is fundamental to an increased awareness of critical situations in large power systems.

A lack of knowledge about a system event, dubbed "situational awareness" has been identified as a cause of four of the six major North American blackouts [5]. While protective relays can protect against local faults and disturbances, little focus has historically been given to protection and monitoring of the wider area of a power system. Traditionally, monitoring the power system as a whole has been difficult due to the lack of accurate, up to date measurements. Synchronized phasor measurements increase situational awareness through systems designed to monitor wide areas of a power system.

1.2 Wide Area Monitoring Systems

The overall objective of the Wide Area Monitoring or Wide Area Measurement System (WAMS) is to provide a more complete knowledge of the power system at large. Hadley et. al. [6] describe a WAMS as a system which complements the existing

Supervisory Control and Data Acquisition System (SCADA) to help manage and understand large, complex power systems. Wide Area Measurement Systems serve to complement SCADA systems by providing real time data for increased situational awareness and event analysis [6]. The first wide area monitoring systems were designed for state estimation and only utilized line flow measurements [1], [2]. However, once synchrophasors were introduced into state estimation, other applications and implementations like the WECC WAMS were created.

As a result of an increased want for dynamic power system information, the Department of Energy (DOE) helped to create the first synchrophasor based WAMS project in conjunction with the Western Electricity Coordination Council (WECC) [7]. Since its inception the (WECC) has gained much experience from DOE's project. One benefit of this project has been increased development of EPRI's FACTS system which pairs with WAMS measurements to provide wide area control. As of 2004, the WECC WAMS contained 60 PMUs and 11 phasor data concentrators (PDCs). The WECC system has led to many other WAMS and led to two IEEE standards. Initially, standard 1344-1995 developed basic measurement ideas and data formatting. The new standard C37.118 was created to address issues like the definition of an "Absolute Phasor," TVE, and PMU compliance tests.

One use of WAMS which was not initially intended, but has grown out of years of experience is dynamic modeling of the power system and its validation. Thus, when a dynamic event occurs, actual measurements can be compared with simulated ones to determine the validity of a model. Assuming accurate system models, WAMS could

ultimately be used for very tight closed loop system control. Control applications with delays on the order of 100s of milliseconds are possible with synchrophasors.

Initial research in WAMS, however, has been mostly concerned with monitoring to provide better situational awareness. In particular, the area of event detection has received much interest. However, some issues arise which do not allow efficient detection of such disturbances. For instance, the reaction time necessary for an operator to see changes in phasor measurements and take action may be too long. Additionally, the change may be so slight as to be unnoticeable to the naked eye. For these reasons, it is more efficient to detect system events and disturbances using synchronized phasor measurements.

Event detection can be broken down into three separate stages [8]. Detecting the event, extracting event information, and classifying the event are essential to the event detection process. Detection of an event is quite similar to detection of edges in the field of image processing. While various methods have been proposed, most are quite similar to one another. Extracting relevant event information, however, will depend on the specific application. For instance, bus voltage angles are dependent upon many aspects of the power system, so the non-relevant information must be identified and discarded. Once the relevant information is obtained, event classification is performed using the many solutions devised in the area of pattern recognition.

For instance, using simple calculations, the authors of [9] designed a logic based algorithm to detect voltage instability. Information required of the algorithm included phasor measurements, real and reactive power flows, and frequency information. The

logic based approach was divided into two steps. The first step was labeled as filtering and the second step was labeled as ranking and analysis. Filtering of system disturbances was based on voltage deviation, but subsequent ranking was based on voltage, frequency, and voltage angle deviation.

Another approach to detection of voltage instability was investigated in a paper by Khatib et. al. [10]. The authors of this paper utilized decision trees (DT) due to their inherently quick computation time and success in previous research. In order to build the decision tree (DT), training data was used for a number of prototypical sample cases. Throughout the design of the DT, choices were made as to which cases were secure and which were insecure. These choices then dictated where the tree's nodes branched off. Therefore, in order for the decision tree to be most useful, its set of training cases had to encompass all cases for which a test was desired. Herein is the downfall of the decision tree approach. In order to train the decision tree, five loading conditions were used with various simulated contingencies and the subsequent margin to voltage collapse was then calculated. In total, 166 different scenarios under five loading conditions were simulated for a total of 830 sets of data points. The authors point out that the placement of PMUs and the types of phasors (i.e. voltage magnitude and angles, current magnitude and angles) utilized in computation were critical to the algorithm's success. Although no quantitative results were mentioned, the accuracy was said to be comparable to the previous decision tree algorithm whose misclassification rate was cited as 18% [11].

The decision tree type algorithms above attempt to provide not only a means of classifying voltage instability, but also the basis for such classification. In other words,

the decision can be traced from the root of the tree to show the foundation of the solution. Other authors have devised event detection algorithms whose solution is not so easily traced to its roots. Artificial Neural Networks (ANNs) were used in [12] to detect fault locations in double circuit transmission lines. Faults in parallel lines create a non-linear impact on mutual coupling in unfaulted phases which prompted the authors to forgo an attempt at a deterministic model. Rather, a two pronged approach was utilized. First, prototypical features were extracted using unsupervised learning. Next, supervised, on-line classification was performed on those features. The authors cited a misclassification rate of 1% out of 100 test cases. In the same article, an ANN approach to voltage instability detection was also mentioned. In both cases, a large number of operating conditions had to be simulated to train the systems.

Smith and Wedeward [13] utilized a constrained optimization approach for both detection and localization of power system events. The authors simulated the dynamic nature of line outages in MATLAB then used the results to perform online constrained optimization resulting in time-series data for bus voltage magnitudes and angles. The difference in bus voltage magnitudes and bus voltage angles were then used to determine and locate line outages. Performance of the algorithm was measured based on the proximity in number of buses to the actual buses involved in the outage. On average, the optimization algorithm was 5 to 6 buses away from the true outage buses. Here again, the authors created an algorithm whose results were not directly traceable to the source and whose computation time was debilitating.

Tate and Overbye described how synchrophasor measurements can be used for detecting single and double line outages [14], [15]. Using the DC approximations of a power flow, the authors were able to detect line outages with varying degrees of accuracy. Distribution factors based on the DC power flow assumptions were pivotal in creating the line outage detection algorithm and well known in the area of contingency analysis [16]. Since synchrophasors can measure voltage angles in near real-time, Overbye and Tate showed that the same DC distribution factors can be used in on-line analysis. One potential downfall of this algorithm was that it requires a line flow measurement on every line in addition to phasor angle measurements. Once detected, a line outage was classified using an exhaustive nearest neighbor search based on the Euclidean distance measure. PMU placement is also mentioned as being critical to this algorithm's success since it is assumed that only a few key buses will be measured.

While many of the aforementioned papers utilize bus voltage magnitudes and angles as indication of power system events, the authors in [17] use frequency deviations at wall outlets as indication of power system events. A study was performed with 10 frequency monitoring devices geographically dispersed across the United States. Both location and magnitude of generator tripping were studied. Utilizing the relative time of frequency deviation between the different monitoring locations, the events were localized on Cartesian coordinates. By first assuming the rate at which the electromechanical wave propagates, the authors are then able to solve a least squares problem to find the probable hypocenter of the event. Solutions based on Newton's method and gradient descent were also examined. In all three cases, the power system event was first localized in Cartesian

coordinates then possible events were derived based on knowledge of actual system topology and measurement location. A common thread here and with many other event detection algorithms is the sensitivity to PMU placement.

1.3 PMU Placement

With critical applications such as power system control and wide area protection, the location of a phasor measurement becomes increasingly important. Due to high cost, it is unrealistic to place stand-alone phasor measurement units at every bus in a power system. To help mitigate this cost, phasor measurement units are being integrated into microprocessor based protective relays. Still, not every bus in a power system will require even a microprocessor based relay.

Many methods for PMU placement and optimizing such placement have been proposed. Baldwin et. al. and Clements [18], [19] described power system observability and its use in PMU placement. Locating a power system's PMUs based on bus connectivity alone, however, does not take into full consideration the effect of the sensitivity of certain buses to changes in the power system. Zhao [20] compared multiple methods of PMU placement constrained by sensitivity indices. While sensitivity constraint did improve each method, the quickest and most simple solution was via linear programming [21]. In most cases, it has been shown that complete power system observability can be achieved with PMU installations on one third of the total number of buses.

1.4 Thesis Overview

The main focus of this thesis is to examine a technique for detecting line outages using bus voltage angle measurements on some or all of the buses in a power system. Design goals for the proposed algorithm include minimizing the time of computation as well minimizing the number of required system measurements. For this reason, bus voltage angles via synchrophasors were used as the primary measurement type. Vutsinas [22] provides proof that bus voltage angles, in addition to current magnitudes, are the major polarizing quantities between differing system contingencies. Therefore, a technique known as Principal Component Analysis (PCA) is performed on the difference in pre and post contingency voltage angles and is examined in Chapter 2. These results not only show the usefulness of such a technique, but they also led to the development of a novel algorithm based on the DC power flow assumptions. A derivation of the proposed algorithm is presented along with a detailed numerical example in Chapter 3. Throughout Chapter 4, the new algorithm is compared to the algorithm devised by Overbye and Tate in [14] and [15] using steady state MATLAB simulations on a reduced 47 bus test system. Finally, the dynamic efficacy of the proposed algorithm is examined in 0 using actual PMU data and a PSS/E simulation.

CHAPTER TWO

PRINCIPAL COMPONENT ANALYSIS

2.1 PCA Principles

The technique known as Principal Component Analysis (PCA) is primarily used as a tool for reducing the dimensionality of large data sets [23]. In a power system with many PMUs and large numbers of measurements, data reduction techniques like PCA will prove invaluable. PCA reduces the dimensionality of a data set by transforming the data to a new set of (possibly fewer) variables which both remove correlation and retain as much of the original variation as possible [24]. Many variations and techniques similar to PCA exist which utilize these same basic ideas. In an effort to encourage further research into PCA in power systems, some basic principles of PCA will be described. Next, the use of PCA in detecting human faces in images known as eigenfaces will be presented. The eigenfaces techniques will be adapted to power systems in section 2.3. Finally example of this adaptation will be presented. The following is a derivation of PCA adapted from [24] to be used as a basis for later discussion.

Suppose the following Figure 2.1 shows a two dimensional dataset consisting of students' class attendance in relation to their overall grade. Clearly, the students with the highest overall average attend class the most. Visually, the trend appears like a line with positive slope toward the right of the figure. In fact line could be drawn on the figure to approximate the relationship between the two variables. The trend is easy to see from the figure, but if more factors are also considered (i.e. time spent studying, extracurricular activities, additional tutoring) the relation between these factors is less obvious and

cannot be represented by a line in two dimensions. PCA can help elucidate these more complicated trends. Note that in a power system, the data will almost never be as simple as shown below.

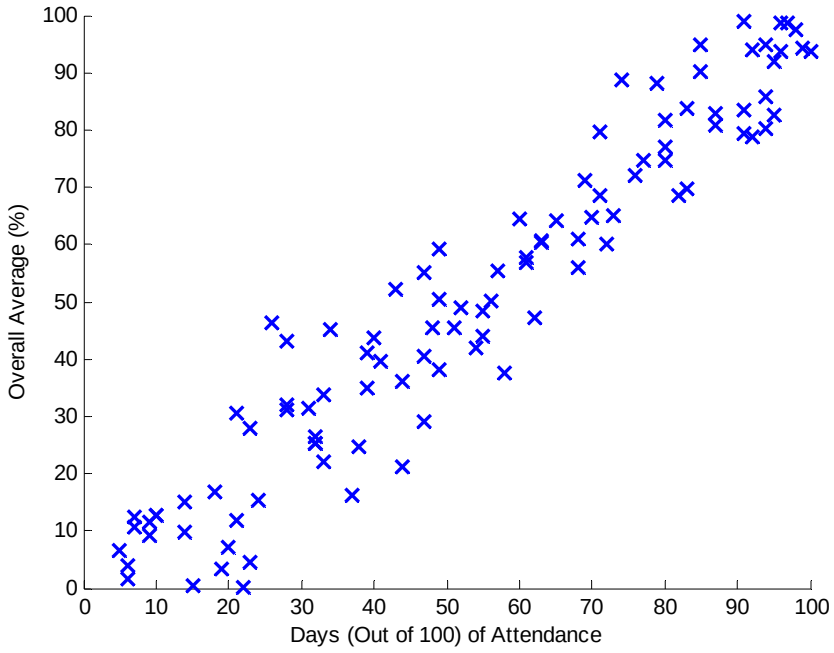


Figure 2.1 – Two-dimensional class attendance plot.

Each of the data points above contains some common attribute which determines the largest extent to which it varies and some secondary attribute which determines the remaining degree of variation. It should be noted that, in general, PCA is used on data sets whose dimensionality is much larger than two. Two dimensional data is used to allow the dataset to be plotted. Using principal component analysis, either of these two common attributes can be removed so that only one remains. For notational purposes, Table 1 below shows a sample of the data above. Student attendance will be labeled x_1 and the overall score will be labeled x_2.

Table 2.1 – Sample classroom attendance values.

Attendance (x_1)	Overall Score (%) (x_2)
69	71.17
71	68.69
76	71.94
77	74.73
74	88.83
80	74.61
79	88.11

Consider the components x_1 and x_2 to be part of the two dimensional data set $\mathbf{x}$. The first goal of PCA is to find a number of linear functions having maximum variance which describe the data. In this way, a linear function will describe the attribute which makes the data vary the most and another linear function will describe the remaining variation. To begin assume a linear function $\mathbf{v}_1^{\mathbf{T}}\mathbf{x}$ exists describing the greatest variation written such that:

$$\mathbf{v}_1^{\mathbf{T}}\mathbf{x} = v_{11}x_1 + v_{12}x_2 = \sum_{i=1}^{2} v_{1i}x_i \tag{2.1.1}$$

Another similar linear function $\mathbf{v}_2^{\mathbf{T}}\mathbf{x}$ which is totally uncorrelated with $\mathbf{v}_1^{\mathbf{T}}\mathbf{x}$ can be written as:

$$\mathbf{v}_2^{\mathbf{T}}\mathbf{x} = v_{21}x_1 + v_{22}x_2 = \sum_{i=1}^{2} v_{2i}x_i \tag{2.1.2}$$

In order for both functions to be uncorrelated, any variation described by one function cannot also be described by the other. The two functions can then be thought of as separate components which when summed, described the entire dataset. For our example, only two such linear functions can be created in this way. However, as many

linear functions can be written as the dimension of the data set with the requirement that each is mutually uncorrelated with the others. From linear algebra, we know that two uncorrelated vectors are orthogonal to one another if the projection of one onto the other has zero length. Before describing the process for calculating these linear functions of $\boldsymbol{x}$, first examine the result plotted against the original data. Clearly, the lines are orthogonal to one another and the trend described earlier is blatantly clear.

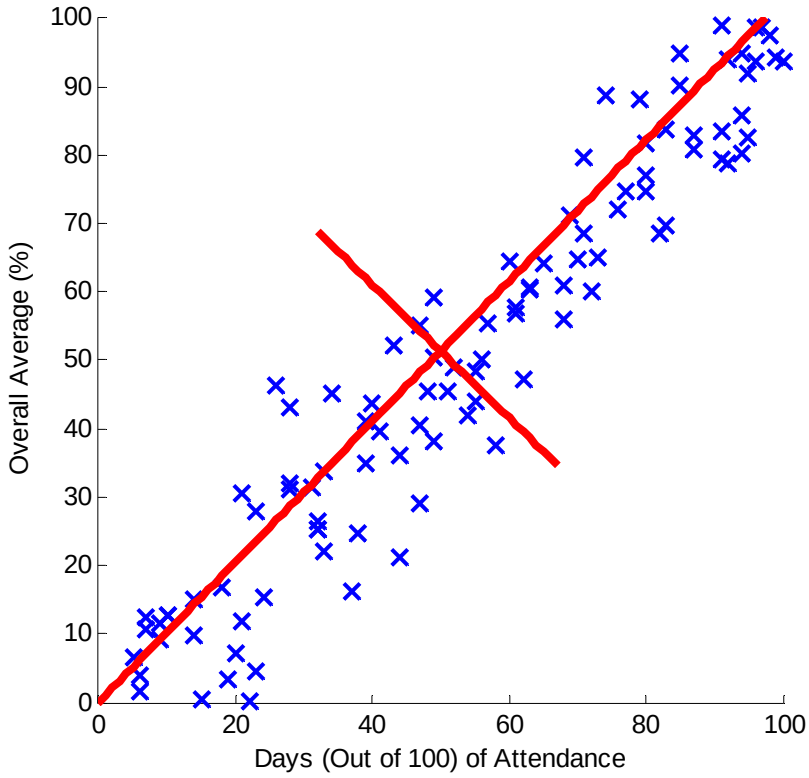


Figure 2.2 – Classroom attendance showing optimal linear functions.

Figure 2.2 above shows the directions of maximum variance, not the degree to which they vary. The lengths of the lines above differ merely to emphasize the fact that the degree of variance is unequal between the two. A common stipulation is to constrain

the direction vectors to length one, so that $\|\mathbf{v_1}^T\mathbf{v_1}\| = 1$ and $\|\mathbf{v_2}^T\mathbf{v_2}\| = 1$. Therefore, it is possible to form a constrained optimization problem using LaGrange multipliers [25]. Since the goal is to find vectors which optimize the variance, the objective function can be written as $\text{var}[\mathbf{v_1}^T\mathbf{x}] = \mathbf{v_1}^T\boldsymbol{\Sigma}\mathbf{v_1}$ where $\boldsymbol{\Sigma}$ is the covariance matrix of $\boldsymbol{x}$. Thus, the goal of the optimization is to maximize the following function:

$$\mathbf{v_1}^T\boldsymbol{\Sigma}\mathbf{v_1} - \lambda\left(\mathbf{v_1}^T\mathbf{v_1} - 1\right) \tag{2.1.3}$$

In order to maximize this function, the first derivative is taken with respect to $\boldsymbol{v_1}$, and the point where the result equals zero is determined as follows:

$$\boldsymbol{\Sigma}\mathbf{v_1} - \lambda\mathbf{v_1} = 0 \tag{2.1.4}$$

Equivalently:

$$\left(\boldsymbol{\Sigma} - \lambda\mathbf{I}\right)\mathbf{v_1} = 0 \tag{2.1.5}$$

This form indicates that $\boldsymbol{\lambda}$ is an eigenvalue of $\boldsymbol{\Sigma}$ with corresponding eigenvector $\boldsymbol{v_1}$. However, both λ and $\boldsymbol{v_1}$ are still unknown, so the question is which of the eigenvalue, eigenvector pairs to choose. Since the dimensionality of the problem is two, there will be two such eigenvalue, eigenvector pairs. From above, $\boldsymbol{\Sigma}\mathbf{v_1} = \boldsymbol{\lambda}\mathbf{v_1}$, and recall that the objective is to maximize $\mathbf{v_1}^T\boldsymbol{\Sigma}\mathbf{v_1}^T$. Thus, the objective function can be rewritten as follows:

$$\mathbf{v_1}^T\boldsymbol{\Sigma}\mathbf{v_1} = \mathbf{v_1}^T\lambda\mathbf{v_1} = \lambda\mathbf{v_1}^T\mathbf{v_1} \tag{2.1.6}$$

and since $\|\mathbf{v_1}^T\mathbf{v_1}\| = 1$

$$\lambda\,\mathbf{v_1}^T\mathbf{v_1} = \lambda(1) = \lambda \tag{2.1.7}$$

In other words, the objective function is a scalar, so the maximum value of λ is simply the largest eigenvalue. This relationship indicates that $\boldsymbol{v_1}$ is the eigenvector of $\boldsymbol{\Sigma}$ which corresponds to the largest eigenvalue which will be called λ_1. The vector $\mathbf{v_1}$, when multiplied by $\boldsymbol{x}$, is called the first principal component (PC) of $\boldsymbol{x}$. Using the technique mentioned above will elicit the first PC, but in order to find the remaining principal components, a slightly modified technique is used.

In order to find the second principal component from the second line function or $\mathbf{v_2}^{\mathrm{T}}\mathbf{x}$, the objective is still to maximize $\mathbf{v_2}^{\mathbf{T}}\boldsymbol{\Sigma}\mathbf{v_2}^{\mathbf{T}}$, such that $\|\mathbf{v_2}^{\mathbf{T}}\mathbf{v_2}\| = 1$, but an additional constraint must also be met. The new constraint assures that the second principal component is completely uncorrelated with the first. Essentially, the constraint causes the covariance between the first and second principal components to be zero:

$$\mathrm{cov}(\mathbf{v_1}^{\mathbf{T}}\mathbf{x}, \mathbf{v_2}^{\mathbf{T}}\mathbf{x}) = 0 \tag{2.1.8}$$

by definition:

$$\mathrm{cov}(\mathbf{v}_1^T\mathbf{x}, \mathbf{v}_2^T\mathbf{x}) = \mathbf{v}_1^T\boldsymbol{\Sigma}\mathbf{v}_2 = \mathbf{v}_2^T\boldsymbol{\Sigma}\mathbf{v}_1 = \mathbf{v}_2^T\lambda_1\mathbf{v}_1 = \lambda_1\mathbf{v}_2^T\mathbf{v}_1 = \lambda_1\mathbf{v}_1^T\mathbf{v}_2 = 0 \tag{2.1.9}$$

Or

$$\lambda_1\mathbf{v}_1\mathbf{v}_2 = 0 \tag{2.1.10}$$

Since $\boldsymbol{\Sigma}$ is a positive semi-definite matrix, λ_1 is necessarily non-zero, so the constraint becomes $\mathbf{v_1}^{\mathbf{T}}\mathbf{v}_2 = 0$. Now, the optimization problem can be formulated as follows:

$$\mathbf{v}_2^T\boldsymbol{\Sigma}\mathbf{v}_2 - \lambda\left(\mathbf{v}_2^T\mathbf{v}_2 - 1\right) - \phi\mathbf{v}_2^T\mathbf{v}_1 \tag{2.1.11}$$

Differentiation with respect to $\mathbf{v}_2$ yields:

$$\mathbf{\Sigma v}_2 - \lambda \mathbf{v}_2 - \phi \mathbf{v}_1 = 0 \tag{2.1.12}$$

If the equation above is pre-multiplied by $\mathbf{v}_1$, the result is:

$$\mathbf{v}_1^T \mathbf{\Sigma v}_2 - \mathbf{v}_1^T \lambda \mathbf{v}_2 - \mathbf{v}_1^T \phi \mathbf{v}_1 = 0 \tag{2.1.13}$$

From the equation above, the first two terms are zero, so only $\mathbf{v}_1{}^T \phi \mathbf{v}_1 = 0$ is left. Therefore ϕ must be zero since $\mathbf{v}_1{}^T \mathbf{v}_1 = 1$. When $\phi = 0$, however, the optimization problem becomes:

$$\mathbf{v}_2^T \mathbf{\Sigma v}_2 - \lambda \left(\mathbf{v}_2^T \mathbf{v}_2 - 1 \right) \tag{2.1.14}$$

This is the same as before, so $\mathbf{v}_2$ is the eigenvector corresponding to eigenvalue λ where $\lambda = \mathbf{v_2}^\mathbf{T} \mathbf{\Sigma v_2}^\mathbf{T}$ is to be maximized. The vector $\mathbf{v}_2$ is distinct from $\mathbf{v}_1$, so λ is distinct from λ_1 which means that $\lambda = \lambda_2$, the second largest eigenvalue of $\mathbf{\Sigma}$ and $\mathbf{v_2}^\mathrm{T}\mathbf{x}$ is the second principal component. Thus, it is possible to reason inductively, that the remaining principal components can be found in a similar fashion, removing all previous PCs' correlation from the new PC. Without delving further into the subject, it is sufficient to state that many variations exist on the derivation and practical calculation of principal components. Many applications of PCA also exist.

2.2 PCA in Face Recognition

One of the primary uses of PCs, dimension reduction, is possible by selecting fewer principal components than the total dimensionality of a data set, but which still capture the largest amount of variance in the data set. Using similar techniques, applications other than dimension reduction are possible. Turk and Pentland [26] used a technique they called "eigenfaces" to both detect and classify human faces. The process

was two-fold. First a set of prototype or training faces was used to create a face space. Next, when an unknown image was to be classified, the new image was projected onto each of the training faces to produce a set of weights. To determine if the new image was a face, the distance between the face space and the new image was found. Finally, if the new image was identified as a face, it was classified by comparing its weights to known weights.

Since images are two-dimensional, the pixels of each of the training and any new image were stacked column by column to form one long column vector. For the set of training faces, an average face was determined by finding the row average of all training faces. Next, the difference between the training faces and the average face was found. This resultant matrix of mean-centered faces was subjected to a principal component analysis to find a set of eigenvalues and eigenvectors which optimally described the variance of the matrix. The eigenvectors are dubbed eigenfaces. Only a small number of eigenfaces which describe most of the variation must were retained. When a new image was encountered, it was first mean-centered, and subsequently projected onto each of the retained eigenfaces. This projection resulted in a number of weights equal to the number of retained eigenfaces. The weights were stacked vertically in a vector then compared to known weight vectors. Identification was based on which weight vector was the closest in terms of Euclidean distance from the calculated weight vector of the unknown face.

2.3 PCA Adapted to Line Outage Detection

Using ideas similar to Turk and Pentland [26], line outages can be detected in a power system. Rather than columns of pixels, however, the bus voltage angular differences can be used. Assume an m-bus power system has the following steady-state angles before and after a line outage:

$$\begin{bmatrix} \theta_{1,pre} \\ \theta_{2,pre} \\ \vdots \\ \theta_{m,pre} \end{bmatrix} \xrightarrow{\text{Line Outage}} \begin{bmatrix} \theta_{1,post} \\ \theta_{2,post} \\ \vdots \\ \theta_{m,post} \end{bmatrix}$$

The difference in these two vectors can be formed as follows:

$$\boldsymbol{\theta}_{\mathbf{post}} - \boldsymbol{\theta}_{\mathbf{pre}} = \boldsymbol{\Delta\theta} \tag{2.3.1}$$

Of course, this analysis assumes that measurements of the bus voltage angles are available at every bus. While this assumption is not realistic it can be relaxed later.

Given a set of possible or typical operating conditions before an outage, many of these $\boldsymbol{\Delta\theta}$ vectors can be formed. For *n* simulated conditions, a matrix can be formed as shown below.

$$\mathbf{T} = \begin{bmatrix} \Delta\theta_{1,1} & \Delta\theta_{1,2} & \cdots & \Delta\theta_{1,n} \\ \Delta\theta_{2,1} & \Delta\theta_{2,2} & \cdots & \Delta\theta_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ \Delta\theta_{m,1} & \Delta\theta_{m,2} & \cdots & \Delta\theta_{m,n} \end{bmatrix} \tag{2.3.2}$$

Each of the columns is an angular difference for a specific line outage at a given loading \ generation condition. Using the principal component techniques on the covariance

matrix of **T** described above, the set of optimal eigenvectors $\boldsymbol{v_m}$ and eigenvalues λ_m can then be found as follows:

$$\Sigma = \frac{1}{m-1}\left[\mathbf{T}-\boldsymbol{\mu}_\mathbf{T}\right]\left[\mathbf{T}-\boldsymbol{\mu}_\mathbf{T}\right]^T \text{ and } \Sigma \mathbf{v_m} = \lambda_m \mathbf{v_m} \tag{2.3.3}$$

In the equation above $\boldsymbol{\mu}_\mathbf{T}$ is the row mean or a column vector of averages taken across all the columns (operating conditions) of **T**. Similar to the eigenfaces technique, a small number of vectors called "principal outage vectors" can be retained which describe the maximum amount of variation in the matrix **T**. From here, the process follows along exactly with the eigenfaces technique. A numerical example of the application of this technique is shown below.

2.4 Principal Outage Vectors Example

The following is a brief example of the principal outage vector technique using a 6-bus test system from Wood & Wollengberg [16]. The system data can be found in Appendix A. Simulations were performed using MATPOWER in MATLAB [27]. All power flow results were calculated using the full Newton-Raphson power flow solution. Using the data provided in Appendix A as a base case, 100 separate operating conditions were simulated. In each case, a random set of load values was created using a Gaussian normal distribution with a standard deviation of 30MW and a mean centered at the base case value for each bus. The power flow solution was then calculated for each of the operating conditions. Next, each line was removed individually from the system at each operating condition and the power flow solution was again calculated. From the pre-outage and post-outage bus voltage angles, a vector of angular differences $\boldsymbol{\Delta\theta}$ was

formed yielding 1100 such vectors (100 loading conditions, 11 lines outaged). The matrix **T** was then formed where each column was an angular difference as shown below. Note that bus 1 is the reference bus, so its angle and angular difference will always be zero.

$$\mathbf{T} = \begin{bmatrix} \Delta\theta_{1,1,1} & \Delta\theta_{1,2,1} & \cdots & \Delta\theta_{1,11,1} & \Delta\theta_{1,1,2} & \cdots & \Delta\theta_{1,11,100} \\ \Delta\theta_{2,1,1} & \Delta\theta_{2,2,1} & & \Delta\theta_{2,11,1} & \Delta\theta_{2,1,2} & & \Delta\theta_{2,11,100} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \Delta\theta_{6,1,1} & \Delta\theta_{6,2,1} & \cdots & \Delta\theta_{6,11,1} & \Delta\theta_{6,1,2} & \cdots & \Delta\theta_{6,11,100} \end{bmatrix} \tag{2.4.1}$$

The subscript format for each element can be written as $\Delta\theta_{bus,\ line,\ condition}$ where each column represents the change in angles at all buses due to a specific loading / generation condition, 100 in total. Next, PCA was performed on the **T** matrix to yield six eigenvalue, eigenvector pairs as shown below. During the principal component analysis the row mean was determined to be:

$$\boldsymbol{\mu}_{\mathbf{T}} = \begin{bmatrix} 0.00 \\ 0.70 \\ 0.67 \\ 1.00 \\ 0.98 \\ 1.22 \end{bmatrix} \tag{2.4.2}$$

The matrix **V** below contains the eigenvectors and the column vector $\boldsymbol{\lambda}$ contains the eigenvalues. Column *i* in **V** corresponds to the eigenvalue in row *i* of $\boldsymbol{\lambda}$.

$$\mathbf{E} = \begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.90 & 1.00 & 0.23 & 0.37 & -0.04 \\ 0.00 & 1.00 & -0.18 & 0.44 & -1.00 & 0.35 \\ 0.00 & 0.79 & -0.30 & 0.02 & -0.06 & -1.00 \\ 0.00 & 0.85 & 0.03 & -1.00 & -0.16 & 0.20 \\ 0.00 & 0.88 & -0.58 & 0.21 & 0.96 & 0.35 \end{bmatrix} \tag{2.4.3}$$

$$\boldsymbol{\lambda} = \begin{bmatrix} 0.00 \\ 17.35 \\ 0.47 \\ 0.37 \\ 1.25 \\ 1.58 \end{bmatrix} \tag{2.4.4}$$

Note that the first column of **V** provides no useful information since its eigenvalue is zero. This should be clear because column one corresponds to the system reference whose angular difference is necessarily zero. Also, note the locations of the largest eigenvalues. Bus 2 and Bus 6 have the two highest values. Next, each mean-centered column of **T** was projected, individually onto each column of **V** (the principal outage vectors) to produce column vectors of weights.

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_6 \end{bmatrix} \text{ where } w_i = \mathbf{v_i}^T \left[\boldsymbol{\Delta\theta_i} - \boldsymbol{\mu_T} \right] \tag{2.4.5}$$

This produced a set of 1100 weight vectors ***w*** corresponding to the weights of known line outages under varying conditions. These are the prototypes which can be used for line outage identification.

A series of new, random loading conditions was then generated which utilized the same normal distribution centered at the base case with 30MW standard deviation. These new loading conditions represent test data used to test the efficacy of the principal outage vector technique. As before, a power flow solution was calculated both before and after each line in the system was removed for each of a set of 100 loading conditions and the angular difference was subsequently found. For each test case, the row mean of the prototype set **T** was subtracted from the angular difference. Then, the test vectors were projected onto the six dimensional principal outage vector space to produce weight vectors, $\boldsymbol{w}$ by applying Equation (2.4.5). In order to determine which line outage the weights correspond to, a nearest neighbor search was performed. Nearest neighbor search was utilized since the underlying statistical nature of the problem was unknown. Using Euclidean distance, the weight vector from the prototype set which was closest to each weight vector of the test set was flagged as the line outage class for the corresponding test vector. As a measure of accuracy, the success rate for this test was calculated as:

$$\%\ Success = \frac{Correct\ \ Identifications}{Total\ \#\ of\ \ Tests} \times 100 \tag{2.4.6}$$

where the *Total # of Tests* was 1100 (100 test conditions and 11 simulated line outages). Shown below are the results for 5 separate iterations of this test; each iteration containing randomly generated loading, different from the last.

Table 2.2 – Success of principal outage vectors, full coverage.

Iteration	*% Success*
1	99%
2	99%
3	98%
4	99%
5	96%

Although the results of the tests above are promising, they rely on the assumption that a voltage angle measurement is available at every bus in the power system. Realistically, this is never the case. Rather than assuming angle measurements at every bus, a new simulation was created where only the angles at buses 2 and 6 were used for both the prototype and test sets. The success rate is shown below.

Table 2.3 – Success of principal outage vectors, two PMUs.

Iteration	*% Success*
1	98%
2	89%
3	90%
4	88%
5	90%

The result of the analysis above shows that, given a large number of simulations under typical operating conditions, it is entirely possible to identify line outage with a high degree of accuracy. New angular difference vectors can be mapped on to the principal outage vector space to determine their similarity to known line outage classes. The mathematical reasoning for this type of analysis is well known and its efficacy has been proven in the area of image processing with the technique known as “eigenfaces.” Therefore, it is entirely possible to create a line outage detection system using these techniques.

For very large systems, the required simulations may consume more time, but since they can be performed off-line, the greatest bottleneck is in searching the weight vectors for closest matches. In this way, a greater number of off-line simulations results in a longer on-line search time. The analysis above utilized nearest neighbor search, but other more efficient search methods could also be investigated. For a given system, the number of simulations required can be quite small. In fact, despite using hundreds of simulations for the system given in Appendix A, further analysis showed that only about ten total simulations were required.

One potential caveat with this system and many others like it is that mathematical reasoning does not directly apply to the problem at hand. To recapitulate, a principal outage vector system may be constructed and function properly, but from a power systems point of view, it is difficult to explain "why" the system works. Therefore, the focus of Chapter 3 is to show an analytical basis for a similar technique which does not require such a large number of off-line simulations. In fact, the only off-line data which is calculated comes from the system impedance matrix.

CHAPTER THREE

LINE OUTAGE DETECTION

3.1 From PCA to LOD

As was shown in Chapter 2, principal component analysis can be used to reduce the dimensionality of a dataset. Reduced dimensionality allows for a data set to be more easily visualized. PCA can be also be used for other applications like detecting faces in images. Most of these techniques utilize very little information about the underlying structure of the data. The eigenfaces technique simply requires that it be trained on images which must differ enough to be salient, but must be centered in a common location. No knowledge is required of the actual physical structure of the human faces being examined. Although, as was shown in Chapter 2, the eigenfaces technique can easily be adapted to power systems, it is not specifically tailored to such an application. Many areas of power systems research rely heavily on the electrical model of the power system. With such a model available, it is possible that techniques like principal component analysis may be used to gain even further insight into power system operation.

In this chapter PCA is utilized to arrive at a new algorithm for detecting line outages in power systems. Although PCA is not directly applied in the algorithm, its utility as an exploratory analysis tool is exemplified. This novel line outage detection algorithm is based on the DC power flow assumptions which are briefly described in section 3.3. A line outage detection algorithm created by authors Overbye and Tate is then described. Overbye and Tate's algorithm also makes use of the DC power flow

assumptions presented to be presented in section 3.3. Next, the theory behind a new, proposed line outage detection algorithm is presented followed by an example of the novel LOD method.

3.2 PCA of Line Outages

In section 2.3, principal component analysis was used to help identify line outages in a fashion similar to eigenfaces. PCA was performed on a set of data which included every line outage, but under various loading conditions. In essence, this created groupings where each group consisted of a certain line outage under various conditions. In order to identify a line outage correctly, a similar loading / generation condition must have already been simulated. Clearly, it is impossible to simulate the gamut of feasible conditions. One particularly glaring shortfall of this method is its lack of generality. In this case, more simulations allow the algorithm to be more general. Therefore, in an effort to find a more extensible method, an attempt was made at using principal component analysis in a different fashion as described below.

Rather than combining all line outages into a single dataset, each line outage was considered as its own dataset. As before, line outages were simulated under various conditions and a set of data whose columns corresponded to randomized loading conditions was subjected to PCA. In this case, however, the eigenvector corresponding to the largest eigenvalue was the only piece of information used to identify a line outage. The reasoning behind this is that the removal of a line is inherently due to a removal of impedance which will change the direction of the corresponding bus voltage angles. PCA can be used to characterize the directions of variability in the bus voltage angles. A

difference in the angles before and after the outage is utilized so that the greatest direction of variability will be due to the outage. Some of this variability will also be caused by noise and the inherently non-linear nature of bus voltage angles. To complicate matters, the impedances are all scaled by injections into the system. The resulting angle changes consist of both a direction and a magnitude where the magnitude is directly proportional to the bus injections. Since the injections will not necessarily be known, the most telling piece of information is the direction of change of bus voltage angles. Quantitatively, this direction is the eigenvector in question. It can be postulated then that each line outage will have a relatively distinct direction of change.

After examining the eigenvectors for line outages under various conditions, it was discovered, as postulated, that the eigenvectors for a given line outage were relatively the same. Regardless of the loading imposed, in general, line outages produced unique eigenvectors. It is important to note that these simulations were performed using full the AC Newton-Raphson power flow method of solution. As will be described in the following section, AC power flow is a non-linear process, so the result is found through iteration. The non-linearity is required for accuracy, but identifying the analytical reasoning for line outages is much more difficult. As a result, the same PCA method was attempted using the DC power flow assumptions (to be reviewed in the following section). Here, it was discovered that the eigenvector corresponding to the largest eigenvalue was not only similar between outages, but was exactly the same. In other words, regardless of what loading / generation conditions were imposed on the system, the eigenvector corresponding to the largest eigenvalue was the same for a certain line

outage. In fact, it was found, this eigenvector was the only eigenvector whose corresponding eigenvalue was significant. This was important since it indicated that the rank of the underlying matrix was unity. To investigate the basis for why this occurred, the process of determining the power flow in a system will be described in the next section.

3.3 Review of Power Flow

In general, the power flow in a power system is governed by basic electric circuit theory. A power flow study is performed in order to determine where and to what degree the active and reactive powers flow [28]. Beginning from Ohm's law and the definition of complex electric power, the following power flow equations are derived:

$$P_i = \sum_{n=1}^{N} |Y_{in}||V_i||V_n| \cos(\delta_{in} + \theta_n - \theta_i) \tag{3.3.1}$$

$$Q_i = -\sum_{n=1}^{N} |Y_{in}||V_i||V_n| \sin(\delta_{in} + \theta_n - \theta_i) \tag{3.3.2}$$

Where N is the number of buses, and i is the bus at which the real power P_i and the reactive power Q_i are injected, the admittance of a branch element in the power system is defined as:

$$|Y_{in}| \angle \delta_{in} \tag{3.3.3}$$

and the bus voltage magnitude and angle at bus i is

$$|V_i| \angle \theta_i \tag{3.3.4}$$

The power flow solution, then, is a process of solving the power flow equations above such that the active power generated equals the active power loss plus the real powers of

the loads. Similarly, the reactive power generated must equal the reactive powers of the connected loads.

Since the power flow problem is non-linear in nature, most solution methods use an iterative approach to arrive at a solution. As with other similar problems, the power flow equations can be linearized about a stable operating point using the Newton-Raphson method. In an attempt provide a faster, though less accurate solution, the DC power-flow was created. A DC power flow represents an entirely linear set of equations which do not require iteration. Some assumptions are made to arrive at the DC power flow solution. First, many large systems have branch impedances whose real part is insignificant compared to the imaginary part:

$$z = r + jx \text{ where } r \ll x \implies z \approx jx \tag{3.3.5}$$

It is important to note that since the impedance is approximately equal to the reactance, the j can be dropped as long as it is known that all calculations are performed on the imaginary components only. Also, in general, if an angle is represented in radians, the sine of that angle is approximately equal to the angle itself:

$$\sin(\theta) \approx \theta \tag{3.3.6}$$

Lastly, when expressed using the per-unit system, the voltages at every bus are approximately equal to 1. With only the real part of the impedance remaining and since the angle δ_{ij} of each impedance is 90°. Thus, the power flow equations become:

$$P_i = \sum_{n=1}^{N} |Y_{in}|(\theta_n - \theta_i) \tag{3.3.7}$$

$$Q_i \approx 0 \tag{3.3.8}$$

Additionally, the power flowing through a single branch from bus *i* to bus *j* can then be approximated as:

$$\begin{aligned} P_{ij} &\cong \frac{|V_i||V_j|}{} \sin\left(\theta_{ij}\right) \\ &\cong \frac{\theta_i - \theta_j}{|X_{ij}|} \end{aligned} \tag{3.3.9}$$

The real power injected at any bus can then be expressed as a sum of the incident branch flows which consist of admittances and bus voltage angles. Therefore, a relation between bus power injections and bus voltage angles can be written in matrix form:

$$\mathbf{P} = \mathbf{Y}\boldsymbol{\theta} \tag{3.3.10}$$

More often, the quantity of interest is the bus voltage angle, since it can be used to determine the line flows as in Equation (3.3.9). For this reason, the DC power flow equations can be expressed in terms of an admittance matrix, **Y**, or an impedance matrix **X**:

$$\boldsymbol{\theta} = \mathbf{Y}^{-1}\mathbf{P} \text{ or } \boldsymbol{\theta} = \mathbf{X}\mathbf{P} \tag{3.3.11}$$

Due to their linear nature, the DC power flow equations are useful in many applications. One particularly important application is in the area of contingency analysis. During normal operation, it is often unrealistic to solve a full power flow in the case of some system contingency. Instead, a set of so-called linear distribution factors is used to quickly calculate the change in line flows or bus voltage angles when system contingencies occur. Overbye and Tate have also shown that such distribution factors

can be used in conjunction with line flow measurements to detect line outages [14] and [15].

3.4 Review of Line Outage Detection

Given line flow measurements and a small number of synchrophasor bus angle measurements, Overbye and Tate (O&T) [14] and [15] have shown that it is possible to detect line outages in a power system. The general process of detecting line outages using their algorithm consists of two steps. First, a model of the power system is analyzed off-line to determine the effect of line outages on bus voltage angles. The change in bus voltage angles is calculated using distribution factors. Step two consists of monitoring synchrophasor measurements on-line for abrupt changes. After an abrupt change occurs, the resulting steady state measurements are compared to the simulations from step one. One or more lines in the system are then identified as being removed.

The off-line analysis utilizes quantities known as power transfer distribution factors or PTDFs which are derived from the DC power flow assumptions. Using the PTDF relating a power transfer between bus *i* and bus *j* from the removal of line *l*, the power injected into the system can be expressed as:

$$\tilde{P}_l = \frac{-P_{ij}}{1 + PTDF_{l,ij}} \tag{3.4.1}$$

With this change in power injected to the system, each of the bus voltage angles will change. However, only a subset of these buses will be observed using PMUs, so the buses to be examined are selected as follows:

$$\mathbf{K} = \begin{bmatrix} I_{K\times K} & 0_{K\times(N-K)} \end{bmatrix} \tag{3.4.2}$$

Where I_{KxK} is the size K identity matrix and the remaining part of the matrix is filled with zeros. Then, angle changes at the observable buses can be written as a function of the bus selection matrix **K** and the DC impedance matrix **X** as:

$$\begin{aligned} \Delta\theta_{calc,l}^{\tilde{P}_l} &= \mathbf{KX}\begin{bmatrix} \vdots \\ \tilde{P}_l \\ -\tilde{P}_l \\ \vdots \end{bmatrix} \begin{matrix} \\ \leftarrow bus\ i \\ \leftarrow bus\ j \\ \\ \end{matrix} \\ &= \tilde{P}_l\mathbf{KX}\begin{bmatrix} \vdots \\ 1 \\ -1 \\ \vdots \end{bmatrix} \begin{matrix} \\ \leftarrow bus\ i \\ \leftarrow bus\ j \\ \\ \end{matrix} \\ &= \tilde{P}\Delta\tilde{\boldsymbol{\theta}}_{\mathbf{calc,l}} \end{aligned} \tag{3.4.3}$$

Thus, a vector of calculated angle changes due to any line outage in the system can be formed.

The on-line analysis relies on the ability to accurately detect system events. By continuously monitoring all bus voltage angles for changes greater than some threshold, it is possible to discriminate system events from normal operating conditions. This threshold τ is dependent upon system parameters, but O&T recommend 0.57 degrees as a starting point based on the IEEE standard for synchrophasor total vector error (TVE) [3]. PMUs are required to maintain noise below this level for normal operation. Once a system event is detected, the difference between the resulting steady state angles after the outage and the steady state angles before the outage is calculated. These angles are stored in a vector called $\Delta\theta_{observered}$.

With a set of calculated, prototypical vectors $\Delta\theta_{calc,l}^{\tilde{P}_l}$ for each line l and an observed vector $\Delta\theta_{observered}$ it is then possible to find the closest match out of the calculated vectors with the single observed vector. However, one potential caveat exists. Since the length of the $\Delta\theta_{observered}$ vector will depend greatly on the loading in the system, the vector must be normalized to a length of one. The same normalization must also be done with each $\Delta\theta_{calc,l}^{\tilde{P}_l}$ vector. Once each vector is normalized, the closest match is determined as follows:

$$\min \left\| \Delta\theta_{observered} - \Delta\theta_{calc,l}^{\tilde{P}_l} \right\| \tag{3.4.4}$$

Performing the minimization above requires one practical modification. The minimization essentially attempts to find the shortest distance between the observed angle changes and all possible angle changes. Distance between any two points is a function of both the direction and magnitude of a straight line between them (assuming Euclidean distance). This distance is more highly dependent upon the scaling of each vector. Scaling comes as a result of the injected powers in the DC power flow. Thus, it becomes necessary to remove the scaling from all of the vectors, so that their lengths are normalized. Overbye and Tate use a value dubbed the NAD or normalized angle distance metric. Figure 3.1 below shows the utility of the NAD.

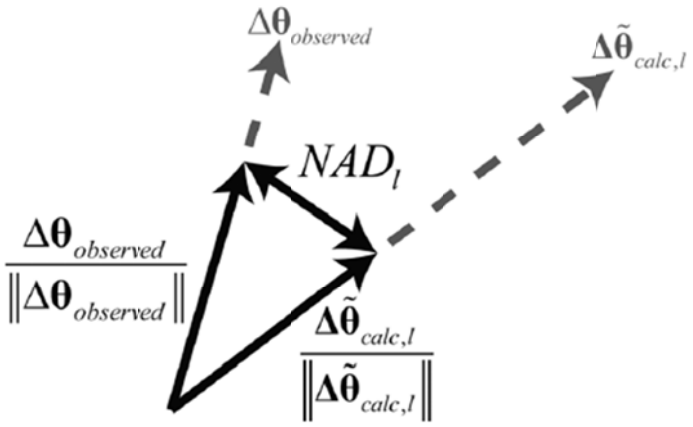


Figure 3.1 – Visual representation of the normalized angle distance (NAD) [14].

The distance between the two vectors above is shorter when their lengths are normalized to 1. Plus, the distance between each vector is no longer dependent upon the scaling of the vector. With a list of NADs between the measured angle changes and all possible calculated angle changes, it is then possible to compare the distances and accurately pick the shortest distance.

3.5 Proposed Line Outage Detection Method

While the method proposed by Overbye and Tate is extremely useful and, as will be shown later, highly accurate, it still requires line flow information in addition to synchrophasor angle measurements. Line flow information is often available, but not necessarily in the same location as synchrophasor measurements. In addition, the lack of a line flow measurement at a given location causes the algorithm to be more prone to error. However, using knowledge of the system impedance and topology along with synchrophasor angle measurements, it is possible to create a similar algorithm which

does not require line flow measurements. The derivation of a new method for line outage detection is presented below.

Using the DC power flow assumptions, it is possible to view the system impedance matrix **X** as a linear transformation. Since **X** is a mapping from vectors in the space of injected powers to the space of bus voltage angles, it can be viewed as the matrix representation of a linear transformation between two finite dimensional vector spaces. Thus, in order to detect line outages, there must be some way to characterize the vectors in the range of **X** as belonging to a specific subset. Each subset represents the possible bus voltage angles which may occur due to an individual line outage. If only two PMU measurements are available the angles lie in an ellipse in two dimensions, but would lie in an ellipsoid if more PMU measurements are available. Figure 3.2 shows this concept diagrammatically for two PMU measurements.

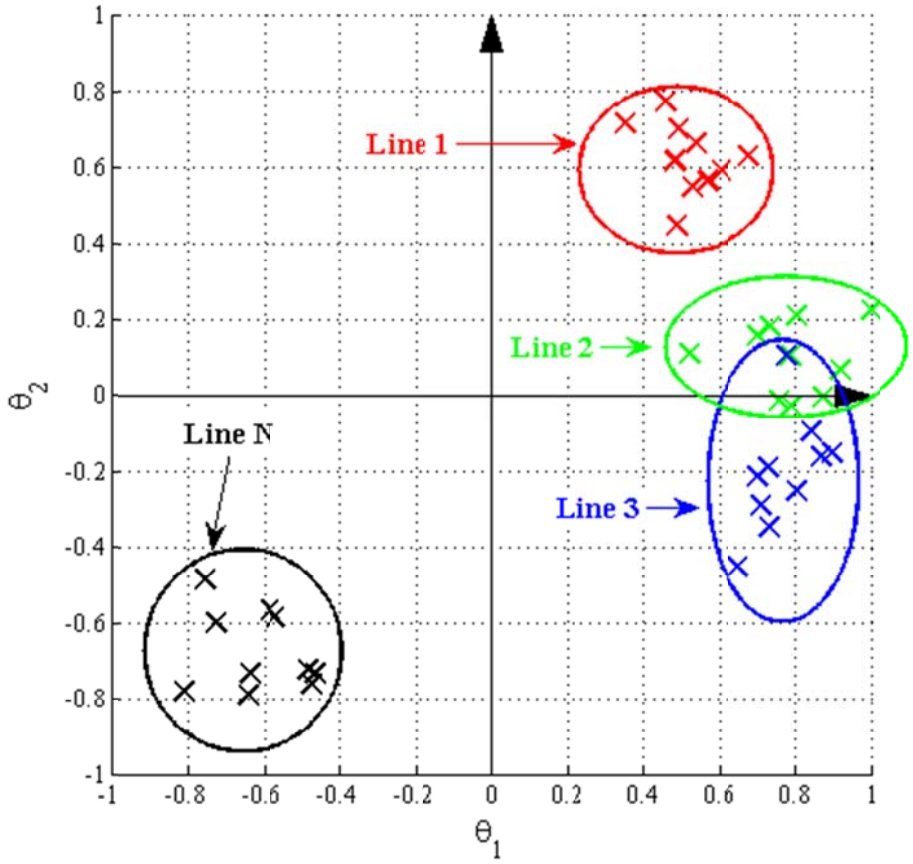


Figure 3.2 – Bus voltage angle difference groupings.

The figure above shows points which correspond to the two bus voltage angles θ_1 and θ_2. The points which lie inside each circle correspond to the angles after a single line is removed, but under different generation \ loading conditions. As shown in the figure above, some line outages are indistinguishable from others. For example, Line 2 and Line 3, when removed, create angles which may not be totally distinguished. This fact is unavoidable when only bus voltage angles are examined. Plus, the size of the system under consideration and the number of bus voltage angles considered greatly determine the distinguishability between line outages.

Seemingly, if the region where each line outage may occur is known through simulation or past experience, then detecting a line outage only requires plotting the angles and determining the region in which they lie. However, this approach would require simulation or historical data for every line outage and every possible generation / loading condition. Realistically, this is impossible. Plus, it is feasible that the bus voltage angles may fall into one of these line outage regions during normal operation regardless of whether an outage has occurred. To combat this problem, the bus voltage angles are continuously examined until one of them changes abruptly, then the difference between the steady state angles before and after the change are examined.

Before a line outage occurs, it is assumed that the linear transformation matrix **X** has been calculated. During and after the outage itself, it is also assumed that a certain number of bus voltage angles are measurable via PMUs. The only unknown quantities then are the injected real powers at each bus.

Since bus voltage magnitudes have been shown to provide the most telling information about power system events, it can be reasoned that their difference before an event and after an event describes the true character of said event. The model of a power system before an outage:

$$\boldsymbol{\theta}_{pre} = \mathbf{XP} \tag{3.5.1}$$

and the same system after a line outage is as follows:

$$\boldsymbol{\theta}_{post} = \left[\mathbf{X} + \Delta\mathbf{X}\right]\mathbf{P} \tag{3.5.2}$$

Due to the line outage, the impedance matrix is modified. The character of this modification is well known, but may be easier to visualize in terms of admittance:

$$\mathbf{Y} = \mathbf{X}^{-1} \Rightarrow \left[\mathbf{X} + \Delta\mathbf{X}\right] = \left[\mathbf{Y} + \mathbf{\Delta Y}\right]^{-1} \tag{3.5.3}$$

The line between bus *i* and bus *j* whose admittance is y_{ij} can be removed from the original admittance matrix **Y** to yield the new admittance matrix $\hat{\mathbf{Y}}$ as follows:

$$\hat{\mathbf{Y}} = \mathbf{Y} + \begin{matrix} \\ (i) \\ \\ (j) \\ \\ \end{matrix} \begin{matrix} \begin{matrix} \quad (i) \quad & \quad (j) \quad \end{matrix} \\ \begin{bmatrix} -y_{ij} & \cdots & y_{ij} \\ \vdots & \ddots & \vdots \\ y_{ij} & \cdots & -y_{ij} \end{bmatrix} \end{matrix} \tag{3.5.4}$$

Typically, this same operation can be modeled with impedances by adding another artificial line of negative impedance equal to the original in parallel with the original line. To remove the effect of the artificial line Kron Reduction is then performed [28].

In the equation above, the negative of the admittance is on the main diagonal, but the actual admittances are in row *i*, column *j* and row *j*, column *i*. For a 3x3 admittance matrix, when removing a line between bus 1 and bus 3, the above equation could be written as:

$$\mathbf{Y} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & Y_{23} \\ Y_{31} & Y_{32} & Y_{33} \end{bmatrix} + \begin{bmatrix} -y_{13} & 0 & y_{13} \\ 0 & 0 & 0 \\ y_{13} & 0 & -y_{13} \end{bmatrix} \tag{3.5.5}$$

Using the Sherman-Morrison-Woodbury (SMW) matrix identity, it is possible to determine the impedance matrix form of this equivalent admittance matrix form. The SMW matrix identity is simply a method for finding the inverse of a matrix when the matrix is updated with a rank *k* update:

$$\left(\mathbf{A}+\mathbf{UCV}\right)^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U}\right)^{-1}\mathbf{VA}^{-1} \tag{3.5.6}$$

Rewriting the above equation, it is possible to arrive at a form similar to the Woodbury identity. Again, assuming a three bus system:

$$\begin{aligned}\tilde{\mathbf{Y}} &= \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & Y_{23} \\ Y_{31} & Y_{32} & Y_{33} \end{bmatrix} + \begin{bmatrix} -y_{13} & 0 & y_{13} \\ 0 & 0 & 0 \\ y_{13} & 0 & -y_{13} \end{bmatrix} \\ &= \mathbf{Y} + \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} -y_{13} & 0 & y_{13} \end{bmatrix} \\ &= \mathbf{Y} + \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} y_{13} \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \\ &= \mathbf{Y} - \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} y_{13} \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}\end{aligned} \tag{3.5.7}$$

Now, we can invert the result obtained above using the SMW identity.

$$\underbrace{\mathbf{Y}}_{\mathbf{Y_{aa}}} - \underbrace{\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}}_{\mathbf{Y_{ab}}} \underbrace{\begin{bmatrix} y_{13} \end{bmatrix}}_{\mathbf{Y_{bb}}} \underbrace{\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}}_{\mathbf{Y_{ab}^T}} \tag{3.5.8}$$

The expression above is made up of four separate pieces which can be rewritten as follows:

$$\left(\mathbf{Y_{aa}} - \mathbf{Y_{ab}Y_{bb}Y_{ab}^T}\right)^{-1} = \mathbf{Y_{aa}^{-1}} - \mathbf{Y_{aa}^{-1}Y_{ab}}\left(\mathbf{Y_{bb}^{-1}} + \mathbf{Y_{ab}^T Y_{aa}^{-1} Y_{ab}}\right)^{-1}\mathbf{Y_{ab}^T Y_{aa}^{-1}} \tag{3.5.9}$$

Equivalently, the original admittance matrix is simply the inverse of the original impedance matrix:

$$\left(\mathbf{Y_{aa}} - \mathbf{Y_{ab}}\mathbf{Y_{bb}}\mathbf{Y_{ab}^{T}}\right)^{-1}$$
$$= \mathbf{X} - \mathbf{X}\begin{bmatrix}1\\0\\-1\end{bmatrix}\left(\left[-y_{13}\right]^{-1} + \begin{bmatrix}1 & 0 & -1\end{bmatrix}\mathbf{X}\begin{bmatrix}1\\0\\-1\end{bmatrix}\right)^{-1}\begin{bmatrix}1 & 0 & -1\end{bmatrix}\mathbf{X} \tag{3.5.10}$$

The second term which is subtracted from the original impedance matrix is made up of three separate pieces. The first piece on the left can be rewritten as follows:

$$\mathbf{X}\begin{bmatrix}1\\0\\-1\end{bmatrix} = \begin{bmatrix}X_{11} - X_{13}\\X_{21} - X_{23}\\X_{31} - X_{33}\end{bmatrix} = \begin{bmatrix}col.\ i\ -\ col.\ j\end{bmatrix} \tag{3.5.11}$$

The rightmost piece is simply the transpose of the leftmost piece and can be rewritten similarly:

$$\begin{bmatrix}1 & 0 & -1\end{bmatrix}\mathbf{X} = \begin{bmatrix}X_{11} - X_{31} & X_{12} - X_{32} & X_{13} - X_{33}\end{bmatrix} = \begin{bmatrix}\quad row\ i\ -\ row\ j\quad\end{bmatrix} \tag{3.5.12}$$

and

$$\begin{aligned}&\left[-y_{13}\right]^{-1} + \begin{bmatrix}1 & 0 & -1\end{bmatrix}\mathbf{X}\begin{bmatrix}1\\0\\-1\end{bmatrix}\\ &= \left[-y_{13}\right]^{-1} + \begin{bmatrix}X_{11} - X_{31} & X_{12} - X_{32} & X_{13} - X_{33}\end{bmatrix}\begin{bmatrix}1\\0\\-1\end{bmatrix}\\ &= \left[-y_{13}\right]^{-1} + \left[X_{11} - X_{31} - \left(X_{13} - X_{33}\right)\right] \leftarrow X_{13} = X_{31}\\ &= x_{13} + X_{11} + X_{33} - 2\left(X_{13}\right)\end{aligned} \tag{3.5.13}$$

Taking the inverse of the result above simply yields a scalar in the case of a single line removal:

$$\left(x_{13}+X_{11}+X_{33}-2\left(X_{13}\right)\right)^{-1}=\frac{1}{x_{13}+X_{11}+X_{33}-2\left(X_{13}\right)} \tag{3.5.14}$$

With each of the three rewritten pieces, it is easy to see how a line impedance is removed from a system impedance matrix:

$$\begin{bmatrix} col.\ i\ -\ col.\ \mathrm{j} \end{bmatrix}\left[\frac{1}{Z_{ii}+Z_{jj}-2Z_{ij}+z_{13}}\right]\begin{bmatrix} \quad row\ \mathrm{i}\ -\ row\ j \quad \end{bmatrix}= \\ =\mathbf{X}\begin{bmatrix}1\\0\\-1\end{bmatrix}\left(\left[-y_{13}\right]^{-1}+\begin{bmatrix}1 & 0 & -1\end{bmatrix}\mathbf{X}\begin{bmatrix}1\\0\\-1\end{bmatrix}\right)^{-1}\begin{bmatrix}1 & 0 & -1\end{bmatrix}\mathbf{X} \tag{3.5.15}$$

Thus, the Kron Reduction which is typically used to add a new loop element to an impedance matrix is nothing more than an application of the SMW matrix identity. The typical form of a Kron Reduction is:

$$\mathbf{K}-\mathbf{L}\mathbf{M}^{-1}\mathbf{L}^{T} \tag{3.5.16}$$

Or

$$\mathbf{X}-\Delta\mathbf{X} \tag{3.5.17}$$

Now, with the ability to model the change in the power system due to a line removal, it is possible to determine analytically the effect of a line outage on bus voltage angles. In the equation above, the term $\Delta\mathbf{X}$ is the representation of the line removal. The question however, is how to isolate this portion so that, when PMU measurements are used, only the change in the impedance matrix is characterized. To accomplish this, the difference in pre and post outage angles must be used.

In order to examine the difference in pre and post outage angles, the power system can be modeled using the DC power flow assumptions. As shown above, the models before and after an outage are:

$$\begin{aligned} \boldsymbol{\theta}_{pre} &= \mathbf{XP} \\ \boldsymbol{\theta}_{post} &= \left[\mathbf{X} + \Delta\mathbf{X}\right]\mathbf{P} \end{aligned} \tag{3.5.18}$$

However, as was shown above, the model after the outage can also be written as:

$$\boldsymbol{\theta}_{post} = \left[\mathbf{X} + \Delta\mathbf{X}\right]\mathbf{P} = \left[\mathbf{X} - \mathbf{L}\mathbf{M}^{-1}\mathbf{L}^{T}\right]\mathbf{P} \tag{3.5.19}$$

To characterize their difference (the impedance change), the post outage angles are subtracted from the pre outage angles:

$$\begin{aligned} \boldsymbol{\theta}_{pre} - \boldsymbol{\theta}_{post} &= \mathbf{XP} - \left[\mathbf{X} - \mathbf{L}\mathbf{M}^{-1}\mathbf{L}^{T}\right]\mathbf{P} \\ &= \mathbf{L}\mathbf{M}^{-1}\mathbf{L}^{T}\mathbf{P} \\ \Delta\boldsymbol{\theta} &= \left[\Delta\mathbf{X}\right]\mathbf{P} \end{aligned} \tag{3.5.20}$$

The vector $\Delta\boldsymbol{\theta}$ is the image of the line outage in terms of bus voltage angles. Here, as before, the assumption is that an angle measurement is available at every bus. Also, as before, this assumption can be relaxed without loss of generality. The vector $\Delta\boldsymbol{\theta}$ can be found off-line, for every line outage since the matrix $\Delta\mathbf{X}$ can be calculated from the topology and the list of branch impedances. The impedance change can be shown to be a rank one matrix as illustrated by the admittance change matrix below:

$$\begin{aligned}\mathbf{\Delta Y} &= \begin{bmatrix} -y_{13} & 0 & y_{13} \\ 0 & 0 & 0 \\ y_{13} & 0 & -y_{13} \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} -y_{13} & 0 & y_{13} \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} -y_{13} \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \\ &= -y_{13}\mathbf{u}\mathbf{u}^T \end{aligned} \tag{3.5.21}$$

By definition, the vector multiplication $\mathbf{u}\mathbf{u}^T$ results in a rank one matrix whose rank does not change by the pre-multiplication by any scalar y_{13} [29]. In general, though, the inverse of a matrix does not necessarily have the same rank of the original matrix. In this case though, the rank of the impedance change matrix can be shown to be of rank one as well. For the removal of a single impedance, the matrix **M** in the Kron Reduction expression is simply a scalar. Therefore:

$$\mathbf{L}m^{-1}\mathbf{L}^T = m^{-1}\mathbf{L}\mathbf{L}^T \tag{3.5.22}$$

Once again, the vector multiplication, by definition results in a rank one matrix and the pre-multiplication by a scalar does not change the rank of a matrix. To address the issue of incomplete observability of a power system, simply examine what happens when one or more rows of the impedance change matrix is removed. This action is equivalent to having a reduced number of PMU measurements, but stacking them in a vector as usual. The result is still a multiplication of two vectors which are pre-multiplied by a scalar

As a linear operator, since the impedance change matrix is of rank one, the number of linearly independent columns is one. Restated, the impedance change matrix

has a range space made up of vectors which are scaled versions of one vector, the linearly independent column of the impedance change matrix. This means that if the scaling is removed from the vector $\Delta\boldsymbol{\theta}$ then, simply, every line outage can be represented by one single vector. Mathematically speaking, the range of the impedance change matrix has one linearly independent basis vector. It is always possible to find this linearly independent vector and simply force the length of the vector to one, thereby removing scaling from the vector. The denominator in the equation below is the 2-norm of the angle change vector.

$$\Delta\boldsymbol{\theta}_{\mathbf{norm}} = \frac{\Delta\boldsymbol{\theta}}{\left\|\Delta\boldsymbol{\theta}\right\|_2} \tag{3.5.23}$$

If the measured $\Delta\boldsymbol{\theta}$ vector is also normalized to rank one, the vectors will be identical, assuming the DC power flow assumptions. Thus, this technique should produce identical results to the OT method, but without the requirement that every line in the power system have a line flow measurement available. The problem then, is how to deal with the inaccuracies involved with the DC power flow assumptions.

The figures below show the complete line outage detection algorithm including both off-line and on-line analyses.

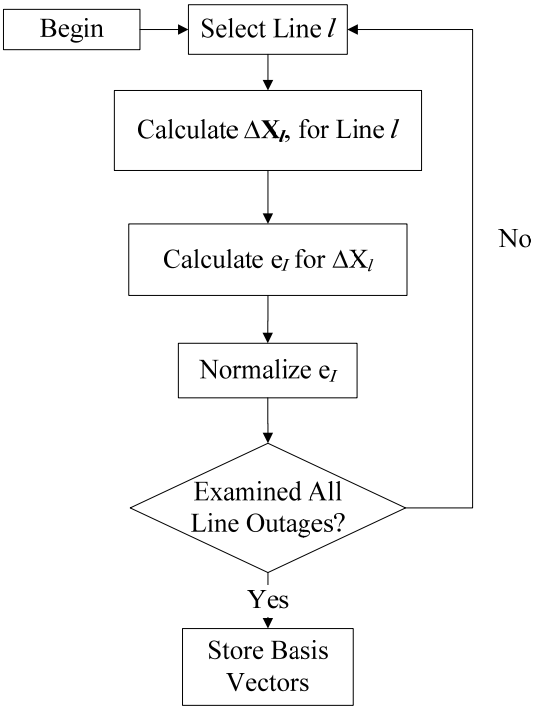


Figure 3.3 – Program flow for off-line part of line outage detection.

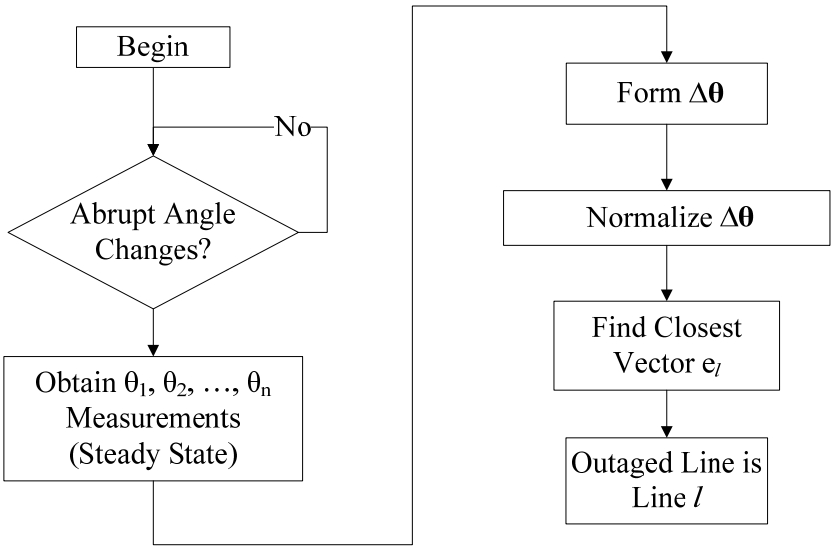


Figure 3.4 – Program flow for on-line part of line outage detection.

3.6 Novel Method Example

To show the utility of the novel method of line outage detection, the method will be used for a small test system. The six bus system from [16] will again be used. This system was chosen since it contains only six buses making it feasible to show all matrix calculations. Plus, the system contains lines which can easily be detected as being outaged as well as lines which are not so easily detected under certain conditions. Below are the **Y** matrix and its inverse, the **X** matrix as used in DC power flow.

$$\mathbf{Y} = \begin{bmatrix} 13.33 & -5.00 & 0.00 & -5.00 & -3.33 & 0.00 \\ -5.00 & 27.33 & -4.00 & -10.00 & -3.33 & -5.00 \\ 0.00 & -4.00 & 17.85 & 0.00 & -3.85 & -10.00 \\ -5.00 & -10.00 & 0.00 & 17.50 & -2.50 & 0.00 \\ -3.33 & -3.33 & -3.85 & -2.50 & 16.35 & -3.33 \\ 0.00 & -5.00 & -10.00 & 0.00 & -3.33 & 18.33 \end{bmatrix} \tag{3.6.1}$$

Note, that for the **X** matrix, only 5 buses are shown with nonzero elements since bus 1 is used as a reference and always assumed to have an angle of zero. Therefore, it will suffice to show only the 5 nonzero buses.

$$X = \begin{array}{c} \\ ① \\ ② \\ ③ \\ ④ \\ ⑤ \\ ⑥ \end{array} \begin{array}{c} \begin{array}{cccccc} ① & ② & ③ & ④ & ⑤ & ⑥ \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0941 & 0.0805 & 0.0630 & 0.0643 & 0.0813 \\ 0 & 0.0805 & 0.1659 & 0.0590 & 0.0908 & 0.1290 \\ 0 & 0.0630 & 0.0590 & 0.1009 & 0.0542 & 0.0592 \\ 0 & 0.0643 & 0.0908 & 0.0542 & 0.1222 & 0.0893 \\ 0 & 0.0813 & 0.1290 & 0.0592 & 0.0893 & 0.1633 \end{bmatrix} \end{array} \tag{3.6.2}$$

Therefore, in steady state, before an outage occurs the bus voltage angles are found to be:

$$\boldsymbol{\theta}_{\mathbf{pre}} = \begin{bmatrix} -2.9024 \\ -3.1679 \\ -4.7632 \\ -5.6902 \\ -5.7418 \end{bmatrix} \tag{3.6.3}$$

Where the injected powers, **P** are initially:

$$\mathbf{P} = \begin{bmatrix} 0.5 \\ 0.6 \\ -0.7 \\ -0.7 \\ -0.7 \end{bmatrix} \tag{3.6.4}$$

Now, each line removal in the system can be modeled as follows. For line 1, from bus 1 to bus 2, with a series reactance of 0.2 p.u., the $\Delta\mathbf{X}$ matrix can be formed:

$$\begin{aligned} \Delta\mathbf{X} &= \begin{bmatrix} \\ col.\ 1 - col.\ 2 \\ \\ \end{bmatrix} \begin{bmatrix} \dfrac{1}{X_{11} + X_{22} - 2X_{12} - z_{12}} \end{bmatrix} \begin{bmatrix} \quad row\ 1 - row\ 2 \quad \end{bmatrix} \\ &= \begin{bmatrix} -0.0941 \\ -0.0805 \\ -0.0630 \\ -0.0643 \\ -0.0813 \end{bmatrix} \begin{bmatrix} \dfrac{1}{0 + 0.0941 - 0 - 0.2} \end{bmatrix} \begin{bmatrix} \quad row\ 1 - row\ 2 \quad \end{bmatrix} \\ &= \begin{bmatrix} -0.0837 & -0.0716 & -0.0560 & -0.0572 & -0.0723 \\ -0.0716 & -0.0612 & -0.0479 & -0.0489 & -0.0618 \\ -0.0560 & -0.0479 & -0.0375 & -0.0383 & -0.0484 \\ -0.0572 & -0.0489 & -0.0383 & -0.0391 & -0.0494 \\ -0.0723 & -0.0618 & -0.0484 & -0.0494 & -0.0624 \end{bmatrix} \end{aligned} \tag{3.6.5}$$

The new system impedance matrix after line removal is $\left[\mathbf{X} - \Delta\mathbf{X}\right]$, so it is equivalent to negate each element and take the sum instead. As a linear transformation, $\Delta\mathbf{X}$ has a rank

of unity as previously shown, so we must find a single vector to serve as a basis for the range space. Using elementary row operations, the transpose of the matrix can be rewritten in reduced row echelon form. The single non-zero row indicates that the matrix has a rank of unity.

$$rref\left(\Delta\mathbf{X}^T\right) = \begin{bmatrix} 1.0000 & 0.8554 & 0.6691 & 0.6836 & 0.8636 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \tag{3.6.6}$$

The single basis vector e_l for the range of $\Delta\mathbf{X}$ is then simply the first row of the reduced row echelon form.

$$e_l = \begin{bmatrix} 1 \\ 0.8554 \\ 0.6691 \\ 0.6836 \\ 0.8636 \end{bmatrix} \tag{3.6.7}$$

However, in order to remove any effect of scaling, the vector must be normalized to length one where the Euclidean norm is:

$$\begin{aligned} \left\|e_l\right\|_2 &= \sqrt{1^2 + \left(0.8554\right)^2 + \left(0.6691\right)^2 + \left(0.6836\right)^2 + \left(0.8636\right)^2} \\ &= 1.8419 \end{aligned} \tag{3.6.8}$$

$$e_{l,norm} = \left(\frac{1}{1.8419}\right) \cdot \begin{bmatrix} 1 \\ 0.8554 \\ 0.6691 \\ 0.6836 \\ 0.8636 \end{bmatrix} = \begin{bmatrix} 0.5429 \\ 0.4644 \\ 0.3633 \\ 0.3712 \\ 0.4689 \end{bmatrix} \tag{3.6.9}$$

Note that the formation of this vector assumes that PMU angle measurements will be available at every bus in the system. Continuing with this assumption, the basis vector for every line outage in the system can be found.

Table 3.1 – Basis vectors, e_l for line outages 1 through 6.

Line / Bus	1	2	3	4	5	6
2	0.5429	0.4060	0.3292	-0.1331	0.5295	-0.4452
3	0.4644	0.3802	0.4644	0.8348	0.3661	0.1533
4	0.3633	0.6503	0.2774	-0.0391	-0.6442	-0.1309
5	0.3712	0.3495	0.6249	0.2583	0.1721	0.8642
6	0.4689	0.3816	0.4567	0.4660	0.3754	0.1194

Table 3.2 – Basis vectors, e_l for line outages 7 through 11.

Line / Bus	7	8	9	10	11
2	-0.1292	0.1755	0.0153	0.0147	0.1856
3	0.4876	0.8155	-0.7322	0.3407	0.4184
4	-0.0380	0.0516	0.0045	-0.5000	0.0546
5	0.2509	-0.3406	-0.0297	0.7280	-0.3603
6	0.8253	0.4307	0.6803	0.3222	0.8110

Continuing with the example for line 1 outaged, the angles in degrees after the outage can be calculated as follows:

$$\boldsymbol{\theta}_{\mathbf{post}} = \begin{bmatrix} -5.4827 \\ -5.3751 \\ -6.4897 \\ -7.4542 \\ -7.9701 \end{bmatrix} \tag{3.6.10}$$

Assuming that the angles before the outage and the angles after the outage were measured without error and the DC power flow assumptions are used, the angular difference can be found.

$$\Delta\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathbf{pre}} - \boldsymbol{\theta}_{\mathbf{post}} = \begin{bmatrix} -2.9024 \\ -3.1679 \\ -4.7632 \\ -5.6902 \\ -5.7418 \end{bmatrix} - \begin{bmatrix} -5.4827 \\ -5.3751 \\ -6.4897 \\ -7.4542 \\ -7.9701 \end{bmatrix} = \begin{bmatrix} 2.5803 \\ 2.2071 \\ 1.7264 \\ 1.7640 \\ 2.2283 \end{bmatrix} \tag{3.6.11}$$

Now, the scaling must be removed by normalizing this angular difference to a length of one.

$$\left\|\Delta\boldsymbol{\theta}\right\|_2 = \sqrt{2.5803^2 + 2.2071^2 + 1.7264^2 + 1.7640^2 + 2.2283^2} = 4.7526 \tag{3.6.12}$$

$$\Delta\boldsymbol{\theta}_{norm} = \left(\frac{1}{4.7526}\right) \cdot \begin{bmatrix} 2.5803 \\ 2.2071 \\ 1.7264 \\ 1.7640 \\ 2.2283 \end{bmatrix} = \begin{bmatrix} 0.5429 \\ 0.4644 \\ 0.3633 \\ 0.3712 \\ 0.4689 \end{bmatrix} \tag{3.6.13}$$

Clearly, this vector is what would be expected since it represents a vector in the range space of $\Delta\mathbf{X}$ and by definition any vector in the range of a linear transformation is a linear combination of the basis vectors. Removal of scaling essentially recovers the basis vector. Additionally, due to the formation of the $\Delta\mathbf{X}$ matrix, the assumption that a PMU measurement is available at every bus can be relaxed.

Now examine what happens when fewer buses are monitored. The optimal PMU locations for the 6-bus system under question are at buses 3 and 6. These were found using integer linear programming in MATLAB with the `bintprog` command. The angles before and after an outage will be identical in this case, but only 2 of the 6 measurements will be available. However, the calculation of $\Delta\mathbf{X}$ and the basis vector of the linear transformation will be slightly different. Instead of the transformation being a

6x6 matrix, it becomes a 2x6 matrix where the row numbers correspond to the buses being measured. For line 1, the transformation is:

$$-\mathbf{\Delta X} = \begin{bmatrix} 0.0716 & 0.0612 & 0.0479 & 0.0489 & 0.0618 \\ 0.0723 & 0.0618 & 0.0484 & 0.0494 & 0.0624 \end{bmatrix} \tag{3.6.14}$$

Again, we find the reduced row echelon form of $\Delta\mathbf{X}^{\mathrm{T}}$

$$rref(\mathbf{\Delta X}^T) = \begin{bmatrix} 1 & 1.0096 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.6.15}$$

And normalize the basis vector

$$\|e_l\|_2 = \sqrt{1^2 + 1.0096^2} = 1.4210 \tag{3.6.16}$$

$$e_{l,norm} = \left(\frac{1}{1.4210}\right) \cdot \begin{bmatrix} 1 \\ 1.0096 \end{bmatrix} = \begin{bmatrix} 0.7040 \\ 0.7105 \end{bmatrix} \tag{3.6.17}$$

Once again, these basis vectors can be formed for all such line outages in the system:

Table 3.3 – e_l for line outages 1 through 6 with PMUs at bus 3, bus 6.

		Lines										
		1	2	3	4	5	6	7	8	9	10	11
Buses	3	0.704	0.706	-0.713	-0.873	0.698	-0.789	0.509	-0.884	-0.733	-0.727	0.458
	6	0.710	0.708	-0.701	-0.487	0.716	-0.614	0.861	-0.467	0.681	-0.687	0.889

Using the same angles, the pre and post outage difference can be found:

$$\mathbf{\Delta\theta} = \mathbf{\theta_{pre}} - \mathbf{\theta_{post}} = \begin{bmatrix} -2.9024 \\ \mathbf{-3.1679} \\ -4.7632 \\ -5.6902 \\ \mathbf{-5.7418} \end{bmatrix} - \begin{bmatrix} -5.4827 \\ \mathbf{-5.3751} \\ -6.4897 \\ -7.4542 \\ \mathbf{-7.9701} \end{bmatrix} = \begin{bmatrix} 2.5803 \\ \mathbf{2.2071} \\ 1.7264 \\ 1.7640 \\ \mathbf{2.2283} \end{bmatrix} \rightarrow \begin{bmatrix} 2.2071 \\ 2.2283 \end{bmatrix} \tag{3.6.18}$$

And once again, the vector is normalized:

$$\left\|\Delta\boldsymbol{\theta}\right\|_2 = \sqrt{2.2071^2 + 2.2283^2} = 3.1363 \tag{3.6.19}$$

$$\Delta\boldsymbol{\theta}_{norm} = \left(\frac{1}{3.1363}\right)\cdot\begin{bmatrix}2.2071\\2.2283\end{bmatrix} = \begin{bmatrix}0.704\\0.710\end{bmatrix} \tag{3.6.20}$$

Similar to the case when a measurement is available at every bus, the vector obtained from angle measurements matches exactly with the vector obtained through calculation. It is important to note that these results rely on a few assumptions. The main assumptions are the DC power flow assumptions. These assumptions only hold true when the resistance in the lines is much smaller than the impedance which is not the case in distribution systems. Plus, it is assumed that a PMU can measure with perfect accuracy. As will be shown in Chapter 4, these assumptions can be relaxed without a significant increase in the detection error.

CHAPTER FOUR

LINE OUTAGE METHOD COMPARISON

4.1 System and Simulation Description

Throughout this chapter, the line outage detection method presented by Overbye and Tate (OT) [14] and [15] will be compared to the new method described in 0. In order to compare the two methods, MATLAB simulations were performed for each method using a test system. All of the MATLAB code can be found in Appendix B and Appendix C. The test system is a reduced equivalent of a portion of the Tennessee Valley Authority (TVA) 500kV transmission system. A one-line diagram of the test system is shown below in Figure 4.1. PMUs are installed at buses 3, 5, 6, 16, 26, 45 shown on Figure 4.1 using arrows. First, the OT method will be used to detect all line outages under a specific loading condition. Next, the proposed method will be used to detect all line outages under the same loading conditions.

Before moving to the results, it is important to note the physical location of the PMUs in the figure below. Buses 3, 5, and 6 are adjacent, so it seems unnecessary to place a PMU at each of these buses. The reason three PMUs are installed instead of one is that this is a reduced system. Thus, the representation in the figure below shows the buses being adjacent, but in truth many more buses and lines separate these adjacent buses. The equivalent lines between buses 3,5, and 6 merely represent the complex structure between the buses in a more simple fashion.

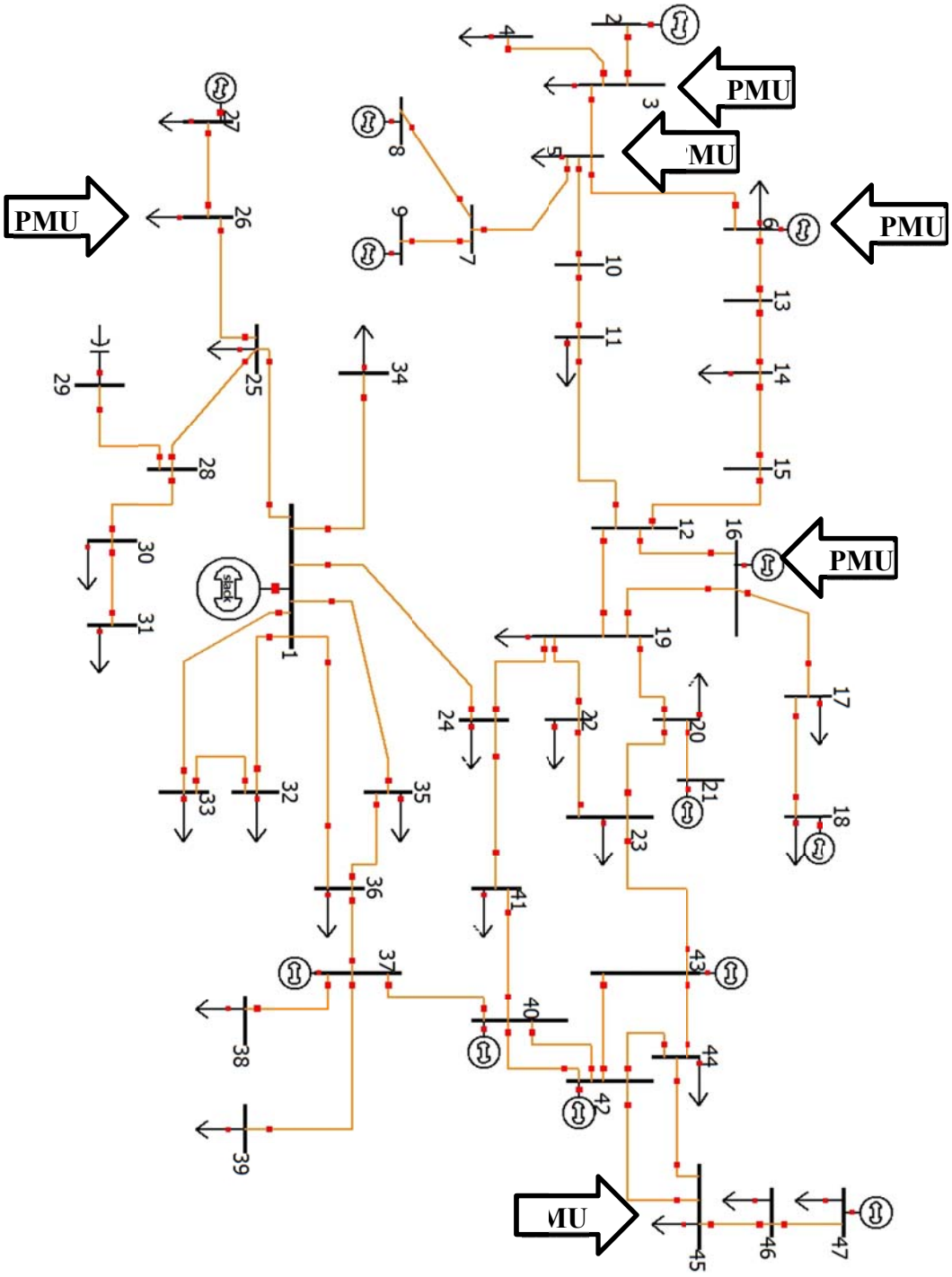


Figure 4.1 – One line diagram of 47 bus reduced TVA system.

For each case, three sets of simulations were performed. The first simulation uses the actual PMU locations. The second simulation uses the integer programming approach to determine the optimal number and location of PMUs. Finally, the last simulation assumes that a PMU measurement is available at every bus. In each case, the system will be simulated in MATLAB. The angles before and after each line outage will be simulated as ideally measured but with a Gaussian noise signal added after the power flow solution. In general, each of the three simulations consists of a simulated off-line portion and a simulated on-line portion. During the off-line portions, prototypical values are found for the line outages. During the on-line portions, measured values are simulated for line outages. A simulated measurement is created by scaling all the loads in the system using a Gaussian random number generator with a mean of 1 and standard deviation of 0.2. The randomly scaled loads can then be anywhere from 80% to %120 of their original base value. This original value was used to create the prototypical values.

4.2 OT Method

As described in section 3.4, the first step in the Overbye and Tate method of line outage detection is the calculation of $\Delta\theta_{calc,l}^{\tilde{P}_l}$. This step requires calculation of the PTDFs for every line or every desired line regardless of the number of available PMU measurements. The PTDFs were calculated using MATPOWER. A sample of the full PTDF matrix is shown below showing the buses were PMUs are currently installed. For readability, only the lines with the ten highest MW flows in the base case were examined. It is entirely feasible to examine every line in the system, but the most highly loaded lines are the most important when outaged.

Table 4.1 – PTDF at 'from' bus for ten highest loaded lines.

Line	PTDF at From Bus
1	0.6926
2	0.6330
3	0.4127
4	0.7937
5	0.1602
6	0.0693
7	0.8510
8	0.8157
9	0.1220
10	0.0587

Once the desired PTDFs are found, the relevant impedance information is taken from the DC power flow **X** matrix via the PMU selection matrix **K**. In this case, **K** has four rows and the number of columns is equal to the number of buses. The **K** matrix is post multiplied by the **X** matrix then scaled by $\tilde{P}_l$ to form the calculated vector of angle changes, $\Delta\theta_{calc,l}^{\tilde{P}_l}$.

Table 4.2 shows the angle change at the buses where PMUs are installed for the top ten line outages. Using only the top ten lines, however is realistic in that their outage will create the most drastic change in power flow. Tate [8] describes which lines in a system can and cannot be detected using line outage detection. For instance, double circuit lines, or radial lines connected to boundary buses are either indistinguishable or entirely undetectable. Further analysis is given in [8]. Note that the first row in the table below is zero since the PMU is installed at the reference bus and all other angle changes are recorded in reference to this bus angle.

Table 4.2 – Calculated angle change at PMU buses due to top ten line outages

PMU	Line 1 Outage	Line 2 Outage	Line 3 Outage	Line 4 Outage	Line 5 Outage	Line 6 Outage	Line 7 Outage	Line 8 Outage	Line 9 Outage	Line 10 Outage
1	21.059	0.5244	3.3956	15.342	-0.097	-1.0061	9.8338	14.8375	0.7705	0.3933
2	21.059	0.5244	3.3956	15.342	-0.097	-1.0061	9.8338	14.8375	0.7705	0.3933
3	21.059	0.5244	3.3956	15.342	-0.097	-1.0061	11.5628	12.9534	0.7705	0.3933
4	21.059	0.5244	2.8471	15.342	0.097	-1.0061	0.0000	0.0000	0.7705	0.3933
5	0.000	0.0000	0.0000	0.000	0.000	0.0000	0.0000	0.0000	0.0000	0.0000
6	14.387	-0.6368	0.0000	4.843	2.313	-1.6149	0.0000	0.0000	1.1544	1.7043

With the data in the table above, the top ten line outages can be identified, once they are detected. Chapter 5 will describe how the detection process is carried out. For this example however, it is assumed that a line outage can be accurately detected. In order to illustrate the O&T algorithm, each of the top ten loaded lines was removed and the resulting bus voltage angles were found. The angular differences are shown below in Table 4.3.

Table 4.3 – Simulations of angle changes due to outages.

PMU	Line 1	Line 2	Line 3	Line 4	Line 5	Line 6	Line 7	Line 8	Line 9	Line 10
1	11.581	0.3942	4.7696	18.9610	-0.2319	-2.9945	5.5689	8.2073	0.8041	0.3336
2	11.581	0.3941	4.7676	18.9605	-0.2319	-2.9945	5.5672	8.1988	0.8041	0.3336
3	11.582	0.3943	4.7729	18.9617	-0.2319	-2.9944	6.5286	7.1703	0.8041	0.3336
4	11.562	0.3889	-3.6033	18.9321	-0.2318	-2.9958	0.0684	0.0861	0.8035	0.3334
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	7.7303	-0.4066	0.0438	5.8523	6.6119	-4.8646	0.0298	0.0410	1.2132	1.3956

Next, for each column in Table 4.3, the normalized angle distance (NAD) was found between the angles in all columns of

Table 4.2. Each column in the table below represents the ten line outages from simulated measurements and each row represents the ten line outages from calculation using the O&T method. The smallest NAD in each column is bordered in bold.

Table 4.4 – NAD between measurements (columns) and calculations (rows).

	Line 1	Line 2	Line 3	Line 4	Line 5	Line 6	Line 7	Line 8	Line 9	Line 10
Line 1	0.0070	0.7850	0.9342	0.1758	1.2196	0.3511	0.5980	0.5968	0.3159	0.7749
Line 2	0.8409	0.0683	0.9948	0.6847	0.9192	1.1521	0.7219	0.7212	1.1227	1.3419
Line 3	0.9661	0.9969	0.0399	0.9428	1.3861	1.0622	0.4594	0.4570	1.0502	1.2310
Line 4	0.1655	0.6235	0.9084	0.0034	1.3517	0.5196	0.5354	0.5341	0.4850	0.9310
Line 5	1.2356	0.9676	1.3823	1.3640	0.0133	0.9342	1.3644	1.3644	0.9656	0.5231
Line 6	0.3522	1.0909	1.0332	0.5172	0.9276	0.0058	0.8027	0.8020	0.0299	0.4448
Line 7	0.6012	0.6792	0.4151	0.5411	1.3708	0.8123	0.0074	0.1401	0.7882	1.1203
Line 8	0.5997	0.6779	0.4128	0.5394	1.3708	0.8113	0.1393	0.0070	0.7873	1.1200
Line 9	0.3193	1.0628	1.0215	0.4849	0.9570	0.0392	0.7804	0.7796	0.0035	0.4773
Line 10	0.7937	1.3819	1.2220	0.9458	0.4972	0.4523	1.1262	1.1258	0.4871	0.0135

From the results above, each of the ten line outages was correctly identified when taken out of service. It is important to note that all of the smallest NADs are less than 0.1. If a system contains many lines which are unable to be correctly identified this NAD level can be used as a threshold.

4.3 Proposed Method

As with the OT method, the proposed method begins with off-line calculation of angle changes. Once again, the top ten line outages were examined assuming the actual PMU locations. Calculation of the angle changes can be accomplished in many ways. In this case, however, the singular value decomposition of the impedance change matrix $\mathbf{\Delta\theta}$ was found. The SVD attempts to find optimal orthonormal bases for both the null space and range space. Therefore, the orthonormal bases for the range space for each line outage are simply the corresponding vector e_l.

Table 4.5 – Calculated angle changes using the proposed method.

PMU	Line 1 Outage	Line 2 Outage	Line 3 Outage	Line 4 Outage	Line 5 Outage	Line 6 Outage	Line 7 Outage	Line 8 Outage	Line 9 Outage	Line 10 Outage
1	0.4732	0.4274	0.5197	0.4939	-0.0416	-0.3899	0.5437	0.6017	0.4002	0.2095
2	0.4732	0.4274	0.5197	0.4939	-0.0416	-0.3899	0.5437	0.6017	0.4002	0.2095
3	0.4732	0.4274	0.5197	0.4939	-0.0416	-0.3899	0.6393	0.5253	0.4002	0.2095
4	0.4732	0.4274	-0.4357	0.4939	-0.0416	-0.3899	0.0000	0.0000	0.4002	0.2095
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.3233	-0.5190	0.0000	0.1559	0.9965	-0.6259	0.0000	0.0000	0.5996	0.9080

To test the efficacy of the proposed algorithm and compare it with the OT method, the ten lines above were removed and measurements were simulated. Table 4.6 shows the actual values from the simulated line angle change measurements. Table 4.7 below shows the same information as Table 4.6, but the columns have been normalized to a length of one.

Table 4.6 – Simulated measurements of angle changes.

PMU	Line 1	Line 2	Line 3	Line 4	Line 5	Line 6	Line 7	Line 8	Line 9	Line 10
1	11.5812	0.3942	4.7696	18.9610	-0.2319	-2.9945	5.5689	8.2073	0.8041	0.3336
2	11.5809	0.3941	4.7676	18.9605	-0.2319	-2.9945	5.5672	8.1988	0.8041	0.3336
3	11.5817	0.3943	4.7729	18.9617	-0.2319	-2.9944	6.5286	7.1703	0.8041	0.3336
4	11.5624	0.3889	-3.6033	18.9321	-0.2318	-2.9958	0.0684	0.0861	0.8035	0.3334
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	7.7303	-0.4066	0.0438	5.8523	6.6119	-4.8646	0.0298	0.0410	1.2132	1.3956

Table 4.7 – Normalized version of Table 4.6.

PMU	Line 1	Line 2	Line 3	Line 4	Line 5	Line 6	Line 7	Line 8	Line 9	Line 10
1	0.4745	0.4455	0.5292	0.4943	-0.0350	-0.3881	0.5444	0.6018	0.3992	0.2157
2	0.4744	0.4454	0.5289	0.4943	-0.0350	-0.3881	0.5443	0.6012	0.3992	0.2157
3	0.4745	0.4457	0.5295	0.4944	-0.0350	-0.3881	0.6382	0.5258	0.3992	0.2157
4	0.4737	0.4396	-0.3998	0.4936	-0.0350	-0.3883	0.0067	0.0063	0.3989	0.2155
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.3167	-0.4596	0.0049	0.1526	0.9975	-0.6304	0.0029	0.0030	0.6023	0.9022

Next, the distance of the absolute value between each column of Table 4.5 and Table 4.7 was found. The resulting distances are shown in Table 4.8. In each column, the shortest distance has been bordered in bold. Similar to the OT method, each line was

correctly identified as being taken out of service. Also, similar to the OT method, the distances are all less than 0.01.

Table 4.8 – Distances between each column of Table 4.5 and Table 4.7.

	Line 1	Line 2	Line 3	Line 4	Line 5	Line 6	Line 7	Line 8	Line 9	Line 10
Line 1	0.0070	0.2230	0.3284	0.1655	1.1004	0.3522	0.6012	0.5997	0.3193	0.7938
Line 2	0.1483	0.0683	0.4772	0.3197	0.9676	0.1985	0.6792	0.6779	0.1653	0.6490
Line 3	0.3409	0.5442	0.0398	0.1882	1.3509	0.6663	0.4151	0.4128	0.6353	1.0763
Line 4	0.1758	0.3900	0.1690	0.0034	1.2375	0.5172	0.5411	0.5394	0.4849	0.9459
Line 5	1.1058	0.9192	1.3640	1.2453	0.0133	0.8012	1.3708	1.3708	0.8318	0.3603
Line 6	0.3511	0.1363	0.6721	0.5196	0.7838	0.0058	0.8122	0.8113	0.0392	0.4524
Line 7	0.5980	0.7177	0.4465	0.5354	1.3601	0.8027	0.0074	0.1392	0.7804	1.1262
Line 8	0.5969	0.7168	0.4447	0.5341	1.3600	0.8019	0.1401	0.0070	0.7797	1.1259
Line 9	0.3159	0.1007	0.6385	0.4850	0.8165	0.0300	0.7882	0.7873	0.0035	0.4871
Line 10	0.7749	0.5712	1.0676	0.9310	0.3606	0.4448	1.1203	1.1200	0.4773	0.0136

4.4 Comparison

Each of the line outage detection algorithms described rely upon the DC power flow assumptions. In addition, both attempt to model the effect of a line outage on the change in one or more bus voltage angles. Thus, each algorithm should perform equally well in terms of line outage identification success. In fact, under further examination, the two algorithms produce identical results. The table below illustrates the similarities between the two algorithms.

Table 4.9 – Algorithm comparison.

Requirements	**OT Algorithm**	**Proposed Algorithm**
DC Power Flow Assumptions	Yes	Yes
Synchrophasor Angles	Yes	Yes
Line Flow Measurements	Yes	No
PTDF Matrix	Yes	No
Impedance Change Matrix	No	Yes
System Impedance Matrix	Yes	Yes
Nearest Neighbor Search	Yes	Yes

The following explanation shows why the proposed algorithm for line outage identification produces identical results as the OT algorithm without the requirement of line flow measurements. As a step in calculating the NAD in the OT method both the calculated angle changes and the measured angle changes are divided by their lengths. In essence, the step is the same as normalizing the PMU measurement vectors to unit length. In addition, the division by length step removes all scaling due to bus injections. Stated differently, only the relationship between the line flow measurements remains as opposed to the absolute magnitude of flow. This does not indicate, however, that the OT does not require line flow measurements. Line flow measurements are required to calculate $\Delta\theta_{calc,l}^{\tilde{P}_l}$ for each outage. However, if each $\Delta\theta_{calc,l}^{\tilde{P}_l}$ is normalized, the result is the same as if the angle changes were calculated from the proposed method. The normalized version of

Table 4.2 is shown below in Table 4.10. These results are not only close, but identical to those for the proposed method.

Table 4.10 – Normalized version of

Table 4.2.

PMU	Line 1 Outage	Line 2 Outage	Line 3 Outage	Line 4 Outage	Line 5 Outage	Line 6 Outage	Line 7 Outage	Line 8 Outage	Line 9 Outage	Line 10 Outage
1	0.4732	0.4274	0.5197	0.4939	-0.0416	-0.3899	0.5437	0.6017	0.4002	0.2095
2	0.4732	0.4274	0.5197	0.4939	-0.0416	-0.3899	0.5437	0.6017	0.4002	0.2095
3	0.4732	0.4274	0.5197	0.4939	-0.0416	-0.3899	0.6393	0.5253	0.4002	0.2095
4	0.4732	0.4274	-0.4357	0.4939	-0.0416	-0.3899	0.0000	0.0000	0.4002	0.2095
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.3233	-0.5190	0.0000	0.1559	0.9965	-0.6259	0.0000	0.0000	0.5996	0.9080

Although not investigated in this thesis, it is possible that a more robust algorithm for line outage detection could be created using a combination of the OT method and the proposed method. For instance, the line flow measurements and bus injection measurements may be used to augment the distances calculated between measured and calculated angle changes. To further illustrate the utility of the proposed line outage

detection algorithm, a dynamic simulation of the 47 bus TVA system above was constructed in the following chapter.

CHAPTER FIVE

DYNAMIC SIMULATIONS

5.1 Dynamic Simulation Description

Identifying a line outage in a real system relies heavily on the assumption that the line outage event can first be detected. The primary focus of this thesis is on the line outage identification methods, but Overbye and Tate explore detection in greater depth [14], so their findings will be utilized in this thesis. Section 5.2 will cover the findings of Overbye and Tate so that their results may be utilized in the sections to follow. A dynamic simulation constructed in PSS/E v32 is then presented using the same 47 bus system, but with generator dynamical models. Line outages are examined based on their detectability and their likelihood of correct identification. Before concluding, the proposed method is utilized on PMU data from the full, non-reduced system which consists of 6000+ buses.

5.2 Detection of a Possible Outage

Any type of event detection algorithm generally consists of two parts. The first is the detection of the event and the second is the identification of the event. For the purposes of line outage detection PMU measurements may be constantly monitored for an abrupt change in the bus voltage angle. Although the idea is simple, in practice such a technique is complicated by noise and non-outage events like capacitor switching. The task then, is to decide on the criteria which separate line outages from everything else.

Before any processing can be performed, the PMU angles must first be filtered to remove erroneous high frequency content. By applying a low pass filter to the PMU

signals, events such as momentary lapses in communication and noise can be removed to prevent false indication of power system events. While it may seem counter intuitive to remove the high frequency content from the signal, line outage detection only requires that a change from one angular value to another be seen. Much research has been completed on the topic of filtering PMU signals. For line outage detection Overbye and Tate propose an order 61 FIR filter with a Hamming window and a cutoff frequency of 0.1Hz. By design, the electromechanical oscillations are kept below 0.1Hz and the cutoff frequency was chosen for this reason. Once the PMU angles are filtered, the process of detecting abrupt changes then begins.

Abrupt changes in a signal can also be thought of as edges, visually the change looks like the edge of a cliff. This analogy is utilized in image processing to detect edges in an image. By examining the first derivative of the intensity values of pixels, a very primitive edge detection system can be constructed. The same applies to any signal since we know that the derivate can be thought of as the slope of a line tangent to the curve or signal. Thus, when small numbers result from the derivative, not much change is happening in the signal. When large numbers are encountered, drastic or possibly abrupt changes are occurring in the signal. All of this description is qualitative however, so the terms “small” and “large” in terms of the derivative must be defined. A threshold, τ is used to mark the distinction between what is small and what is large.

For real signals, the analytical derivative cannot be applied since a signal is actually a discrete sampled version of the continuous one. The typical form of an analytical derivative is shown below which is simply a representation of the slope of a

line tangent to the function *f* at a point *a*. The term *h* is simply the distance between the two points which are evaluated in the numerator. As *h* approaches zero in the limit, the equation below becomes the exact analytical form of a derivative.

$$f'(a) = \frac{f(a+h) - f(a)}{h} \tag{5.2.1}$$

In the discrete time version, *h* can never truly reach zero but can be as small as one sample. Therefore, a discrete approximation to the equation above is shown below.

$$f'[a] = f[a+1] - f[a] \tag{5.2.2}$$

Or, more simply, the discrete approximation to a derivative can be found by taking the difference between successive samples. Equation (5.2.2) requires future knowledge of the signal however, so the equivalent equation below can be utilized:

$$f'[a-1] = f[a] - f[a-1] \tag{5.2.3}$$

Some caveats arise with this expression though. For instance, when a quick disturbance occurs such that the sampling rate of the PMU is too low, the event may be missed. In addition, it is necessary to determine not only when the event starts, but when the event also ends. Any major change in the power system topology will create a transient condition which diminishes after some time. Therefore, an indication of the end of the event may simply occur when the angular difference between two consecutive samples reaches zero, or very near zero. The figure below illustrates this method.

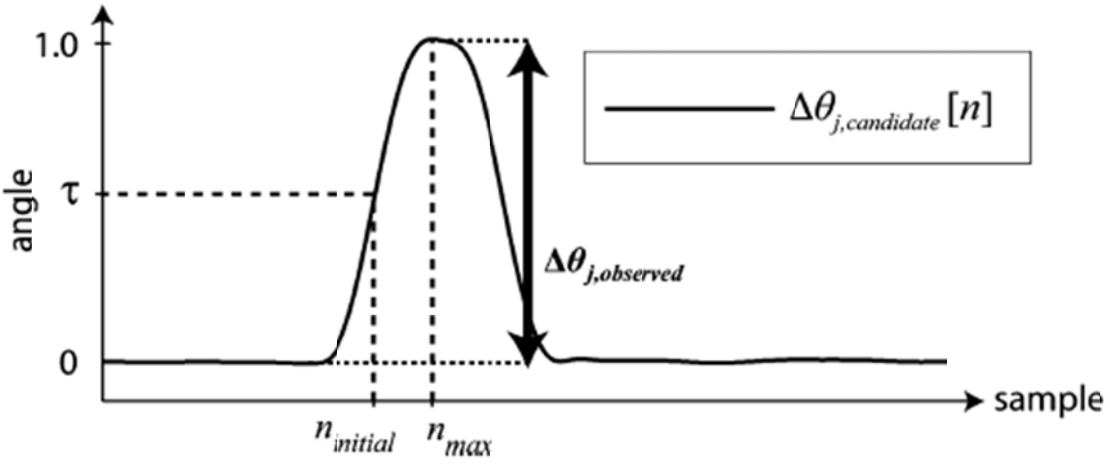


Figure 5.1 – Locating the event in time [14].

Another parameter γ must be introduced here to prevent early indication of steady state values. Many small decreases in the angular difference will occur before the overall decrease in angular difference is found. The parameter, γ, is simply a threshold to determine where, in time, the event actually ends. For the purpose of the following simulations, γ is set equal to τ. When the maximum angular difference is determined and the peak of the event occurs, this sample is labeled n_{max} and is used to correlate the event across all relevant PMUs.

Once indication of a possible line outage is triggered by the derivative approximation, the task is to determine the final steady state value of all bus voltage angles. This requires a new parameter *N* to determine when the PMU measurement has settled from its post outage, transient condition, to its steady state value. The figure below shows how *N* can be chosen so that steady state angles are used for the post outage angle measurements.

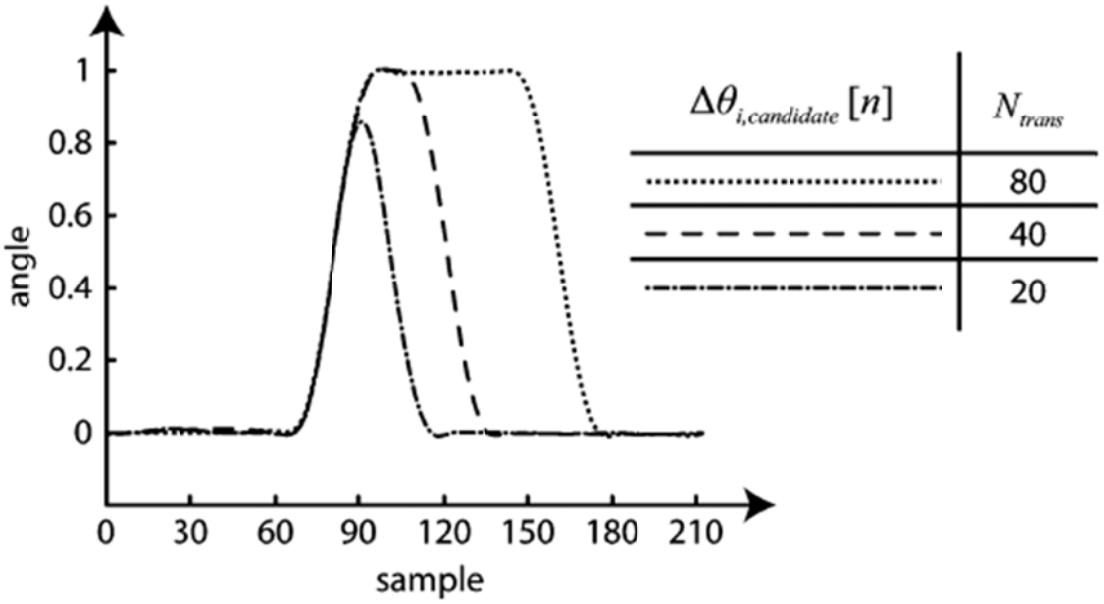


Figure 5.2 – Selection of the N parameter [14].

The N parameter will depend heavily on the degree of damping of the post-event transients. If N is too small, the angular difference calculated may not have reached steady state values. If N is too large, the angular difference calculated will take longer to process and the final steady state values of angles could represent changes in the system other than the line outage alone.

5.3 PSS/E Simulation

The reduced 47 bus system was modeled in PSS/E v32 using a simulation time step of 0.008333 or two samples per cycle. As in 0, each of ten critical lines was sequentially removed. Next, the PMU bus voltage angle measurements were simulated using the waveforms generated by PSS/E. To simulate a real PMU signal, the bus voltage angle waveforms were down-sampled to 30 samples / s or one sample every 0.0333 seconds. The down-sampling as well as the dynamic signal processing was

performed using Python and the Python modules known as SciPy and NumPy. In order to automate the PSS/E simulations, the Python module named PssPy was also utilized. Shown below is the removal of the line from bus 1 to bus 24 at t = 3 seconds. All angles are referenced to bus 1.

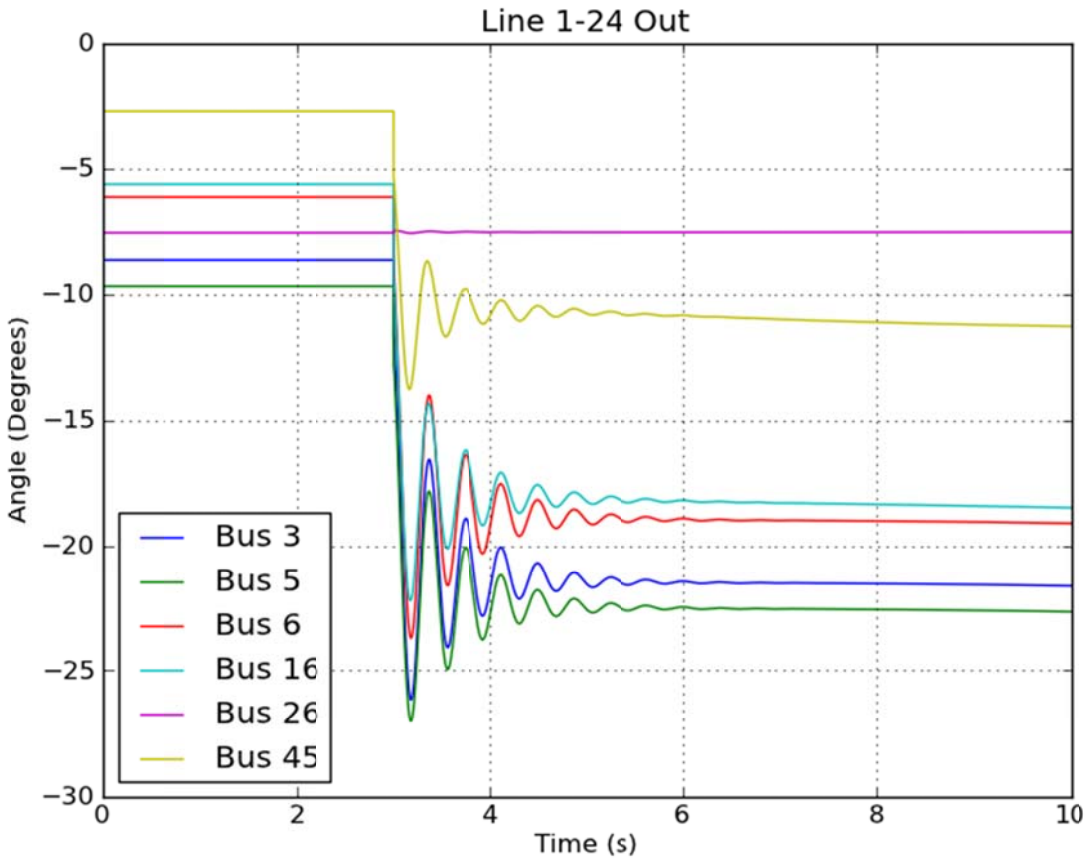


Figure 5.3 – Simulated PMU bus voltage angles when line 1 – 24 is removed.

Next, a low-pass, order 61 FIR filter with a cutoff frequency of 0.1 Hz was used to remove any noise or unintended step or impulse changes in the angles. Since the signals are simulated, no noise of any kind is present, but its application must be taken into consideration to show how the angle signals are changed by the filter. The magnitude and phase response of the filter are shown below in Figure 5.4.

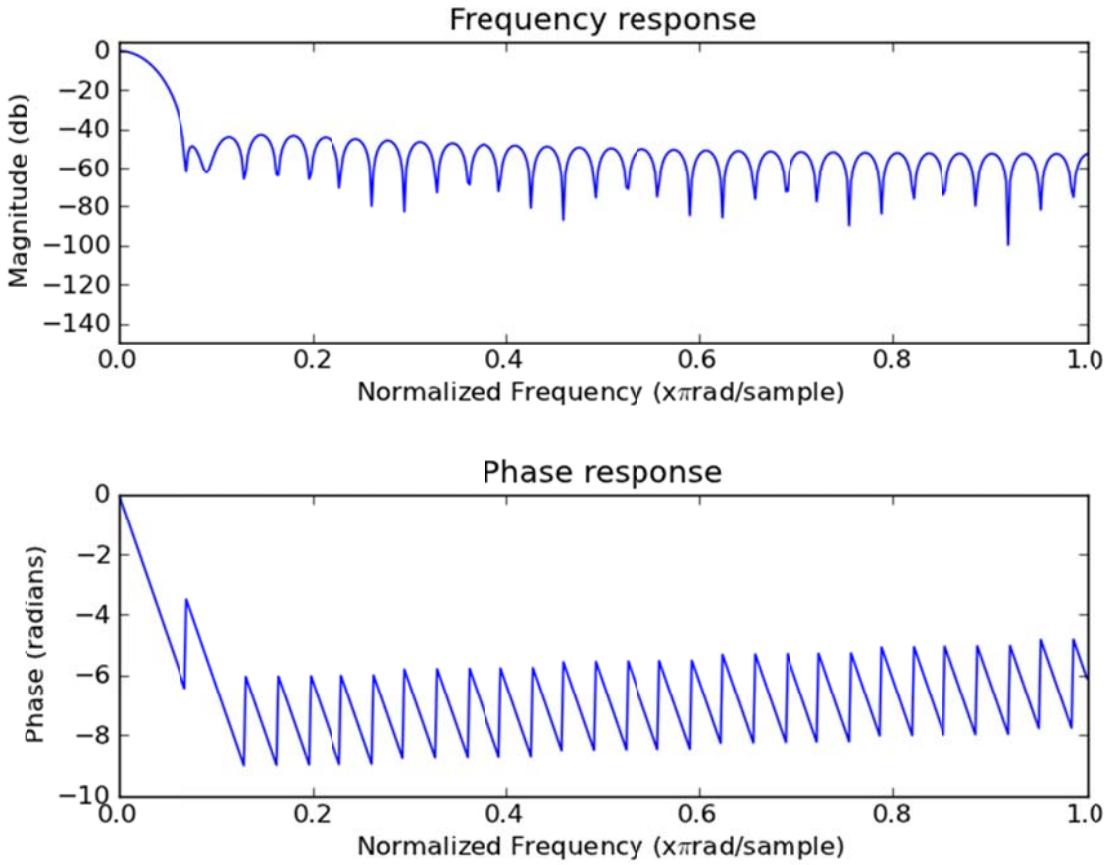


Figure 5.4 – Response of 61 order, FIR low pass filter.

The FIR filter above was implemented using a linear constant coefficient difference equation [31]. Filter coefficients were determined using the `firwin` function from the SciPy module for Python. Tate [8] showed the effectiveness of FIR filters in similar application. After applying the low pass filter to the line 1-24 outage from Figure 5.4, the result is shown in below.

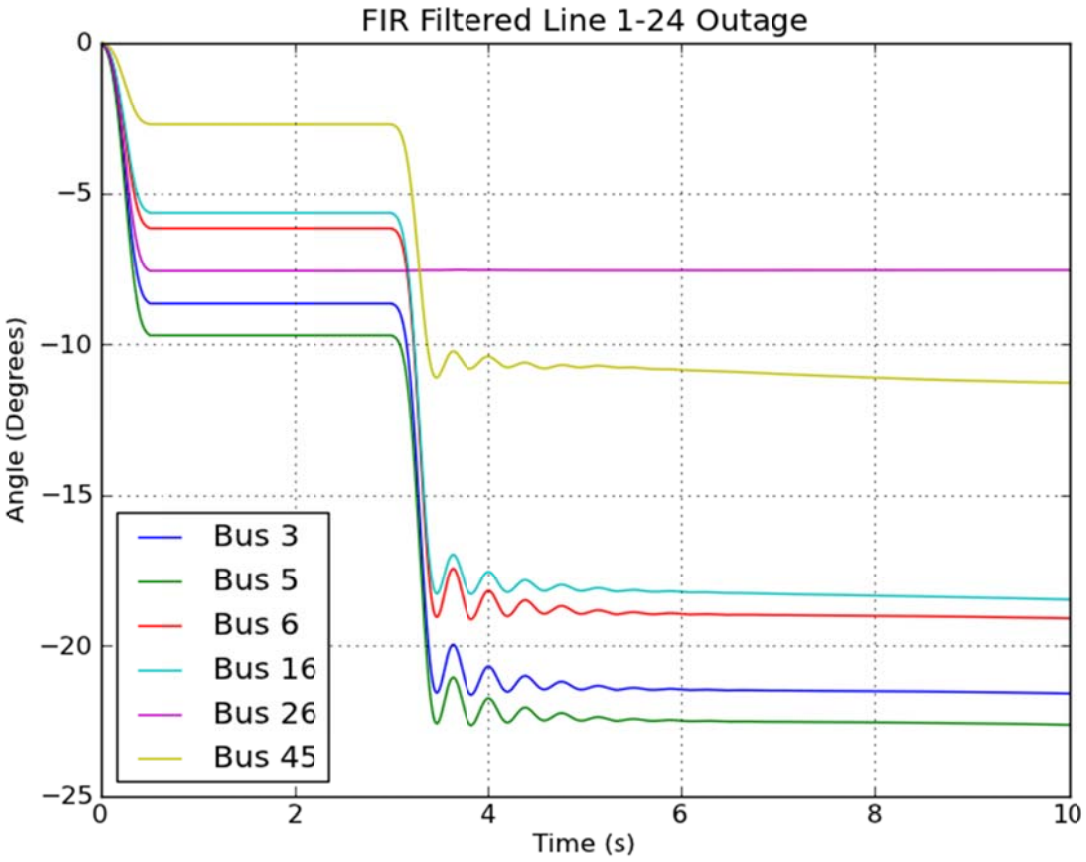


Figure 5.5 – FIR filtered version of Figure 5.3.

At each of the six buses with PMUs, the detection algorithm was then applied. In order for the true event time to be determined, detection was performed at each bus separately, and the maximum and minimum time values were then used. The first step in the detection algorithm was the derivative approximation with N=60 samples. Figure 5.6 shows the result of this step.

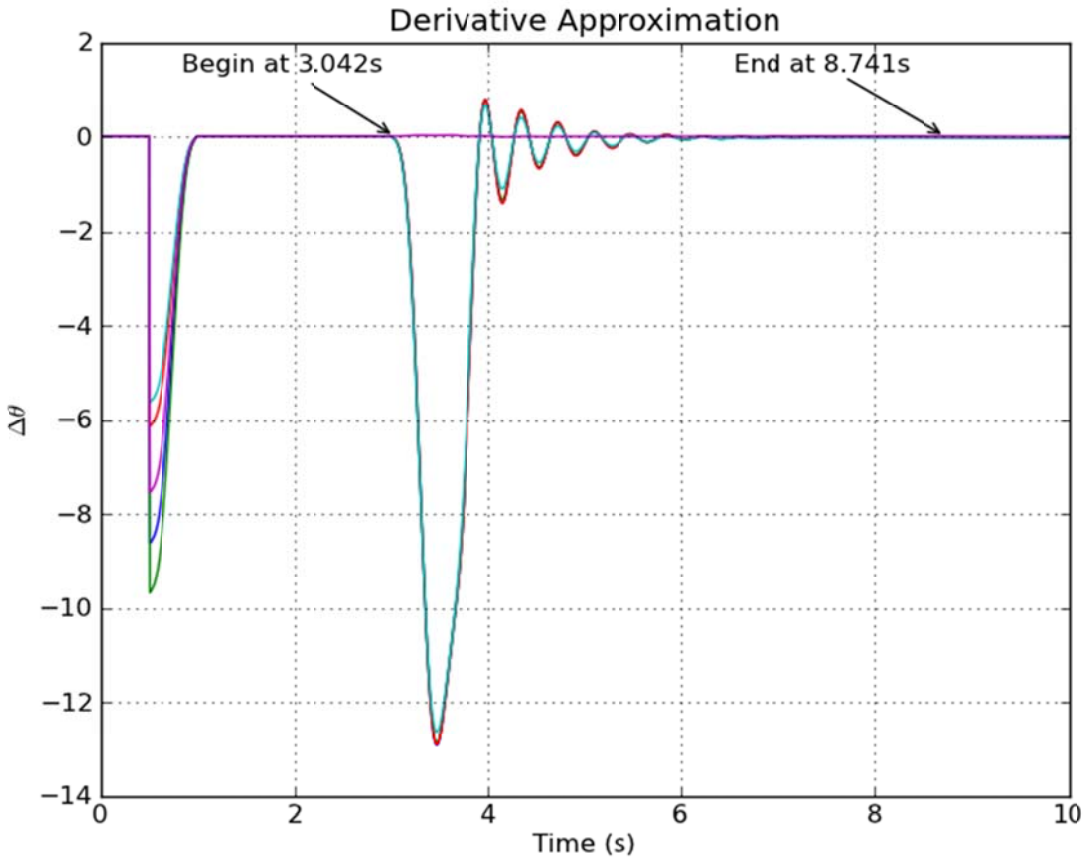


Figure 5.6 – Derivative approximation applied to Figure 5.5.

As seen in the figures above, the low pass filter has a built in delay of $\frac{N-1}{2} = 30 \text{ samples} = 1 \text{ second}$ and since the data must be zero-padded before filtering, an artifact is created at t = 1 second. This artifact also shows up in the derivative approximation. For this reason, the rest of the detection algorithm is delayed for 2 seconds to allow start-up effects to die out. Next, the derivative approximation is examined for abrupt changes above the threshold. For the purposes of this simulation a threshold of 0.05 degrees was used. Overbye and Tate show that a reasonable

assumption is between 0.01 and 0.57 degrees. These bounds come from manufacturers' noise tolerance and PMU standards. The same threshold was also used to determine the end of the event. These two times are annotated in Figure 5.6 above. The same process was performed for each of the ten line outages. In order to determine the angular difference, the pre and post event, low pass filtered bus voltage angles at the times indicated above were used. The resultant e_l basis vectors after scaling are shown in the table below.

Table 5.1 – Simulated angle change vectors using the proposed method.

PMU	Line 1	Line 2	Line 3	Line 4	Line 5	Line 6	Line 7	Line 8	Line 9	Line 10
1	0.476	0.399	0.520	0.495	-0.025	-0.355	0.543	0.610	0.418	0.234
2	0.475	0.386	0.521	0.494	-0.027	-0.356	0.642	0.511	0.413	0.228
3	0.476	0.403	0.519	0.495	-0.025	-0.355	0.542	0.606	0.419	0.237
4	0.472	0.346	-0.435	0.491	-0.030	-0.358	-0.006	-0.018	0.403	0.214
5	-0.001	0.000	0.001	-0.001	-0.001	0.006	0.000	0.000	0.001	-0.001
6	0.314	-0.641	-0.007	0.159	0.999	-0.702	-0.002	-0.005	0.563	0.890

Similar to the process performed in Chapter 4, the basis vectors in the table above were compared with the calculated basis vectors to find the distances between each. The result is shown in the table below. For seven out of the ten line outages, the algorithm correctly ranks the line. While there are three line outages which are identified incorrectly, in each of these cases, the correct ranking is still in the top two or three. In order to better tune the algorithm, the parameters above may have to be changed based on a particular system or a particular situation.

Table 5.2 – Distance between simulations (columns) and calculations (rows).

	Line 1	Line 2	Line 3	Line 4	Line 5	Line 6	Line 7	Line 8	Line 9	Line 10
Line 1	0.0102	0.3674	0.3291	0.1692	1.1194	0.4450	0.5994	0.5880	0.2686	0.7490
Line 2	0.2256	0.1567	0.5369	0.3835	0.9338	0.2318	0.7191	0.7096	0.0541	0.5444
Line 3	0.3253	0.6816	0.0067	0.1741	1.3762	0.7611	0.4476	0.4358	0.5917	1.0427
Line 4	0.1629	0.5346	0.1667	0.0045	1.2581	0.6114	0.5367	0.5246	0.4383	0.9060
Line 5	1.1026	0.7719	1.3494	1.2323	0.0301	0.6948	1.3609	1.3582	0.8603	0.3886
Line 6	0.3548	0.0489	0.6605	0.5108	0.8164	0.1017	0.8040	0.7956	0.0791	0.4177
Line 7	0.5978	0.7950	0.4527	0.5407	1.3827	0.8787	0.1388	0.0830	0.7489	1.0931
Line 8	0.5963	0.7956	0.4501	0.5391	1.3826	0.8779	0.0737	0.1236	0.7485	1.0933
Line 9	0.3220	0.0694	0.6291	0.4785	0.8467	0.1350	0.7818	0.7731	0.0465	0.4502
Line 10	0.7961	0.4408	1.0734	0.9400	0.3766	0.3586	1.1272	1.1223	0.5337	0.0452

5.4 PMU Data

PMU data for the 6000+ bus system, which was reduced for the previous analysis, are examined in the following section. For consistency, only the PMU locations used in the 47 bus reduced system are used for 6000+ bus system. In this way, the following analysis will show the effect of the power system dynamics and the generality of the proposed line outage detection algorithm. In the full 6000+ bus system, a different slack bus is utilized than the reduced 47 bus system. The slack bus in the larger system was equivalenced, so a different bus was chosen as the reference. Since the proposed line outage detection algorithm is based on relative angular change and all scaling due to bus injection is removed, the algorithm should perform well for any choice of slack or loading \ generation condition. Figure 5.7 below shows a three line to ground fault with 2 seconds of pre-fault data and approximately 16 seconds afterward. Corrective action was taken at approximately 0.5 seconds after the fault.

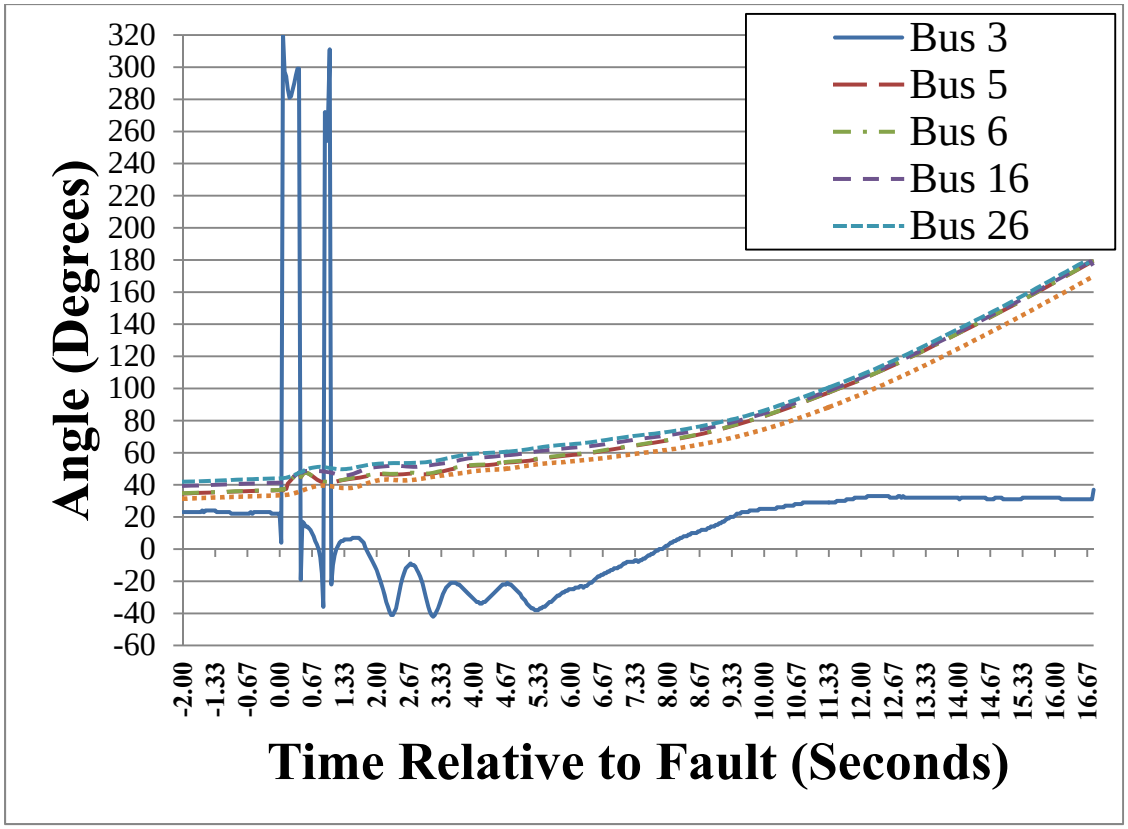


Figure 5.7 – PMU data during a line to ground fault.

As described above, the first step in processing this data is to apply an order 61 FIR filter with a Hamming window and cutoff frequency of 0.1Hz. The resultant angles are shown below in Figure 5.8. This figure shows an important aspect of the filtering algorithm. Unfortunately, the data that was provided only starts 2 seconds before the event. Therefore, the startup effects of the low-pass filter begin to interfere with the event itself. This occurs because the data must be pre-pended with zeros to allow the causal FIR filter to operate. If instead, the data is pre-pended with its edge value (the angles at t = -2s), this startup effect can be diminished as shown in Figure 5.9.

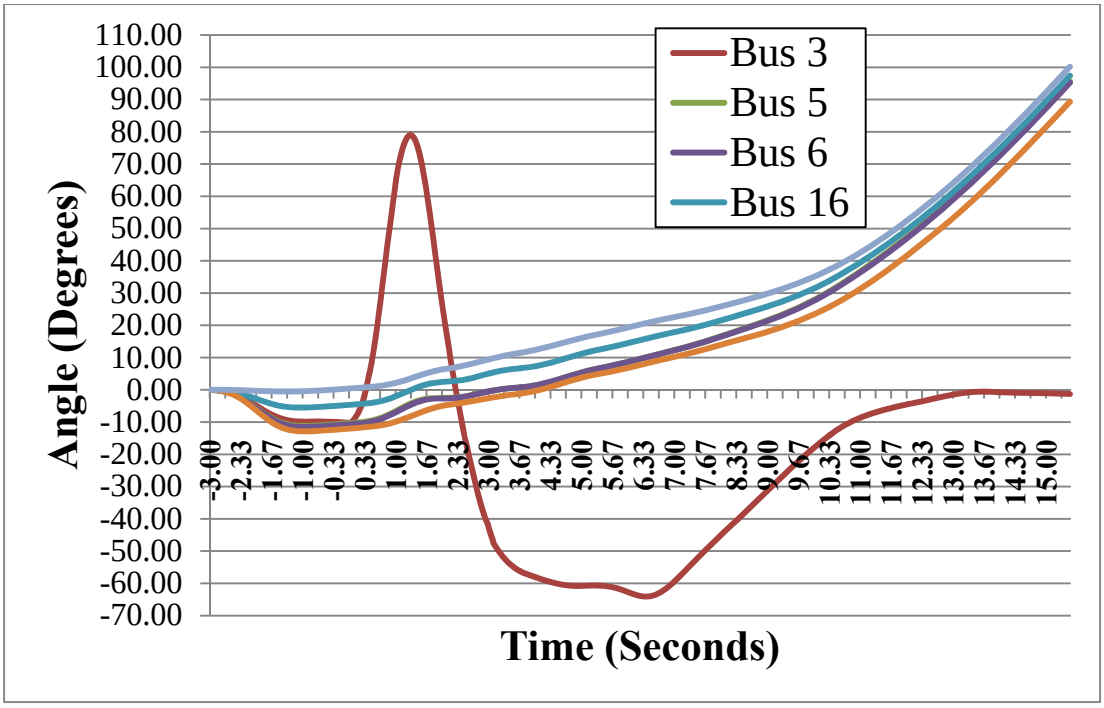


Figure 5.8 – Low Pass filtered PMU measurements.

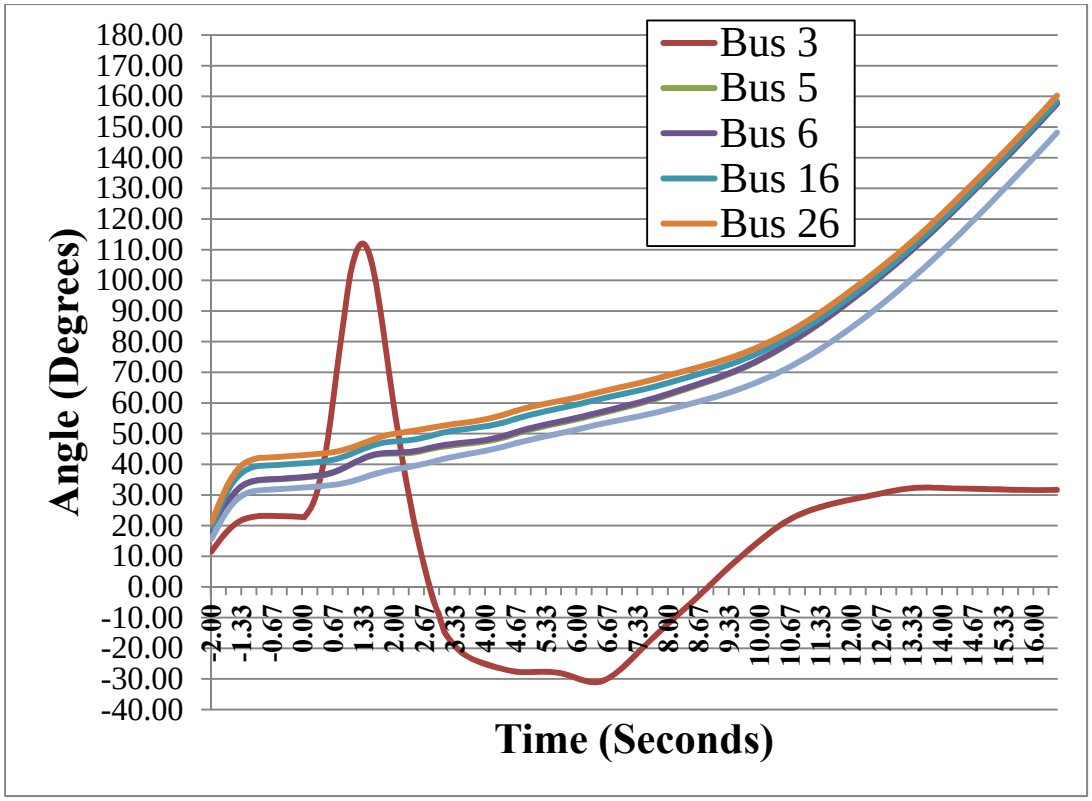


Figure 5.9 – Low pass filtered angles with pre-pended edge value.

After low-pass filtering the angles, the angular difference was found as shown below. Unfortunately, the data provided does not fully allow the algorithm to come to completion since the angular difference does not fall below the threshold for 5 out of the 6 angle measurements.

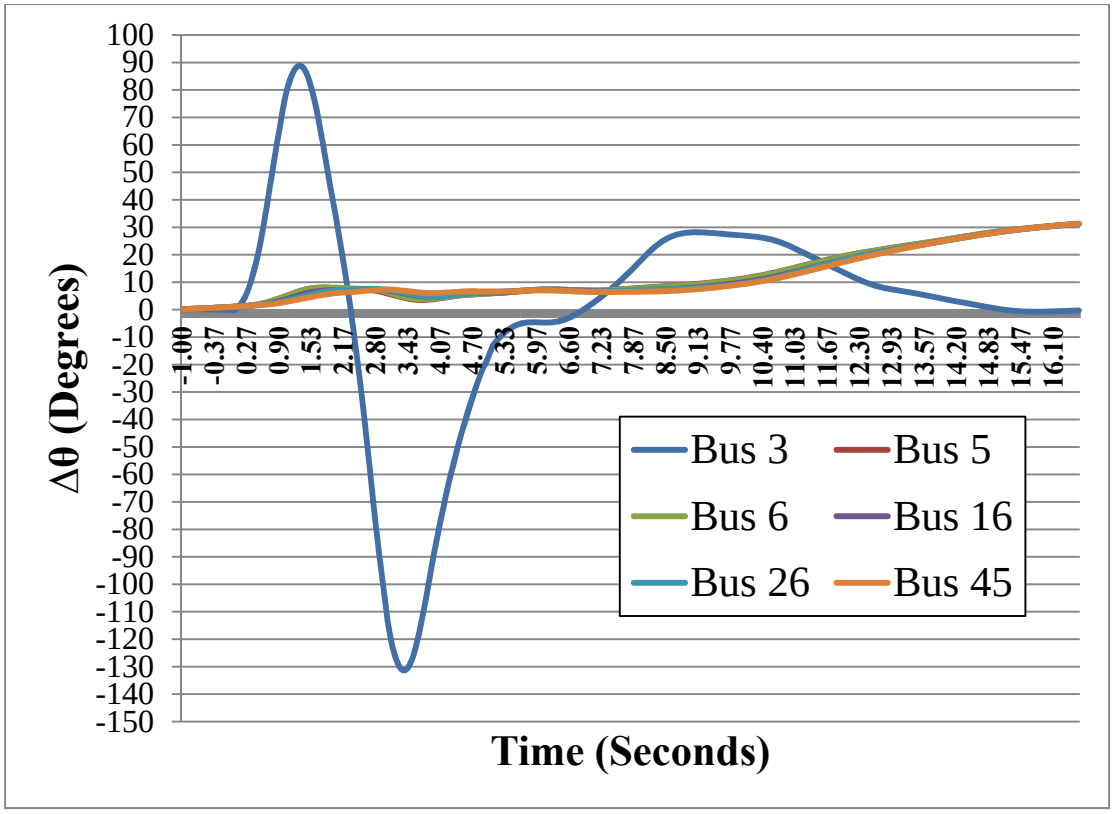


Figure 5.10 – Angular difference for PMU data.

CHAPTER SIX

CONCLUSION AND FUTURE RESEARCH

Modern power systems have become increasingly complex. The mere speed at which electricity travels may lead one to believe that controlling such a force is impossible. Electric power systems as a whole, however, do not change drastically from second to second. This fairly constant characteristic is what has allowed power systems to operate, in large part, with minimal closed loop, on-line control. As power systems have grown, the possibility for major catastrophes has also grown. No longer can a power system operate without automated intervention. In order to prevent large scale blackouts and interruption of service to essential loads, closed loop automated control must become the norm for power system operation. The basis for any type of such control is a synchronized measurement across the entire power system. With an accurate, up-to-date snapshot of all or part of the power system many new applications become possible.

While completely automated power system control and operation may not be possible in the near future, it is feasible that small strides can be made today. Preventing large cascading blackouts caused by something as simple as a single line outage is a very real possibility. The 2003 blackout, along with most of the major U.S. blackouts before it, was caused by a lack of information and communication, also dubbed situational awareness. Such blackouts may be caused by the outage of a single line. In this case, traditional power system protection schemes should prevent local area events from affecting the wider power system. If one or more levels of backup protection fail,

however, wide area protection may not exist to prevent a catastrophe. Synchronized phasor measurements can be used to create wide area monitoring systems to prevent such catastrophes.

In Chapter 2 it was shown that principle component analysis can be used to elicit information from wide area measurements. The method of line outage detection which was presented, dubbed "Principle Outage Vectors," requires a possibly large number of simulations similar to the Eigenfaces technique used in facial recognition. This method may not necessarily be practical, but it does show the utility of the PCA technique. Such statistical techniques may prove more useful in wide area monitoring systems. However, an analytical basis for the results garnered from PCA is not always available. In order to create a new, more deterministic algorithm, the line outage problem was explored more deeply in Chapter 3.

The two methods of line outage detection explored in Chapter 3 are distinct but rely upon similar assumptions. Overbye and Tate assumed that the topology and line flow data for the entire system will be available. In addition, they assumed that all synchrophasor measurements and line flow measurements are aggregated at a central location. In an effort to make an algorithm which is both more efficient and more universal, the proposed method does not rely on line flow measurements in any capacity. Rather, the algorithm was created under the assumption that a line outage will impose some basic characteristics on the bus voltage angles which are independent of anything but the topology of the system. Therefore, the bus voltage angles alone can be used to detect line outages.

Both the OT algorithm and the novel algorithm described in Chapter 3 rely on the DC power flow assumptions. Since the power flow is non-linear in reality, the DC power flow assumptions are an important tool for simplified power flow analysis. In order to compare both of these algorithms, steady state simulations were performed in Chapter 4 using a 47 bus reduced equivalent system. After applying both algorithms, it was discovered that the results from each are identical. Through further investigation, the calculated bus voltage angle changes in each case are identical after normalization. The majority of the scaling comes from the injected powers in the DC power flow assumptions. Thus, when normalizing the bus voltage angles, the effect of these injections is effectively removed. Plus, since both the proposed and OT algorithms utilize a normalization step, it is easy to see why their results might be identical.

Chapter 5 consisted of dynamic simulations using the proposed algorithm. The intention of this research was focused more so on the identification aspect of the algorithm, so much of the work performed by Overbye and Tate on event detection was replicated. As expected, the line outage detection method performs reasonably well for dynamically generated PMU measurements. However, some tuning of key parameters is required for specific systems with a given number of PMU measurements. While these dynamic simulations still depend solely on bus voltage angles, most PMUs can supply much more information.

The bus voltage angles depend on many system parameters other than the topology alone. However, it has been empirically shown that the bus voltage angles and line current magnitudes are the two most telling aspects of power system topology

changes [22]. Thus, it is possible that an algorithm which utilizes a combination of the two approaches above could minimize the required number of line flow measurements while maximizing the line outage detection accuracy. In addition, the DC power flow assumptions utilized above are simple, but with a reduced degree of accuracy. More accurate line outage detection algorithms such as the Principle Outage Vector technique may need to rely on a full Newton Raphson power flow solution. In fact, when examining the proposed algorithm, the method lends itself to such an approach. Since Newton-Raphson power flow uses an approximation to the derivative at a single point, known as the Jacobian, the method is quite similar to the line outage detection method proposed above. The proposed method utilizes the difference between bus voltage angles which is equivalent to the second term of the Kron Reduction. As seen above, a discrete difference is also an approximation to a derivate. Since the second term in the Kron Reduction has a rank of one, all angular differences are simply scaled versions of the basis. It is possible that the same reasoning could be applied to the Jacobian. The Jacobian, however, is typically full rank, so a more in depth analysis would be required.

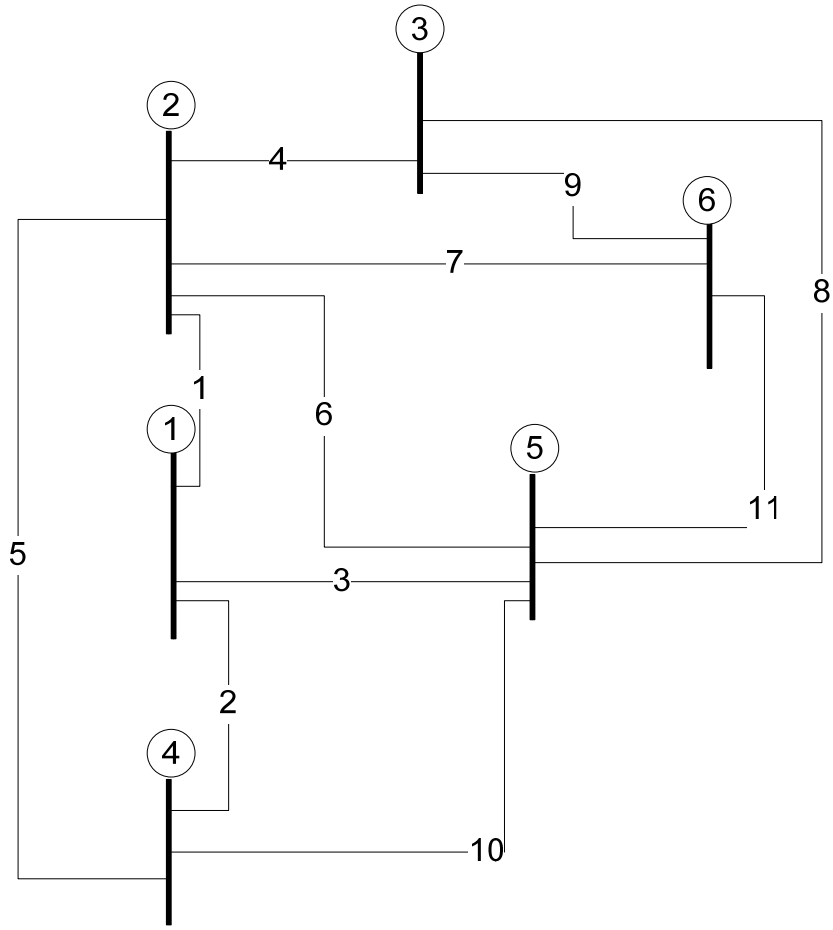
In summary, the conclusions of this thesis are two-fold. First, statistical techniques such as PCA are invaluable to the future synchrophasor applications. The accuracy of algorithms derived from PCA is extremely high. However, PCA based algorithms tend to be less concise and are only somewhat based on the underlying structure of the problem. Next, Overbye and Tate's algorithm for line outage detection, while useful, was improved upon. The OT algorithm requires line flow measurements on every line in the system to be effective. This requirement is both unnecessary and likely

impossible in some cases. Lastly, despite the inherent complexity of the power system the DC power flow assumptions can provide valuable insight into both line outage detection and power system operation as a whole.

APPENDICES

Appendix A

Six Bus Test System Parameters



From Bus	To Bus	R(pu)	X(pu)	BCAP/2 (pu)
1	2	0.10	0.20	0.02
1	4	0.05	0.20	0.02
1	5	0.08	0.30	0.03
2	3	0.05	0.25	0.03
2	4	0.05	0.10	0.01
2	5	0.10	0.30	0.02
2	6	0.07	0.20	0.0025
3	5	0.12	0.26	0.0025
3	6	0.02	0.10	0.01
4	5	0.20	0.40	0.04
5	6	0.10	0.30	0.03

Bus Number	Bus Type	V Schedule	P_{gen}	P_{load}	Q_{load}
1	Swing	1.05			
2	Gen.	1.05	0.50		
3	Gen.	1.07	0.6		
4	Load			0.7	0.7
5	Load			0.7	0.7
6	Load			0.7	0.7

Appendix B

Overbye and Tate Method

```
%typical usage of OandT_sim
function OandT_Run

   %lines of interest
   lines=[1,27,20,29,54,43,19,18,50,51];

   %locations of actual PMU installations
   actual_PMUs=zeros(1,47);
   actual_PMUs(3)=1;
   actual_PMUs(5)=1;
   actual_PMUs(6)=1;
   actual_PMUs(16)=1;
   actual_PMUs(26)=1;
   actual_PMUs(45)=1;

   [Oe,Fe,Ae]=OandT_sim(loadcase('TVA'),actual_PMUs,lines);

   Oe
   Fe
   Ae

end

%OANDT_SIM      Simulate line outage detection using Overbye, Tate
Method.
%  [Osuccess,Fsuccess,Asuccess]=OANDT_SIM(cse,actual_PMUs,lines)
%
%  Calculate the success rates in detecting line outages for full PMU
%  coverage, optimal PMU coverage, and a vector of actual PMU
locations.
%  The optimal PMU locations are determined using integer programming.
%  Line outage detection is performed by simulating line outages using
DC
%  power flow assumptions and pre-outage line flows, then comparing the
%  result to another simulation with randomized loading/generation.  It
is
%  assumed that the power injections are constant throughout the event.
%
%Usage:
%     cse         - system case given in MATPOWER format
%     actual_PMUs - for the N bus base_case system, PMUs is a 1xN
vector
%                   containg 1's and 0's where column i is 1 if a PMU
is
%                   installed at bus i.
%     lines       - line numbers to be studied.  For all lines, use
%                   lines=[1:num_lines]
%Note:
```

```
%     Requires MATPOWER and optimization toolbox.

function [Osuccess,Fsuccess,Asuccess]=OandT_sim(cse,actual_PMUs,lines)

   %number of buses in the system
   buses=size(cse.bus,1);

   %number of branches in the system
   branches=size(cse.branch,1);

   %begin optimal PMU placement
   TPMUs=build_PMUs(cse);

   %b vector for optimization, for complete observability at each bus
   b=ones(buses,1);

   %determine optimal PMU locations
   PMUs=bintprog(b,-TPMUs,-b)

   %if only one PMU is returned, try again with 2 PMUs observing each
bus
   if(sum(PMUs)==1)

         %new b vector with 2 PMUs observing each bus
         b=2*ones(buses,1);

         %rerun optimization
         PMUs=bintprog(b,-TPMUs,-b)

   end

   %calculate success rates
   Osuccess=1-OandT(cse,PMUs,lines)/branches;
   Fsuccess=1-OandT(cse,ones(1,buses),lines)/branches;
   Asuccess=1-OandT(cse,actual_PMUs,lines)/branches;

end

%Overbye and Tate line outage detection for given PMUs and lines
function errors=OandT(base_case, PMUs, lines)

   %suppress MATPOWER output
   opt=mpoption('OUT_ALL',0,'VERBOSE',0);

   %number of lines in the system
   num_lines=size(base_case.branch,1);

   %number of buses in the system
   num_bus=size(base_case.bus,1);
```

```
   %create a copy of the case for changing the load values
   new_case=base_case;

   %run the base case Newton Raphson load flow
   results=runpf(base_case,opt);

   %get the bus voltage angles before outage
   pre_angles=results.bus(:,9);

   %form the matrix of power transfer distribution factors (PTDFs)
   ptdf=makePTDF(base_case.baseMVA,base_case.bus,base_case.branch);

   %determine the power injections required to force the line flow to
zero
   %for each outaged line
   for branch=1:size(lines,2);

         Plhat(branch)=results.branch(lines(1,branch),14)/(1-
ptdf(lines(1,branch),results.branch(lines(1,branch),1)));

   end

   %form the B matrix
   [B, Bf, Pbusinj, Pfinj] = makeBdc(base_case.baseMVA, base_case.bus,
base_case.branch);

   %form the PMU connection matrix
   K=zeros(sum(PMUs),num_bus);

   %initialize PMU counter
   num_pmu=0;

   %cycle through buses looking for PMUs
   for b=1:num_bus

         %if a PMU exists at a bus add a one to the connection matrix
         if PMUs(b)
            num_pmu=num_pmu+1;
            K(num_pmu,b)=1;
         end

   end

   %prefill K*X product result with zeros.
   KBinv=zeros(sum(PMUs),num_bus);

   %KBinv = K * inv(B) = K * X - > only take rows of X with PMUs

KBinv(1:end,2:num_bus)=K(1:end,2:num_bus)*full(inv(B(2:num_bus,2:num_bu
s)));
```

```
   %prefill calculated delta theta with zeros
   delta_t_l=zeros(size(KBinv,1),size(base_case.branch,1));

   %cycle through branches formin vector of injections 1 = in, -1 = out
   for br=1:size(lines,2)

      %prefill with zeros
      inj=zeros(num_bus,1);

      %cycle through buses
      for bus=1:size(base_case.bus,1)
         if(bus==base_case.branch(lines(1,br),2))
            inj(bus,1)=1;
         elseif(bus==base_case.branch(lines(1,br),1))
            inj(bus,1)=-1;
         end
      end

      %calculated change in bus voltage angles
      delta_t_l(:,br)=Plhat(br)*KBinv*inj;

   end

   %pre-fill vector of simulated bus voltage angle differences
   dt=zeros(sum(PMUs),1);

   %cycle through branches and simulate line outages
   for branch=1:size(lines,2)

      %calculate angles after the outage
      outage_angles=lo_angles(base_case,lines(1,branch),0);
      num_pmu=0;
      for b=1:num_bus
         if(PMUs(b))
            num_pmu=num_pmu+1;
            dt(num_pmu,branch)=pre_angles(b,1)-outage_angles(b,1);
         end
      end

   end

   %determine the normalized angular distances between simulated and
   %calculated line outages
   for line_int=1:size(lines,2)
      for line_out=1:size(lines,2)


NAD(line_int,line_out)=min(norm(dt(:,line_out)./norm(dt(:,line_out))-
delta_t_l(:,line_int)./norm(delta_t_l(:,line_int))),...
```

```
norm(dt(:,line_out)./norm(dt(:,line_out))+delta_t_l(:,line_int)./norm(d
elta_t_l(:,line_int))));

      end
   end

   %initialize errors
   errors=0;

   %vector to determine troublesome lines
   error_place=zeros(size(lines,2),1);

   %rank lines in terms of closeness to actual outaged line
   for line_rank=1:size(lines,2)

      %find minimum normalized angular distance in each column
      [val,I]=min(NAD(:,line_rank));

      %search each row for the minumum
      idx=knnsearch(val,NAD(:,line_rank),1);
      if(line_rank~=idx)

         errors=errors+1;
         error_place(line_rank,1)=1;

      end

   end

end

%return the bus angles for a line outage on case mpc
function angles=lo_angles(mpc,line,flag)

   %don't display pf solution
   if flag==1
      opt=mpoption('PF_DC',1,'OUT_ALL',0,'VERBOSE',0);
   elseif flag==2

opt=mpoption('PF_ALG',3,'PF_MAX_IT_FD',5,'OUT_ALL',0,'VERBOSE',0);
   else
      opt=mpoption('OUT_ALL',0,'VERBOSE',0);
   end

   %switch out the specific line
   mpc.branch(line,11)=0;

   %run the power flow with the line outage
   results=runpf(mpc,opt);
```

```
   %return the angles only
   angles=results.bus(:,9);

   %switch the line back into service
   mpc.branch(line,11)=1;

end
```

Appendix C

B.2 Outage Vector Method

```
%typical usage of LOD_sim
function LOD_Run
  %array of PMU bus numbers for actual PMU installations
  actual_PMUs=[3,5,6,16,26,45];

  %array of lines of interest, top ten loaded lines
  lines=[1,27,20,29,54,43,19,18,50,51];

  [Oe,Fe,Ae]=LOD_sim(loadcase('TVA'),actual_PMUs,lines);

   Oe
   Fe
   Ae

end

%LOD_SIM      Simulate line outage detection using normalized vectors.
%  [Osuccess,Fsuccess,Asuccess]=LOD_SIM(cse,actual_PMUs,lines)
%
%  Calculate the success rates in detecting line outages for full PMU
%  coverage, optimal PMU coverage, and a vector of actual PMU
locations.
%  The optimal PMU locations are determined using integer programming.
%  Line outage detection is performed by simulating line outages,
finding
%  the normalized bus voltage angular differences, then comparing the
%  result to another simulation with randomized loading/generation.  It
is
%  assumed that the power injections are constant throughout the event.
%
%Usage:
%     cse         - system case given in MATPOWER format
%     actual_PMUs - for the N bus base_case system, PMUs is a 1xN
vector
%                   containg the bus numbers of actual PMU locations.
%     lines       - line numbers to be studied.  For all lines, use
%                   lines=[1:num_lines]
%Note:
%     Requires MATPOWER and optimization toolbox.

function [Osuccess,Fsuccess,Asuccess]=LOD_sim(cse,actual_PMUs,lines)

   %suppress MATPOWER output
   opt=mpoption('OUT_ALL',0,'VERBOSE',0);

   %make a copy of the case for changing the load values
   new_case=cse;
```

```
   %number of buses in the system
   num_bus=size(cse.bus,1);

   %number of branches in the system
   branches=size(cse.branch,1);

   %begin optimal PMU placement
   TPMUs=build_PMUs(cse);

   %b vector for optimization, for complete observability at each bus
   b=ones(num_bus,1);

   %determine optimal PMU locations
   PMUs=bintprog(b,-TPMUs,-b);

   %if only one PMU is returned, try again with 2 PMUs observing each
bus
   if(sum(PMUs)==1)

         %new b vector with 2 PMUs observing each bus
         b=2*ones(num_bus,1);

         %rerun optimization
         PMUs=bintprog(b,-TPMUs,-b);

   end

   %MATPOWER function to build B matrix
   [BBUS, BF, PBUSINJ, PFINJ]=makeBdc(cse.baseMVA,cse.bus,cse.branch);

   %Invert non-slack bus portion of B matrix
   Xp=inv(full(BBUS(2:end,2:end)));

   %prefill X matrix with zeros, so that slack row/column is zero
   X=zeros(num_bus);

   %copy non-zero elements
   X(2:end,2:end)=Xp;

   %array of PMU bus numbers for the optimal PMU installations
   optim_PMUs=find(PMUs);

   %cycle through lines to study
   for br=1:size(lines,2)

      %X matrix after line removal

X2=add_lp_to_bus_nc(X,cse.branch(lines(1,br),1),cse.branch(lines(1,br),
2),cse.branch(lines(1,br),4));
```

```
      %difference in X matrices = L*M^(-1)*L.'
      dx=X-X2;

      %perform singular value decomposition to find optimal orthonormal
      %basis for the range of dx. Alternately, could find rref of dx
      [u,s,v]=svd(dx);

      %prototypical outage vectors for full coverage
      full_proto(:,br)=v(:,1);

      %prototypical outage vectors for actual coverage

actual_proto(:,br)=full_proto(actual_PMUs,br)/norm(full_proto(actual_PM
Us,br));

      %prototypical outage vectors for optimal coverage

optim_proto(:,br)=full_proto(optim_PMUs,br)/norm(full_proto(optim_PMUs,
br));

   end

   %create empty vector for load zones
   zones=[];

   %cycle through buses to determine the load zones
   for b=1:num_bus

      %find indices of the current zone in the running list of zones
      [r,c]=find(zones==cse.bus(b,11));

      %if it isn't already in the list, add it
      if isempty(r)

         zones(end+1,1)=cse.bus(b,11);

      end

   end

   %randomize the loading at each load zone
   for i=1:size(zones,1)

      ld_vec(1,i)=0.2*(randn(1))+1;

   end

   %scale the load for the new case
   new_case.bus=scale_load(ld_vec,cse.bus);
```

```
   %determine outage vector simulated measurements for optimal coverage
   ovO_meas=outage_vectors(cse,optim_PMUs,lines,0);

   %determine outage vector simulated measurements for full coverage
   ovF_meas=outage_vectors(cse,[1:num_bus],lines,0);

   %determine outage vector simulated measurements for actual coverage
   ovA_meas=outage_vectors(cse,actual_PMUs,lines,0);

   %initialize errors to zeros
   Oerror=0;

   Ferror=0;

   Aerror=0;

   %cycle through lines and search for closest match between simulated
and
   %calculated outage vectors for all three cases
   for b=1:size(lines,2)

       idx=knnsearch(abs(ovO_meas(:,b).'),abs(optim_proto.'),1);

       if idx ~= b
          Oerror=Oerror+1;

       end

       idx=knnsearch(abs(ovF_meas(:,b).'),abs(full_proto.'),1);

       if idx ~= b

          Ferror=Ferror+1;

       end

       idx=knnsearch(abs(ovA_meas(:,b).'),abs(actual_proto.'),1);

       if idx ~= b

          Aerror=Aerror+1;

       end

   end

   %calculate success rates
   Osuccess=1-Oerror/branches;
   Fsuccess=1-Ferror/branches;
```

```
   Asuccess=1-Aerror/branches;

end


%OUTAGE_VECTORS      Normalized vector bus voltage angular differences.
%  V=OUTAGE_VECTORS(base_case, PMUs, lines, flag) calculates the
difference between pre
%  and post outage bus voltage angles, then normalizes the length of
the
%  vector to one.
%
%Usage:
%     base_case - system case given in MATPOWER format
%     PMUs      - for the N bus base_case system, PMUs is a 1xN vector
%                 containg 1's and 0's.  If bus m has a PMU measurment,
%                 PMUs(1,m) will be 1, otherwise 0. This vector can be
%                 found using the optimal placement of PMUs, for
example,
%                 with integer programming.
%     lines     - line numbers to be studied.  For all lines, use
%                 lines=[1:num_lines]
%     flag      - flag to determine which power flow solution type to
use
%                 flag = 1 - uses DC power flow assumptions
%                 flag = 2 - uses Decoupled power flow assumptions
%                 flag = other - uses full Newton Raphson power flow

function V=outage_vectors(base_case, PMUs, lines, flag)

   %suppress output from MATPOWER
   opt=mpoption('OUT_ALL',0,'VERBOSE',0);

   %run the base case power flow
   results=runpf(base_case,opt);

   %number of lines in the system
   num_lines=size(base_case.branch,1);

   %number of buses in the system
   num_bus=size(base_case.bus,1);

   %bus voltage angles in the base case
   pre_angles=results.bus(:,9);

   %create empty vector to hold bus numbers of PMU locations
   pmu_buses=[];

   %make sure the PMU vector is the right dimension
   if size(PMUs,1)>size(PMUs,2)
```

```
      PMUs=PMUs.';

   end

   %create vector of bus numbers where PMUs are installed
   for bus=1:size(PMUs,2)

      %if a PMU exists on a bus add the bus to the list
      if PMUs(bus)
         pmu_buses=[pmu_buses bus];
      end

   end

   %create array of prototype vectors
   evec_proto=zeros(sum(PMUs),num_lines);

   %cycle through lines
   for branch=1:size(lines,2)

      %determine the angles at each bus for each line outage
      out_ang(:,branch)=lo_angles(base_case,lines(1,branch),flag);

      %create vector of pre and post outage angles
      %if the system has m PMUs, delta_theta will be m x 2
      delta_theta(:,branch)=[pre_angles(:)-out_ang(:,branch)];

   end

   %only select the rows corresponding to the PMUs that were given
   V=delta_theta(PMUs.',:);

   %normalize the vectors
   for br=1:size(lines,2)

      V(:,br)=V(:,br)/norm(V(:,br));

   end

end

%return the bus angles for a line outage on case mpc
function angles=lo_angles(mpc,line,flag)

   %don't display pf solution
   if flag==1
      opt=mpoption('PF_DC',1,'OUT_ALL',0,'VERBOSE',0);
   elseif flag==2

opt=mpoption('PF_ALG',3,'PF_MAX_IT_FD',5,'OUT_ALL',0,'VERBOSE',0);
```

```
   else
      opt=mpoption('OUT_ALL',0,'VERBOSE',0);
   end

   %switch out the specific line
   mpc.branch(line,11)=0;

   %run the power flow with the line outage
   results=runpf(mpc,opt);

   %return the angles only
   angles=results.bus(:,9);

   %switch the line back into service
   mpc.branch(line,11)=1;

end


%BUILD_PMUS Create PMU connection matrix for integer programmiing
%     TPMU=build_PMUs(base_case) returns the connection matrix for
%     determing the optimal PMU placement using integer programming.
%
%Usage:
%     base_case is a MATPOWER case


function TPMU=build_PMUs(base_case)

   %number of branches in the system
   branches=size(base_case.branch,1);

   %Create PMU location matrix
   TPMU=eye(size(base_case.bus,1));

   %cycle through all branches to determine the connection matrix
   for branch=1:branches

      %obtain from and two bus numbers for MATPOWER cases or simple
brach
      %matrices
      if(isstruct(base_case))
         fr=base_case.branch(branch,1);
         to=base_case.branch(branch,2);
      else
         fr=base_case(branch,1);
         to=base_case(branch,2);
      end

      %if two buses are connected, set the corresponding element to 1
      TPMU(fr,to)=1;
```

```
      TPMU(to,fr)=1;

   end
end
```

REFERENCES

[1] E. O. Schweitzer III and D. E. Whitehead, “Real world synchrophasors solutions,” *IEEE PES, 62nd Annual Conference for Protective Relay Engineers*, pp. 536-547, 2009.

[2] A. G. Phadke, "Synchronized phasor measurements-a historical overview," *IEEE/PES Asia Pacific Transmission and Distribution Conference and Exhibition,* 2002, vol. 1, pp. 476-479.

[3] "IEEE Standard for Synchrophasors for Power Systems," *IEEE Std C37.118-2005 (Revision of IEEE Std 1344-1995)* , vol., no., pp.0_1-57, 2006.

[4] 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.

[5] US-Canada Power System Outage Task Force, Final Report on August 14, 2003 the Blackout in the United States and Canada, 2004. [Online]. Available: https://reports.energy.gov/BlackoutFinal-Web.pdf.

[6] Hadley, et al. “Securing Wide Area Measurement Systems”, Pacific Northwest national Laboratory. June 2007. [Online]. Available: http://www.oe.energy.gov/DocumentsandMedia/Securing_WAMS.pdf

[7] J.Y. Cai, Z. Huang, J. Hauer and K. Martin. “Current status and experience of WAMS implementation in North America,” *Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES* pp. 1-7.

[8] J., Tate, "Event detection and visualization based on phasor measurement units for improved situational awareness". Ph.D. diss., University of Illinois at Urbana-Champaign ,2008. In Dissertations & Theses: Full Text [database on-line]; available from http://www.proquest.com (publication number AAT 3337937; accessed January 17, 2011).

[9] A. Tiwari and V. Ajjarapu,“Event identification and contingency assessment for voltage stability via PMU,” *Power Symposium, 2007. NAPS '07. 39th North American* pp. 413-420.

[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 *IEEE Power Engineering Society General Meeting,* 2004, vol. 1, pp. 582-588.

[11] R.F. Nuqui, A.G. Phadke, R. P. Schulz, N.B. Bhatt, “Fast On-line Voltage Security Monitoring Using Synchronized Phasor Measurements and Decision Trees”, *IEEE 2201 Power Engineering Society Winter Meeting*, vol. 3, Jan. 28-Feb. 1, 2001, pp 1347 -1352.

[12] R. Aggarwal and Y. Song, “Artificial neural networks in power systems,” *PEJ*, pp. 279–287, December 1998.

[13] M.J. Smith and K. Wedeward, "Event detection and location in electric power systems using constrained optimization," *Power & Energy Society General Meeting, 2009. PES '09. IEEE* , vol., no., pp.1-6, 26-30, July 2009.

[14] J. E. Tate and T. J. Overbye, “Line outage detection using phasor angle measurements,” *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1644-1652, Nov. 2008.

[15] J. E. Tate and T. J. Overbye; , "Double line outage detection using phasor angle measurements," *Power & Energy Society General Meeting, 2009. PES '09. IEEE* , pp.1-5, 26-30 July 2009.

[16] A. Wood and B. Wollenberg, *Power Generation, Operation, and Control*. New York: Wiley, 1984.

[17] R.M. Gardner, J.K. Wang, Liu Yilu, "Power system event location analysis using wide-area measurements," *Power Engineering Society General Meeting, 2006. IEEE* , pp.7

[18] T.L. Baldwin, L. Mili, M. B. Boisen, and R. Adapa, “Power system observability with minimal phasor measurement placement”, *IEEE Trans. Power Syst.*, vol. 8, no.2, pp. 707-715, May 1993.

[19] K. A. Clements, “Observability Methods and Optimal Meter Placement,” *Electrical Power & Energy Systems*, vol. 12, No. 2, pp. 88-93, Aptil 1990.

[20] Z. Zhao, "Sensitivity constrained PMU placement utilizing multiple methods". M.S. diss., Clemson University, 2010. In Dissertations & Theses @ Clemson University [database on-line]; available from http://www.proquest.com (publication number AAT 1475592; accessed January 18, 2011).

[21] B. Gou, “ Optimal placement of PMUs by integer linear programming,” *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1525-1526, Aug. 2008.

[22] M. Vutsinas, "Contingency analysis using synchrophasor measurements". M.S. diss., Clemson University, 2008. In Dissertations & Theses @ Clemson University [database on-line]; available from http://www.proquest.com (publication number AAT 1461004; accessed January 24, 2011).

[23] J. E. Jackson, *A User's Guide to Principal Components*. New York: John Wiley & Sons, Inc 1991.

[24] I.T. Jolliffe, *Principal Component Analysis*. New York: Springer 2002.

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

[26] M.A. Turk and A.P Pentland, "Face Recognition Using Eigenfaces," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. pp. 586–591.

[27] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER's Extensible Optimal Power Flow Architecture," *Power and Energy Society General Meeting, 2009 IEEE*, pp. 1-7, July 26-30 2009.

[28] J. Grainger and W. Stevenson, *Power System Analysis*, New York: McGraw Hill, 1994.

[30] A. Michel and C. Herget, *Applied Algebra and Functional Analysis*, New York: Dover Publications, Inc. 1981.

[31] L. Ludeman, *Fundamentals of Digital Signal Processing*, John Wiley & Sonse, 1986.