

Conditioning of FNET Data and Triangulation of Generator Trips in the Eastern
Interconnected System

By

Robert Matthew Gardner

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State
University in partial fulfillment of the requirements for the degree of

Master of Science
In
Electrical Engineering

Approved by:

Dr. Yilu Liu (Chair)
Dr. Virgilio Centeno
Dr. Jaime De La Reelopez

29 July 2005
Blacksburg, Virginia U.S.A.

Keywords: Frequency Data Recorder, FNET, Eastern Interconnected System,
Electromechanical Wave, Frequency Perturbation/Wave Front/Disturbance,
Hypocenter, Hypocentral Loci

Conditioning of FNET Data and Triangulation of Generator Trips in the Eastern Interconnected System

Robert Matthew Gardner

ABSTRACT

Using data from the frequency disturbance recorders (FDRs) that comprise the nation-wide frequency monitoring network known as FNET, disturbances in the eastern interconnected system (EI) have been monitored and recorded over the past several years. Analysis of this and other data by a wide variety of research scientists and engineers has rendered the idea that frequency disturbances from generator trips, transmission line trips, load trips, and other events, travel with finite speed as electromechanical waves throughout any power system (in this case the EI). Using FNET data as a tool, it is possible to measure and output the arrival times of these disturbance waves with a time resolution of 100 ms.

To observe with certainty the arrival time of the frequency disturbance waves, field data collected by the FDRs must first be conditioned in a robust manner. The current method that uses the moving mean of raw FDR data is analyzed and two computationally efficient robust methods are suggested in this report. These new methods that rely on robust statistics are more resistant to the effect of outliers contained within the raw FDR data. Furthermore, like the moving mean, these methods smooth the raw data without removing the general trend.

Having recorded and conditioned the FDR data, three conventional triangulation techniques taken from the field of seismology are proposed and analyzed. This study reconfirms the fact that the EI is not a medium of continuous elasticity through which the frequency perturbations travel but rather a discontinuous patchwork of varying elasticities. Within this report, nine generator trip events are analyzed and the aforementioned triangulation methods are applied. The advantages and disadvantages of each method are discussed. To conclude, axioms of future research are proposed and delineated.

Acknowledgement

I would like express my deepest appreciation to Dr. Yilu Liu for her patient guidance and gracious provision throughout this study. Without her support, the completion of this study would not have been possible. She is respected and appreciated for her personal and professional mentoring.

Also, I would like to thank Dr. Martin Chapman, director of the Virginia Tech Seismological Observatory, for his assistance and guidance in the field of seismology.

To those who came before me and made the way plain, I am indebted. I am thankful for the quality research and work done especially by Shu-Jen Tsai, Jian Zuo, and Zhian Zhong.

I gratefully acknowledge the financial support from the National Science Foundation of the United States.

I am thankful for the endless love that my family has shown me. Thanks to my parents and parents-in-law for the gracious love and support they have shown me and my wife, Molly, as we simultaneously worked to complete our theses.

Finally, I would like to express gratitude to my lovely and beautiful wife Molly, for her love, support, and sacrifices without which I would have certainly failed.

To my beloved family

Acronyms and Symbols

EI: Eastern (United States) Interconnected System

FNET: United States Frequency Monitoring Network

FDR: Frequency Disturbance Recorder

PMU: Phasor Measurement Unit

GPS: Global Positioning System

NERC¹: North American Electric Reliability Council

ECAR: East Central Area Reliability Coordination Agreement

SERC: Southeastern Electric Reliability Council

MAIN: Mid-America Interconnected Network

FRCC: Florida Reliability Coordinating Council

GUI: Graphical User Interface

LS: Least-Squares

ML: Maximum Likelihood

MAD: Median Absolute Deviation

¹ <http://www.nerc.com>

Table of Contents

ABSTRACT.....	ii
Acknowledgement	iii
Acronyms and Symbols	v
List of Figures	viii
List of Tables	x
Chapter 1 Introduction	1
1.1 Scope.....	1
1.2 Electromechanical Wave Propagation	2
1.3 The Role of FNET	3
1.4 Organization of the Study	9
Chapter 2 Data Conditioning and Curve Fitting	11
2.1 The Nature of FDR Data.....	11
2.2 The Currently Used Data Smoothing Technique.....	12
2.3 Two Robust and Computationally Efficient Techniques.....	16
2.3.1 The Moving Median	16
2.3.2 Curve Fitting with Outlier Rejection	18
2.4 Summary	23
Chapter 3 Linear Triangulation Methods.....	35
3.1 The Limited Contributions from Seismology	35
3.2 The Least-Squares Approach.....	40
3.3 Problems with the Over-Constrained System	49
3.4 Summary	57
Chapter 4 Non-Linear Triangulation Methods	58
4.1 Solutions to Non-Linear Systems of Equations	58
4.2 The Gradient Search Method	67
4.3 Summary	76
Chapter 5 Conclusions and Future Work.....	77
5.1 Conclusions.....	78
5.2 Future Work	80
References.....	83
Appendix I MATLAB Scripts for Data Conditioning and Curve Fitting.....	87
Appendix II MATLAB Scripts for Event Location.....	107

AII.1	Least-Squares Method:	107
AII.2	Newton Method:	108
AII.3	Gradient Search Method:	109

List of Figures

Figure 1-1: Location of FDRs used in this study.....	3
Figure 1-2: Location of base cases used in this study.....	5
Figure 1-3: A cartoon of an example frequency event.	8
Figure 2-1: Problematic FDR data with signal quality issues highlighted.	12
Figure 2-2: The Gaussian distribution.	13
Figure 2-3: Raw FDR data plotted with moving mean.....	14
Figure 2-4: Raw perturbed data plotted with moving mean.	15
Figure 2-5: The Laplacian Distribution	17
Figure 2-6: Raw perturbed data with moving mean and moving median.....	18
Figure 2-7: Sample FDR data recording generator trip.	19
Figure 2-8: Sample FDR data with outliers identified.....	20
Figure 2-9: Raw FDR data plotted with moving median and fitted curve.....	22
Figure 2-10: Disturbance wave detection error with static threshold.	24
Figure 2-11: Magnitude of frequency derivative of moving median and fitted curve (see figure 2-9).	25
Figure 2-12: Detection times for static threshold and slope-based methods (see figure 2-8).	26
Figure 2-13: FDR data conditioning GUI developed for FNET (Base Case 1)...	27
Figure 2-14: FDR data conditioning GUI developed for FNET (Base Case 2)...	28
Figure 2-15: FDR data conditioning GUI developed for FNET (Base Case 3)...	28
Figure 2-16: FDR data conditioning GUI developed for FNET (Base Case 4)...	29
Figure 2-17: FDR data conditioning GUI developed for FNET (Base Case 5)...	29
Figure 2-18: FDR data conditioning GUI developed for FNET (Base Case 6)...	30
Figure 2-19: FDR data conditioning GUI developed for FNET (Base Case 7)...	30
Figure 2-20: FDR data conditioning GUI developed for FNET (Base Case 8)...	31
Figure 2-21: FDR data conditioning GUI developed for FNET (Base Case 9)...	31
Figure 3-1: Example of seismograph readings taken at a single observatory station [53].	36
Figure 3-2: Jeffreys-Bullen travel-time curves [54].	38
Figure 3-3: Jeffreys-Bullen plot of wave paths [54].	39
Figure 3-4: Linear triangulation of base case 1.	42
Figure 3-5: Linear triangulation of base case 2.	43
Figure 3-6: Linear triangulation of base case 3.	43
Figure 3-7: Linear triangulation of base case 4.	44
Figure 3-8: Linear triangulation of base case 5.	44
Figure 3-9: Linear triangulation of base case 6.	45
Figure 3-10: Linear triangulation of base case 7.	45

Figure 3-11: Linear triangulation of base case 8.	46
Figure 3-12: Linear triangulation of base case 9.	46
Figure 3-13: Plot of LS error versus assumed continuum speed.	49
Figure 3-14: Cartoon of hypocenter triangulation by increasing radii.	50
Figure 3-15: Flow chart for algorithm to find hypocentral loci.	51
Figure 3-16: Hypocentral loci that meet constraints of base case 1.	52
Figure 3-17: Hypocentral loci that meet constraints of base case 2.	53
Figure 3-18: Hypocentral loci that meet constraints of base case 3.	53
Figure 3-19: Hypocentral loci that meet constraints of base case 4.	54
Figure 3-20: Hypocentral loci that meet constraints of base case 5.	54
Figure 3-21: Hypocentral loci that meet constraints of base case 6.	55
Figure 3-22: Hypocentral loci that meet constraints of base case 7.	55
Figure 3-23: Hypocentral loci that meet constraints of base case 8.	56
Figure 3-24: Hypocentral loci that meet constraints of base case 9.	56
Figure 4-1: Newton triangulation of base case 1.	61
Figure 4-2: Newton triangulation of base case 2.	61
Figure 4-3: Newton triangulation of base case 3.	62
Figure 4-4: Newton triangulation of base case 4.	62
Figure 4-5: Newton triangulation of base case 5.	63
Figure 4-6: Newton triangulation of base case 6 (enlarged for clarity).	63
Figure 4-7: Newton triangulation of base case 7.	64
Figure 4-8: Newton triangulation of base case 8 (enlarged for clarity).	64
Figure 4-9: Newton triangulation of base case 9.	65
Figure 4-10: Plot of Newton error versus assumed continuum speed.	67
Figure 4-11: Flow chart of algorithm to find hypocentral coordinates using gradient search.	69
Figure 4-12: Gradient Search triangulation of base case 1.	70
Figure 4-13: Gradient Search triangulation of base case 2.	71
Figure 4-14: Gradient Search triangulation of base case 3.	71
Figure 4-15: Gradient Search triangulation of base case 4.	72
Figure 4-16: Gradient Search triangulation of base case 5.	72
Figure 4-17: Gradient Search triangulation of base case 6.	73
Figure 4-18: Gradient Search triangulation of base case 7.	73
Figure 4-19: Gradient Search triangulation of base case 8.	74
Figure 4-20: Gradient Search triangulation of base case 9.	74
Figure 4-21: Gradient Search error versus number of active constraints.	76
Figure 5-1: Performance of triangulation methods.	79

List of Tables

Table 1-1: FDR unit location data.	6
Table 1-2: Base case location data.....	7
Table 2-1: Base case perturbation wave form arrival data (Cases 1-3).	32
Table 2-2: Base case perturbation wave form arrival data (Cases 4-6).	33
Table 2-3: Base case perturbation wave form arrival data (Cases 7-9).	34
Table 3-1: Numerical results for linear triangulation method.	47
Table 4-1: Numerical results for Newton-Raphson triangulation method.....	66
Table 4-2: Numerical results for Gradient Search triangulation method.....	75

Chapter 1

Introduction

1.1 *Scope*

The purpose of this thesis is to take a first step in the direction of developing a reliable and comprehensive method of triangulating the approximate, yet accurate, location of disturbances that affect power system frequency. The stage for this study is set in the Eastern United States Interconnected System (EI). Within the EI there are ten Frequency Disturbance Recorders (FDRs) from which data is drawn to drive the analysis contained within this report. Since utmost importance is placed upon producing results that can be practically implemented with computational efficiency, no simulated data has been introduced into this study. Hence the results conveyed in this report are relevant for immediate implementation. Information from nine generator trip events with known generator locations provides the data that drives this study.

A first effort is made to properly condition the frequency data from the nine events under study. The noisy nature of the data received from the FDRs demands intermediate steps between observation, calculation, and triangulation. Using robust methods, steps are taken to estimate the true value of frequency at each FDR location during generator trips and, consequentially, other frequency perturbing events.

The prevailing effort in this study, however, is the mathematical triangulation of the generator trip locations solely from frequency data observed by the ten relevant FDRs in the EI. Concepts taken from both mathematics and seismology are used in an attempt to compose a solution that deals with the vastly dynamic and heterogeneous nature of the EI. Methods suited for solving both linear and non-linear sets of equations are used for the purpose of triangulation.

The prevention of catastrophic power system failures relies on the timely triangulation of events such as generator trips [1].

1.2 *Electromechanical Wave Propagation*

Scientists and engineers have noted that frequency perturbations apparently travel as electromechanical waves [2-4]. From the inception of a disturbance with nebulous location (herein categorized as either a generator trip, load trip, or transmission line trip of any mentionable magnitude), there begins a progression, in this case within the EI, of phase angle disturbances that travel with finite speed. Having been studied with great acuity, this distribution of phase angle disturbances in time and space can be viewed as a wave front of varying frequency perturbations that progresses outward from its point of origin [5, 6] . The mechanics and details of how this wave travels have also been studied at great length and it has been suggested that there exist at least three elements that dictate speed of its progression. These are surrounding mesh network impedance, rotational inertia of nearby machines, and location of disturbance [2, 7]. Others have modeled the behavior of these frequency perturbation waves as traveling through a medium of constant elasticity or a continuum [8-10]. In the effort to triangulate the origin of a generator trip, we are acutely interested in how this frequency perturbation wave travels. If we can measure and record the progression of this wave in much the same manner as seismologists track the progression of earthquakes, then it is hoped that we can triangulate the hypocenter of our electrical disturbance in the vein of how a seismologist would triangulate the hypocenter of an earthquake [11].

1.3 The Role of FNET

Since the creation of the phasor measurement unit (PMU), the FDR, and the associated FNET, many have remarked upon the importance and use of such global positioning system (GPS) time-synchronized devices [2, 12-16]. As mentioned earlier, we are not basing this study on the constrained world of computer aided simulations. Rather, we are basing our calculations on data recorded by ten FDRs each of which are a part of FNET. The following figure shows the placement of the FDRs used in this study. Each unit location is marked with a blue dot. It is, however, important to note that not all FDRs currently in operation are listed. Only the units that provided data for the nine generator trip cases used in this study are presented.

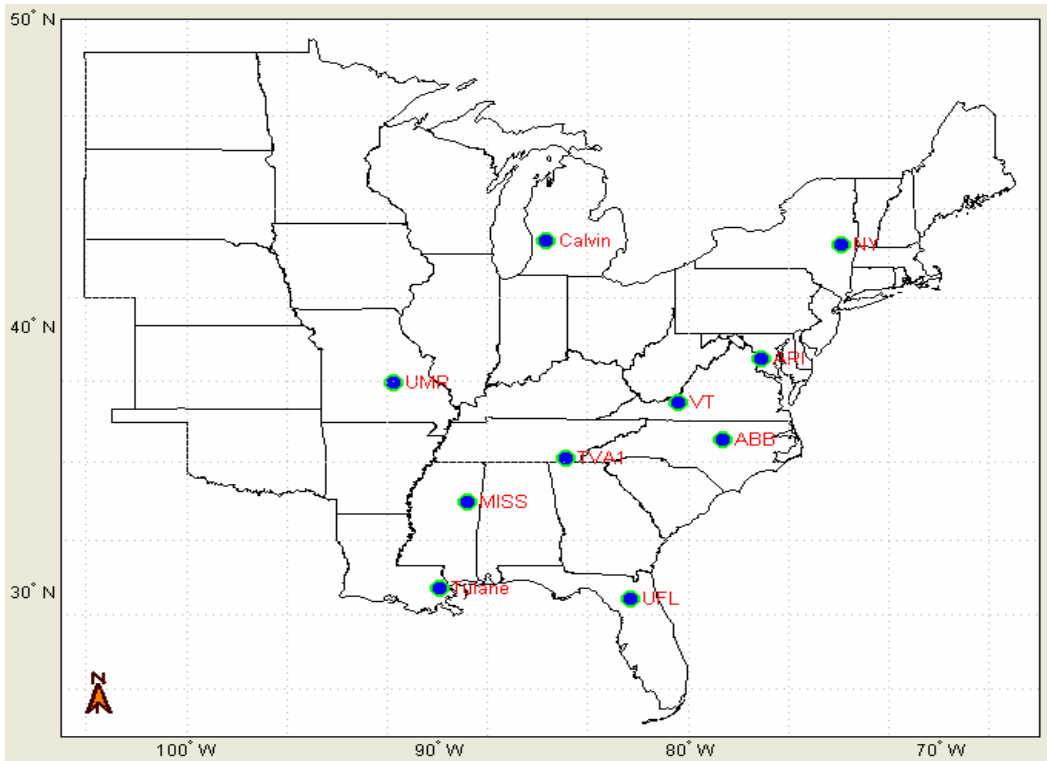


Figure 1-1: Location of FDRs used in this study.

There has been a significant push recently to use PMU and FDR data for the purpose of predictive control for frequency stability [17]. Conversely, the FDRs that compose FNET will be used in the opposite manner for this study. We will use these units for the purpose of triangulation after a contingency. It is hoped that excerpts from this study will be helpful in the eventual release of an automatic, fast-acting, post-mortem, event triangulation utility. To that end, FNET will provide the infrastructure for the estimator we wish to build. Although the location estimation effort is quite different from the familiar area of power system state estimation, there are a few common principles that are used in both. The number of units, unit location, and conditioning of unit data are important factors to both areas [18-21]. As the development and furtherance of FNET is yet a growing science, the placement of FDRs throughout the EI is sparse and spotted. Furthermore, there is also no control within our domain over the power system events that serve as the base cases for this study. Hence this first effort is made with the knowledge that our information is scarce in the hope that at least a foundation can be laid in the area of power system event triangulation.

With our set of ten FDRs, we have observed nine events. Each event, however, was not observed by every FDR in our set of ten. Rather, each of these nine events was measured only with a subset of our ten FDR units. These events that serve as base cases are important in the development of a triangulation algorithm because they give us insight about the behavior of the EI in an event. Furthermore, for each base case we have confirmation on general location and approximate trip amount [22-24]. These base cases do not provide us, however, with information on the exact tripping times of the generators in question. The following figure shows the locations of the base cases that have been used for this study in relation to the FDR units. Base cases are plotted with squares and the FDRs are also plotted as blue dots.

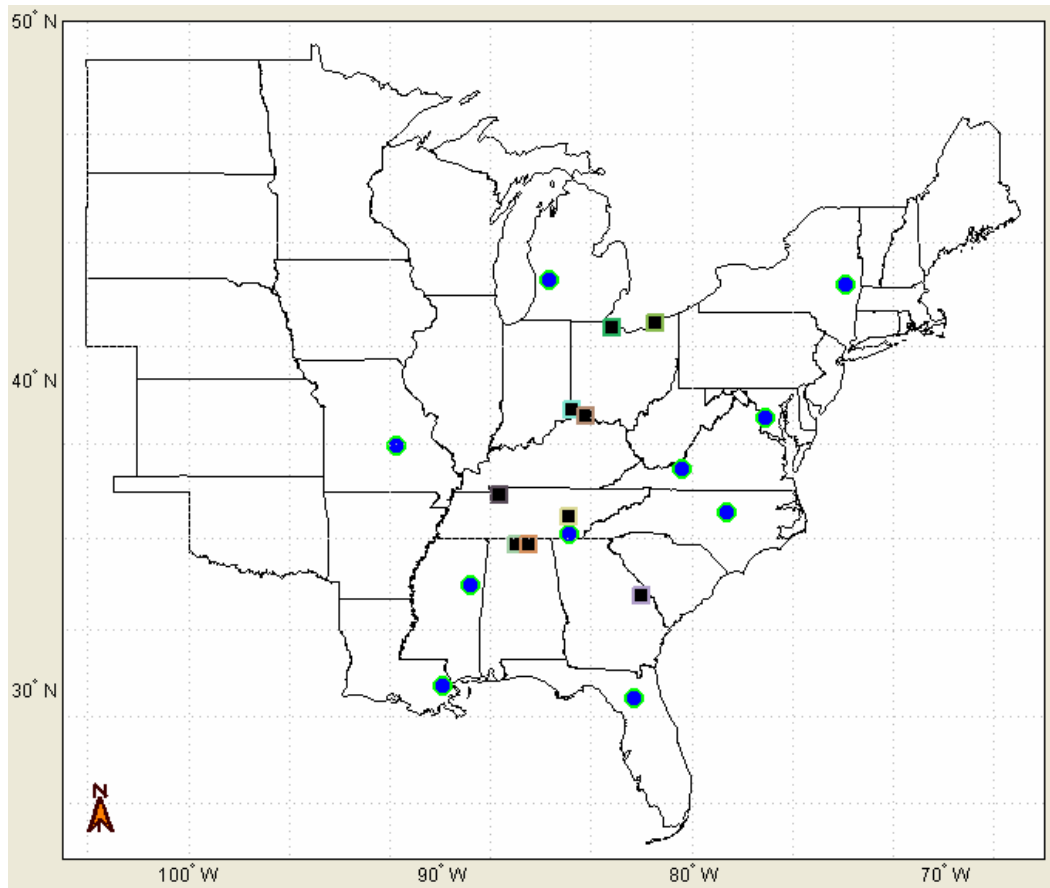


Figure 1-2: Location of base cases used in this study.

The following tables list pertinent information about both the FDR units and the base cases used in this study.

Table 1-1: FDR unit location data.

<i>Unit Number</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>6</i>	<i>7</i>	<i>9</i>	<i>11</i>	<i>17</i>	<i>20</i>
Unit Name	NY	UMR	ARI	VT	ABB	MISS	UFL	Calvin	Tulane	TVA1
NERC Region²	NPCC	MAIN/SERC	SERC	ECAR/SERC	SERC	SERC	FRCC	ECAR	SERC	SERC
Latitude³	42.8018	37.9487	38.8210	37.2327	35.8220	33.4567	29.6742	42.9613	30.0658	35.1313
Longitude³	-73.9281	-91.7658	-77.0862	-80.4284	-78.6587	-88.8222	-82.3363	-85.6557	-89.9313	-84.8750

² As determined by <http://www.nerc.com>

³ As determined by the U.S. Gazetteer, <http://www.census.gov/cgi-bin/gazetteer>

Table 1-2: Base case location data.

<i>Case Number</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
Date	8/4/2004	9/19/2004	11/23/2004	1/26/2005	2/11/2005	3/21/2005	4/2/2005	4/22/2005	4/29/2005
Plant Name	Davis Besse	Watts Bar	Browns Ferry	East Bend	Browns Ferry	Cumberland	Eastlake	Zimmer	Votgle-Wilson
Nearest Town	Oak Harbor	Spring City	Athens	Rabbit Hash	Athens	Cumberland City	Eastlake	Moscow	Waynesboro
State	Ohio	Tennessee	Alabama	Kentucky	Alabama	Tennessee	Ohio	Ohio	Georgia
NERC Region⁴	ECAR	SERC	SERC	ECAR	SERC	SERC	ECAR	ECAR	SERC
Latitude⁵	41.5116°	35.6874°	34.7860°	30.0324°	34.7860°	36.3822°	41.6596°	38.8603°	33.0900°
Longitude⁵	-83.1467°	-84.8641°	-86.9599°	-84.7414°	-86.9599°	-87.6440°	-81.4306°	-84.2285°	-8201.36°
FDR Set	{2,3,4,6,7,9}	{1,2,3,4,6,7}	{2,3,4,6,7,11}	{2,3,4,6,7,11}	{2,3,4,6,7,9,11}	{2,3,4,6,7,9,11,17,20}	{2,3,4,6,7,9,11,17,20}	{2,3,4,6,7,9,11,20}	{2,3,4,6,7,9,11,17,20}

⁴ As determined by <http://www.nerc.com>

⁵ As determined by the U.S. Gazetteer, <http://www.census.gov/cgi-bin/gazetteer>

The base case events in the above table are observed by looking for departures from 60 Hz (or nominal frequency) in the frequency data supplied by the available and relevant FDR units. A trigger device has already been developed that looks for a 5 mHz departure from nominal frequency over a period of 4 s or longer [2, 23]. Once an event has been detected and triggered, the actual electromechanical wave front is visible in the time-staggered frequency decline observed by the FDRs. However, the drop in frequency is not detected by every unit at the same time. There is a definite delay. This idea leads to the proposal of this thesis: There exists a method or methods that can use the delay between the inception of an observable event and the detection thereof solely by frequency examination to triangulate with reasonable accuracy the location of the inciting event. The following cartoon graphically explains the above stated.

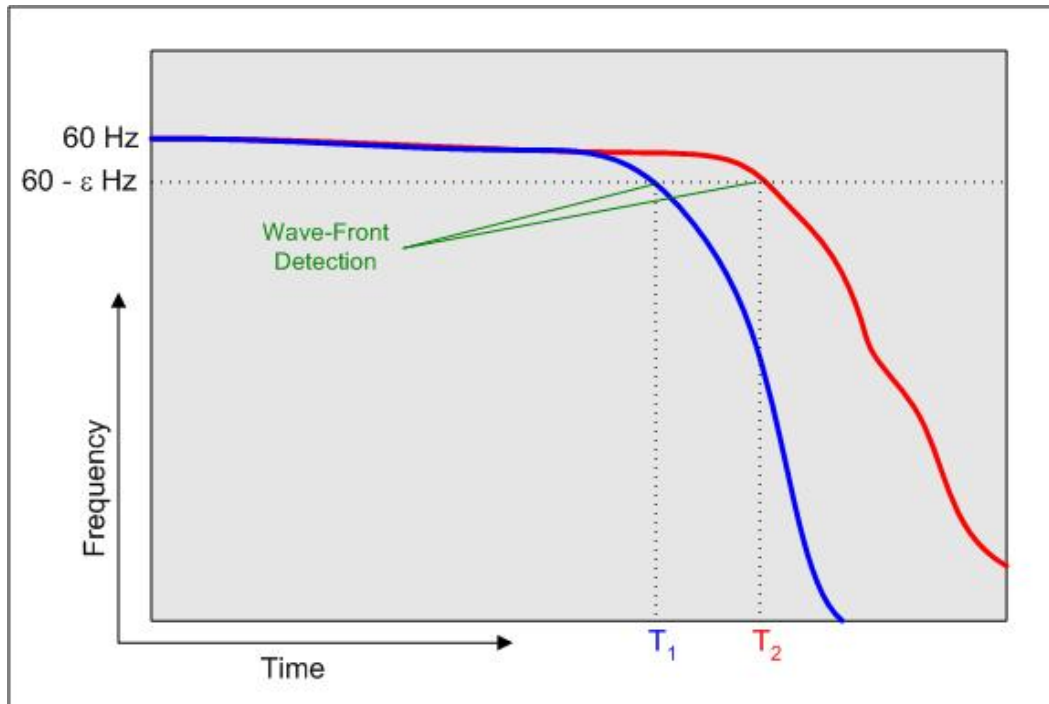


Figure 1-3: A cartoon of an example frequency event.

The above figure represents an imaginary event detected by two FDR units. The unit associated with the data plotted in blue was the first to observe the event at time T_1 . The second unit (data in red) observed this event at time T_2 . The detection times (T_1 and T_2) are determined in the following manner: First, a confidence threshold, ϵ , is determined. This threshold has been fixed at 0.0025 Hz. This is justified by both legacy and the idea that 0.0025 Hz is half of the event detection threshold of 0.005 Hz (from above) [2, 23]. In this event, nominal frequency just happens to be 60 Hz. However, the field measurements rarely render nominal frequency exactly at 60 Hz. Therefore the second step is to form $f_{threshold}$, the frequency at which we shall note the event detection time for each unit. This quantity can be defined as:

$$f_{threshold} = f_{Nominal} - \epsilon$$

Finally, the detection times are documented as the time when the measured frequency data crosses the line determined by $f_{threshold}$.

This thesis presents another, more scientifically consistent method of determining the disturbance wave front arrival times at the end of Chapter 2. This method is not based on the selection of an arbitrary threshold.

1.4 Organization of the Study

Chapter 2 describes the data conditioning that makes the steps in Section 1.3 possible. Since the data delivered by the FDR units is not as smooth as the ideal data in the cartoon of the previous section, the data must undergo a conditioning process as later described. Section 2.1 describes in particular terms the problems that exist with FDR data. Section 2.2 covers the current method of dealing with the noisy nature of FDR data and explains why this method is not sufficient. Section 2.3 submits two robust methods taken from the area of statistics that de-noise the FDR data and remove unwanted outliers. Lastly in

Chapter 2, Section 2.4 summarizes the main points of robust outlier detection and removal while also displaying the graphical user interface (GUI) that has been developed expressly for the purpose of smoothing data collected from FDRs.

Chapter 3 discusses a linear technique conventionally used for the purpose of seismic hypocenter triangulation. In Section 3.1, principles from elementary seismic location theory are delineated. The solution (or best approximation) of the linear triangulation equation set proposed in Section 3.1 is discussed in terms of the least-squares problem in Section 3.2. Section 3.3 sequentially adds constraints as determined by the detection times associated with each FDR. This process shows why a linear solution method alone is prone to failure. Finally, the results of Chapter 3 are discussed in Section 3.4.

Chapter 4 discusses the solution of non-linear sets of distance equations and the minimization of constraint equations using the Newton's and gradient search methods, respectively. The results of these processes are plotted and discussed in Sections 4.1 and 4.2. Lastly in Chapter 4, Section 4.3 summarizes the contributions of non-linear theory to triangulation of power system events.

Chapter 5 concludes the thesis with a summary of findings from the research into triangulation thus far. Finally, a proposal for future research is submitted in the last part of Chapter 5.

Chapter 2

Data Conditioning and Curve Fitting

2.1 *The Nature of FDR Data*

In order to use data collected from FNET, one must first understand its nature. The data is by nature riddled with noise and outliers. We do not consider these noisy FNET data to be mistake-ridden, but rather we consider these data to be error-ridden. If indeed the data were mistake-ridden, then we could make improvements by simply eliminating the source of mistakes and no statistical treatment would be necessary. However, according to the latest science available to the FNET research group, we consider this data to be mistake free. The frequency data, is perhaps error-ridden, meaning that in the data recorded by each FDR, there are measurements that depart from the true value of the frequency [25]. What is the source of the error that renders the FDR measurements so noisy? Aside from the inescapable experimental errors that arise from the ever-present although minuscule measurement error, each FDR is prone to measuring noise from the utility's distribution network. A key benefit of FNET is that it relies upon FDRs that are, in general, installed to take measurements of frequency at the distribution level. Furthermore, these units are not in any manner protected or buffered from other devices also connected to the distribution network that may cause disturbances in voltage signals or otherwise degrade local power quality. Many have researched and explained the power quality problems that appear at the distribution level [26-29]. Although the FDRs read and measure the small disturbances at the distribution level, we do not wish to consider these relatively minute disturbances, that are not a true reflection of power system frequency, when analyzing large scale power system contingencies [12]. These local disturbances become to us as noise with possible outliers. The following figure is

a prime example of a set of problematic data recorded by an older FDR running a previous, now updated and replaced, firmware version.

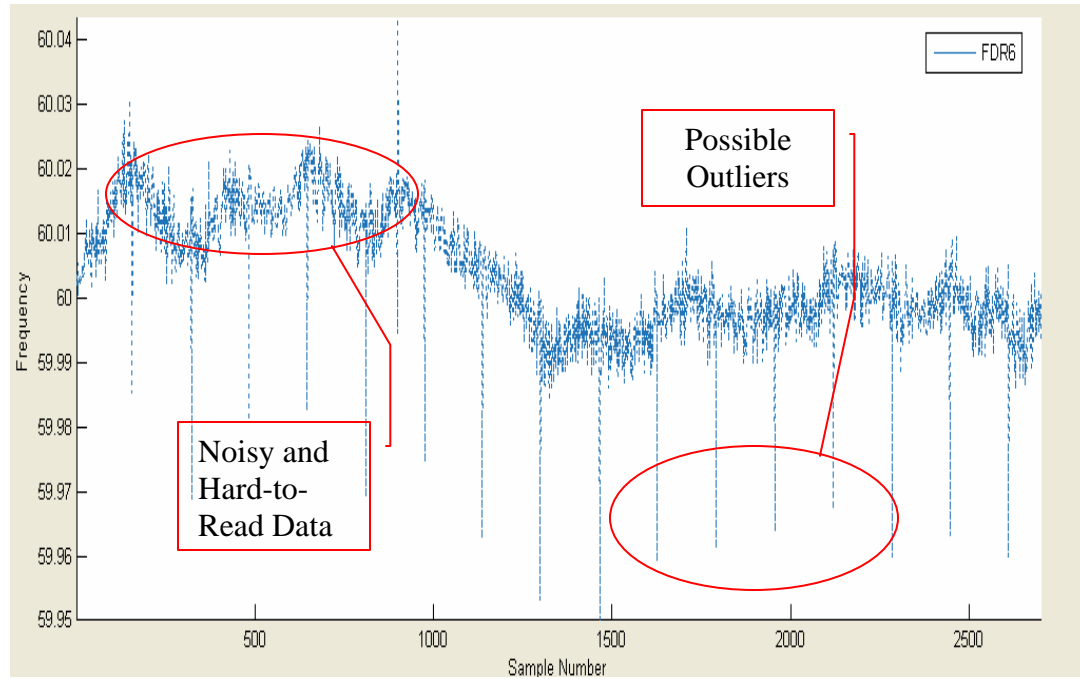


Figure 2-1: Problematic FDR data with signal quality issues highlighted.

From the above figure, it is obvious that the frequency waveform as measured at the distribution level is quite noisy and hard to read. The next sections discuss a few techniques used to estimate the true value of the frequency. These techniques draw ideas from the realms of conventional and robust statistics to deal with noisy data [24, 30-33]. An effort is made to keep the complexity of data conditioning to a minimum for the purpose of computational efficiency.

2.2 The Currently Used Data Smoothing Technique

The current method used by the FNET research group to estimate the true value of the frequency, f_t , measured by each individual FDR is the “moving average” or “moving mean” [23]. The moving mean is defined as follows [34]:

given a sequence $\{f_i\}_{i=1}^N$, an n -moving mean is another sequence $\{\bar{f}_i\}_{i=1}^{N-n+1}$ defined from f_i by taking the mean as:

$$\bar{f}_i = \frac{1}{n} \sum_{j=1}^{i+n-1} f_j$$

Thus the estimate of the true value of the frequency is defined as:

$$\hat{f}_t \equiv \{\bar{f}_i\}_{i=1}^{N-n+1}$$

Although the actual true value of the frequency f_t only exists in theory. This approach estimates f_t and we write this value as \hat{f}_t . The moving mean is actually a moving least squares estimator that minimizes the following objective function:

$$\phi(f_t) = \sum_{i=1}^n r_i^2 = \sum_{i=1}^n (f_i - f_t)^2$$

It is important to note that the least squares (LS) estimator is the maximum (ML) likelihood, or optimal, estimator of normally distributed data [35]. Normal, or Gaussian, distributed data has a probability density function of the form [36-38]:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

And can be viewed pictorially as:

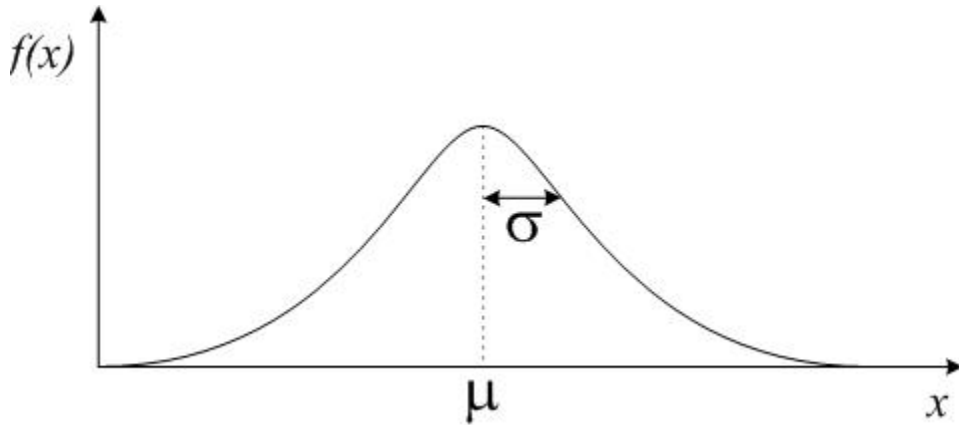


Figure 2-2: The Gaussian distribution.

The following figure shows the raw data, f and the estimate of the true value using the moving mean, \hat{f}_t .

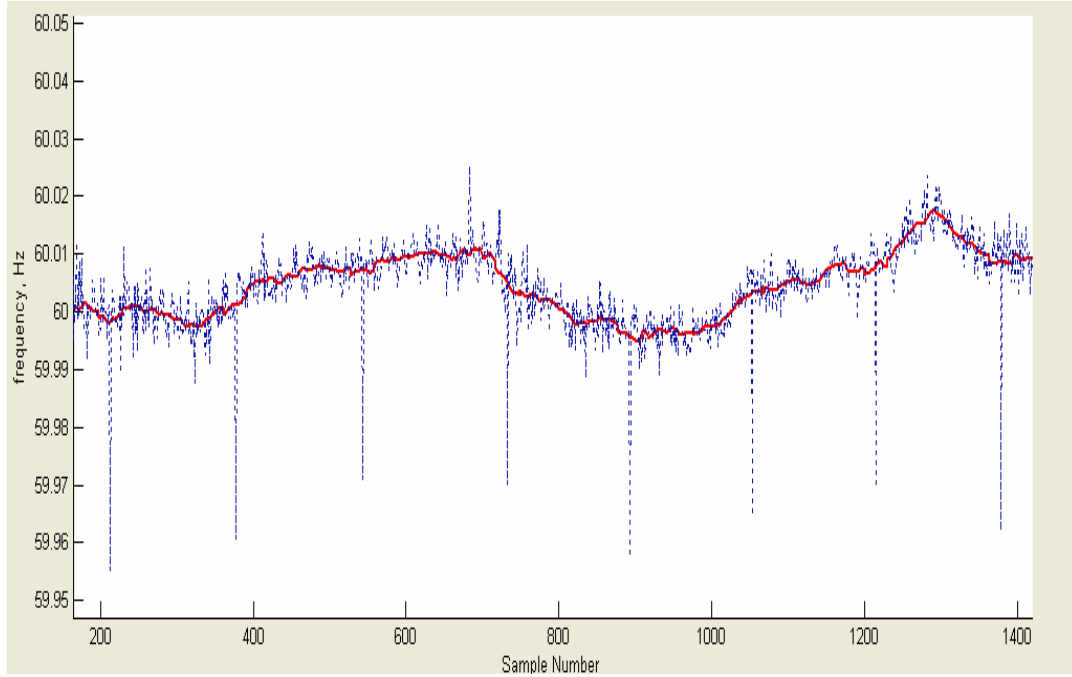


Figure 2-3: Raw FDR data plotted with moving mean.

The above figure shows both the raw FDR data plotted in blue and the moving mean plotted in red. A window size of 30 ($n = 30$) has been selected for the computation of the moving mean. From this figure the smoothing capabilities of the moving mean are evident and, at first glance, this method looks an excellent estimator of true value of the frequency. However, let us introduce a disturbance, not unlike one that might result from an actual scenario, and examine the moving mean closer. In the next figure, the actual data has been perturbed by adding eight data points at 59.9 Hz near one another. It is clearly visible, that the moving mean is not robust against these outliers because the mean is also perturbed. The mean is perturbed because we are breaking from Gaussianity by adding outliers

that are common to real data. As can be seen from figure 2-2, the Gaussian distribution has relatively thin tails and such outliers as added are not accommodated by the tails in the distribution. Therefore the LS estimator which is optimal at the Gaussian distribution will break down when non-Gaussian data is introduced.

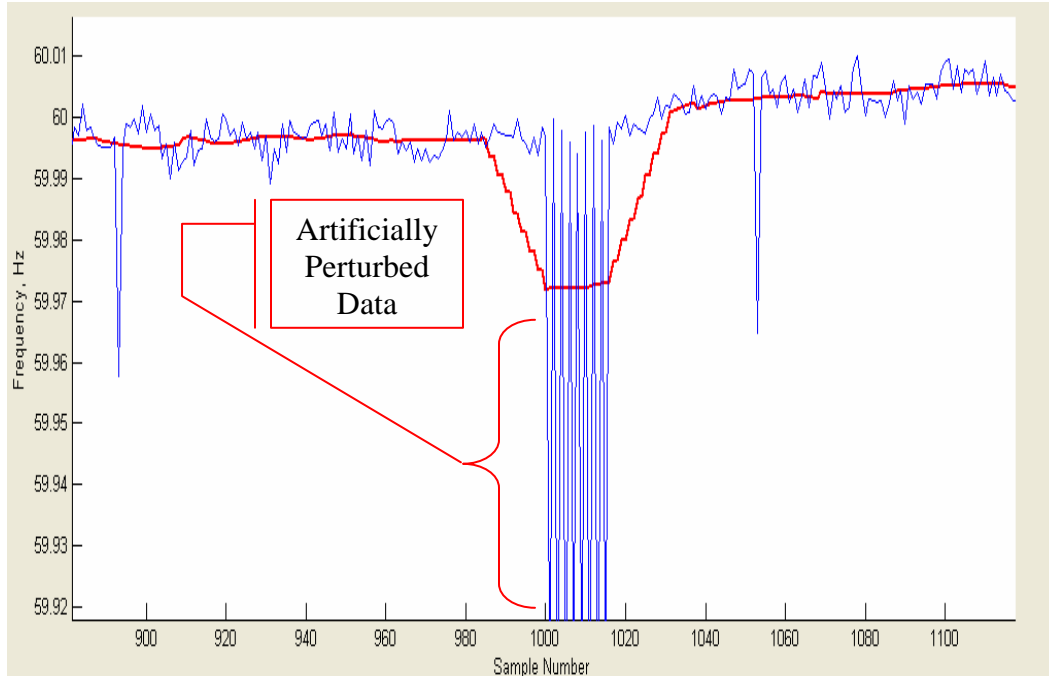


Figure 2-4: Raw perturbed data plotted with moving mean.

In the figure above, the data plotted in blue is the raw data and the data plotted in red is the moving mean with a window size of 30. Such a disturbance that lasts less than 2s is not uncommon at the distribution level [26]. We seek an estimator of f_t that is robust against such common occurrences of outliers. We turn from classical statistics to robust statistics for the solution [30, 31].

2.3 Two Robust and Computationally Efficient Techniques

The goal of this section is to present two techniques that are computationally efficient but also robust against outliers with the nature of those presented in the previous section. These two techniques are the moving median and polynomial curve-fit with outlier rejection.

2.3.1 The Moving Median

We define the moving median in much the same way as the moving mean. The concept of a moving window each encompassing 30 elements remains the same but the mean is replaced with the median. In closed form, the moving median is defined: given a sequence $\{f_i\}_{i=1}^N$, an n -moving median is another sequence $\{\bar{f}_i\}_{i=1}^{N-n+1}$ defined from f_i by taking the median as [34, 36, 39]:

$$\bar{f}_i = \text{median}[f_i, f_{i+1}, \dots, f_{i+n-1}]$$

The estimation of the true value of the frequency can then be defined as:

$$\hat{f}_t \equiv \{\bar{f}_i\}_{i=1}^{N-n+1}$$

The moving median is the ML estimator of the Laplacian or Double-Exponential distribution which minimizes the L_1 norm given by [35]:

$$\phi(f_t) = \sum_{i=1}^n |f_i|$$

And the probability density function of the Laplacian distribution is given by [35-38]:

$$f(x) = \frac{1}{2a} e^{-\frac{|x-b|}{a}}$$

The following figure presents the Laplacian distribution pictorially. Notice that the Laplacian distribution has thicker tails than those of the Gaussian distribution.

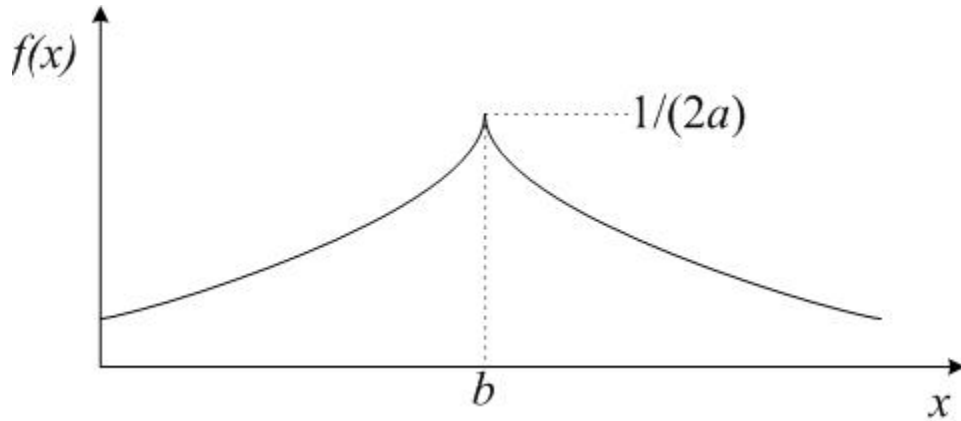


Figure 2-5: The Laplacian Distribution

The following figure shows a sample of perturbed FDR data plotted along with both the moving mean and the moving median. The raw data is plotted in blue, the moving mean in red, and the moving median in green. From this figure the robustness of the moving median becomes evident as the moving median is not at all perturbed by the simulated disturbance. The thicker tails in the Laplacian distribution allow for the greater possibility of outliers and thus the moving median does not break down when outliers are present. The “ride-through” capability of the median is quite desirable for our application, thus we shall consider the moving median as a viable computationally efficient replacement of the moving mean. Nonetheless, over the few-second period of the following figure, the moving median still does not seem smooth enough. Hence, we turn to the world of curve fitting.

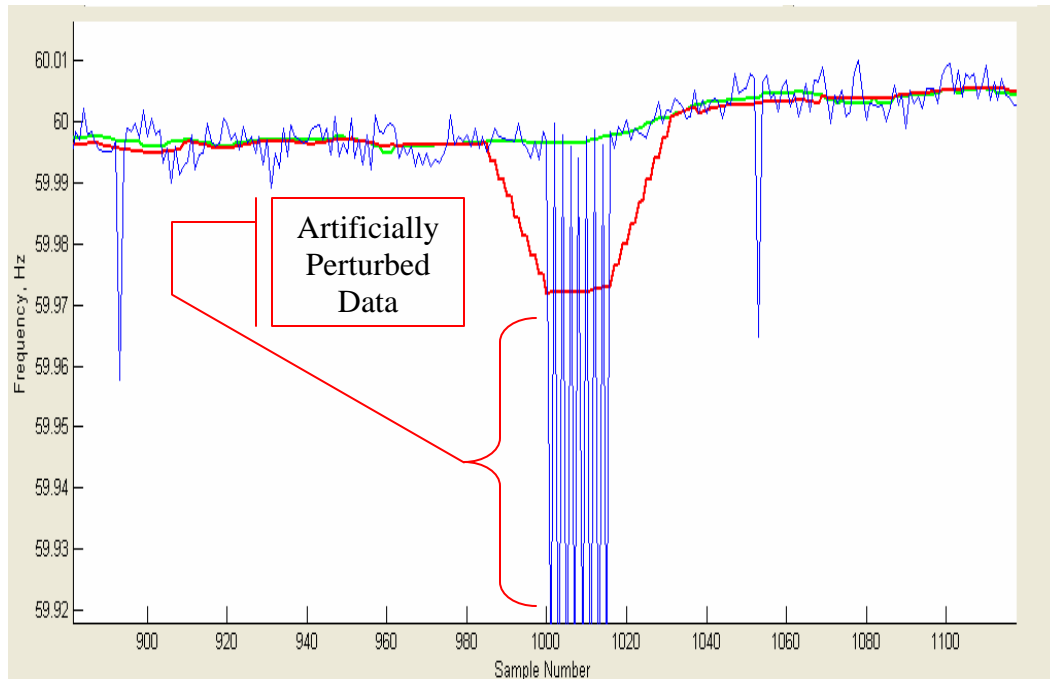


Figure 2-6: Raw perturbed data with moving mean and moving median.

2.3.2 Curve Fitting with Outlier Rejection

Having examined the moving mean and the moving median and yet desiring even more data smoothing capabilities in the estimation of the true value of the frequency, we resort to curve fitting. In the next section, for the purpose of determining the wave front arrival times, we shall also desire a smooth wave form in order to evaluate the derivative. For consistency, the FDR data that is to be fitted is placed in a ~10-20s window. We desire the point at which the electromechanical wave was noticed to be placed near the center of this window. This step places a relatively flat section of data on either side of the disturbance in question. The figure below explains this step graphically.



Figure 2-7: Sample FDR data recording generator trip.

Before we begin the curve fitting process, it is important to identify and reject any outliers that might be contained within the sample of original data. The following steps are used in the outlier rejection process and are taken from the field of robust statistics [32, 33, 40].

1. The sample to be fit is broken down into sub-samples of window size n . This method of outlier detection is highly sensitive to the size of the sub-sample windows and thus after due experimentation we choose a window size of 30 ($n = 30$).

The following steps are repeated for each of the sub-samples created in Step 1.

2. The median is evaluated as follows:

$$\hat{z}_{median} = median[z_1, z_2, \dots, z_n]$$

3. The median-absolute-deviation or MAD is calculated as:

$$MAD = 1.4826 \times b_n \times median|z_i - \hat{z}_{median}| \text{ for } i = 1, 2, \dots, n$$

4. The bias factor, b_n , of Step 3 is calculated as:

$$b_n = \frac{n}{n - 0.8}$$

5. Standardized residuals are formed from the median and MAD of the sub-sample in the following manner:

$$|r_{si}| = \frac{|z_i - \hat{z}_{median}|}{MAD}$$

6. The elements associated with residuals with a value larger than 2.57 are classified and discarded as outliers.

The following figure shows a set of frequency data with outliers having been identified using the previous steps (outliers identified with red asterisks).

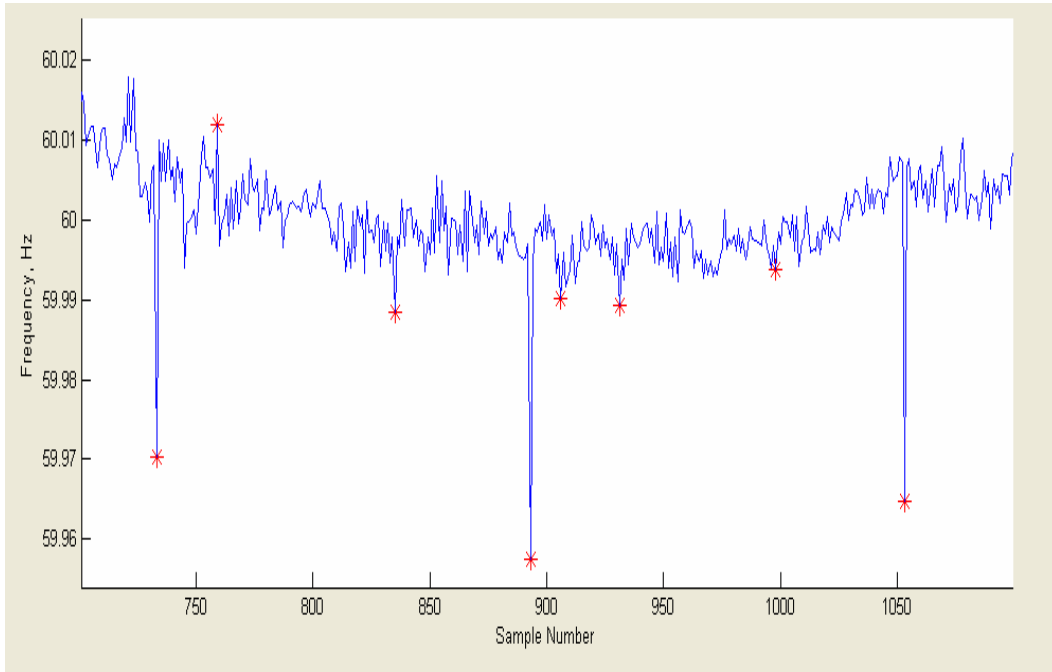


Figure 2-8: Sample FDR data with outliers identified.

Having chosen the sample of data to be fitted and having detected the outliers, we can now proceed with fitting the data with a polynomial curve. Since the current FDRs record the frequency every 0.1 s, ten seconds worth of

data amounts to 100 samples. By experimentation we can observe that, with the nature of our data and with 100 samples, a desirable close but smooth fit can be obtained by a seventh order polynomial. This equation describes in closed form what we desire:

$$f_t = \alpha_7 t^7 + \alpha_6 t^6 + \alpha_5 t^5 + \alpha_4 t^4 + \alpha_3 t^3 + \alpha_2 t^2 + \alpha_1 t + \alpha_0$$

However for the sake of versatility, we generalize:

$$f_t = \alpha_m t^m + \dots + \alpha_2 t^2 + \alpha_1 t + \alpha_0$$

We use the following method to calculate the coefficients to be used in the curve fitting. These ideas come from classical curve fitting and matrix theory [35, 39, 41-43].

1. We first define the matrix expression that accommodates each sample within the sample set of n where $n = 100$ with a polynomial of order m as follows:

$$\underline{F} = \underline{H}\underline{\alpha}$$

2. \underline{F} , \underline{H} , and $\underline{\alpha}$ are defined as:

$$\underline{F} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix} \quad \underline{H} = \begin{bmatrix} t_1^m & t_1^{m-1} & \dots & t_1^2 & t_1 & 1 \\ t_2^m & t_2^{m-1} & \dots & t_2^2 & t_2 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ t_{n-1}^m & t_{n-1}^{m-1} & \dots & t_{n-1}^2 & t_{n-1} & 1 \\ t_n^m & t_n^{m-1} & \dots & t_n^2 & t_n & 1 \end{bmatrix} \quad \underline{\alpha} = \begin{bmatrix} \alpha_m \\ \alpha_{m-1} \\ \vdots \\ \alpha_2 \\ \alpha_1 \\ \alpha_0 \end{bmatrix}$$

3. The matrices from Step 2 are altered such that any elements that were identified as outliers along with the associated time values are not included.
4. In all likelihood, the matrices from Step 2 form an over-constrained system. Therefore, in general, there will be no exact solution for the α -coefficients. The pseudo-inverse can be used, however, to fit the data

with the proper coefficients in the least squares sense. By definition, the pseudo-inverse is the solution operator for the minimum-norm least squares problem [44]. Hence we form the pseudo-inverse, \underline{H}^\dagger , of \underline{H} :

$$\underline{H}^\dagger = (\underline{H}^t \underline{H})^{-1} \underline{H}^t$$

5. We can then use the pseudo-inverse to solve for the α -coefficients:

$$\underline{\alpha} = \underline{H}^\dagger \underline{F}$$

6. Finally, we use these coefficients to plot the m^{th} , in our case seventh, order polynomial.

The following figure displays the raw data plotted along with both the moving median and the seventh order polynomial fit. The raw data is plotted in blue, the moving median in green, and the fitted curve in red.

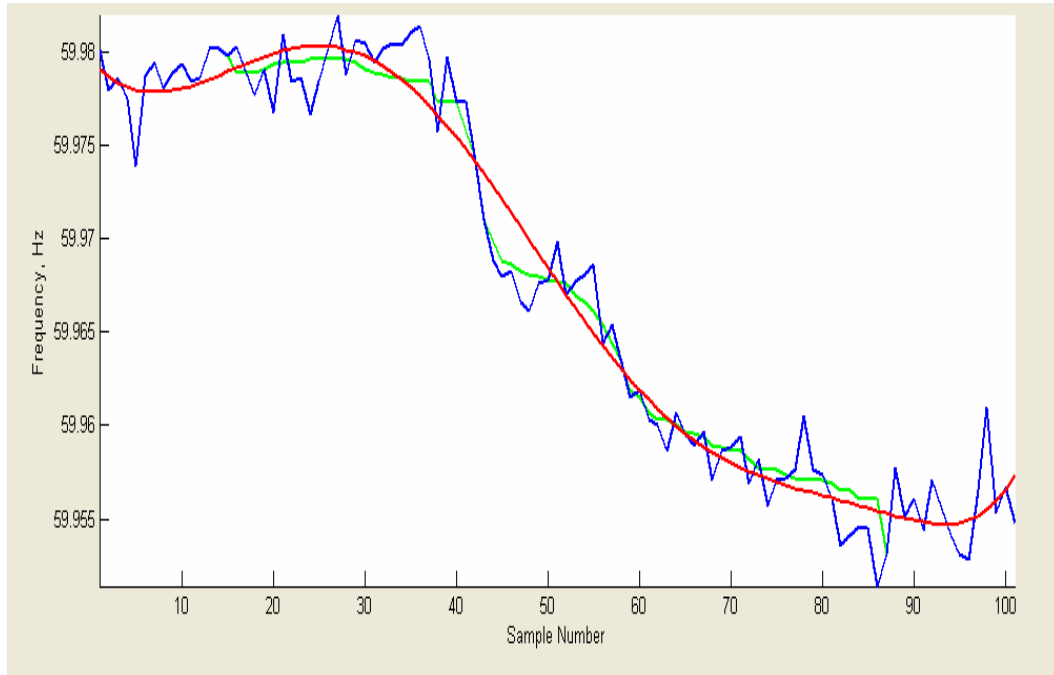


Figure 2-9: Raw FDR data plotted with moving median and fitted curve.

2.4 Summary

The data conditioning discussed in this chapter is beneficial because it gives us a non-relativistic method with which to determine the actual arrival time of the frequency disturbance waves. We depart from the classically used method described in Section 1.3 because it sets an arbitrary threshold by which every frequency waveform is analyzed. This old method might work if we could be assured that every frequency disturbance wave, as measured by the FDRs, has the same magnitude of perturbation and the same waveform shape. This ideal is however foolish to assume. The following figure gives an illustration of why this is so. The static threshold is represented by the dotted line and three imaginary frequency waveforms are plotted. There is no accommodation for waveform shape or magnitude with a static threshold. If, as in the case below, one or a few waveforms is shaped differently from the prevailing majority, the arrival time for the unusual waveforms might be assessed incorrectly without an objective arrival time detection algorithm.

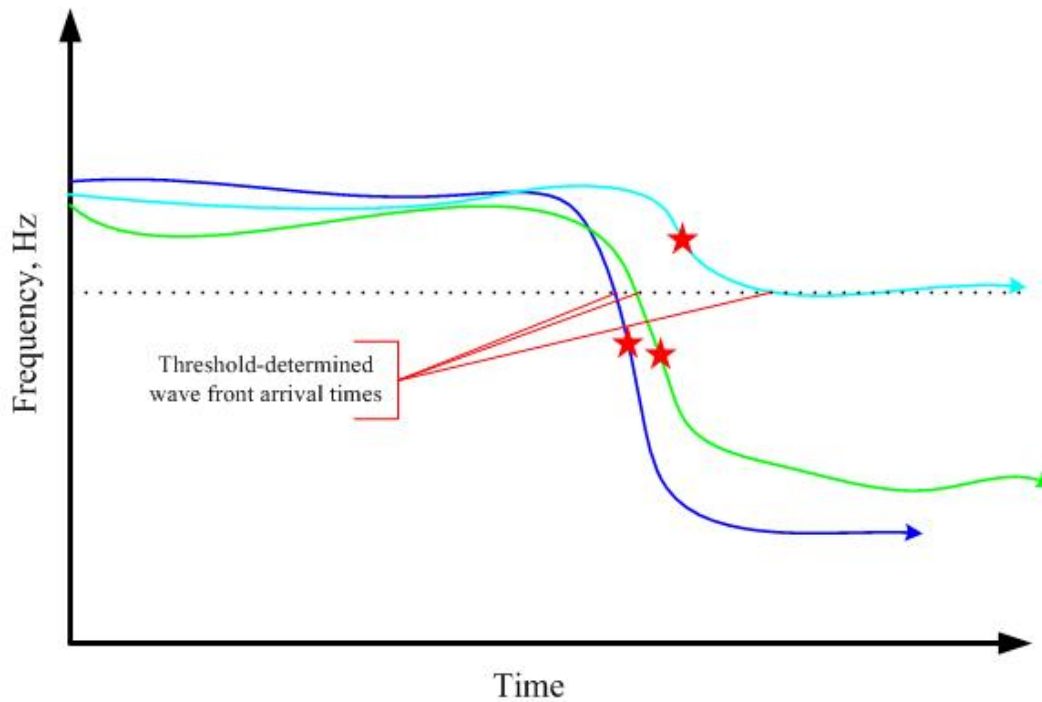


Figure 2-10: Disturbance wave detection error with static threshold.

Yet another problem with the static threshold method of determining the arrival time of frequency disturbance wave fronts is the fact that several thresholds must be determined in order to classify the event properly. What if, for instance, the contingency was load rejection as opposed to a loss of generation? An upper threshold would be necessary in this case. To avoid this problem along with the others, let us define the frequency perturbation arrival/measurement time as the moment when the frequency waveform is changing the most (i.e., the maximum value of the magnitude of the slope). The red stars in the above graph represent these moments. This new wave front detection method is yet another reason to use the curve fitting method instead of the moving median. Referring back to figure 2-9, we examine the magnitude of the derivative for both the moving median and the fitted curve. The moving median derivative is plotted in red and the fitted curve derivative in blue. The derivative of the moving median

is quite jumpy compared to the derivative of the fitted curve. Ignoring the derivative at the boundary, there is a definite maximum in the magnitude of the derivative of the fitted curve. The maximum value of the magnitude of the derivative of the moving median could be much more difficult to assess. This problem is made evident for this case (and in general) by the spikes visible in the following figure.

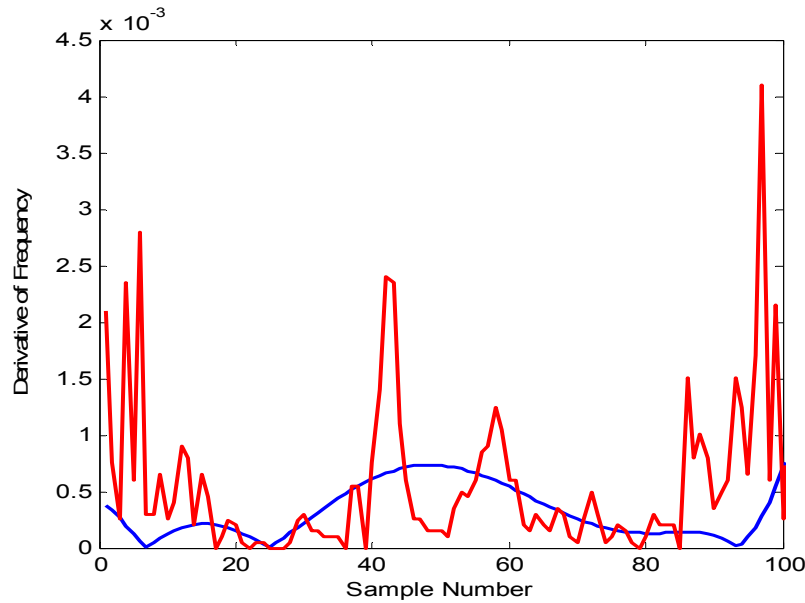


Figure 2-11: Magnitude of frequency derivative of moving median and fitted curve (see figure 2-9).

Figure 2-12 shows where the old static threshold method and the new slope-based method identify the official measurement time of the electromechanical wave. Since at this juncture, the new slope based wave front detection algorithm is not designed for use in fast-action closed loop control, the slight time delay, Δt , is irrelevant.

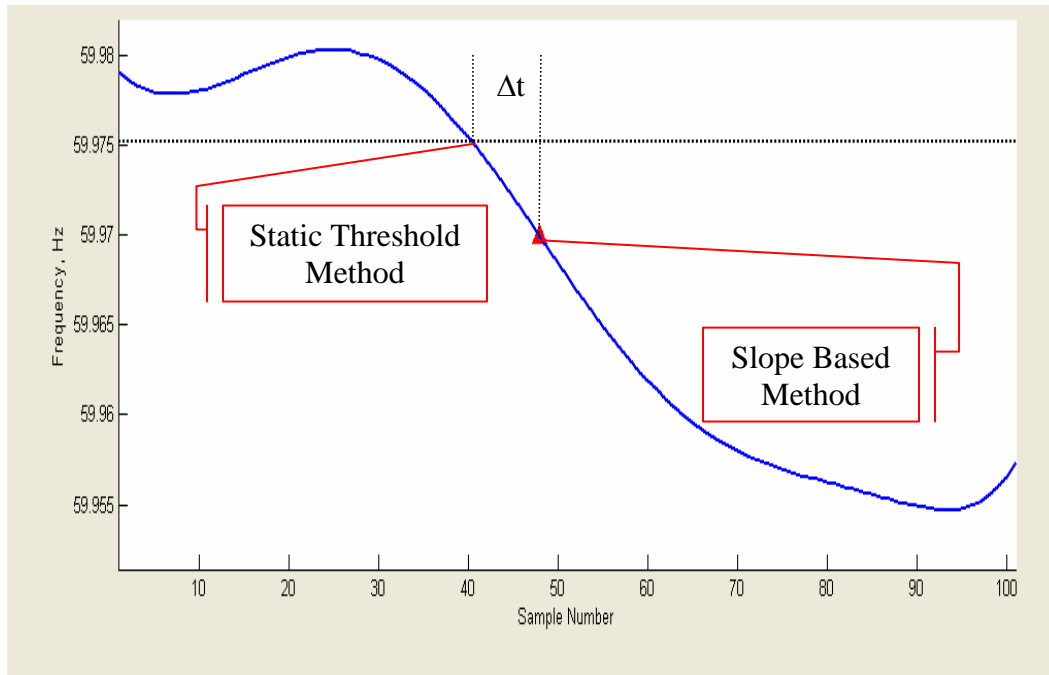


Figure 2-12: Detection times for static threshold and slope-based methods (see figure 2-8).

As a part of the work done in the accomplishment of this thesis, a GUI was developed expressly for the purpose of fitting FDR data. The figures below show the GUI with data plotted for each first base case. Notice that the placement of red triangles plotted on each curve represents the set wave front detection time for that particular curve. Table 2-1 shows all pertinent base case information including the recorded wave front detection times.

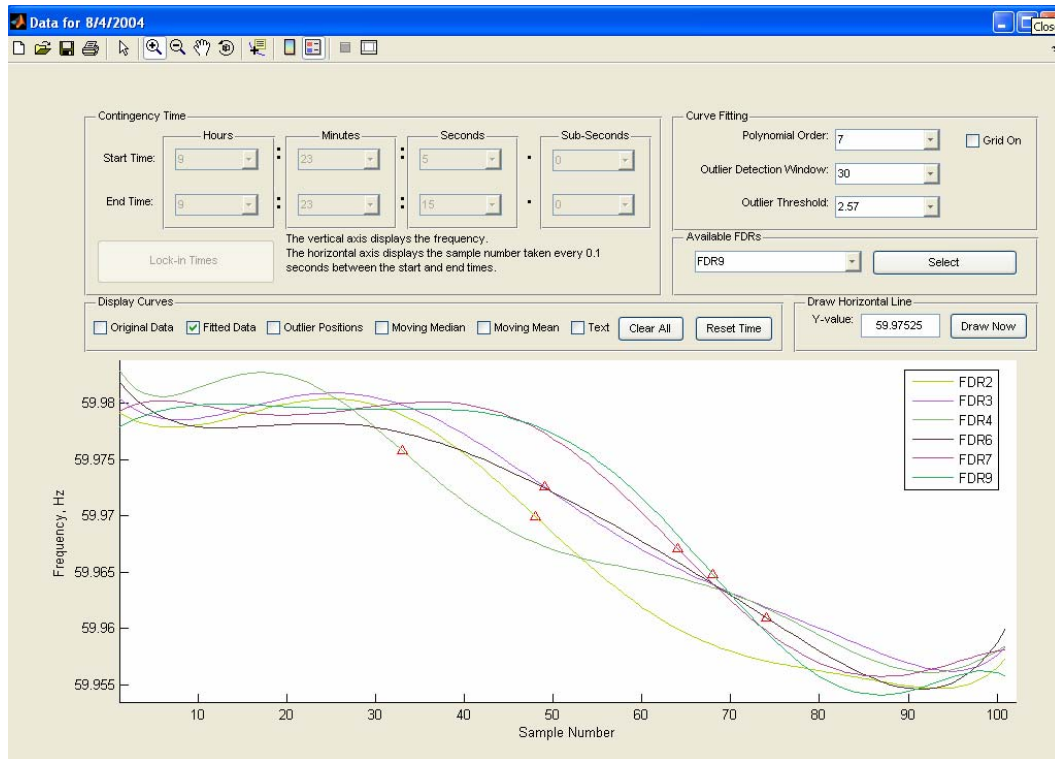


Figure 2-13: FDR data conditioning GUI developed for FNET (Base Case 1).

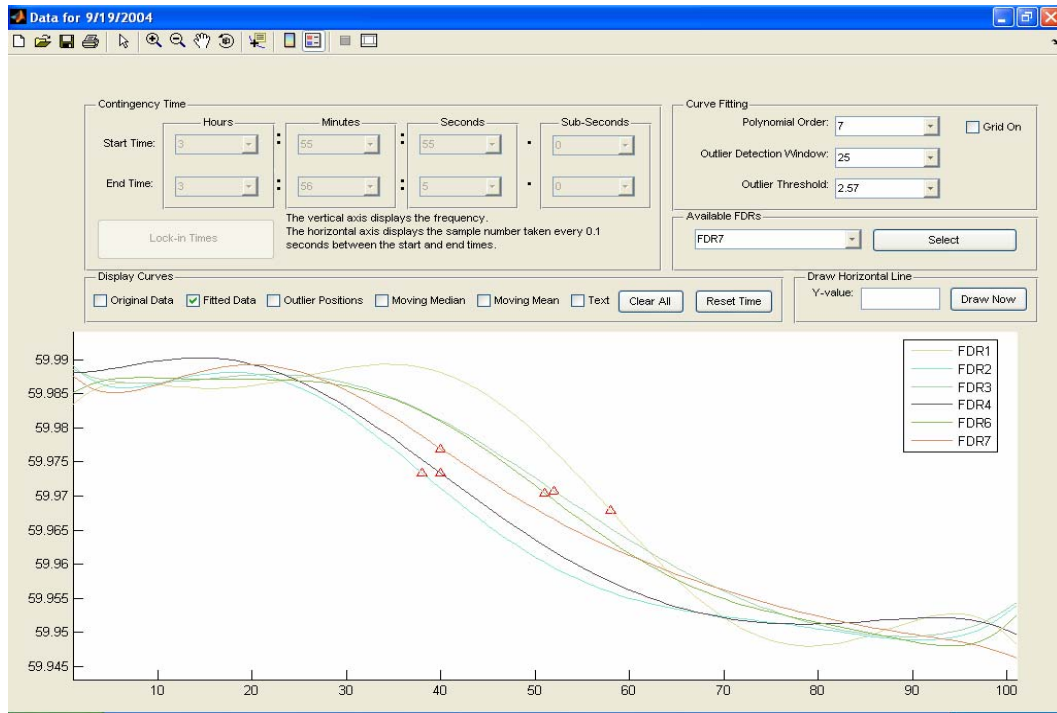


Figure 2-14: FDR data conditioning GUI developed for FNET (Base Case 2).

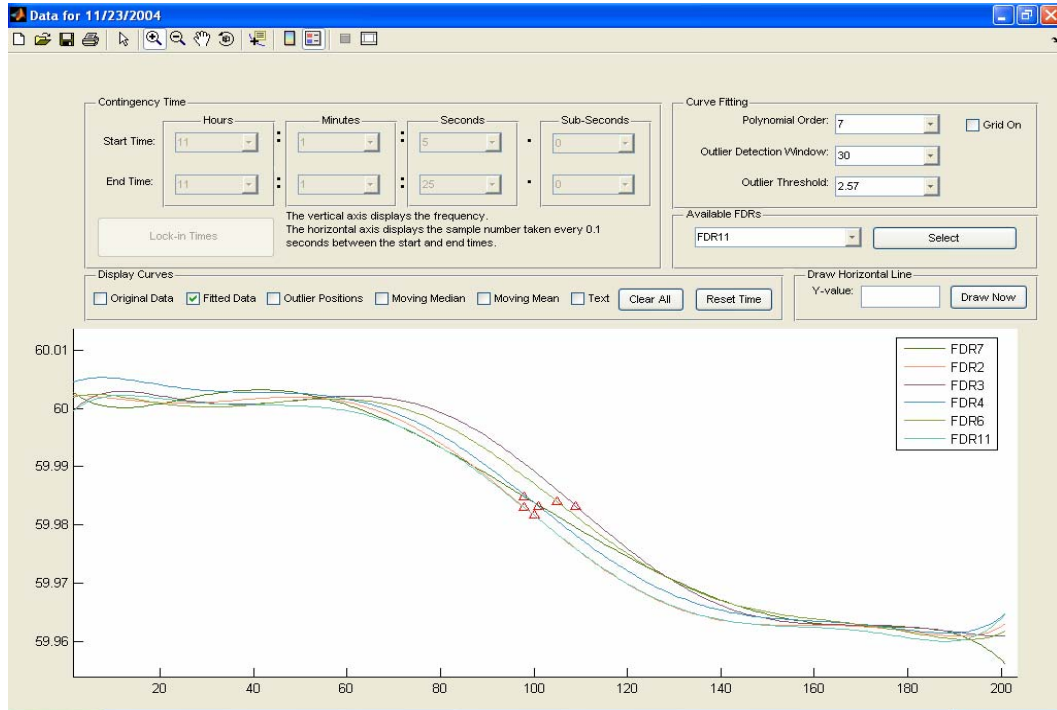


Figure 2-15: FDR data conditioning GUI developed for FNET (Base Case 3).

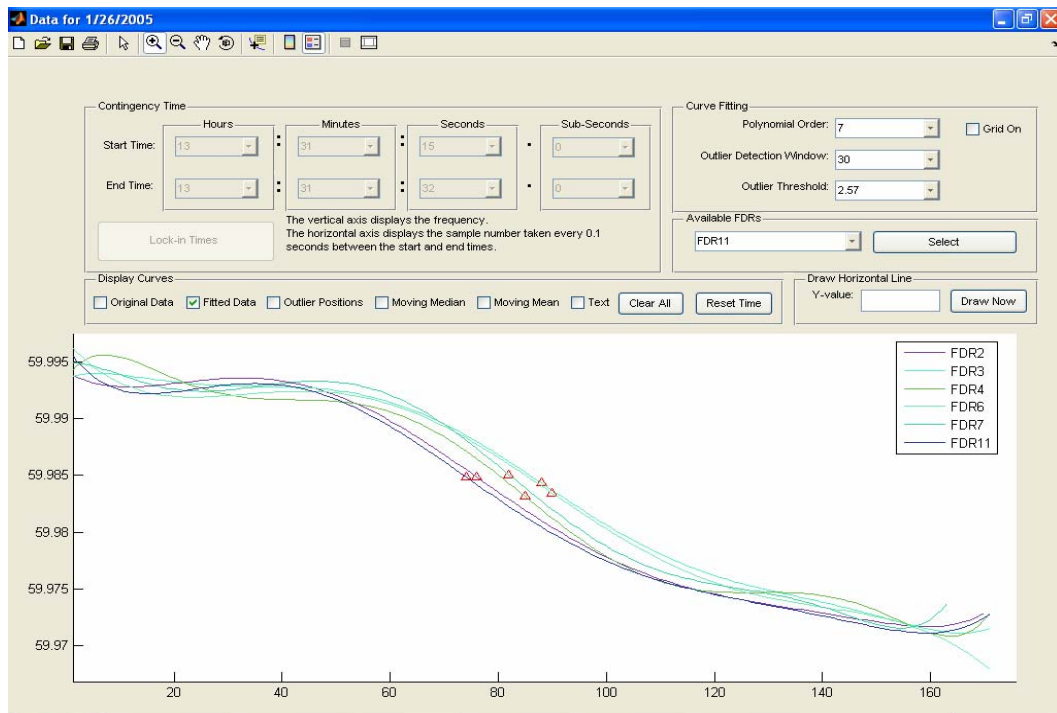


Figure 2-16: FDR data conditioning GUI developed for FNET (Base Case 4).

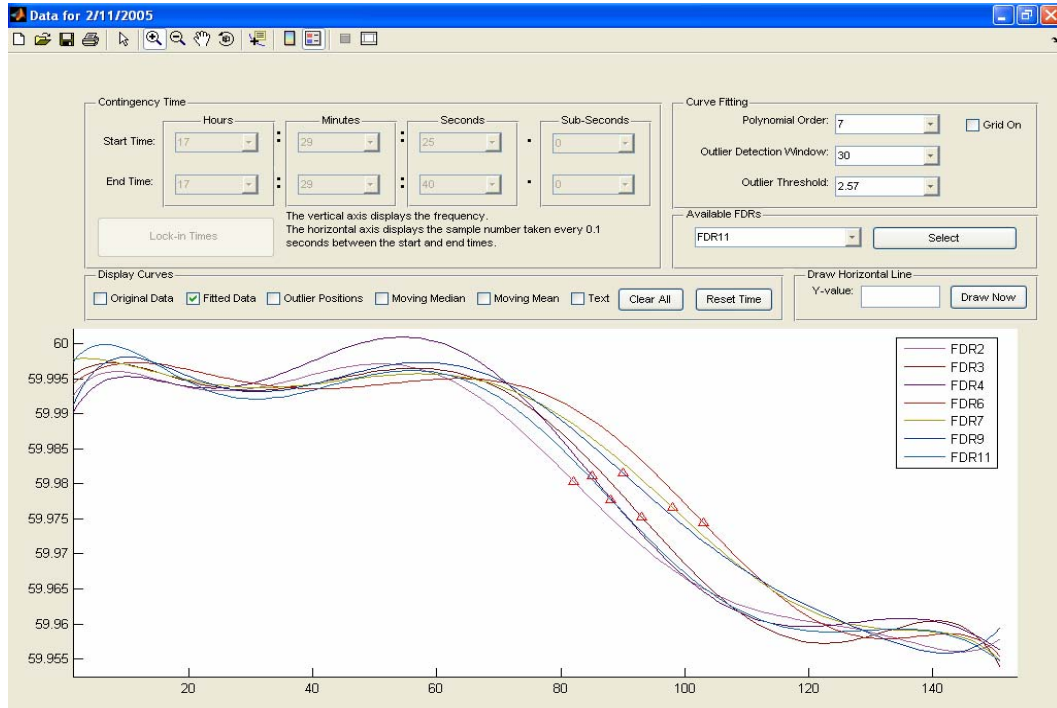


Figure 2-17: FDR data conditioning GUI developed for FNET (Base Case 5).

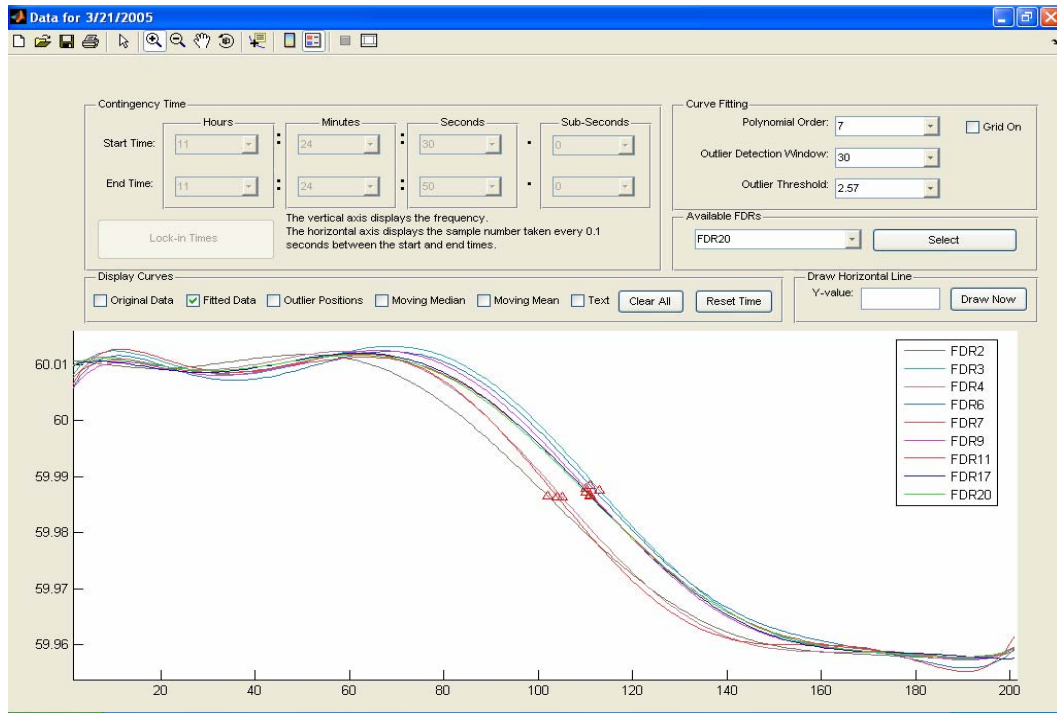


Figure 2-18: FDR data conditioning GUI developed for FNET (Base Case 6).

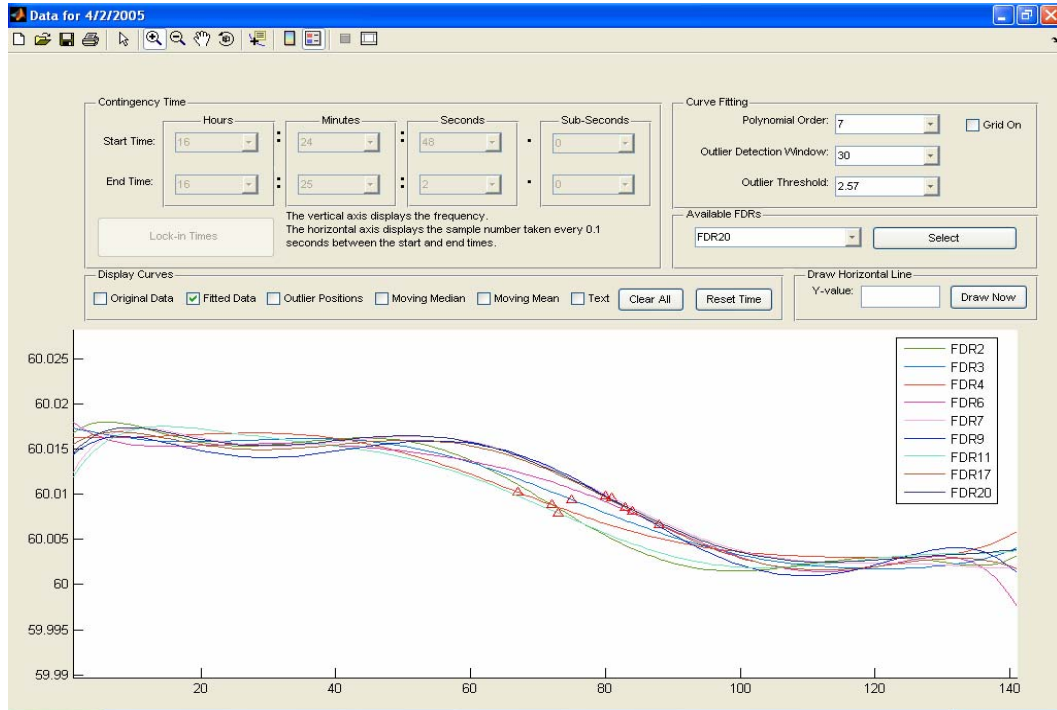


Figure 2-19: FDR data conditioning GUI developed for FNET (Base Case 7).

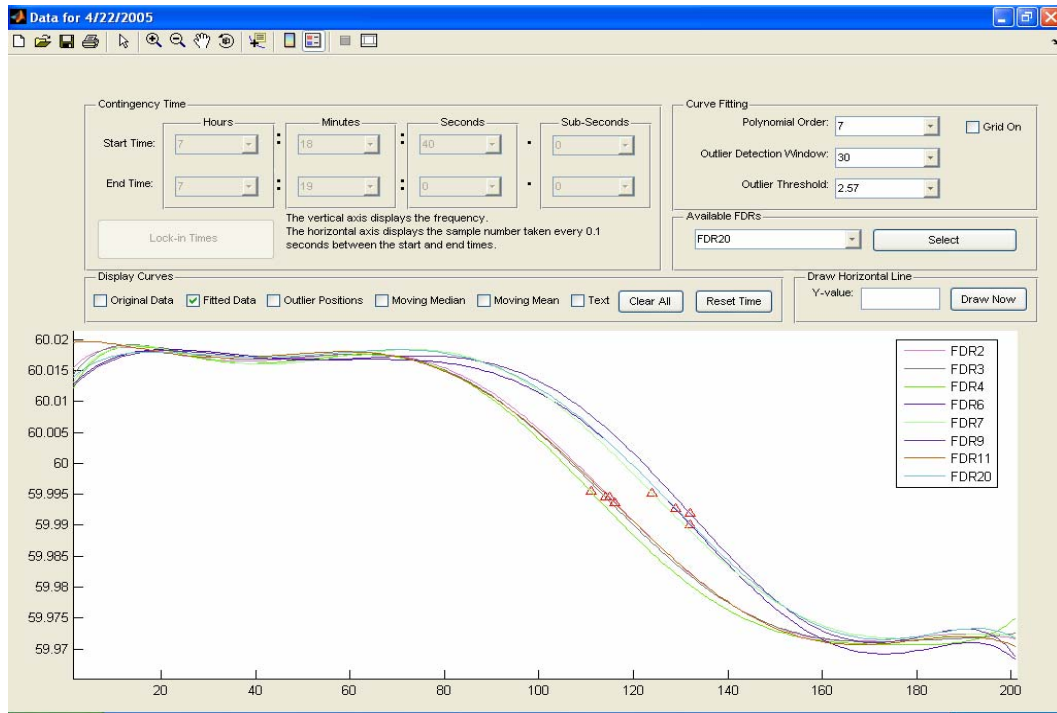


Figure 2-20: FDR data conditioning GUI developed for FNET (Base Case 8).

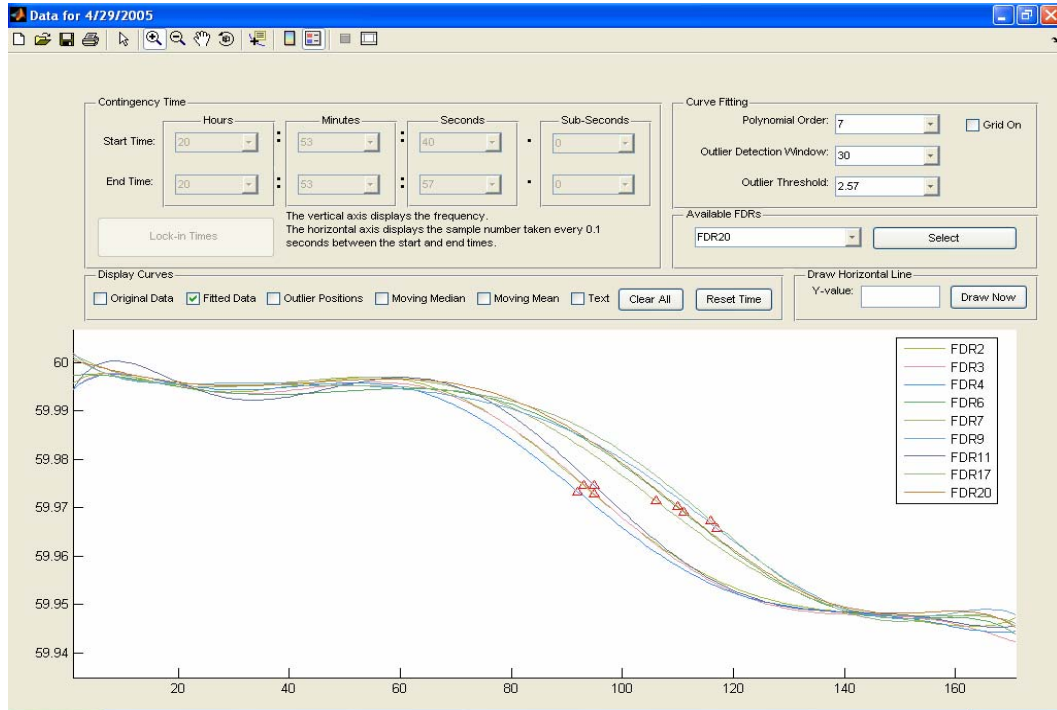


Figure 2-21: FDR data conditioning GUI developed for FNET (Base Case 9).

Table 2-1: Base case perturbation wave form arrival data (Cases 1-3).

CASE:	1	2	3
DATE:	4-Aug-2004	19-Sep-2004	23-Nov-2004
Plant Name:	Davis Besse	Watts Bar	Browns Ferry
Nearest Town:	Oak Harbor	Spring City	Athens
State:	OH	TN	AL
NERC Region:	ECAR	SERC	SERC
	↓Latitude↓ ↓Longitude↓	↓Latitude↓ ↓Longitude↓	↓Latitude↓ ↓Longitude↓
Coordinates:	41.512° -83.147°	35.687° -84.864°	34.786° -86.960°
↓ Unit ↓	↓Detection Time ↓	↓Detection Time ↓	↓Detection Time ↓
FDR1	--	3.56.00.0	--
FDR2	9.23.09.8	3.55.58.8	11.01.14.8
FDR3	9.23.09.9	3.56.00.2	11.01.15.9
FDR4	9.23.08.3	3.55.59.0	11.01.15.1
FDR6	9.23.12.4	3.56.00.1	11.01.15.5
FDR7	9.23.11.4	3.55.59.0	11.01.14.8
FDR9	9.23.11.8	--	--
FDR11	--	--	11.01.15.0
FDR17	--	--	--
FDR20	--	--	--

Table 2-2: Base case perturbation wave form arrival data (Cases 4-6).

CASE:	4	5	6
DATE:	26-Jan-2005	11-Feb-2005	21-Mar-2005
Plant Name:	East Bend	Browns Ferry	Cumberland
Nearest Town:	Rabbit Hash	Athens	Cumberland City
State:	KY	AL	TN
NERC Region:	ECAR	SERC	SERC
Coordinates:	↓Latitude↓ ↓Longitude↓	↓Latitude↓ ↓Longitude↓	↓Latitude↓ ↓Longitude↓
	39.032° -84.741°	34.786° -86.956°	36.382° -87.644°
↓ Unit ↓	↓Detection Time ↓	↓Detection Time ↓	↓Detection Time ↓
FDR1	--	--	--
FDR2	13.31.22.6	17.29.33.2	11.24.40.2
FDR3	13.31.23.8	17.29.34.3	11.24.41.3
FDR4	13.31.23.5	17.29.33.5	11.24.40.5
FDR6	13.31.24.0	17.29.35.3	11.24.41.1
FDR7	13.31.23.2	17.29.34.8	11.24.40.4
FDR9	--	17.29.34.0	11.24.41.0
FDR11	13.31.22.4	17.29.33.8	11.24.41.1
FDR17	--	--	11.24.41.1
FDR20	--	--	11.24.41.0

Table 2-3: Base case perturbation wave form arrival data (Cases 7-9).

CASE:	7	8	9
DATE:	2-Apr-2005	22-Apr-2005	29-Apr-2005
Plant Name:	Eastlake	Zimmer	Votgle-Wilson
Nearest Town:	Eastlake	Moscow	Waynesboro
State:	OH	OH	GA
NERC Region:	ECAR	ECAR	SERC
Coordinates:	↓Latitude↓	↓Latitude↓	↓Latitude↓
	↓Longitude↓	↓Longitude↓	↓Longitude↓
	41.660°	38.860°	33.090°
	-81.431°	-84.228°	-82.014°
↓ Unit ↓	↓Detection Time ↓	↓Detection Time ↓	↓Detection Time ↓
FDR1	--	--	--
FDR2	16.24.55.2	7.18.51.5	20.53.49.3
FDR3	16.24.55.5	7.18.51.4	20.53.49.5
FDR4	16.24.54.7	7.18.51.1	20.53.49.2
FDR6	16.24.56.8	7.18.53.2	20.53.51.1
FDR7	16.24.56.1	7.18.52.4	20.53.50.6
FDR9	16.24.56.4	7.18.53.2	20.53.51.7
FDR11	16.24.55.3	7.18.51.6	20.53.49.5
FDR17	16.24.56.3	--	20.53.51.6
FDR20	16.24.56.0	7.18.52.9	20.53.51.0

Chapter 3

Linear Triangulation Methods

3.1 *The Limited Contributions from Seismology*

Let us begin work on the triangulation problem by restating the thesis of this report: There exists a possibility of using the delay between the inception of an observable event within the electric utility grid of the EI and the detection thereof solely by frequency examination to triangulate with reasonable accuracy the location of the inciting event. As a first effort, we turn to seismology for inspiration in possible solution methods for our problem. This step is taken because of two common areas between the problem of power system event location and earthquake triangulation: (1) The seismograph is used as a tool to measure disturbances (perturbations on and under the Earth's crust) just as our FDRs measure disturbances, and (2) these seismic perturbations travel with finite speed through diverse strata in the Earth's composition [11, 45-50]. In the study of seismology, however, we encounter many areas in which the paradigm between our perception of the utility grid and the spherical globe breakdown. First, the triangulation of the hypocenter of earthquakes is a much more mature science than is the triangulation of the hypocenter of utility grid disturbances. Today, there are nearly 3,000 seismic observatories around the globe producing hundreds of thousands of seismographs annually [49, 51, 52]. Each of these stations has at least one or more seismic measurement devices, and in the measurement of seismological disturbances, several families and types of waveforms are captured [11, 45-50]. The following figure shows the data measured at only one of the several University of Utah Seismograph Stations.

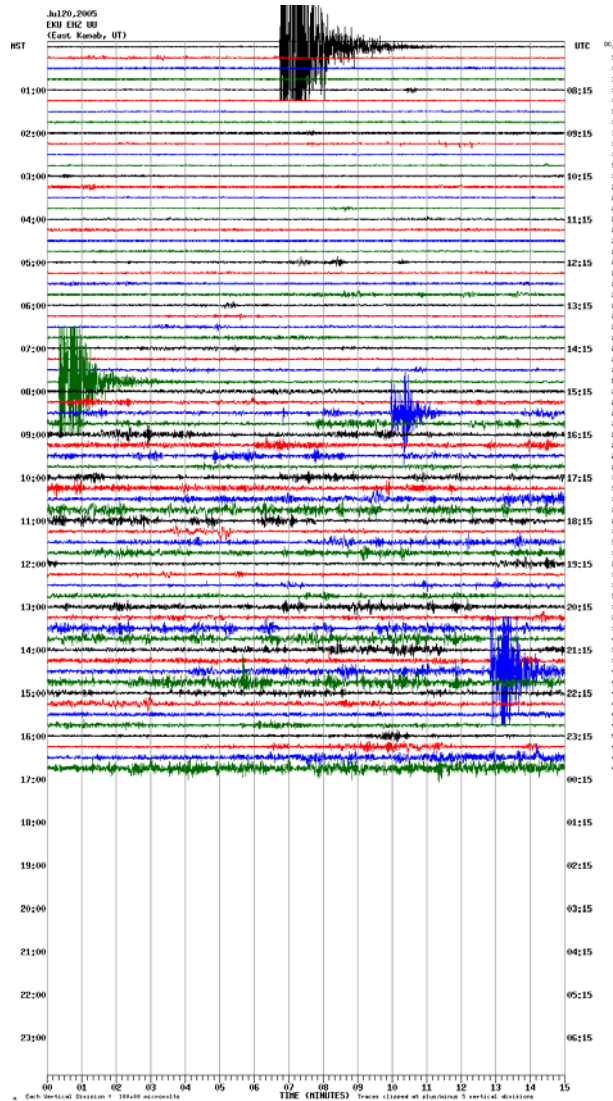


Figure 3-1: Example of seismograph readings taken at a single observatory station [53].

Conversely, we are constrained to measuring only one such signal and that is the aforementioned electromechanical wave that manifests itself in the form of a frequency waveform. In the world of seismology, much can be discovered about the hypocenter of an earthquake by analyzing certain established relationships between the various observed waves in the seismograph. There is

no such analog in our case if we constrain ourselves (as we have) to frequency (and the related angle) waveforms alone.

Thus far we have noticed and agreed upon the fact that the EI grid is a grid of heterogeneous time-varying elasticity. This is also the case with the construction of the Earth [11, 45-50]. Again, however, the study of seismology is thousands of years old. The Chinese have been recording earthquakes with a variety of methods since 1800 BC and measurements more constructive to contemporary science have been recorded since around AD 1750 [49]. A long legacy of measurements, trial, and error has lead to a fairly complex model of the Earth's structure. Thus, velocity models have for many years existed in the form of the notorious Jeffreys-Bullen travel time table (see figure 3-2) which is associated with the path of seismic waves (see figure 3-3) [11, 49, 54].

Seismology is also very much a three-dimensional (3-D) problem. Many ideas in seismology are related to 3-D torsions, stresses, deformations, et cetera. Not only are seismologists concerned with the latitude and longitude of an earthquake, but they also look for the depth of the disturbance below the surface of the Earth [11, 45-50]. We, however, have one less dimension with which to be concerned.

At this juncture, there seem to be few established relationships between advanced seismology and electrical engineering. However, some of the simpler seismological techniques might be of use. When a Cartesian coordinate system is available, the Pythagorean relation is used in seismology for small seismographic networks [49]:

$$(x_i - x_h)^2 + (y_i - y_h)^2 + (z_i - z_h)^2 = V^2(t_i - t_h)^2$$



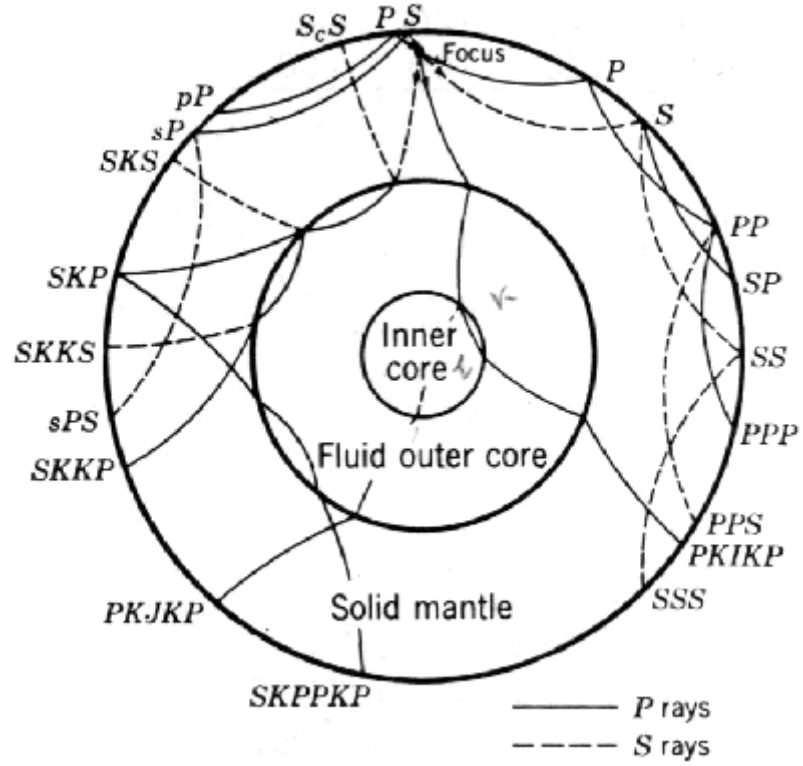


Figure 3-3: Jeffreys-Bullen plot of wave paths [54].

In the Pythagorean relation above, x_i is the latitude coordinate of the i^{th} station, y_i is the longitude coordinate of the i^{th} station, z_i is the altitude coordinate of the i^{th} station, x_h is the latitude coordinate of the hypocenter, y_h is the longitude coordinate of the hypocenter, z_h is the altitude coordinate of the hypocenter, V is the velocity of the medium, t_i is the disturbance measurement time of the i^{th} station, and t_h is the time of origin at the hypocenter. From the data given in the previous chapters, we have enough data to formulate a method of solving this relation. Accurate results are not guaranteed as, at this point, no relationship between the field of seismology and utility grid event location has been proven. To find a solution to the Pythagorean relation, let us first attempt to find set of

equations linear in terms of hypocentral variables x_h , y_h , and t_h (eliminating the altitude component z_h).

3.2 The Least-Squares Approach

Before beginning the solution process, MATLAB's Mapping Toolbox was used to convert the 3-D spherical Greenwich latitude and longitude coordinates to two-dimensional (2-D) Cartesian coordinates. After the solution methods herein have been implemented, the reverse coordinate transformation process was used to change the 2-D coordinates back to 3-D coordinates [55].

Since we have several, say n , FDR unit locations, we can form a system of Pythagorean relationships in the following manner:

$$\begin{aligned}(x_1 - x_h)^2 + (y_1 - y_h)^2 &= V^2(t_1 - t_h)^2 \\(x_2 - x_h)^2 + (y_2 - y_h)^2 &= V^2(t_2 - t_h)^2 \\ \vdots & \quad \quad \quad \vdots \\(x_n - x_h)^2 + (y_n - y_h)^2 &= V^2(t_n - t_h)^2\end{aligned}$$

It is worth noting at this point that we seek a linear system in terms of the hypocentral coordinates such that the least-squares (LS) method of solving an over-constrained system of equations can be used. In general, our system of equations will be over-constrained. By subtracting successive pairs of station equations, a linear equation in terms of the hypocentral coordinates is produced:

$$(x_{i+1} - x_i)x_h + (y_{i+1} - y_i)y_h - V^2(t_{i+1} - t_i)t_h = \frac{1}{2}[V^2(t_i^2 - t_{i+1}^2) + x_{i+1}^2 + y_{i+1}^2 - x_i^2 - y_i^2]$$

The above equation is linear in terms of our three hypocentral coordinates and the LS method can be applied. To accomplish this, we first form the following system of equations while collapsing the right-hand side into a constant C_i :

$$\begin{aligned}
(x_2 - x_1)x_h + (y_2 - y_1)y_h - V^2(t_2 - t_1)t_h &= C_1 \\
(x_3 - x_2)x_h + (y_3 - y_2)y_h - V^2(t_3 - t_2)t_h &= C_2 \\
\vdots & \\
(x_1 - x_n)x_h + (y_1 - y_n)y_h - V^2(t_1 - t_n)t_h &= C_n
\end{aligned}$$

represented in matrix form as:

$$\underline{C} = \underline{H}\underline{x}$$

with matrix variables defined as:

$$\underline{C} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_{n-1} \\ C_n \end{bmatrix} \quad \underline{H} = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 & V^2(t_2 - t_1) \\ x_3 - x_2 & y_3 - y_2 & V^2(t_3 - t_2) \\ \vdots & \vdots & \vdots \\ x_n - x_{n-1} & y_n - y_{n-1} & V^2(t_n - t_{n-1}) \\ x_1 - x_n & y_1 - y_n & V^2(t_1 - t_n) \end{bmatrix} \quad \underline{x} = \begin{bmatrix} x_h \\ y_h \\ t_h \end{bmatrix}$$

The hypocentral coordinates can then be solved for via the LS method using the pseudo-inverse as discussed previously in Section 2.3.2 [11, 44-46]:

$$\underline{H}^\dagger = (\underline{H}^t \underline{H})^{-1} \underline{H}^t$$

yielding the final solution:

$$\underline{x} = \underline{H}^\dagger \underline{C}$$

The main problem with using this method of finding the hypocentral coordinates is that we must assume that the frequency disturbance waves are traveling in a continuum at a constant velocity in each direction. Furthermore, we must also assume a scalar velocity before solving. At this point we have no velocity model for the travel of these frequency perturbation waves so we must make several guesses, solve using each guess, and evaluate the results. The following figures show the results of the LS method, plotted with red asterisks, using velocities that vary from 100 mi/s to 1000 mi/s. We chose this range because in the past we have measured disturbances traveling between 350 mi/s and 660 mi/s and we wish to accommodate a wide range of possible disturbance

travel speeds [2, 23]. In the following figures, the closest solution location is enlarged and pointed out with an arrow. Again, here the base case being triangulated is plotted with a square and the FDR locations are shown using blue dots.

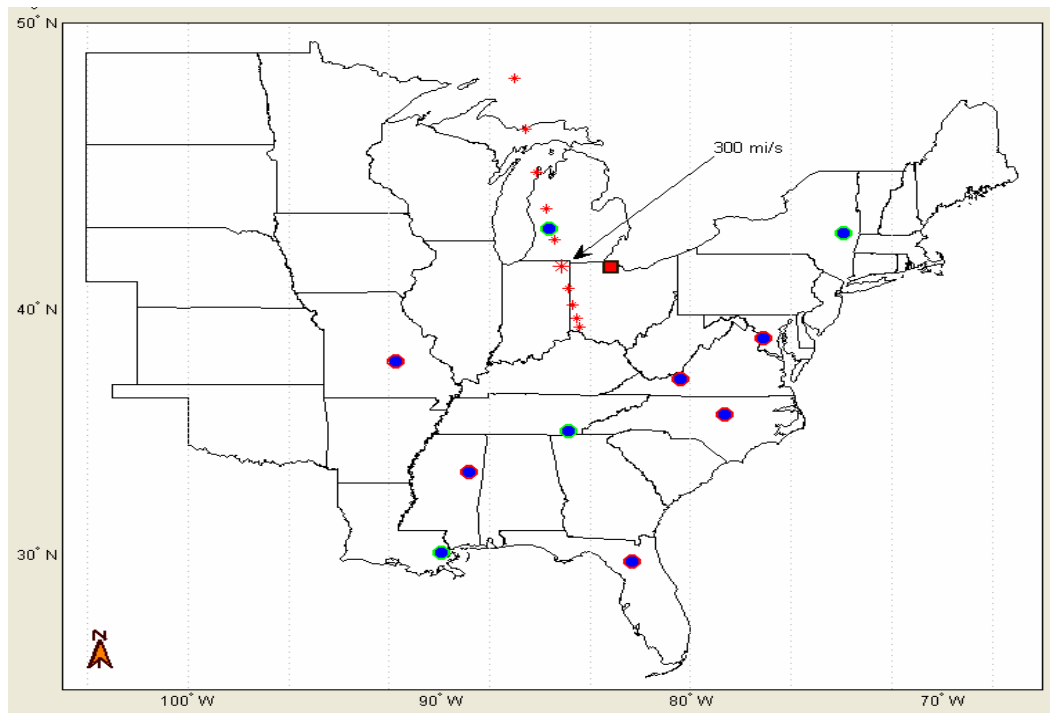


Figure 3-4: Linear triangulation of base case 1.

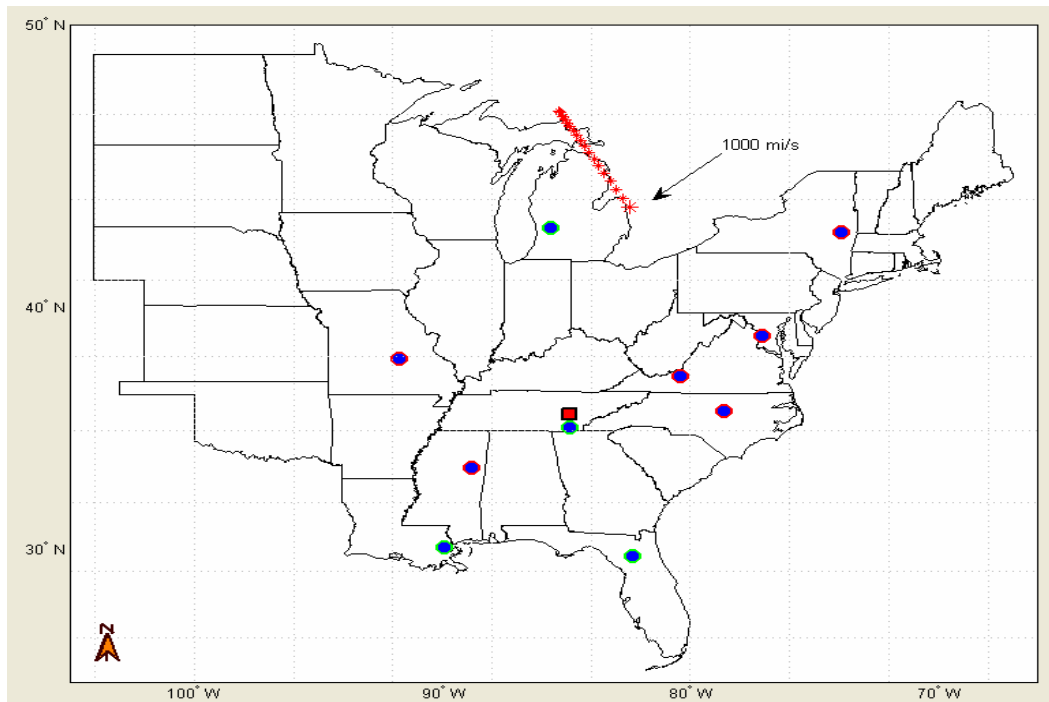


Figure 3-5: Linear triangulation of base case 2.

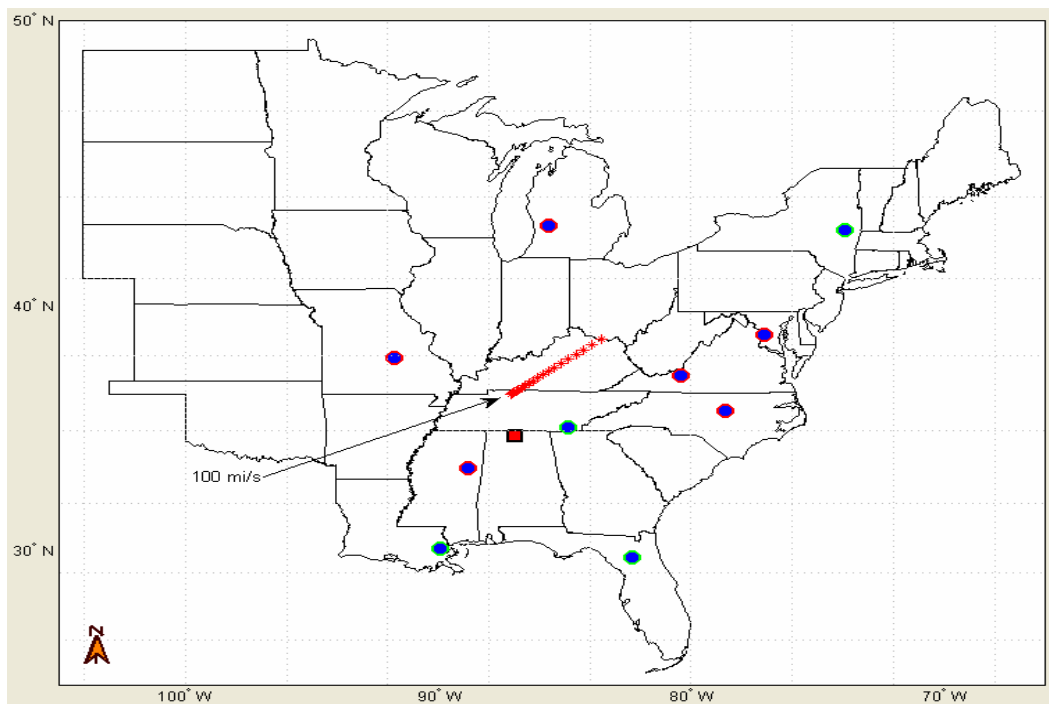


Figure 3-6: Linear triangulation of base case 3.

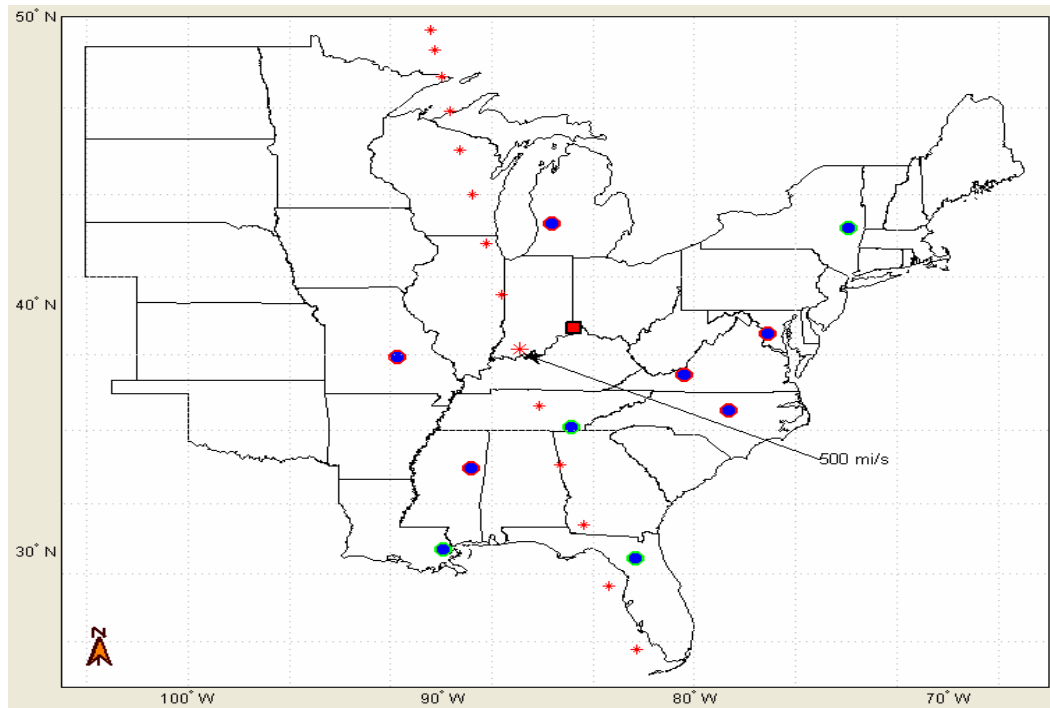


Figure 3-7: Linear triangulation of base case 4.

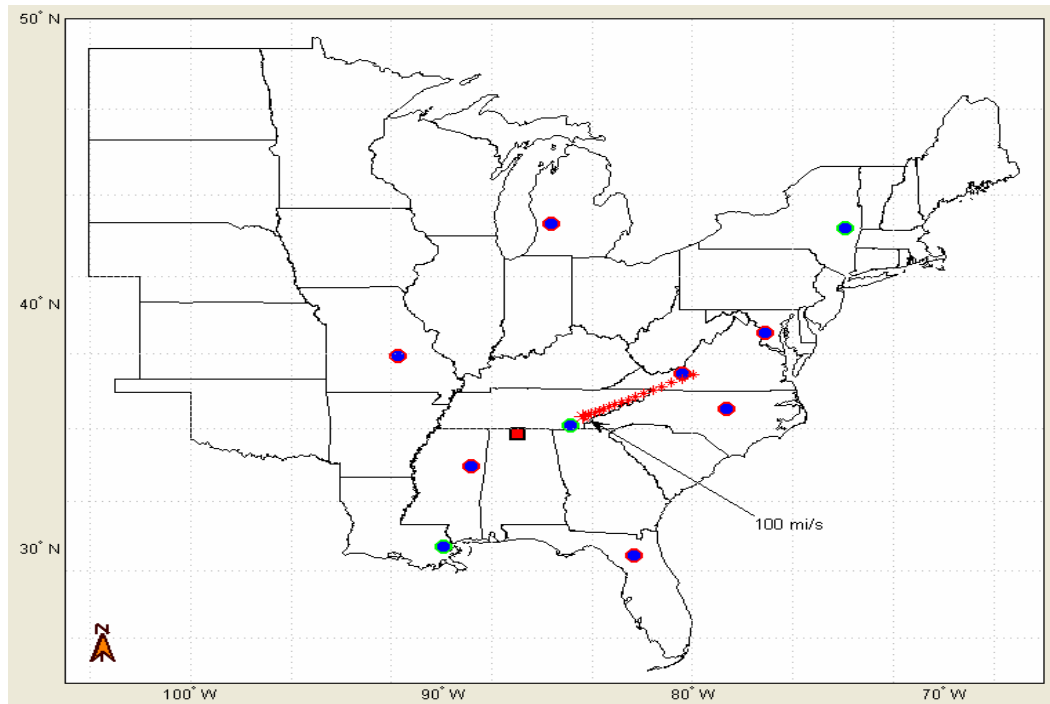


Figure 3-8: Linear triangulation of base case 5.

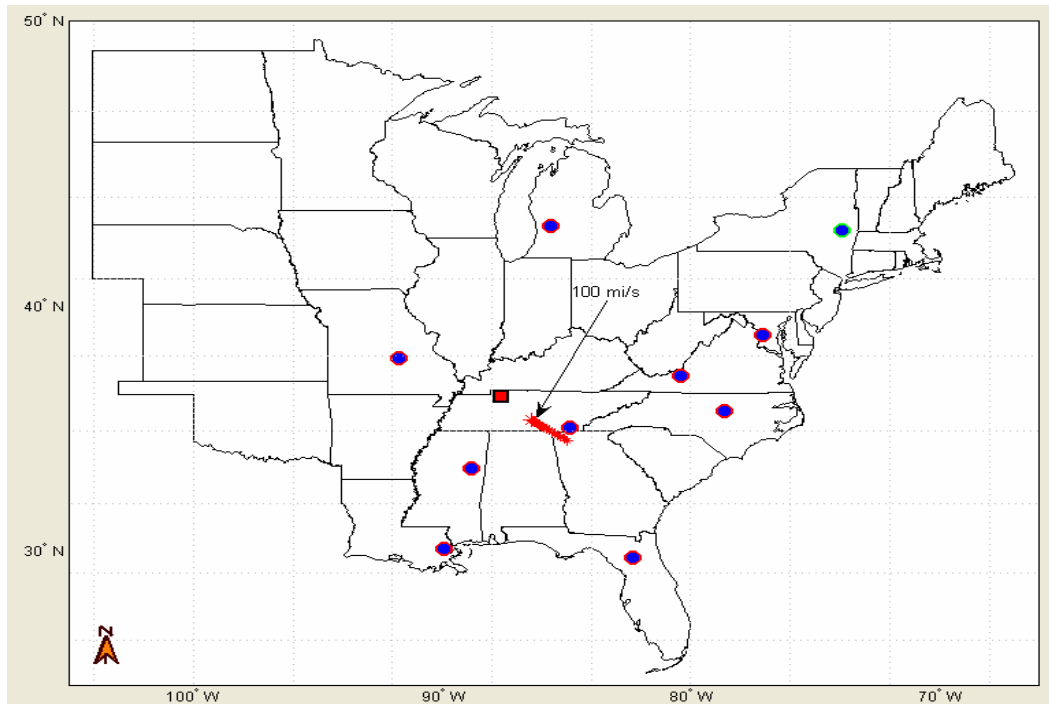


Figure 3-9: Linear triangulation of base case 6.

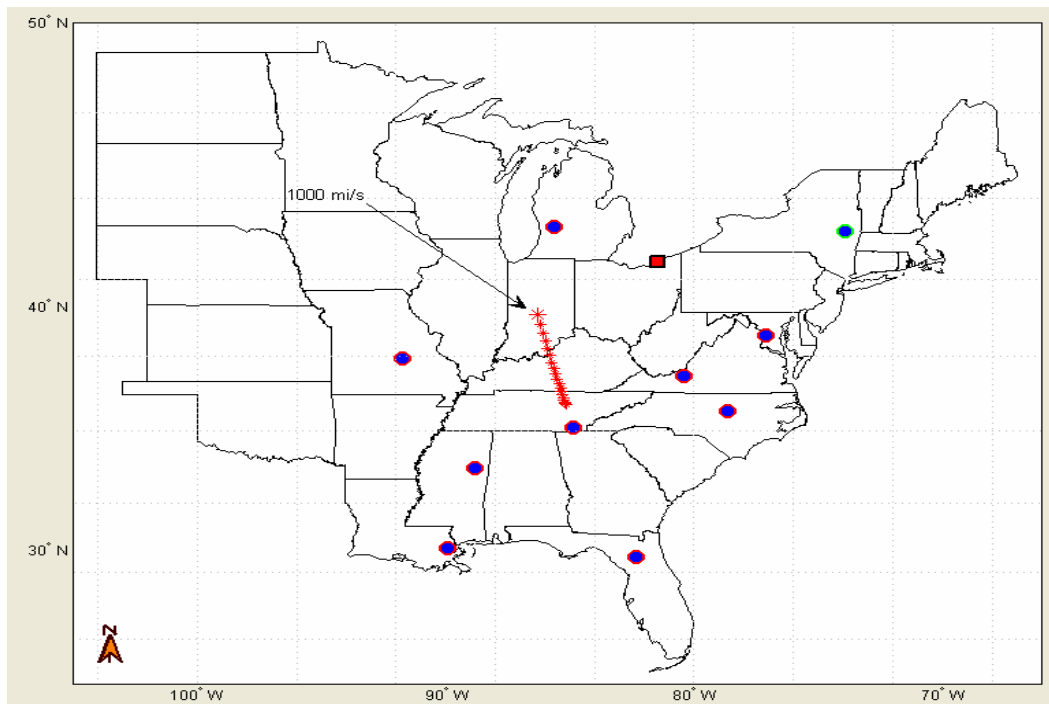


Figure 3-10: Linear triangulation of base case 7.

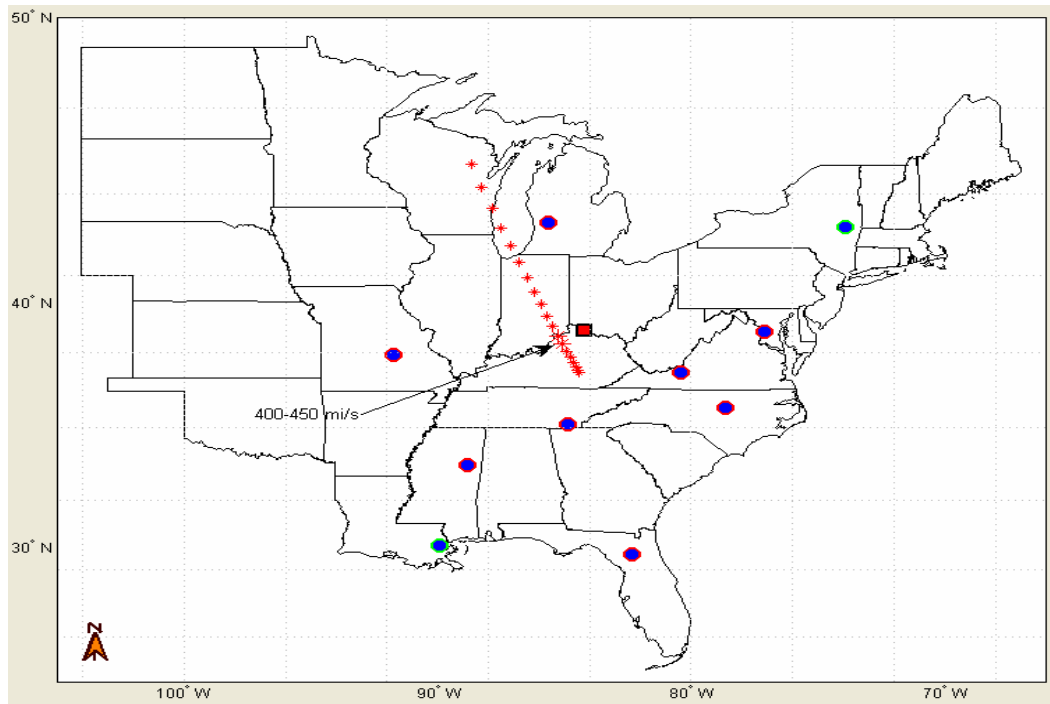


Figure 3-11: Linear triangulation of base case 8.

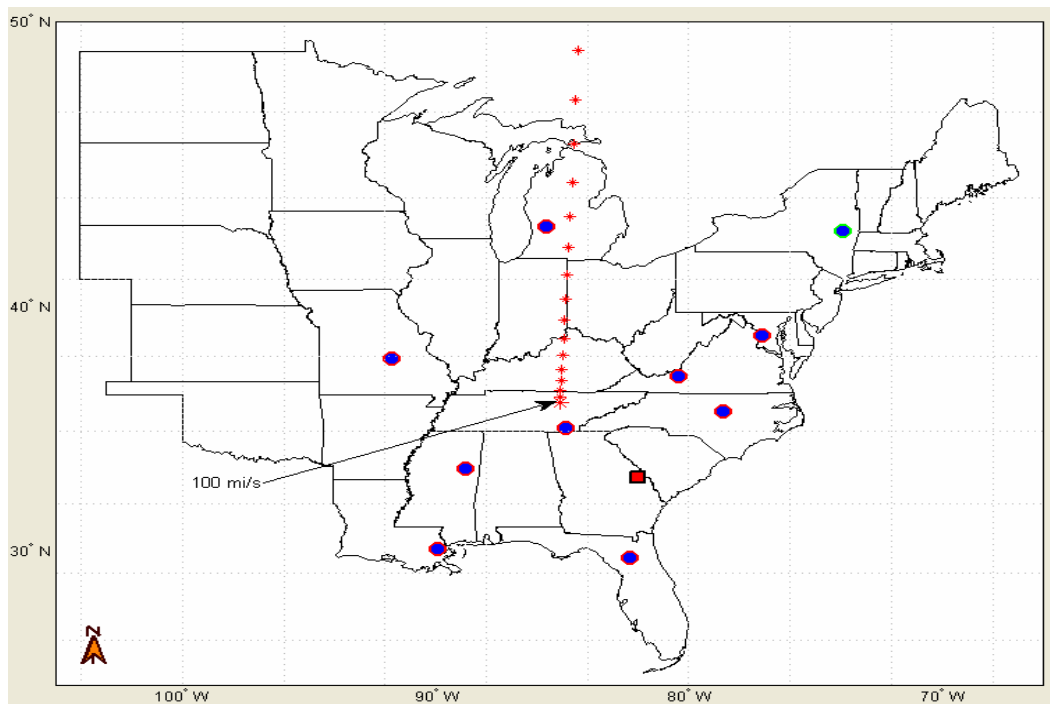


Figure 3-12: Linear triangulation of base case 9.

The following table shows the distances between the actual location of each base case and the loci of solutions from the linear LS triangulation method. Shortest distances are highlighted in red.

Table 3-1: Numerical results for linear triangulation method.

<i>Continuum Speed (mi/s):</i>	<i>Solution Distances from Base Case (mi):</i>								
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
100	171	789	118	779	154	95	435	113	276
150	151	786	120	735	157	96	433	107	287
200	126	781	123	674	162	98	430	99	302
250	105	776	126	596	168	100	425	89	323
300	103	769	130	505	175	103	420	77	349
350	136	761	136	400	184	106	414	66	381
400	199	752	142	287	195	110	408	57	420
450	282	742	150	176	206	114	400	57	466
500	382	731	159	129	219	119	392	67	518
550	498	718	169	223	233	124	383	88	578
600	631	705	181	376	249	130	373	115	645
650	782	691	195	549	266	136	363	148	720
700	954	675	209	735	285	143	352	185	804
750	1151	659	225	933	304	150	341	226	896
800	1381	642	243	1141	325	158	330	270	998
850	1656	625	262	1360	348	166	320	318	1110
900	2001	607	282	1589	372	175	309	370	1233
950	2499	588	304	1828	397	184	300	425	1369
1000	3350	570	327	2078	423	194	292	484	1520

As can be seen from the above figures and table, there exist a few problems with the LS method of triangulating the disturbance origin. At least four possible reasons exist for these discrepancies:

- Too few FDR output points for each second
- Lack of sufficient measurement units
- The heterogeneous nature of the EI grid
- The time- and condition-varying nature of the EI grid
- Lack of event progression repeatability due to system topology changes

Each FDR samples the frequency at 10 samples per second. If frequency perturbations actually travel between 100 mi/s and 1000 mi/s in the EI, then errors could be between 10 mi and 100 mi. Furthermore, these error estimates can be multiplied by each additional degree of freedom that is added into the system. Another obvious problem is the lack of sufficient measurement units available. There are vast spaces in the above maps where there is very little indication of frequency. The solution of this over-constrained linear system of equations cannot account in any manner for the heterogeneous nature of the EI grid in the LS sense. And finally, if base case three and five are examined using the data listed in tables 2-1 and 2-2, one shall quickly notice a lack of consistency between the timing of the two cases although they were located in the same plant. In the following figure we see a plot of the assumed continuum velocity versus the distance calculated between the LS solution and the true location for each of the nine base cases in this study. From this figure and the above table, it is easy to see that there exists no common continuum speed that can be assumed in the honing of a LS solution.

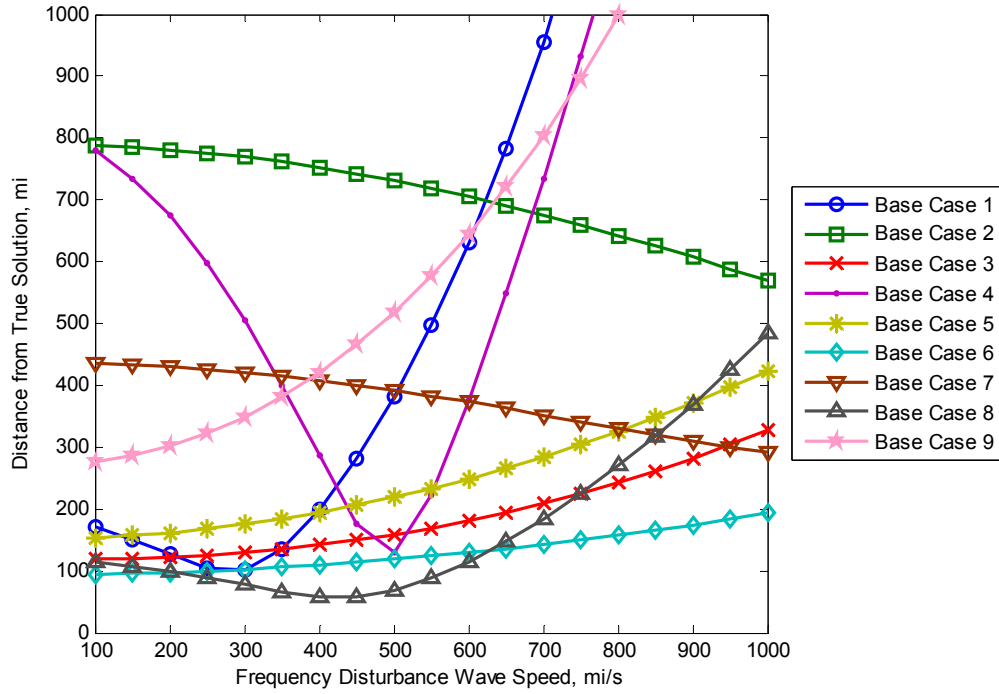


Figure 3-13: Plot of LS error versus assumed continuum speed.

3.3 Problems with the Over-Constrained System

Consider the figure below. If each dot represents an FDR location and the number of each dot represents the order in which a disturbance was detected, then one might consider using the following triangulation method: If a circle of increasing radius is drawn around each of n points and is enlarged until there exists a neighborhood of at least n intersections in an arbitrarily small area with the following constraint:

$$r_1 \leq r_2 \leq \dots \leq r_n$$

where r_i is associated with the i^{th} measurement unit and all measurement units are numbered in ascending order according to disturbance detection time, then we can assert that the disturbance hypocenter is located within the neighborhood that satisfies these constraints.

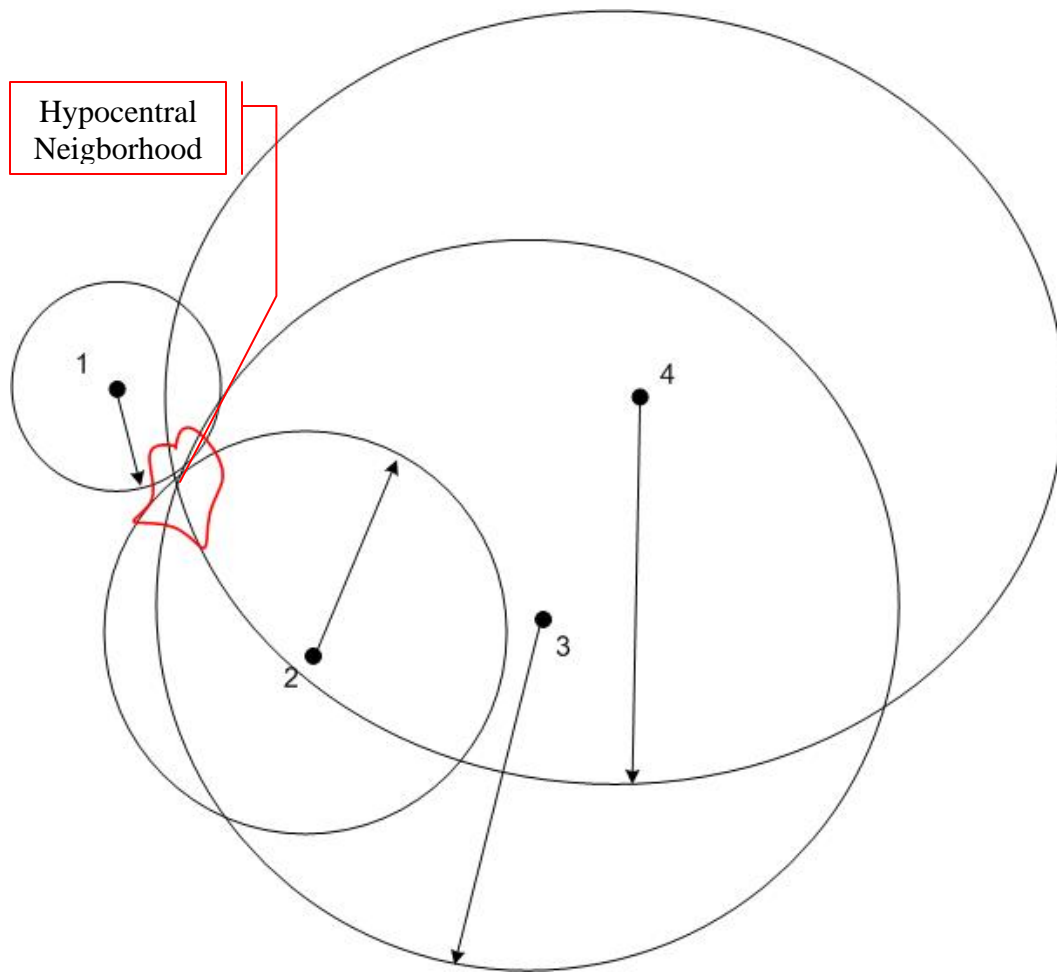


Figure 3-14: Cartoon of hypocenter triangulation by increasing radii.

In an effort to find the hypocenter using the above method, we use MATLAB to search for points within the solution space (the area in and around the EI). This search is accomplished by breaking the entire solution space into a grid of candidate points consisting of coordinates with a latitude and longitude. From each candidate point, radii are constructed to each of the base case locations. If the radii constraint from above is satisfied, then the point is considered a part of the hypocenter loci, else the point is disregarded. This method is iterated until every candidate point in the grid has been tested. See the following flow chart.

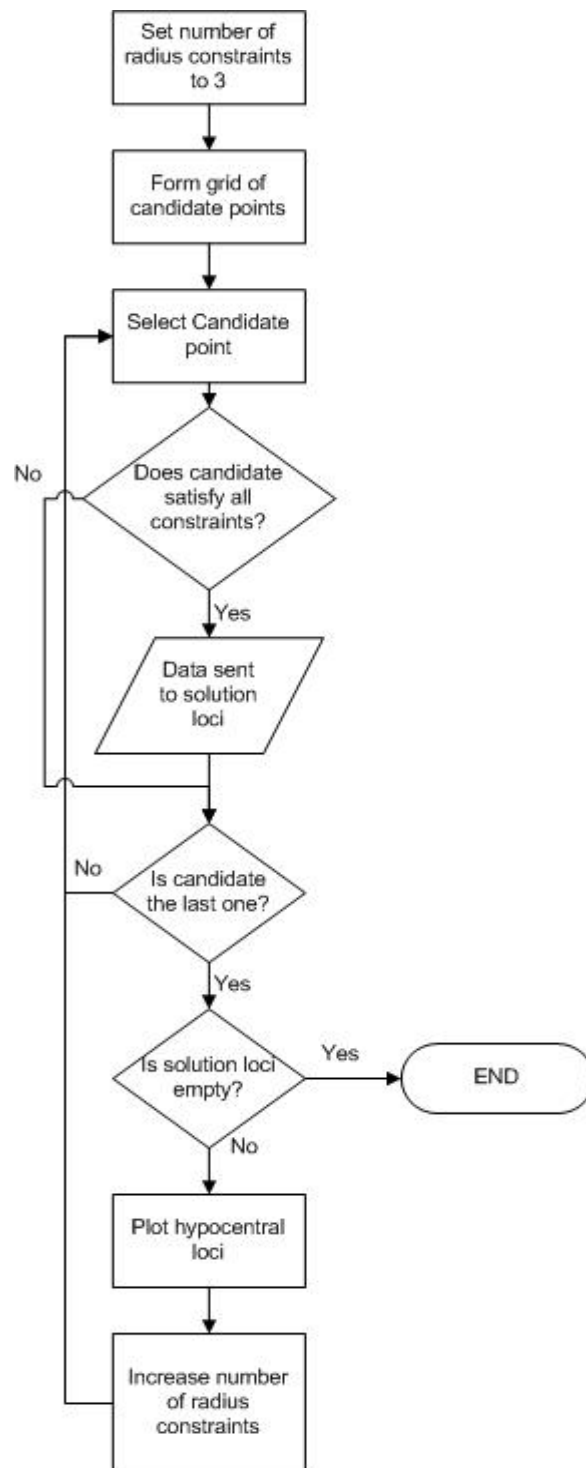


Figure 3-15: Flow chart for algorithm to find hypocentral loci.

The following figures show the hypocentral loci for each base case with a varying number of constraints added. After about four radial constraints are added, in most cases there exists no point within the solution space that satisfies the radii constraints. In the following figures, brighter shades of red represent the loci of points that meet a higher number of constraints on the solution space. Logically, loci that satisfy $n + 1$ constraints will also be in the solution set for n constraints. For each patch of red, the numbers of satisfying constraints are listed. As is evident from the following plots, there seems to be a problem in dealing with our over-constrained system in a linear manner.

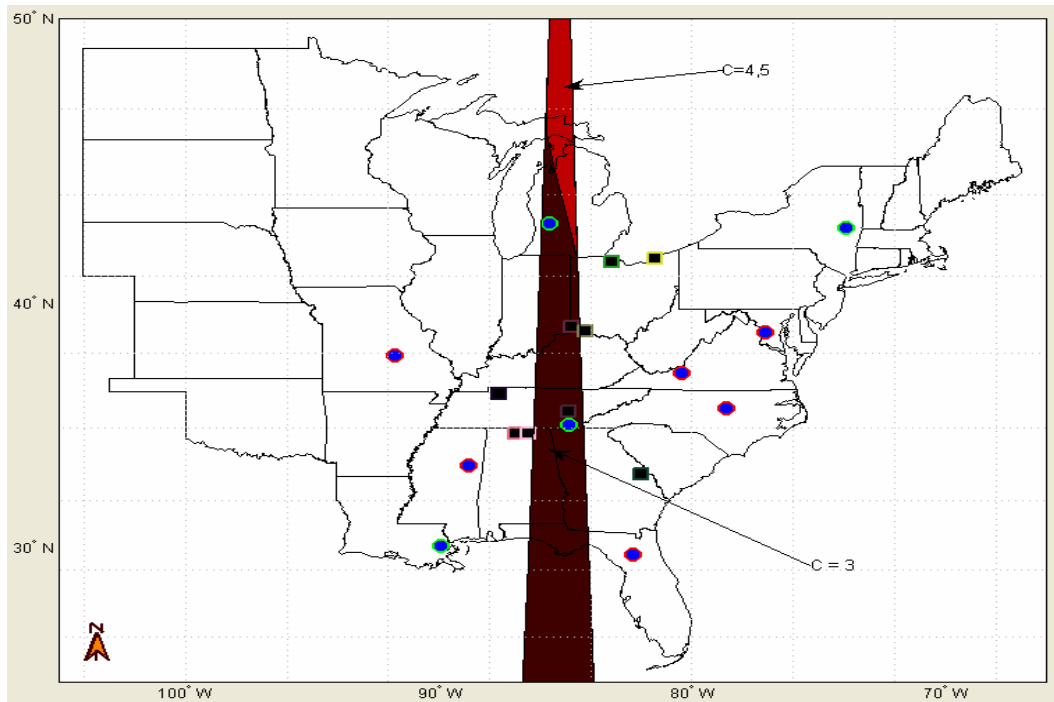


Figure 3-16: Hypocentral loci that meet constraints of base case 1.

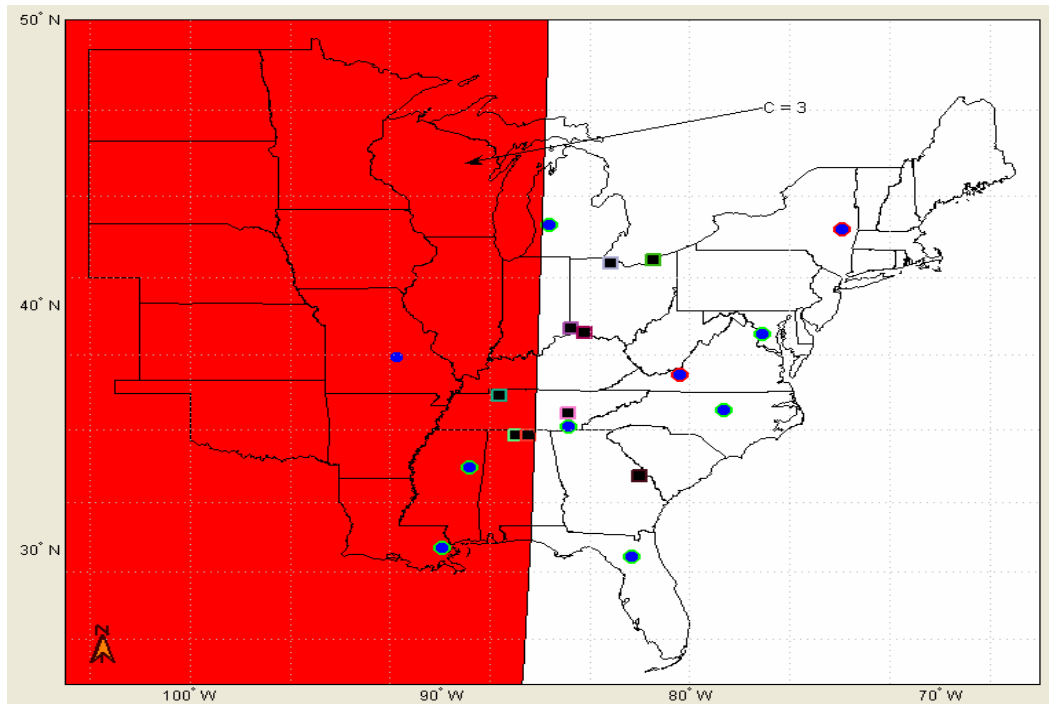


Figure 3-17: Hypocentral loci that meet constraints of base case 2.

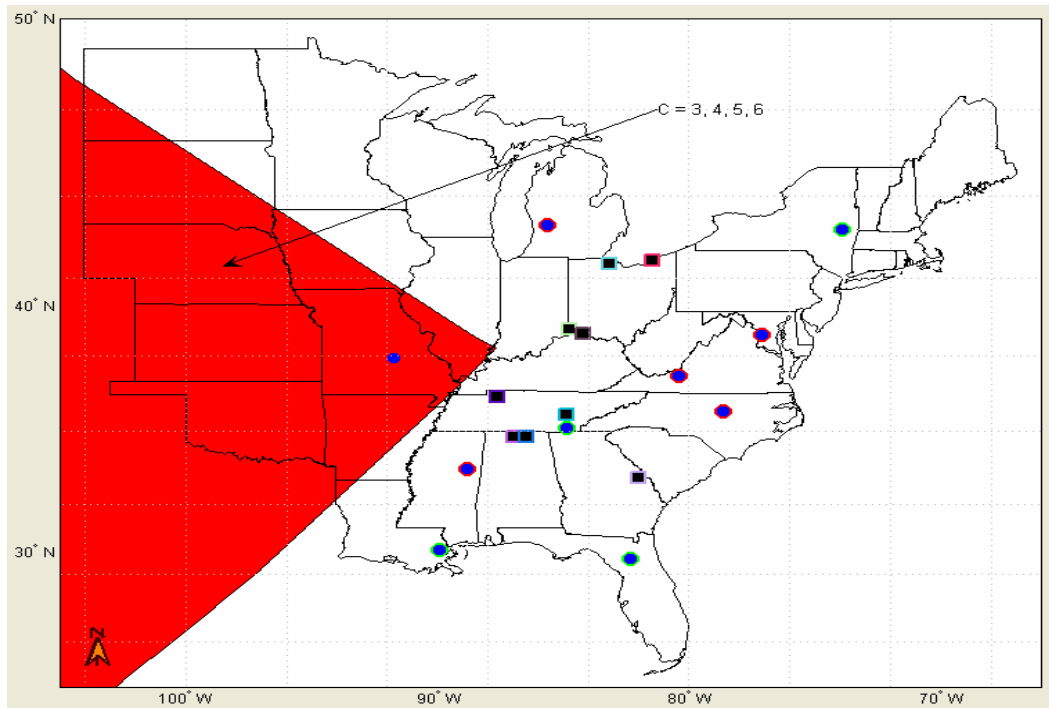


Figure 3-18: Hypocentral loci that meet constraints of base case 3.

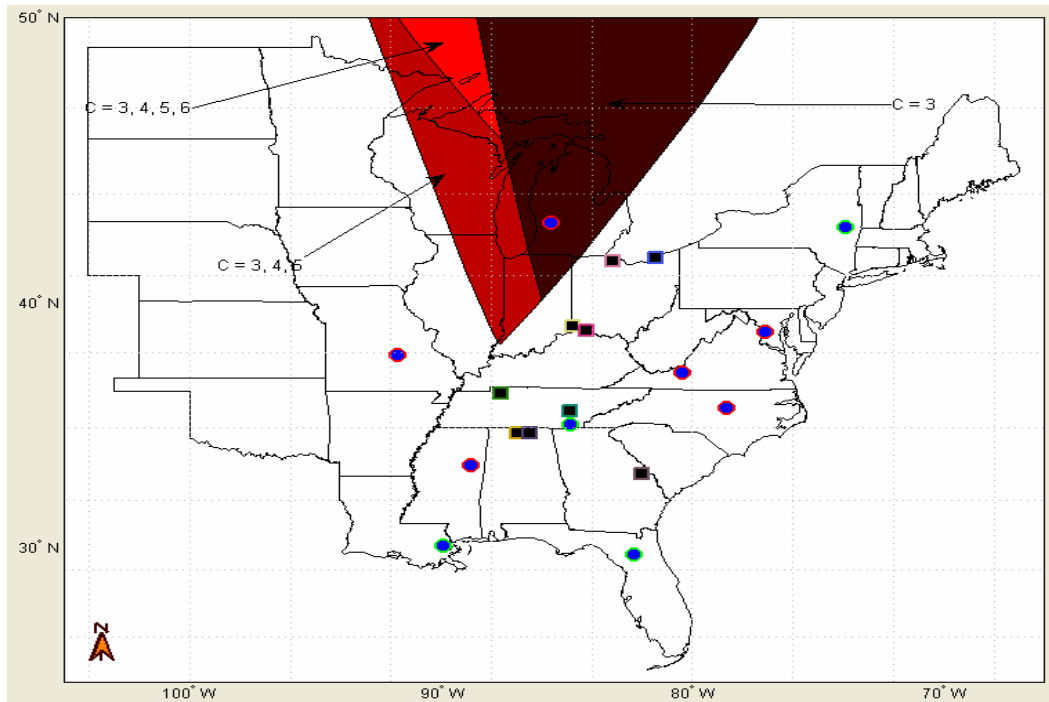


Figure 3-19: Hypocentral loci that meet constraints of base case 4.

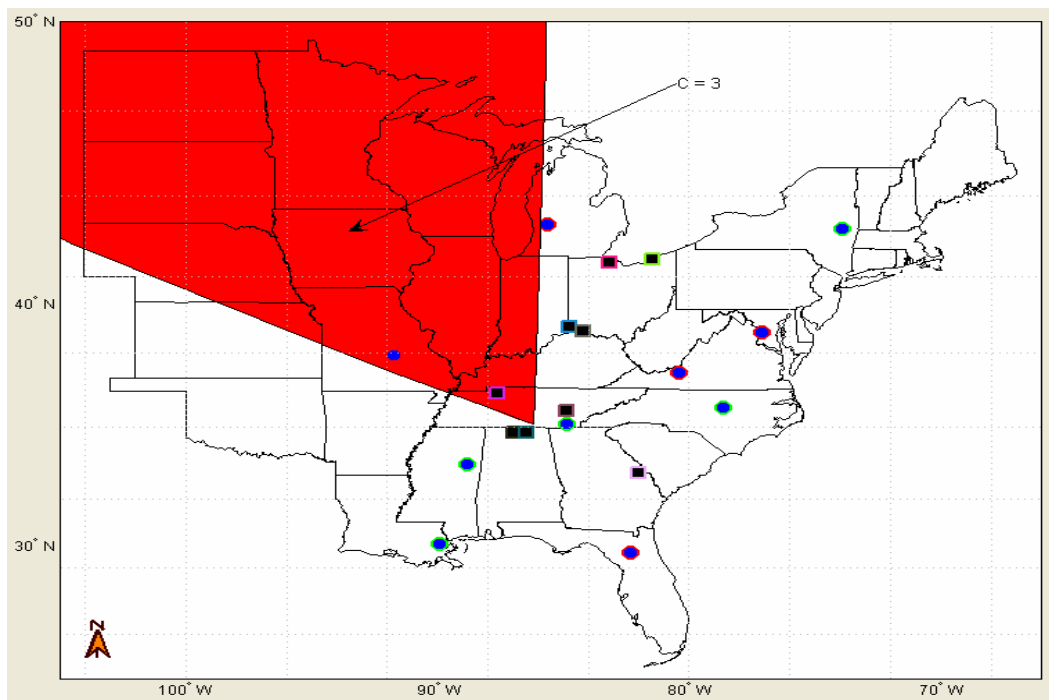


Figure 3-20: Hypocentral loci that meet constraints of base case 5.

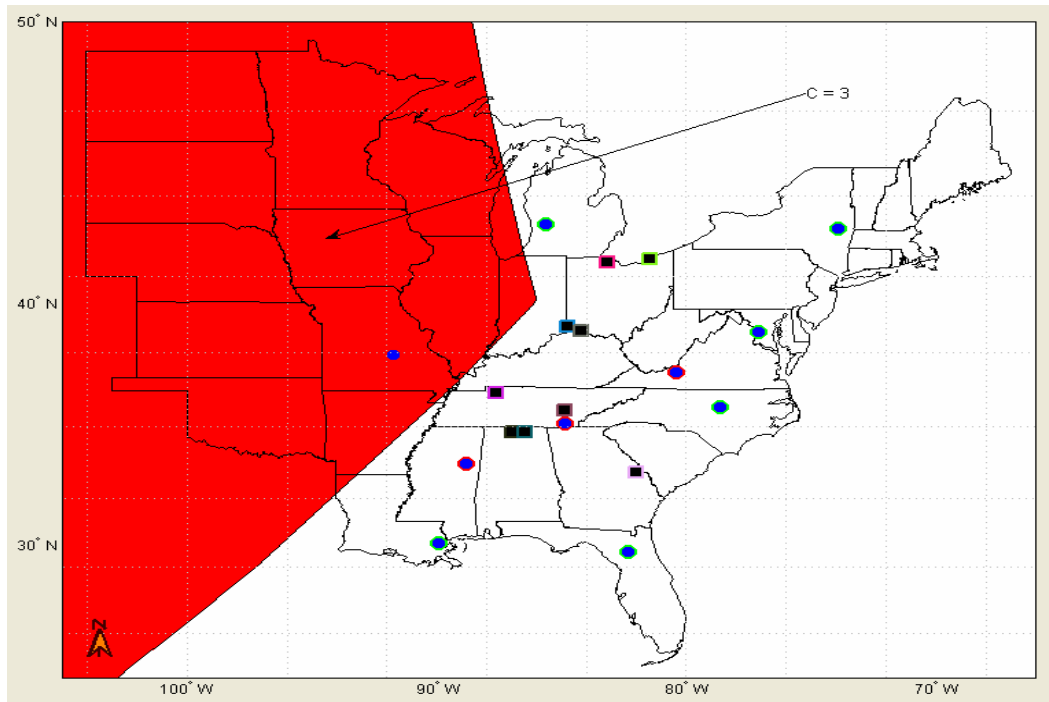


Figure 3-21: Hypocentral loci that meet constraints of base case 6.

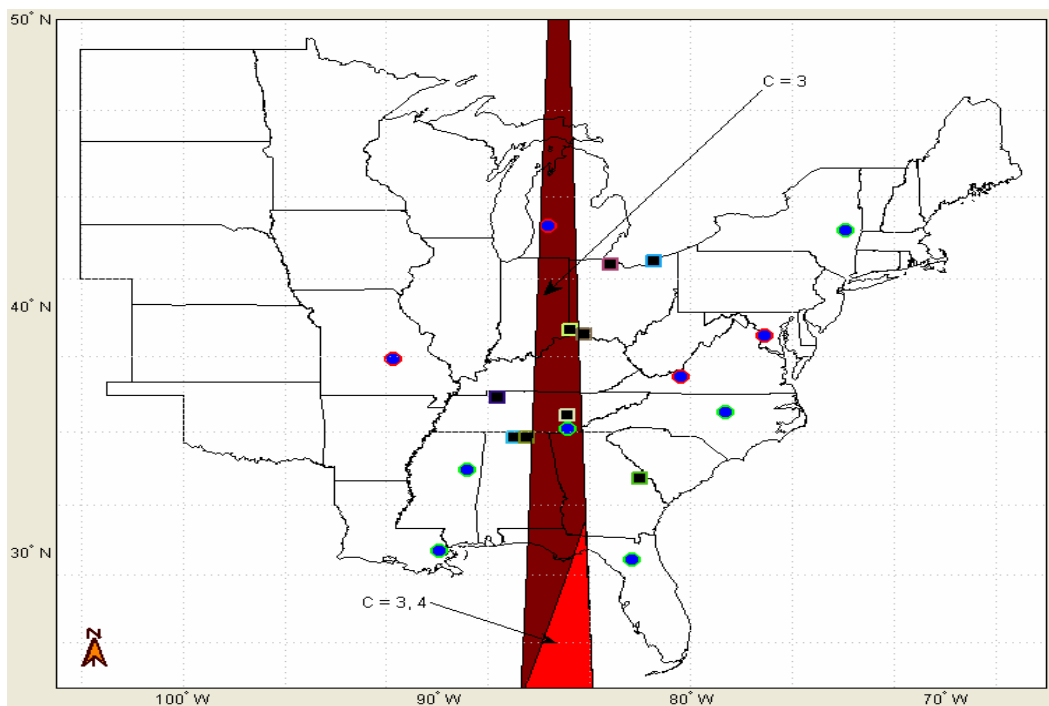


Figure 3-22: Hypocentral loci that meet constraints of base case 7.

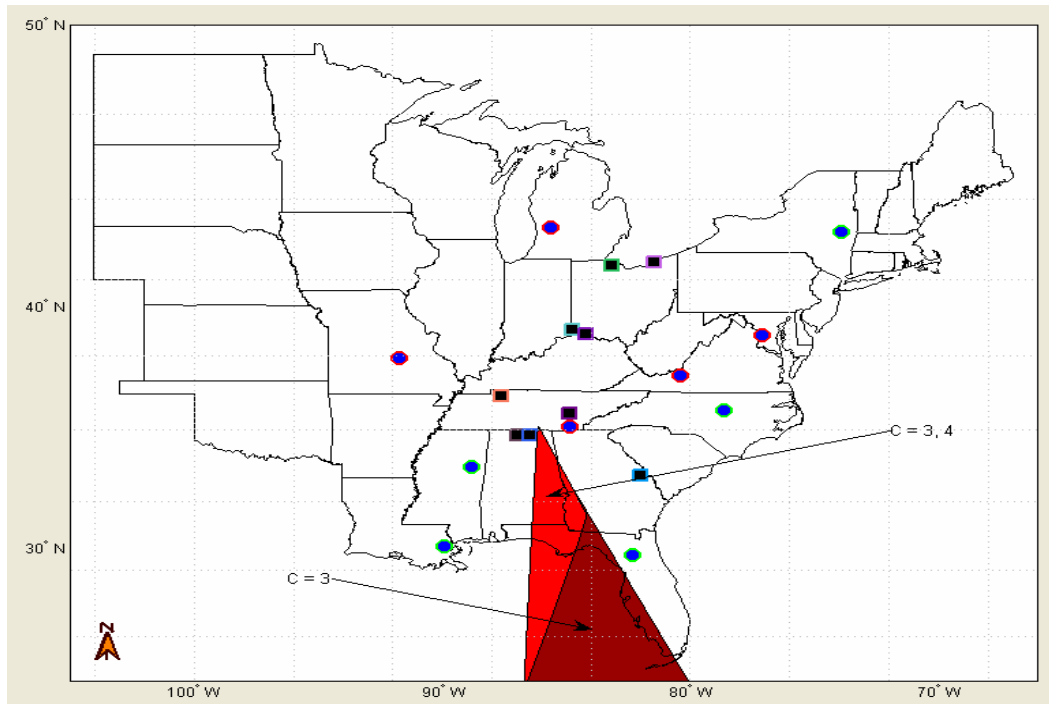


Figure 3-23: Hypocentral loci that meet constraints of base case 8.

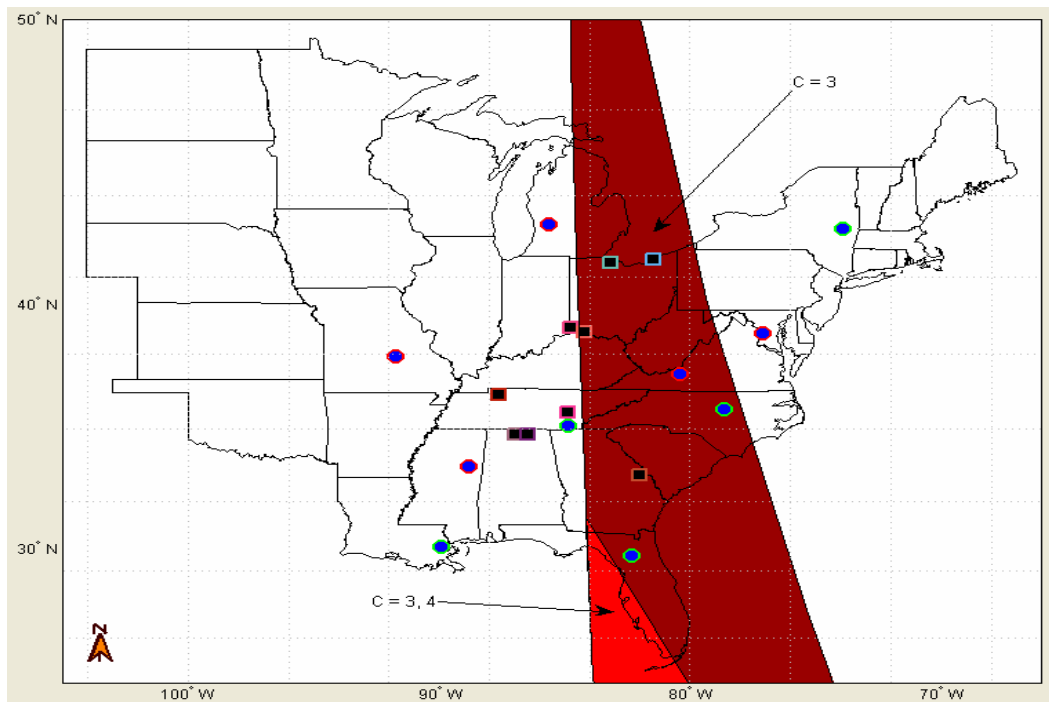


Figure 3-24: Hypocentral loci that meet constraints of base case 9.

3.4 Summary

In this chapter we have attempted the triangulation of disturbance location via the search for a LS solution to a set of over-constrained linear equations. While in some base cases there existed a solution associated with an assumed continuum velocity, not all of the base cases enjoyed satisfactory results. Even for the base cases that enjoyed a close solution based on an assumed continuum velocity, from figure 3-13 and table 3-1 we note that there was no consistency for assumed continuum velocities that minimized the distance between the LS solution and the actual location. In the previous section we empirically proved that methods based on the idea of increasing radii will also not work for our problem or at least have not worked in the past as determined by the base cases in this study. In many cases the system is over-constrained and a solution does not exist within our solution space. We now turn to two methods of solving non-linear sets of equations to further study the problem of power system event location.

Chapter 4

Non-Linear Triangulation Methods

We begin this chapter by reexamining the Pythagorean relation in a different light. To refresh, we write the relation as follows:

$$(x_i - x_h)^2 + (y_i - y_h)^2 + (z_i - z_h)^2 = V^2(t_i - t_h)^2$$

From this relation a linear set of equations was formed and then, using the pseudo-inverse, we found the LS solution. In this chapter we also start with the Pythagorean relation, in like manner as unto Chapter 3 ignoring the altitude parameters, and seek to find an alternate solution to this nonlinear algebraic equation:

$$(x_i - x_h)^2 + (y_i - y_h)^2 = V^2(t_i - t_h)^2$$

Written in broader terms as:

$$\underline{f}(\underline{\phi}) = \underline{y}$$

Where:

$$\underline{f}(\underline{\phi}) = \begin{bmatrix} f_1(\underline{\phi}) \\ f_2(\underline{\phi}) \\ \vdots \\ f_n(\underline{\phi}) \end{bmatrix} \quad \underline{\phi} = \begin{bmatrix} x_h \\ y_h \\ t_h \end{bmatrix} \quad \underline{y} = \begin{bmatrix} V^2 t_1^2 \\ V^2 t_2^2 \\ \vdots \\ V^2 t_n^2 \end{bmatrix}$$

Such that:

$$(x_i - x_h)^2 + (y_i - y_h)^2 - V^2(2t_i t_h - t_h^2) = V^2 t_i^2$$

4.1 Solutions to Non-Linear Systems of Equations

Newton's method and the closely related Newton-Raphson iterative method of solving nonlinear systems of equations are common to the study of power systems. The following is a short overview of the Newton's method [56, 57]:

We first place our equations in the form:

$$\underline{f}(\underline{\phi}) - \underline{y} = 0$$

Introducing the invertible $n \times n$ square matrix \underline{J} , we add $\underline{J}\underline{\phi}$ to both sides and rearrange:

$$\underline{J}\underline{\phi} = \underline{J}\underline{\phi} + \underline{y} - \underline{f}(\underline{\phi})$$

The following is obtained after multiplying both sides by \underline{J}^{-1} :

$$\underline{\phi} = \underline{\phi} + \underline{J}^{-1}[\underline{y} - \underline{f}(\underline{\phi})]$$

In the above equation, the updated $\underline{\phi}$ is on the left side and is calculated from the previous $\underline{\phi}$ on the right side. The process is iterative which actually leads to writing the above equation as:

$$\underline{\phi}(i+1) = \underline{\phi}(i) + \underline{J}^{-1}(i)[\underline{y} - \underline{f}(\underline{\phi}(i))]$$

The matrix \underline{J} is the Jacobian as defined by:

$$\underline{J}(i) = \left. \frac{\partial \underline{f}}{\partial \underline{\phi}} \right|_{\underline{\phi}=\underline{\phi}(i)} = \begin{bmatrix} \frac{\partial f_1}{\partial \phi_1} & \frac{\partial f_1}{\partial \phi_2} & \dots & \frac{\partial f_1}{\partial \phi_n} \\ \frac{\partial f_2}{\partial \phi_1} & \frac{\partial f_2}{\partial \phi_2} & \dots & \frac{\partial f_2}{\partial \phi_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \phi_1} & \frac{\partial f_n}{\partial \phi_2} & \dots & \frac{\partial f_n}{\partial \phi_n} \end{bmatrix}_{\underline{\phi}=\underline{\phi}_i}$$

A method similar to Newton's method can be implemented using the `fsolve` command in MATLAB's Optimization Toolbox [55]. This method has been used for the triangulation of contingency location for each of the nine base cases. In each case, the initial guess used for $\underline{\phi}_0$ was created as follows:

$$\underline{\phi}_0 = \begin{bmatrix} x_0 \\ y_0 \\ t_0 \end{bmatrix} = \begin{bmatrix} \frac{1}{k} \sum_{i=1}^k x_i \\ \frac{1}{k} \sum_{i=1}^k y_i \\ \min(t_i) \end{bmatrix}$$

where x , y , and t , are the latitude, longitude, and arrival time parameters associated with each of the k base cases that measured the disturbance respectively. In general, when using such iterative techniques for the solution of nonlinear sets for equations, it is imperative to understand that the more variables that exist to be solved for, the closer the initial guess needs to be to the actual solution. This reason is why, again, we assuming a continuum by using the same velocity parameter, V , for each equation. As in Chapter 3, we shall solve each set of equations for a continuum velocity value varying from 100 mi/s to 1000 mi/s by increments of 50 mi/s.

The following figures show the hypocentral solution set using the above nonlinear solution method where each element of the solution set is plotted with a green triangle. The closest solution location is enlarged and pointed out with an arrow. Again, the base case being triangulated is plotted with a square and the FDR locations are shown using blue dots.

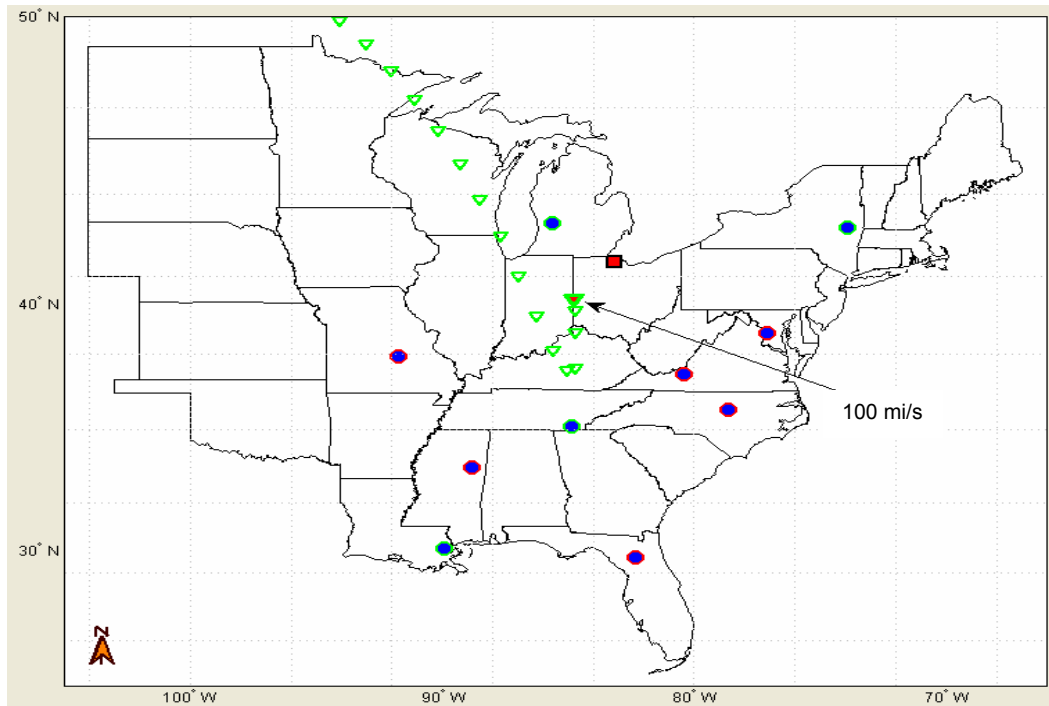


Figure 4-1: Newton triangulation of base case 1.

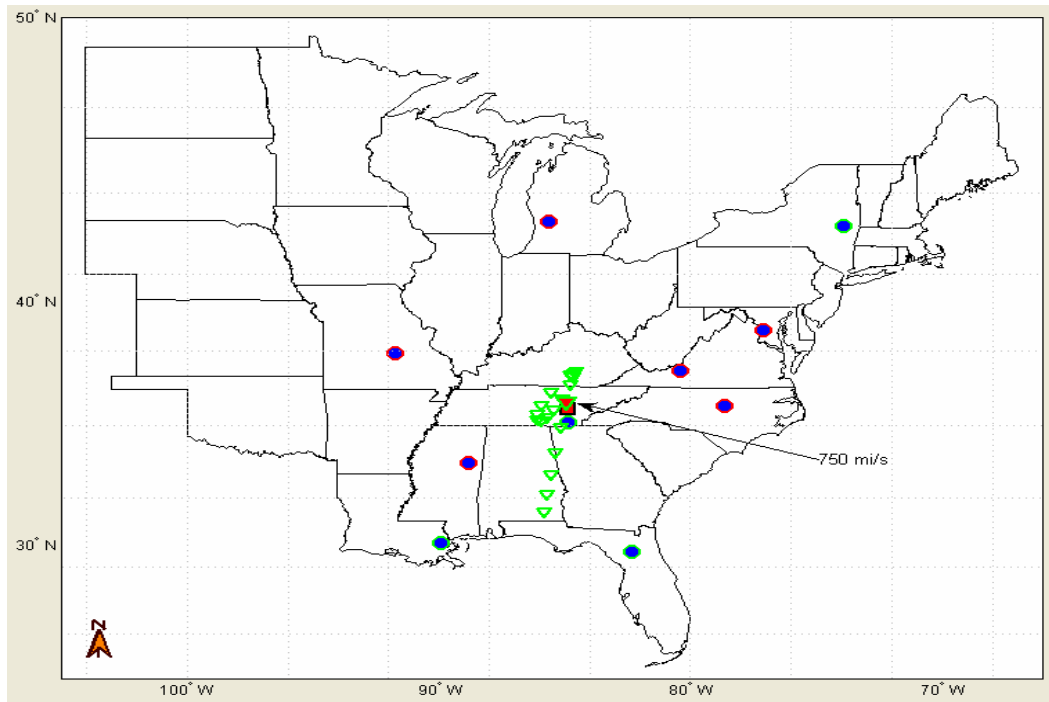


Figure 4-2: Newton triangulation of base case 2.

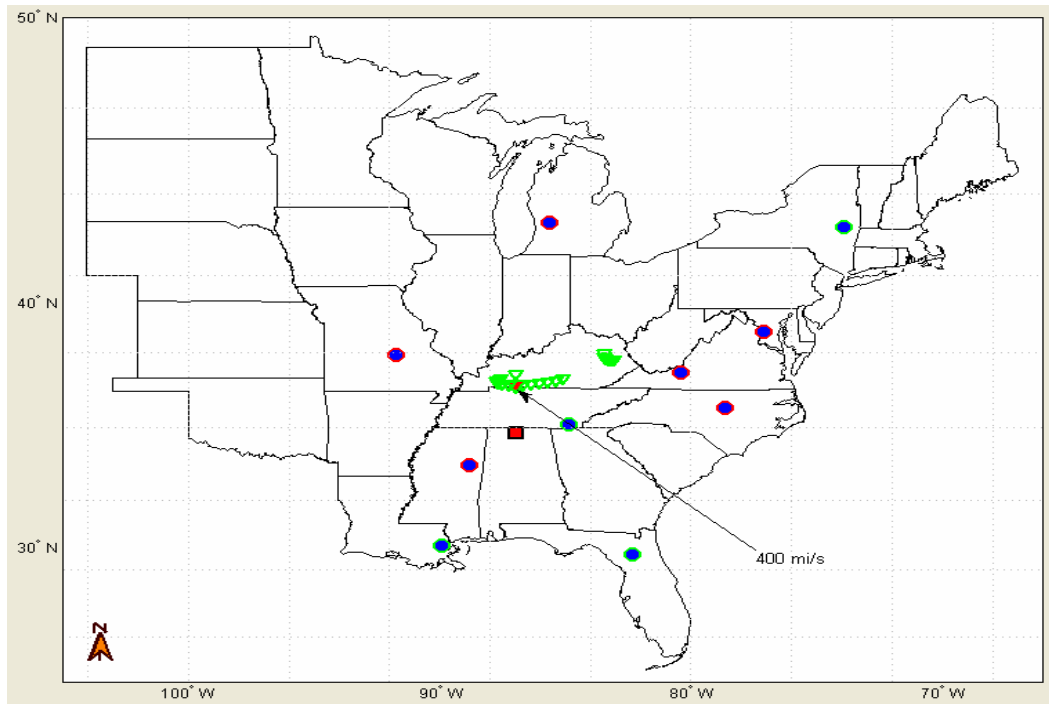


Figure 4-3: Newton triangulation of base case 3.

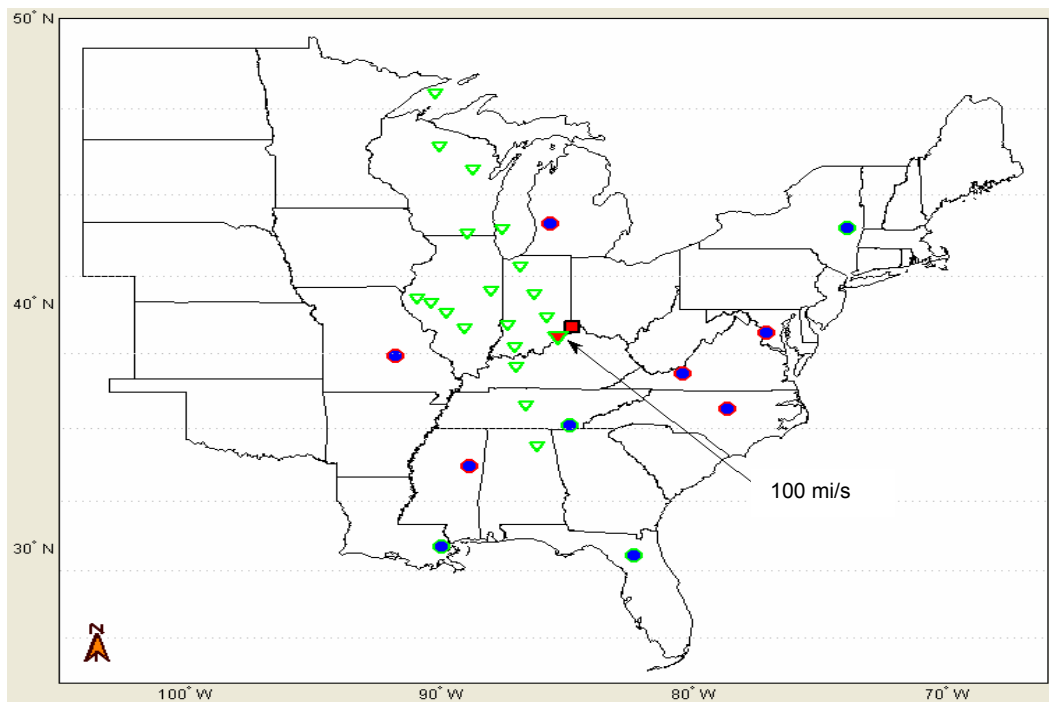


Figure 4-4: Newton triangulation of base case 4.

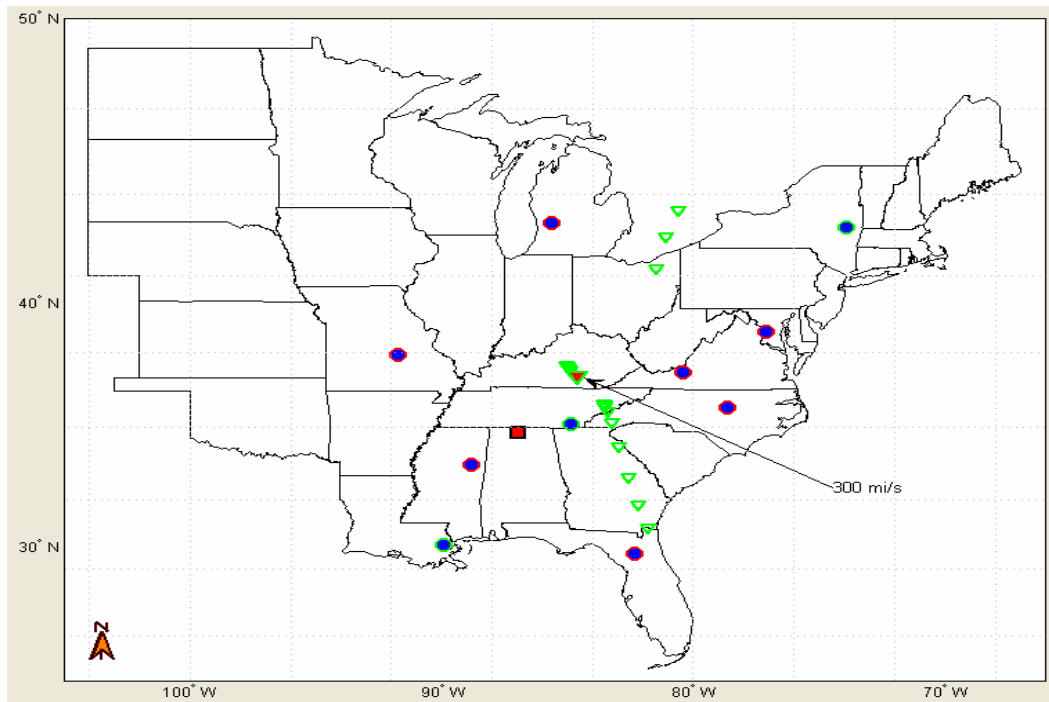


Figure 4-5: Newton triangulation of base case 5.

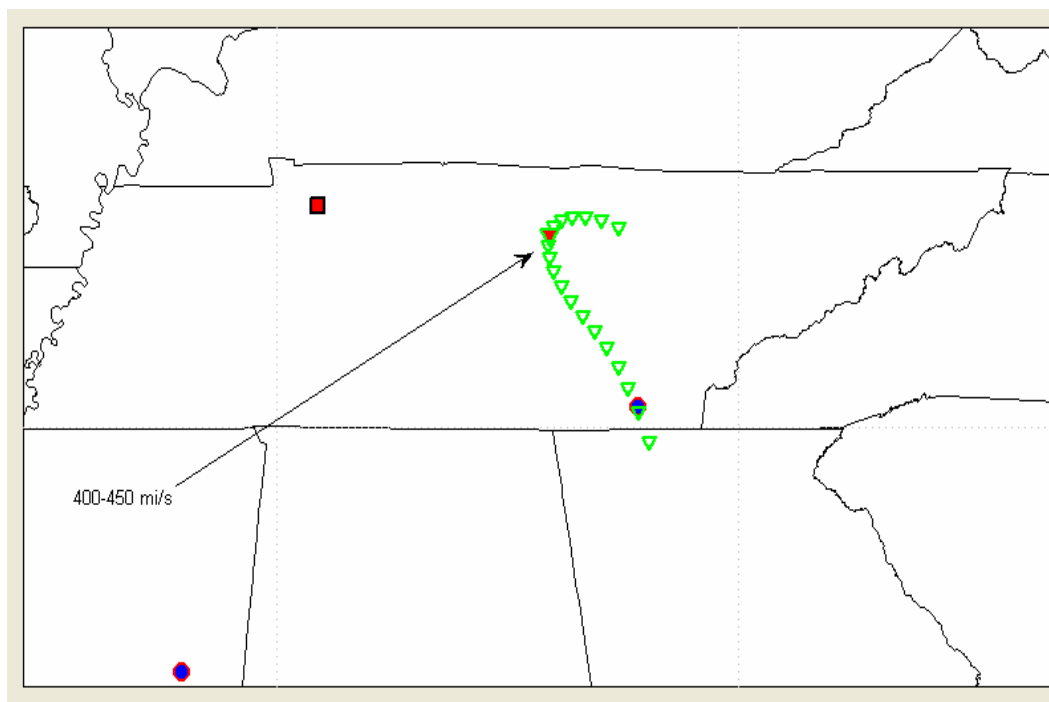


Figure 4-6: Newton triangulation of base case 6 (enlarged for clarity).

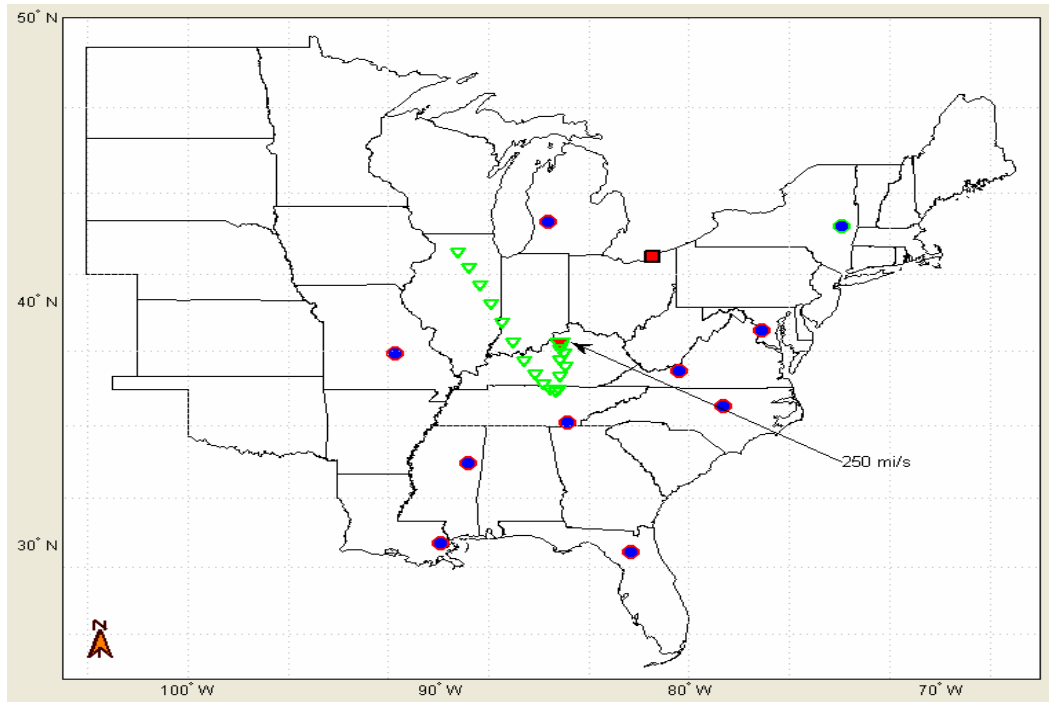


Figure 4-7: Newton triangulation of base case 7.

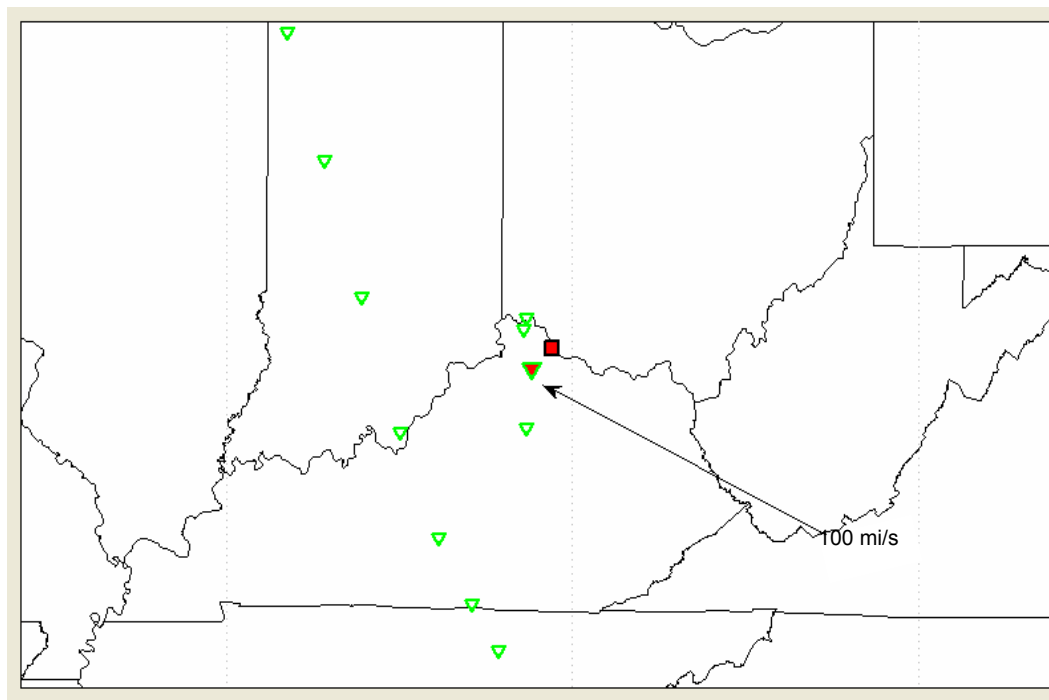


Figure 4-8: Newton triangulation of base case 8 (enlarged for clarity).

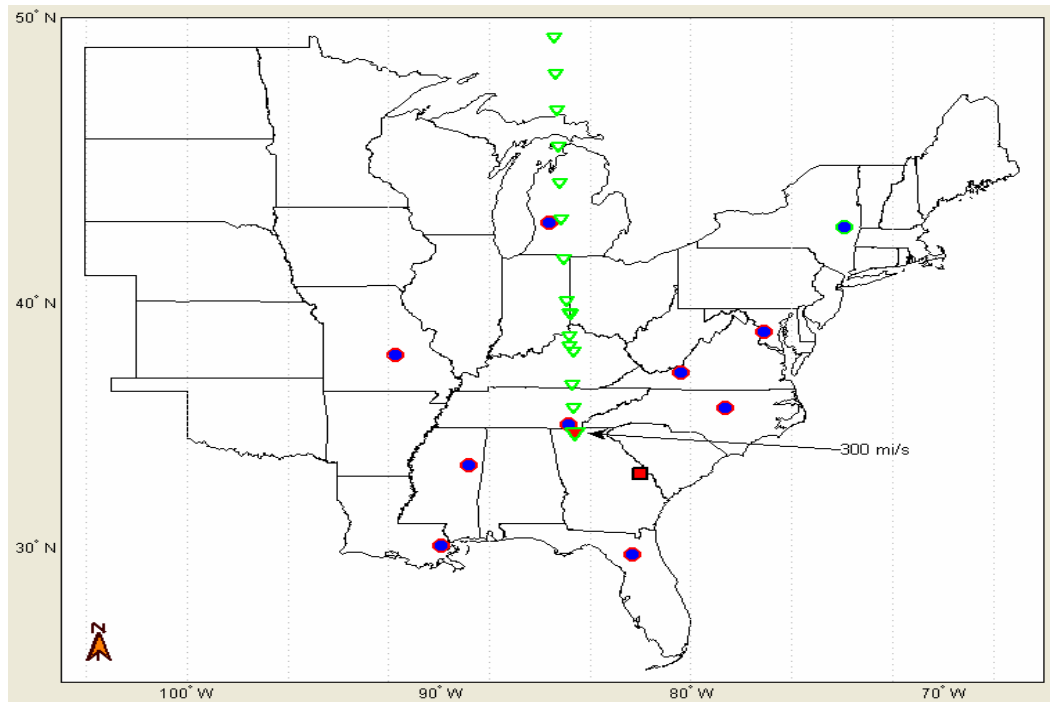


Figure 4-9: Newton triangulation of base case 9.

The following table shows the distances between the actual location of each base case and the loci of solutions from the linear LS triangulation method. Shortest distances are highlighted in red. The following figure is a plot of the data shown in the table below. It can be seen that, again, there is no common continuum velocity that can be assumed since there is no common velocity value that minimizes every curve.

Table 4-1: Numerical results for Newton triangulation method.

<i>Continuum</i> <i>Speed</i> <i>(mi/s):</i>	<i>Solution Distances from Base Case (mi):</i>								
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
100	129	30	184	40	208	146	346	18	416
150	202	31	171	61	214	137	320	24	473
200	291	52	158	119	215	129	305	21	469
250	304	68	147	194	211	123	304	50	373
300	266	75	139	294	204	118	315	258	188
350	216	71	134	452	209	114	340	214	241
400	202	60	132	645	209	112	378	184	294
450	243	58	134	532	209	112	411	161	390
500	313	94	139	327	209	114	419	134	507
550	394	105	144	198	209	118	419	107	613
600	475	101	150	138	210	123	415	122	709
650	556	90	155	134	214	129	404	177	799
700	634	68	163	160	232	137	389	243	886
750	708	18	297	236	283	145	372	307	971
800	779	57	293	335	346	153	361	369	1054
850	846	129	288	228	409	161	359	428	1138
900	909	194	287	267	535	170	367	484	1221
950	969	252	288	303	617	179	381	538	1305
1000	1025	305	295	335	685	190	402	589	1390

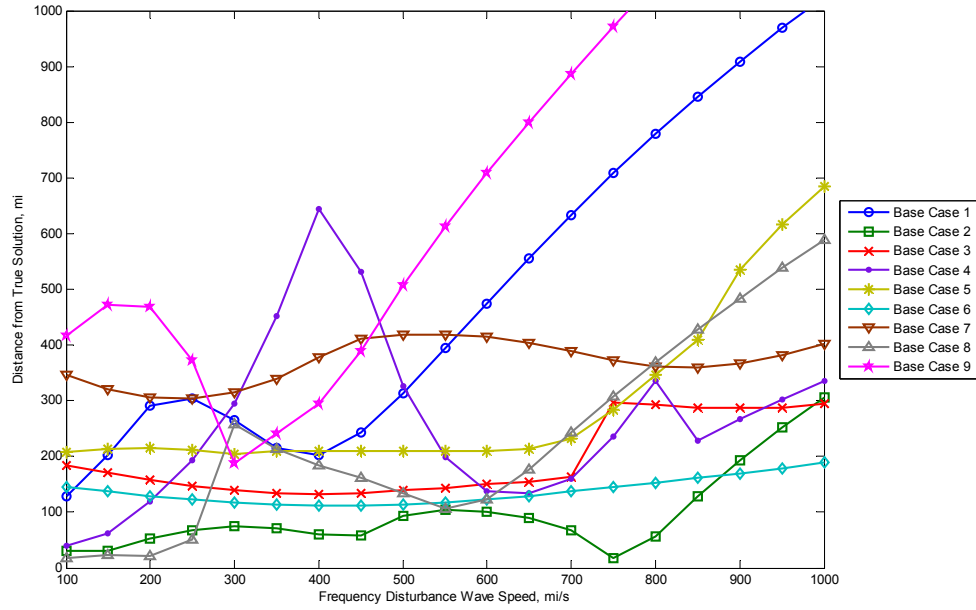


Figure 4-10: Plot of Newton error versus assumed continuum speed.

4.2 The Gradient Search Method

Thus far, the main factor that seems to be inhibiting progress is the inability to accurately and consistently determine a mean continuum velocity. This method, therefore, does not require any assessment or approximation of the velocity of the electromechanical wave front. The consequence, however, is that the event occurrence time is not calculated.

The method used here is based upon minimizing the distance between the purported event location and each FDR. This method, however, would lead to the same solution for each case where the observation set of FDRs is the same. Therefore, the constraint first mentioned in Section 3.3 is again applied to the minimization process as follows:

$$r_1 \leq r_2 \leq \dots \leq r_n$$

where r_i is associated with the i^{th} measurement unit and all measurement units are numbered in ascending order according to disturbance detection time. The following is the objective function that we wish to minimize:

$$\phi(x_h, y_h) = \sum_{i=1}^n r_i^2$$

Where r_i is defined as:

$$r_i = \sqrt{(x_i - x_h)^2 + (y_i - y_h)^2}$$

Again, the variable conventions are consistent with the rest of this report. Our objective function, ϕ , can be minimized via finding the steepest descent using the gradient search method [55, 58]. The basics of the gradient search method can be seen in the following steps. First the gradient of the objective function is calculated:

$$\nabla \phi(x_h, y_h) = \begin{bmatrix} \frac{\partial \phi}{\partial x_h} \\ \frac{\partial \phi}{\partial y_h} \end{bmatrix}$$

The gradient is negated to find the steepest descent, multiplied by a scalar, α , and then added to the initial guess to find an updated guess as follows:

$$\begin{bmatrix} x_h(i+1) \\ y_h(i+1) \end{bmatrix} = \begin{bmatrix} x_h(i) \\ y_h(i) \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial \phi}{\partial x_h} \\ \frac{\partial \phi}{\partial y_h} \end{bmatrix}$$

The process is iterated until convergence is found. The command for minimization using gradient search in MATLAB (with the Optimization toolbox) is `fmincon`. The trouble with this method, however, is that it is very sensitive to the order in which the FDRs measured the disturbance. Hence, the process is repeated several times depending on how many FDRs measured the disturbance. If n FDRs measured the disturbance, then the process is repeated $n-2$ times with

the above distance constraints added incrementally beginning with three constraints determined according to which FDRs first measured the disturbance. The flowchart below illustrates this process further.

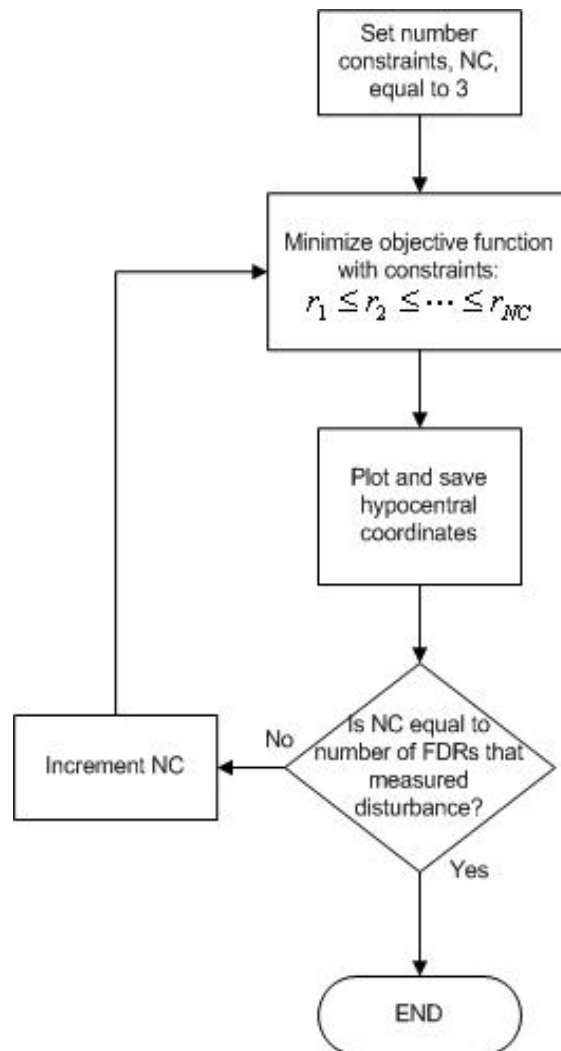


Figure 4-11: Flow chart of algorithm to find hypocentral coordinates using gradient search.

The following figures show the results of this triangulation method. Each green diamond represents a minimization with number of constraints, NC. The

closest results are noted with red and the number of constraints active during that particular calculation.

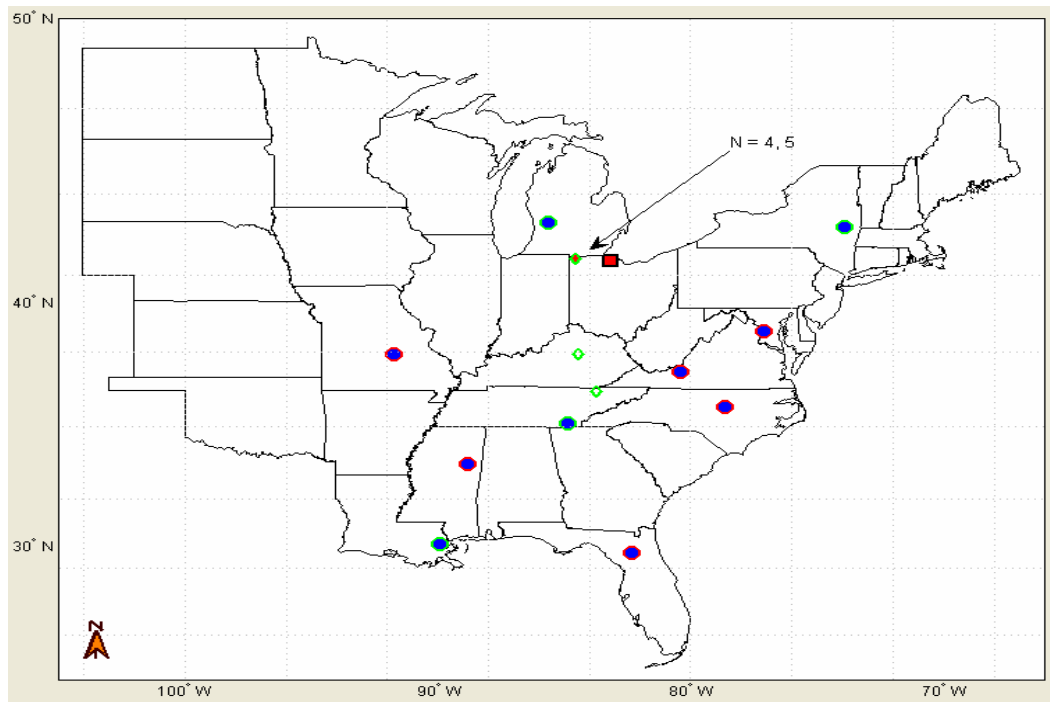


Figure 4-12: Gradient Search triangulation of base case 1.

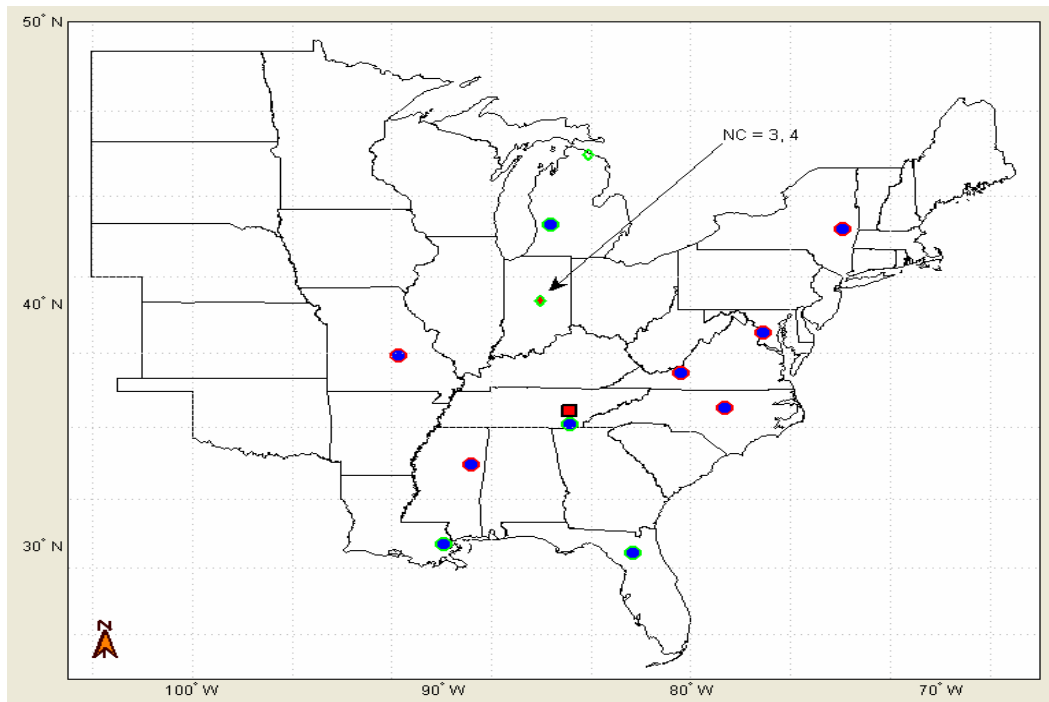


Figure 4-13: Gradient Search triangulation of base case 2.

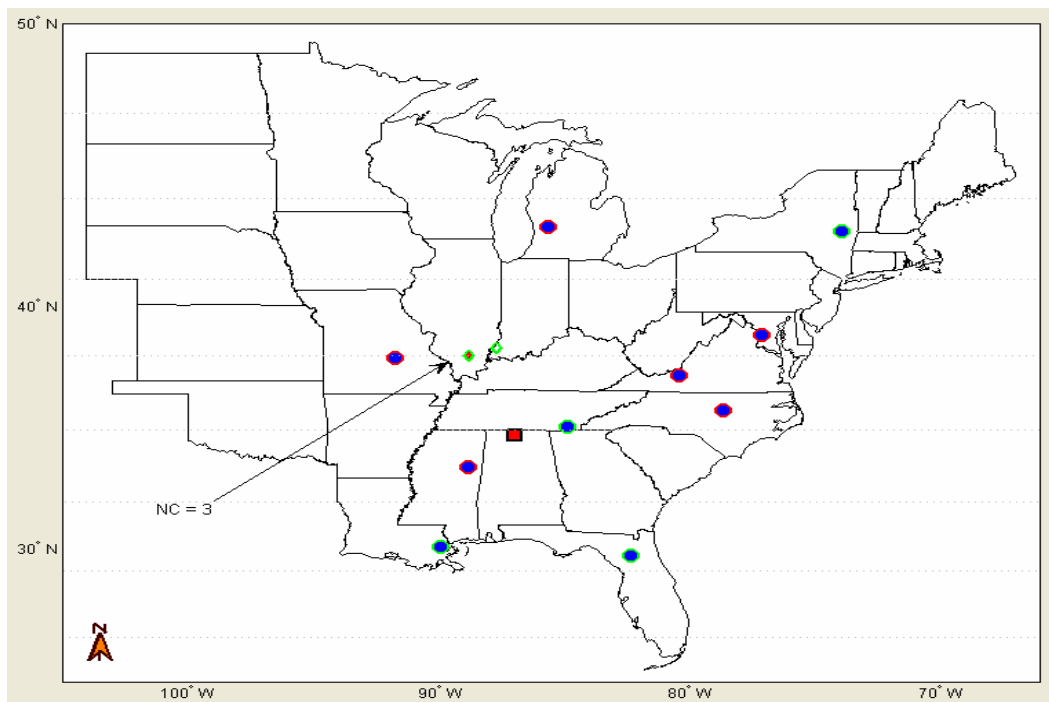


Figure 4-14: Gradient Search triangulation of base case 3.

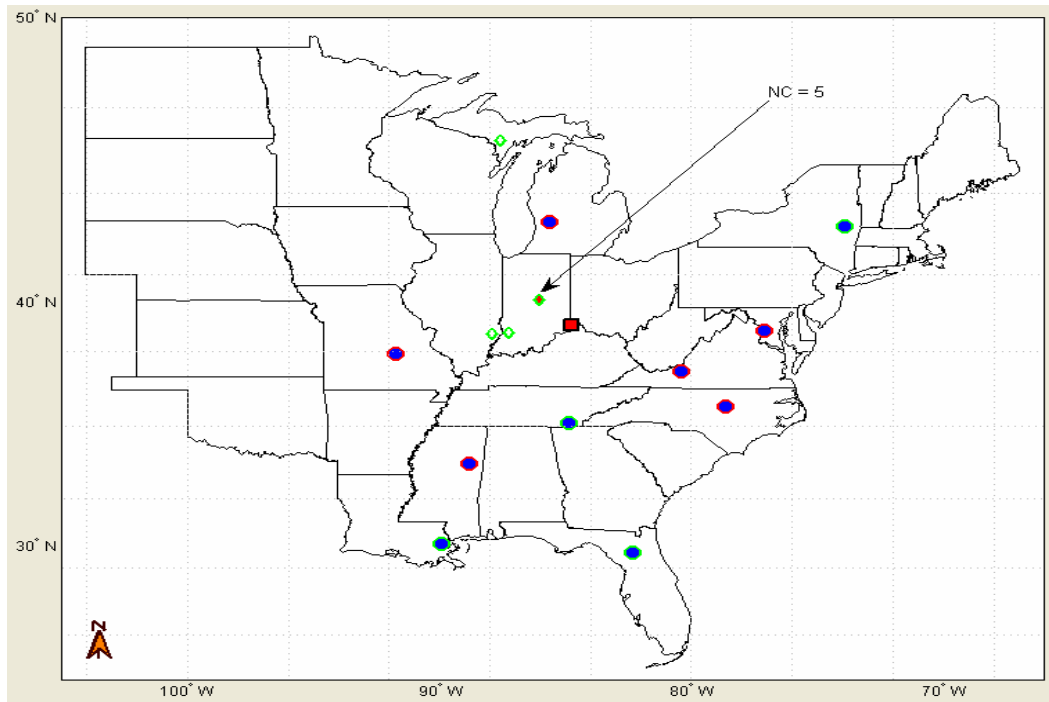


Figure 4-15: Gradient Search triangulation of base case 4.

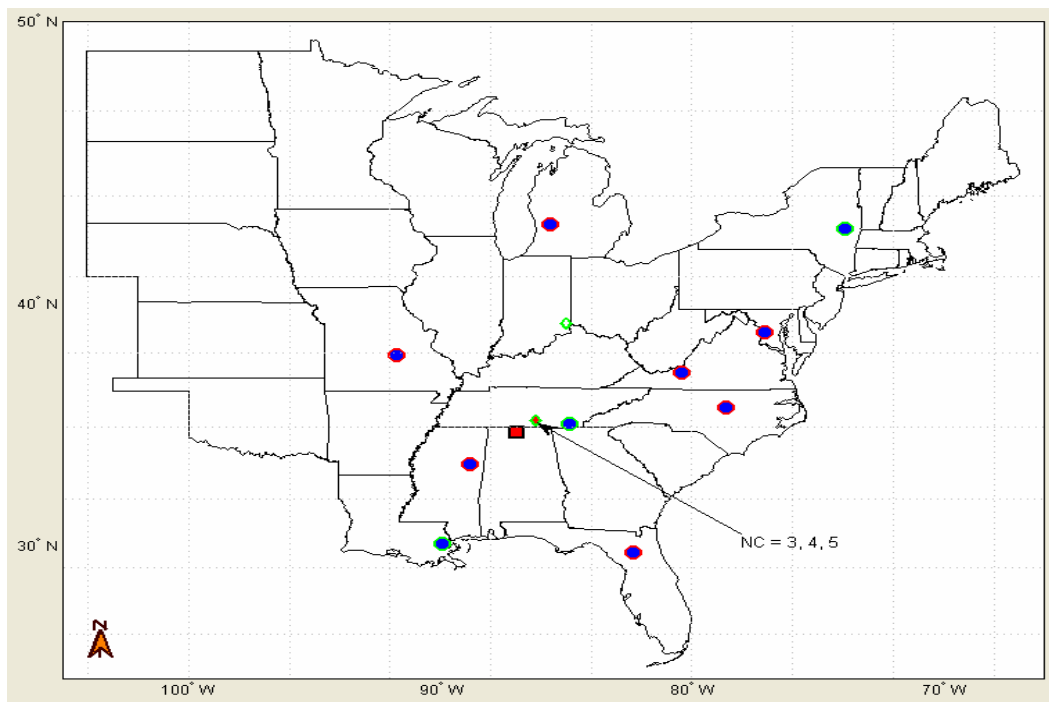


Figure 4-16: Gradient Search triangulation of base case 5.

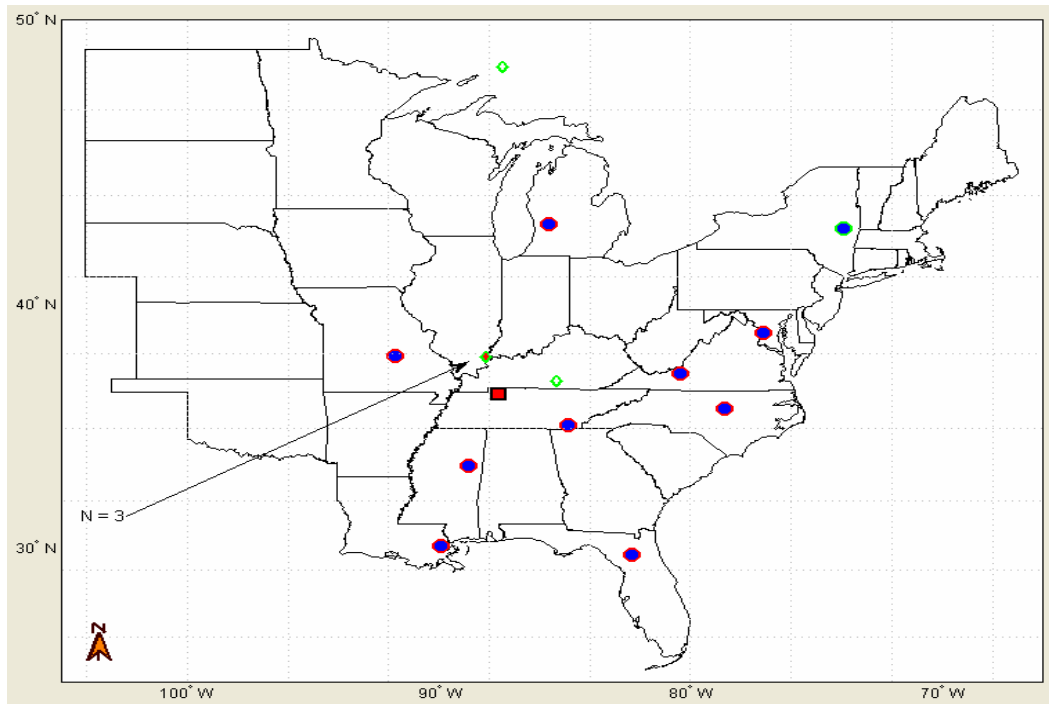


Figure 4-17: Gradient Search triangulation of base case 6.

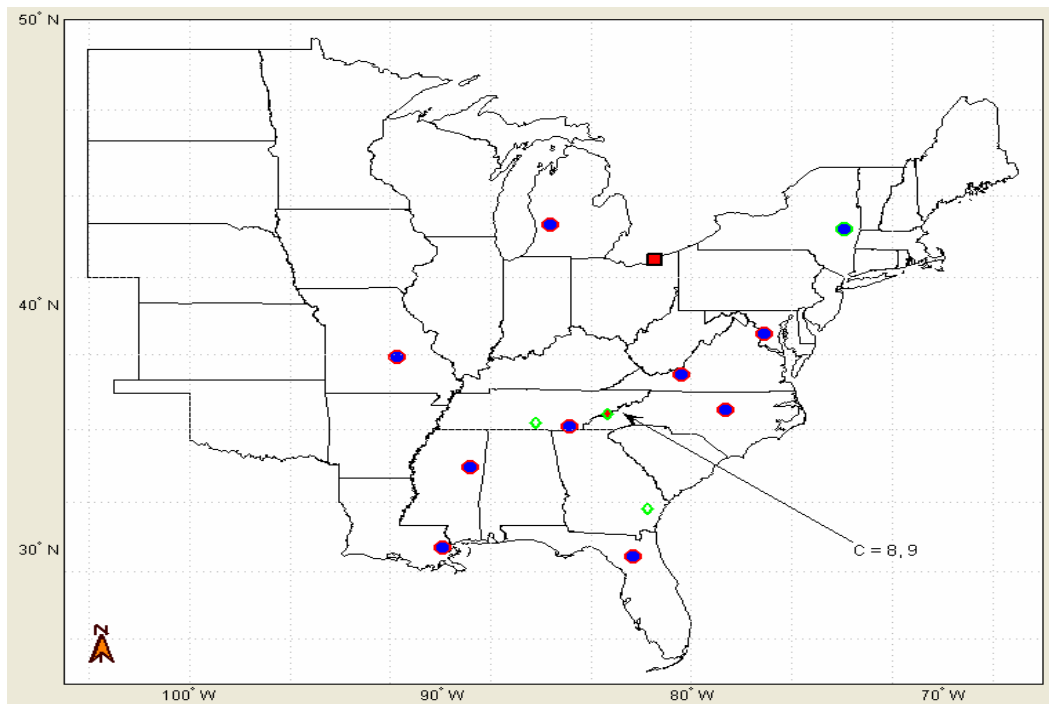


Figure 4-18: Gradient Search triangulation of base case 7.

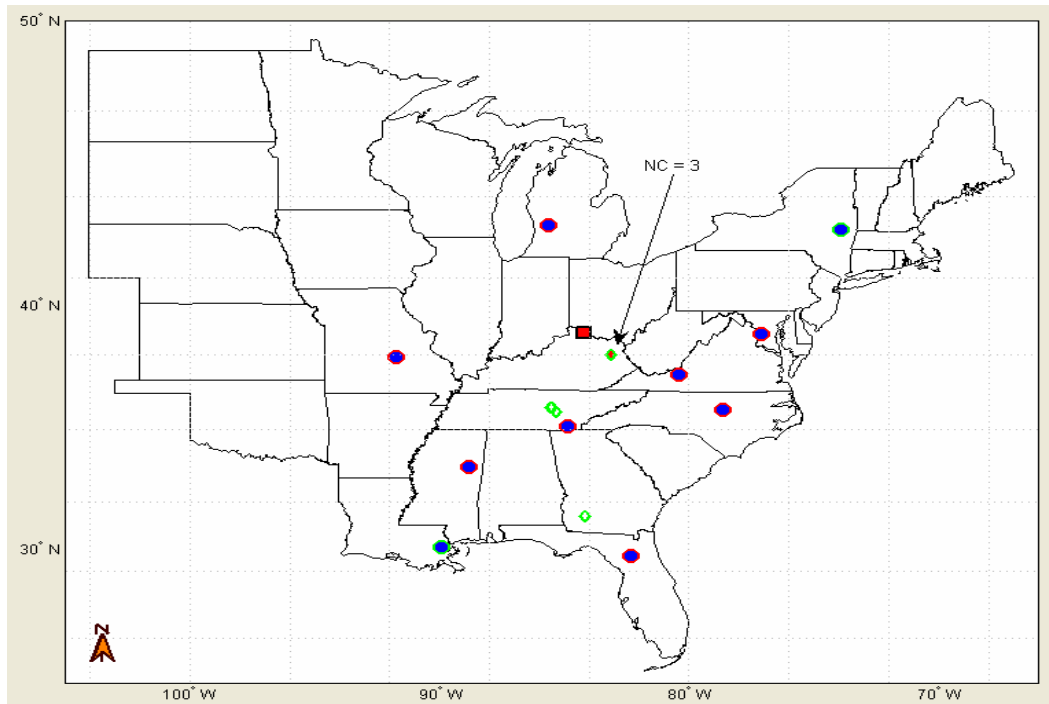


Figure 4-19: Gradient Search triangulation of base case 8.

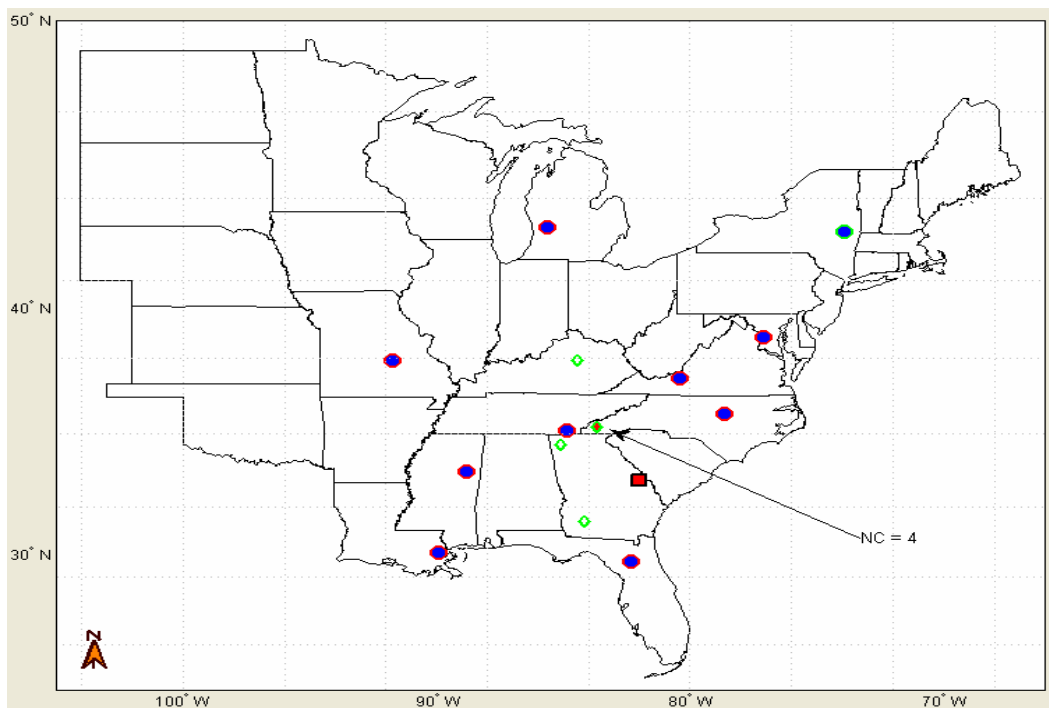


Figure 4-20: Gradient Search triangulation of base case 9.

The following table shows the distance from the true solution versus the number of constraints placed upon the minimization for each of the nine base cases. The constraints below, n , refer to the ascending order of the first n radii associated with the first n units to measure the disturbance under study. This table, along with the following figure shows us that in general, the (constrained) gradient search method seems to work best with fewer than five constraints. In five of the nine cases, the solution error was minimized with three constraints. This notion might be promising for the future of FNET as the number of FDRs increases and the resolution of the solution space increases.

Table 4-2: Numerical results for Gradient Search triangulation method.

<i>Active Constraints:</i>	<i>Solution Distances from Base Case (mi):</i>								
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
3	257	309	245	171	56	107	510	85	360
4	72	309	246	137	56	834	510	516	172
5	72	1009	246	97	56	834	689	225	178
6	353	678	246	492	322	134	689	216	202
7	--	--	--	--	322	134	689	216	202
8	--	--	--	--	--	134	428	216	202
9	--	--	--	--	--	134	428	--	202

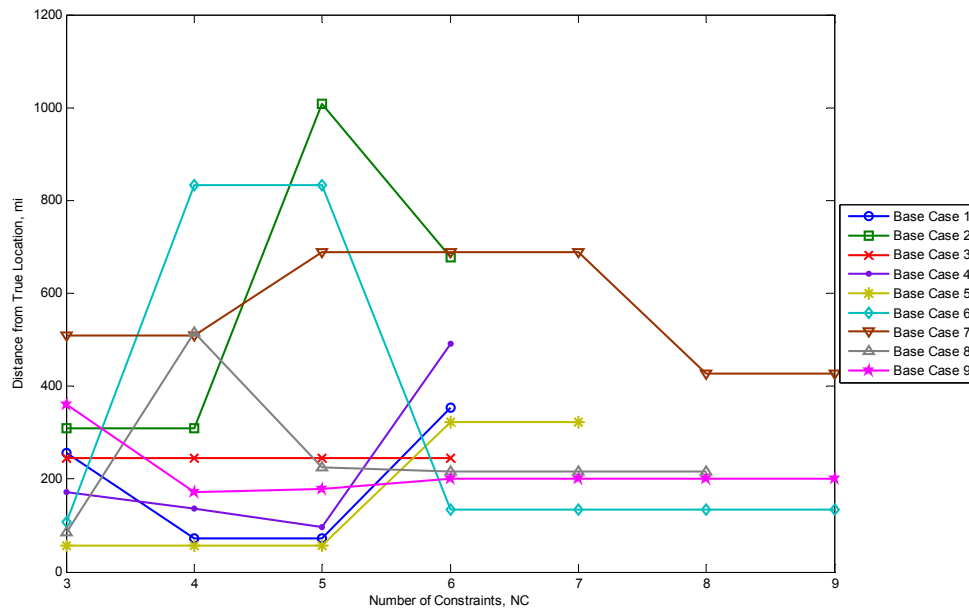


Figure 4-21: Gradient Search error versus number of active constraints.

The above plot shows pictorially that, in general, the gradient search method more accurately triangulates the location of the frequency disturbance with fewer constraints.

4.3 Summary

This chapter opened by attempting to find a better solution to the triangulation problem by solving an over-constrained set of nonlinear equations as defined by the Pythagorean relation from Chapter 3. In most cases the results were better but there is yet work to be done to achieve a satisfactory triangulation method. Unlike the LS approach taken in Chapter 3, Newton's method generally gave better results when the assumed continuum velocity, V , was less than 450 mi/s. In some cases, with an assumed V , Newton's method produced results within 20 miles of the actual hypocenter. The problem still exists, however, of needing to know which V to specify. Other than the general need for V to be less

than 450 mi/s, there is little indication of a value that would work universally for all possible cases.

To avoid the need of specifying V , we turned to the gradient search method and eliminated the element of solving for time all together. The gradient search method was used to minimize the sum of the distances (squared) from the candidate hypocenter to each valid FDR. Knowing that this method would always give the same answer in all situations with the same set of valid FDRs, we placed an increasing number of constraints on the method such that the distance from the FDR that first measured the disturbance to the candidate location would be less than the distance from the FDR that measured the disturbance second, and so fourth. We learned empirically that this method produced best results when the number of constraints was between three and five. Under such constraints, we were able to triangulate the disturbance with an error less than 60 miles. However, like the LS method of Chapter 3, when this triangulation method guessed incorrectly, it did so with huge error. Such volatility is extremely undesirable and problematic. Nevertheless, this method might show some promise when the number of FDRs in the field is increased.

Chapter 5

Conclusions and Future Work

The thesis opened with a suggestion for the treatment of FDR data. We concluded that the nature of our data made the moving median more suitable for data conditioning purposes than the moving mean due to the robustness of the median against data outliers. We then improved upon the moving median with the goal of further smoothing the FNET data such that the point of maximum slope could be clearly and easily determined. To accomplish this smoothing, we used the LS method as a means of fitting a seventh order polynomial to a small

window of our data. We also used the median and MAD as an aid in rejecting outliers.

The second part of this thesis attempted to use a conventional triangulation method based upon the Pythagorean relation to triangulate the origin of generator trips within the EI using FNET data captured by FDRs. Data collected from nine previous generator trips and ten FDRs were analyzed and the performance of the triangulation method presented.

5.1 Conclusions

The data conditioning portion of the thesis was quite conclusive. The moving median should henceforth replace the moving mean as used to smooth FDR data. Furthermore, a GUI was developed and is now in use that aids in the conditioning, viewing, and analysis of FNET data.

The second part of this thesis highlighted areas where future work is needed in the area of power system generator trip triangulation. The following figure presents the performance of each of the three triangulation methods set forth in Chapters 3 and 4. The average distance between all of the nine base cases from the actual hypocenter and the solved location are plotted for varying values of assumed continuum velocity, V .

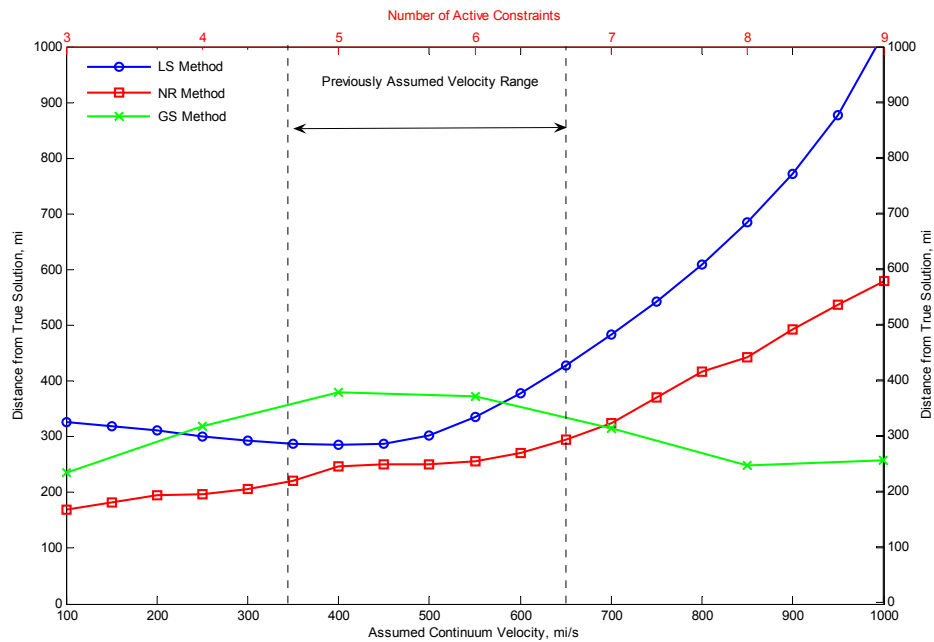


Figure 5-1: Performance of triangulation methods.

The above figure shows us several things about the triangulation methods discussed within this thesis. Looking at the vertical axis, we notice that the average distance between the true location and the solved location is at a minimum around 280 miles for the LS method, 180 miles for Newton's method, and 230 miles for the gradient search method. In short, the precision of these methods leaves much to be desired. From the data tables in the previous chapters, we see that the accuracy is also rather low. From the above figure we can see the how the behavior of the LS and Newton's methods depends on the assumed V (bottom axis). The LS and Newton's methods seem to lose precision very quickly after V exceeds approximately 600 mi/s. Nevertheless, the precision of the LS method seems optimal for an assumed V within the velocity window previously assumed to be the range of velocities for power system disturbances within the EI (~350-650 mi/s) [2, 23]. The precision and accuracy of the gradient search method (associated with top axis) seem the lowest at this juncture. From our

limited study with only a few FDRs, there is no noticeable trend associated with the number of active constraints and the precision of the triangulation algorithm.

The vast and heterogeneous nature of the EI likely plays a major role in the imprecision of these triangulation methods. With the entire expanse of the EI as a backdrop, the triangulation methods within this report base their results on but a small understanding of the grid via the measurements of only ten FDRs spaced hundreds of miles apart. These uncertainties are unsettling in an environment where even modest uncertainty can seriously affect results in a negative way. To improve the precision of the triangulation, at a minimum, the following must occur:

- More FDRs must be placed.
- New units must increase data output rate.

A larger number of FDRs will increase our knowledge about the propagation of frequency perturbation. At the time of this study, the base case information was sparse and the FDRs that experienced the base cases were few. Furthermore, as stated in Chapter 3, the sampling time of 0.1 s is too slow for the precision we seek. From the results in this study, it seems possible that in 0.1 s a frequency disturbance wave could travel upwards of 100 miles.

5.2 Future Work

There are five main areas where I feel future research is necessary in the area of power system event triangulation. They are as follows:

- A large number of base cases should be simulated, possibly using simulation software such as PSS/E, in order to derive as accurate a velocity model of the EI as possible. Simulation may be key in forming a velocity model of the EI because, in such a closed environment, the

inception time of any simulated event is both controlled and known. For the existing base cases, there is no information about the actual inception time of the inciting generator trip. To create a velocity model, the time of the event must be known, and it must be known with high accuracy. Hence, to complement simulation data, it is necessary to seek timing data from PMUs near base case events as they occur.

- Conventional mathematics based on the Pythagorean relation as contained herein may not be the best way to approach this problem anymore. Such methods are very rigid and deterministic and thus cannot robustly deal with the errors that are common in FNET data. Perhaps techniques taken from neural and fuzzy theory would be beneficial in creating a system that could learn from previous patterns and suggests new solutions. Further research in neural networks suggested.
- Obviously, the topology of the EI is in some way associated with the propagation of frequency disturbance waves. A triangulation method that in some manner takes into account the model of the EI would be ideal. Before the controllability or the observability of the EI can be assessed, the sheer size of the state-space model requires some form of model reduction. To aid in such an effort, perhaps the Virginia Tech supercomputer, System X, would be of use.
- A re-examination of continuum modeling of power systems is also suggested. In this research, however, the calculation of the discriminant would use FNET data. The discriminant, which aids in the determination of velocity and timing of electromechanical waves is defined as [6, 9, 10, 59]:

$$D_f = \omega + Z_0 p$$

$$D_r = \omega - Z_0 p$$

Where p is defined as follows:

$$p = -b \frac{\partial \delta}{\partial x}$$

To calculate p , an FDR has been placed arbitrarily close to the unit at Virginia Tech in an effort to calculate the above derivative. The calculation of the discriminant at different point throughout the grid could be beneficial to the efforts of forming a velocity model.

- Finally, a study of the tools currently used in range finding might also prove helpful. Several other technologies such as SONAR and RADAR use range finding regularly. A specific area study in the area of range finding called *passive ranging* seems helpful since this area has been developed for signals arriving from objects with nebulous signal emission time[60-63].

References

1. De La Ree, J., et al., *Catastrophic failures in power systems: causes, analyses, and countermeasures*. Proceedings of the IEEE, 2005. **93**(5): p. 956.
2. Tsai, S.S., *Study of Global Power System Frequency Behavior Based on Simulations and FNET Measurements*, in *Electrical and Computer Engineering*. 2005, Virginia Tech: Blacksburg. p. 247.
3. Huang, L., *Electromechanical Wave Propagation in Large Electric Power Systems*, in *Electrical and Computer Engineering*. 2003, Virginia Tech: Blacksburg.
4. Tsai, S.S., et al. *Study of global frequency dynamic behavior of large power systems*. 2004.
5. Pierre, J.W., D.J. Trudnowski, and M.K. Donnelly, *Initial results in electromechanical mode identification from ambient data*. Power Systems, IEEE Transactions on, 1997. **12**(3): p. 1245.
6. Thorp, J.S., C.E. Seyler, and A.G. Phadke, *Electromechanical wave propagation in large electric power systems*. Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on [see also Circuits and Systems I: Regular Papers, IEEE Transactions on], 1998. **45**(6): p. 614.
7. Machowski, J., J. Bialek, and J.R. Bumby, *Power system dynamics and stability*. 1997, New York: John Wiley. xxii, 461.
8. Parashar, M., *Continuum Modeling of Power Networks*, in *Electrical Engineering*. 2003, Cornell University: Ithaca.
9. Parashar, M., J.S. Thorp, and C.E. Seyler, *Continuum modeling of electromechanical dynamics in large-scale power systems*. Circuits and Systems I: Regular Papers, IEEE Transactions on [see also Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on], 2004. **51**(9): p. 1848.
10. Thorp, J.S., et al., *The large scale electric power system as a distributed continuum*. Power Engineering Review, IEEE, 1998. **18**(1): p. 49.
11. Aki, K. and P.G. Richards, *Quantitative seismology: theory and methods*. 1980, San Francisco: W. H. Freeman. 2 v. (xiv, 932, 16).
12. Bertsch, J., et al. *Experiences with and perspectives of the system for wide area monitoring of power systems*. 2003.
13. Bin, Q., et al. *Internet based frequency monitoring network (FNET)*. 2001.
14. Phadke, A.G., *Synchronized phasor measurements in power systems*. Computer Applications in Power, IEEE, 1993. **6**(2): p. 10.

15. Horowitz, S.H. and A.G. Phadke, *Power system relaying*. 2nd ed. Electronic & electrical engineering research studies. Lines and cables for power transmission series; 7. 1995, Taunton, Somerset, England, New York: Research Studies Press; Wiley. xiv, 319.
16. Cease, T.W. and B. Feldhaus, *Real-time monitoring of the TVA power system*. Computer Applications in Power, IEEE, 1994. **7**(3): p. 47.
17. Larsson, M. and C. Rehtanz. *Predictive frequency stability control based on wide-area phasor measurements*. 2002.
18. Baldwin, T.L., et al., *Power system observability with minimal phasor measurement placement*. Power Systems, IEEE Transactions on, 1993. **8**(2): p. 707.
19. Abdel-Rahman, K., et al. *Internet based wide area information sharing and its roles in power system state estimation*. 2001.
20. Abur, A. and A. Gómez Expósito, *Power system state estimation: theory and implementation*. Power engineering. 2004, New York, NY: Marcel Dekker. xiv, 327 p.
21. Burnett, R.O., Jr., et al., *Synchronized phasor measurements of a power system event*. Power Systems, IEEE Transactions on, 1994. **9**(3): p. 1643.
22. www.Genscape.com. 2004-2005, Genscape, Inc.
23. Zhong, Z. and Y. Liu, *Estimation of Event Location and Tripped Generation of Generator Tripping Fault Based on FNET Frequency Information*, in *Final Report Submitted to ABB/TVA*. 2004, Virginia Tech: Blacksburg.
24. <http://www.census.gov/cgi-bin/gazetteer>. 2005, US Census Bureau.
25. Lyon, A.J., *Dealing with data*. 1st ed. 1970, Oxford: Pergamon Press. xvii, 392.
26. Dugan, R.C. and R.C. Dugan, *Electrical power systems quality*. 2nd ed. 2003, New York: McGraw-Hill. xv, 528 p.
27. Trudnowski, D.J. and J.E. Dagle, *Effects of generator and static-load nonlinearities on electromechanical oscillations*. Power Systems, IEEE Transactions on, 1997. **12**(3): p. 1283.
28. NERC, *Understand and Calculate Frequency Response*. 2003, NERC Training Resources Working Group.
29. YongJune, S., et al. *Time-frequency analysis of power system disturbance signals for power quality*. 1999.
30. Staudte, R.G. and S.J. Sheather, *Robust estimation and testing*. Wiley series in probability and mathematical statistics. Applied probability and statistics. 1990, New York: Wiley. xix, 351 p.
31. Rousseeuw, P.J. and A.M. Leroy, *Robust regression and outlier detection*. Wiley series in probability and mathematical statistics. Applied probability and statistics. 1987, New York: Wiley. xiv, 329 p.

32. Hoaglin, D.C., F. Mosteller, and J.W. Tukey, *Understanding robust and exploratory data analysis*. Wiley classics library ed. 2000, New York: Wiley. xx, 447 p.
33. Huber, P.J., *Robust statistics*. Wiley series in probability and statistics. 2004, Hoboken, N.J.: Wiley-Interscience. ix, 308 p.
34. Weisstein, E.W. "Moving Average". [cited; Available from: <http://mathworld.wolfram.com/MovingAverage.html>.
35. Mili, L., *Robust Estimation and Filtering (ECE 5714)*. 2005, Virginia Tech.
36. Milton, J.S. and J.C. Arnold, *Introduction to probability and statistics: principles and applications for engineering and the computing sciences*. 3rd ed. Schaum's solved problems series. 1995, New York: McGraw-Hill. xx, 811.
37. Helstrom, C.W., *Probability and stochastic processes for engineers*. 2nd ed. 1991, New York Toronto Canada: Macmillan; Collier Macmillan Publishers. xi, 610.
38. Bendat, J.S. and A.G. Piersol, *Random data: analysis and measurement procedures*. 2nd ed. 1986, New York: Wiley. xvii, 566.
39. Walpole, R.E. and R.H. Myers, *Probability and statistics for engineers and scientists*. 5th ed. 1993, New York Toronto New York: Macmillan; Maxwell Macmillan Canada; Maxwell Macmillan International. xiv, 766.
40. Hampel, F.R., *Robust statistics: the approach based on influence functions*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. 1986, New York: Wiley. xxi, 502 p.
41. Daniel, C., F.S. Wood, and J.W. Gorman, *Fitting equations to data: computer analysis of multifactor data*. 2d ed. 1980, New York: Wiley. xviii, 458.
42. Lancaster, P. and K. Salkauskas, *Curve and surface fitting: an introduction*. Computational mathematics and applications. 1986, London; Orlando: Academic Press. 280.
43. Arlinghaus, S.L., *Practical handbook of curve fitting*. 1994, Boca Raton, Fla: CRC Press. 249.
44. Franklin, J.N., *Matrix theory*. 1968, Englewood Cliffs, N.J.: Prentice-Hall. xii, 292.
45. Menke, W., *Geophysical data analysis: discrete inverse theory*. 1984, Orlando, Fla.: Academic Press. xii, 260.
46. Menke, W., *Geophysical data analysis: discrete inverse theory*. Rev. ed. International geophysics series; v. 45. 1989, San Diego: Academic Press. xii, 285.

47. Meju, M.A., *Geophysical data analysis: understanding inverse problem theory and practice*. Course notes series; v. 6. 1994, Tulsa, OK: Society of Exploration Geophysicists. iv, 296 p.
48. Chapman, M.C., *Director of Virginia Tech Seismological Observatory*. 2005: Blacksburg, Virginia. p. Personal communication regarding seismological triangulation.
49. Bullen, K.E. and B.A. Bolt, *An introduction to the theory of seismology*. 4th ed. 1985, Cambridge [Cambridgeshire]; New York: Cambridge University Press. xvii, 499 p.
50. Gibowicz, S.a.J. and A. Kijko, *An introduction to mining seismology*. International geophysics series; v. 55. 1994, San Diego: Academic Press. x, 399.
51. *International Seismological Centre* (<http://www.isc.ac.uk/>). 2005.
52. *United States Geological Survey* (<http://www.usgs.gov>). 2005.
53. *Recent Earthquakes in the Intermountain West* (<http://www.seis.utah.edu/helicorder/heli/utah>). 2005, University of Utah Seismograph Stations.
54. *Seismology and Global Structure*, University of Alberta (<http://www-geo.phys.ualberta.ca/~vkrav/Geoph110/seisoutline.htm>).
55. *MATLAB 7*. 2004, The Mathworks, Inc.
56. Kundur, P., N.J. Balu, and M.G. Lauby, *Power system stability and control*. 1994, New York: McGraw-Hill. xxiii, 1176 p.
57. Glover, J.D. and M.S. Sarma, *Power system analysis and design*. 2nd ed. 1994, Boston: PWS Pub. xix, 583 p.
58. Wood, A.J. and B.F. Wollenberg, *Power generation, operation, and control*. 2nd ed. 1996, New York: J. Wiley & Sons. xv, 569.
59. Phadke, A.G. and J.S. Thorp, *Computer relaying for power systems*. Electronic & electrical engineering research studies. Lines and cables for power transmission series; 3. 1988, Taunton, Somerset, England New York: Research Studies Press;Wiley. x, 289 p.
60. Hebert, M. *Active and passive range sensing for robotics*. 2000.
61. Jeffers, R., B. Breed, and B. Gallemore. *Passive range estimation and range rate detection*. 2000.
62. Pieper, R., A. Cooper, and G. Pelegris. *Dual baseline triangulation*. 1995.
63. Gregoire, D.G. and G.B. Singletary. *Advanced ESM AOA and location techniques*. 1989.

Appendix I

MATLAB Scripts for Data Conditioning and Curve Fitting

The following MATLAB 7 (R14SP1) code is for the data conditioning GUI developed for FNET:

```
function varargout = CurveGUI(varargin)
warning('off','MATLAB:nearlySingularMatrix');
% CURVEGUI M-file for CurveGUI.fig
%   CURVEGUI, by itself, creates a new CURVEGUI or raises the
existing
%   singleton*.
%
%   H = CURVEGUI returns the handle to a new CURVEGUI or the
handle to
%   the existing singleton*.
%
%   CURVEGUI('CALLBACK',hObject,eventData,handles,...) calls
the local
%   function named CALLBACK in CURVEGUI.M with the given input
arguments.
%
%   CURVEGUI('Property','Value',...) creates a new CURVEGUI or
raises the
%   existing singleton*. Starting from the left, property
value pairs are
%   applied to the GUI before CurveGUI_OpeningFunction gets
called. An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to CurveGUI_OpeningFcn via
varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI
allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.
```

```

% Edit the above text to modify the response to help CurveGUI

% Last Modified by GUIDE v2.5 30-Jun-2005 14:18:01

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @CurveGUI_OpeningFcn, ...
                  'gui_OutputFcn',    @CurveGUI_OutputFcn, ...
                  'gui_LayoutFcn',    [], ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before CurveGUI is made visible.
function CurveGUI_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
% varargin     command line arguments to CurveGUI (see VARARGIN)


% Choose default command line output for CurveGUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes CurveGUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);


disp('Loading requested FDR data.  Please Wait...');
disp('NOTE: This GUI is primarily designed to examine data on
short intervals (less than 10 minutes).');
global FDR_STRUCT
FDR_STRUCT = load(varargin{1}, 'FDR*');

```

```

pack
disp('Files loaded. Check for error messages or warnings.');
```

```

set([handles.uipanel2 handles.uipanel3 handles.uipanel4
handles.uipanel9],...
    'Visible','off');
```

```

set([handles.checkbox1 handles.checkbox2 handles.checkbox3...
handles.checkbox4 handles.checkbox5
handles.checkbox6], 'Value',1);
set(hObject, 'ToolBar', 'figure');
```

```

set(hObject, 'Name', 'FNET Frequency Data Curve Fitting Utility');
```

```

global rawhds fithds fithdism dispcurve outhds linhds texhds
medhds avghds
outhds = [];
rawhds = [];
fithds = [];
fithdism = [];
linhds = [];
texhds = [];
medhds = [];
avghds = [];
dispcurve = [];
```

```

% --- Outputs from this function are returned to the command
line.
```

```

function varargout = CurveGUI_OutputFcn(hObject, eventdata,
handles)
% varargout cell array for returning output args (see
VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```

% Get default command line output from handles structure
varargout{1} = handles.output;
```

```

% --- Executes on selection change in popupmenu1.
```

```

function popupmenu1_Callback(hObject, eventdata, handles)
% hObject handle to popupmenu1 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```

% Hints: contents = get(hObject,'String') returns popupmenu1
contents as cell array
% contents{get(hObject,'Value')} returns selected item
from popupmenu1
```

```

% --- Executes during object creation, after setting all
properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu2
contents as cell array
%         contents{get(hObject,'Value')} returns selected item
from popupmenu2

% --- Executes during object creation, after setting all
properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject      handle to popupmenu3 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu3
contents as cell array
%          contents{get(hObject,'Value')} returns selected item
from popupmenu3

% --- Executes during object creation, after setting all
properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject      handle to popupmenu3 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu5.
function popupmenu5_Callback(hObject, eventdata, handles)
% hObject      handle to popupmenu5 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu5
contents as cell array
%          contents{get(hObject,'Value')} returns selected item
from popupmenu5

```

```

% --- Executes during object creation, after setting all
properties.
function popupmenu5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu6.
function popupmenu6_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu6
contents as cell array
%         contents{get(hObject,'Value')} returns selected item
from popupmenu6

% --- Executes during object creation, after setting all
properties.
function popupmenu6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on selection change in popupmenu7.
function popupmenu7_Callback(hObject, eventdata, handles)
% hObject      handle to popupmenu7 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu7
contents as cell array
%           contents{get(hObject,'Value')} returns selected item
from popupmenu7

% --- Executes during object creation, after setting all
properties.
function popupmenu7_CreateFcn(hObject, eventdata, handles)
% hObject      handle to popupmenu7 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu8.
function popupmenu8_Callback(hObject, eventdata, handles)
% hObject      handle to popupmenu8 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu8
contents as cell array
%           contents{get(hObject,'Value')} returns selected item
from popupmenu8

% --- Executes during object creation, after setting all
properties.
function popupmenu8_CreateFcn(hObject, eventdata, handles)
% hObject      handle to popupmenu8 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB

```

```

% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu9.
function popupmenu9_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu9 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu9
contents as cell array
%         contents{get(hObject,'Value')} returns selected item
from popupmenu9

% --- Executes during object creation, after setting all
properties.
function popupmenu9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu9 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
set([handles.popupmenu1 handles.popupmenu2 handles.popupmenu3...
handles.popupmenu5 handles.popupmenu6 handles.popupmenu7...

```



```

handles.popupmenu8 handles.popupmenu9
handles.pushbutton1],...
    'Enable','Off');
set([handles.uipanel2 handles.uipanel3 handles.uipanel4
handles.uipanel9],...
    'Visible','on');
set(handles.popupmenu13,'Value',9)
global Time
str1 = get(handles.popupmenu1,'String');
idx1 = get(handles.popupmenu1,'Value');
Time.start.hour = str2double(str1{idx1});
str2 = get(handles.popupmenu3,'String');
idx2 = get(handles.popupmenu3,'Value');
Time.start.min = str2double(str2{idx2});
str3 = get(handles.popupmenu6,'String');
idx3 = get(handles.popupmenu6,'Value');
str4 = get(handles.popupmenu8,'String');
idx4 = get(handles.popupmenu8,'Value');
Time.start.sec = str2double([str3{idx3} '.' str4{idx4}]);
Time.start.str = [str1{idx1} ':' str2{idx2} ':' str3{idx3} '.'
str4{idx4}];
clear str1 idx1 str2 idx2 str3 idx3 str4 idx4
str1 = get(handles.popupmenu2,'String');
idx1 = get(handles.popupmenu2,'Value');
Time.end.hour = str2double(str1{idx1});
str2 = get(handles.popupmenu5,'String');
idx2 = get(handles.popupmenu5,'Value');
Time.end.min = str2double(str2{idx2});
str3 = get(handles.popupmenu7,'String');
idx3 = get(handles.popupmenu7,'Value');
str4 = get(handles.popupmenu9,'String');
idx4 = get(handles.popupmenu9,'Value');
Time.end.sec = str2double([str3{idx3} '.' str4{idx4}]);
Time.end.str = [str1{idx1} ':' str2{idx2} ':' str3{idx3} '.'
str4{idx4}];
global FDR_STRUCT
s = fieldnames(FDR_STRUCT);
set(handles.popupmenu10,'String',s);
clear str1 str2 str3 str4 idx1 idx2 idx3 idx4

% --- Executes on selection change in popupmenu10.
function popupmenu10_Callback(hObject, eventdata, handles)
% hObject      handle to popupmenu10 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu10
contents as cell array

```

```

%         contents{get(hObject,'Value')} returns selected item
from popupmenu10

% --- Executes during object creation, after setting all
properties.
function popupmenu10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu10 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global FDR_STRUCT Time dispcurve ldata
%Get unit number to plot
str = get(handles.popupmenu10,'String');
idx = get(handles.popupmenu10,'Value');
fdrstr = str{idx};
clear str
if isempty(find(dispcurve == idx))
    dispcurve(length(dispcurve)+1)=idx; clear idx
    FDR = getfield(FDR_STRUCT,fdrstr);
    startidx = find(FDR.date(:,4)==Time.start.hour &
FDR.date(:,5)==...
        Time.start.min & FDR.date(:,6)==Time.start.sec);
    if isempty(startidx)
        beep
        disp('The starting time you entered is not recorded in
this unit''s data set. ');
        return
    else
        fprintf('Starting index: %d \n',startidx);
    end
    endidx = find(FDR.date(:,4)==Time.end.hour &
FDR.date(:,5)==...

```

```

        Time.end.min & FDR.date(:,6)==Time.end.sec);
    if isempty(endidx)
        beep
        disp('The ending time you entered is not recorded in this
unit''s data set. ');
        return
    else
        fprintf('Ending index: %d \n',endidx);
    end
    if startidx>endidx
        delete(gcf);
        beep
        display('Program closed because of the following error:
contingency start time larger than contingency end time. ');
        return
    end
    %Get polynomial fit order
    str = get(handles.popupmenu1,'String');
    idx = get(handles.popupmenu1,'Value');
    o = str2double(str{idx});
    %Get outlier rejection window size
    str = get(handles.popupmenu12,'String');
    idx = get(handles.popupmenu12,'Value');
    w = str2double(str{idx});
    %Get outlier rejection threshold
    str = get(handles.popupmenu13,'String');
    idx = get(handles.popupmenu13,'Value');
    th = str2double(str{idx});
    clear str idx
    %Fit and plot data
    set(gcf,'Name',[ 'Data for ' num2str(FDR.date(1,2)) '/' ...
        num2str(FDR.date(1,3)) '/' num2str(FDR.date(1,1))]);
    global rawhds fithdls fithdlsm outhdls medhds avghds
texhds
    color = [rand, rand, rand];
    hold on;
    Y = FDR.freq(startidx:endidx);
    ldata = length(Y);
    rejidx = [];
    q = floor(w/2);
    Ym(1:q) = Y(1:q);
    Ya(1:q) = Y(1:q);
    for i=q+1:ldata-w+q
        Ym(i) = median(Y(i-q:i+q));
        Ya(i) = mean(Y(i-q:i+q));
    end
    Ym(ldata-w+q+1:ldata)=Y(ldata-w+q+1:ldata);
    Ya(ldata-w+q+1:ldata)=Y(ldata-w+q+1:ldata);

```

```

medhdls(length(medhdls)+1)=plot(handles.axes1,1:length(Ym),Ym,...
    'Color',color,'linestyle','--');
if ~get(handles.checkbox4,'Value')
    set(medhdls(:),'Visible','off');
end
setappdata(medhdls(end),'unit',fdrstr);
clear Ym

avghdls(length(avghdls)+1)=plot(handles.axes1,1:length(Ya),Ya,...
    'Color',color,'linestyle','-');
if ~get(handles.checkbox5,'Value')
    set(avghdls(:),'Visible','off');
end
setappdata(avghdls(end),'unit',fdrstr);
clear Ya
for i=1:ldata-w+1
    Yt = Y(i:i+w-1);
    medYt = median(Y);
    bm = w/(w-0.8);
    MAD = 1.4826*bm*median(abs(Yt-medYt));
    r = abs(Yt-medYt)/MAD;
    for ii=1:w
        if r(ii)>th
            outhdls(length(outhdls)+1) =
plot(handles.axes1,...
            ii+i-1,Y(ii+i-1),'r*');
            if ~get(handles.checkbox3,'Value')
                set(outhdls(:),'Visible','off');
            end
            setappdata(outhdls(end),'unit',fdrstr);
            rejidx(length(rejidx)+1)=ii+i-1;
        end
    end
end
Y=Y';
l = length(Y);
lsidx = 1;
for i=1:l
    if isempty(find(rejidx==i))
        for ii=1:(o+1)
            H(lsidx,ii) = i^(o-ii+1);
        end
        lsidx = lsidx + 1;
    end
end
Yraw = Y;
Y(rejidx)=[];
Hinv = (H'*H)^-1*H';

```

```

XLS = Hinv*Y;

rawhdls(length(rawhdls)+1)=plot(handles.axes1,1:length(Yraw),Yraw
,...
    'Color',color,'linestyle',':');
if ~get(handles.checkbox1,'Value')
    set(rawhdls(:),'Visible','off');
end
setappdata(rawhdls(end),'unit',fdrstr);
for i=1:ldata
    y(i)=0;
    for ii=1:(o+1)
        y(i)=y(i)+XLS(ii)*i^(o-ii+1);
    end
end
plotdata(:,1) = 1:length(y);
plotdata(:,2) = y;
derpd = tder(plotdata);
[c,i] = max(abs(derpd(11:end-10,2)));

fithdls(length(fithdls)+1)=plot(handles.axes1,i+10,plotdata(i+1
0,2),'r^');

fithdls(length(fithdls)+1)=plot(handles.axes1,plotdata(:,1),...
    plotdata(:,2),'Color', color,'linestyle','-');
texhdls(length(texhdls)+1)=text((length(texhdls)+5),y(1),...
Time.start.str,'Color',color,'Rotation',90,'VerticalAlignment',...
.
'top','Margin',2,'FontWeight','bold','BackgroundColor','r');
texhdls(length(texhdls)+1)=text(length(y)-length(texhdls),...
    y(length(y)),Time.end.str,'Color',color,'Rotation',90,...
'VerticalAlignment','bottom','Margin',2,'FontWeight','bold',...
    'BackgroundColor','r');
if ~get(handles.checkbox2,'Value')
    set(fithdls(:),'Visible','off');
    set(fithdls(:),'Visible','off');
end
if ~get(handles.checkbox6,'Value')
    set(texhdls(:),'Visible','off');
end
setappdata(fithdls(end),'unit',fdrstr);
axis tight
str = get(handles.popupmenu10,'String');
clear fdrstr
for i=1:length(dispcurve)
    idx = dispcurve(i);

```

```

        fdrstr{i} = str{idx};
    end
    legend(fithdls,fdrstr);
    clear str idx
    clear y Y rejidx ldata XLS lsidx Hinv r Yt medYt color FDR bm
MAD fdrstr
else
    beep
    disp('The data for this unit has already been plotted.');
```

end

```

% --- Executes on button press in checkbox1.
function checkbox1_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global rawhdls
if ~isempty(rawhdls)
    if get(hObject,'Value')
        set(rawhdls,'Visible','On');
    else
        set(rawhdls,'Visible','Off');
    end
end
end

% --- Executes on button press in checkbox2.
function checkbox2_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global fithdls fithdls
if ~isempty(fithdls)
    if get(hObject,'Value')
        set(fithdls,'Visible','On');
        set(fithdls,'Visible','On');
    else
        set(fithdls,'Visible','Off');
        set(fithdls,'Visible','Off');
    end
end
end

% --- Executes on button press in checkbox3.
function checkbox3_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
```

```

% handles      structure with handles and user data (see GUIDATA)
global outhdls
if ~isempty(outhdls)
    if get(hObject,'Value')
        set(outhdls,'Visible','On');
    else
        set(outhdls,'Visible','Off');
    end
end

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject      handle to popupmenu1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu1
contents as cell array
%      contents{get(hObject,'Value')} returns selected item
from popupmenu1

% --- Executes during object creation, after setting all
properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to popupmenu1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu12.
function popupmenu12_Callback(hObject, eventdata, handles)
% hObject      handle to popupmenu12 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu12
contents as cell array

```

```

%         contents{get(hObject,'Value')} returns selected item
from popupmenu12

% --- Executes during object creation, after setting all
properties.
function popupmenu12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu12 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu13.
function popupmenu13_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu13 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu13
contents as cell array
%         contents{get(hObject,'Value')} returns selected item
from popupmenu13

% --- Executes during object creation, after setting all
properties.
function popupmenu13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu13 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```



```

function edit1_Callback(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of
edit1 as a double

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton3 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
global linhdls ldata
try
    val = str2double(get(handles.edit1,'String'));
    if isnumeric(val)

linhdls(length(linhdls)+1)=plot(handles.axes1,1:ldata,ones(1,ldata
a)*val,'r-');
    else
        set(handles.edit1,'String','');
    end
catch
    disp('No reference data.  No line drawn.');
```

```

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton4 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
global dispcurve linhdls rawhdls fithdls fithdls1m outhdls texhdls
medhdls avghdls
set(texhdls,'Visible','off');
delete(texhdls);
set(linhdls,'Visible','off');
delete(linhdls);
set(rawhdls,'Visible','off');
delete(rawhdls);
set(fithdls,'Visible','off');
delete(fithdls);
set(fithdls1m,'Visible','off');
delete(fithdls1m);
set(outhdls,'Visible','off');
delete(outhdls);
set(medhdls,'Visible','off');
delete(medhdls);
set(avghdls,'Visible','off');
delete(avghdls);
legend('off');
outhdls = [];
rawhdls = [];
fithdls = [];
fithdls1m = [];
linhdls = [];
medhdls = [];
avghdls = [];
texhdls = [];
dispcurve = [];

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton5 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
set([handles.uipanel2 handles.uipanel3 handles.uipanel4...
handles.uipanel9], 'Visible', 'off');
set([handles.popupmenu1 handles.popupmenu2 handles.popupmenu3...
handles.popupmenu5 handles.popupmenu6 handles.popupmenu7...
handles.popupmenu8 handles.popupmenu9
handles.pushbutton1], ...
'Enable', 'on');

```

```

% --- Executes on button press in checkbox4.
function checkbox4_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global medhdlg
if ~isempty(medhdlg)
    if get(hObject,'Value')
        set(medhdlg,'Visible','On');
    else
        set(medhdlg,'Visible','Off');
    end
end

% --- Executes on button press in checkbox5.
function checkbox5_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global avghdlg
if ~isempty(avghdlg)
    if get(hObject,'Value')
        set(avghdlg,'Visible','On');
    else
        set(avghdlg,'Visible','Off');
    end
end

% --- Executes on button press in checkbox6.
function checkbox6_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global texhdlg
if ~isempty(texhdlg)
    if get(hObject,'Value')
        set(texhdlg,'Visible','On');
    else
        set(texhdlg,'Visible','Off');
    end
end

% --- Executes on button press in checkbox7.
function checkbox7_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox7 (see GCBO)

```

```
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
if get(hObject,'Value')
    grid on
    grid minor
else
    grid off
end
```

Appendix II

MATLAB Scripts for Event Location

The following MATLAB 7 (R14SP1) code is for the triangulation algorithm developed for FNET:

All.1 Least-Squares Method:

```
% --- Executes on button press in triangulate.
function triangulate_Callback(hObject, eventdata, handles)
% hObject    handle to triangulate (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global FREQ currentbc bcnunits vu testvel TRIHANDLE
TEST_RESULTS
curunit = FREQ.Unit.Unit;
vu.num = [];
vu.lat = [];
vu.lon = [];
vu.del = [];
for idx = 1:length(curunit)
    if ~isempty(curunit(idx).UnitDelay)
        vu.num(length(vu.num)+1) = curunit(idx).UnitNumber;
        vu.lat(length(vu.lat)+1) = curunit(idx).UnitLat;
        vu.lon(length(vu.lon)+1) = curunit(idx).UnitLon;
        vu.del(length(vu.del)+1) = curunit(idx).UnitDelay;
    end
end
[x,y] = grn2eqa([24 50],[-105 -66]);
lb = [min(x); min(y)];
ub = [max(x); max(y)];
testidx = 1;
for vidx = 100:50:1000
    testvel = vidx*FREQ.vel.convm2e;
    for idx = 1:length(vu.num)
        if idx == length(vu.num)
            nidx = 1;
        else
            nidx = idx+1;
        end
        [xi,yi] = grn2eqa(vu.lat(idx),vu.lon(idx));
        [xni,yni] = grn2eqa(vu.lat(nidx),vu.lon(nidx));
    end
end
```

```

        C(idx) = 0.5*((testvel^2)*(vu.del(idx)^2-
vu.del(nidx)^2)+xni^2+yni^2-xi^2-yi^2);
        H(idx,1) = xni-xi;
        H(idx,2) = yni-yi;
        H(idx,3) = -testvel^2*(vu.del(nidx)-vu.del(idx));
    end
    XLS = (H'*H)^-1*H'*C';
    [lat,lon] = eqa2grn(XLS(1),XLS(2));
    TRIHANDLE(length(TRIHANDLE)+1) = plotm(lat,lon,'r*');
    drawnow
    TEST_RESULTS(testidx,1) = vidx;
    TEST_RESULTS(testidx,2) = lat;
    TEST_RESULTS(testidx,3) = lon;
    testidx = testidx+1;
end

```

All.2 Newton Method:

```

% --- Executes on button press in triangulate.
function triangulate_Callback(hObject, eventdata, handles)
% hObject      handle to triangulate (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
global FREQ vu TRIHANDLE MASTER testvel TEST_RESULTS
set(hObject, 'String','Please Wait...','ForegroundColor','r');
set([handles.basecase handles.newunit],'Enable','off');
curunit = FREQ.Unit.Unit;
vu.num = [];
vu.lat = [];
vu.lon = [];
vu.del = [];
for idx = 1:length(curunit)
    if ~isempty(curunit(idx).UnitDelay)
        vu.num(length(vu.num)+1) = curunit(idx).UnitNumber;
        vu.lat(length(vu.lat)+1) = curunit(idx).UnitLat;
        vu.lon(length(vu.lon)+1) = curunit(idx).UnitLon;
        vu.del(length(vu.del)+1) = curunit(idx).UnitDelay;
    end
end
[x,y] = grn2eqa([24 50],[-105 -66]);
testidx = 1;
for vidx = 100:50:1000
    testvel = vidx*FREQ.vel.convm2e;
    x0 = [min([FREQ.Unit.Unit(:).UnitDelay]); mean(x(:));
mean(y(:))];

```

```

options=optimset('Display','iter');           % Option to display
output
X = fsolve(@trifun,x0,options);               % Call optimizer
[lat,lon] = eqa2grn(X(2),X(3));
if (lat<50)&(lat>24)&(lon<-66)&(lon>-105)
    TRIHANDLE(length(TRIHANDLE)+1) =
plotm(lat,lon,'gv','linewidth',2);
    drawnow
end
TEST_RESULTS(testidx,1) = vidx;
TEST_RESULTS(testidx,2) = lat;
TEST_RESULTS(testidx,3) = lon;
testidx = testidx+1;
end
set(hObject, 'String','Triangulate','ForegroundColor','g');
%Function to construct target equation = sum(ri^2)
function trif = trifun(X)
global FREQ vu testvel
for idx = 1:length(vu.num)
    [x(idx),y(idx)] = grn2eqa(vu.lat(idx),vu.lon(idx));
    trif(idx) = (x(idx)-X(2))^2+(y(idx)-X(3))^2-
testvel^2*(vu.del(idx)-X(1))^2;
end
trif = trif';

```

All.3 Gradient Search Method:

```

% --- Executes on button press in triangulate.
function triangulate_Callback(hObject, eventdata, handles)
% hObject    handle to triangulate (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global FREQ vu TRIHANDLE MASTER testvel TEST_RESULTS
set(hObject, 'String','Please Wait...','ForegroundColor','r');
set([handles.basecase handles.newunit ],'Enable','off');
curunit = FREQ.Unit.Unit;
vu.num = [];
vu.lat = [];
vu.lon = [];
vu.del = [];
for idx = 1:length(curunit)
    if ~isempty(curunit(idx).UnitDelay)
        vu.num(length(vu.num)+1) = curunit(idx).UnitNumber;
        vu.lat(length(vu.lat)+1) = curunit(idx).UnitLat;
        vu.lon(length(vu.lon)+1) = curunit(idx).UnitLon;
    end
end

```

```

        vu.del(length(vu.del)+1) = curunit(idx).UnitDelay;
    end
end
[x,y] = grn2eqa([24 50],[-105 -66]);
lb = [min(x); min(y)];
ub = [max(x); max(y)];
x0 = [mean(x(:)); mean(y(:))];
[b,IX] = sort(vu.del);
vulen = length(vu.num);
testidx = 1;
for n = 3:vulen
    outtemp = vu;
    vu.del = vu.del(IX(1:n));
    vu.num = vu.num(IX(1:n));
    vu.lat = vu.lat(IX(1:n));
    vu.lon = vu.lon(IX(1:n));
    X = fmincon(@distfun,x0,[],[],[],[],lb,ub,@dnonlcon);
    [lat,lon] = eqa2grn(X(1),X(2));
    TRIHANDLE(length(TRIHANDLE)+1) =
plotm(lat,lon,'gd','linewidth',2);
    drawnow;
    TEST_RESULTS(testidx,1) = n;
    TEST_RESULTS(testidx,2) = lat;
    TEST_RESULTS(testidx,3) = lon;
    testidx = testidx+1;
    vu = outtemp;
end
drawnow;
set(hObject, 'String', 'Triangulate', 'ForegroundColor', 'g');
set([handles.basecase handles.newunit ], 'Enable', 'on');

%Minimization Function
function dfun = distfun(X)
global vu
dfun = 0;
for idx = 1:length(vu.num)
    [x,y] = grn2eqa(vu.lat(idx),vu.lon(idx));
    dfun = dfun + (x-X(1))^2 + (y-X(2))^2;
end
%Non-linear constraint function
function [c,ceq] = dnonlcon(X)
global vu
for idx = 1:length(vu.num)-1
    [x1,y1] = grn2eqa(vu.lat(idx),vu.lon(idx));
    [x2,y2] = grn2eqa(vu.lat(idx+1),vu.lon(idx+1));
    c(idx) = (x1-X(1))^2+(y1-X(2))^2-(x2-X(1))^2-(y2-X(2))^2;
end
ceq = [];

```