Frequency Monitoring Network (FNET)
Algorithm Improvements and Application Development

Tao Xia

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

Doctor of Philosophy
in
Electrical Engineering

Committee Members:

Yilu Liu (Chair)

Virgilio A. Centeno

Jaime De La Ree

Richard W. Conners

Irene E. Leech

December 7, 2009
Blacksburg, Virginia

Keywords:
FNET, Event Location Estimation, Phase Angle Algorithm, Angle-based
Oscillation Detection, Modal Analysis

Frequency Monitoring Network (FNET)
Algorithm Improvements and Application Development

Tao Xia

(Abstract)

The Internet Based real-time GPS synchronized wide-area Frequency Monitoring Network (FNET) is an extremely low cost and quickly deployable wide-area frequency measurement system with high dynamic accuracy which consists of Frequency Disturbance Recorder (FDR) distributed to more than 100 places around North America and an Information Management System situated at Virginia Tech. Since its first FDR deployment in 2003, the FNET system has been proved to be able to reliably receive phasor data accurately measured at and instantaneously sent via the Internet from different locations of interest, and efficiently run the analyzing program to detect and record significant system disturbances and subsequently estimate the location of disturbance center, namely the event location, in the electric grid based on the information gathered. The excellent performance of the FNET system so far has made power grid situation awareness and monitoring based on distribution level frequency measurements a reality, and thus advances our understanding of power system dynamics to a higher level and in a broader dimensionality.

Chapter1 and Chapter 2 of this dissertation briefly introduce the genesis and the architecture of the FNET system, followed by a summary of its concrete implementations. Chapter 3 and Chapter 4 outline FNET frequency estimation algorithm and phase angle estimation algorithm, including their attributes and the new methodologies to enhance them. In Chapter 5, the report discusses the algorithms developed at FNET to detect the frequency disturbance and estimate the disturbance location by the triangulation procedure using real-time frequency data and geographic topology of the FNET units in the power grid where the disturbance occurs. Then, the dissertation proceeds to introduce the FNET angle-based power system oscillation detection and present some research about Matrix Pencil Modal Analysis of FNET phase angle oscillation data in the following two chapters. Lastly, the content of this report is summarized and the future work envisioned in Chapter 8.

Acknowledgments

I would like to express my deepest gratitude to my advisor, Dr. Yilu Liu for her patient guidance and constant encouragement throughout this study. Her expertise and financial support was indispensable for the completion of this dissertation. I would also like to thank Dr. Virgilio A. Centeno, Dr. Jaime De La Ree, Dr. Richard W. Conners, and Dr. Irene E. Leech for serving on my committee. I thank them for their interest and concern. I am greatly appreciative of the help given by the graduate students and visiting scholars in FNET and the Power Systems Laboratory. Their friendship has allowed me to successfully accomplish my academic pursuit. My Ph.D studies at Virginia Tech have provided some of my most precious memories, and I will cherish them for the rest of my life.

To many of the sleepless nights in Blacksburg

Table of Contents

List of Figures

List of Tables

Chapter 1: FNET System Architecture

1.1 Background and Motivation

Power systems have been becoming more and more complex, as a result of a considerable variety of new components being added, such as HVDC and power electronic devices, and numerous recent technologies continually being put into application, such as distributed generation. These characteristics of the evolutional trend of power systems has ushered in the era of bulk power system, in the process of which the status of a power system becomes more and more difficult to measure, estimate and comprehend because of the ever-increasing complexity of the new bulk power systems.

Understanding the complexities of nowadays interconnected power grids and the need for proper planning, comprehensive maintenance, and sound operating practices is the key to maintain system reliability, preventing potential hazards due to contingencies, and meet the challenge of the colossal energy demand for power in the new century. Advanced analyses, operational tools and the application of latest technologies, especially the situational awareness of the power system, are imperative to tackle phenomena (stability detection, reactive power management, load encroachment, etc.) that persist to affect grid reliability and threaten the normal operation of power systems.

Meanwhile, drastic changes in the electric industry have taken place in the last twenty years, mainly because of the deregulation of electricity markets. Under the deregulated market condition, the transmission utilities find it much more difficult to upgrade the transmission systems to keep up with the growing loads and the increasing generation than in the past. The customers try to buy cheaper electricity which might be generated in a remote area, and make the transmission distance longer and transmission network more complex. Many tie lines between control areas are driven to operate near their maximum capacity, and the entire transmission networks are very often exploited at their capacity limits. The super-grid has also become so venerable that the events in one area can cascade and have significant impact on the remote areas. Those stressed operating conditions can increase the inter-area oscillation between different control areas, endanger the power system steady operation, and even break up the system.

All these changes are putting pressure for the development of new tools to monitor and analyze system operation, and take measures to realize transient stability control if necessary. By making greater use of new types of sensors, communication systems, and distributed computer networks, modern power systems are expanding the scope of power system engineering and thereby starting to meet the challenges posed by the recent full-scale changes and the deregulation trend of the power systems.

Among all the newly-developed applications that aim at satisfying those new technological demands, Wide area measurement system (WAMS) opens a new avenue to power system stability analysis and control, and it has been increasingly attracting attention in recent years, since it is a powerful tool for power system monitoring, protection and control, and has been widely used in the energy management systems (EMS) of power systems. The snapshot of the system dynamic behavior can be obtained and updated on the order of milliseconds by WAMS, and the rapid development of WAMS in recent years also facilitates the simulation validation work because WAMS can record the synchronized phasors during the dynamics.

PMUs, a research effort that has been in the spotlight in the past 20 years as a key technology of WAMS, are used to measure the power systems dynamic data synchronized by GPS satellites, such as voltages, current, angle and frequency, which can provide real-time information about the state of large portions of national and transnational grids and allow obtaining a state estimate which is a true snapshot of the power system, making the technology for real-time supervision of power systems clearly at hand.

Along with the innovational and even revolutionary effects it brings to power system monitoring and control applications, the PMU development has certain disadvantages manifested in the exertions put forth previously to apply the technology to practice, such as its relatively high cost and inconvenience of installation, which to some extent hinders it to be more quickly and widely utilized in power systems. Aiming at accounting for those problems, Frequency Monitoring Network (FNET) of Virginia Tech has come up with an extremely low cost and quickly deployable wide-area frequency measurement system with high dynamic accuracy. Nearly 100 sensors have been deployed around the North America power grids so far. Algorithms and applications have been explored, proposed, examined, tested, compared, and put online in FNET, which makes possible and available the wide-area situational awareness and disturbance analysis to power system researchers, operators, supervisors, customers, and policy-makers. The technology is not presented as a replacement for existing technologies such as SCADA and PMU technology, but rather, as a support thereunto.

1.2 FNET Overall Structure

System frequency is one of the most important parameters of the power system. Area frequency can vary over a small range due to generation-load mismatches. Many power system protection and control applications require accurate and fast estimation of area system frequency. Frequency information has also been used routinely in the control and management of generations. Fast and accurate real time measurement of frequency is an important component in the operation of the transmission system. Moreover, if the rate of change of frequency is known, more accurate adjustments can be made to restore the system back to its normal operation point quickly[1].

Over the years, many advances had been made in the areas of accurate and efficient frequency information retrieval, and many measurement devices had been researched and developed by power system engineers, most of which, however, had assumed a single system frequency and accordingly used long periods of data averaging in order to achieve good estimation accuracy until the PMU came into its very being. In the early 1990s the synchronized phasor measurement unit (PMU) was developed and later commercialized. Since then, a number of applications have been proposed that require wide-area measurement systems. The results of all earlier efforts clearly point to the need for much wider measurement coverage, coverage that can be quickly and economically obtained[2].

The objective of the frequency monitoring network (FNET) effort is to create an extremely low cost and quickly deployable wide-area frequency measurement system with high dynamic accuracy and that requires a minimal installation cost. All these features are possible in FNET due to the fact that the power system frequency can be accurately measured and global positioning system (GPS) synchronized at the 110-V distribution voltage level of a typical office outlet[3][4].

The FNET system is a wide-area sensor network consisting of high-precision frequency disturbance recorders and a central processing server. Figure 1.1 shows the framework of the FNET system.

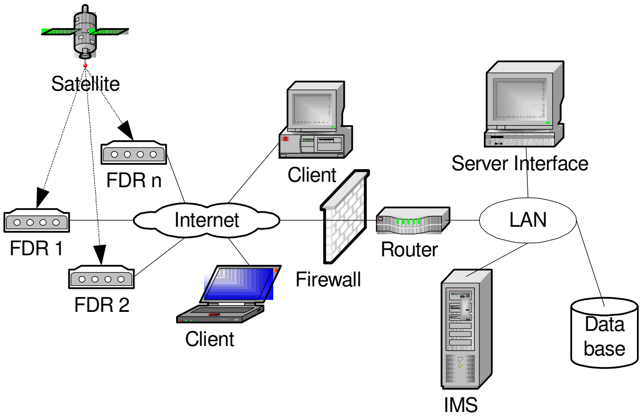


Fig. 1.1 Frequency Monitoring Network System Architecture

The FNET system consists of two major components: a) Frequency Disturbance Recorders (FDRs), which perform local GPS synchronized frequency measurements and send data to a server through the Internet; b) The Information Management System (IMS) which includes data collection and storage service, data communication service, database operation service, and web service. The FDRs sample the voltage at various locations; calculate the frequency, angle, and magnitudes of the voltage; timestamp the results using GPS-based synchronized timing; and send back the data to the IMS through the Internet. The IMS receives, classifies, and manages the data. The disturbance detecting trigger is embedded in the IMS and based on the data stored in the database. Furthermore, the IMS estimates the event location and then produces the corresponding event map after detecting a disturbance. The result is then uploaded to the website for easy reference.

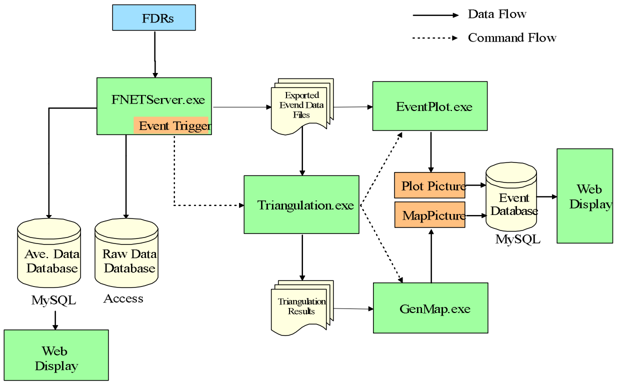


Fig. 1.2 FNET Software Framework

The entire FNET program consists of four individual modules including the FNET main server (FNETServer.exe), the event plotting module (EventPlot.exe), the triangulation module (Triangulation.exe), and the event map generation module (GenMap.exe), shown in

Figure 1.2.

The FNET main server acts as the IMS that collects the frequency data from the Internet, saves the data to both a Microsoft® Access database and a MySQL® database, and checks for events. An on-line event trigger is also implemented on the server. When an event is detected on-line, server exports event data to hard disk files; records the event date, time, and zone information into EventTime.txt; and then calls the triangulation module.

The triangulation module, which is also referred to as the event location estimation module, reads in data from exported data files, performs data conditioning, and estimates the event location. After the event location is estimated, the module writes the triangulation results to a text file and then calls both the event plotting module and the event map generation module. The event plotting module plots the exported event frequency data and saves it to a file for web display. The event map generation module displays the event's location coordinates on a U.S. map along with information regarding the event type and estimated trip amount.

Chapter 2: FNET Technical Details

2.1 FNET Implementation

2.1.1 Frequency Disturbance Recorder

An FDR unit consists of a voltage transducer, a low pass filter, an analog to digital (A/D) converter, a GPS receiver, a microprocessor, and network communication modules, as shown in Figure 2.1. The voltage transducer takes an analog voltage signal from an 110V wall outlet and converts it to acceptable A/D levels, the low pass filter eliminates the high frequency components, and the A/D converter transforms the analog signal into digital data. A microprocessor is used to generate the sampling pulses synchronized to the PPS from the GPS receiver integrated into the FDR. The phase angle, frequency, and rate of change of frequency are computed, using phasor techniques developed specifically for single phase measurements. The computed values are time stamped, and transferred to the IMS via the Internet.

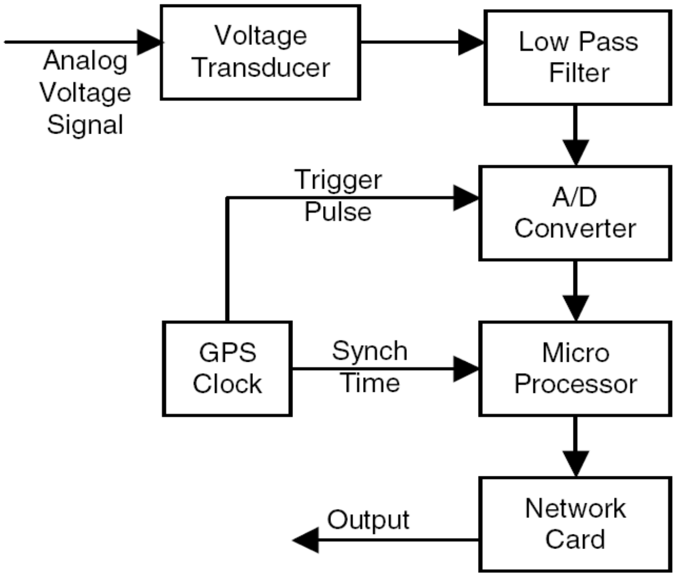


Fig. 2.1 FDR Hardware Structure

There are two generations of FDRs, which are shown in Figure 2.2. The right one is generation I and the left one generation II.

Static frequency computations are usually done on voltage waveforms due to their relatively unchanging nature during normal operating conditions. The FDR unit makes frequency calculations using algorithms of phasor analysis and signal resampling techniques. The complexity of the calculations is minimized to allow the microprocessor time to complete its other tasks and prevent data overflow. The current version of FDRs has a sampling rate of 1440Hz and the resulting frequency accuracy is ±0.0005Hz or better.

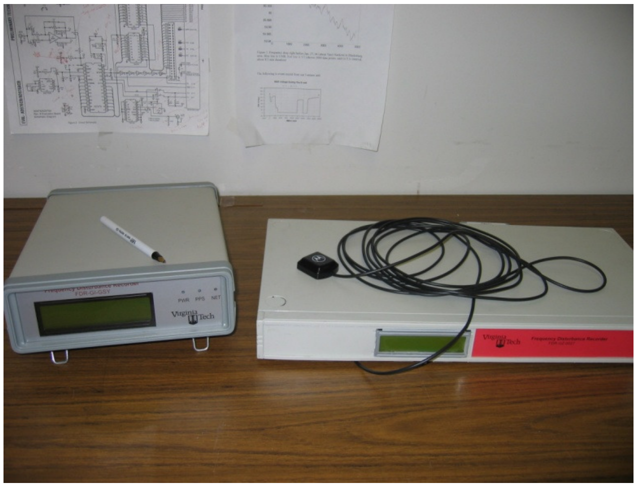


Fig. 2.2 Frequency Disturbance Recorders

Time synchronization of measured data is an important aspect of FNET. A GPS clock integrated into the FDR device provides phase and frequency-locked sample pulses. In conjunction with the front end topology, the GPS clock permits precise comparison of sample data taken at any point in the system. Data sampling is synchronized to a time reference. In the hardware design, each FDR has a GPS clock to generate the sampling pulses that drive the A/D converter to ensure the synchronized phasor data acquisition. It also supports the accurate time for the phasor data output.

The microprocessor of the FDR is the core of the device, which is responsible for the required computation of the phasor measurements of the local area of installation. It also coordinates different parts and elements of the sensor to help perform the pre-designed functions, such as parsing the GPS time stamp information transmitted from the GPS receiver module, encapsulating the measured electric data and the time stamp into an IP packet, and communicating the IP packets to the Internet interface module to be sent to the server. Two different types of microprocessors, MPC555 and TI TMS320F2407, are utilized in generation I FDR and generation II FDR respectively.

In order for the code to be easily developed and maintained, the C language is primarily employed in the FNET system. Programmers do not need to understand the architecture of the microprocessor being used, and the code possesses a very satisfactory portability, which enables us to transplant nearly the entirety of the firmware of generation I FDR to a different microprocessor of generation II FDR with little effort.

The two versions of the firmwares are fairly similar to each other, whose basic software frames are alike, with only details of implementation being different to account for the specific peripherals of the main microprocessors and particular modules of the two devices. The remainder of this subsection uses generation II as an example to illustrate the software structure as well as the functionalities and coordination of different modules.

In addition to the primary microprocessor TI TMS320F2407, generation II FDR has C8051 as its auxiliary co-microprocessor, helping to execute certain ancillary functions. In a nutshell, the TMS320F2407 chiefly deals with executing A/D conversions upon sampling pulses, performing frequency, voltage magnitude, and voltage angle calculations based on new sample value, and sending the results up to C8051; while, the C8051 basically receives the measurement results from TMS320F2407, analyze the synchronous clock information from the GPS receiver and translate it into time stamps, wraps the measurements and time stamps, and then send the IP packets to the Internet module to be sent out. The specifics of the code for both TMS320F2407 and C8051 are presented in the following 2 diagrams.

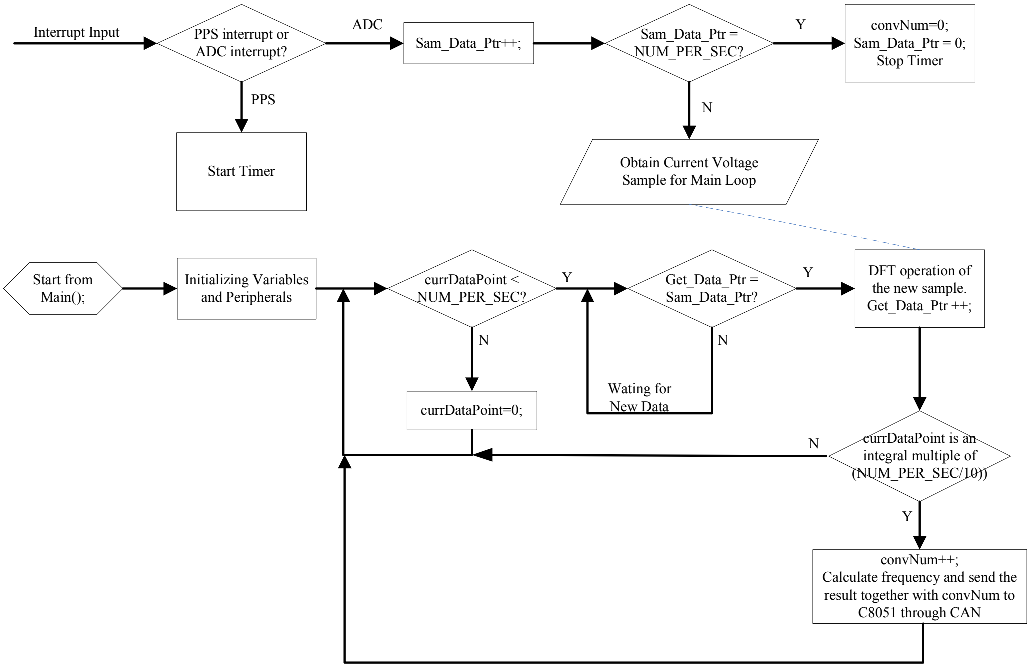

Interrupt Input
PPS interrupt or ADC interrupt?
ADC
Sam_Data_Ptr++;
Sam_Data_Ptr = NUM_PER_SEC?
Y
convNum=0; Sam_Data_Ptr = 0; Stop Timer
PPS
Start Timer
N
Obtain Current Voltage Sample for Main Loop
Start from Main();
Initializing Variables and Peripherals
currDataPoint < NUM_PER_SEC?
Y
Get_Data_Ptr = Sam_Data_Ptr?
Y
DFT operation of the new sample. Get_Data_Ptr ++;
N
N
currDataPoint=0;
Wating for New Data
N
currDataPoint is an integral multiple of (NUM_PER_SEC/10))
Y
convNum++; Calculate frequency and send the result together with convNum to C8051 through CAN

Fig. 2.3 DSP Program Flowchart

Figure 2.3 is the DSP program flowchart. There are two major parts of the code, the interrupt service routine (ISR) and the main() function respectively. The DSP utilizes two interrupts to respond to the PPS and ADC inputs. Once there is an interrupt request, either by PPS or by ADC, the DSP pushes the current state of execution to the stack or other form of direct fast data storage, and then enters the ISR, in the beginning of which certain bit fields signify which of the two types of interrupts the request is. After the interrupt attribute is determined, the ISR goes to different braches accordingly. If the request is an ADC, the ISR first increments the sample sequence counter Sam_Data_Ptr by 1, followed by judging whether or not the variable is over NUM_PER_SEC, the total sampling points in one second or the sampling rate. The DSP then resets counter variables, such as convNum and Sam_Data_Ptr, if Sam_Data_Ptr equals to the limit, which indicates the sampling point will cross the current second boundary of GPS time at the next sampling point. The timer to trigger the A/D conversion is also stopped for sake of the impending boundary-crossing. Otherwise, the ISR reads in the voltage sample from the A/D converter and make it ready for the main program to process. If a PPS interrupt is detected, the timer for A/D conversion is started as the means of synchronization with the GPS time.

The main() loop of the DSP program, which runs continuously before being shut down, starts with initializing the variables and peripherals. The ever-running feature of the main() function is caused by the loop that compare Get_Data_Ptr with Sam_Data_Ptr. As can be seen from the flowchart, the program repeats an infinite loop if Get_Data_Ptr is equal to Sam_Data_Ptr, and the loop is only exited until the Sam_Data_Ptr is modified by A/D conversion ISR. A series of operations are carried out once the loop stops, including DFT calculation at every sampling point, computing frequency and sending phasor data together with tenth-of-a-second counter convNum to C8051 through CAN when currDataPoint is an integral multiple of (NUM_PER_SEC/10). The program goes back to the infinite loop that waits for the new data after the operations are finishes, preceded by resetting currDataPoint if it is not less than NUM_PER_SEC. Counter currDataPoint is used to coordinate the pace of the computation in main() function and the pace of the A/D conversion.

Figure 2.4 is the C8051 program flowchart. Similarly, there are also two major parts of the code, the interrupt service routines (ISRs) and the main() function. The difference of the way C8051 firmware manages interrupts from the way DSP does is the fact that there are three different interrupts serving three different types of interrupt requests (IRs), each of which has its dedicated interrupt entry address. The ISRs here are CAN interrupt, PPS interrupt and Serial Port interrupt.

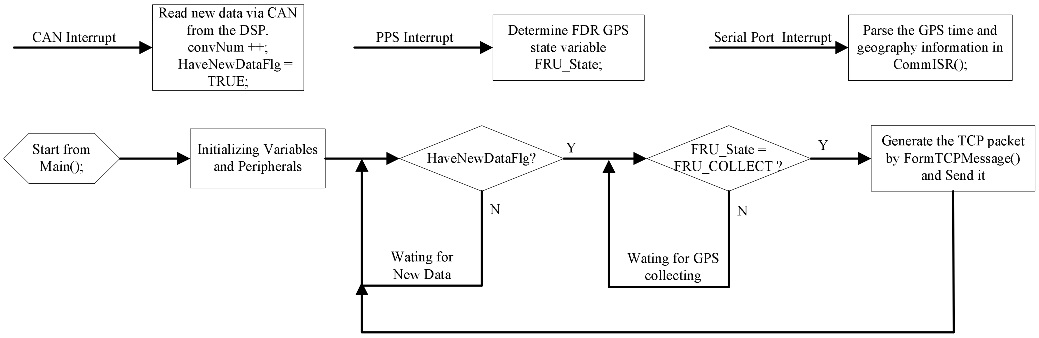

CAN Interrupt
Read new data via CAN from the DSP. convNum ++; HaveNewDataFlg = TRUE;
PPS Interrupt
Determine FDR GPS state variable FRU_State;
Serial Port Interrupt
Parse the GPS time and geography information in CommISR();
Start from Main();
Initializing Variables and Peripherals
HaveNewDataFlg?
Y
N
FRU_State = FRU_COLLECT ?
Y
N
Generate the TCP packet by FormTCPMessage() and Send it
Wating for New Data
Wating for GPS collecting

Fig. 2.4 C8051 Program Flowchart

Likewise, the main() loop of the C8051 program starts with initializing the variables and peripherals before repeating the infinite loop that checking the status of the variable HaveNewDataFlg that is set to TRUE by the CAN interrupt ISR upon the arrival of a new packet of phasor data transmitted from the DSP module. There too is another infinite loop to wait on the recovery of the GPS information when GPS signal errs. When the HaveNewDataFlg is refreshed by the CAN interrupt ISR and the FRU_State is set to be FDR_COLLECT, the program generates the newest TCP packet by FormTCPMessage() function by integrating the phasor data from the DSP board, and the parsed GPS time received from the serial port. The packet then is sent to the MOXA box, which is the module to send the data to the server by the Internet.

In the PPS interrupt ISR, the program determines if the GPS state variable FRU_State is in the collecting status by checking whether or not the time interval of the most recent two adjacent PPS interrupts is between 900 ms and 1100 ms. It is managed by the Serial Port interrupt ISR to parse the GPS time and geographic information in CommISR() function, rendering a set of variables containing the GPS information for other parts of the program to use. The CAN interrupt ISR primarily reads new data via CAN from the DSP, then increment convNum counter by 1 and set flag variable HaveNewDataFlg to be TRUE. The HaveNewDataFlg sign communicates to the main() function that a new phasor data is ready for processing.

2.1.2 FDR Placement

The FDR units are designed to record dynamic frequency information for power system analysis (and control in the future). The placement of FDR units is an important issue for the FNET system. Their location should be selected to effectively reflect the different frequency clusters of inter-area oscillations and to cover as broad an area as possible, in order to capture dynamic behavior of larger system disturbances. Just like PMU placement, FDR locations should represent the system frequency, effectively describe the behavior of major inter-connected systems, and provide information on the large area load behavior. The present placement of FDRs is aimed to cover all the regional reliability regions that compose North American Electric Reliability Council (NERC). In the future FNET system, more FDR units will be located close to major generation centers, major transmission tie lines and load concentrated areas for local frequency oscillation mode study. Up to now there have been approximately 100 measurement devices deployed all around the North America. Figure 2.5 shows the locations of the units that are deployed in the FNET system. Thanks to their minimal installation cost and plug and play operation, FDR units can be easily relocated, if necessary.

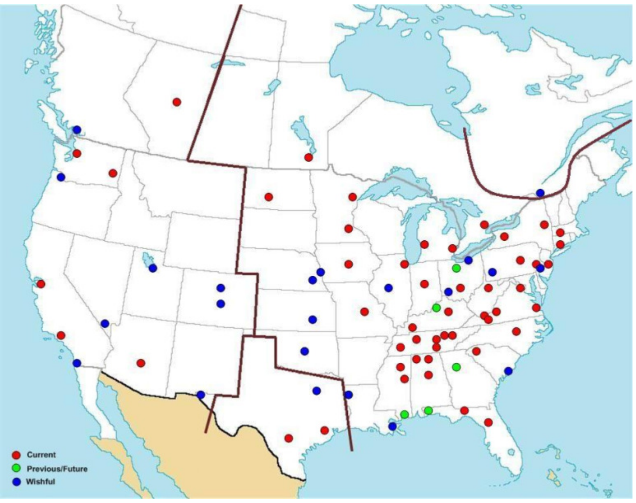


Fig. 2.5 FDR Distribution Map

2.1.3 FNET Server Hardware and Software Configuration

The server receives data from different FDR units, processes the data, manages the database, performs data analysis, and supports the web service for Internet users. The frequency IMS server can be physically located anywhere. The server has following system configurations:

a) Hardware configuration: Two quad CPU (Intel Xeon CPU 3.0GHz), 16GB memory servers with 6TB hard disks are employed to meet the need of reliable data transmission, processing and web service. The backup data is stored both on the hard disk of the server and in an external hard drive.

b) Server software configuration: Windows Server 2003 is the host operating system. A commercial database is installed at the database server machine, as the backend database. C++ is employed for the whole server program design and communication. Internet standard web browsers are used as the interface for the user to access the frequency data.

The server mentioned above is an abstract notion of the entire set of the server functionalities in a virtue computer. Practically, it consists of a number of computers working in synergy to realize all the functionalities demanded by the FNET system. There are four servers in the FNET server system whose schematic layout diagram is shown in Figure 2.6.

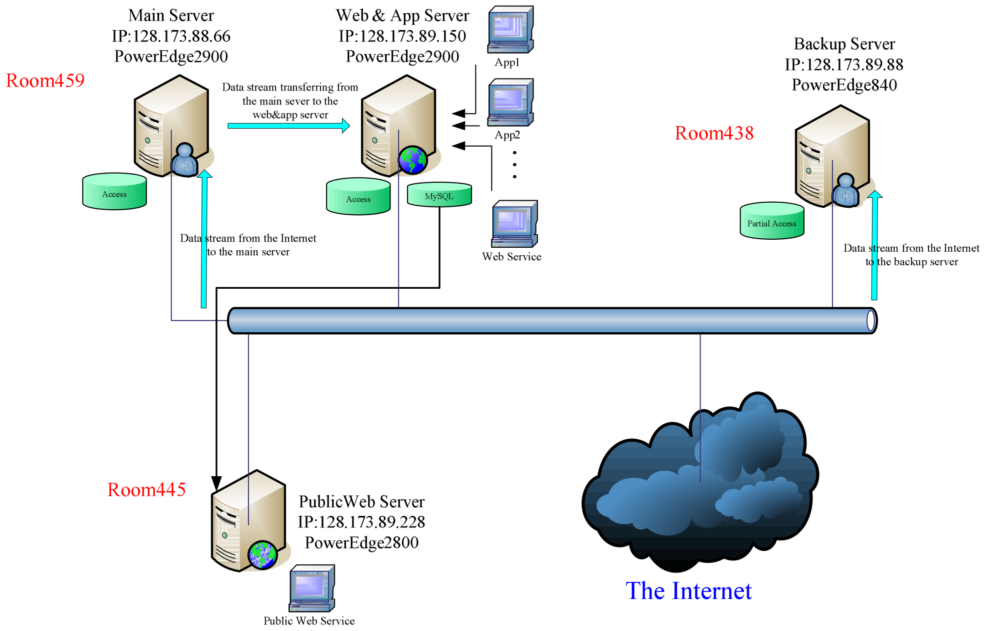


Fig. 2.6 FDR Server Layout

The main server and the backup server are the two destinations that the FDRs send measurement data to. The main server is the pivotal machine of the server system, the assigned functions of which are saving data to Access database and forwarding data to as many machines as needed. This data dissemination feature of the main server is rather useful a means to expand the flexibility of the server system. The application server is one of IP addresses the main server forwards the data to, and it is loaded with an assortment of applications which include the data saving to MySQL database for web display, the on-line event trigger, the event location estimation and web service, etc. For simplicity sake, the public web server only works as a standalone website server to show the introduction of FNET, the average frequency table, the frequency distribution map. It retrieves data from the MySQL database on the application server. The backup server is a duplicate of the main server as a standby substitute of the main server in case it does not work properly.

2.2 FNET Data Communication

2.2.1 Introduction of Socket Communication

In computer networking, an Internet socket (or commonly, a network socket or socket) is the endpoint of bidirectional communication flow across an Internet Protocol-based computer network, such as the Internet. Internet sockets are an application programming interface (API) in an operating system, used for in inter-process communication. Internet sockets constitute a mechanism for delivering incoming data packets to the appropriate application process or thread, based on a combination of local and remote IP addresses and port numbers. Each socket is mapped by the operational system to a communicating application process or thread.

A socket address is the combination of an IP address (the location of the computer) and a port (the communication service type) into a single identity, much like one end of a telephone connection is between a phone number and a particular extension line at that location.

An Internet socket is characterized by a unique combination of the following:

♦ Protocol (TCP, UDP or raw IP). Consequently, TCP port 53 is not the same socket as UDP port 53.

♦ Local socket address (Local IP address and port number)

♦ Remote socket address (Only for established TCP sockets. As discussed in the Client-Server section below, this is necessary since a TCP server may serve several clients concurrently. The server creates one socket for each client, and these sockets share the same local socket address.)

The operating system forwards incoming IP packets to the corresponding application or service process by extracting the socket address information from the IP and transport protocol headers.

Within the operating system and the application that created a socket, the socket is referred to by a unique integer number called socket identifier or socket number.

In Internet standards, in many textbooks as well in this article, the term "socket" refers to an entity that is uniquely identified by the socket number. In other textbooks, the socket term refers to a local socket address, i.e. a "combination of an IP address and a port number". In the original definition of socket given in RFC 147 as it was related to the ARPA network in 1971, a socket was "specified as a 32 bit number with even sockets identifying receiving sockets and odd sockets identifying sending sockets." Today, however, sockets are bidirectional.

Communicating local and remote sockets are called socket pairs. Each socket pair is described by a unique 4-tuple struct consisting of source and destination IP addresses and port numbers, i.e. of local and remote socket addresses. As seen in the discussion below, in the TCP case, each unique socket pair 4-tuple is assigned a socket number, while in the UDP case, each unique local socket address is assigned a socket number.

There are several Internet socket types. What FNET uses is Stream sockets, also known as connection-oriented sockets, which use Transmission Control Protocol (TCP) or Stream Control Transmission Protocol (SCTP).

Computer processes that provide application services are called servers, and create sockets on start up that are in listening state. These sockets are waiting from initiatives from client programs.

A TCP server may serve several clients concurrently, by creating a child process for each client and establishing a TCP connection between the child process and the client. Unique dedicated sockets are created for each connection. These are in established state, when a socket-to-socket virtual connection or virtual circuit (VC), also known as a TCP session, is established with the remote socket, providing a duplex byte stream.

A server may create several concurrently established TCP sockets with the same local port number and local IP address, each mapped to its own server-child process, serving its own client process. They are treated as different sockets by the operating system, since the remote socket address (the client IP address and/or port number) are different, i.e. since they have different socket pair tuples.

Development of application programs that utilize this API is called socket programming or network programming.

These are examples of functions or methods typically provided by the API library:

♦ socket() creates a new socket of a certain socket type, identified by an integer number, and allocates system resources to it.

♦ bind() is typically used on the server side, and associates a socket with a socket address structure, i.e. a specified local port number and IP address.

♦ listen() is used on the server side, and causes a bound TCP socket to enter listening state.

♦connect() is used on the client side, and assigns a free local port number to a socket. In case of a TCP socket, it causes an attempt to establish a new TCP connection.

♦ accept() is used on the server side. It accepts a received incoming attempt to create a new TCP connection from the remote client, and creates a new socket associated with the socket address pair of this connection.

♦ send() and recv(), or write() and read(), or recvfrom() and sendto(), are used for sending and receiving data to/from a remote socket.

♦ close() causes the system to release resources allocated to a socket. In case of TCP, the connection is terminated.

2.2.2 Structure of the FNET Socket Communication

The overall hierarchy of FNET socket communication is basically a 2-tier structure with each tier associated with a C++ socket class derived from the base socket class CAsyncSocket{}. The two derived classes are MyServerSocket{} and MySocket{}. The MyServerSocket{} class is on the higher level in relation with MySocket{} as far as the structural layout is concerned. A MyServerSocket{} class object is declared for a port dedicated for a specific unit, in the constructor of which a MySocket{} class object is declared. This means a MySocket{} object is bound to a MyServerSocket{} object and therefore subordinate to it, and the two objects are performing different functions of socket communication assigned to them. A schematic diagram that illustrates the FNET socket communication and the relations between the two socket objects is shown in Figure 2.7.

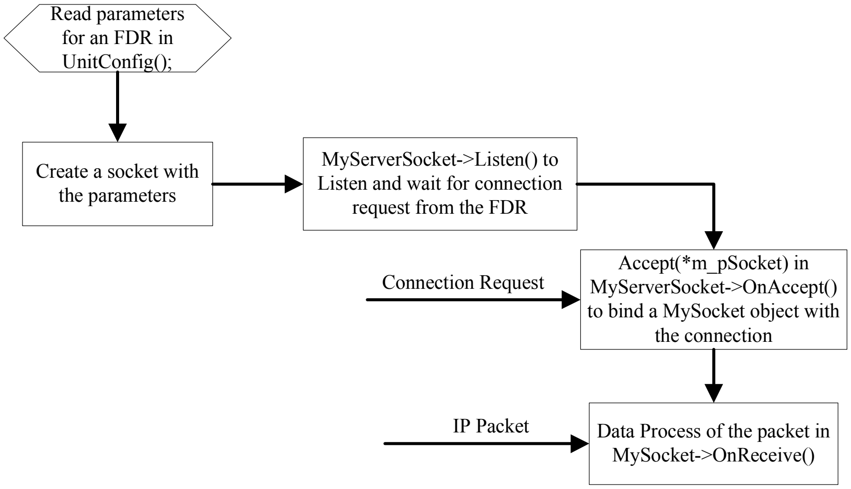

Read parameters for an FDR in UnitConfig();
Create a socket with the parameters
MyServerSocket->Listen() to Listen and wait for connection request from the FDR
Connection Request
Accept(*m_pSocket) in MyServerSocket->OnAccept() to bind a MySocket object with the connection
IP Packet
Data Process of the packet in MySocket->OnReceive()

Fig. 2.7 FNET Socket Communication Overall Structure

The sever program reads the unit parameters, including socket port number, from the UnitConfig() function during initialization. In the meantime, the server allocate memory for a MyServerSocket{} class to construct a socket object, and then create the Windows socket and attach it using create() function inherited from CAsyncSocket{} base class. If the socket is created successfully, the program proceeds to open the listen() function to waiting for the connection request sent by individual FDRs from the Internet. Once the request comes, the server program is directed to the OnAccept() function according to the 'Message Based, Event Driven' mechanism of the Windows API. A MySocket object m_pSocket, which is created in the constructor of MyServerSocket{} object, is bound to the connection requested by the specific FDR by Accept(*m_pSocket) instruction in OnAccept(). An exception information is posted if there is error in the execution of Accept(*m_pSocket) instruction, otherwise the Internet connection between the FDR and the designated port of the server is established.

Upon the establishment of the connection, the server program starts to receive IP packets with voltage phasor data contained therein sent by the FDR from the Internet in the overridden function MySocket->OnReceive(). The ProcessData() function is then invoked to process the data unwrapped, extracted and analyzed from the IP packets. The procedures in proper pre-set sequence are to be executed in an endless loop until the server program is manually shut down and OnClose() functions are correspondingly called to terminate the running Internet connection.

As can be seen from the previous explanation, the key parts of FNET socket communication are within the three functions: FNETServerDlg->ReadUnitConfig(), MyServerSocket->OnAccept() and MySocket->OnReceive(). The ReadUnitConfig() function creates MyServerSocket{} objects according the unit information recorded in the UnitConfig.txt file when the main dialogue class object is being initialized. As the server creates the sockets one at a time for each unit, it also invokes some MyServerSocket{} functions to set certain relevant running parameters for each unit, such as setting the interconnection affiliation information of the unit by calling the SetLocation() function, setting unit geographic information by calling SetUnitInfo() function, etc. It needs to be noted that the primary task of the MyServerSocket{} class is to listen to the Internet and to immediately establish a TCP connection between the server and the FDR once the connection request is received, and the task is designed to be fulfilled in the OnAccept() function. After the connection is set up, the communication activities turn to the MySocket{} object to receive, analyze and process the data from the FDR, which are mainly performed in OnReceive() function. The data reception is implemented by the Receive((void*)&m_buffer[…]) instruction with the data being saved to the array m_buffer[]. The data processing carried out in OnReceive() function after being validated by ValidateMessage() function comprises saving data to databases, updating the man machine interface (MMI) of the server with the new measurement, on-line event detecting (on-line triggering) and exporting data to text files for triangulation module if an event is detected.

One complication about the socket communication part of the server program that needs to be explained is the memory leak issue, which is caused by the instruction Accept(*m_pSocket) and successfully tackled by adding to new code below before the instruction.

```
if ( m_bConnected )
{//Add by Tao, May. 1, 2007, for the memory leak issue.
delete m_pSocket;
m_pSocket = new CMySocket(m_pDlg);
m_pDlg->m_SystemInfo.Format("Abnormal connection request report!");
m_pDlg->UpdateData(false);
}
```

Numerous unexpected abnormalities would occur in the communications between the FDRs and the server, one of which is the intermittent interrupts of the Internet connection. When the Internet connection between an FDR and the server is resumed after being interrupted for uncertain reasons, a new connection request will be made by the FDR to the server, and the server will respond to the request by executing the instruction Accept(*m_pSocket). The system deems the resumed same connection a new one and it intends to associate a new MySocket{} object with the connection. The code shown above checks if the connection has already been established before by the flag variable m_bConnected, and it deletes the current MySocket{} object and create a new one if so. As a result, the memory will not increase because it deletes the old object. The old server code does not contain the code above, and therefore the Accept() function will allocate new memory to the requested connection because it does not overwrite the old MySocket{} object for the seemingly new connection. Consequently, the memory keeps growing due to the high likelihood of the intermittent connection interruption, and the memory leak will be greatly exacerbated if the FDR goes wrong and keeps making connection requests as opposed to sending measurement data.

The memory leak problem is well resolved after the code described above is added to the server code. However, the on-line trigger loop-out issue arises without due considerations to the on-line trigger variables. This is because there is a flag variable m_bMonitoring of the socket object that determines if the on-line trigger function should be executed or branched for the specific FDR, and this variable will be reset by the code that handles the memory leak if the connection between the FDR and the server has been interrupted and then resumed. The design of the server program allows certain number of the FDRs run the on-line trigger function to monitor the frequency disturbances. When the Internet connections of all those monitoring units have been interrupted, there then will be no one-line trigger function of the socket objects running to detect frequency events, hence the trigger loop-out issue. The solution for this problem is to move the parameter value-assigning functions of the MySocket{} class, SetMonitor() and SetLocation() specifically, from before the memory leak handling code to after it. By doing that, the on-line monitoring flag and the FDR location variable are given the right values that are preserved in the corresponding MyServerSocket{} object, which makes the on-line trigger work correctly even when the connection has been interrupted.

2.2.3 Forwarding Feature of the FNET Socket Communication

For data reliability considerations, the main server deals solely with receiving and transferring the data, leaving the data processing functions that complicate the running of the program to another server that is less critical. Due to the fact that it relieves the main server of many roles that have been proven to potentially cause it to bog down in some unpredicted conditions, such as writing data to MySQL database, on-line event trigger and so on, this policy can greatly improve the security and completeness of FNET data, which is the foundation and source of every facet of the FNET research and hence their utmost importance.

The data transferring function, or data forwarding activity, is implemented in the main server by creating FwdSocket{} class objects that correspond to the IPs the main server are required to distribute the data to, and then re-routing data to those IPs right after the data has been received and confirmed. Apart from saving data to the Access database, the main server acts in likeness of a data communication relay that receives the data and passes on to as many destinations as the system needs, which remarkably increases the flexibility of the server function assignment and the data layout.

The general structure of how the data forwarding techniques work bears a great resemblance of the overall structure of the FNET socket communication depicted in Figure 2.7. The first variation of the data forwarding scheme of the socket technique is that there is an FwdSocket{} class in addition to the MyServerSocket{} class and the MySocket{} class. A MySocket{} object is created in the constructor of MyServerSocket{} class as the Receiving socket object for the Accept() function; FwdSocket{} objects whose IPs are stipulated in the IPConfig.txt file as the IPs the server needs to broadcast the data to are created in the constructor of MySocket{} class as the carrier of the socket object for the sending the data out to other servers. Furthermore, in the ProcessData() function, the data forwarding program tries to connect to the machines that has one of the IPs pre-set in configuration text file, sets the data in a certain form, and calls the OnSend() function to distribute the data to the destinations. The data are sent to the Internet by the Send() function of the FwdSocket{} class.

2.3 FNET Database Systems

2.3.1 Overview of the FNET Database Techniques

As for the database part of FNET applications, the database system has been practically divided by early developers into two database management systems (DBMSs) with different emphases, one of which is for the data storage, the other for the web display. To meet those needs, the database management systems (DBMSs) FNET uses are Microsoft Access and Sun

Microsystems MySQL.

Microsoft Access is a relational database management system from Microsoft that combines the relational Microsoft Jet Database Engine with a graphical user interface and software development tools. It can be used as a small database programming systems which can contain the end-user application programming within the database as a single item. It stores data in its own format based on the Access Jet Database Engine, and it can also import or link directly to data stored in other Access databases, Excel, SharePoint lists, text, XML, Outlook, HTML, dBase, Paradox, Lotus 1-2-3, or any ODBC-compliant data container including Microsoft SQL Server, Oracle, MySQL and PostgreSQL. For such database applications as in FNET, it is expedient for all programming to be stored with the data, which results in a single file and is easier to develop a broad spectrum of functionalities

MySQL is also a relational database management system (RDBMS) for FNET web display applications. It is rated as one of the two the best overall performers. Its popularity for use with web applications is closely tied to the popularity of PHP, which is often combined with MySQL. Several high-traffic web sites (including Flickr, Facebook, Wikipedia, Google (not for searching), Nokia and YouTube) use MySQL for its data storage and logging of user data. It is fast and stable, and is possessed of a high scalability. It also supports a vast majority of features considered important by the database community, such as transactions, row-level locking, foreign keys, sub-queries, and full text searching. It is thought to be one of the most widely-used back-end databases.

A back-end database is a database that is accessed by users indirectly through an external application rather than by application programming stored within the database itself or by low level manipulation of the data (e.g. through SQL commands). A back-end database stores data but does not include end-user application elements such as stored queries, forms, macros or reports.

There are varied software API methods for using database management systems (DBMS). In the FNET system, ODBC and OLE DB (object-linking and embedding database) are used for database manipulation.

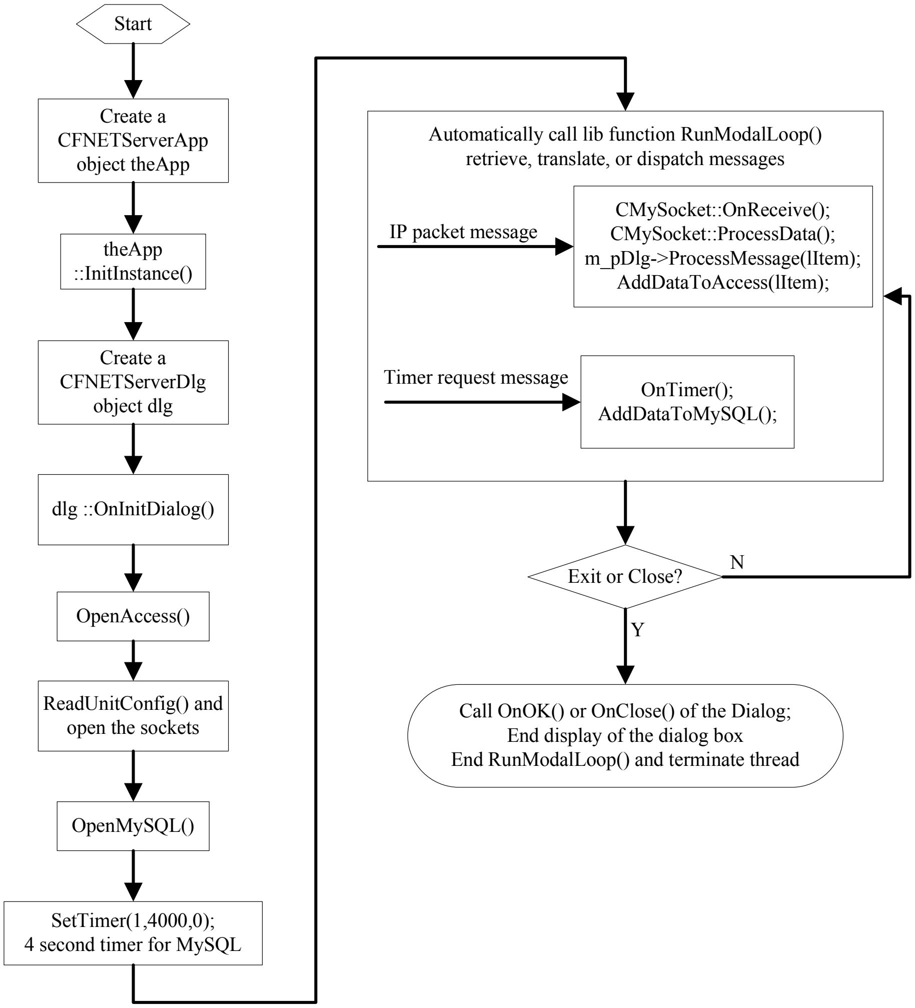

Start
Create a CFNETServerApp object theApp
theApp ::InitInstance()
Create a CFNETServerDlg object dlg
dlg ::OnInitDialog()
OpenAccess()
ReadUnitConfig() and open the sockets
OpenMySQL()
SetTimer(1,4000,0);
4 second timer for MySQL
Automatically call lib function RunModalLoop()
retrieve, translate, or dispatch messages
IP packet message
CMySocket::OnReceive();
CMySocket::ProcessData();
m_pDlg->ProcessMessage(lItem);
AddDataToAccess(lItem);
Timer request message
OnTimer();
AddDataToMySQL();
Exit or Close?
N
Y
Call OnOK() or OnClose() of the Dialog;
End display of the dialog box
End RunModalLoop() and terminate thread

Fig. 2.8 FNET Socket Database System

In light of the particular characteristics of the two database systems, the manner in which the FNET server organizes the databases is illustrated in Figure 2.8.

When the server program starts, the InitInstance() function of a CFNETServerApp{} object theApp is automatically executed and a CFNETServerDlg{} object dlg is created in the function. When the dialog box dlg is being initialized, the function OnInitDialog() is invoked on the MFC conventional basis, and the two functions, OpenAccess() and OpenMySQL(), are called to open the two databases. In the meantime, the sockets whose port numbers specified in the UnitConfig.txt are created and set to listen to the data sent from the FDRs, and a system timer is arranged to start and execute at a 4-second interval. After the initialization work is successfully finished, the main serve program runs on a loop to wait on messages to interrupt and then process the message accordingly. Once the Exit or Close buttons are clicked, the program will end the dialog box and the loop, and terminate the thread to conclude the program.

Two interrupts are served in the system loop applied in the RunModalLoop() function, the IP packet interrupt and the timer interrupt. In the service program for the IP packet interrupt, the server updates the data that has newly come in to all its modules, including adding the new measurements contained in the packet to the Access database by calling the function AddDataToAccess() to archiving all the measurements, which includes the frequency, the voltage magnitude, the voltage angle and the time stamp, in full completeness. The operation rate for the Access database is 10 times per second for the N running FDRs, which means it is 10*N times per second in total. In the service subroutine for the timer interrupt, the AddDataToMySQL() function simply saves the average frequency over the most recent 4 seconds to the MySQL data for the use of the FNET web services.

2.3.2 FNET Access Database

The MFC provides a number of database application support classes, and the ODBC database support class CDatabase{} is the one that FNET uses for Access database manipulation. The detailed flowchart of the Access database operation is shown in Figure 2.9.

When the server program starts up, a CDatabase{} object m_pConn is created in the constructor of the main dialog box object dlg, and the OpenAccess() function is called in the initialization of the dialog box to open the destination Access database. The ODBC connection is specified in the OpenEx() function of the CDatabase{} class by equating DSN to the data source FNET that has been defined to be db1.mdb in the same folder as the server program. The definition of the data source is performed at the operating system (OS) level, for instance it is set at the Control Panel/ODBC source in the Windows server OS.

If there is a new IP packet that requests processing after the Access database is connected, the OnReceive() function is invoked to answer the request and the measurement data are received and then validated. When the data is successfully validated, the AddDataToAccess() function is carried out in the ProcessMessage() function that handles updating new data to different part of the application.

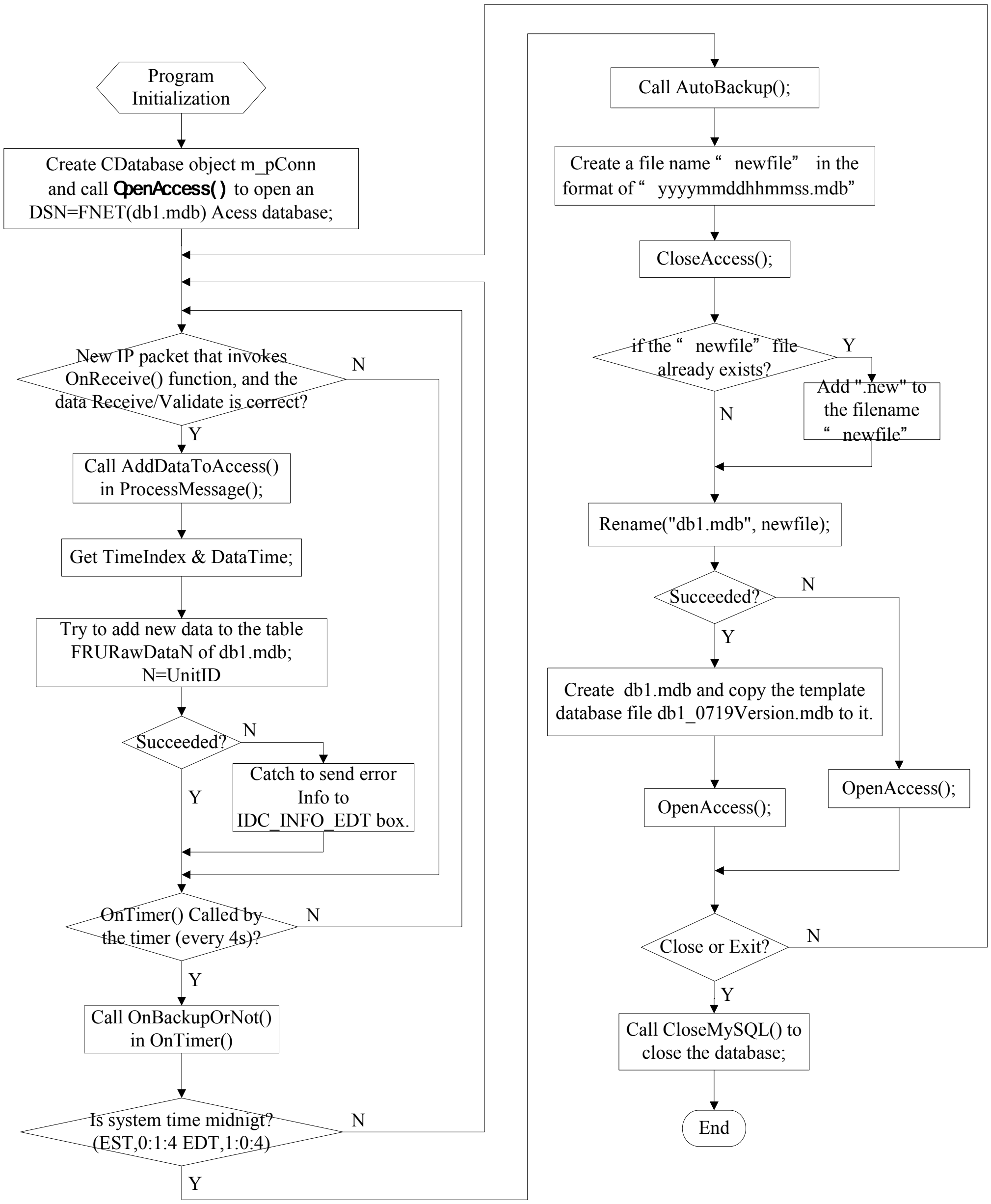

Program Initialization
Create CDatabase object m_pConn and call OpenAccess() to open an DSN=FNET(db1.mdb) Acess database;
New IP packet that invokes OnReceive() function, and the data Receive/Validate is correct?
N
Y
Call AddDataToAccess() in ProcessMessage();
Get TimeIndex & DataTime;
Try to add new data to the table FRURawDataN of db1.mdb; N=UnitID
Succeeded?
N
Y
Catch to send error Info to IDC_INFO_EDT box.
OnTimer() Called by the timer (every 4s)?
N
Y
Call OnBackupOrNot() in OnTimer()
Is system time midnigt? (EST,0:1:4 EDT,1:0:4)
N
Y
Call AutoBackup();
Create a file name “ newfile” in the format of “ yyyymmddhhmmss.mdb”
CloseAccess();
if the “ newfile” file already exists?
Y
N
Add ".new" to the filename “ newfile”
Rename("db1.mdb", newfile);
Succeeded?
N
Y
Create db1.mdb and copy the template database file db1_0719Version.mdb to it.
OpenAccess();
OpenAccess();
Close or Exit?
N
Y
Call CloseMySQL() to close the database;
End

Fig. 2.9 FNET Access Database Flowchart

The AddDataToAccess() function is passed the argument lItem that contains the time stamp, the measurements and the unit number, from the upper level functions. Afterwards, the function decomposes the time and measurements, assemble them in the right format and save the data into the corresponding table of the database by executing an insert SQL statement. The table names are in the ‘FRURawDataN’ format where N is the unit number. There is an exception handling mechanism that acts and reports to the server interface the alarm of the abnormality the when the execution of the SQL statement goes wrong.

If the data keep coming in and the Access database remains the same without renewal, the size of the database will exceed the limit some time and the data sent after that critical point will be left unsaved. The server solves the issue by running the AutoBackup() function at the midnight of the Eastern Time (EST or EDT) to rename the db1.mdb file in the ‘yyyymmddhhmmss.mdb’ format, and copy the template database db1_0719Version.mdb to a new db1.mdb. The connection between the server program and the data source is suspended during the process of the data source file renewal. The check on the midnight time is done in the timer service routine OnTimer() which is executed every 4 seconds.

When the Close or Exit button is clicked, the Access database service is discontinued along with the close of the main server program.

2.3.3 FNET MySQL Database

CRecordset{} class, another ODBC database support class in MFC hierarchy, is employed in the FNET MySQL database system, and a new ODBC type class CFNETRecord{} is derived from that base class and then used as the core component of the MySQL database application.

Similar to the Access database, a CFNETRecord{} object m_rs is created in the constructor of the main dialog box object dlg, and the source MySQL database is opened in the initialization of the dialog box. The data source is also set at the Control Panel/ODBC source in the Windows server Operating System, which is to bind a MySQL database to a data source name referenced by the object m_rs. However, as opposed to specifying the data source in the OpenEx() function of the CDatabase{} class, the CRecordset{} class framework calls GetDefaultConnect() function to get the default connection string for the data source of the CFNETRecord{} object. In addition, the framework calls DoFieldExchange() function to automatically exchange data between the field data members of the object and the corresponding columns of the current record on the data source.

Generally speaking, the data is added to the MySQL database by executing the AddDataToMySQL() function every 4 seconds in the OnTimer() message service routine. The server program checks whether or not the object allows updates by invoking the CanUpdate() function, and proceeds to add a new record to the table by the function AddNew(). The record's fields are initially Null, which, in database terminology, means "having no value". To complete the operation, the Update() function must be called to save the changes to the data source.

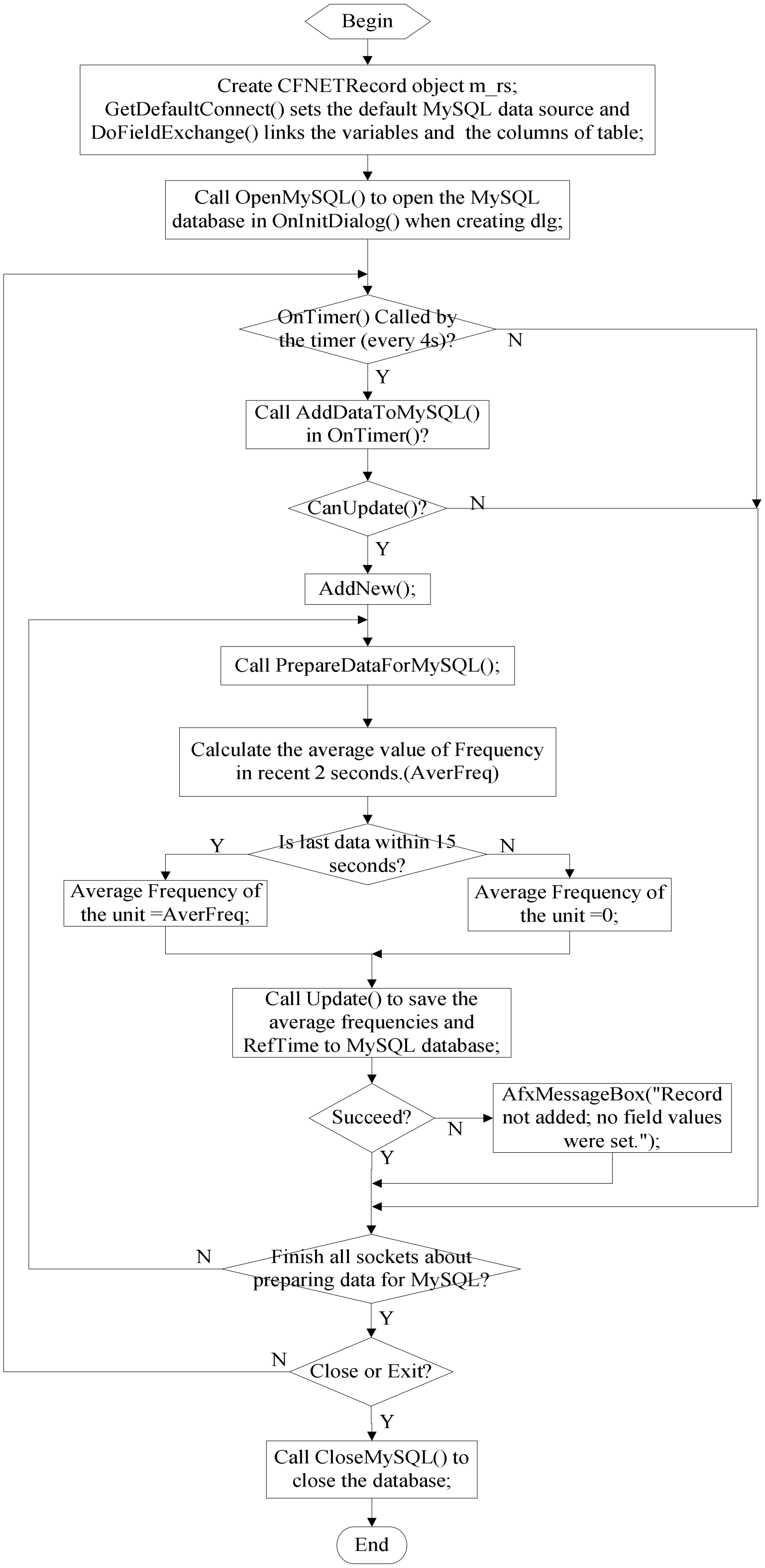

Begin
Create CFNETRecord object m_rs;
GetDefaultConnect() sets the default MySQL data source and
DoFieldExchange() links the variables and the columns of table;
Call OpenMySQL() to open the MySQL
database in OnInitDialog() when creating dlg;
OnTimer() Called by
the timer (every 4s)?
N
Y
Call AddDataToMySQL()
in OnTimer()?
CanUpdate()?
N
Y
AddNew();
Call PrepareDataForMySQL();
Calculate the average value of Frequency
in recent 2 seconds.(AverFreq)
Y
Is last data within 15
seconds?
N
Average Frequency of
the unit =AverFreq;
Average Frequency of
the unit =0;
Call Update() to save the
average frequencies and
RefTime to MySQL database;
Succeed?
N
AfxMessageBox("Record
not added; no field values
were set.");
Y
N
Finish all sockets about
preparing data for MySQL?
Y
N
Close or Exit?
Y
Call CloseMySQL() to
close the database;
End

Fig. 2.10 FNET MySQL Database Flowchart

Before the Update() function, the PrepareDataForMySQL() function is called to prepare the data for the updating. It basically calculates the frequency average of the most recent 2 seconds and compares the GPS time of the last data point with the current system time to determine whether the data is too old by the time difference. If the data is not interrupted, the time stamp of the last data point should be very close to the current system time and the time difference should be almost zero, otherwise the time difference is a large number. The pre-set threshold for this logic is 15 seconds.

If the data is determined to be old, zero will be assigned to the frequency field data member of the object, whereas if the data is sufficiently recent, the calculated frequency average will be assigned to the field data. The Update() function then is called to complete the operation. A piece of error information will be displayed in the Windows MessageBox if the process is not successful.

The aforementioned procedures are done to treat with one IP packet from a certain FDR, and they are repeated for all the FDR IP packets until there is no unhandled IP packet in queue.

In like manner, when the Close or Exit button is clicked, the MySQL database service is discontinued along with the close of the main server program. The MySQL database flowchart is shown in Figure 2.10.

2.4 FNET Website

FNET servers host three FNET websites, namely, FNET private website, FNET event database website and FNET public website.

The FNET private website, hosted on the application server, provides a brief introduction of FNET, lists the detailed FDR deployment information, as well as retrieves frequency data from the MySQL database and displays them in the forms of table, map and gradient.

As an advertisement version of the FNET private website, the FNET public website, hosted on the public website server, similarly acquires frequency data from the MySQL database, but displays them only in the forms of table and gradient. It also provides a brief introduction of FNET, but it does not list the FDR deployment information because of safety considerations. Instead, it gives four event analysis examples, which comprise of FNET frequency plot, event location estimation and movie/report link, and adds ‘FNET Gallery’ column and ‘Contact us’ column to offer more information of FNET.

The FNET event database website, hosted on the application server, makes available for the subscribed users the complete repository of the frequency plots and location estimations of all the events that have been detected, analyzed and saved by the triangulation module. The website runs as both the interface of the postmortem event analysis database and the situational awareness of the North America power systems with extremely easy reference, which to a great extent facilitates the research of the frequency behavior of large-scale power grids and leads to the better understanding of or the contribution to the knowledge in that area.

The FNET private website and the FNET event database website require a password to access, whereas the FNET public website is freely open to all Internet users. All the website construction efforts use PHP embedded with Javascript as the development scripting languages, and are built upon Windows Server 2003 version Internet Information Services (IIS).

2.4.1 FNET Private Website

The private website uses the HTTP/HTTPS service supplied by IIS 6.0 for the web server programs to operate by accepting HTTP requests from the client, and providing an HTTP response to the client. The web server programs are written in PHP coupled with Javascript. This method can enhance the performance with respect to the data retrieving function, data display function and so on, by the mixture coding of the two scripting languages.

In order for the web service programs to run, the PHP engine version 5.2.3 is installed to parse the scripts after the IIS platform is set up. The PHP scripts are kept as human-readable source code and will be compiled at runtime by the PHP engine, which increases their execution speed. Also, php5isapi.dll needs to be added to the Application Extension Mapping so that the PHP files are parsed correctly by the right engine. Since the scripts require interface with MySQL database, the MySQL database management system is to be installed and related databases to be imported from an external storage medium.

There are 9 PHP files that are stored on the application server in the folder C:\Inetpub\wwwroot, the default folder for IIS applications. The PHP files that serve the private website are ‘index.php’, ‘newFreqVals.php’, and ‘cdgradgen.php’, while the PHP files ‘fnetevdb.php’, ‘logincheck.php’, ‘logout.php’, ‘setpw.php’, ‘fnetevdbcont.php’, and ‘chknpass.php’ serve the event database website.

The ‘index.php’ script first displays the interface to authenticate the subscribed users for security purposes.

The authentication page stays until correct username and password are input or gives the incorrect input message when wrong information is keyed in. The script then retrieves FDR location data and the most recent 4 second frequency averages from table fdrloc2 and table serverinfo of the MySQL database ‘fnetserver’ as it switches to the FNET server web display page. There are five buttons, ‘About FNET’, ‘Table Display’, ‘Map Display’, ‘Map Gradient’, and ‘FDR Locations’, on the page that link to different contents to fill bottom frame of the page.

The ‘About FNET’ directs to a brief introduction of FNET. The ‘FDR Locations’ lists the geographic information of all the FDRs. The rest of the buttons show the power system frequency distribution throughout the North America in different forms.

The default frame content is ‘Table Display’, which means the frequency averages are displayed in a table format when the web display page is entered after the username and password authentication. Script ‘newFreqVals.php’ is invoked by a frame replace function in a timer service subroutine every 4 seconds to refresh the frequency average data in the table. The same timer service subroutine also calls ‘cdgradgen.php’ to display frequency gradient on a US and Canada map by tinting each pixel of the map with certain color in the color scheme according to the trigonometric analysis result based on the frequency values of different FDRs and the geographical coordinates of the FDR location if ‘Map Gradient’ is activated. When the button ‘Map Display’ is hit, the website loads a flash mapDisplay.swf to show the frequency information in discrete dots on the map that correspond to where the FDRs are located. It displays the dots in colors associated with their frequency value and registers the accurate frequency when a specific FDR dot is pointed at.

2.4.2 FNET Event Database Website

The event database website starts with a login page run by the script ‘fnetevdb.php’, in which script ‘logincheck.php’ is linked to authenticate the user.

If the authentication is successful, ‘fnetevdbcont.php’ automatically runs and the recorded events database interface shows up with a drag down box to choose which date the user wants to browse. A new entry is inserted to the log_file table of the MySQL fnet-security database after each successful login. All the past events saved in the table fnetevent of the MySQL database fnetserver are filled in the drag down box, and if a date that has frequency events is selected, all the events of that date are listed for the viewers. Once an event is chosen, the detailed information of the event is displayed, including the event date, the time, the size, the event type, the unit detection order, the event plot chart and the event location map. The event plot chart and event location map are obtained from the folder C:\Inetpub\wwwroot\eventdbpics in term of the names of the two graphic files that are also stored in table fnetevent together with other parameters of the event.

If the authentication fails, meaning the username and password the user inputs cannot be found in the user_rec table of the MySQL fnet-security database, the web page automatically switches to the ‘logout.php’ page. If the user is new and leaves the password blank, the website goes to the ‘setpw.php’ page which allows the user to set the password he desires and saves it to the database on the server. A ‘chknpass.php’ script is called to check if the passwords input twice are identical.

The MySQL database is pivotal to the functions of the website. Not only is it accessed by the website service scripts, but also it is accessed and updated by the server program and the triangulation module.

There are two MySQL databases: database fnetserver, which includes table fdrloc2, fnetevent and serverinfo, and database fnet-security, which includes table log_file and user_rec. The server program receives the data coming from the FDRs and writes the frequency average to the serverinfo table, and triangulation module writes the result to the fnetevent table once there is a frequency disturbance. When a new FDR is deployed, the information of the FDR is input to the fdrloc2 table manually. The table user-rec is for manually adding or deleting the subscribers of FNET event database, and the table log_file is used to keep record of the usage of the event database website by the subscribers.

The organization of the PHP files of the private website and the FNET event database website is shown in Figure 2.11.

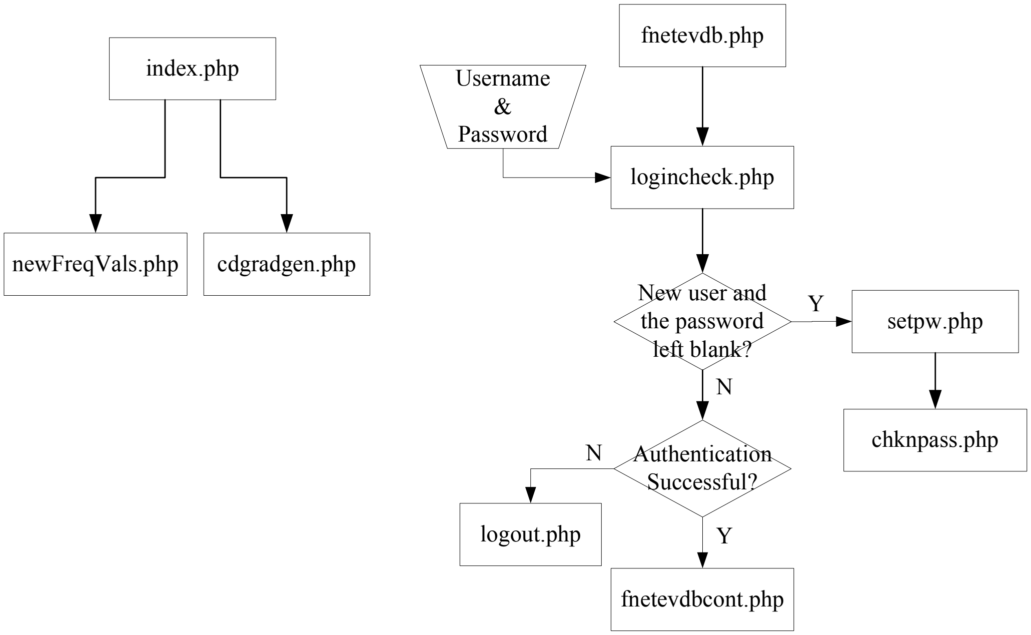


Fig. 2.11 Organization of the PHP Files

2.4.3 FNET Public Website

The FNET public website is a simplified version of the private website with some new features added unto it. It can be regarded as an advertisement version of FNET private website just as it was mentioned in the previous section. The same procedures to install the PHP engine and the MySQL database management system should be conducted to run the public website correctly.

There are 3 PHP files in the folder C:\Inetpub\wwwroot: ‘index.php’, ‘newFreqVals.php’, and ‘cdgradgen.php’. No authentication mechanism is considered in the public website, and therefore the website directly goes to the page that introduces FNET and FNET Consortium without username and password interface when browsed on. There are six buttons, ‘About FNET’, ‘Table Display’, ‘Map Gradient’, ‘Sample Events’, ‘FNET Gallery’,and ‘Contact Us’, on the page that link to different contents to fill bottom frame of the page, which is different from those of the private website.

The ‘About FNET’ directs to a brief introduction of FNET and the FNET Consortium, which is the default frame content of the web page.

The scripts on the public website server retrieves FDR location data and the most recent 4 second frequency averages from table fdrloc2 and table serverinfo of the MySQL database fnetserver from the application server via the Internet. There is an IP permission setting on the application server to disable any direct access to the MySQL database on the server except from the public website server. Otherwise, a cyber security issue could be raised upon the safety of the MySQL database on the application server.

Unlike the private website, the public one only displays the 4 second frequency average in the table format and map gradient format when ‘Table Display’ and ‘Map Gradient’ are clicked on. The script ‘newFreqVals.php’ and ‘cdgradgen.php’ are likewise invoked by a frame replace function in a timer service subroutine every 4 seconds to refresh the frequency average data in the table and the frequency gradient on the map.

The new features added to the public website are ‘Sample Events’, ‘FNET Gallery’, and ‘Contact Us’. The ‘Sample Events’ gives four event analysis examples, which comprise of FNET frequency plot, event location estimation and movie/report link, by which the viewer can get a rough concept of the major researches FNET conducts. The ‘FNET Gallery’ column displays the photographs of FNET professors, students, alumni, and visiting scholars. The ‘Contact us’ column offers the contact information of FNET director Dr. Yilu Liu.

Chapter 3: FNET Frequency Estimation Algorithm and its Evaluation

System frequency is one of the most important parameters of a power system, especially for the monitoring and control of the power system. It is a direct index of the healthiness of the operation of power systems.

It is well known that any unbalance between generation and load is faithfully reflected by a change in the operating frequency. Area frequency can vary over a small range due to generation-load mismatches. When the power consumed by loads and losses is greater than the generated power, the system operating frequency decreases, resulting in a situation classified as ‘the underfrequency condition’. This occurs when there is a loss of one or more generators, or when there is an increase in power demand during certain peak period of the day. When loads are suddenly lost in a system, the input power will be greater than the consumed power. The extra power generated will be absorbed by the rest of the loads and the generator inertia, resulting in an increase in the operating frequency. This is classified as ‘the overfrequency condition’. In either case, a change in frequency can jeopardize the efficiency and safety of the power system and may cause it to collapse.

It is therefore essential to maintain the frequency of a power system close to its nominal value. In well-regulated systems, any significant variation from nominal frequency forces generator governors, under/overfrequency relays, and other devices to take corrective actions intended to bring the frequency back to the nominal values as soon as possible. Frequencies lower than the nominal value indicates that the system is overloaded. Underfrequency relays are used to detect overloaded conditions and disconnect load blocks to restore the frequency to its normal value. These relays provide outputs when the frequency decreases to specified thresholds. The conditions of surplus of power generation are detected by overfrequency relays provided at generator terminals.

Overfrequency relays are also used to protect generators from overspeeding during start-ups. During start-up and shut-down of a generator, its frequency varies over a wide range. If the generator is connected to a transformer, both the generator and the transformer can experience overexcitation. Excessive volts-per-hertz (V/Hz) indicates that overexcitation has occurred. One method of estimating this parameter is to measure the voltage magnitude and frequency and then calculate volts-per-hertz. Volts-per-hertz relays that provide accurate measurements over 10 to 70 Hz range would be desirable for use on steam turbine driven generators.

Frequency information has also been used routinely in the control and management of generation[5]. Fast and accurate real time measurement of frequency is an important component in the operation of the transmission system. Moreover, if the rate of change of frequency is known, more accurate adjustments can be made to quickly restore the system back to its normal operation point[6]. Consequently, accurate and fast estimation of area system frequency is of vital importance to power-system protection-and-control applications.

The problem of determination of accurate frequency in power networks, however, has grown more complex in the recent past decades. There are multiple reasons for this. The dynamic balance between load and generation, a prerequisite for stable power system operation, has become more difficult to maintain because the expansion of the transmission network does not follow the growth of the system. The direct consequence is that the security margins are generally smaller and quite often power systems operate at the brink of instability, possibly resulting in a blackout. Such operating practices imposed by the practical reasons are further aggravated due to the effects of deregulation. Non-utility generation and wheeling may reduce the stability margins of a normally secure system. It is therefore very important for utilities to develop means to monitor and control the dynamics of the power system[7].

In the past forty years, various techniques have been developed to measure power system frequency. Recently, more sophisticated and accurate algorithms are being developed to take advantage of the powerful computing capability of digital computers. Several projects have been launched for wide-area measurement systems in the past 15 years[8][9][10]. The results of all earlier efforts clearly point to the need of a real time, continuous, comprehensive and uniform-measurement system.

FNET, an internet-based real-time Global Positioning System (GPS) synchronized wide-area frequency monitoring network, has been developed at Virginia Tech, which has satisfactorily met the needs and advanced the wide-area measurement technology to a more economical and practical level.

In the FNET system, an algorithm that employs the relationship between phasor angles and deviated frequency[11] calculates both frequency and its rate of change. The accuracy is ± 0.0005 Hz which is already very acceptable, but precise frequency estimation in a wider range is desired. Also, some practical problems which affect the output estimated frequency have occurred within the system that has been set up and running at Virginia Tech. This very chapter covers the topics about the fundamentals of the algorithm, the sources of the error, some practical problems, and how to improve the accuracy of the frequency estimation algorithm used in the FNET system.

3.1 The Frequency Estimation Algorithm of FNET

3.1.1 The Fundamental Theory of the Algorithm

A pure sinusoidal waveform can be expressed as

$$x(t) = \sqrt{2}X\cos(\omega t + \varphi) \qquad (3\text{-}1)$$

Equation (3-1) can be rewritten into (3-2) at the sampling rate of N times per cycle

$$x_k = \sqrt{2}X\cos(\frac{2\pi}{N}k + \varphi) \qquad (3\text{-}2)$$

The Discrete Fourier Transform (DFT) of (3-2) is given by

$$\overline{X} = \frac{1}{\sqrt{2}}(X_c + jX_s) \qquad (3\text{-}3)$$

where $X_c = \frac{2}{N}\sum_{0}^{N-1} X_k \cos\frac{2\pi}{N}k$ and

$$X_c = \frac{2}{N}\sum_{0}^{N-1} X_k \cos\frac{2\pi}{N}k \cdot$$

Next, the recursive phasor computation technique is used to obtain the *r*th phasor, which is given below

$$\overline{X}^{(r)} = \overline{X}^{(r-1)} + j\frac{1}{\sqrt{2}}\frac{2}{N}(x_{N+r} - x_r)\exp\left(-j\frac{2\pi}{N}(r-1)\right) \qquad (3\text{-}4)$$

From (3-4), the phasor of a sinusoidal wave at frequency $f = f_0 + \Delta f$ can be found to be

$$\overrightarrow{X}^{(r)}_{f_0+\Delta f} = \overrightarrow{X}^{(0)}_{f_0}\frac{\sin\left(\frac{N\Delta\omega\Delta t}{2}\right)}{N\sin\left(\frac{\Delta\omega\Delta t}{2}\right)}\exp(jr\Delta\omega\Delta t)\exp\left(j(N-1)\frac{\Delta\omega\Delta t}{2}\right) \tag{3-5}$$
$$+\overrightarrow{X}^{*(0)}_{f_0}\frac{\sin\left(\frac{N(\omega+\omega_0)\Delta t}{2}\right)}{N\sin\left(\frac{(\omega+\omega_0)\Delta t}{2}\right)}\exp\left(-jr(\omega+\omega_0)\Delta t\right)\exp\left(-j\frac{(N-1)(\omega+\omega_0)\Delta t}{2}\right)$$

where $\overrightarrow{X}^{(0)}_{f_0}$ is the initial computation of the phasor at f_0, and Δt is the sampling interval.

For small values of Δf , the following simplifications are introduced[12]

$$\frac{\sin\left(\frac{N(\omega+\omega_0)\Delta t}{2}\right)}{N\sin\left(\frac{(\omega+\omega_0)\Delta t}{2}\right)}\cong 0 \quad \text{and} \quad \frac{\sin\left(\frac{N\Delta\omega\Delta t}{2}\right)}{N\sin\left(\frac{\Delta\omega\Delta t}{2}\right)}\cong 1 \cdot \tag{3-6}$$

Therefore, (3-5) reduces to

$$\overrightarrow{X}^{(r)}_{f_0+\Delta f} = \overrightarrow{X}^{(0)}_{f_0}\exp(jr\Delta\omega\Delta t)\exp\left(j(N-1)\frac{\Delta\omega\Delta t}{2}\right)\cdot \tag{3-7}$$

By denoting the angle of the new phasor as ψ_r and the angle of the previous phasor as ψ_{r-1}, one can approximate the new phasor angle as

$$\psi_r = \frac{\Delta f}{f_0}\frac{2\pi}{N} + \psi_{r-1}\cdot \tag{3-8}$$

Now, using the definition of derivative,

$$\frac{d\psi}{dt} = \lim_{t\to 0}\frac{\psi_r - \psi_{r-1}}{t} \approx \frac{\psi_r - \psi_{r-1}}{{}^{1}\!/_{Nf_0}} = 2\pi\Delta f \cdot \tag{3-9}$$

The system frequency can now be computed with

$$f = f_0 + \Delta f = f_0 + \frac{1}{2\pi}\frac{d\psi}{dt}, \tag{3-10}$$

and the rate of change of frequency is

$$\frac{df}{dt}=\frac{1}{2\pi}\frac{d^2\psi}{dt^2}. \tag{3-11}$$

By computing the phasor, the harmonic noise can be partially eliminated. With the help of a low-pass filter, the high frequency noise does not interfere with the frequency computation. With the help of least error square computation on redundant number of phasor angles, the accuracy of both the frequency computation and the rate of change of frequency computation will be greatly improved. Since the technique is based on sampled phasor data and because most microcomputer-based power-system devices installed already have the ability to compute phasors, the method can be added for low incremental cost to microcomputer-based devices that are performing other functions, without additional hardware.

It is seen that the simplifications of (3-6) are made to obtain the relationship between phasor angle and deviated frequency. This approximation will inadvertently introduce some error in the estimation. There are many ways to alleviate the above approximation error, such as to use the positive sequence voltage waveform as the input instead of the single-phase voltage waveform, which will be discussed in details in the next chapter.

The difference between the ordinary DFT and recursive DFT is illustrated in (3-12) through (2-14).

$$\overline{x}_0=\frac{2}{N}\left(x_0e^{-j0\frac{2\pi}{N}}+x_1e^{-j\frac{2\pi}{N}}+x_2e^{-j2\frac{2\pi}{N}}+\ldots+x_{N-2}e^{-j(N-1)\frac{2\pi}{N}}+x_{N-1}e^{-j(N-1)\frac{2\pi}{N}}\right) \tag{3-12}$$

$$\overline{x}_1=\frac{2}{N}\left(x_1e^{-j0\frac{2\pi}{N}}+x_2e^{-j\frac{2\pi}{N}}+x_3e^{-j2\frac{2\pi}{N}}+\ldots+x_{N-1}e^{-j(N-1)\frac{2\pi}{N}}+x_Ne^{-j(N-1)\frac{2\pi}{N}}\right) \tag{3-13}$$

$$\overline{x}_1=\frac{2}{N}\left(x_1e^{-j\frac{2\pi}{N}}+x_2e^{-j2\frac{2\pi}{N}}+x_3e^{-j3\frac{2\pi}{N}}+\ldots+x_{N-1}e^{-j(N-1)\frac{2\pi}{N}}+x_Ne^{-jN\frac{2\pi}{N}}\right) \tag{3-14}$$

Equation (3-12) is the phasor computation at the sampling point k=0. Equations (3-13) and (3-14) are the phasor computations at the next sampling point for the ordinary DFT and recursive DFT respectively. As it manifests itself very clearly in the equations, every term in ordinary DFT of the current sampling point is different from those of the previous sampling point, whereas most of the terms, which are colored in red, in the recursive DFT are the same as the previous sampling point except the first and the last terms. This implies the computation of the recursive DFT is reduced to only a fraction of that of the ordinary DFT, because the recursive DFT only needs to calculate the last term of (3-14) , add it to the previous phasor result and subtract the first term of (3-12), whereas the ordinary DFT needs to calculate every term and sum them up. However, the result of the recursive DFT is the same as that of the sum of the result of the ordinary DFT plus $\frac{2\pi}{N}$.

The effect the recursive DFT with respect to the ordinary DFT is illustrated in Figure 3.1. The sequence of the powers of the first primitive *n*th root of unity serves as a frame of reference for the sampling array of the input phasor, which means the DFT result is the relative phasor between the sequence of the powers and the sampling array. When the sequence of the powers starts at the positive real axis, the DFT result is the input phasor, whereas when the sequence of the powers starts at the vector whose phase angle is θ, the DFT result is the input phasor with $-\theta$ phase angle displacement. It can then be deduced that if the sequence of the powers is stationary, the frame of the reference is stationary, and that if the sequence of the powers keeps increasing, the frame of reference is rotating.

As can be seen from (3-12) through (3-14), the sequence of the powers remains from 0 to $N-1$ for the ordinary DFT but incrementally changes for the recursive DFT, causing the frame of reference to stay at the positive real axis for the ordinary DFT but to rotate at angular velocity of $2\pi f_0$ for the recursive DFT. Due to that effect, the phase angle of the ordinary DFT resultant phasor rotates at an angular velocity of $2\pi ft$ as the sampling point increments, whereas the phase angle of the recursive DFT resultant phasor rotates at an angular velocity of $2\pi(f-f_0)$, which is shown in Figure 3.1.

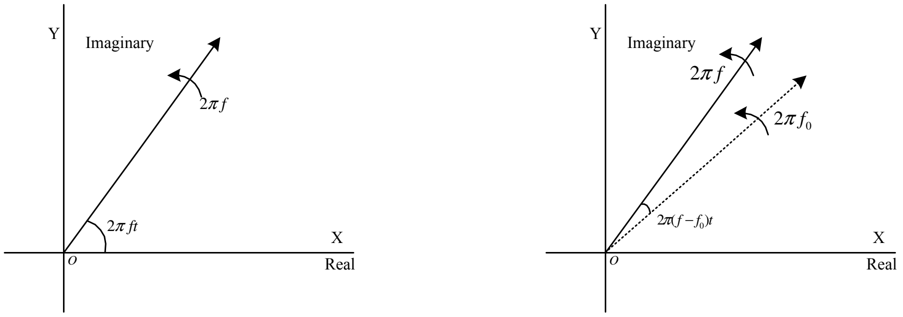


Fig. 3.1 The Illustration of the Effect of the Recursive DFT

3.1.2 Technical Details of the Algorithm

Assuming there are N sampling points $\{x_k\}, k \in [1, N]$ per cycle of fundamental frequency, the phasor of fundamental frequency component is:

$$\overline{X}_1 = \frac{1}{\sqrt{2}}\left(\frac{2}{N}\sum_{k=1}^{N} x_k \cos\left(\frac{2\pi}{N}k\right) - j\frac{2}{N}\sum_{k=1}^{N} x_k \sin\left(\frac{2\pi}{N}k\right)\right) \qquad (3\text{-}15)$$

Denoting $\frac{2}{N}\sum_{k=1}^{N} x_k \cos\left(\frac{2\pi}{N}k\right)$ by $X_c^{(1)}$ and $-\frac{2}{N}\sum_{k=1}^{N} x_k \sin\left(\frac{2\pi}{N}k\right)$ by $X_s^{(1)}$ and using the recursive formula derived in [11], each successive phasor can be calculated with

$$\begin{aligned} X_c^{(k+1)} &= X_c^{(k)} + \frac{2}{N}\left(x_{k+1} - x_{k+1-N}\right)\cos\left(\frac{2\pi}{N}k\right) \\ X_s^{(k+1)} &= X_s^{(k)} - \frac{2}{N}\left(x_{k+1} - x_{k+1-N}\right)\sin\left(\frac{2\pi}{N}k\right) \end{aligned} \qquad (3\text{-}16)$$

Hence, a new phasor is available for every incoming sample after initialization. The angle of *k*th phasor is given by:

$$\phi(k) = \tan^{-1}\frac{-X_s^{(k)}}{X_c^{(k)}} \qquad (3\text{-}17)$$

Assuming the phasor angle vary as polynomial function with respect to the sampling number. Detail analysis in [11] shows that the linear function does not have enough accuracy while cubic function does not improve accuracy much but carrying on heavy computation burden. Therefore the quadratic function is used:

$$\phi(k) = a_0 + a_1 k + a_2 k^2 \qquad (3\text{-}18)$$

Considering the computation window to be M to estimate the coefficients $a_k, k = 0,1,2$:

$$\begin{aligned} \phi(1) &= a_0 + a_1 1 + a_2 1^2 \\ \phi(2) &= a_0 + a_1 2 + a_2 2^2 \\ \vdots \quad & \quad \vdots \quad \vdots \\ \phi(M) &= a_0 + a_1 M + a_2 M^2 \end{aligned} \qquad (3\text{-}19)$$

Rewrite (3-19) in matrix form:

$$\begin{bmatrix} \phi(1) \\ \phi(2) \\ \vdots \\ \phi(M) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1^2 \\ 1 & 2 & 2^2 \\ \vdots & \vdots & \vdots \\ 1 & M & M^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \qquad (3\text{-}20)$$

or $\Phi = \mathbf{Xa}$ (3-21)

The unknown coefficient matrix $\mathbf{a}$ can be estimated by the least square error method:

$$\mathbf{a} = [\mathbf{X}^T\mathbf{X}]^{-1}\mathbf{X}^T\Phi \qquad (3\text{-}22)$$

Here, the pseudo-inverse matrix $\left[\mathbf{X}^T\mathbf{X}\right]^{-1}\mathbf{X}^T$ is known as the gain matrix and can be computed off-line. One thing has to be remembered is that since the least error square approximation is actually doing a curve fit, all the phasor angles, ϕ s, must be made monotonic, i.e., there should not be any wrap-around of angles. This can be easily done by appropriately adding $\pm 2\pi$. Also, to keep the curve fitting uniform in all intervals and the arithmetic errors at the minimum, it is best to make ϕ_1 to be zero. Hence, ϕ_1 is subtracted from all subsequent phasor angles in the computation window.

Once the values of a_1 and a_2 are known, one can proceed to compute the frequency and the rate of change of frequency.

Taking the derivative of (3-18) with respect to k

$$\frac{d\phi}{dk} = a_1 + 2a_2k \qquad (3\text{-}23)$$

The relationship between sample number and time can be expressed as

$$k = Nf_0t \text{ and } \frac{dk}{dt} = Nf_0 \qquad (3\text{-}24)$$

where N is the number of phasor angles used in the computation per cycle and f_0 is the nominal frequency.

Using Chain Rule,

$$\frac{d\phi}{dt} = \frac{dt}{dk}\frac{d\phi}{dt} = Nf_0\frac{d\phi}{dk} = Nf_0\left(a_1 + 2a_2k\right) \text{ or}$$

$$\frac{d\phi}{dt} = Nf_0\left(a_1 + 2a_2Nf_0t\right) \tag{3-25}$$

In previous section, it is shown that $\frac{d\phi}{dt} \approx \Delta\omega$. Hence

$$\Delta\omega \approx Nf_0\left(a_1 + 2a_2Nf_0t\right) \Rightarrow \Delta f \approx \frac{1}{2\pi}Nf_0\left(a_1 + 2a_2Nf_0t\right) \tag{3-26}$$

where t determines which instant inside the computation window the computed frequency corresponds to.

Now,

$$\frac{d\Delta\omega}{dt} = 2\pi\frac{d\Delta f}{dt} = 2\pi\frac{df}{dt} \approx \frac{d^2\phi}{dt^2} = \frac{d}{dt}\left(Nf_0a_1 + 2\left(Nf_0\right)^2 a_2t\right) = 2\left(Nf_0\right)^2 a_2 \Rightarrow$$

$$\frac{df}{dt} \approx \frac{1}{2\pi}2\left(Nf_0\right)^2 a_2 \tag{3-27}$$

Equations (3-26) and (3-27) are the main formulas used for calculating the frequency and the rate of change of frequency. It is seen from (3-27) that the rate of change of frequency is assumed to be constant within the computation window. It has to be kept in mind that the frequency estimation obtained at the end of the computation window is actually the frequency at the half cycle point before the end of the window. This is because the phasor computed at the end of one cycle actually represents the phasor at the center. The frequency and the rate of change of frequency estimations obtained above are already rather accurate. However, since more precise frequency estimation is needed, one more step is required.

It has been shown earlier that the frequency estimation will be more accurate when the actual frequency approaches the frequency established for the sampling rate. Hence, resampling the waveform with the estimated frequency and using the new phasors to perform another corrective estimation will be a rather attractive solution to achieve precise measurement.

To explain the idea of resampling in details, assume the nominal frequency is 60 Hz and the sampling rate is 1440 samples per second, which corresponds to $\frac{1440}{60} = 24$ samples per cycle. When the frequency has changed to 55 Hz, each cycle will now have $\frac{1440}{55} = 26.18$ samples. By re-normalizing through resampling, one tries to do a mathematical interpolation so that there will always be 24 samples per cycle regardless of the waveform frequency.

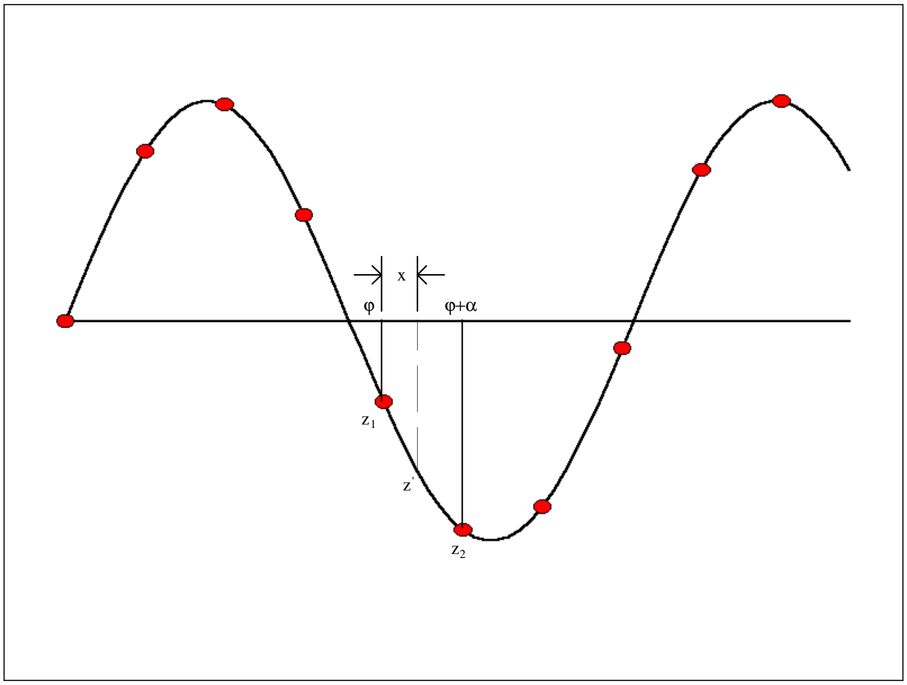


Fig. 3.2 Illustration of the Resampling Technique

Refer to Figure 3.2 for the resampling algorithm, the values of z_1 and z_2 can be represented by the following two equations

$$z_1 = Z_m \sin(\varphi) \qquad (3\text{-}28)$$

$$z_2 = Z_m \sin(\varphi + \alpha) = Z_m \sin\varphi \cos\alpha + Z_m \cos\varphi \sin\alpha \qquad (3\text{-}29)$$

where Z_m is the amplitude of the waveform,

φ is a sample instant and is an arbitrary known value,

$\alpha = 2\pi f_{new} / (Nf_0)$ is the interval between two samples at the new frequency.

Combining (3-28) and (3-29) results in

$$Z_m \cos\varphi = \frac{(z_2 - z_1 \cos\alpha)}{\sin\alpha} \qquad (3\text{-}30)$$

Let x be the fractional distance between z_1 and z_2, the re-sampled point z' is then given by

$$\begin{aligned} z' &= Z_m \sin(\varphi + x\alpha) \\ &= Z_m \sin\varphi \cos x\alpha + Z_m \cos\varphi \sin x\alpha \qquad (3\text{-}31) \\ &= z_1 \cos x\alpha + (z_2 - z_1 \cos\alpha) \frac{\sin x\alpha}{\sin\alpha} \end{aligned}$$

After resampling points have been found, the phasor angles of the re-sampled data are computed and another estimation using (3-22), (3-25) and (3-27) is made to obtain the corrective frequency $\Delta f'$, and the final rate of change of frequency measurements.

The rate of change of frequency computed here is the final estimation of the rate. The final frequency estimation is computed using

$$f_{final} = f_0 + \Delta f + \Delta f' \qquad (3\text{-}32)$$

The number of cycles employed to estimate the frequency is set to be 6 nominal cycles with the sampling rate being 24 points per second. After initialization, each new phasor is calculated every $\frac{1}{24 \times 60} = 0.6944$ millisecond. However, only 4 phase angles per cycle (i.e., $4 \times 6 = 24$ phase angles for the frequency computation window) are used to compute the coefficients a_0, a_1 and a_2, and then the frequency. The sliding window is 8 cycles long with 6 cycles for estimating frequency. The reason to use 8 cycle sliding window is:

1) It takes one cycle of sampled data to compute the first phasor value.

2) When frequency is less than 60 Hz, we need more data to satisfy the 6 cycle criterion for resampling. Assuming f = 55 Hz, the number of samples needed to compute the frequency becomes $24 \times 6 \times \frac{60}{55} \approx 158$ samples. Therefore, the number of cycles in 60 Hz changes to

$\frac{158}{24} \approx 7$ cycles.

Another issue is about the treatment of the term *t* in (3-25), which is not a problem when the frequency does not change or changes very slowly in that a_2 is zero or negligible if the rate of change of frequency is zero or very small. However, it does cause some error to the frequency estimation when the power system undergoes drastic change and the system frequency deviates from the nominal value at a great extent, such as large inter-area oscillation and islanding. In those situations, it is of great practical importance to designate an appropriate *t* value. For the FNET applications, $C = 3N$ and $C = \left(\left(N\left(\frac{6}{2} - \frac{1}{2}\right) - \left(6N - 6N\frac{f_0}{f_r}\right)\right)\frac{f_r}{f_0} + \frac{N}{2}\right)$ are adopted to address this issue in the frequency estimation $\Delta f' \approx \frac{1}{2\pi} N f_r \left(a_1 + 2a_2 C\right)$ for the first frequency estimation and the frequency estimation after resampling, and an adequately accurate result can be achieved by this means.

3.2 Mathematical Examination and Simulation Results

The importance of accurate measurements of frequency is articulated in the beginning of this chapter. The fundamental theory and technical details of the frequency algorithm employed to calculate the frequency are presented afterwards. The performance of the algorithm then becomes a significant issue needed to be well tended to.

In order to make certain an accurate evaluation of the algorithm so that a firm foundation of data validity can be laid for the applications based on FNET frequency measurements, this sub-chapter thoroughly checks, examines and exemplifies the precision of the algorithm and the behaviors of the frequency measurements under different circumstances that correspond to some power system unusual operations. Tools such as VC++ and MATLAB are used to carry out the research.

3.2.1 Tests with Constant Frequency Waveform

1) Algorithm Precision Estimation

A program consisting of separate modules is designed to check the computational result of the input with specified frequencies in the interest of making sure the genuine precision of the algorithm. The program itself generates output similar to the ADC sampling, and another module reads in the data with which it then does computational work to examine the accuracy of the algorithm by comparison between the input frequency and the calculation result.

After the computation is done, the measured frequency value will be saved into a file, from which we can obtain data to make the estimated FNET frequency algorithm precision table shown below:

Δf	Final Frequency(Hz)	First Frequency(Hz)	Input contains the 3rd harmonic
0.01Hz	7.88e-8	4.47e-6	2.48e-6
0.1Hz	1.72e-6	1.20e-4	1.38e-5
1.0Hz	1.40e-4	1.40e-3	3.50e-4

Tab. 3.1 The Estimated Precision of the FNET Frequency Algorithm

It can be concluded from the table that when measuring pure sinusoidal waveforms with constant frequency between 59.0Hz and 61.0Hz, the error is less than 0.14 mHz, which satisfactorily meets the demand for the majority of applications in power systems. More importantly, the final frequency is nearly 100 times more precise than first frequency, which demonstrates that re-sampling and recalculating process plays an indispensable role in this algorithm.

2) The Effect of the Harmonics and the Noises on the Algorithm

Real power system voltages are not pure sinusoidal wave. There are many harmonics and noises existing in power systems, which would make this frequency algorithm less precise. It, therefore, bears a significant meaning to learn about how harmonics and noises affect this algorithm.

Power system noises can be viewed as being composed of white noise and spike noise. Both spike noise and white noise can be essentially filtered by analog filters. However, regardless of how well-designed the filters are, there are always residues after the filters. The passed-through amount of the noises can impact the computation result to an appreciable degree. Therefore, it is necessary to look into how the white noise and spike noise act upon the algorithm.

Harmonic noise notoriously has been a major challenge to most power system measurement systems. Low order harmonics will not be filtered by analog filters and they are usually present in the waveform entering the measurement device. In this section, simulations are also performed on waveforms experiencing harmonic distortions while maintaining a constant operating frequency.

The first simulation case to be studies is the effect of the noises on the algorithm. Ten percent Additive White Gaussian Noise (AWGN) is added to the simulative samples provided by the VC++ signal generating module. The waveform after noise addition is displayed in the left image of Figure 3.3. The algorithm examining module then uses the data to calculate the frequencies at different instants and plot them with the time stamps of the instants in the right image of Figure 3.3.

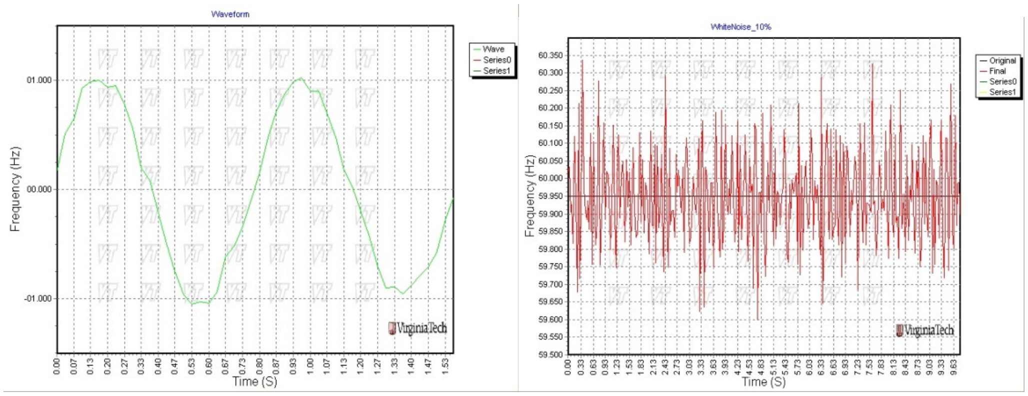


Fig. 3.3 Waveform and Frequency Output of 10% AWGN

The frequency output plot in the right image of Figure 3.3 appears to be very chaotic and randomly around the 59.95 Hz preset input frequency line, which can be deemed as the result of the randomness attribute of the added noise. The error compared to the true input drifts along the time axis at the range nearly as large as from -40 mHz to 40 mHz without any recognizable pattern.

To further the research about the effect of noises and harmonics, the pure sinusoidal signal is added the 3rd order and the 5th order harmonics as opposed to white noise, and the same procedures are followed to come up with the same graphs in comparison with the results given above. The waveforms of the input signal are demonstrated in the left images of Figure 3.4 and Figure 3.5. The output of the frequency calculating and plotting module generates similar frequency plots in the right images of Figure 3.4 and Figure 3.5. The frequency of the signal to be measured is still set to be 59.95 Hz.

As can be observed from the figures, the distortion of the input signals are much greater than that of the input coupled with the AWGN, whereas the errors caused by the interposition of the harmonics are far less than those of the AWGN. If put under the same plotting scale, the error from the harmonics are manifesting itself as a straight line in the two plots without any noticeable deviation from the horizontal line of the preset input frequency, no matter if it is the 3rd or the 5th harmonic. The scale of the frequency output plots is 100 mHz from top to bottom on the graphs.

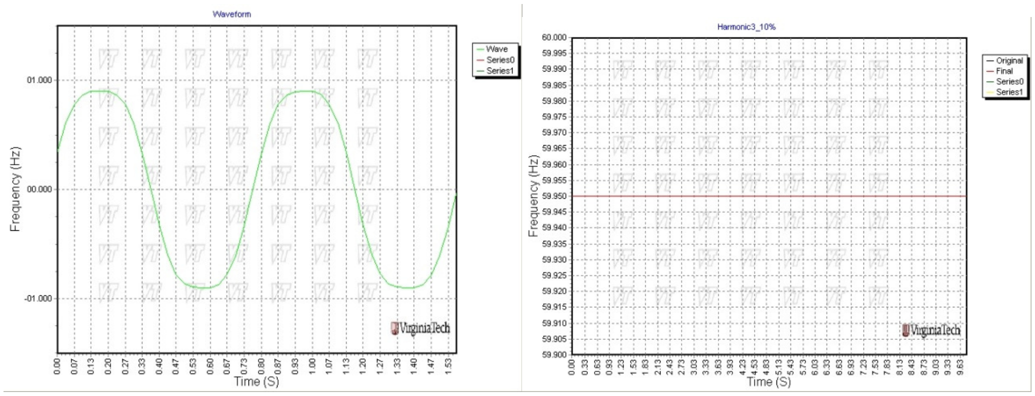


Fig. 3.4 Waveform and Frequency Output of 10% 3rd Harmonic

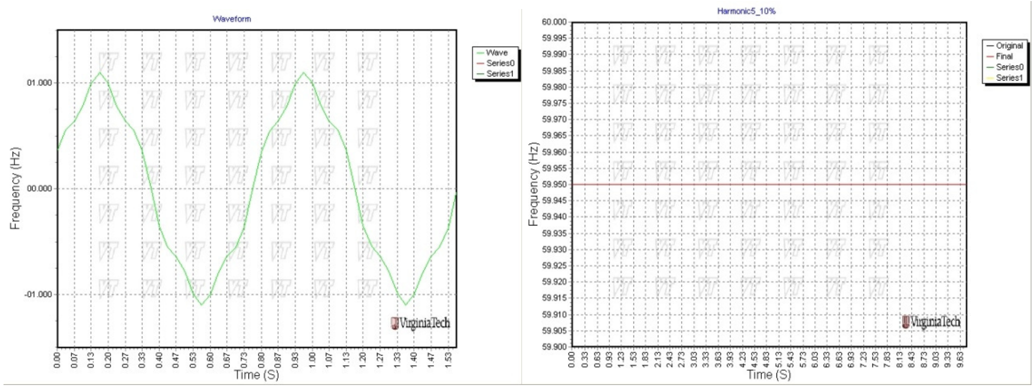


Fig. 3.5 Waveform and Frequency Output of 10% 5rd Harmonic

For purpose of getting a closer look at the behavior of the errors from the harmonics, the frequency data generated by the algorithm after the input being added harmonics in the previous tests are retrieved and plotted in Figure 3.6, the scales of which are magnified 1000 times. In those close-up graphs, it can be seen that the straight lines in the previous graphs become periodic waves and the peak-to-peak magnitude of the waves are less than 1.0 mHz.

A number of experiments are also carried out to supplement the investigation, including the ones that are added only 1% noise and harmonics. On the basis of all those data, Table 3.2 is made to give a quantitative concept of the effect of the noises and harmonics on the algorithm.

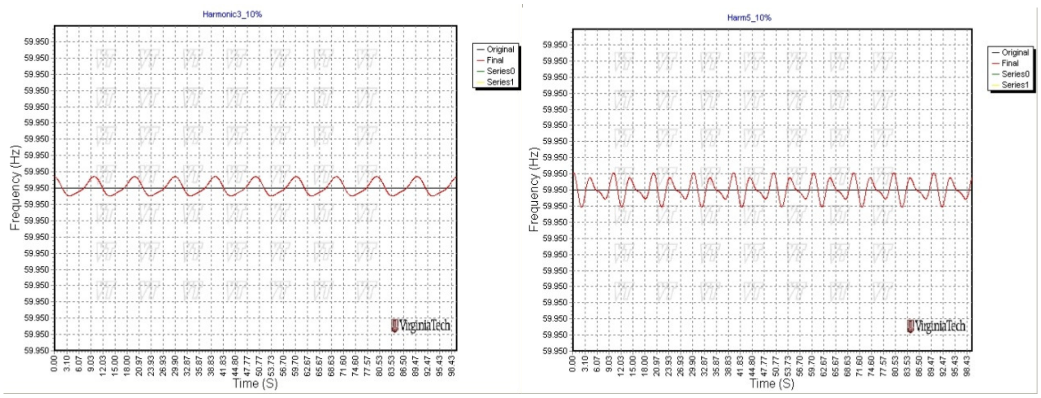


Fig. 3.6 Frequency Plots of the 3rd and 5th Harmonics (magnified 1000 Times)

	Maximum Error (10% noise)	Maximum Error (1% noise)
AWGN Noise	387mHz	39.1mHz
The 3rd harmonic	310μHz	29.2μHz

Tab. 3.2 The Effect of the Noises and Harmonics on the Frequency Algorithm

From the figures and table above, it is evident that random white noise plays a far more important role in affecting this algorithm than the harmonics mentioned here. It is estimated from the number that the error caused by random noise is over 1,000 times larger than that of the 3rd harmonic. This phenomenon results from the orthogonality property of the trigonometric functions, which states that the integral of product of the various two dimensional trigonometric functions, such as $\sin(mx)*\sin(nx)$, $\sin(mx)*\cos(nx)$, $\cos(mx)*\sin(nx)$, $\cos(mx)*\cos(nx)$ is zero when x runs from 0 to 2π. When this property is applied to the DFT algorithm, it denotes that the DFT result of any harmonic is supposed to be zero when the frequency of the fundamental wave is exactly 60 Hz, which implies the superimposed harmonics would have no impact on the DFT result of the fundamental wave. In the presence of a frequency deviation of the fundamental wave, the DFT results of the harmonics are affected by the deviation to some extent, and the slight inaccuracy is then passed on to the estimation of the frequency, as the foregoing simulations exhibit. However, the effect of the harmonics is negligibly small compared to hardware limitations when it comes to practical applications.

Therefore, because of the severe discrepancy between effects of the white noise and the harmonics on the performance of the frequency algorithm, more effort should be made to decrease the random noise when it comes to error-shooting.

In the FNET applications, a certain accuracy level is required for the system to work properly and to meet some industrial standard. Many a simulation is done to find out the largest amounts of the noise or harmonics that can be tolerated by the algorithm before the errors of the results surpass the required level. The empirical parameter for the accuracy requirement in FNET is 0.5 mHz.

As a summary of the simulations undertaken, the results are listed in Table 3.3 below. The tested frequencies are from 53 Hz to 67 Hz, which is greater than the greatest frequency deviation range ever recorded. The column ‘pure wave’ is the list of the simulation results when there is no additive noise and harmonics mixed into the fundamental sinusoidal waveform. The column ‘SNR’ is the list of the tolerances of signal noise ratio in dB to satisfy the maximum error of the 0.5 mHz if AWGN is added. The last four columns are the tolerances of the harmonics of different orders in terms of satisfying the maximum error requirement of 0.5mHz, given by the ratios of the harmonic amount to the total magnitude of the fundamental wave.

Input Frequency	Pure Wave	SNR (db)	Ratio of 2^{nd} Harm	Ratio of 3^{nd} Harm	Ratio of 5^{nd} Harm	Ratio of 9^{nd} Harm
53	0.000242	43	0.0065	0.0236	0.0029	0.0053
54	0.000147	42				
55	0.000142	40	0.0943	0.0029	0.0065	0.0224
56	9.70E-05	39				
57	4.40E-05	38	0.0118	0.0177	0.0118	0.0206
58	1.30E-05	37				
59	2.00E-06	36.5	0.0884	0.053	0.0707	0.1002
59.2	1.00E-06	36.4	0.277	0.2357	0.1945	0.1355
59.4	0	36.3	0.053	0.1296	0.1061	0.1237
59.6	0	36.2	0.4773	1.0312	1.3553	1.0312
59.8	0	36.15	0.3595	0.4007	0.4537	0.3536
60	0	36.1				
60.2	0	36.3	0.3712	0.4361	0.442	0.3418
60.4	0	36.5	0.5009	0.8544	0.9075	0.607
60.6	1.00E-06	36.8	0.0707	0.1414	0.1473	0.165
60.8	1.00E-06	37.2	0.2475	0.2829	0.4243	0.1061
61	2.00E-06	37.4	0.0766	0.0884	0.0884	0.0707
62	1.10E-05	39				
63	3.70E-05	40	0.0136	0.0177	0.0236	0.0318
64	7.90E-05	41				
65	0.000125	43	0.1296	0.0035	0.0041	0.0295
66	0.000143	44				
67	0.000136	45	0.013	0.0071	0.0147	0.0106

Tab. 3.3 Tolerance of Noise and Harmonics with the Error Requirement of Less Than 0.5 mHz

The third column about the SNR of the AWGN to satisfy the 0.5 mHz requirement is also plotted in Figure 3.7 to render a more visualized notion of the effect of AWGN on the performance of the algorithm.

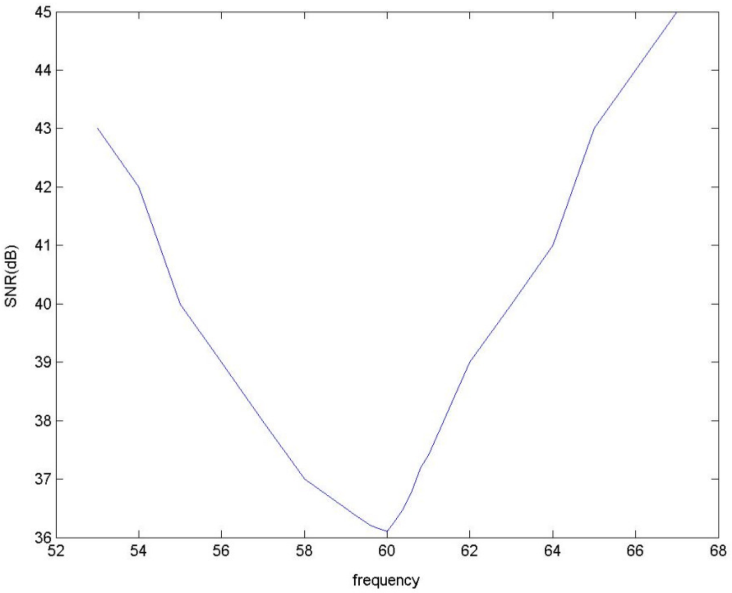


Fig. 3.7 SNR of AWGN Plot

3.2.2 Dynamic Response of the Algorithm

Dynamic response is a crucial part of an algorithm in that the most important information usually exists in the transient period. In this section, the dynamic responses of this algorithm will be discussed briefly.

1) The Responses to Frequency Jumps

Four plots of the frequency responses to abrupt input frequency jumps are presented in Figure 3.8 and Figure 3.9 that are generated by the VC++ program.

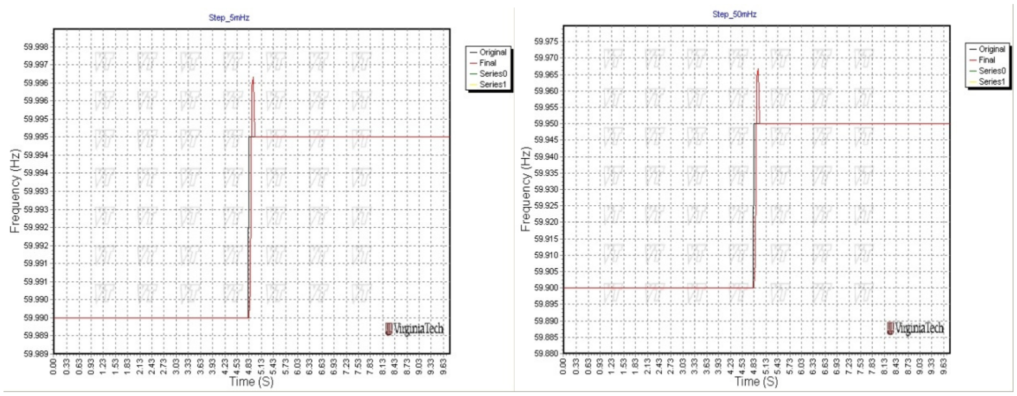


Fig. 3.8 Response to 5mHz and 50 mHz Positive Frequency Jump

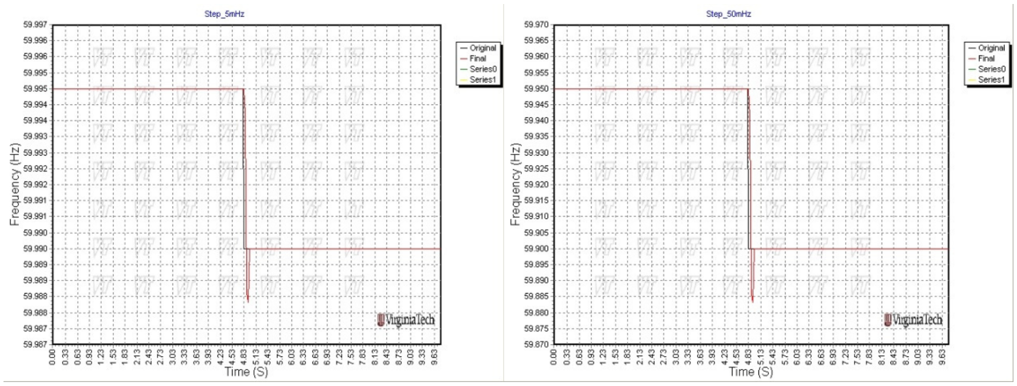


Fig. 3.9 Response to 5mHz and 50 mHz Negative Frequency Jump

In summary of the above 2 figures, Table 3.4 is made to recapitulate the results.

Jump Size	5mHz	50mHz	-5mHz	-50mHz
Error Rate	33.4%	33.5%	33.5%	33.7%

Tab. 3.4 Step Response Maximum Error Table

It can be observed that the error rate, which is the rate of maximum error and the absolute frequency jump value, nearly remains a constant, which is around one third.

2) The Responses to Constant Rate of Change of Frequency

The module that generates simulative samples to feed into the algorithm testing module is set to create a data file containing voltage samples with a constant rate of change of frequency. Thus, the effect of waveform with a constant rate of change of frequency can be studied by the comparison between the frequency output of the algorithm test module and the exact frequency input of the same instant. Errors arising from the rate of change of frequency are examined in detail, and the plots of the preset and the estimation of the frequency are shown in Figure 3.10 through Figure 3.13.

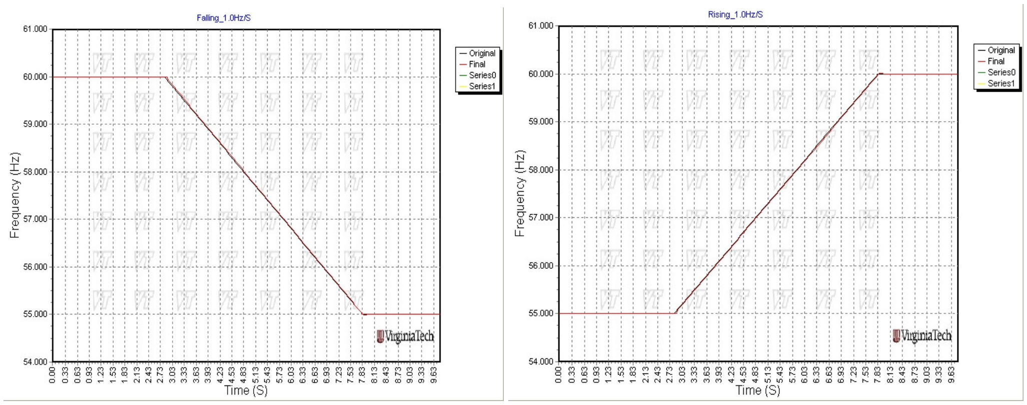


Fig. 3.10 Response to 1.0 Hz/s Frequency Falling and Rising

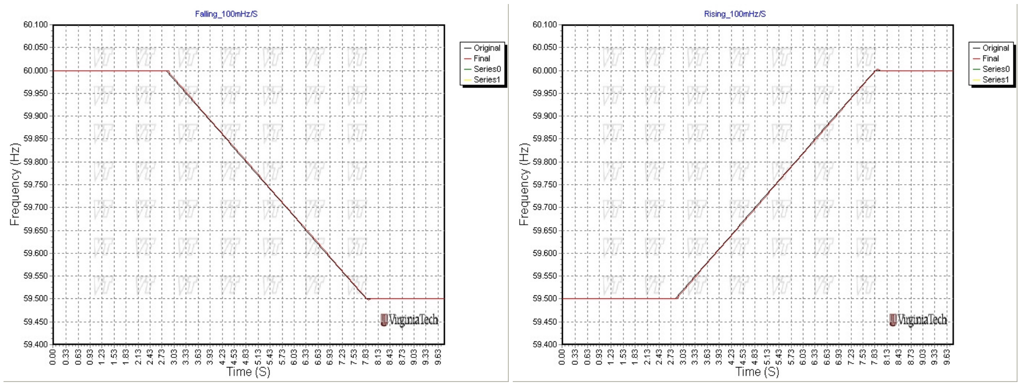


Fig. 3.11 Response to 100 mHz/s Frequency Falling and Rising

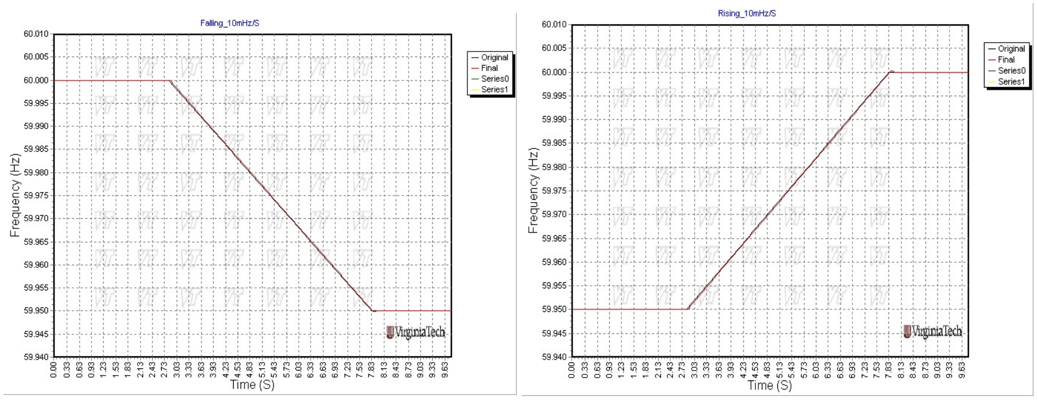


Fig. 3.12 Response to 10 mHz/s Frequency Falling and Rising

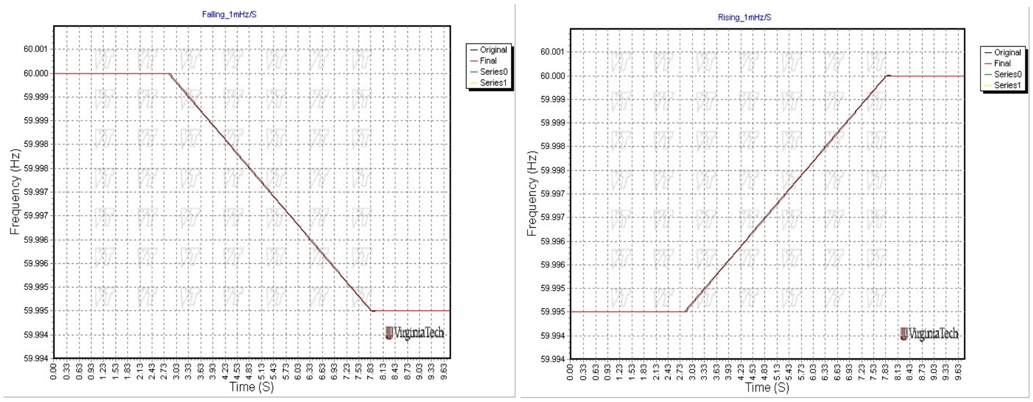


Fig. 3.13 Response to 1 mHz/s Frequency Falling and Rising

From the charts enumerated above, it can be seen that the algorithm handles the input with a constant rate of change of frequency at a reasonable range in power systems sufficiently well, since the black line of the input and the red line of the output are almost overlapping when the rate of change of frequency runs the whole gamut of reasonable values from 1.0mHz/s to 1.0Hz/s.

3) The Responses to Frequency Oscillations

Other than experiencing a constant rate of change of frequency, a power system also occasionally experiences frequency swings. During quasi-steady state operation, the swing is gradual and mild. In this subsection, the algorithm is examined against a waveform experiencing frequency swings. To generate the input waveform with frequency swing, the following equation is used

$$v(t) = V \sin\left(2\pi\left(f_1 t - \frac{A_{osci}\cos(2\pi f_{osci} t)}{2\pi f_{osci}}\right)\right) \tag{3-33}$$

where V is the amplitude of the waveform

f_1 is operating frequency on which the oscillation occurs

A_{osci} is the magnitude of the frequency oscillation

f_{osci} is the frequency of the frequency oscillation.

The frequency deviation is a function of time and is given by

$$\Delta f(t) = f(t) - f_0 = f_1 - f_0 + A_{osci}\sin(2\pi f_{osci} t) \tag{3-34}$$

In the following simulations, A_{osci} and f_{osci} are set to be different values to find out to what extent the two factors affect the performance of the algorithm and which factor contributes more to the computational error.

The first instance sets $A_{osci} = 5mHz$ and $A_{osci} = 50mHz$ with f_{osci} being constant at the value $2Hz$. The results of the two different sets of parameters are plotted in Figure 3.14, and the two graphs are nearly identical, with the difference beyond the resolving power of human eyes. It can be inferred that the magnitude of the oscillation does not contribute to the phase angle delay and amplitude change of the measurement in that the frequency measurement does not vary appreciably when the inputs differ at an order of magnitude.

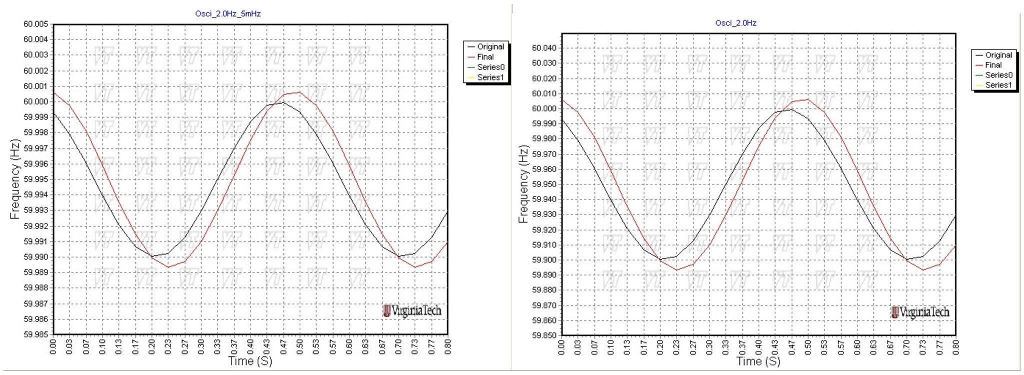


Fig. 3.14 Responses to the Oscillations of Amp = 5mHz and 50 mHz, Freq = 2Hz

What comes next is the impact of the frequency of frequency oscillation of the input signal on the measurement result of the algorithm. The parameter A_{osci} is set to be $50mHz$ as a constant and f_{osci} is set to be $1.0Hz$, $0.5Hz$ and $0.2Hz$ respectively to investigate how those different frequency settings act upon the computation of the algorithm. Those three values together with the diagram of the previous simulation can form a clear picture of the impact in Figure 3.15 and Figure 3.16.

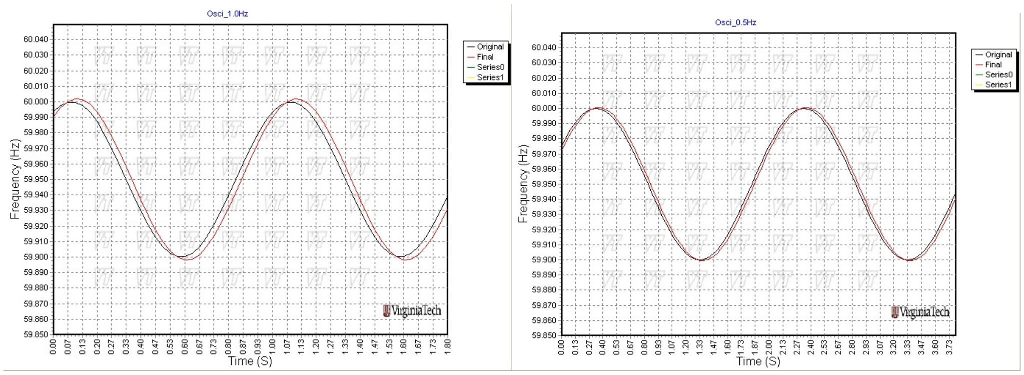


Fig. 3.15 Responses to the Oscillations of Amp = 50mHz, Freq = 1.0Hz and 0.5Hz

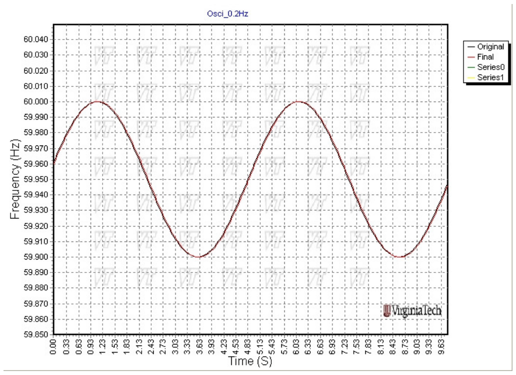


Fig. 3.16 Response to the Oscillation of Amp = 5mHz, Freq = 0.2Hz

It is evident in the graphs above that the frequency of the oscillation conspicuously affects the performance of the frequency algorithm, dissimilar to the effect of the magnitude of the oscillation. When the frequency is adjusted to 1.0 Hz, 0.5 Hz and 0.2 Hz, the estimated frequency is more and more closely approaching the plot of the input frequencies. The speed of the estimation of the algorithm approaching the actual input is such that when the frequency of the oscillation is 2.0 Hz, the discrepancy between the estimation and the true values are too large to consider the results of the algorithm suitable for application if without proper compensation, whereas when the frequency of the oscillation is set to be 0.2 Hz, the two plots almost overlap, which means the estimation of the algorithm is so close to the actual values that the error reaches a negligible level at this oscillation frequency. Noting that from 0.2 Hz to 0.4 Hz is the most common frequency range of the power system oscillations and that the data rate of the FNET system is 10 points per second, we may then assert that the precision of the algorithm under oscillatory circumstances is sufficient for power system applications even when the system is undergoing oscillations within reasonable range.

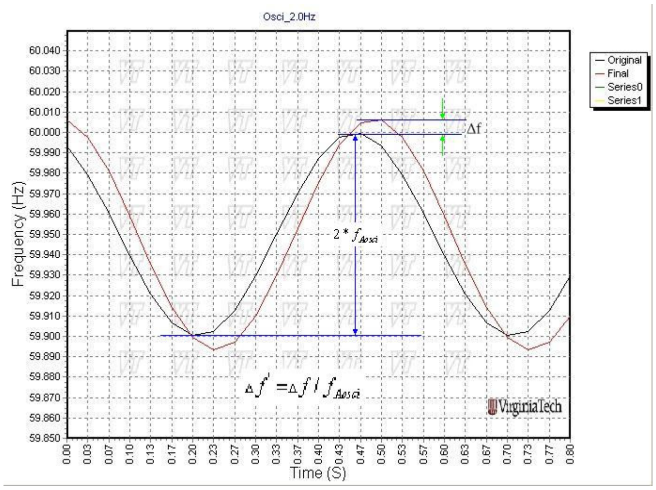


Fig. 3.17 The Definition of the Waveform Gain

A table is produced based on the data collected from the aforementioned cases and the graphs listed above. The definition of waveform gain index Δf is explained in Figure 3.17. In the graph, the black line is the input frequency trajectory and the red line is the plot of the estimated frequencies by the algorithm. There is a magnitude gain present when the input frequency is oscillating. The parameter to evaluate the gain is the waveform gain defined as the ratio of the amount of the magnitude in excess of the actual amplitude of the frequency oscillation compared to the true amplitude of the oscillation.

Oscillation Fre	2.0Hz/s	1.0Hz/s	0.5Hz/s	0.2Hz/s
$\Delta f^{'}$	11.9%	3.5%	0.9%	0.15%

Tab. 3.5 Waveform Gain Statistics

From the graphs and Table 3.5 above, it is obvious that the ratio of the waveform magnification has nothing to do with the oscillation amplitude, whereas it is roughly a quadratic function of the oscillation frequency. And it can be asserted that if the oscillation frequency is less than 1.0Hz/s, the waveform mis-estimation ratio will be less than 3.5%, which fundamentally meets the requirements of power system real-time applications.

3.2.3 Conclusions

This subsection briefly addresses the frequency measurement algorithm used by FNET. Certain analysis to its precision, static properties and dynamic responses are given with concrete simulations and instances. Many intuitive charts and tables are provided to illustrate following points:

1) The accuracy of FNET frequency measurement algorithm is testified to be less than 0.1mHz when the input frequency is within the range between 59Hz to 61Hz and the input waveform is pure sinusoidal.

2) Random noise affects the accuracy of the algorithm the most, while harmonics are nearly negligible when the same amount of noise is superimposed to the fundamental sinusoidal wave. Numerically, the same magnitude of random noise will bring over 1,000 times greater error than harmonics.

3) When a frequency jump occurs, the measurement approximately brings about an error of around one third of the frequency jump.

4) This algorithm does very well in dealing with the smoothly rising and falling

frequency changes, which is the frequency change at a constant rate.

5) The phase shift and amplitude change of the measurements by the algorithm in time of frequency oscillation is predominantly related to the oscillation frequency, instead of the oscillation magnitude. If the oscillation frequency is less than 0.2Hz/s, the distortion will be less than 0.15% of the oscillation magnitude.

Chapter 4: FNET Phase Angle Estimation Algorithm and its Enhancement

Computation of phase angles using the discrete Fourier transform (DFT) algorithm has been proven to be sufficiently accurate in a variety of power system applications when the input signal is three-phase voltage and the voltage frequency is near the nominal value. However, the performance of the DFT algorithm in time of the singe-phase voltage input with a large frequency excursion so far has not been well explored, much less documented. This chapter briefly discusses the importance of the synchronized phase angle measurement in power systems, and brings up the necessity of research on single-phase measurement systems. The performance of the DFT algorithm in power system phase angle measurements is thoroughly analyzed. Corresponding rectifying methods, including quasi-positive-sequence DFT and offset compensation, are proposed and examined by relevant numerical simulations and necessary laboratory tests. The results of the simulations and tests show a perfect match between the derived equations and the phase angle data, and thus verify the effectiveness of the proposed algorithm, which is believed to be able to greatly improve the performance of DFT in the aspect of phase angle calculation when severe power system upsets with remarkable frequency excursions are present.

4.1 Introduction

The phase angle, as the decisive indication of a phasor, is a vital parameter of the power system. The voltage phase angles at different locations in a power system are important variables for describing the state of the system, monitoring the operation of the system, and a number of other functions[13]. It has been recognized that phasors are applicable even when the waveforms are changing rather rapidly and contain significant amounts of transient components[14]. It therefore has been an ongoing endeavor for scientists and engineers to simultaneously take measurements of the phasors, especially the phase angles, all around the whole interconnected power grid at particular instants of time in order to advance the research about the safety, dependability and stability of power systems to a broader dimensionality.

Not until the last two decades did the development of the synchronized phasor measurement technique begin. This was followed closely by widespread application to power systems. The shift occurred when time synchronization with the Global Positioning Satellite (GPS) system became as reliable and as precise as needed by most applications in power system engineering, and digital signal processing techniques could provide accurate measurements of voltages and currents, from which various control and protection functions would flow[15]. It seems clear that the synchronized phasor measurement technique offers a unique capability for tracking system dynamic phenomena in real-time, and offers the possibility of enhancing monitoring, protection, and control functions for the electric power system[14].

Most applications and measuring devices developed so far are based on positive sequence voltages or currents, and usually have three-phase inputs sampled and fed into the microprocessors. The research of three-phase phasor measurements appear in the technical literature from time to time. While at the same time, there are some power system research groups, such as Frequency Monitoring Network (FNET), that only utilize one phase to obtain critical electric variables, and also there are some situations where measurements of single-phase parameters are particularly required, or where three-phase information is not easily obtainable or not accessible at all[16]. In those circumstances, highly sophisticated single-phase measurement procedures need to be researched, discussed, and examined. No highly satisfactory practical method for measuring voltage phasors in single-phase applications and devices has yet been proposed.

In FNET, useful and thorough research about single-phase frequency and phasor measurements has been conducted in the past several years. Certain aspects of the single-phase phasor measurement algorithm will be discussed in the following sections. First, theoretical analyses of the single-phase phasor measurement technique in terms of amplitude and phase angle are presented as the basis of the methods that are then proposed to correct the theoretical errors caused by frequency excursions and the single-phase constraint. After that, necessary simulations are carried out and are shown to verify the proposed method. Lastly, a laboratory experiment is conducted to demonstrate the performance of the phase angle measurements of three frequency disturbance recorders (FDRs) loaded with different algorithms. These FDRs are FNET synchronized single-phase phasor measurement devices. This chapter then proceeds to prove the effectiveness of the proposed algorithm by the comparison among the angle results.

4.2 DFT Phase Angle Measurement Algorithm and its Error Analysis

Different techniques are used to measure phase angles, including discrete Fourier transform technique, zero-crossing technique, and level crossing technique. The DFT technique is the most commonly used and is much clearer in concept with regard to the definition of instantaneous phase angle, whose basic idea is to calculate the voltage phasor for a balanced system operating at off-nominal frequency using the nominal frequency-based voltage samples[17][18]. The focus of this section is the DFT method and the analysis of its performance in the presence of off-nominal operation and single-phase setting.

Consider a sinusoidal input signal of angular frequency ω given by

$$x(t) = \sqrt{2}X\cos(\omega t + \varphi) \qquad (4\text{-}1)$$

where $\omega = 2\pi(f_0 + \Delta f)$, f_0 is nominal frequency and Δf is frequency deviation from nominal value.

According to electrical engineering convention, this signal is usually represented by a complex number

$$x = Xe^{j\varphi} = X\cos\varphi + jX\sin\varphi \qquad (4\text{-}2)$$

which is called the phasor of the input signal.

If the input signal is sampled N times per cycle of the f_0 waveform, the sample set

$$\omega t = \frac{2(f_0 + \Delta f)k\pi}{f_0 N} \quad \text{and}$$

$$x_k = \sqrt{2}X\cos(\frac{2(f_0 + \Delta f)k\pi}{f_0 N} + \varphi) \qquad (4\text{-}3)$$

will be produced.

The DFT of $\{x_k\}$ for the data window between k=0 and k=N-1 can be calculated as

$$\overline{x} = \frac{2}{N}\sum_{k=0}^{N-1} x_k e^{-jk\psi} \tag{4-4}$$

where $\psi = \dfrac{2\pi}{N}$.

Equation (4-3) above can be expressed with the help of Euler's formula as follows

$$x_k = \sqrt{2}X\,\frac{(e^{j(\frac{2(f_0+\Delta f)k\pi}{f_0 N}+\varphi)} + e^{-j(\frac{2(f_0+\Delta f)k\pi}{f_0 N}+\varphi)})}{2} \tag{4-5}$$

Substitute (4-5) into (4-4) and perform a series of development, the phasor can be simplified as

$$\overline{x} = \sqrt{2}Xe^{j\varphi}\frac{\sin(\frac{\pi\Delta f}{f_0})}{N\sin(\frac{\pi\Delta f}{Nf_0})}e^{j(\frac{(N-1)\pi\Delta f}{Nf_0})} + \sqrt{2}Xe^{-j\varphi}\frac{\sin(\frac{\pi\Delta f}{f_0})}{N\sin(\frac{\pi(2f_0+\Delta f)}{Nf_0})}e^{-j(\frac{(N-1)\pi(2f_0+\Delta f)}{Nf_0})} \tag{4-6}$$

For the convenience of analyzing the amplitude and the phase angle, (4-6) can be otherwise written as

$$\overline{x} = \sqrt{2}Xe^{j\varphi}\frac{\sin\left(\frac{\pi\Delta f}{f_0}\right)}{N\sin\left(\frac{\pi\Delta f}{Nf_0}\right)}e^{j\left(\frac{(N-1)\pi\Delta f}{Nf_0}\right)}\left(1+\frac{\sin\left(\frac{\pi\Delta f}{Nf_0}\right)}{\sin\left(\frac{\pi(2f_0+\Delta f)}{Nf_0}\right)}e^{-j\left(2\varphi+\frac{2\pi(N-1)(f_0+\Delta f)}{Nf_0}\right)}\right) \tag{4-7}$$

The first fraction of (4-7) shows that the magnitude of the phasor is in error by an attenuation factor $\dfrac{\sin(\frac{\pi\Delta f}{f_0})}{N\sin(\frac{\pi\Delta f}{Nf_0})}$ as the frequency of the input signal deviates from the nominal value. When the nominal frequency is 60Hz and the sampling rate is 24 points per cycle, a frequency difference of 6Hz (10%) produces only an attenuation of 1.6% according to the attenuation factor expression. For all practical purposes, this effect can be neglected for normal frequency excursions. The second term $e^{j(\frac{(N-1)\pi\Delta f}{Nf_0})}$ in (4-7) represents a phase shift of constant magnitude when a fixed frequency deviation is present, and it shows that the constant phase shift is proportional to the frequency deviation of the input signal. It can be calculated that when the nominal frequency is 60Hz and the sampling rate is 24 points per second, a frequency excursion of 6Hz (10%) generates 17.25 degrees, which is so substantial that certain measures need to be taken in order to compensate.

In addition to the constant attenuation factor and phase shift caused by frequency deviation, the DFT phasor expression also contains the third term $\left(1+\frac{\sin(\frac{\pi\Delta f}{Nf_0})}{\sin(\frac{\pi(2f_0+\Delta f)}{Nf_0})}e^{-j(2\varphi+\frac{(N-1)2\pi(f_0+\Delta f)}{Nf_0})}\right)$, due to the existence of the second term of (4-6).

The phasor of the third term can be illustrated as a small fractional amplitude phasor $\overrightarrow{AB}$ rotating at the end of the unit phasor $\overrightarrow{OA}$ according to the definition of phasor addition. Phasor $\overrightarrow{OB}$ will be the phasor representation of the third term, as is shown in Figure 4.1.

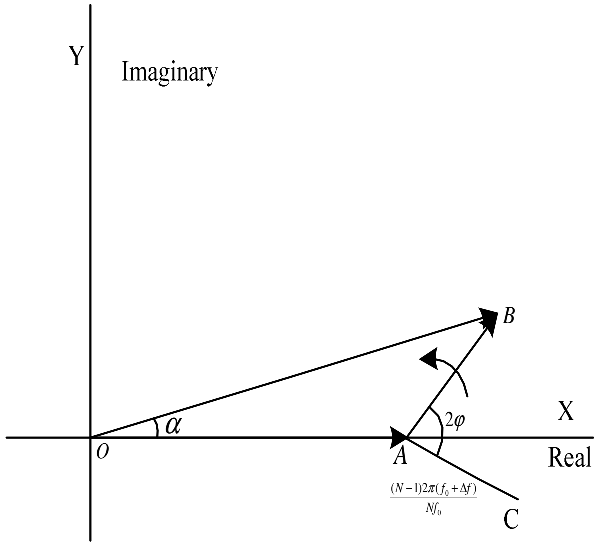


Fig. 4.1 Complex Coefficient of the Third Term

We can write based on the intuitive relations in Figure 4.1 that

$$|AB|=\frac{\sin(\frac{\pi\Delta f}{Nf_0})}{\sin(\frac{\pi(2f_0+\Delta f)}{Nf_0})} \text{ and } \angle BAX=-(2\varphi+\frac{(N-1)2\pi(f_0+\Delta f)}{Nf_0}) \qquad (4\text{-}8)$$

which means angle $\angle BAX$ starts at $\frac{(N-1)2\pi(f_0+\Delta f)}{Nf_0}$ and rotates at twice the angular speed of the measurand as the actual angle of the measurand varies from 0 to 2π.

As can be seen from Figure 4.1, the angle of phasor $\overline{OB}$ can be expressed as

$$\alpha = \angle AOB = \tan^{-1}\frac{|AB|\sin(\angle BAX)}{|OA|+|AB|\cos(\angle BAX)} \qquad (4\text{-}9)$$

For practical power system applications where the frequency deviation is considerably small and the sampling rate N is usually more than 10, the coefficient $\frac{\sin(\frac{\pi\Delta f}{Nf_0})}{\sin(\frac{\pi(2f_0+\Delta f)}{Nf_0})}$ will be remarkably small; for example, the coefficient will be only 0.048 when N=24 and frequency deviation is 6Hz (10%) of 60Hz nominal system. In other words, $|AB|$ is far less than the unit phasor $|OA|$ in 4.1. Therefore, (4-8) can be simplified to a fair approximation as:

$$\begin{aligned}\alpha &= \tan^{-1}\frac{|AB|\sin(\angle BAX)}{|OA|+|AB|\cos(\angle BAX)}\\ &\approx \tan^{-1}(|AB|\sin(\angle BAX))\\ &\approx |AB|\sin(\angle BAX)\end{aligned} \qquad (4\text{-}10)$$

Combining (4-8) and (4-10) yields

$$\alpha \approx -\frac{\sin(\frac{\pi\Delta f}{Nf_0})}{\sin(\frac{\pi(2f_0+\Delta f)}{Nf_0})}\sin(2\varphi + \frac{(N-1)2\pi(f_0+\Delta f)}{Nf_0}) \qquad (4\text{-}11)$$

Apparently, α is a sinusoidal function of the input phase angle φ, and taking into account the fact that $N > 10$ and $\Delta f < 0.1f_0$, the amplitude of the sinusoidal function can be given approximately by

$$\frac{\sin(\frac{\pi\Delta f}{Nf_0})}{\sin(\frac{\pi(2f_0+\Delta f)}{Nf_0})} \approx \frac{\frac{\pi\Delta f}{Nf_0}}{\frac{\pi(2f_0+\Delta f)}{Nf_0}} = \frac{\Delta f}{2f_0+\Delta f} \qquad (4\text{-}12)$$

As a summary of the previous discussion, when the frequency of the input stays at a constant excursion from the nominal value, the measured phase angle by DFT algorithm will consist of three components:

- The actual phase angle φ
- The invariant error $\frac{(N-1)\pi\Delta f}{Nf_0}$ (Fixed Offset)

- The variant sinusoidal form error

$$-\frac{\sin(\frac{\pi\Delta f}{Nf_0})}{\sin(\frac{\pi(2f_0+\Delta f)}{Nf_0})}\sin(2\varphi+\frac{(N-1)2\pi(f_0+\Delta f)}{Nf_0})\ \text{(Sine Variation).}$$

The total systematic error should be

$$\frac{(N-1)\pi\Delta f}{Nf_0}-\frac{\sin(\frac{\pi\Delta f}{f_0})}{\sin(\frac{\pi(2f_0+\Delta f)}{Nf_0})}\sin(2\varphi+\frac{(N-1)2\pi(f_0+\Delta f)}{Nf_0}) \qquad (4\text{-}13)$$

and according to (4-12) we get

$$Error=\frac{(N-1)\pi\Delta f}{Nf_0}-\frac{N\Delta f}{2f_0+\Delta f}\sin(2\varphi+\frac{(N-1)2\pi(f_0+\Delta f)}{Nf_0}) \qquad (4\text{-}14)$$

It can therefore be concluded that the systematic error of phase angle of the DFT phasor measurement is an offset sinusoidal function of twice the input angle plus a starting value, as shown in Figure 4.2.

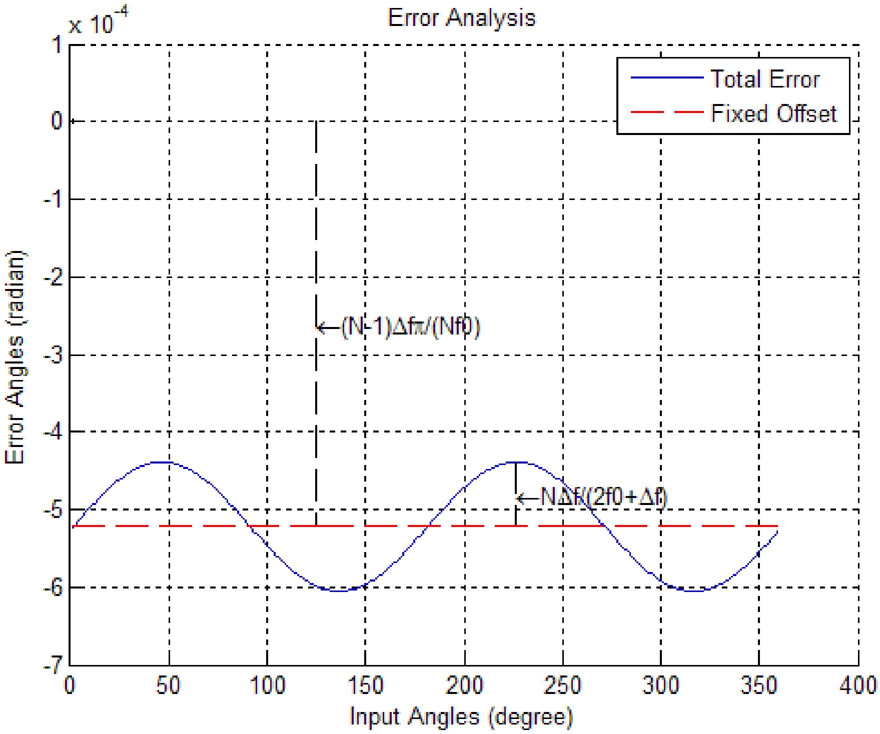


Fig. 4.2 The Phase Angle Error of DFT Algorithm

The content above is basically about the measurement with one voltage input, whereas the power system is a three-phase system, which means there are three separate voltages staggered in phase available for the measurement system when all the three phases of the power systems can be obtained, transduced, transmitted, sampled, and computed. This attribute of the power system provides an advantage in the phase angle measurement as revealed in the rest of this section below.

There might be frequency deviations in power systems, but the frequency is practically deemed as unchanged during the time period of phasor measurement which is usually only one or two cycles. As such, the two terms of (4-6) can be interpreted as the products of $e^{j\varphi}$ or $e^{-j\varphi}$ and their respective coefficients that can be expressed as the two constant complex values below:

$$P = \sqrt{2}X \frac{\sin(\frac{\pi\Delta f}{f_0})}{N\sin(\frac{\pi\Delta f}{Nf_0})} e^{j(\frac{(N-1)\pi\Delta f}{Nf_0})}$$

and $Q = \sqrt{2}X \frac{\sin(\frac{\pi\Delta f}{f_0})}{N\sin(\frac{\pi(2f_0+\Delta f)}{Nf_0})} e^{-j(\frac{(N-1)\pi(2f_0+\Delta f)}{Nf_0})}$ (4-15)

Doing that gives

$$\bar{x} = Pe^{j\varphi} + Qe^{-j\varphi} \quad (4\text{-}16)$$

Now consider three-phase-balanced voltage input at a stable frequency, which is a common phenomenon in the power system. The three phases are $\frac{2}{3}\pi$ out of phase with one another regardless of the frequency deviation of the voltage input. Accordingly, the three complex expressions of the phasor quantities can be written as

$$\begin{aligned} \bar{x}_a &= Pe^{j\varphi} + Qe^{-j\varphi} \\ \bar{x}_b &= Pe^{j(\varphi+\frac{2}{3}\pi)} + Qe^{-j(\varphi+\frac{2}{3}\pi)} \\ \bar{x}_c &= Pe^{j(\varphi-\frac{2}{3}\pi)} + Qe^{-j(\varphi-\frac{2}{3}\pi)} \end{aligned} \quad (4\text{-}17)$$

By convention, the positive sequence phasor is defined as the positive sequence component of phase A, which can be obtained by

$$\bar{x}_+ = \bar{x}_{a+} = \frac{1}{3}(\bar{x}_a + e^{-j\frac{2}{3}\pi}\,\bar{x}_b + e^{j\frac{2}{3}\pi}\,\bar{x}_c) \tag{4-18}$$

Applying (4-17), we can deduce as follows

$$\begin{aligned}
\bar{x}_+ = \bar{x}_{a+} &= \frac{1}{3}\left((Pe^{j\varphi}+Qe^{-j\varphi}) + e^{-j\frac{2}{3}\pi}(Pe^{j(\varphi+\frac{2\pi}{3})}+Qe^{-j(\varphi+\frac{2\pi}{3})}) + e^{j\frac{2}{3}\pi}(Pe^{j(\frac{2\pi}{3})}+Qe^{-j(\varphi-\frac{2\pi}{3})})\right)\\
&= \frac{1}{3}\left((Pe^{j\varphi}+e^{-j\frac{2}{3}\pi}Pe^{j(\varphi+\frac{2\pi}{3})}+e^{j\frac{2}{3}\pi}Pe^{j(\varphi-\frac{2\pi}{3})}) + (Qe^{-j\varphi}+e^{-j\frac{2}{3}\pi}Qe^{-j(\varphi+\frac{2\pi}{3})}+e^{j\frac{2}{3}\pi}Qe^{-j(\varphi-\frac{2\pi}{3f})})\right)\\
&= \frac{1}{3}\left(3Pe^{j\varphi}+Qe^{-j\varphi}(1+e^{-j(\frac{4}{3}\pi+\frac{2\pi}{3})}+e^{j(\frac{4}{3}\pi+\frac{2\pi}{3})})\right)\\
&= Pe^{j\varphi}+\frac{1}{3}Qe^{-j\varphi}(1+e^{-j\frac{4}{3}\pi}+e^{j\frac{4}{3}\pi})
\end{aligned} \tag{4-19}$$

Since $1+e^{-j\frac{4}{3}\pi}+e^{j\frac{4}{3}\pi}=0$, the second term of (19) can be canceled out and thus it can be simplified as

$$\bar{x}_+ = Pe^{j\varphi} = \sqrt{2}X\,\frac{\sin(\frac{\pi\Delta f}{f_0})}{N\sin(\frac{\pi\Delta f}{Nf_0})}\,e^{j(\frac{(N-1)\pi\Delta f}{Nf_0})}\,e^{j\varphi} \tag{4-20}$$

The phasor of a certain phase is defined as equal to the positive sequence component of that phase when the three phases are balanced in a power system. Recognizing that fact and comparing (4-7) and (4-20), the sine variation part of the error of the measured phase angle by DFT algorithm is canceled out by the balancing effect of the three-phase system regardless of the frequency deviation of the input voltage, which is thought to be a bonus merit of the three-phase structure of the power system and might have been beyond the original intentions of the pioneers who first established the power systems.

4.3 Methods to Improve Single-Phase DFT Phase Angle Measurements

By comparison of the analyses of the phase angle results of the three-phase and the single-phase DFT phasor measurement algorithms, a fixed offset in both conditions is observed, whereas the three-phase condition does not have the sinusoidal error because of the canceling out effect of the second term of (4-6) for balanced three phase inputs. That being the case, appropriate strategies may need to be introduced in order to cope with those systematic errors. For the fixed offset, it is conceivable that negative compensation can be added to the phase angle in order to balance the error which is proportional to the frequency deviation of the input when accurate frequency can be computed based on not-as-accurate phase angles by the metrics explained in literature[19][20]. For the sinusoidal waveform error, to the extent that the error is caused by the single-phase condition, approaches that can imitate the three-phase condition using single-phase inputs would provide a novel means to eliminate the error, or at least reduce a great amount of the error.

4.3.1 Fixed Offset Balancing Method

The prevailing approach power system engineers use to get the frequency of the voltage input after the phase angles are obtained is to calculate the derivative of the phase angles. It is clear in 4.2 that the fixed offset error is constant at any angle of input, and therefore the derivative of the phase angles is not affected by the inaccuracy in the form of identical angular displacement.

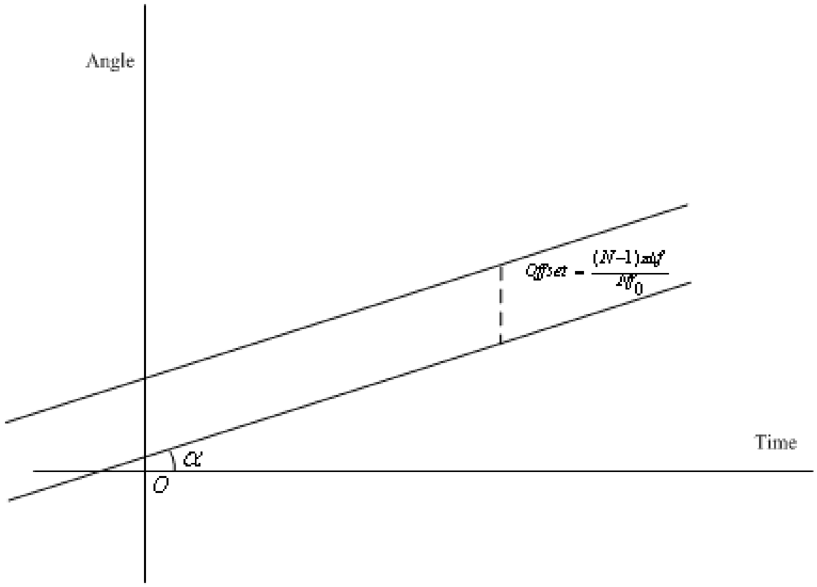


Fig. 4.3 The Fixed Offset Balancing Method

Assume the frequency of the input f is calculated to a satisfactory precision, then the frequency deviation can readily be obtained as $\Delta f = f - f_0$, where f_0 is the nominal frequency value for calculating frequency in the measurement algorithm. Based on the frequency excursion, the fixed offset can be calculated according to (4-12) as

$$Offset = \frac{(N-1)\pi\Delta f}{Nf_0} \qquad (4\text{-}21)$$

The phase angle can then be compensated by subtracting the result from the original value. The elements and procedures of the fixed offset balancing method are illustrated in Figure 4.3.

4.3.2 Sinusoidal Error Reducing Method

According to previous exploration, the sine variation error of the DFT phase angle measurement can be handled by the balancing effect of the three-phase input in power systems, which usually make available three voltages from different phases for measurement applications. However, that is not always the case, and it is highly possible that single-phase voltages or currents are used for power system applications, either because the application is a single-phase one or only one phase is procurable. In the absence of the access to all three of the phases, certain corrective actions need to be taken to address the sinusoidal error of the DFT phase angle measurement.

The sampling mechanism in the phasor measurement algorithm decides the sampling frequency is N times the nominal frequency where N is an integer. As a result the phase angle elapse during one sampling interval is $\frac{2\pi}{N}$ when the frequency of the input signal is precisely the nominal value. In order for us to acquire the three phases to cancel out the second term as in the previous section, two imitative phases must be constructed by relevant electrical phasor knowledge. Recognizing that the amount that the three phases are out of phase with one another is $\frac{2\pi}{3}$ in a three-phase system, it can be easily conceived that when N is an integer multiple of 3, $-\frac{N}{3}$ points after the current one can be considered an imitation of phase B and $\frac{N}{3}$ points after the current one can be considered an imitation of phase C.

However, the input frequency is not exactly the nominal one in most practical cases, and the change of phase angle during one sampling interval would be $\frac{2\pi f}{Nf_0} = \frac{2\pi(f_0 + \Delta f)}{Nf_0}$ instead of $\frac{2\pi}{N}$. Consequently, the phase angle amount that three phases are away from one another becomes $\frac{2\pi(f_0+\Delta f)}{3f_0}$ when $-\frac{N}{3}$ and $\frac{N}{3}$ points after the point of interest are employed to simulate the other two phases. Therefore, if the expression of phase A is the same as in (4-17), the expressions of imitative phase B and phase C must be modified as below

$$\bar{x}_b = (Pe^{j(\varphi - \frac{2\pi(f_0+\Delta f)}{3f_0})} + Qe^{-j(\varphi - \frac{2\pi(f_0+\Delta f)}{3f_0})})$$

$$\bar{x}_c = (Pe^{j(\varphi + \frac{2\pi(f_0+\Delta f)}{3f_0})} + Qe^{-j(\varphi + \frac{2\pi(f_0+\Delta f)}{3f_0})}) \qquad (4\text{-}22)$$

Following the same procedures in (4-19) to calculate the positive sequence phasor using imitative phase B and phase C, we can write

$$\begin{aligned}
\bar{x}_+ = \bar{x}_{a+} &= \frac{1}{3}\left((Pe^{j\varphi}+Qe^{-j\varphi}) + e^{-j\frac{2}{3}\pi}(Pe^{j(\varphi+\frac{2\pi(f_0+\Delta f)}{3f_0})} + Qe^{-j(\varphi+\frac{2\pi(f_0+\Delta f)}{3f_0})}) + e^{j\frac{2}{3}\pi}(Pe^{j(\varphi-\frac{2\pi(f_0+\Delta f)}{3f_0})} + Qe^{-j(\varphi-\frac{2\pi(f_0+\Delta f)}{3f_0})})\right) \\
&= \frac{1}{3}\left((Pe^{j\varphi} + e^{-j\frac{2}{3}\pi}Pe^{j(\varphi+\frac{2\pi(f_0+\Delta f)}{3f_0})} + e^{j\frac{2}{3}\pi}Pe^{j(\varphi-\frac{2\pi(f_0+\Delta f)}{3f_0})}) + (Qe^{-j\varphi} + e^{-j\frac{2}{3}\pi}Qe^{-j(\varphi+\frac{2\pi(f_0+\Delta f)}{3f_0})} + e^{j\frac{2}{3}\pi}Qe^{-j(\varphi-\frac{2\pi(f_0+\Delta f)}{3f_0})})\right) \\
&= \frac{1}{3}\left(Pe^{j\varphi}(1+e^{j\frac{2\pi\Delta f}{3f_0}} + e^{-j\frac{2\pi\Delta f}{3f_0}}) + Qe^{-j\varphi}(1+e^{-j(\frac{4}{3}\pi+\frac{2\pi\Delta f}{3f_0})} + e^{j(\frac{4}{3}\pi+\frac{2\pi\Delta f}{3f_0})})\right) \qquad (4\text{-}23)
\end{aligned}$$

The parts in (4-23) that are different from (4-19) are $1+e^{j\frac{2\pi\Delta f}{3f_0}}+e^{-j\frac{2\pi\Delta f}{3f_0}}$ and $1+e^{-j(\frac{4}{3}\pi+\frac{2\pi\Delta f}{3f_0})}+e^{j(\frac{4}{3}\pi+\frac{2\pi\Delta f}{3f_0})}$. Keeping in mind that $\frac{2\pi\Delta f}{3f_0}$ is a very small variable, we can simplify the two terms to be (4-24) and (4-25).

$$\begin{aligned}
&1+e^{j\frac{2\pi\Delta f}{3f_0}}+e^{-j\frac{2\pi\Delta f}{3f_0}} \\
&= 1 + (\cos\frac{2\pi\Delta f}{3f_0} + j\sin\frac{2\pi\Delta f}{3f_0}) + (\cos\frac{2\pi\Delta f}{3f_0} - j\sin\frac{2\pi\Delta f}{3f_0}) \\
&= 1 + 2\cos\frac{2\pi\Delta f}{3f_0} \\
&= 3 - 2(1-\cos\frac{2\pi\Delta f}{3f_0})
\end{aligned}$$

$$= 3 - 4\sin^2 \frac{\pi\Delta f}{3f_0} \tag{4-24}$$

$$\begin{aligned}
&1 + e^{-j(\frac{4\pi}{3}+\frac{2\pi\Delta f}{3f_0})} + e^{j(\frac{4\pi}{3}+\frac{2\pi\Delta f}{3f_0})} \\
&= 1 + \left(\cos(\frac{4\pi}{3} - \frac{2\pi\Delta f}{3f_0}) + j\sin(\frac{4\pi}{3} - \frac{2\pi\Delta f}{3f_0}) \right) + (\cos(\frac{4\pi}{3} - \frac{2\pi\Delta f}{3f_0}) - j\sin(\frac{4\pi}{3} - \frac{2\pi\Delta f}{3f_0}) \Big) \\
&= 1 + 2\cos(\frac{4\pi}{3} + \frac{2\pi\Delta f}{3f_0}) \\
&= 1 + 2\left(-\frac{1}{2}\cos\frac{2\pi\Delta f}{3f_0} + \frac{\sqrt{3}}{2}\sin\frac{2\pi\Delta f}{3f_0} \right) \\
&= \left(1 - \cos\frac{2\pi\Delta f}{3f_0} \right) + \sqrt{3}\sin\frac{2\pi\Delta f}{3f_0} \\
&= 2\sin^2\frac{\pi\Delta f}{3f_0} + \sqrt{3}\sin\frac{2\pi\Delta f}{3f_0}
\end{aligned} \tag{4-25}$$

Because higher order term $\sin^2\frac{\pi\Delta f}{3f_0}$ is infinitesimal compared to $\sin\frac{2\pi\Delta f}{3f_0}$ and therefore ignored when $\frac{2\pi\Delta f}{3f_0}$ approaches zero, (4-24) and (4-25) can be approximately given by (4-26), if the frequency deviation is sufficiently small.

$$\begin{aligned}
&1 + e^{j\frac{2\pi\Delta f}{3f_0}} + e^{-j\frac{2\pi\Delta f}{3f_0}} = 3 - 4\sin\frac{\pi\Delta f}{3f_0} \approx 3 \\
&1 + e^{-j(\frac{4}{3}\pi+\frac{2\pi\Delta f}{3f_0})} + e^{j(\frac{4}{3}\pi+\frac{2\pi\Delta f}{3f_0})} = 2\sin^2\frac{\pi\Delta f}{3f_0} + \sqrt{3}\sin\frac{2\pi\Delta f}{3f_0} \approx \frac{2\sqrt{3}\pi\Delta f}{3f_0}
\end{aligned} \tag{4-26}$$

Substituting (4-26) into (4-23), we get

$$\bar{x}_+ = Pe^{j\varphi} + Qe^{-j\varphi}\frac{2\sqrt{3}\pi\Delta f}{9f_0} \tag{4-27}.$$

After the same procedures from (4-6) to (4-14), the DFT phase angle measurement error of this method can be specified as

$$Error = \frac{(N-1)\pi\Delta f}{Nf_0} - \frac{2\sqrt{3}\pi\Delta f}{9f_0}\frac{N\Delta f}{2f_0+\Delta f}\sin(2\varphi + \frac{(N-1)2\pi(f_0+\Delta f)}{Nf_0}) \tag{4-28}$$

From (4-28), it can be concluded that by simulating two other phases from a single-phase sinusoidal input and then synthesizing the three complex DFT results to form a virtual positive sequence phasor, the sinusoidal error has been greatly reduced to only a negligible fraction of the original value while the fixed offset of the error remains unchanged.

The attenuation effect of the reduction is $\frac{2\sqrt{3}\pi\Delta f}{9f_0}$ as shown in (4-28), and it is less than 0.1% when the frequency deviation is no greater than 50 mHz.

4.3.3 Practical Equation

When the sampling rate N is an integer multiple of 3, and the vector of the samples is x_k, the phasor at k=0 is $\bar{x} = \frac{2}{N}\sum_{k=0}^{N-1} x_k e^{-jk\frac{2\pi}{N}}$.

According to the discussion above, the actual phasor $\bar{x}_a$ and the simulative $\bar{x}_b, \bar{x}_c$ can be expressed as

$$\bar{x}_a = \frac{2}{N}\sum_{k=0}^{N-1} x_k e^{-jk\frac{2\pi}{N}}, \bar{x}_b = \frac{2}{N}\sum_{k=0}^{N-1} x_{k-\frac{N}{3}} e^{-jk\frac{2\pi}{N}}, \bar{x}_c = \frac{2}{N}\sum_{k=0}^{N-1} x_{k+\frac{N}{3}} e^{-jk\frac{2\pi}{N}} \qquad (4\text{-}29)$$

From the definition of the positive sequence component given in (4-18), we can develop the result of substituting (4-29) to (4-12) as below

$$\begin{aligned}
\bar{x}_+ = \bar{x}_{a+} &= \frac{1}{3}\left(\frac{2}{N}\sum_{k=0}^{N-1} x_k e^{-jk\frac{2\pi}{N}} + e^{j\frac{2}{3}\pi}\frac{2}{N}\sum_{k=0}^{N-1} x_{k-\frac{N}{3}} e^{-jk\frac{2\pi}{N}} + e^{-j\frac{2}{3}\pi}\frac{2}{N}\sum_{k=0}^{N-1} x_{k+\frac{N}{3}} e^{-jk\frac{2\pi}{N}}\right) \\
&= \frac{2}{3N}\left(\sum_{k=0}^{N-1} x_k e^{-jk\frac{2\pi}{N}} + \sum_{k=0}^{N-1} x_{k-\frac{N}{3}} e^{-j(k-\frac{N}{3})\frac{2\pi}{N}} + \sum_{k=0}^{N-1} x_{k+\frac{N}{3}} e^{-j(k+\frac{N}{3})\frac{2\pi}{N}}\right) \\
&= \frac{2}{3N}\left(\sum_{k=0}^{N-1} x_k e^{-jk\frac{2\pi}{N}} + \sum_{k=-\frac{N}{3}}^{N-\frac{N}{3}-1} x_k e^{-jk\frac{2\pi}{N}} + \sum_{k=\frac{N}{3}}^{N+\frac{N}{3}-1} x_k e^{-jk\frac{2\pi}{N}}\right) \qquad (4\text{-}30)
\end{aligned}$$

After decompose the three summations above into three equal segments and then integrate the similar items, (4-30) can be written as

$$\bar{x}_+ = \frac{2}{3N}\left(\sum_{k=-\frac{N}{3}}^{-1} x_k e^{-jk\frac{2\pi}{N}} + 2\sum_{k=0}^{\frac{N}{3}-1} x_k e^{-jk\frac{2\pi}{N}} + 3\sum_{k=\frac{N}{3}}^{\frac{2N}{3}-1} x_k e^{-jk\frac{2\pi}{N}} + 2\sum_{k=\frac{2N}{3}}^{N-1} x_k e^{-jk\frac{2\pi}{N}} + \sum_{k=N}^{\frac{4N}{3}-1} x_k e^{-jk\frac{2\pi}{N}}\right) \qquad (4\text{-}31)$$

It is evident in (4-31) that by just expanding the sampling window to 5/3 cycles and assigning different weights to the different segments of the sample vector, the sinusoidal error of phase angles can be nearly eliminated without much change in DFT expression. Because the idea of (4-31) is much related to the concept of the positive sequence phasor, the technique may hence be referred to as ‘quasi-positive-sequence DFT’.

The error attenuation effect of the ‘quasi-positive-sequence DFT’ is expounded in Figure 4.4. The left figure shows if the input is balanced three-phase voltages, the fixed offset of the angle errors of the three phases are the same whereas the sinusoidal components of the angle errors are 120 degrees out of phase with one another. The positive sequence transform of the vectors of the three phases will add up the sinusoidal components of the angle errors of the three phases, which aggregates to zero because they are evenly displaced. In the right figure, since the imitative phases are not exactly 120 degrees out of phase with one another, the sinusoidal components of the angle errors are not evenly displaced. As a result, the summation of the sinusoidal components of the angle errors will generate a residue because of the unbalance when the positive sequence transform of the vectors of the three phases is carried out.

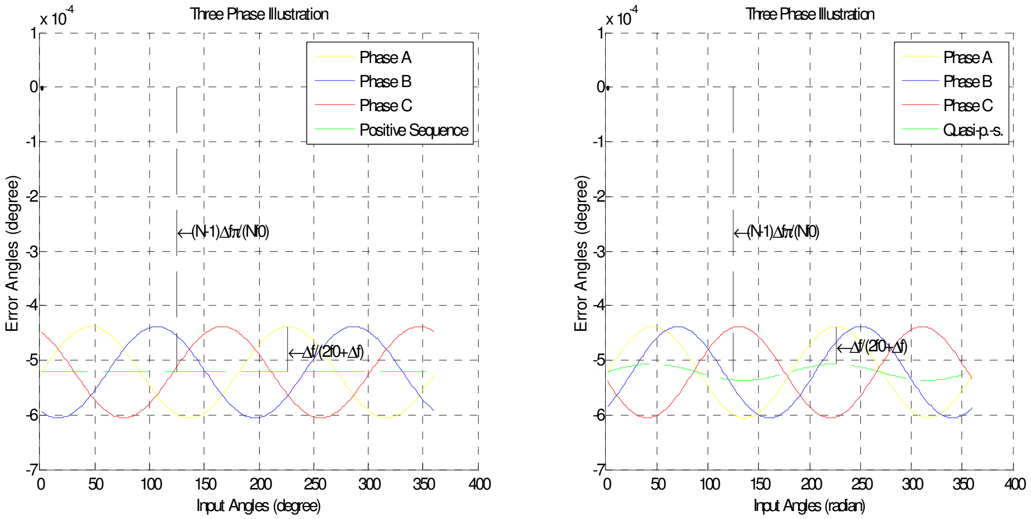


Fig. 4.4 The Error Attenuation Effect of the ‘Quasi-positive-sequence DFT’

4.4 Residue Problem and its Solution in FNET Practice

What has been previously theorized is more of the numerical analyses and mathematical deductions, which is based on the assumption that the sampling clock is perfect. For example, there are 24 sampling points per cycle in many power system measurement system applications for the nominal 60 Hz fundamental wave frequency, meaning that there are 1440 sampling points per second. Therefore the sampling clock will be 1440 Hz synchronized by the GPS. However the sampling clock generated by clock division is not perfectly 1440 Hz clock pulse for some practical reasons, which inevitably brings about a residue problem. This residue problem then time skews in measurement synchronization and induce inaccuracy in the phase angle computation and resultantly the frequency estimation. The cause of residue problem and the solution is described as follows.

4.4.1 Sampling Clock Generation Mechanisms in the FDRs

There are two generations of FDRs that have been developed in FNET. Their hardware designs with respect to the sampling clock generation mechanism are quite similar. Therefore, it is technologically proper for the Generation II FDRs, also known as Tsinghua units, to be used as a representative.

Generation II FDRs use Texas Instrument TMS320LF2407A as its microprocessor. The sampling clock is generated by an event-manager module on the chip. The EV Control Registers and Control Logic is the component that manages and orchestrates the functionalities of the module, including sending ADC start of conversion signal to the on-chip A/D conversion microprocessor. The DSP core interfaces with the control component, the timers, the capture units and the compare units via the data/address buses, and the reset, clock and interrupt signals.

The event-manager modules include general-purpose (GP) timers, full-compare/PWM units, capture units, and quadrature-encoder pulse (QEP) circuits. EVA and EVB timers, compare units, and capture units function identically. However, timer/unit names differ for EVA and EVB. The structure diagram is displayed in Figure 4.5.

There are two GP timers in TMS320LF24xx series chips that can be operated independently or synchronized with each other. The compare register associated with each GP timer can be used for compare function and PWM-waveform generation. There are three continuous modes of operations for each GP timer in up- or up/down-counting operations. Internal or external input clocks with programmable prescaler are used for each GP timer. GP timers also provide the time base for the other event-manager submodules: GP timer 1 for all the compares and PWM circuits, GP timer 2/1 for the capture units and the quadrature-pulse counting operations. Double-buffering of the period and compare registers allows programmable change of the timer (PWM) period and the compare/PWM pulse width as needed.

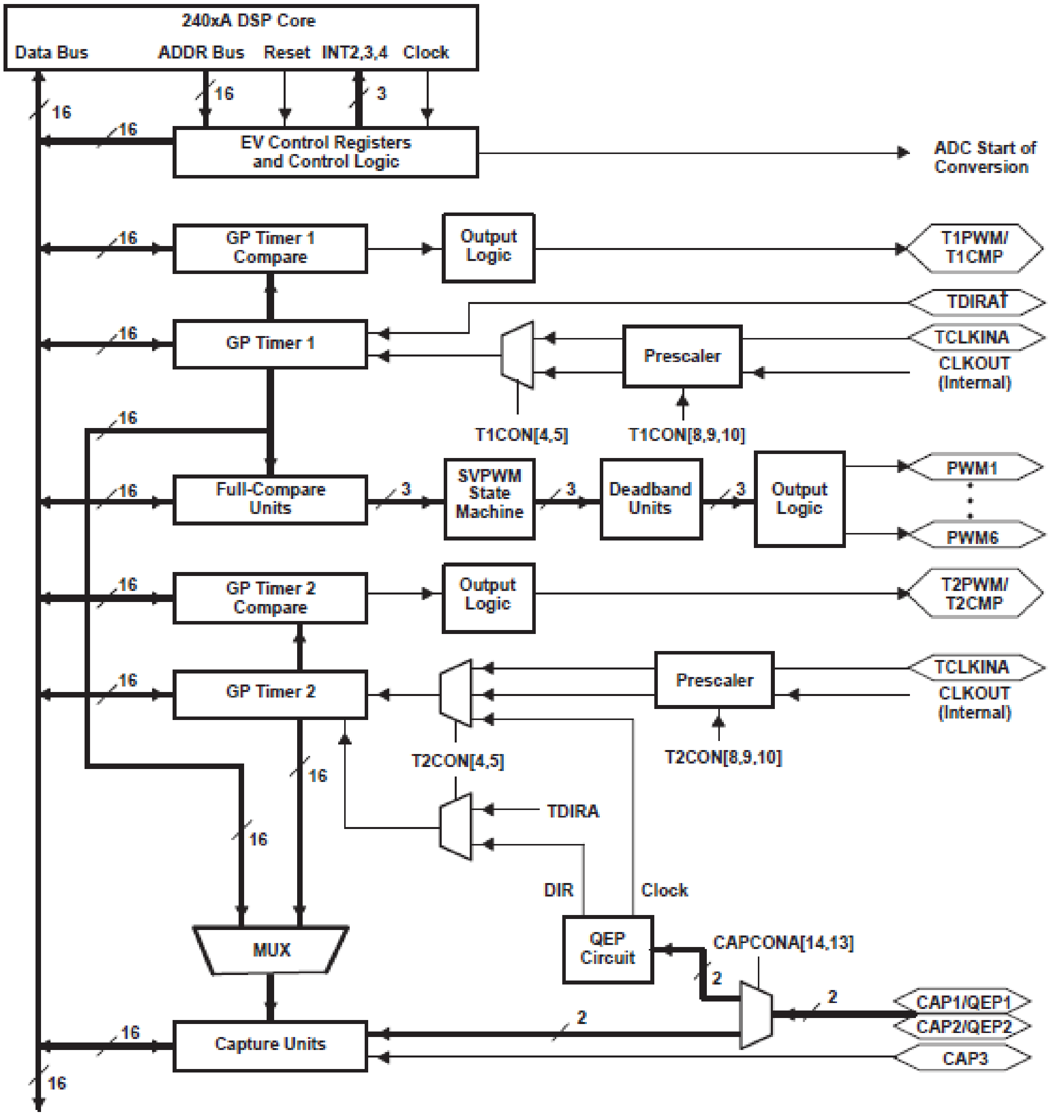


Fig. 4.5 MIOS Block Diagram

Three full-compare units use GP timer1 as the time base and generate six outputs for compare and PWM-waveform generation using programmable deadband circuit. The state of each of the six outputs is configured independently. The compare registers of the compare units are double-buffered, allowing programmable change of the compare/PWM pulse widths as needed. Two capture inputs (CAP1 and CAP2 for EVA; CAP4 and CAP5 for EVB) can be used to interface the on-chip QEP circuit with a quadrature encoder pulse.

The capture unit provides a logging function for different events or transitions. The values of the selected GP timer counter is captured and stored in the two-level-deep FIFO stacks when selected transitions are detected on capture input pins, CAPx (x = 1, 2, or 3 for EVA; and x = 4, 5, or 6 for EVB). The capture unit consists of three capture circuits.

The generation II FDRs do not use the on-chip ADC module, but use an external, more accurate 16-bit A/D converter on the extended system of the PCB board instead. The DSP code sets the register TxCON to stipulate the mode of timer 1, and the registers TxCMPR and TxPR to assign the period of the T1CMP output signal, which is used to start the external ADC module on a constant-frequency basis. Once the A/D procedure is completed, a signal is sent to the CAP3 pin of the microprocessor to make an interrupt request so that the new A/D conversion result can be tended to. The DSP core responds to the interrupt request and enters the IRS subroutine. The timer is immediately halted upon entering the subroutine in which the new sample is read in and made ready for processing. Before leaving the subroutine, the program starts the timer to repeat the sampling-processing cycle. The capture unit 3 of the event-manger module deals with the PPS of the GPS signal, involving resetting some counter variables and flag variables to facilitate the running of the program.

In a nutshell, the event-manager module generates the desired sampling frequency and it works as a frequency divider. The amount of the total system clock pulses between two consecutive PPS divided by sampling number per second is the number of system clock pulses between adjacent sampling points:

$$N_{clkpus\_sampling} = floor(N_{clkpus\_PPS} / M) \qquad (4\text{-}32)$$

where, $N_{clkpus\_sampling}$ is the system clock pulses between adjacent sampling pulses

N_{clkpus_PPS} is number of the system clock pulses between two consecutive PPS

M is the sampling points per second

The parameter $N_{clkpus\_sampling}$ is 20 M for the generation I FDRs and 30 M for the generation II FDRs according to their input clock frequency and the timer that is used as the ADC starting signal. The parameter M is 1440 for the 60 Hz, 24 samples per cycle data acquisition system.

4.4.2 Residue Problem and its Previous Treatments

For generation I FDRs,

$$N_{clkpus\_sampling} = floor(N_{clkpus\_PPS} / M) = 20{,}000{,}000/1440 = floor(13888.8889) = 13888 .$$

For generation II FDRs,

$$N_{clkpus\_sampling} = floor(N_{clkpus\_PPS} / M) = 30{,}000{,}000/1440 = floor(20833.3333) = 20833.$$

As the consequence, a residue forms because of the fraction of the quotient of N_{clkpus_PPS} divided by M , which can be calculated as follows:

$$\text{Residue} = 20{,}000{,}000 - (1440 * 13888) = -1280 \text{ (Generation I)}$$

$$\text{Residue} = 30{,}000{,}000 - (1440 * 20833) = 480 \text{ (Generation II)}$$

The residues are 64 μs and 16 μs respectively for the two generations of FDRs if converted into the time equivalents. Even though the residue is seemingly negligible and nothing near where it is comparable to the order of the time of the transient or subtransient phenomena, it still causes a fair amount of sampling point time shift every second. If left undelt with, the residue from each second would accumulate and one sampling point could be lost or added just as the intercalary days are added to leap years because of the indivisibility of the fact that the solar year does not have a whole number of days.

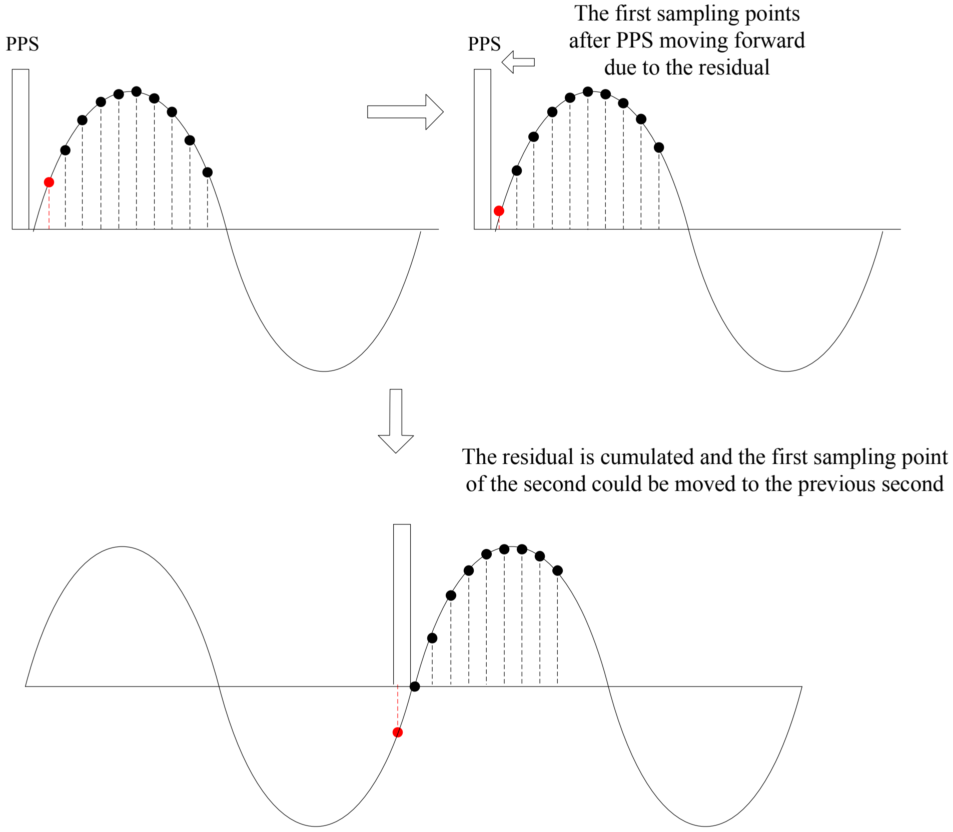

PPS
PPS
The first sampling points after PPS moving forward due to the residual
The residual is cumulated and the first sampling point of the second could be moved to the previous second

Fig. 4.6 Sampling Time Shift Caused by the Residue Problem

If there is no residue and the grid frequency is perfectly 60 Hz, the first sampling point of each second starts at the same position referring to the beginning of each second. However, as shown in Figure 4.6, the residue problem causes the first sampling point of each second moving forward. After certain time, the residue accumulates and thus drives the sampling point, which would be the first sampling point after PPS if there were not residue, go into the previous second shown as in the bottom diagram of Figure 4.6. Since the algorithm only allows 1440 sampling points per second to be used for the frequency estimation, the extra sampling point is consequently discarded. From the standpoint of calculation, this is tantamount to moving the whole sampling sequence one point ahead from the discarded point. Hence there would be a relatively abrupt "jump" (discontinuity) in the sampling voltage waveform used for phase angle and frequency computations.

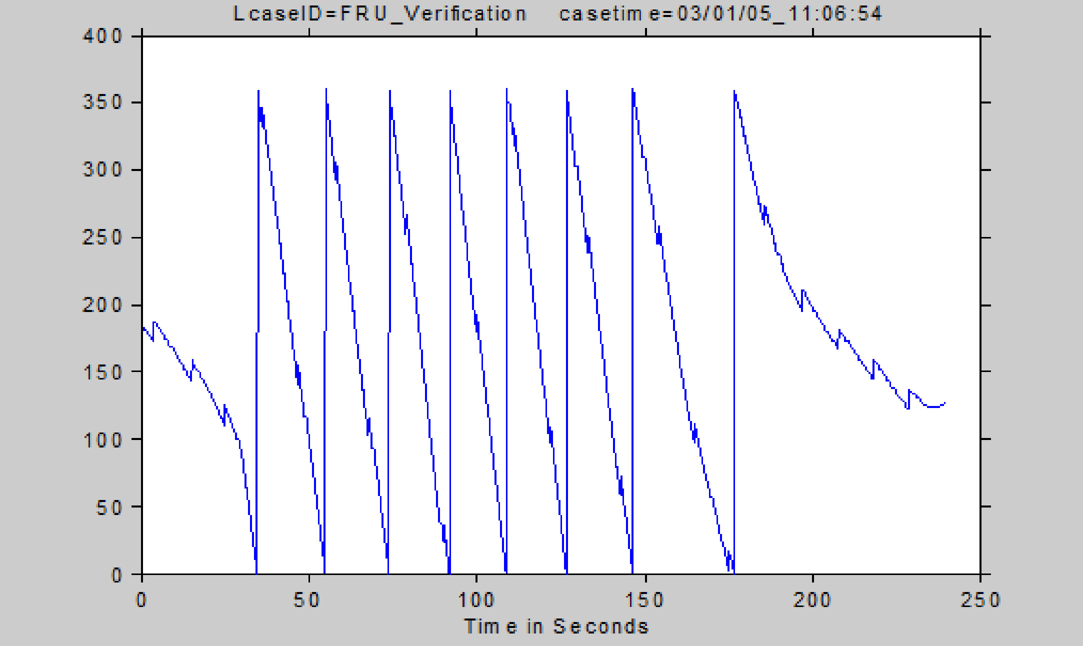
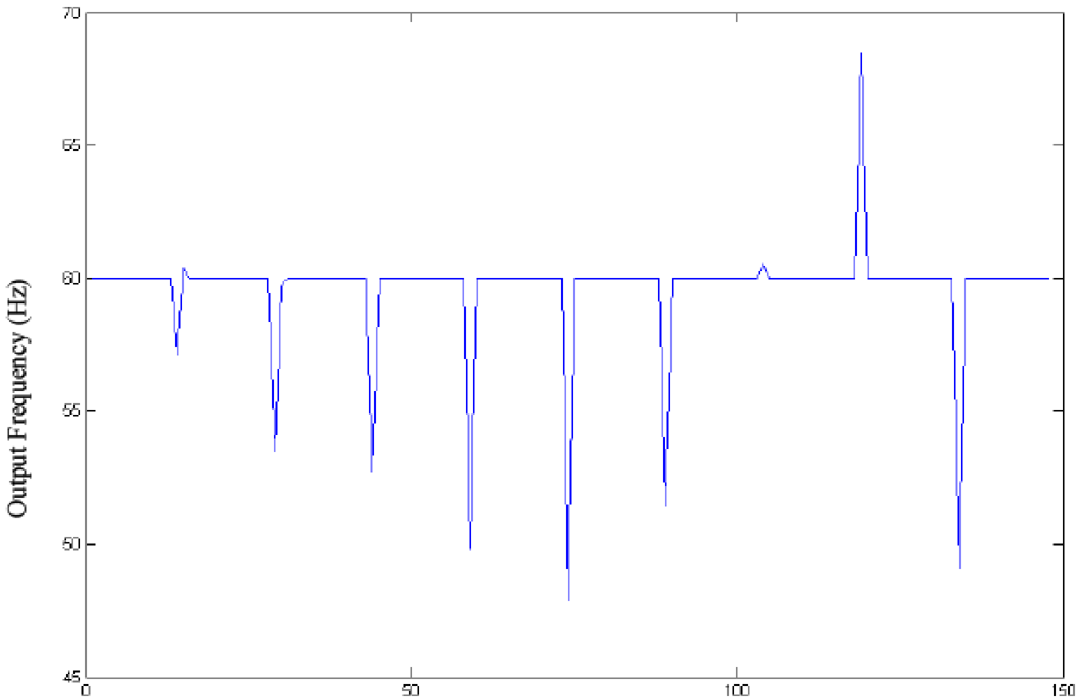


Fig. 4.7 Angle Discontinuity Caused by the Residue Problem

From another perspective, the 64 μs per second is an error of 6.4 μs per sampling interval, given that FNET data rate is 10 points per second. The error of 6.4 μs is equal to 0.13824 degree in a 60 Hz system, which indicates that there is a 0.13824 degree per sampling point systematic deviation in FNET angle measurements. The erroneous phase angle deviation keeps adding up to the cumulative error until the leap sampling point arrives and the FDR discards it. Figure 4.7 exhibits the effect the angle deviation. In the graph, every 10 seconds the phase angle jumps and the maximum phase angle error can be as large as 10 to 20 degrees. Note that the vertical 360 degree angle rise is the angle wrapping effect, and that the 10 to 20 degree angle error occurs at the small tooth-shaped jumps, such as second 195 and second 205.

According to the deductions in the previous chapter, the frequency is calculated by the derivative of the phasor angles and thus a spike forms in time of the phase angle jump because the leap sampling point is discarded. The fact that the “spikes” in frequency estimation algorithm shown in Figure 4.8 are caused by the loss of a voltage sampling point instead of real frequency change in a power system implies that a systematic inaccuracy of the frequency estimation algorithm exists, which resultantly detracts from the merit of the frequency estimation algorithm.



Fi

g. 4.8 Test Result of Losing Sampling Point

One approach to resolve the problem is to correct the residue every second. Therefore even though the residue still exists, it will be compensated every second and will not be accumulated. To synchronize the sampling clock with the PPS and correct the residue problem every second, we define a fixed waiting time that the sampling clock will wait to start after the edge of PPS:

$$t_w = n_w \times t_{sys}$$

$$t_{sys} = 1 / f_{sys}$$

where n_w is the number of system clock to be counted for the waiting, t_{sys} is the time interval of system clock pulse.

The waiting time t_w should be as small as possible to make the sampling clock start every second as close as possible to the edge of PPS.

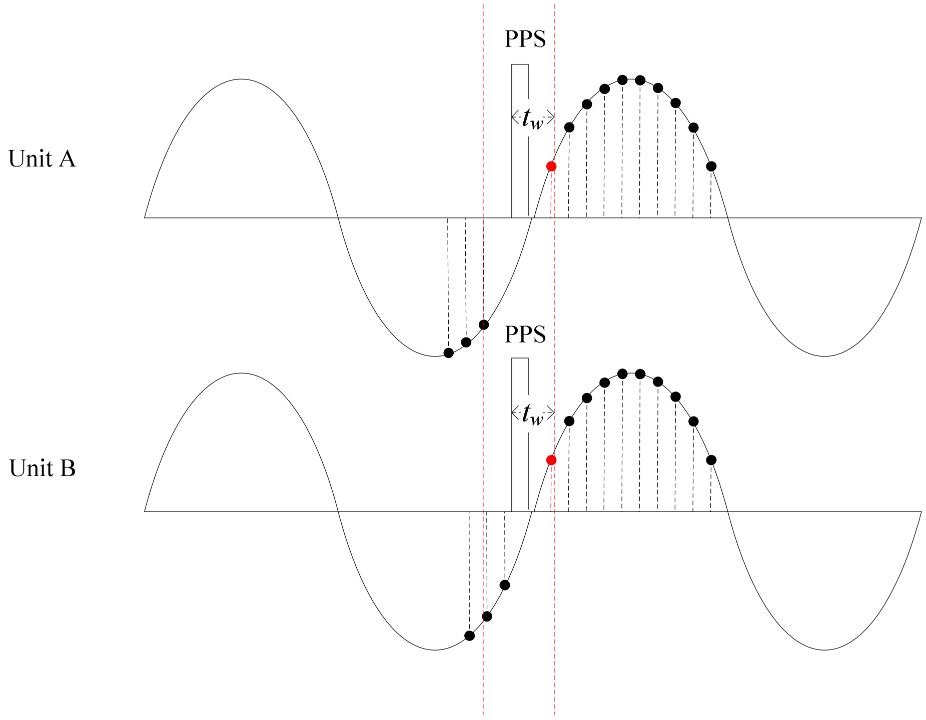


Fig. 4.9 Sampling Clock Waiting Time

The two units A and B in Figure 4.9 have different residues at the end of every second (caused by the slight different in system clock, actually the difference is not as significant as shown here). However the residue problem will be corrected at the beginning of every new second by waiting for a fixed time after the PPS rising edge. Therefore the hardware differences are confined within each second and will not accumulate. In the meanwhile, all the units' sampling clocks are synchronized by the PPS.

Although waiting for a fixed time after PPS corrects the residue problem every second, it brings about another problem that the sampling interval between the last pulse of previous second and the first pulse of current second is different from other sampling intervals. The problem is illustrated in Figure 4.10.

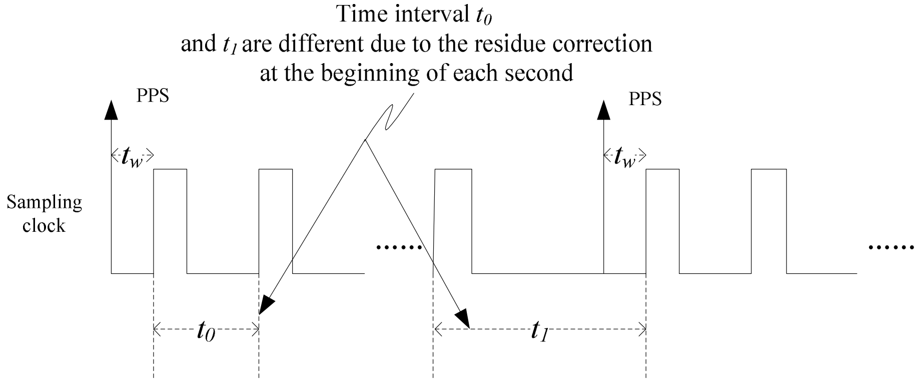

Time interval t_0
and t_1 are different due to the residue correction
at the beginning of each second
PPS
PPS
t_w
t_w
Sampling
clock
......
......
t_0
t_1

Fig. 4.10 Different Sampling Intervals due to the Residue Correction

As we can see from Figure 4.10, the sampling intervals of t_0 and t_1 are different. The algorithm requires the computation window cross several fundamental frequency cycles. For static frequency waveform, the longer the computation window, the better result of the least square error estimation of the frequency. However the computation window size could not be infinite due to the dynamic of the frequency. Considering power system frequency characteristics and the frequency dynamic requirement, 8 fundamental frequency cycles are used for frequency estimation. Since FDR output is 10 frequency estimations per second, the first frequency estimation output is at the 0.1 second of each second, which is 6 cycles. If the window size of the first estimation is set to be 6 cycles, it maximizes the window length to avoid using the sampling interval t_1, which otherwise produces frequency spikes.

4.4.3 The Current Solution to the Residue Problem

The method in previous section can limit the erroneous systematic angle deviation caused by the residue to each second and thus greatly improve the angle accuracy. It also eliminates the frequency jitters when the accumulated residue is such that the extra sampling point is discarded.

However, the residue still exists and can restrict the application of the phase angle and frequency measurements to some extent in the situations where very refined phase angle or frequency data are required.

The comparison between the phase angle measurements of the algorithm without residue compensation and the algorithm with residue compensation every second is displayed in Figure 4.11. It is a generation drop accompanied by oscillation event. The FDRs 6,11,15 are the sensors that have not been put the residue compensation into, whereas the rest of the sensors are downloaded with the firmware that has residue treatment feature in it. The benchmark for the phase angle measurements are the phase angle output of a PMU, which is a commercial phasor measurement device and has been precisely calibrated. Obviously, the red, cyan, yellow lines, which represent FDRs 6,11,15 respectively, are outstanding from the rest of the sensors. The phase angle plots from those outstanding devices are all periodical sawtooth shape waves. The period is approximately 16 s and the maximum angle mismatch from the PMU reference is about 15 degrees. The two parameters are roughly coherent with the residue obtained from the specifications of the generation I FDR hardware design. The plot waveform of the FDRs with residue compensation is also sawtooth shape, but the size is nothing near those of FDR 6,11,15. To some degree, they may be deemed as the miniature of the plots of those outstanding ones. The period of the regular ones whose residues are dealt with by compensation are exactly one second, which is the way it is supposed to be as to its code modification. The maximum angle error is squeezed to only 1.5 degrees as demonstrated in the graph below and by the analysis of the data collected.

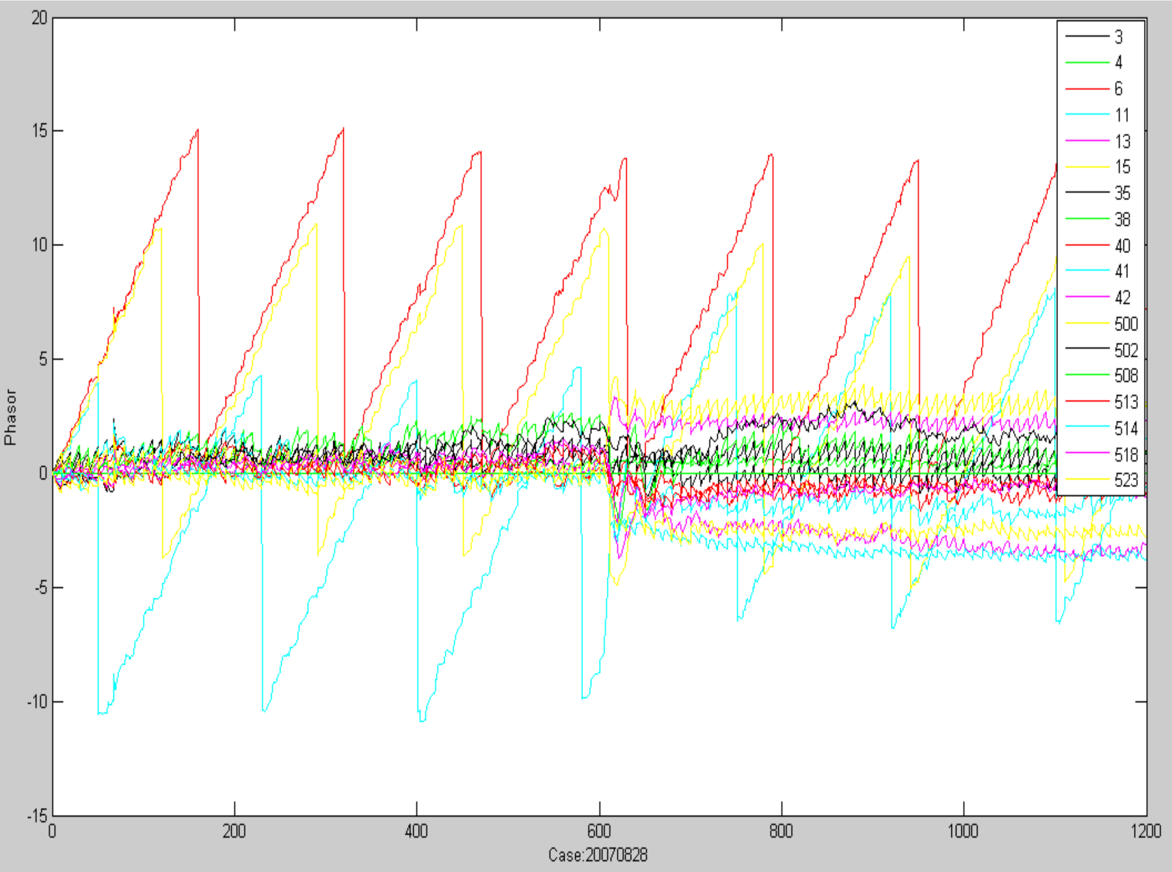


Fig. 4.11 Phase Angle Measurements in time of an Event

There is a one-order-of-magnitude improvement in the accuracy of the phase angle measurements when the residue compensation is introduced to the FDRs. However, seeing as the sawtooth error of the phase angle data is comparable to the phase angle change caused by the generation trip event around time index 600 in Figure 4.11, the accuracy of the phase angles after the residue treatment is still not up to par with the requirement of power system applications. In order to better take advantage of the phase angle measurements in FNET system, the residue issue needs to be further addressed and a more sophisticated mechanism should be put forward to enhance the accuracy so that it satisfies certain standards and it is therefore suitable for more extensive applications in FNET.

An experiment is performed to more thoroughly examine the residue, quantify the factors that contribute to the error, and proceed to determine the best correction methods to them. The experiment connected a PMU and a generation II FDR to a circuit that provides a 60 Hz signal and simultaneously feed the two devices. The measurement results are collected by the specific data concentrator programs of the two devices, and then are converted into a uniform MATLAB format for the ease of the analysis program. The data are processed by a MATLAB analysis program upon the completion of the format conversion, and the analysis program will supply with phase angle plots in the form of the comparison of FDR phase angles with those of the PMU. A 10 second long sample plot is presented blow.

The plot shows there is a very regular sawtooth-shape deviation from the PMU phase angle reference whose period is exactly 1.0 second. Figure 4.12 can be looked upon as a magnified graph over a ten second area of a unit of which the residue is compensated every second. The average angle drift rate is -1.305 degrees per second from the calculation of the 10 second data in the graph.

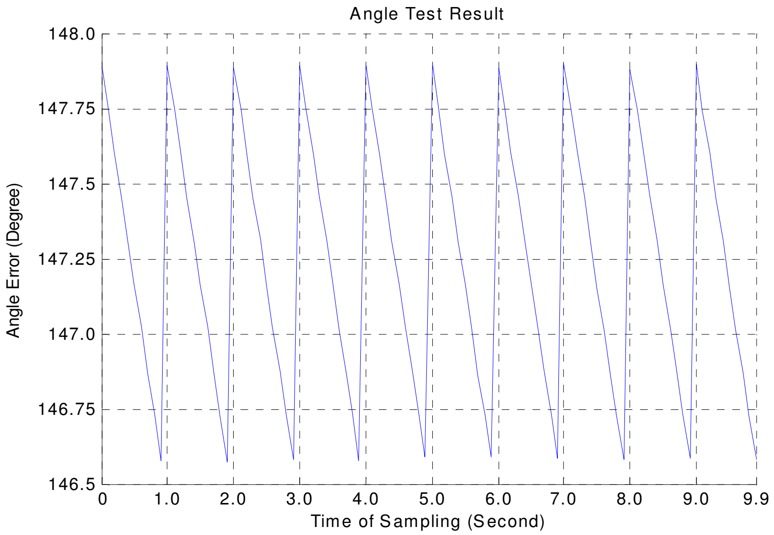
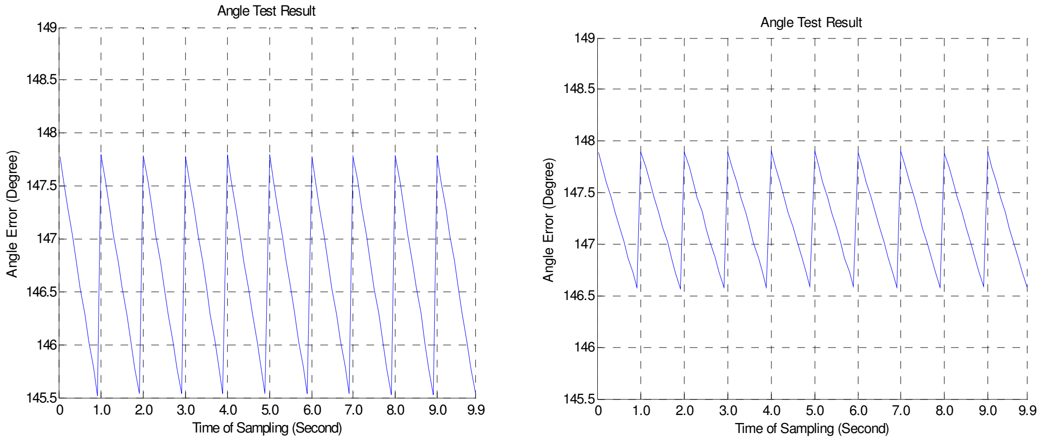


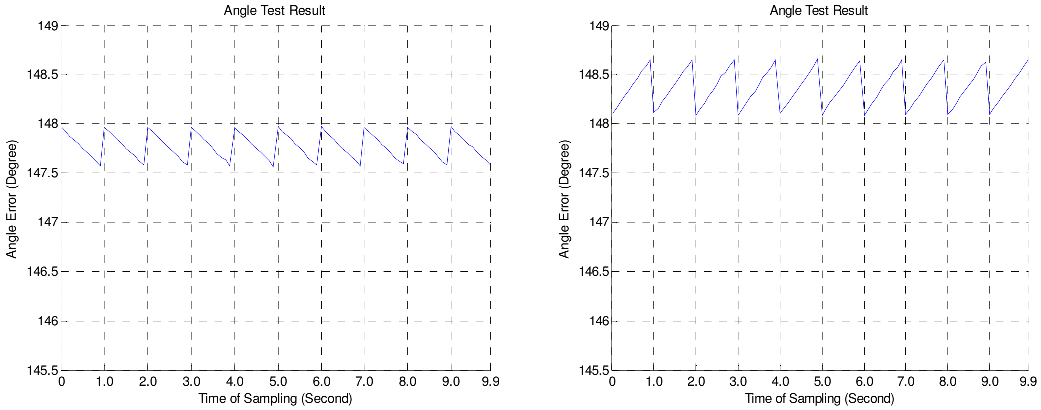
Fig. 4.12 Phase Angle Comparison of FDR and PMU

The discussion above tends to raise the query that whether or not changing the parameter of the sampling module can help to reduce the error. To test out that query, parameter $N_{clkpus\_sampling}$ in (4-32) is set to different values when downloading the firmware to the FDR, and measurements sent by the FDR with varied sampling clock pulse parameters are then gathered, processed and analyzed. The four $N_{clkpus\_sampling}$ values are 20832, 20833, 20834, and 20835 respectively, and their phase angles are plotted in the graphs of Figure 4.13.



CNT=20833-1 Error=-2.2532

CNT=20833 Error=-1.305



CNT=20833+1 Error=-0.38544

CNT=20833+2 Error=0.55135

Fig. 4.13 Phase Angle Plots of Different Sampling Parameters

The average angle drift rates of the four $N_{clkpus\_sampling}$ parameters appear to demonstrates a trend when $N_{clkpus\_sampling}$ increments from 20832 to 20835, which is apparently a linear increase from -2.532 degrees at $N_{clkpus\_sampling}$ being 20832 to 0.55135 degree at $N_{clkpus\_sampling}$ being 20835. The four pairs of parameters are plotted in a Cartesian coordinate system that uses the offset of $N_{clkpus\_sampling}$ to its theoretical value 20833 as the X-axis and the average angle drift rate as the Y-axis. The offset of $N_{clkpus\_sampling}$ to its theoretical value is the difference between $N_{clkpus\_sampling}$ and 20833, which are -1, 0, 1, and 2 respectively for the four $N_{clkpus\_sampling}$ values in Figure 4.13. A linear fit is executed using the MATLAB linear fit tool to find out the coefficients of the linear fit and then plot the line. The plot is exhibited in Figure 4.14.

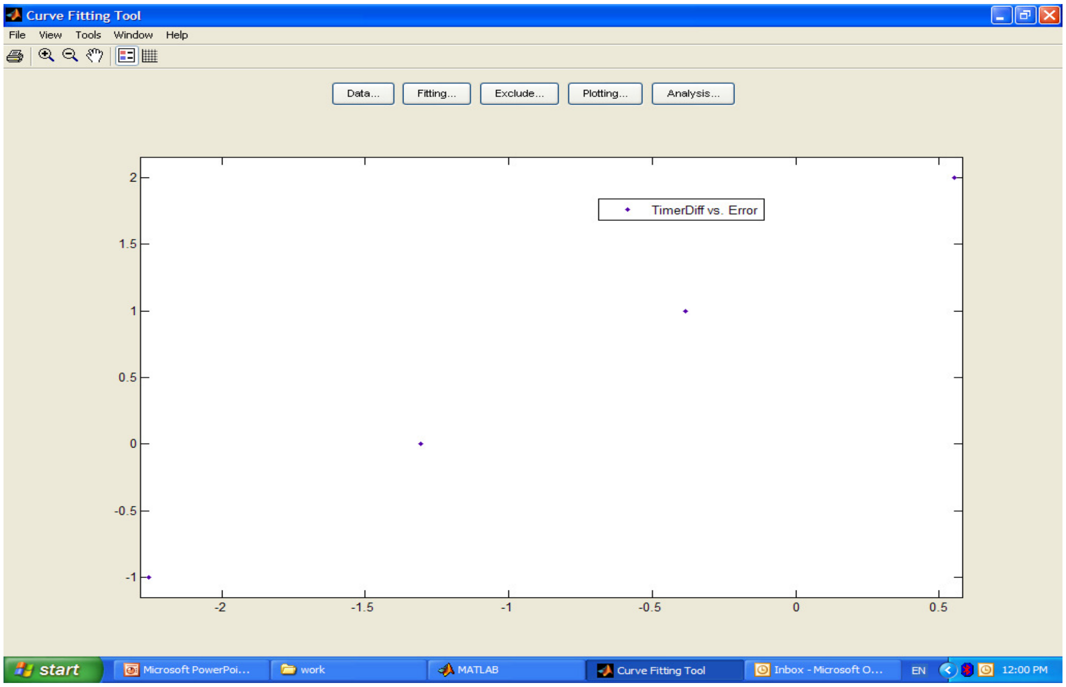


Fig. 4.14 Angle Drift Rates versus Parameter Offsets

The four dots are virtually all on the line without any appreciable straying from the line of the fitting result in Figure 4.15. The linear model for the linear regression computation takes the form:

$$N_{clkpus\_sampling} - 20833 = AngleDriftRate * p1 + p2 \qquad (4\text{-}33)$$

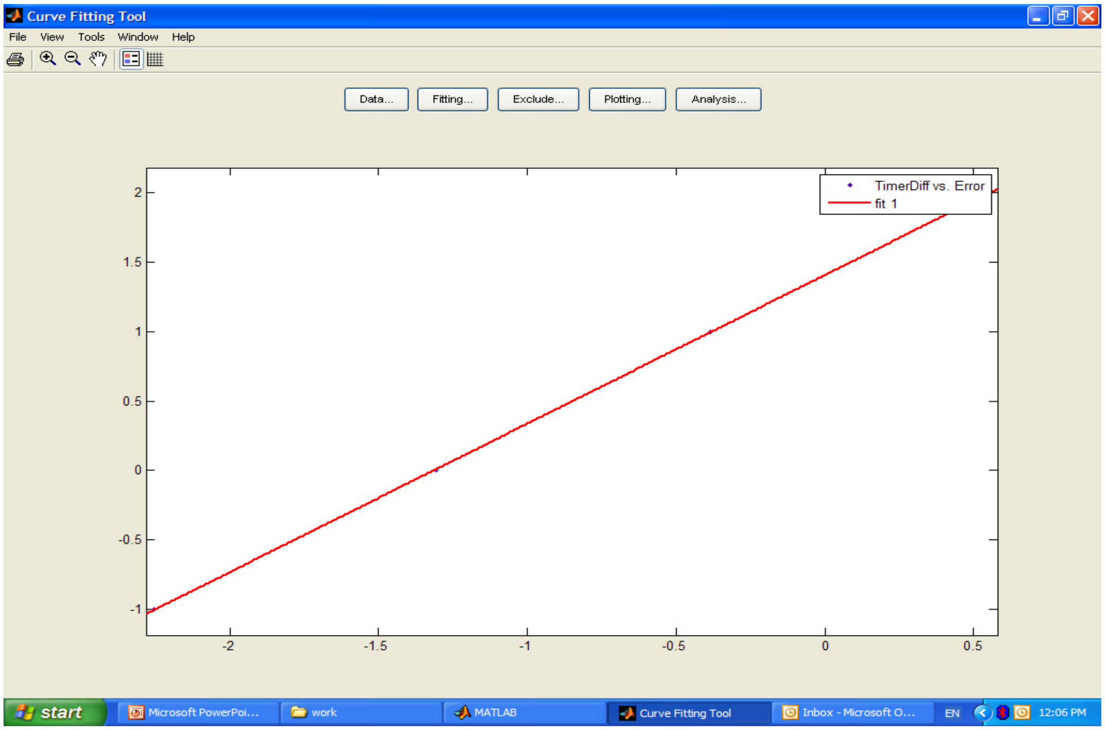


Fig. 4.15 Linear Fit of the Angle Drift Rates versus Parameter Offsets

The parameters $p1$ and $p2$ in (4-33) are estimated to be 1.071 and 1.409 from the data of the four dots on the graph. Let $AngleDriftRate = 0$, the parameter $N_{clkpus\_sampling}$ will be 20834.409, meaning that when $N_{clkpus\_sampling}$ is set to be 20834.409, the angle drift rate will be 0. No angle drift implies the residue no longer exists and the phase angle sawtooth error problem of both generations of FNET devices is completely resolved. One may wonder why the value of the parameter $N_{clkpus\_sampling}$ is different from the value calculated by $(N_{clkpus\_PPS} / M) = 20833.3333$. The reason is that there is hardware response time for the interrupt to be answered, for the ADC module to communicate to the DSP upon its completion of the A/D conversion, and possible congenital inaccuracy compensation of the DSP clock in the hardware system. All those factors together may take a toll on the clock pulse number per sampling, which reasonably consumes around one system clock time.

The foregoing analysis seems to point to the fact that a fractional $N_{clkpus\_sampling}$ parameter is necessary to fully solve the residue problem once and for all, whether it be 20834.409 of the laboratory result or 20833.333 of the theoretical deduction in the previous section. To that end, it is imperative for us to take stock of the structure and mechanism of the sampling of FNET devices.

As previously stated, the registers TxCMPR and TxPR of timer 1 assign the period of the T1CMP output signal, which is used to start the external ADC module on a constant-frequency basis. How the two registers coordinate is illustrated in Figure 4.16. The T1CON register sets the mode of the GP timer 1 compare module. The T1CNT is driven by 30 MHz system clock, increasing by 1 every system clock. When the T1CNT register reaches the value of the T1PR, it automatically rolls over and be reset to 0. The output of pin T1PWM/T1CMP is actually controlled by the result of the comparison between T1CNT and T1CMP, being set to be 1 when T1CNT is greater than or equal to T1CMP and 0 when T1CNT is less than or T1CMP. By acting so, the T1CMP register can adjust the duty cycle of the output of the pin T1PWM/T1CMP, thus making it possible to provide pulse-width modulated rectangular waves.

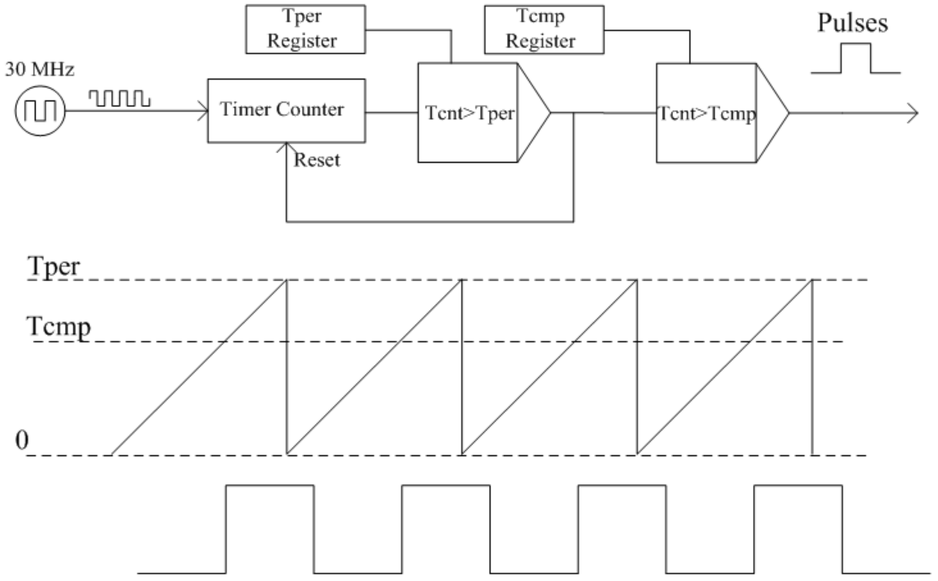


Fig. 4.16 Compare Unit Output Mode 1

Because only the rising edge of the rectangular waveform is utilized to trigger the A/D conversion in the FNET application, the duty cycle is not of much use to the sampling clock generation. The falling edge of the rectangular wave is not drawn right at the status change of the comparison between T1CNT and T1CMP, and as a result the rectangular wave shows like a square wave. This does not affect the analysis in that only the rising edge of the waveform is important to the analysis here.

As evidenced in Figure 4.16, the register T1PR plays a decisive role in determining the sampling interval of the A/D module, whereas the T1CMP is not quite relevant to this issue. The focus of this problem is then residing in how to manipulate the T1PR register in such a way that fractional number of system clocks in the sampling interval can be achieved.

The crux of the problem in this issue lies in the accumulation of the discrepancy between the fractional number quotient of the system clock frequency divided by the sample rate and the closest integral number that the register can only be assigned over a period of time. The time period is about 15 seconds in the original code and 1 second in the modified code, rendering the maximum angle drift being 15 degrees and -1.305 degrees respectively. Although the register cannot be assigned a fractional value, if the residue can be rectified every sampling interval, the systematic angle error caused by residue can be amply reduced. If the sampling rate is 1440 points per second and the maximum angle drift per second is 1.305 degrees, the maximum angle error stemming from the residue will be

1.305/1440=0.00090625 degree, which is absolutely inconsequential for any power system application.

The approach to create fractional number for the T1PR register, or in other words to compensate the residue every sampling point, is illustrated in Figure 4.17. This is a method to realize the value of Tper+0.5 in the T1PR register. In the first two A/D conversion ISR, the T1PR is set to be Tper; while in the last two A/D conversion ISR, the T1PR is set to be Tper+1. By doing that, the average T1PR value on a whole will be Tper+0.5.

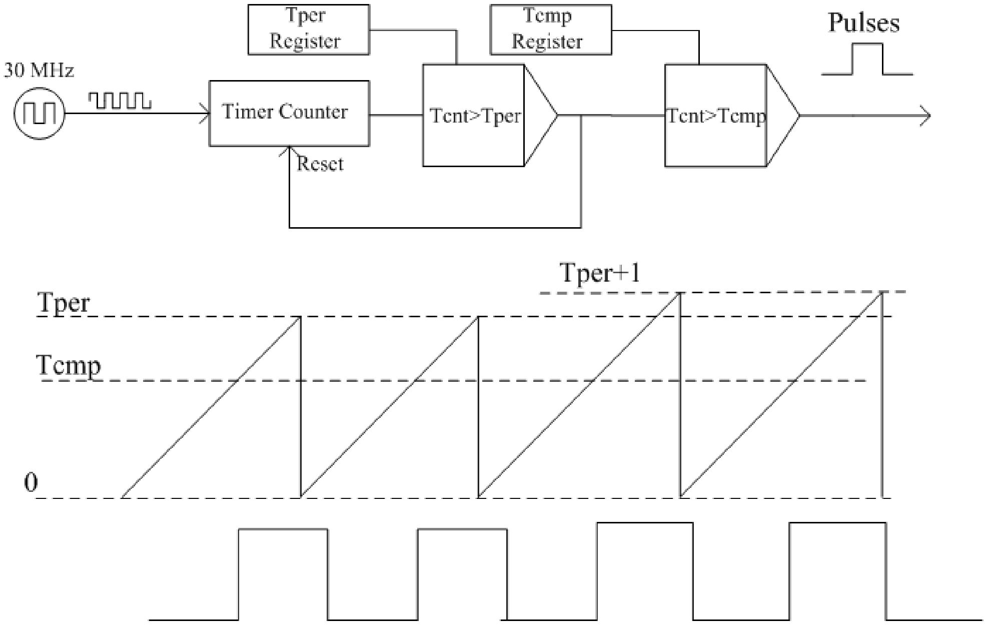


Fig. 4.17 Compare Unit Output Mode 2

Other fractional T1PR values, such as 0.405, can be produced in a similar manner by alternating different integral T1PR values at corresponding proportions. The practical approach to accomplish that goal is to keep track of the integral part of the product of the fractional quotient of N_{clkpus_PPS} / M and the counter of the sampling point sequence at each sampling point, and then assign the difference between that integral part value of the current point and that of the previous point to the T1PR register as its new value. Thus, the sum of the T1PR register value over a certain time period will always keep up with the nearest integer of the system clock number constituting that time period, meaning the residue is eliminated every sampling point free of the erroneous residue accumulation. The detailed code for this change in the angle computation algorithm is listed in Appendix.

A number of experiments are carried out to try out the effect of the complete residue rectifying approach mentioned above. The results show the sawtooth error totally disappears as evidenced in Figure 4.18.

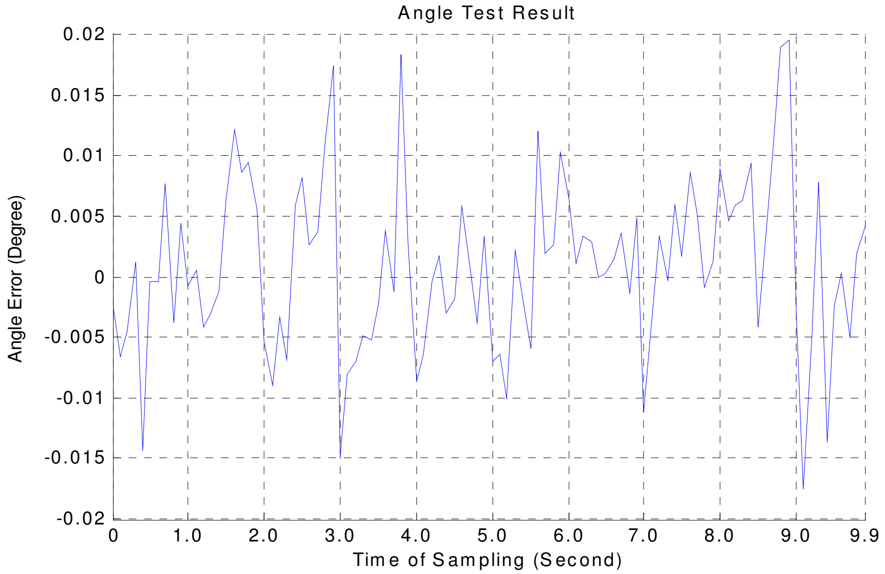


Fig. 4.18 Test Result of Complete Residue Compensation

The errors shown in Figure 4.18 are obviously random noise caused by numerous factors, including the hardware accuracy, the A/D module resolution, the precision of the GPS signal itself, and the limit of the FNET angle representation. Although the plot in Figure 4.18 is much more jittery than Figure 4.12 and also the errors seem to be larger than those of Figure 4.12, those numbers in Figure 4.18 are factually very small and nearly negligible compared to those in Figure 4.12 seeing that the scale used in Figure 4.18 is 0.04, whereas the scale used in Figure 4.12 is 4.5 degrees. It would make the plot look like a straight line if put into the same scale as Figure 4.12. The maximum deviation of the angle measurements obtained from Figure 4.18 is 0.0196 degree and the standard deviation is 0.0072 degree, which are too small to be taken into account for any power system application.

4.4.4 The Effect on the Frequency Computation

In FNET, there was a constant frequency drift in frequency measurements, which was a phenomenon that an invariable frequency deviation from the input frequency value, even when the signal source is stable, reliable and well calibrated. The angle measurements used to calculate the frequency are smooth and the frequency data themselves fluctuate to a very subdued extent, which leaves the phenomenon difficult to explain. For the simplicity of resolving it, the problem was attributed to being caused by the hardware imperfections and the component parameter instability.

By virtue of the fact that the frequency drift is constant on the whole, an offset compensation coefficient can be added to the final frequency result to make up for the drift. The effectiveness of that approach is quite satisfactory and therefore it had been a common practice for FNET to correct the frequency in that way.

However, after delving into the problem, we can find out that the frequency offset may result from the sawtooth error of the phase angle measurement, knowing that the frequency is by definition the derivative of the phase angles and the sawtooth is the phase angle change with a relatively constant slope. That being assumed, the experiment is performed to provide data for analysis. In the experiment, the input signal to the FDR was set to be 60.000 Hz, and 1 minutes/600 samples worth of data were collected onto the server.

According to the data gathered, the average frequency during the 1 minute time period was 59.9959 Hz. Since the period register is assigned 20833 for the test FDR, the maximum drifting angle per second value calculated with the data in this experiment is -1.305 degree, as can be corroborated by the previous result.

Since the sawtooth error is rectified at every whole number second and increases or decreases at fractional number seconds within a specific second, the angle drift maximizes at 0.9 second after the whole number second. Therefore, the maximum angle drift per second in the angle measurement denotes the difference of the angle at a whole number second subtracted from the angle at 0.9 second after the whole number second. A -1.305 degree over a 0.9 second time period is equivalent to -1.305/360 circle over 0.9 second time period, which by frequency definition equals to -1.305/(0.9*360) = -0.0041 Hz.

It hence can be identified that the sawtooth error of maximum -1.305 degrees can cause a frequency error of -0.0041 Hz. If subtracting the -0.0041 Hz frequency error from the 60.000 Hz, which is the input frequency, we get 59.9959 Hz, which is exactly the number that is calculated from the data gathered in the experiment. The consistency of the frequency error calculated from the experiment data and the frequency error obtained from the derivative of the sawtooth angle measurements attests to the fact that the constant frequency offset previously thought to issue from the hardware imperfections actually arises from the sawtooth-shaped errors of the angle measurements. If the sawtooth-shaped errors are corrected, the frequency offset issue will accordingly be fixed.

An experiment that aims to confirm the proposition that the frequency offset is caused by the sawtooth angle error is tried and the comparison between the frequency plot when the angle errors are fully compensated in the new DSP firmware and the frequency plot when the angle errors are not compensated in the old DSP firmware is presented in Figure 4.19 with the red plot representing the old firmware and the blue plot representing the new firmware. The frequency plot of the new firmware centers around the horizontal straight line 60.000 Hz and the average of the frequency data in the time period is exactly 60.000 Hz, which suggests that the constant frequency offset results from the sawtooth angle error and can be entirely eradicated if the sawtooth angle error is rectified.

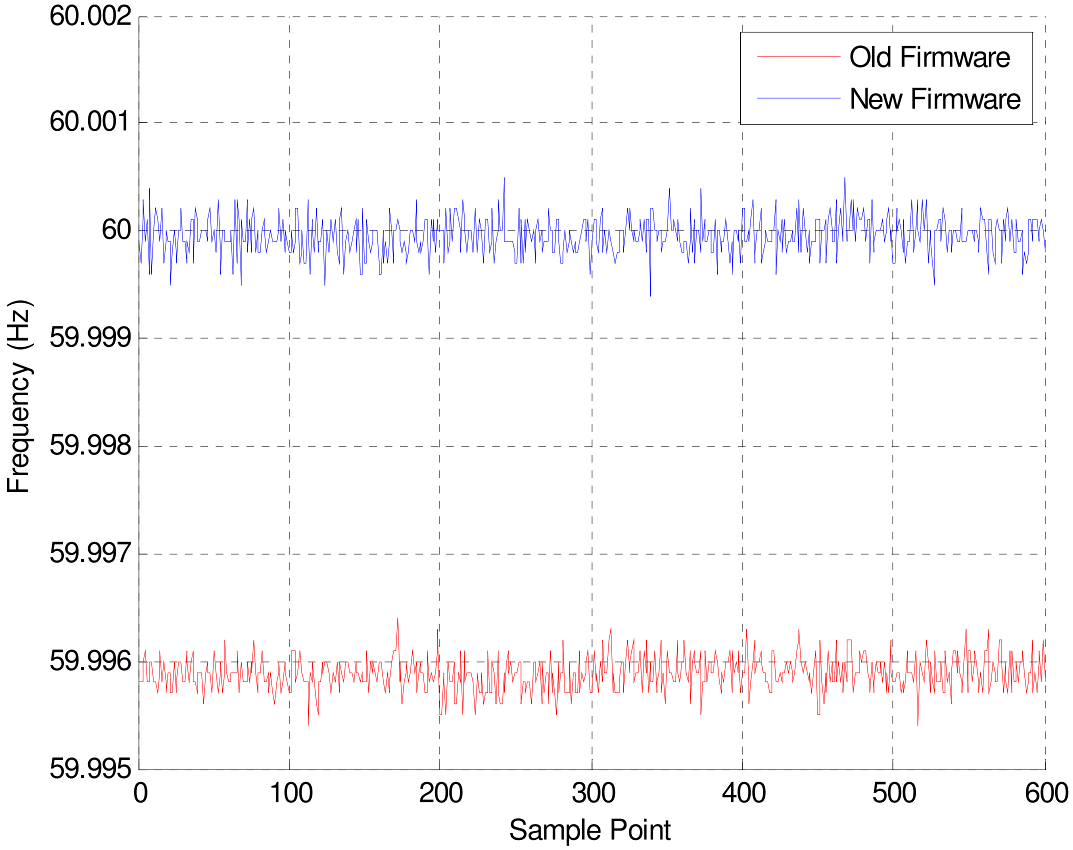


Fig. 4.19 Comparison of Frequency Measurements

In the next sections of this paper, simulations and laboratory tests are conducted to prove the validity and efficacy of the phase angle offset compensation and the quasi-positive-sequence DFT.

4.5 Numerical Simulations and Laboratory Experiments

In the first one of the two MATLAB simulation examples, the frequency of the input signal is 59 Hz, and the systematic angle errors of the proposed quasi-positive-sequence DFT method and the conventional DFT technique are plotted in Figure 4.20. The X-axis is the phase angle of the input signal which changes from 0 to 1200 degrees. As can be seen from the plot, the offsets of the two methods are equal, whereas the sinusoidal variation error of the quasi-positive-sequence DFT method is remarkably reduced compared to that of the conventional DFT technique. The attenuation of the sinusoidal error of the quasi-positive-sequence DFT is 0.019954, only a 0.99% variation from the approximation given by (4-28), 0.020153.

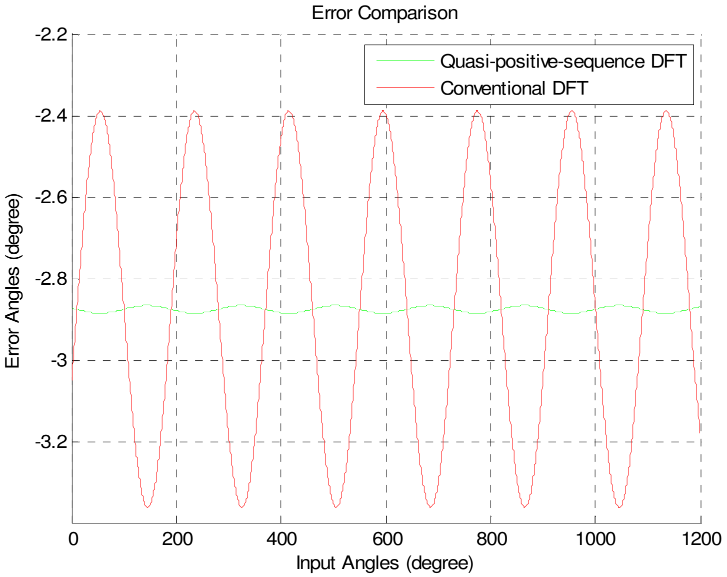


Fig. 4.20 Error Comparison for 59 Hz Input

In the second simulation example, the frequency of the input signal is 60.1 Hz, and the errors of the two methods are plotted in Figure 4.21 in the similar way to Figure 4.20. The most noticeable feature in which Figure 4.21 differs from Figure 4.20 is that the sinusoidal part of the error of the quasi-positive-sequence DFT method seems to vanish compared to that of the conventional DFT technique, notwithstanding the fact that the offsets of the two methods appear to be almost the same. The absolute values of the conventional DFT error decrease from the order of 2.4 to 3.3 degrees to the order of 0.24 to 0.33 degree as the ratio of the frequency deviation of the input signal with respect to the nominal frequency changes from 1/60 to 1/600, which is a very convincing proof to (4-21). The attenuation of the sinusoidal part of the error of the quasi-positive-sequence DFT algorithm is 0.0020174 in this simulation example and the attenuation value from (4-28) is 0.0020153, only 0.104% variation in between. The accuracy of the estimation of the attenuation effect by (4-28) increases from 99.0% to 99.9% as the frequency deviation decreases from 1.0 Hz to 0.1 Hz, as displayed from the four decimal attenuations.

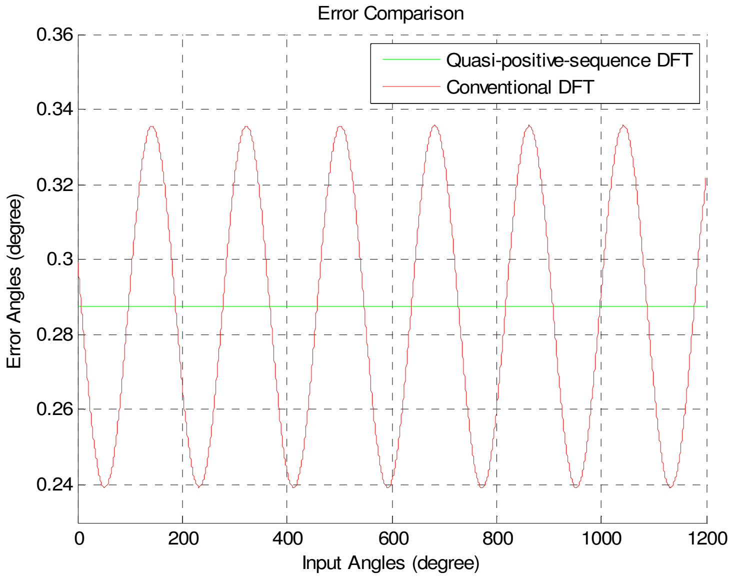


Fig. 4.21 Error Comparison for 60.1 Hz Input

With such a deeply attenuated sinusoidal part of the error, the error curve of the proposed method looks like a straight line without fluctuation, as shown in the green line Figure 4.21. However, with a close-up view of the error curve of the quasi-positive-sequence DFT method in Figure 4.22, the sinusoidal component of the error is still present with a totally negligible amount (0.0002 degree of peak-to-peak value) compared to the fixed offset.

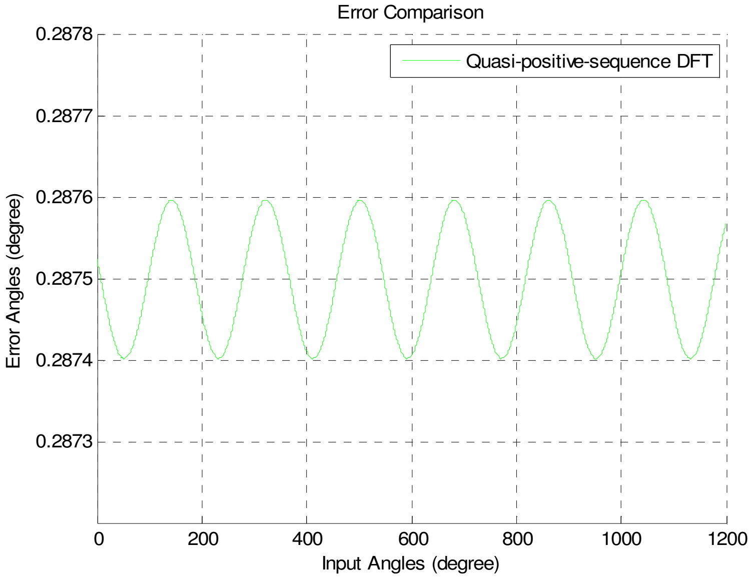


Fig. 4.22 Close-up Error Curve

It can be easily understood in the discussion and plots that the sine variation to fixed offset ratio of the error will be reduced to the order of the ratio of the frequency deviation to the nominal frequency by the cancellation function of the quasi-positive-sequence DFT method, which is the direct proof for the equations derived in the previous sections. The frequency deviation is in practice very small in power system applications, and therefore, by implementing the new angle algorithm stated in (4-21) and (4-31), the accuracy of the single-phase angle measurement will be improved conspicuously, which will be examined in the next section of this paper.

Experiments as well were performed in the Virginia Tech power laboratory to examine and verify the analysis and the results from the previous sections. The experimental settings are shown in Figure 4.23. A Protection Relay Test Set with GPS synchronization function is used as the signal generator, which provides a testing signal to three FDRs downloaded with three different firmwares: conventional DFT algorithm, quasi-positive-sequence DFT algorithm, and quasi-positive-sequence DFT algorithm with offset compensation, respectively. The measurement results from the three FDRs are transmitted to and stored in a local data concentration server for analysis.

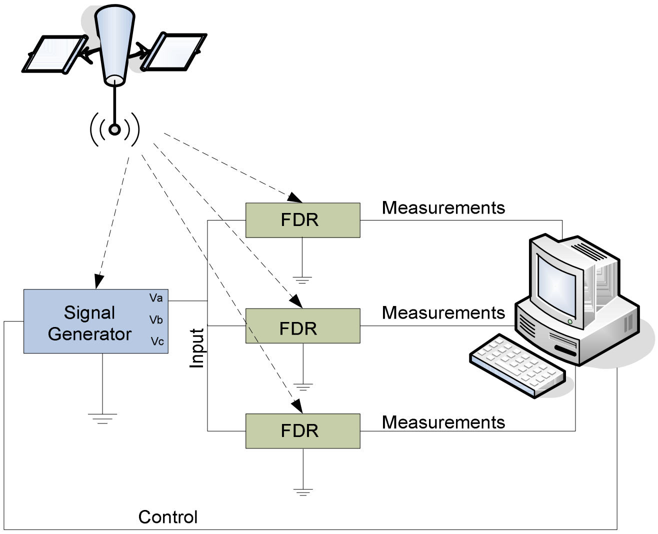


Fig. 4.23 Experiment Settings

In the first experiment, phase angle measurements were collected and ten seconds worth of data is plotted in Figure 4.24, with the frequency of the signal source set to 60.3 Hz. Clearly, there is an obvious sinusoidal component in the error of the conventional DFT algorithm, and the sinusoidal component is eliminated when using the quasi-positive-sequence DFT algorithm, which is considered to be an immediate affirmation of the previous explications and simulations. The error can be further cancelled to almost zero by deducting the fixed offset from the measurement after calculating the frequency based on the derivative of the angle measurements. The discrepancies between the experimental results and the theoretical values calculated by the aforementioned equations are shown in Table 4.1. The satisfactory compliance of the experimental results with the theoretical values is clearly manifested in the table, and it can also be seen that the errors of the experiment results are as small as about 0.02 degree, which can be legitimately explained by the hardware resolution limit and the GPS signal accuracy. Keep in mind that the Pulse per second (PPS) of GPS signal is 1 us in precision, which is equivalent to 0.022 degree in 60 Hz nominal frequency power systems.

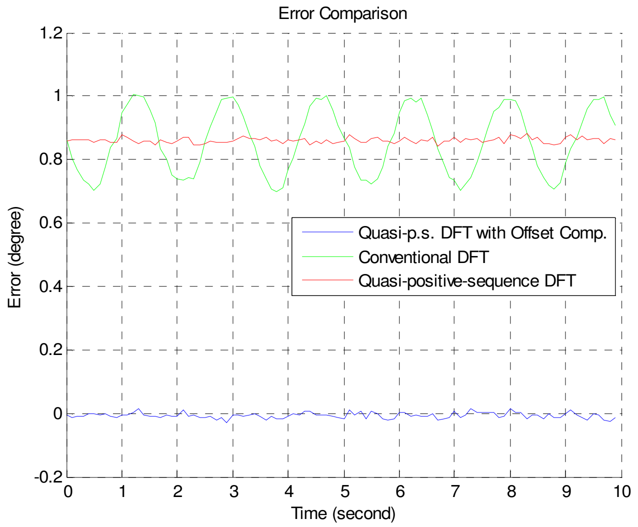


Fig. 4.24 Plot of FDR Angle Errors at 60.3 Hz

	Experiment Results	Theoretical Values
Sine Variation (peak-peak value)	0.2962	0.2862
Fixed Offset	0.8606	0.8625
99.7% Precise Deviation	0.0276	0

Tab. 4.1 Summary and Comparison of the Results (in degrees) at 60.3 Hz

Nearly identical procedures were conducted in the second experiment, except that the setting of the input frequency was changed to 59.5 Hz. Phase angle measurements were likewise collected and the same amount of data is plotted in Figure 4.25. The discrepancies between the experiment results and the theoretical values are similarly shown in Table 4.2. The same phenomena can be observed in Table 4.2, and they can further substantiate the validity of the cancellation effect of the quasi-positive-sequence DFT algorithm on the sinusoidal part of the systematic error, as well as the compensation effect of the offset balancing method on the fixed offset part of the systematic error.

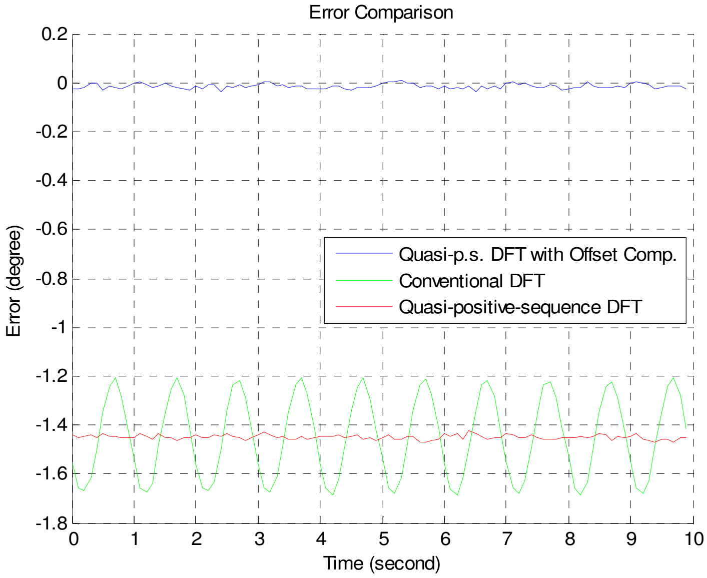


Fig. 4.25 Plot of FDR Angle Errors at 59.5 Hz

	Experiment Results	Theoretical Values
Sine Variation (peak-peak value)	-0.4679	0.4770
Fixed Offset	-1.4457	-1.4375
99.7% Precise Deviation	0.0298	0

Tab. 4.2 Summary and Comparison of the Results (in degrees) at 59.5 Hz

The total vector error (TVE) shown in this section, as small as about 0.035% and 0.053% after implementing quasi-positive-sequence DFT and the offset compensation, sufficiently meets IEEE synchrophasor standard C37.118, which stipulates that TVE should be less than 1% when frequency excursion reaches 0.5 Hz during extreme power system upsets[17]. When the conventional DFT algorithm is utilized, however, the TVE is 0.82% and 2.54% individually for the sinusoidal component and the fixed offset of the error, which closely approaches and well surpasses the limit, respectively. This effect exceptionally evinces the efficacy of the proposed methods in terms of reducing the error and the necessity of adopting those methods to power system applications when satisfying the IEEE standard is concerned.

4.6 Conclusion

The phase angle error of the conventional FFT algorithm is analyzed and new solutions that can rectify the error are presented in this chapter. The following conclusions have been drawn with adequate theoretical analyses, MATLAB simulations, and laboratory experiments:

1) A systematic error is present in the DFT phase angle measurements caused by the frequency excursion when the power system frequency deviates from the nominal value.
2) The error comprises two components, the fixed offset and the sinusoidal variation, if the input signal is single-phase voltage. However, it only contains the fixed offset component when the input is three-phase voltage.
3) Both of the two components of the error are proportional to the frequency excursion, and the error will exceed the limit stipulated in IEEE standard if no measure is taken to correct the error.
4) The sinusoidal component can largely be alleviated by the quasi-positive-sequence DFT algorithm and the fixed offset can be eliminated by compensation after the frequency is obtained based on the uniformly shifted phase angles.
5) The numerical simulations prove the effectiveness of the proposed methods to rid the measurements of the error, and the laboratory test results explicitly exhibit the satisfying performance of the proposed methods in a practical system.

Chapter 5: Aspects of Disturbance Triggering and Disturbance Location Estimation

In the demonstration of the utility of distribution level wide-area measurements, system frequency is of capital importance due to the ability to accurately estimate the parameter at virtually any voltage level in the interconnection and the faithfulness to reflect the system dynamic behavior even when the severe disturbances, such as fairly large amount generation trip, loading shedding and not satisfactorily mitigated interarea oscillation are present. Power system frequency is theoretically homogenous all across different parts of an interconnection system during steady state, and is practically very close to the theoretical condition with inappreciable fluctuations in different parts. But this is not the case during system dynamics caused by disturbances which induce active power imbalance and the system is operating without homogenized frequencies at different parts of the system before the new equilibrium of active power has been achieved. At this period of time, system is dominated by the swing equation, which describes the association between the generator rotor speed change and the active power imbalance. Further study of the power system in a continuous mode reveals that the frequency deviation propagates as electromechanical wave[21][22][23][24][25][26][27], or so-called 'Frequency Wave'. Observations from synchronized wide-area frequency measurement find that, during the transient period, the rate of change of system frequency is related to the distance between the measurement point to the disturbance location, or disturbance source. The distance here is defined as electrical distance which is generally inversely proportional to the inductance between the 'source' and the measurement. For the simplicity of the research, we assume the electrical distance is proportion to the geographic distance between the source and the measurement point. Using the frequency wave information to estimate the location of the disturbance source is of great research interest in FNET applications. This chapter examines the meaning of changes in system frequency in a wide-area context. Classically, due to a general lack of high-resolution frequency data, 'accurate measurement of system response is difficult unless the frequency deviation... is significant[28].' However, using the FNET system, we demonstrate the ability to detect power system disturbances, classify the disturbances, and estimate the magnitude of the disturbances.

5.1 Disturbance Trigger

When a generator trip or a load rejection occurs in the electric grid, the imbalance between generation and load causes sudden frequency changes within the system. Typical frequency excursions of generation-loss and load-drop events in three North America Interconnections manifest themselves in different fashions specific to their particular properties. Although all the generation-loss events have sharp frequency drops and load-drop events, sudden frequency rises, frequency excursions in three Interconnections demonstrate quite distinct characteristics. The difference indicates different generation/load characteristic, and control schemes in the three Interconnections[29]. The comparison of frequency excursions of generation-loss and load-drop events in three Interconnections is exhibited in Figure 5.1 and Figure 5.2.

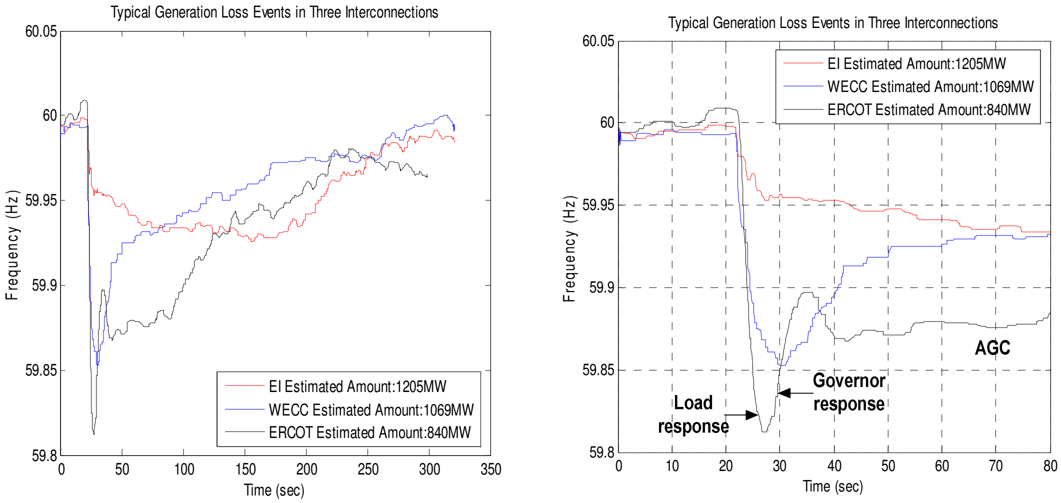


Fig. 5.1 Generation-loss Like Events in Three Interconnections

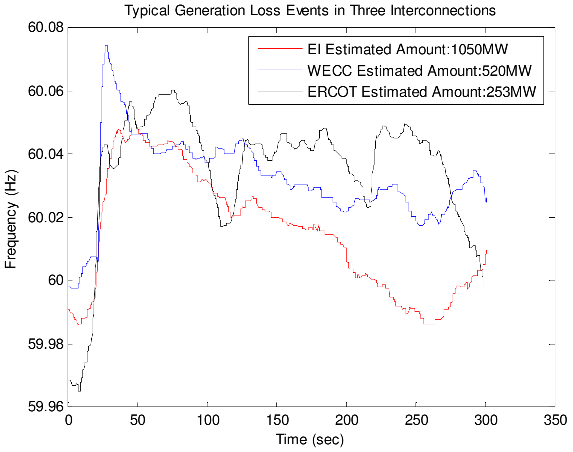


Fig. 5.2 Load-drop Like Events in Three Interconnections

The generation/load loss amount can be estimated from the equation:

$$\Delta P = \beta \Delta f \qquad (5\text{-}1)$$

where β is the frequency response characteristic[30] or load-frequency sensitivity coefficient[31]; Δf is the total frequency change amount following the disturbance by $\Delta f = f_{ss2} - f_{ss1}$, where f_{ss1} is the pre-disturbance steady state frequency, f_{ss2} is the post-disturbance steady state frequency. The amount of frequency change from certain amount of generation/load loss varies in different Interconnections, because the EI, the WECC and the ERCOT systems have different β values, which can be found in [30].

Frequency excursions also indicate characteristics of load profiles and control practices in the three Interconnections. As in the right image of Figure 5.1, the rate of change of frequency is not constant following the initial drop. Rather, the rate decreases after 4 to 5 seconds following the trip because of the load sensitivity to frequency[32]. When frequency drops, motor-type loads will slow down, which results in less load-generation mismatch and therefore the rate of frequency change decreases. Then the governor control of generators starts to arrest the frequency decline within 10 seconds and tends to halt it, a procedure known as primary control.

Governor control can bring frequency back to some extent but it does not necessarily recover frequency to the pre-disturbance level. The Automatic Generation Control (AGC) actions and reserve development, known as secondary control, continue to recover the frequency by replacing the amount of lost generation, which generally takes 5 to 15 minutes.

Therefore, the frequency's rate of change can be used as an indicator for grid disturbances as mentioned above. Likewise, the difference between pre- and post-disturbance frequency averages can also be utilized as a criterion to distinguish the abnormality of the event condition and normal operation condition.

The frequency deviation method is illustrated in Figure 5.3, where it shows the method continuously calculates the frequency average of the time period T3 that is immediately prior to the current time, and the frequency average of the time period T1 that is T2+T3 interval before the current time, recording them as f_{post} and f_{pre} respectively. The algorithm then proceeds to calculate the frequency difference between the two averages as Δf . If the frequency difference exceeds a predetermined threshold, then an event has occurred. The time periods T1 and T3 are to reduce the noise to an acceptable amount; the time period T2 is to cover the frequency-change time of an event in order for the frequency averages to faithfully reflect the pre- and post-disturbance conditions.

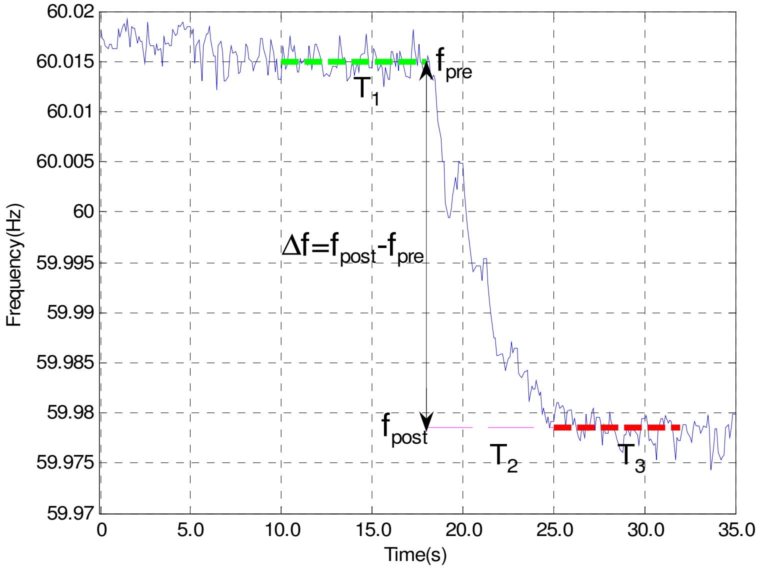


Fig. 5.3 Schematic Diagram of the Frequency Deviation Method

A chart explaining the rate of change of frequency event trigger algorithm is shown in Figure 5.4. This example assumes the use of two FDRs. However, in general, a trigger can be created using data from n FDRs. (Note: It is important to understand that each of the n FDRs should be connected to the same electrical interconnection.) This on-line event trigger algorithm can be described as follows:

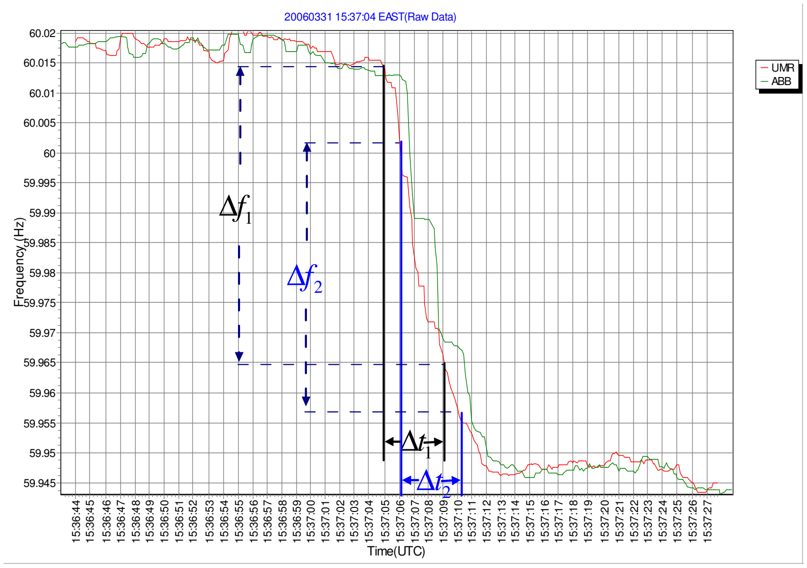


Fig. 5.4 Schematic Diagram of the Event Trigger Algorithm

1. Perform data conditioning using a moving median filter or other data smoothing technique and save the data in a buffer.

2. Calculate $\frac{df}{dt_1}, \frac{df}{dt_2}, \cdots, \frac{df}{dt_n}$ where:

$$\frac{df}{dt_i} \equiv \frac{\Delta f_i}{\Delta t_i} \qquad (4\text{-}1)$$

3. If: $\left|\frac{df}{dt_1}\right|, \left|\frac{df}{dt_2}\right|, \cdots, \left|\frac{df}{dt_n}\right| \geq \tau$ (4-2)

where τ is an empirically determined threshold and n is the number of FDRs used, then the software triggers on the event.

This event trigger is applied continuously to incoming frequency data as it arrives at the IMS. The trigger allows for the capturing of frequency excursions that are further used for the location of the hypocenter of the inciting event.

Interconnection	EUS	WECC	ERCOT
Threshold	0.005 Hz/Sec	0.006 Hz/Sec	0.008 Hz/Sec

Tab. 5.1 Thresholds for the Three Interconnections

The thresholds for each interconnection are different since the inertias of different interconnections are various. The same amount of generation trip or load rejection will cause different rates of change of system frequency. In another word, the load-frequency sensitivities of each interconnection are different. After calculating several months' worth of data for EUS, we found that the rate of change of system frequency is mostly within ±0.002 Hz/sec and seldom reach ±0.004 Hz/sec in normal operation conditions. Correspondingly, 0.005 Hz/sec is employed by FNET as the threshold for EUS. It works well and most of the events detected by the filter are obviously disturbances and some of them are confirmed by power utilities. Table 5.1 lists the threshold for each interconnection:

5.2 Trip Amount Estimation

The estimated amount of tripped generation or load shedding is based on the relationship between frequency change and active power balance.

Consider the generator depicted in Figure 5.5:

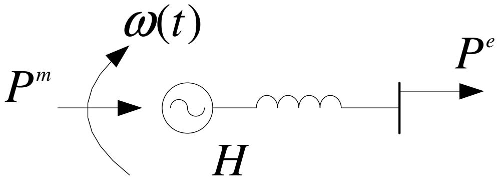


Fig. 5.5 Single Machine Electromechanical Diagram

The dynamics of this system are governed by an extension of Newton's second law of motion applied to rotating masses, the Swing Equation:

$$\frac{2H}{\omega_0}\frac{d\omega(t)}{dt} = P^m - P^e \tag{5-3}$$

where P^m is the mechanical power injection in megawatts (MW) resulting from a torque applied by the machine's turbine onto the generator's shaft.

The parameter P^e is the electrical power in megawatts(MW) withdrawn by an electrical load applied to the generator's terminal. The machine's inertia is represented by H with units of megajoule per megavoltampere (MJ/MVA or seconds) and its rotational velocity in radians per second is $\omega(t)$. Synchronous frequency of the electrical system is denoted by the parameter ω_0. Given the simple system of Figure 5.5 and the Swing Equation (5.3), it becomes quite easy to determine the magnitude of any power mismatch experienced by the generator based upon frequency measurements assuming all system parameters are known.

An interconnection, however, includes many machines and many loads. Therefore, the theory of single-machine single-load behavior needs to be extended to multiple machines connected through some network to multiple loads.

Because the power system has the attribute of superimposing a machine or load with the physics laws being the same, we can construct an aggregate model of the system through appropriate summations as follows:

$$\frac{2H_{system}}{\omega_0}\frac{d\hat{\omega}(t)}{dt} = P^m{}_{system} - P^e{}_{system} \qquad (5\text{-}4)$$

where $H_{system} = \sum_{i=0}^{N} H_i$, $P^m_{system} = \sum_{i=0}^{N} P^m_i$, $P^e_{system} = \sum_{i=0}^{M} P^e_i$, $\hat{\omega}(t) = \frac{1}{p}\sum_{i=0}^{p}\omega_i(t)$.

Equation (5-4) provides a method of describing an entire interconnection aggregated into a single machine as shown in Figure 5.6, which is very similar to Figrure 5.5 with the difference of substituting the system or aggregate components and parameters for the individual components and parameters.

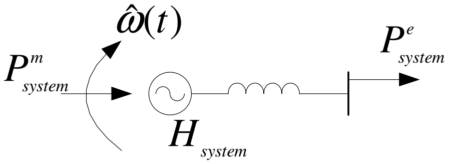


Fig. 5.6 Aggregate Representation of a Synchronous Electrical Interconnection

Because the aggregate swing equation governs the system frequency change all through the time frame of transients, the frequency change over the transient period or the pre- and post-disturbance frequency difference, which is the integral of the instantaneous frequency derivative, generally can be used to estimate the power imbalance. Although the governor determines the final post-disturbance frequency in that it is the generators that increase or decrease the prime mover input to compensate the power imbalance, the swing equation still dictates the linear relation between the frequency change and the trip amount from the point of view of physics. The frequency characteristics of load may affect the frequency change to some extent, but their effect is almost negligible compared to that of the generators.

From the perspective of the governors' effect, we can also posit that the amount of tripped generation is proportional to the frequency drop, and can be calculated by the area frequency response characteristic β and the frequency change.

When an electrical load change occurs, the turbine-generator rotor accelerates or decelerates, and frequency undergoes a transient disturbance. Under normal operating conditions, the rotor acceleration eventually becomes zero, and the frequency reaches a new steady-state, which is shown in the steady-state frequency-power relation for turbine-governor control:

$$\Delta p_m = \Delta p_{ref} - \frac{1}{R}\Delta f \qquad (5\text{-}3)$$

where Δf is the change in frequency, Δp_m is the change in turbine mechanical power output, and Δp_{ref} is the change in a reference power setting. R is called the regulation constant.

The steady-state frequency-power relation for one area of an inter-connected power system can be determined by summing (5-3) for each turbine-generating unit in the area. Noting that Δf is the same for each unit,

$$\begin{aligned}\Delta p_m &= \Delta p_{m1} + \Delta p_{m2} + ...\\ &= (\Delta p_{ref1} + \Delta p_{ref2} + ...) - (\frac{1}{R_1} + \frac{1}{R_2} + ...)\Delta f \qquad (5\text{-}4)\\ &= \Delta p_{ref} - \frac{1}{R}\Delta f\end{aligned}$$

where Δp_m is the total change in turbine mechanical power output, and Δp_{ref} is the total change in a reference power setting within the area. We define the area frequency response characteristic as $\beta = (\frac{1}{R_1} + \frac{1}{R_2} + ...)$.

It is apparent that since Δp_{ref} is fixed during the short time period of an event, the Δp_m, also interpreted as the tripped generation, is directly proportional to the change in frequency.

Referencing the North American Electric Reliability Corporation's (NERC) frequency response standard (example given in Figure 5.7), the following points in Figure 5.4 are defined as:

- Point A: Interconnection frequency immediately before disturbance.
- Point B: Interconnection frequency at the point immediately after the frequency stabilizes due to governor action but before contingent control area takes action.
- Point C: Point of maximum frequency excursion due to the loss of rotating kinetic energy from the interconnection.
- Point D: Interconnection settling frequency 60 seconds after contingency.

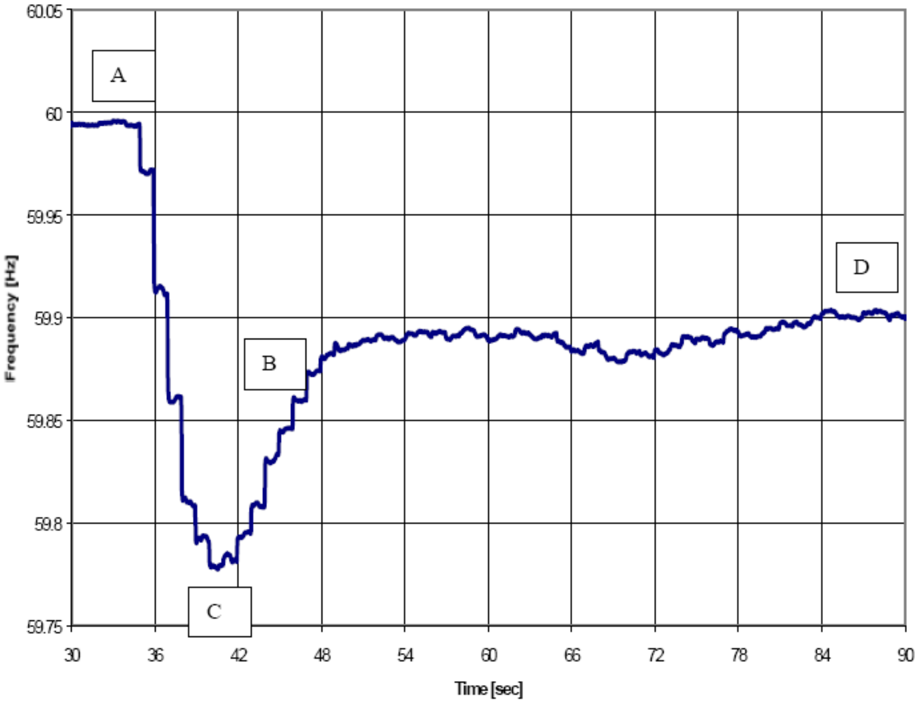


Fig. 5.7 Diagram of the NERC Frequency Response Standard

Because point C is indicative of the transient nature of the system, it should not be used as the post-disturbance frequency. The frequency values between point A and point B also should not employed in any way to calculate the post-disturbance frequency of an event. In fact, it is better for only the data after point B to be utilized in the process of calculating the post-disturbance frequency.

Therefore, in FNET applications, the first step of quantifying the trip amount is to locate point A as the event time. The second step is to calculate the average of the frequency averages of 4 seconds worth of data prior to point A of all the FDRs in the interconnection where the event occurs and assign it as the pre-disturbance frequency. Then the average of the frequency averages of 4 seconds worth of data of all the FDRs in the interconnection between 5 seconds after point A and 9 seconds after point A is calculated and saved to a variable as post-disturbance frequency. Five seconds after point A is the empirical value to avert the effect of the transient frequency data, and it works well to avoid the impact of the overshoot of frequency upon the calculation of the post-disturbance frequency. After the two frequency averages are obtained, the multiplication of the difference of the frequency averages and the area frequency response characteristic β will yield the estimated trip amount. Table 5.2 lists the β values of each interconnection:

Interconnection	EUS	WECC	ERCOT
β	0.005 Hz/Sec	0.006 Hz/Sec	0.008 Hz/Sec

Tab. 5.2 Beta Values for the Three Interconnections

5.3 Event Location Estimation

Power system frequency can tell volumes about the health and state of a power system. Moreover, the manner in which angle and frequency perturbations travel throughout the power grid has been extensively studied in the past[33]. Such perturbations are known to propagate throughout the grid as a function of time and space[34]. Using measurements from FNET, one can observe the electromechanical wave signature of power system events in frequency data. In this section, a brief analysis of several generator trip events is presented. The analysis comprises several event location methods and related observations. The ultimate goal of this research is to formulate a practical event location algorithm that can be used online to locate with sufficient accuracy the hypocenter of power system events irrespective of utility boundaries in real-time. We wish to take the first step in the direction of developing a reliable and comprehensive method of using real-time wide-area frequency measurements to locate approximately the location of power system disturbances, such as generator trips. The power system for this study is set in the EI.

5.3.1 Wave-front Arrival Time

Frequency perturbations have been shown to travel throughout the grid as electromechanical waves dispersing at finite (measurable) speeds[21][33][34]. Hence, the FDRs located throughout the grid detect said waves with unique time delays proportional to, among other factors, the physical distance between each respective unit and the disturbance location. We refer to these various detection times as “wave-front arrival times.” For indeed, these times correspond to the “arrival” of an electromechanical wave caused by the inciting event. Thus, the first step in event location is the determination of the wave-front detected by each unit and its corresponding arrival time.

Listed below is the approach that determines the wave-front arrival times using FNET’s FDRs.

First, after the on-line event trigger detects an event, three seconds worth of data is used to calculate the pre-disturbance frequency as shown in (5-5):

$$f_{pre} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{k}\sum_{j=1}^{k} f_{i,j}\right) \qquad (5\text{-}5)$$

where n is the total number of available units, k is the number of the data points in three seconds, i is the index of units, j is the index of data, $f_{i,j}$ is the corresponding j^{th} frequency measurement of unit i.

Second, a confidence threshold, ε, has been fixed at 8~14 mHz, which is justified empirically by a gracious plenty of previous event cases. Then, the wave-front frequency, the frequency at which we will note the event detection time for each unit, is calculated with f_{pre} and ε. This quantity can be defined as:

$$f_{wavefront} = f_{pre} - \varepsilon \text{ or } f_{wavefront} = f_{pre} + \varepsilon \qquad (5\text{-}6)$$

depending on the type of event at hand. Generator trips in general cause negative frequency excursions and load shedding in general causes positive frequency excursions.

Finally, the detection times are documented as the time when the measured frequency data crosses the line determined by $f_{wavefront}$. The flow chart of wave-front arrival time is shown in Figure 5.8.

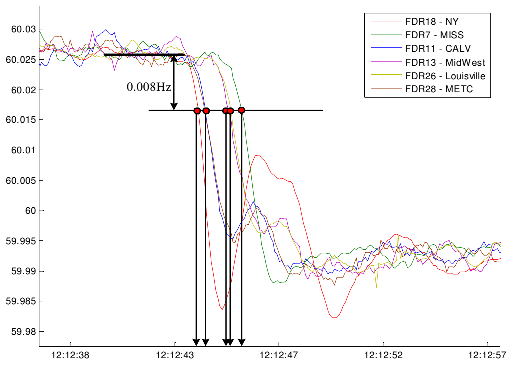


Fig. 5.8 Example of the Fixed Frequency Shift Wave-front Arrival Time Detection Method

5.3.2 The Event Location Estimation of Generation Trips

The Wave-front Arrival Time that FDRs detect electromechanical waves is called Time Delay of Arrival (TDOA) in FNET. In power grid monitoring systems similar to FNET, the proportionality will probably be affected by many factors, for example noise, topology of transmission lines, the algorithm to detect the frequency disturbances, and the precision of sensors. There is a very common pattern that the closer the units to the event location the more proportional their TDOAs to their distances to the disturbance location, which is justified empirically by many previous event cases.

The following figures demonstrate the apparent electromechanical wave propagation throughout the EI due to several generation trip events. The figures indicate generation units that have experienced a 14 mHz drop in frequency due to the generation trip. A complete list of generators of interest in the EI model can be found on the FNET server. This analysis is intended to provide a broad sampling of events from across the EI. It is this analysis that gives us inspiration for the use of wide-area frequency measurements in the location of large-scale (>500 MW) events.

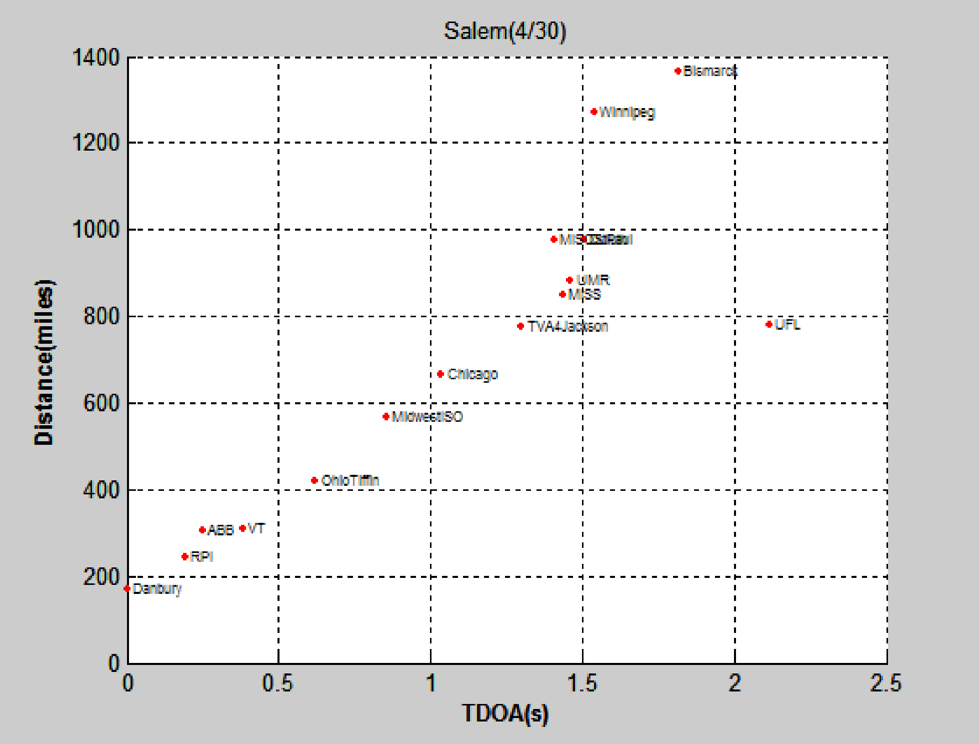


Fig. 5.9 Salem 4/30/2007 Event Distance vs. TDOA Chart

Figure 5.9 is a chart of Salem 4/30/2007 event. The X axis is the TDOA, which is the relative frequency disturbance detection times of all the units with reference to that of the first unit; the Y axis is the geographic distance from each unit to the actual event location. It can be seen from this chart that the dots of the units are approximately linearly distributed, which reflects the validity of the previous assumption.

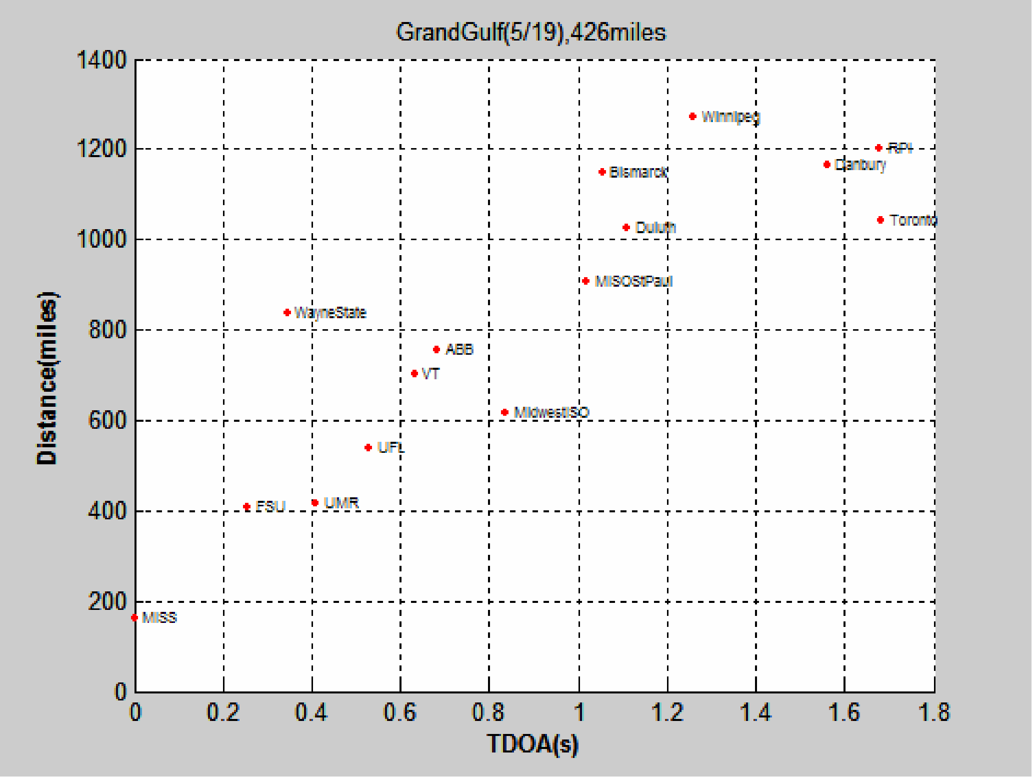


Fig. 5.10 Salem 5/19/2007 Event Distance vs. TDOA Chart

For most cases, however, the proportionality is not as satisfactory as the above one. Figure 5.10 is a chart of Grand Gulf 5/19/2007 event. In this case, the first several units are to some extent proportional, while the last several units deviate from the general trend of the former ones. According to the FNET event database statistics, almost 8 out of 10 cases can be categorized into this pattern as opposed to the previous more linear one. From those observations, it can be generalized that the units that are farther from the event location and/or detect the frequency change later can pollute the proportionality of the TDOA/distance of the units. In light of that phenomenon, it becomes imperative for us to use the first several units instead of all the units to eliminate the pollution so as for the algorithm to generate a better result for the estimation of the event location. Practically, the first 6 has been proven to be the optimal empirical number to utilize the said proportionality to execute the triangulation based on the cases accumulated in FNET.

According to previous assumption and observation, after the on-line event trigger detects an event, the first unit that sensors the frequency perturbation is usually the closest one to the actual event location. Therefore we can have the inference that the possible event location should be in the area that makes any spot in it be closest to the first detected unit. Based on this inference, it is to some degree necessary to use geometric method to determine the exact area that is able to satisfy that rule, but practically that will make too many complications out of this relatively less significant concern. In FNET triangulation application we can approximately get that area by draw a suitable circle around the first unit. Then how to decide the size of that circle comes to be an issue.

With engineering commonsense we can solve this issue by using as a reference the quotient of the total area of the whole electrical interconnection divided by the number of the units within. For instance, if a power interconnection is estimated to be 10000 square miles and there are on average 20 active units in it, we accordingly have 500 square miles per unit. Then it can be reckoned as each unit having 500 square miles as its territory. When a certain unit is detected as the first sensed unit, the 500 square miles around it is hypothesized as the possible area that the event takes place. The radius of that area can be calculated assuming the area to be a circle. With consideration of margin for error, the radius of the possible area around the first detected unit can be defined:

$$r = k\sqrt{\frac{S}{\pi N}} \qquad \text{(5-7)}$$

where r is the radius, S is the total area of the power interconnection, N is the unit number installed in that interconnection, and k is the margin coefficient.

From (5-7), it can be easily concluded: the more units in a power interconnection the smaller the possible area, and thus the more accurate our result.

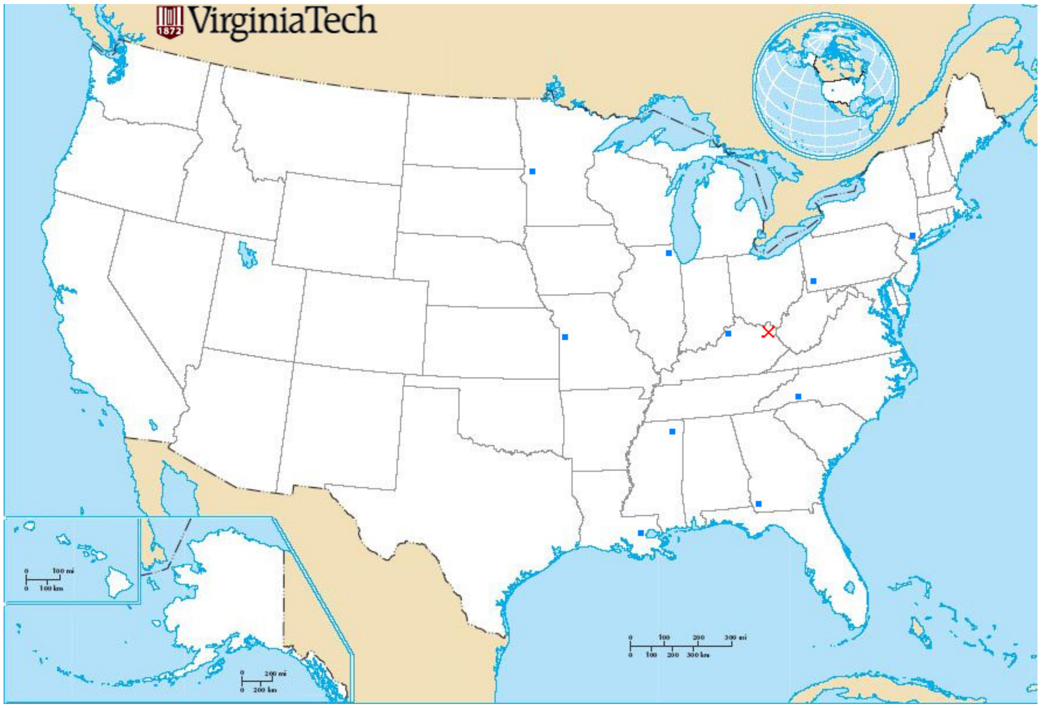
As far as the Eastern Interconnection is concerned, 200 miles is the r value that we can get from the total area of and current unit number in the Eastern Interconnection

Once the possible area in an interconnection is determined, if the event is diagnosed as a generator trip and the event trip amount is already computed, the possible points in the area can only be the power plants in the possible area that have generators whose capacity are over the trip amount. By ruling out power plants under trip amount, we can narrow down the possibilities to just a few locations of power plants and therefore make the algorithm more efficient.

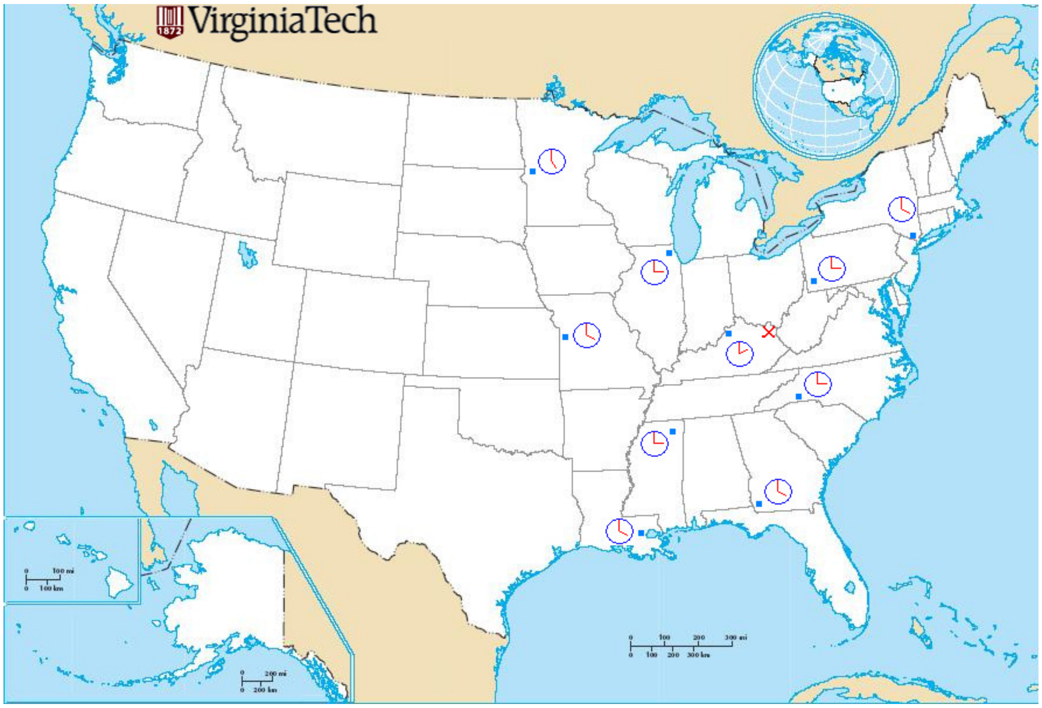
After the possible power plants are chosen, the proportionality of the TDOAs and the physical distances to the disturbance location of the first FDRs are employed to select the most probable location from those power plants. Linear regression is a popular approach to research linear phenomena, and the sum of the squares of the distances from the coordinates of each point to the linear-fit line of all the points can be used as the parameter that represents the proportionality of the points concerned. In this application, it can be assumed that each possible power plant to be the actual event location and then calculate the distances of the FDRs to the hypothesized location. The result can then be obtained by choosing the power plant that has the minimum sum of the squares after using the previously calculated distance and detected TDOA of each FDR.

From the above research and discussion, the generator trip triangulation algorithm can then be described as following steps.

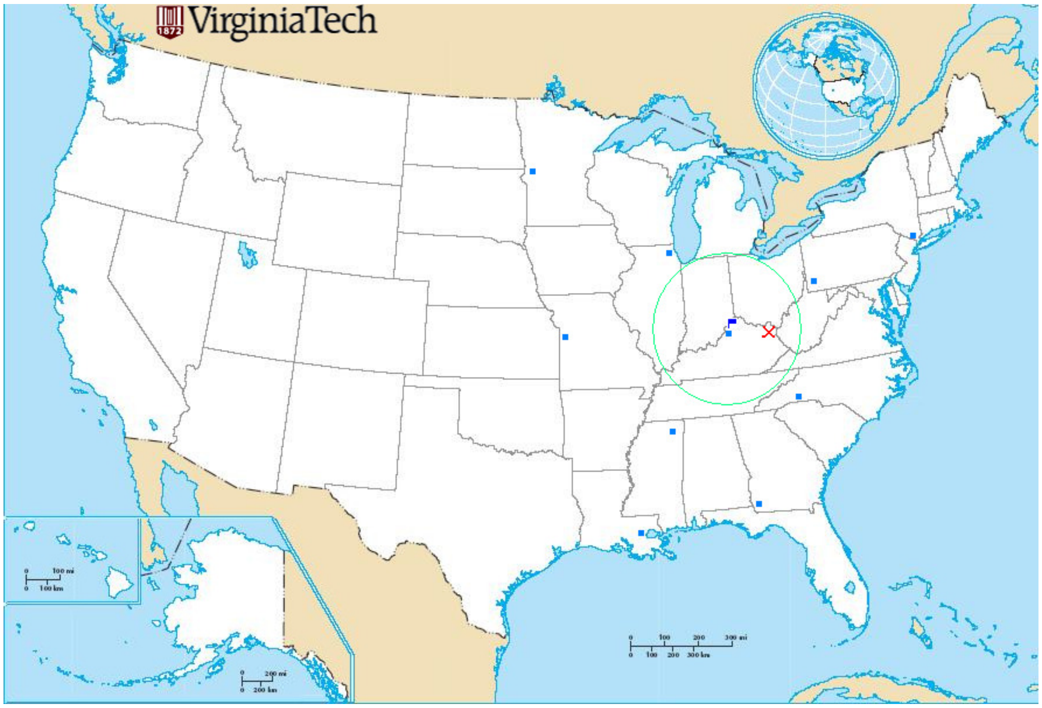
1) Assume an event occurs at the red crossing in EI and there are nine FDRs(blue dots) in EI.



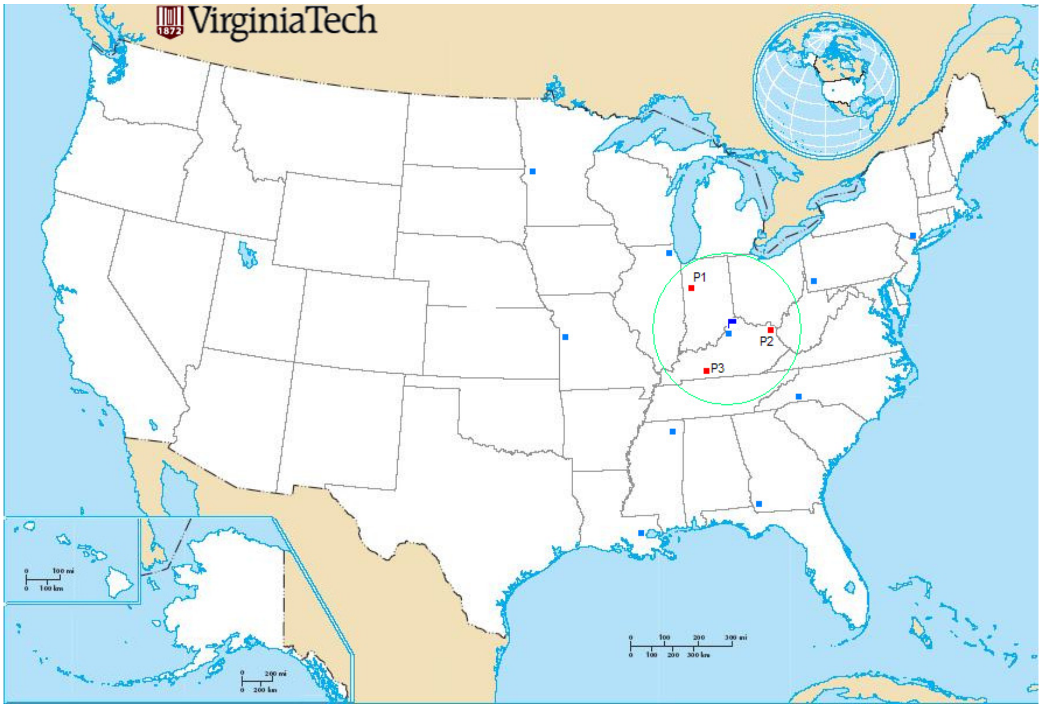
2) Calculate the TDOA of each FDR.



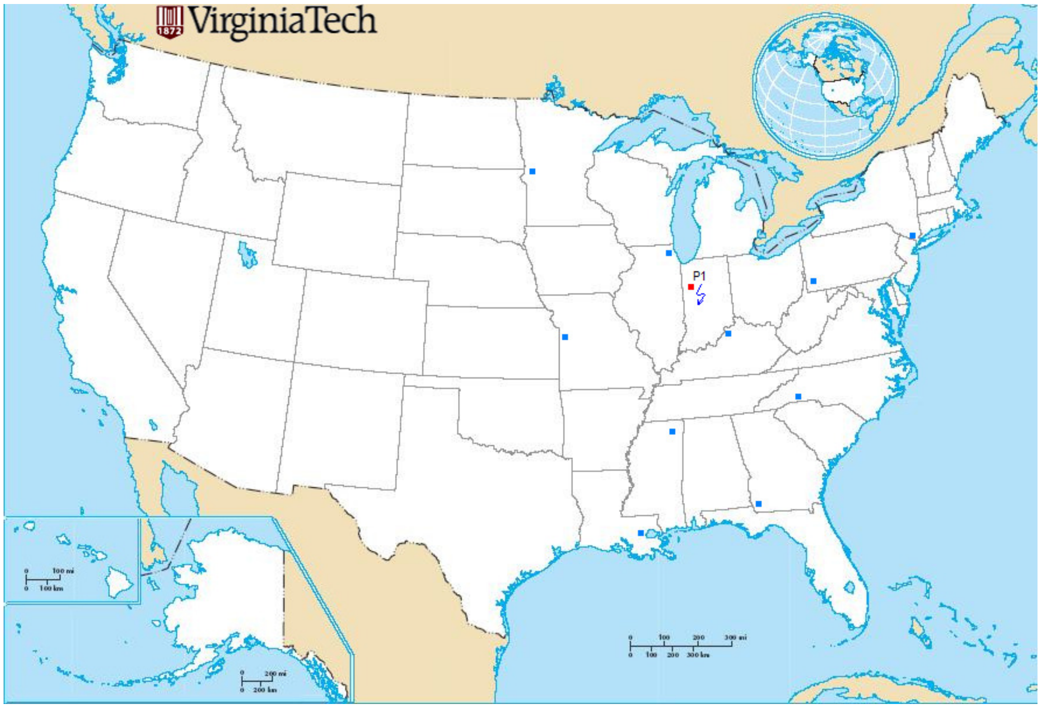
3) Find the first unit that detects the event, for example the one with flag, and then draw of a circle of a certain radius according to the FDR density in the power interconnection.



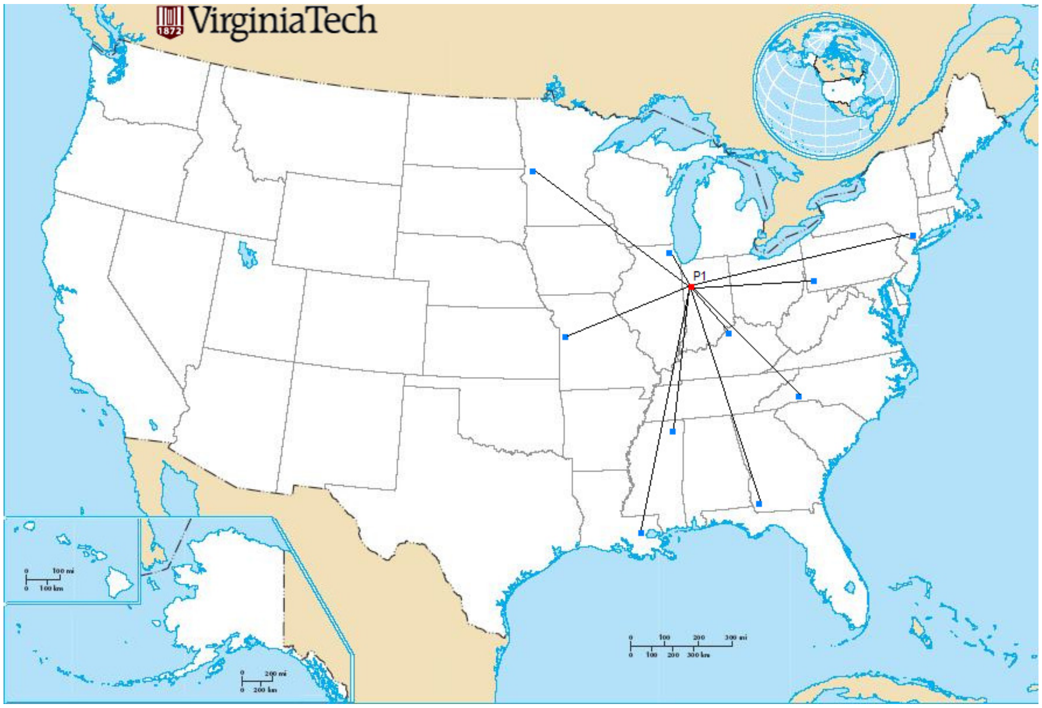
4) Find all the power plants that have generators whose capacity are over the trip amount, for example the red dots P1, P2, and P3.



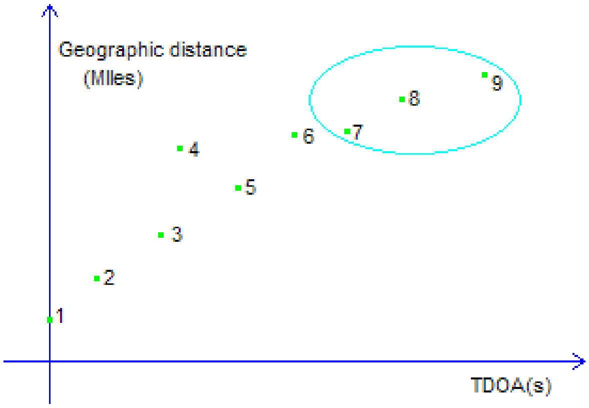
5) Assume the event occurs around plant P1.



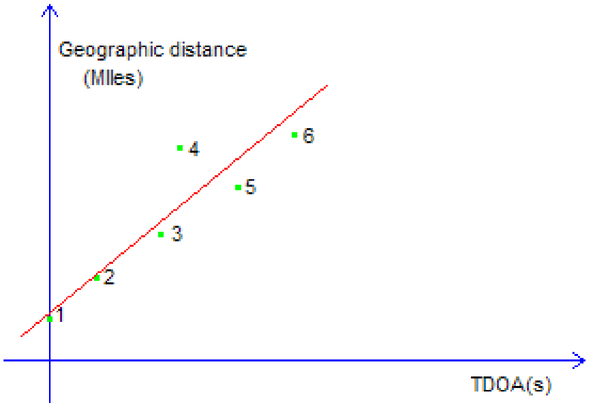
6) Calculate each geographic distance from each FDR to the hypothesized plant (P1).



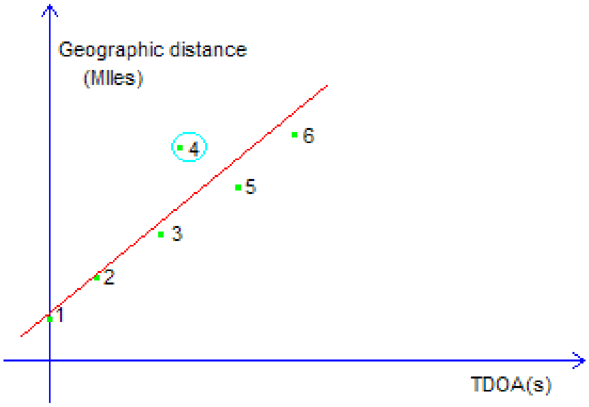
7) List first 6 units (TDOAs and distances to the hypothesized location on Cartesian co-ordinates) and omit the rest 3 units.



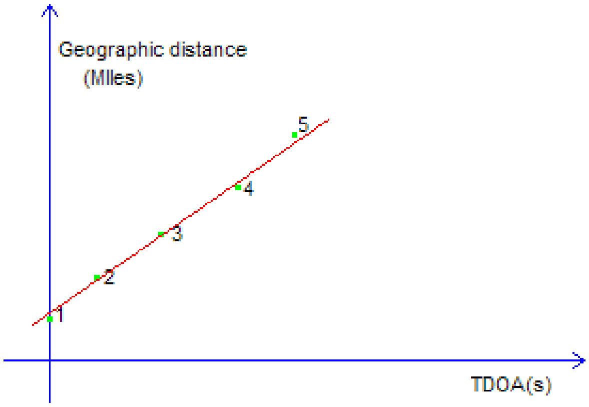
8) Do linear fit based on Distance/TDOA of the first 6 units.



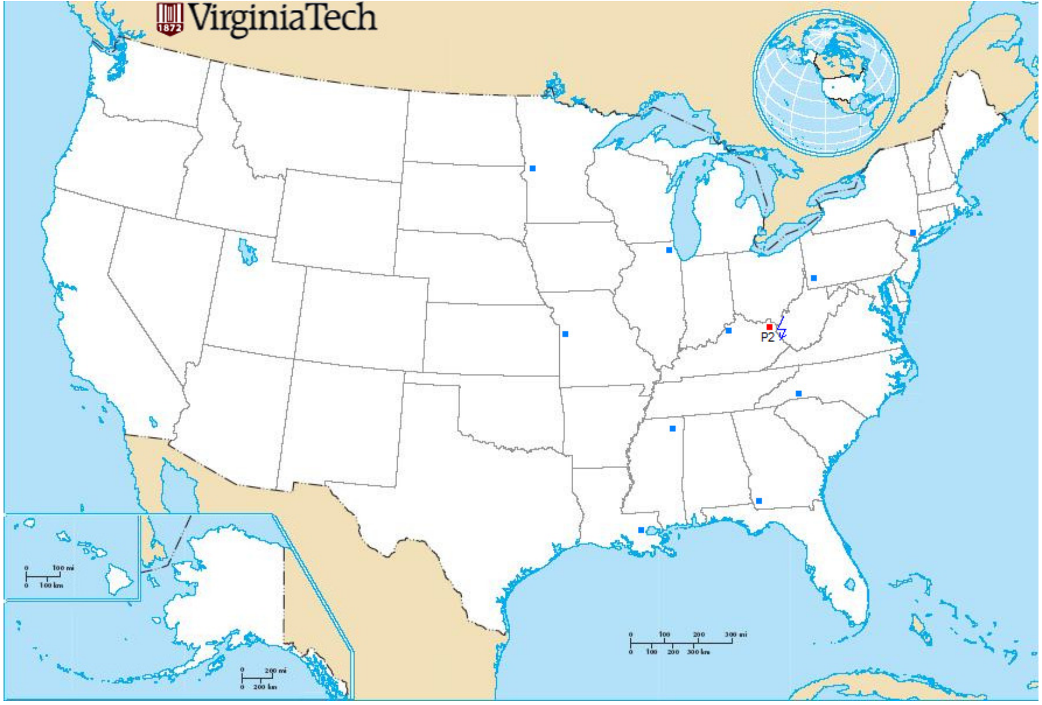
9) Delete the unit that is farthest from the linear-fit line.



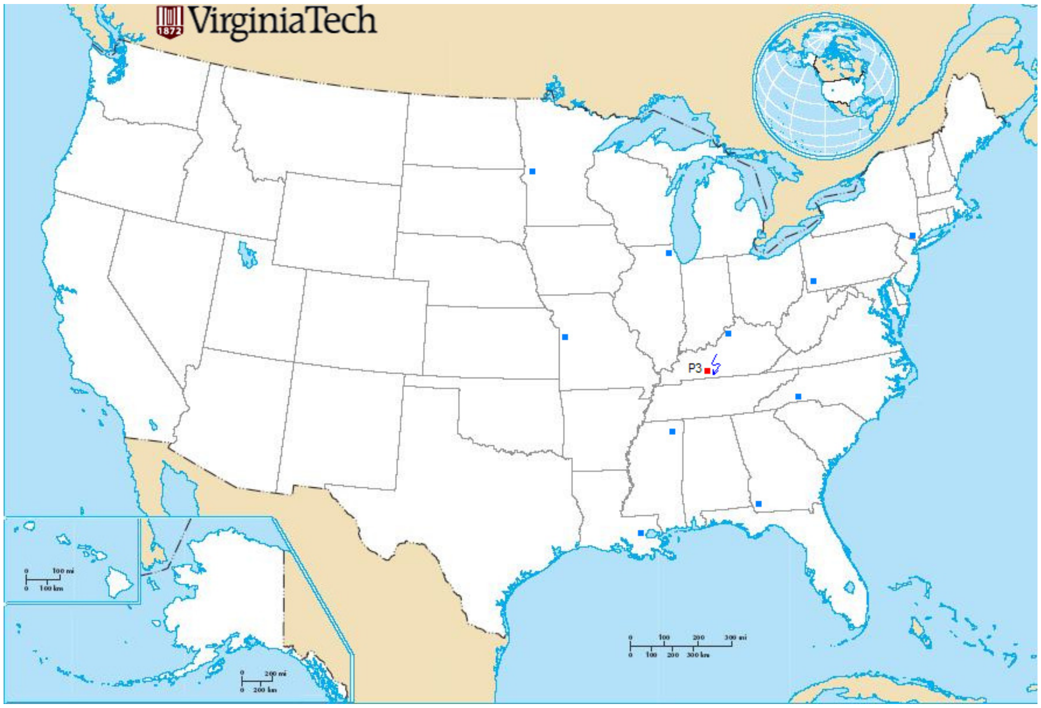
10) Redo linear fit based on the Distances/TDOAs of the remaining 5 units. Calculate the sum of the squares of the distances from the 5 units' coordinates to the linear-fit line.



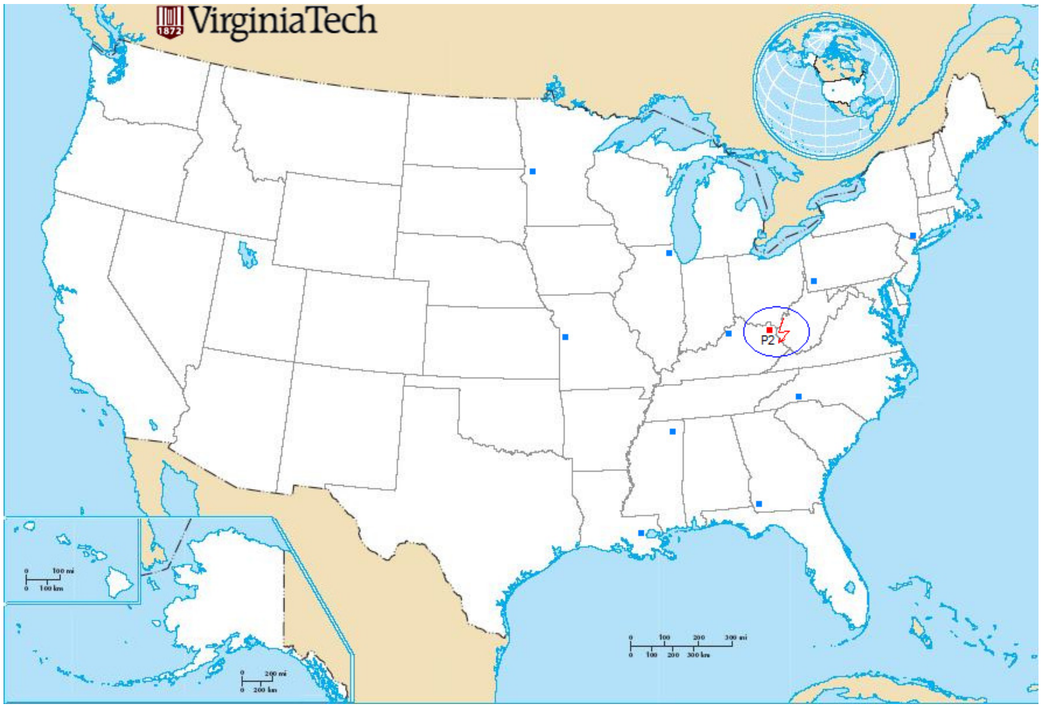
11) Assume P2 to be the event location, do the same procedure from step 6 to step 10 and obtain the new sum of squares.



12) Assume P3 to be the event location, do the same procedure from step 6 to step 10 and obtain the new sum of squares.



13) Find the least sum of squares and the corresponding plant is the estimated event location. For example, if P2 has the least squares, we estimate the area around it is the event location.



This method is referred to as the generating station hypothesis approach, and it is a topic of great research interest in FNET about the event location estimation.

5.3.3 Newton’s Method and Least Squares Method

For some unpredictable reasons, the aforementioned generation trip detection method may encounter certain issues and fail to provide location estimation. The algorithm above checks if the least sum of squares exceeds some limit; if so, it is then judged that the estimation procedures do not work adequately to meet the accuracy requirement of the result. On the other hand, there are load shedding (or pumped storage disconnect) events that in most cases do not occur in a specific power plant. Under those Circumstances, a more general method needs to be adopted to cope with them. The following content of this subsection addresses using Newton’s method or least square method to attempt pinpointing the event locations by an equation-solving means.

A, Newton’s Method

When an event occurs, the locations and the wave-front arrival times of the units are known. The coordinates of the event location (x_e, y_e), the latitude and longitude, are two unknown variables. Event time t_e is the third unknown variable. If we assume that the wave propagation speed V is unique in different directions (as past experience suggests), V will be the fourth unknown vector-valued variable in the event location estimation problem.

Since there are four unknown variables, one of which is vector-valued, standard linear solution methods such as least-squares will not work. Here, we discuss how to use the Newton's method to do the triangulation. There are more than thirty FDRs deployed in EUS. However in this method, we choose the first four units in the response sequence to find event location.

As shown in Figure 5.11, assume the event is located at (x_e, y_e), the event hypocenter, then the distances between event location and an arbitrary FDR location is provided by:

$$L_i^2 = (x_i - x_e)^2 + (y_i - y_e)^2 \qquad (5\text{-}8)$$

where L_i is the distance between FDR_i and the event hypocenter, and x_i and y_i are the latitude and longitude of FDR_i's location respectively.

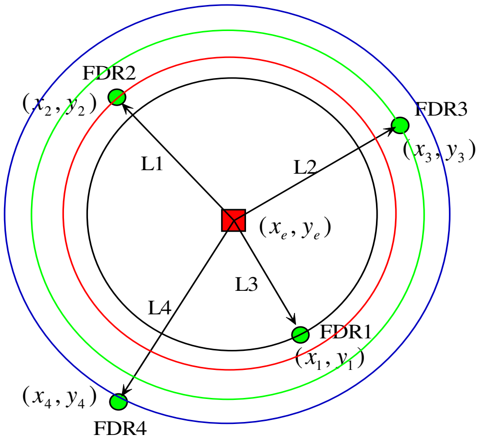


Fig. 5.11 Chart of the Fixed Frequency Shift Wave-front Arrival Time Detection Method

Distance can also be expressed by the product of speed and time:

$$L_i = V\left(t_i - t_e\right) \qquad (5\text{-}9)$$

where t_i is the wave-front arrival time determined by information from FDR_i and t_e is actual event time. Note that t_i is strictly larger than t_e since the event cannot be detected before it occurs. Hence we arrive at the event location estimation equations:

$$\begin{cases} (x_1 - x_e)^2 + (y_1 - y_e)^2 - V^2 (t_1 - t_e)^2 = 0 \\ (x_2 - x_e)^2 + (y_2 - y_e)^2 - V^2 (t_2 - t_e)^2 = 0 \\ (x_3 - x_e)^2 + (y_3 - y_e)^2 - V^2 (t_3 - t_e)^2 = 0 \\ (x_4 - x_e)^2 + (y_4 - y_e)^2 - V^2 (t_4 - t_e)^2 = 0 \end{cases} \tag{5-10}$$

Corresponding initial values of these unknown variables are needed to solve these non-linear equations using Newton's method. The averages of the locations of FDRs are set as the initial values of event location:

$$x_{e0} = (x_1 + x_2 + x_3 + x_4)/4$$
$$y_{e0} = (y_1 + y_2 + y_3 + y_4)/4 \tag{5-11}$$

The time initial is set to the time that is 0.5 second before the earliest wave-front arrive time:

$$t_{e0} = t_1 - 0.5 \tag{5-12}$$

The initial of wave speed is set as 200miles/sec, according to previous experience.

Now we can use Newton's method to calculate the four unknown variables, and get the estimated event location.

B, Least Squares Method

Newton's method discussed above uses only four FDRs' data to compute the event hypocenter, instead of using all of the available information. Moreover, it does not always reach a convergent solution for the set of non-linear equations.

Another triangulation method based on the least squares method is introduced that can use all of available FDRs' information and will almost always give a well conditioned result.

According to (5-8) and (5-9), for each responding FDR, we can write:

$$\begin{aligned}(x_1-x_e)^2+(y_1-y_e)^2&=V^2(t_1-t_e)^2\\(x_2-x_e)^2+(y_2-y_e)^2&=V^2(t_2-t_e)^2\\&\vdots\\(x_n-x_e)^2+(y_n-y_e)^2&=V^2(t_n-t_e)^2\end{aligned} \tag{5-13}$$

where (x_n, y_n) are the (x, y) coordinates of the n^{th} FDR to respond; (x_e, y_e) are the (x, y) coordinates of the hypocenter; t_n is the time at which the n^{th} FDR measures the electromechanical wave; t_e is the time at which the event occurred; and V is the mean velocity at which the frequency perturbation travels. Therefore, V is not a vector-valued variable in this method. The unknowns in the above equation set are: the hypocentral coordinates, (x_e, y_e), and the time at which the event occurred, t_e.

To find a set of solutions, we seek a linear system in terms of the hypocentral coordinates such that the least-squares method of solving an over-constrained system of equations can be used. In general, our system of equations will be over-constrained since many more FDRs typically respond than there are variables. By subtracting successive pairs of equations, a linear equation in terms of the hypocentral coordinates is produced:

$$(x_{i+1}-x_i)x_e-(y_{i+1}-y_i)y_e-V^2(t_{i+1}-t_i)t_e=C_i \tag{5-14}$$

where C_i is defined as:

$$C_i=\frac{1}{2}\left[V^2(t_{i+1}-t_i)+x_{i+1}^2+y_{i+1}^2-x_i-y_i\right] \tag{5-15}$$

Hence, (5-11) can be written for every unit that responds forming a system of equations that can be placed in a matrix to form:

$$\mathbf{C}=\mathbf{Hx} \tag{5-16}$$

with matrix variables defined as:

$$\mathbf{C}=\begin{bmatrix}C_1\\C_2\\\vdots\\C_n\end{bmatrix},\quad \mathbf{x}=\begin{bmatrix}x_h\\y_h\\t_h\end{bmatrix} \tag{5-17}$$

$$\mathbf{H} = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 & V^2\left(t_2 - t_1\right) \\ x_3 - x_2 & y_3 - y_2 & V^2\left(t_3 - t_2\right) \\ \vdots & \vdots & \vdots \\ x_n - x_{n-1} & y_n - y_{n-1} & V^2\left(t_n - t_{n-1}\right) \\ x_1 - x_n & y_1 - y_n & V^2\left(t_1 - t_n\right) \end{bmatrix} \tag{5-18}$$

To solve for $\mathbf{x}$ in (5-16), which is constructed to contain the hypocentral coordinates, we use the pseudo-inverse defined as:

$$\mathbf{H}^{\dagger} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T \tag{5-19}$$

The final least squares solution is:

$$\mathbf{x} = \mathbf{H}^{\dagger}\mathbf{C} \tag{5-20}$$

The least squares based triangulation method is currently used in FNET and has worked well as a complement for the method discussed in the previous subsection on over 1000 event location estimations performed since January 2006.

5.3.4 Adaptive Threshold and Oscillation Method

A, Adaptive Threshold

It has been proven in previous discussion and by many confirmed cases that the wave-front arrival time detection is of pivotal importance to the estimation of the event location. The electromechanical wave travels approximately at 400 miles per second, which connotes one second error in deciding the wave-front arrival time can possibly lead to 400 miles error in the event location estimation. Although the error in the wave-front arrival time detection of an individual FDR can be compensated to some extent by the remaining FDRs with correct detection times, the event location estimation still will more or less be skewed by the inaccurate time detection, especially the first wave-front arrival times that are found to respond to the disturbance the most quickly.

Among all the elements that affect the outcome of the wave-front arrival time determination, the confidence threshold, ε, which has been fixed at 8~14 mHz in the past, is foremost important. The power system transients of a disturbance, dictated by the operating mode, the generation-load unbalance and other contingent parameters, generally drive the frequency of each FDR along a very irregular curve, which complicates the determination of the wave-front arrival times and makes it difficult to set a suitable confidence threshold.

There was a generation trip at 11:15:09 on 10/30/2008 at FirstEnergy's Sammis #7 plant in South East Ohio, 40 miles west of Pittsburgh, PA and 50 miles North of Wheeling, WV. The left map is the online triangulation result of the server using 14 mHz as the confidence threshold. The red circle is the possible area in which the disturbance was estimated to occur and the blue dot is the feedback of the actual location provided by the utility. The verge of the estimation circle was nearly 200 miles from the actual location. Therefore the online estimation of this event is categorized as a failed case.

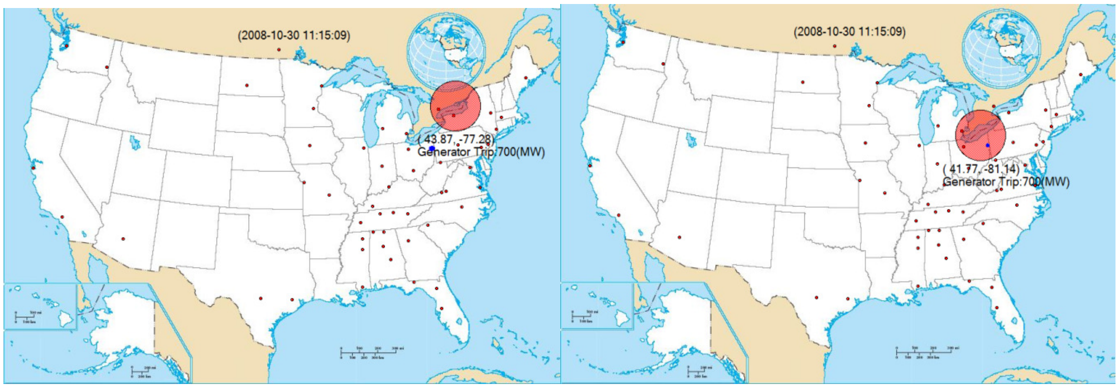


Fig. 5.12 Comparison of the Results by Different Confidence Threshold of the 10/30/2008 Event

The investigation of this event finds the failure of the estimation is caused by the unfitting threshold that determined the wave-front arrival times. The pre-disturbance frequency average is 59.994 Hz, which is shown as a green line in Figure 5.13.

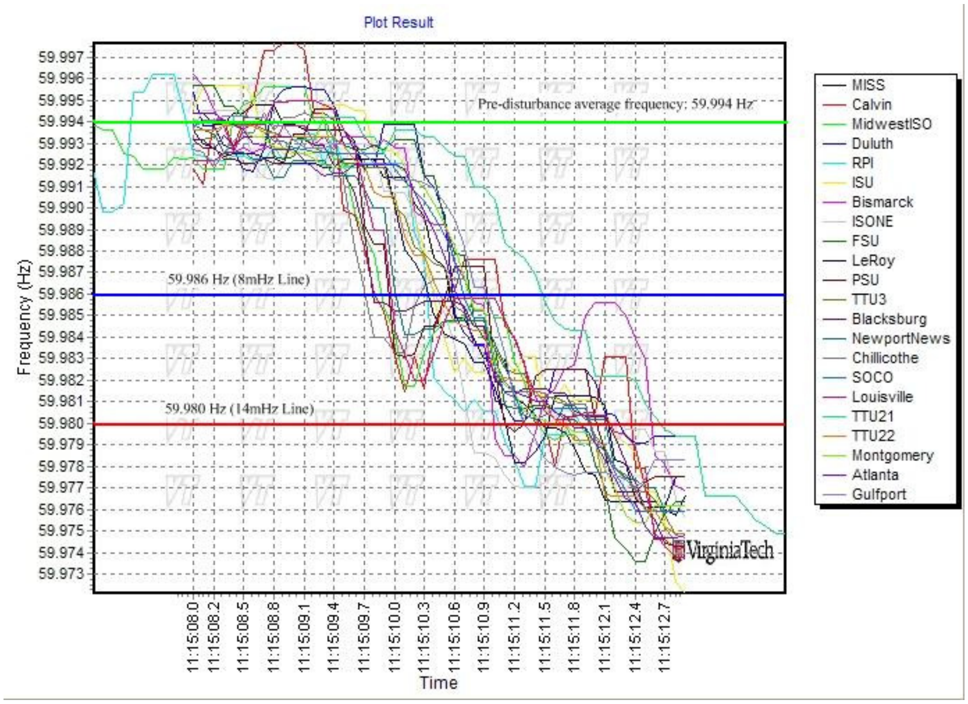


Fig. 5.13 Frequency Plots of the 10/30/2008 Event

When 14 mHz is adopted as the threshold, the cutting line for the wave-front arrival time determination will be red 59.980 Hz line in Figure 5.13. The first intersection where the frequency curve crosses the cutting line is calculated and written down as the wave-front arrival time of each FDR. As can be observed from the figure, the transients of the disturbance caused the frequencies of the FDRs to oscillate and made the sequence of the detected arrival times chaotic after a short period of steady frequency drop, which to a great degree affects the accuracy of the arrival time detection and renders the detected wave-front arrival times improper to represent the real times when the electromechanical waves arrive.

If 8 mHz is adopted as the threshold, however, to avoid the negative effect of the transients upon the wave-front arrival time detection, the cutting line will be the 59.986 Hz blue line, and the intersections of the frequency curve of each FDR crossing the cutting line apparently embody the sequence of the turning points of the FDR frequency signatures in Figure 5.13. The offline triangulation using 8 mHz as the threshold generates the estimation result in the right map of Figure 5.12. There is a great improvement in the accuracy of the estimation of the event location as the actual location in the blue dot is very close to the center of the red circle, which can undoubtedly be ascribed to applying a suitable threshold to the wave-front arrival time detection.

When the electromechanical wave propagates to an area, the frequency of the area changes in accordance with the electromechanical wave and the turning point of the frequency curve is the time when the electromechanical wave arrives. It is that the closer the cutting line to the pre-disturbance frequency average line, the more precise the wave-front arrival detection method quantifies the sequence of the real wave-front arrival times and the intervals between one another. It is therefore reasonable to think of using a threshold as small as possible to determine the arrival times so as to shield the detection of the electromechanical wave arrival times from the disturbance transients by setting the cutting line close to the pre-disturbance frequency, thus improving the accuracy of the triangulation program.

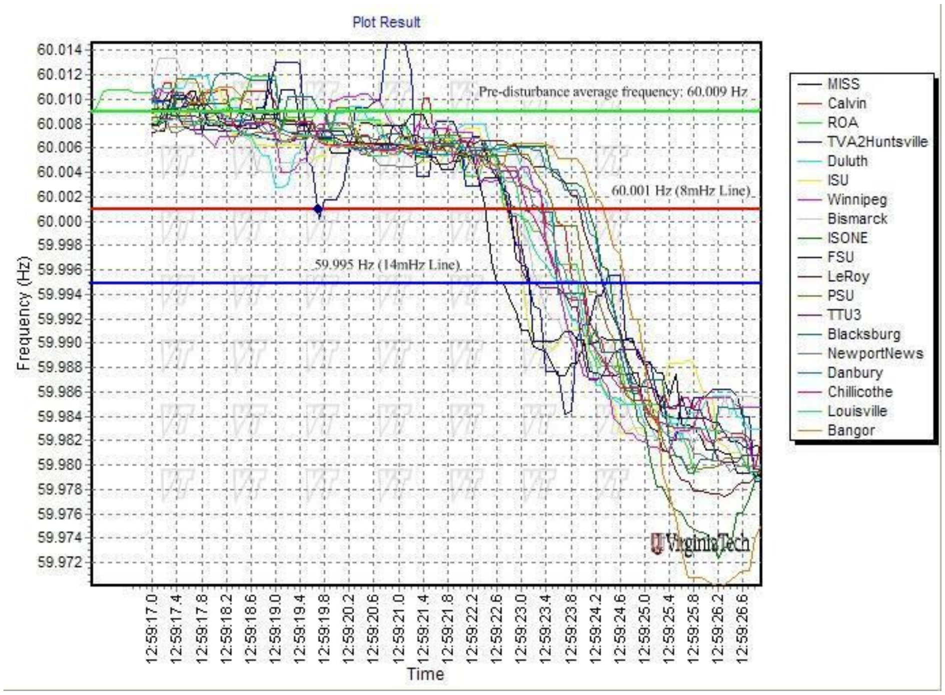


Fig. 5.14 Frequency Plots of the 06/17/2008 Event

However, power systems are not an ideal system without any frequency fluctuation and noise, and the FNET frequency measurement system is not a perfect system immune from errors and misrepresentations. Owing to those practical problems, the triangulation algorithm will not work successfully if the wave-front arrival times are not properly determined when the threshold is set too small. Figure 5.14 is a compelling example of the fact that too small a threshold is as detrimental to the performance of the arrival time detection as too large a threshold. As can be seen from the frequency plots, the pre-disturbance frequency is 60.009 Hz in a green line, and the 8 mHz/14 mHz cutting lines are the red and blue lines respectively. Because the transients of the disturbance start later than the 10/30/2008 event in Figure 5.13, which may be caused by different operating mode, different disturbance origin location or some other system factors, the intersections along the 14 mHz cutting line are all before the transients begin and therefore correctly and clearly express the sequence and the intervals between one another of the wave-front arrival times. On the other hand, if the cutting line is 8 mHz from the pre-disturbance frequency line, the TDOA of the MISS FDR will be assigned to be the intersection of the red line and the MISS frequency curve, the blue dot shown in Figure 5.14, which is obviously an error created by the noise of the frequency data of the MISS device. The error is so ahead of the cluster of the frequency curves caused by the event that it will definitely botch the triangulation algorithm.

There consequently has to be a balance to strike between avoiding the noise of the measurement and preventing the impact of power system transients. The solution to this issue in the FNET triangulation application is the adaptive threshold, the essence of which is to try another threshold after a certain threshold does not give a good result instead of setting a fixed threshold for the entire application. The criterion used to judge if the threshold does not work is to check if the residue of the linear regression exceeds a preset value. The algorithm attempts 3 thresholds, 14 mHz, 10 mHz and 8 mHz in sequence. The algorithm tries 14 mHz first. If the residue is within limit, the program reports the triangulation result, whereas if the residue is over the limit, the program then attempts 10 mHz and finally attempts 8 mHz. After the last threshold is tried and still no satisfactory result is obtained, the program then deems the adaptive threshold method does not work in the case.

By attempting three threshold values, the triangulation application greatly reduces the risk of having too much riding on a single threshold by making each threshold much less mission-critical, which is very conducive to solving many non-convergent cases resulted from noise or transients of the frequency itself and the measurement thereof. The efficacy of using adaptive threshold method has been substantiated by many successful FNNT online triangulation cases.

B, Oscillation Method in Generation Trip Events

All the means mentioned above contribute to a satisfactory estimation of event location. In most cases reported in FNET so far a good estimation result can be achieved by the collective functioning of those individual albeit interrelated approaches. However, there still exist a number of cases where the result is still far off after all those means are exhausted.

A great proportion of those non-convergent cases, viewed from the point of view of oscillations, contain obvious oscillatory element in their frequency signature. Figure 5.15 provides a typical oscillation example of those cases.

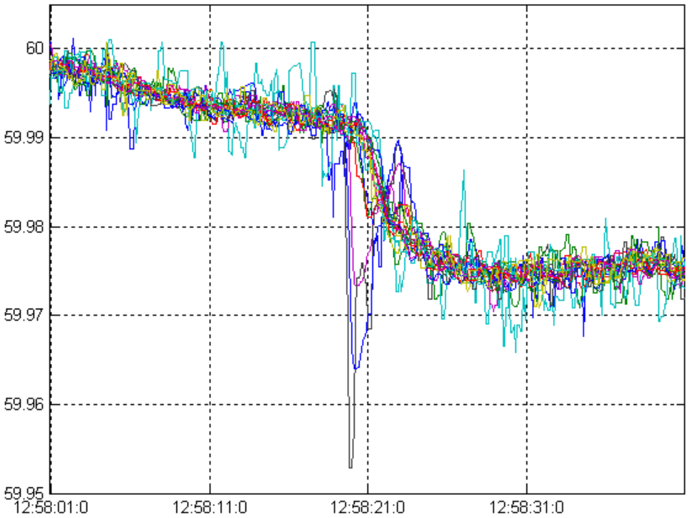


Fig. 5.15 01/21/2008 Generation Trip Event with Large Oscillations

When there is a generation trip, the total generated active power decreases whereas the load stays almost the same as before the event. The unbalance between generation and load then leads to re-distribution of active power among the remaining generators. Because of the complexity of the power system and relatively slow response of the turbine, the rotor of generators will convert its kinetic energy into power to make the mechanical and electrical power equal. This behavior is one reason for power system oscillations, and the magnitude of oscillation can reflect the severity of the unbalance and the electrical distance from the generation trip hypocenter. So the magnitude of oscillation can contribute to determining the generator trip location.

The frequency oscillation and deviation are concomitant and they superimpose upon each other in oscillatory generation trip events, which brings about the issue of how to separate the oscillatory component from the general trend. The process of separating oscillation from the frequency signature is called detrending. One classical way to detrend is to first calculate the 51 moving-median of the frequencies and then subtract the 51 moving-median result from the original values. The difference then will be the oscillation part of the frequency curve of each FDR. To rule out the false positives caused by the noise, a threshold of 10 mHz is set to screen out the FDRs whose oscillation magnitude is too small.

The actual event location of the non-convergent case in Figure 5.14 is the unit 1 of the Allen S King power plant which is situated in Washington, MN. After the detrending procedure, the FDRs whose oscillation magnitude is more than 10 mHz during the event time are presented in Table 5.3. As the event location is known for this case, the distances from those FDRs to the event epicenter are also computed and listed in the table.

FDR Number	40	35	11	41	4
Oscillaation Magnitude (mHz)	33.3	11.3	12.0	20.7	14.0
Distance from the Hypocenter (Mile)	16.49	211.1	381.3	387.3	836.2

Tab. 5.3 FDRs' Oscillation Magnitude and Distance from Event Location

As the event location is known for this case, the relationship between the oscillation magnitude and the distance from the specific FDR to the event location can be investigated. There is a meaningful observation of the relationship between the oscillation magnitude and the distance for this case as well as for many other available cases that the FDR which has the greatest oscillation magnitude is always closest to the event location. This observation is exemplified in Table 5.3 that FDR 40 has the greatest oscillation magnitude, far leading the rest of the oscillation magnitudes of the other FDRs, and it also is the closest one to the actual event location.

In light of the fact that it is a crucial factor that affects the oscillation magnitude, the generation trip amount must be taken into account when taking advantage of the oscillation magnitude in the process of locating the event epicenter. The way to reflect trip amount in the event location process is to use relative oscillation magnitude, which is the quotient of the oscillation magnitude in mHz of a specific FDR divided by the trip amount in Megawatts, instead of using the absolute oscillation magnitude. For example, in Figure 5.15, the total generation trip is 420 MW and the detected absolute oscillation magnitude is 33.3 mHz. The relative oscillation magnitude that is used in the event location program is 33.3 mHz / 420 MW = 0.079 mHz/MW.

The practice of utilizing oscillation to resolve the non-convergent cases is the final resort after all the previous methods cannot give a reasonable estimation. In the FNET event location estimation research effort, the procedures are: first, calculate the oscillation magnitude of each FDR and then obtain the relative oscillation magnitude; second, list the first three FDRs in TDOA and the FDR that has the greatest relative oscillation magnitude; third, if the FDR that has the greatest relative oscillation magnitude is among the first three FDRs in TDOA, search the area in 200 miles radius from the FDR for the nearest generating station that has a generator whose capacity exceeds the estimated trip amount; lastly, designate that generating station as the event location estimation result if a generating station that meets the capacity requirement can be found.

5.3.5 Summary and Performance Evaluation of the Location of Disturbance Algorithm

The methods expounded previously in this section collectively serve the purpose of estimating the event location to researchers' satisfaction when a disturbance in a power system is detected. The primary method is the generating station hypothesis method in subsection 5.3.2, which in most researched cases runs well and provides with decent estimations. The rest of the methods are devised to complement the primary method when the case is non-convergent.

Figure 5.16 explains how all those methods coordinate in the FNET event location estimation research effort. First, the algorithm determines whether it is a generation trip or a load shedding. If it is a load shedding, the server automatically branches to the Newton's method or the Least Squares method module to estimate the event location. If it is a generation trip, the generating station hypothesis approach is tried first to see if it can generate a satisfactory result. If not, the algorithm then goes to the Newton's method or the Least Squares method module. If it is not convergent or there is still too much error in the least squares residue, the server program then resort to the oscillation method. After all the methods are executed and no acceptable estimation is reached, the algorithm reports a failed triangulation instance.

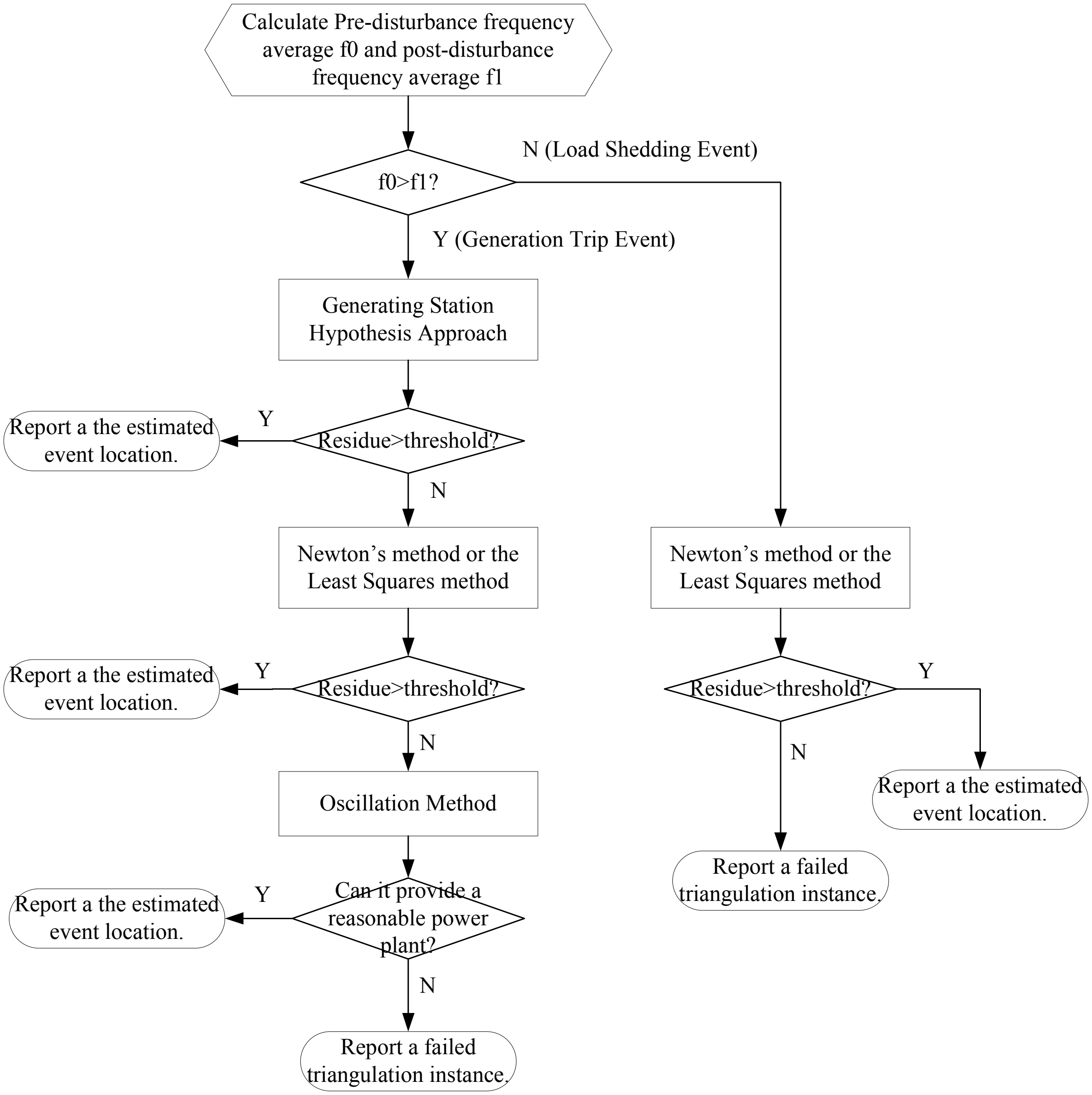

Calculate Pre-disturbance frequency average f0 and post-disturbance frequency average f1
f0>f1?
N (Load Shedding Event)
Y (Generation Trip Event)
Generating Station Hypothesis Approach
Residue>threshold?
Y
Report a the estimated event location.
N
Newton’s method or the Least Squares method
Residue>threshold?
Y
Report a the estimated event location.
N
Oscillation Method
Can it provide a reasonable power plant?
Y
Report a the estimated event location.
N
Report a failed triangulation instance.
Newton’s method or the Least Squares method
Residue>threshold?
Y
Report a the estimated event location.
N
Report a failed triangulation instance.

Fig. 5.16 Flowchart of the Triangulation Research Program

The estimation based on the frequency data and the geographic information of the FDRs is a novel idea to utilize the wide-area measurement capability of FNET. So far, the offline test program has yielded decent results as to the events with confirmed location provided by the utilities.

There are 96 confirmed cases in the time period from 08/06/2007 to 10/26/2008. Statistics show the success and accuracy of the FNET event location estimation algorithm in Table 5.4. The offline estimation of nearly half of the cases taken stock of during the specified period (43.75%) is within 100 miles radius from the actual location, which is categorized as satisfactory in the table. There are 8.33% decent estimations that are in the ring area between 100 miles and 200 miles radius from the actual location. Only approximately one fifth (22.92%) of the cases are more than 200 miles from the actual location of the event.

Category	Satisfactory (<100 miles)	Acceptable (<200 miles)	Too Off (>200 miles)	Non-convergent
Number	42	8	24	22
Percentage	43.75%	8.33%	16.67%	22.92%

Tab. 5.4 Statistics of the FNET Event Location Estimation

Chapter 6: FNET Angle-based Power System Oscillation Detection

Alternating current generators remain in synchronism because of the self-regulating properties of their interconnection. If one machine deviates from its synchronous speed, power is transferred from the other generators in the system in such a way as to reduce the speed deviation. The moments of inertia of the generators also come into play, and result in the speed overcorrecting in an analogous manner to a pendulum swinging about its equilibrium; the pendulum inertia is equivalent to the generator inertia, and the torque on the pendulum due to gravity is equivalent to the synchronizing torque between the generators in the power system. However, generators are much more complicated dynamic devices than are pendulums, and one must not be tempted to put too much emphasis on this analogy. However, it is true to say that power system oscillations are as natural as those of pendulums.

An interconnected power system cannot operate without control. This is affected by a combination of manual operator controls and automatic controls. The operators control the power that the generator supplies under normal operating conditions and the automatic controls come into play to make the fast adjustments necessary to maintain the system voltage and frequency within design limits following sudden changes in the system. Thus, most generators have speed governing system which automatically adjust the prime mover driving the generator so as to keep the generator speed constant, and voltage regulating systems which adjust the generators' excitation to maintain the generator voltages constant. These controls are necessary for any interconnected power system to supply power of the quality demanded by today's electric power users. However, most automatic controls use high gain negative feedback, which, by its active nature, can cause oscillations to grow in amplitude with time. The automatic controls in power systems must, as with other automatic feedback controls, be designed so that oscillations decay rather than grow.

Oscillations were observed in power systems as soon as synchronous generators were interconnected to provide more power capacity and more reliability. Originally, the interconnected generators were fairly close to one another, and oscillations were at frequencies of the order of 0.1 to 2 Hz. Amortiseur (damper) windings on the generator rotor were used to prevent the oscillations amplitudes increasing. Damper windings act like the squirrel cage winding of an induction motor and produce a torque proportional to the speed deviation of the rotor from synchronous speed. They absorb the energy associated with the system oscillations and so cause their amplitudes to reduce.

As power system reliability became increasingly important, the requirement for a system to be able to recover from a fault cleared by relay action was added to the system design specifications. Rapid automatic voltage control was used to prevent the system's generators loosing synchronism following a system fault. Fast excitation systems, however, tend to reduce the damping of system oscillations. Special stabilizing controls (Power System Stabilizers) were designed to damp these oscillations.

In the 1950s and 1960s, electric power utilities found that they could achieve more reliability and economy by interconnecting to other utilities, often through quite long transmission lines. In some cases, when the utilities connected, low frequency growing oscillations prevented the interconnection from being retained. In some instances, lowering automatic voltage regulator gains was all that was necessary to make the system interconnection successful. However, in other cases the interconnection plans were abandoned until asynchronous HVDC interconnections were technically possible. AC tie lines became more stressed, and low frequency oscillations between some interconnected systems were found to increase in magnitude. In the worst cases, these oscillations caused the interconnection to be lost with consequent inability to supply customer load.

From an operating point of view, oscillations are acceptable as long as they decay. However, oscillations are a characteristic of the system; they are initiated by the normal small changes in the system load. An increase in tie line flow of as little as 10 MW may make the difference between decaying oscillations which are acceptable and increasing oscillations which have the potential to cause system collapse. If the there is no warning to the operator if a new operating condition causes an oscillation to increase in magnitude, the operator's manual control might trigger a migration from a decaying oscillation to a increasing oscillation and eventually cause the system to collapse. However, the oscillations are not noticeable by the system's operators unless they have special instrumentation that detects them. This then calls for a wide-area monitoring system on system oscillations, which can inform the system operators of the presence of oscillations so that correction measures can be taken to mitigate the oscillations or at least to prevent them from increasing.

With its wider-area monitoring ability all around the North America power grid, FNET can be utilized to provide an oscillation detection system for the power system operators, policy makers, customers as well as researchers. A module that detects the oscillation of the Eastern Interconnection has been thoroughly researched in FNET, which can save the data to the computer, reports the detection to FNET users and executes certain analyses in terms of generating the parameters of the detected oscillation. It provides valuable data for studies of inter-area oscillations, scenario reconstruction, and model verification, which helps to increase power system engineers' awareness of oscillations sufficiently, to encourage them to treat oscillations seriously and to set up accurate system models in the interest of better grid management.

This chapter starts with a number of observations of FNET on the power system oscillation, demonstrating the feasibility and necessity of the oscillation detection in FNET. Particulars about the implementation of the oscillation detection module are then explained at full length, from data conditioning, real-time processing mechanism, detection algorithm, exception handling to threshold setting. Lastly, a selection of detected oscillations by the module is presented in the purpose of providing concrete instances of how the detection module works.

6.1 FNET Observations on Power System Oscillations

The occurrence of the oscillations in the power system is not a rare phenomenon. As a matter of fact, there are oscillations almost every moment in a power system in the sense that there are disturbances, larger or smaller, constantly present in a power system and the disturbances theoretically can lead to an oscillation to a certain degree. The disturbances cause imbalance between generation and load, which leads to redistribution of real power among the generators. In some cases, the redistribution of real power bring about some real power being sent back and forth from one region to another, and this reciprocal real power exchange in different regions manifest itself in oscillations in many forms, such as voltage oscillation, phase angle oscillation and frequency oscillation. From the perspective of wide-area measurements, some of the oscillations are very small in magnitude and are overwhelmed by the noise. Some are noticeable albeit indistinct of any oscillatory trajectory and some are viewed as pure fluctuations of the system. On the other hand, some of the oscillations exhibit a very clear sinusoidal waveform in the output of those electrical variables.

FNET has demonstrated a preeminent capability of monitoring the electrical and mechanical behaviors of North America power grids since its debut into power system wide-area measurement field in 2004. It has collected and stored on the servers 5 years worth of frequency, phase angle and voltage magnitude data of the three interconnections. Also, the event trigger module has detected thousands of cases of system wide disturbances that are saved in the event database on the servers. Delving into those databases enables us to obtain the frequency plots that obviously contain oscillations, which points to the fact that FNET possesses an outstanding observational ability over the power system oscillations. Figure 6.1 is an example of a generation trip accompanied by oscillation in comparison with a generation trip event without oscillation with respect to their frequency signatures.

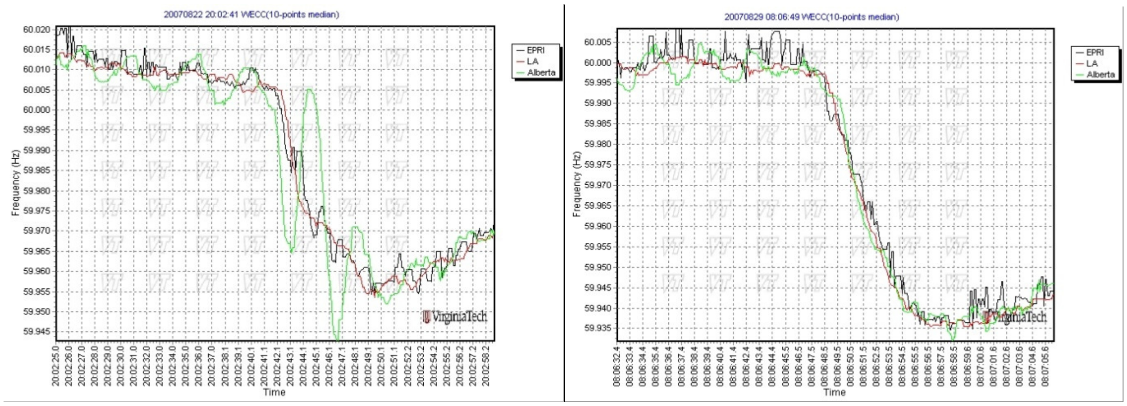


Fig. 6.1 Oscillation in Comparison with non-Oscillation

The two events both occurred in WECC with the same three FDRs reporting data, namely EPRI, LA and Alberta, and the two event dates are a week apart. The frequency of the Alberta FDR oscillated against the frequency of the EPRI and LA FDRs at a peak-to-peak value oscillation magnitude of 30 mHz in the left figure, whereas it followed very well with the frequency of the two FDRs in the right figure. The frequencies of the EPRI FDR and the LA FDR stuck with each other fairly closely because of their geographic proximity.

The comparison of Figure 6.1 plotted by using the FNET frequency data displays the evident difference between the oscillation of frequency and normal frequency drop, which surely justifies the feasibility of taking advantage of FNET data into the applications about detecting power system oscillation all around the FNET network.

A great deal of oscillations can be found in the FNET measurement database and event database. However, since there was not any oscillation detection, the oscillation could only be detected as a generation trip or load shedding event when the event occurs simultaneously with the oscillation. A problem would thus be created that some oscillations can be wrongly detected as a generation trip or load shedding, or the oscillation characteristic of some events can be inappropriately neglected due to the reason that there is not a means for the server to detect the oscillation and then separate them for the normal events. There are actually some straightforward instances in FNET system that can illustrate the problem of not having a tool to detect and report oscillations.

The first is a large-amount generation trip event that occurred in Eastern Interconnection on Aug. 4th, 2007. It was the Jefferson-Greentown 765kV line trip followed by several generator trips. The tripped generating units: 2 Rockport units (2600MW capacity), 1 Petersburg unit (510MW capacity), and 2 Newton units (1120MW capacity). The total loss of generation was about 4000MW. The FNET server triggered the event and the estimated trip amount is 2483 MW. The event location program pinpointed the event location. The map of this event that interprets the locations of the event is shown in Figure 6.2.

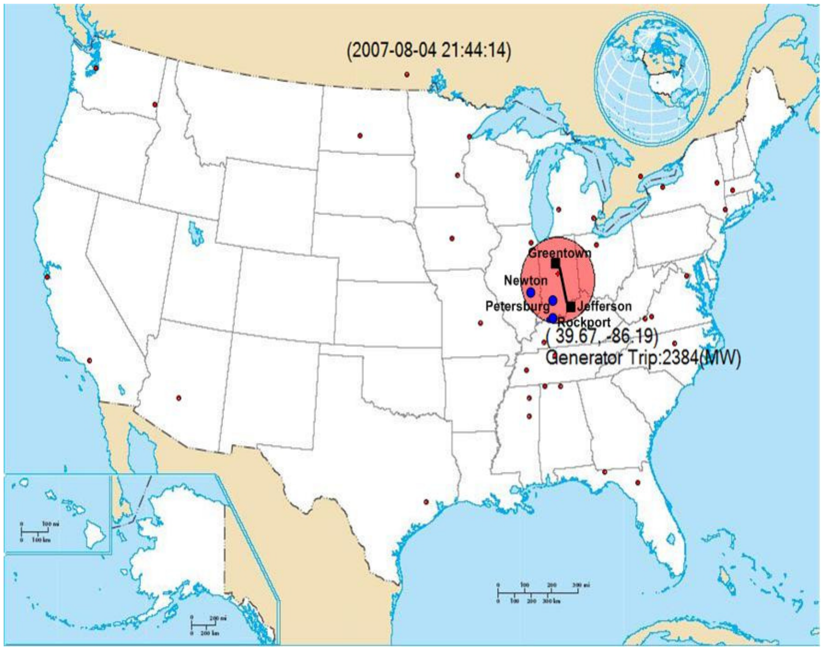


Fig. 6.2 The 08/04/2007 Event

This event was detected by the event trigger module explained in Chapter 5, labeled merely as a generation trip. However, there were significant oscillatory elements contained in the frequency plots displayed in the left image in Figure 6.3. The frequency can be decomposed into the frequency trend and the oscillatory part by detrending. The right image in Figure 6.3 shows the oscillations with the trend of frequency drop taken out after detrending.

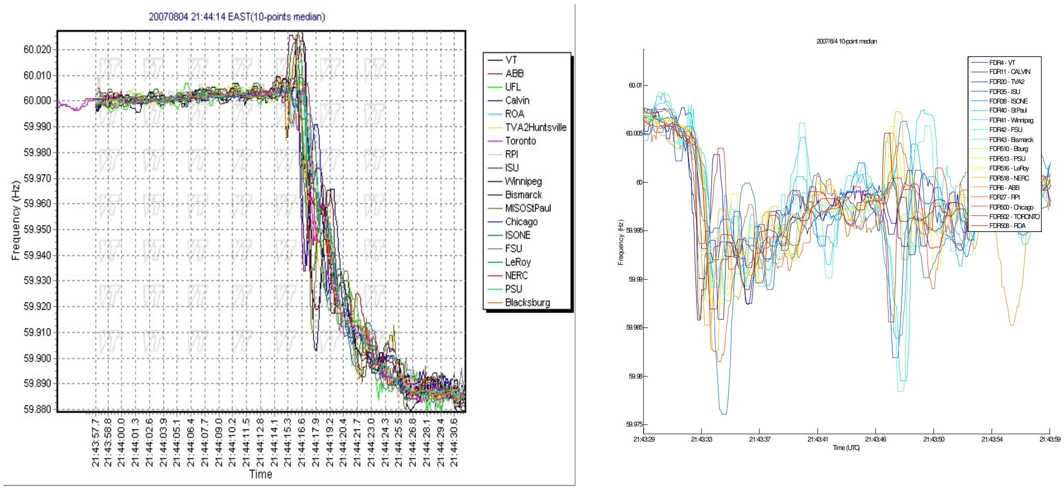


Fig. 6.3 Frequency Plots and Frequency Oscillations of the 08/04/2007 Event

The oscillations in the figure are so pronounced that the analysis of the event would be rather incomplete if the oscillations were left out. It is therefore imperative for the oscillations to be detected by a separate module and the associated data to be saved after the detection for a more comprehensive analysis of a complicated event like this.

There is another significant grid reliability event in South Florida that occurred on Feb. 26th, 2008. An equipment malfunction in a substation near Miami disabled two power distribution lines between Miami and Daytona Beach, and in response, Florida Power & Light's Turkey Point nuclear plant south of Miami stopped operating. The trouble then set off a sequence of events that within two to three minutes had knocked numerous generating facilities (at least 8 power plants) off-line, including the Turkey Point nuclear power plant south of Miami. The outages cut power to 2 to 3 million people during the heat of a day.

This major event caused enormous oscillations in the Eastern Interconnection that were observed quite clearly by many utilities companies in the EI grid. FNET detected this event as a load shedding case and sent to its users the frequency plots of the event which is shown in the left image of Figure 6.4. In the frequency plots, large oscillations can be positively discerned and the minimum frequency deviation even hit the record low, 59.84 Hz, in the EI system. The right image of Figure 6.4 provides the plots of relative angle of a few representative FDRs of the event, which also exhibit tremendous oscillatory curves with the maximum angle peak-to-peak value exceeding 90 degrees. The Chicago unit is chosen as the reference in the angle plots.

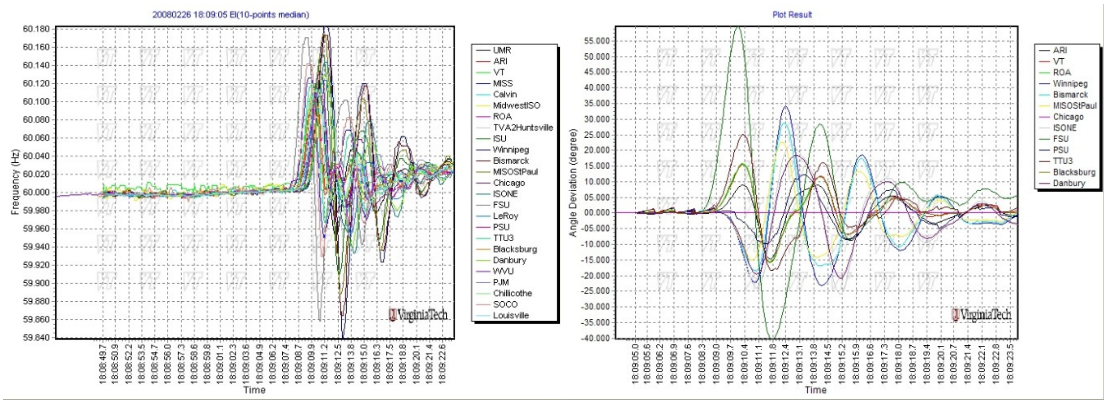


Fig. 6.4 Plots of the Florida Event

The event is identified as a load shedding by FNET event trigger module. However, the frequency plots show there is no significant frequency change before and after the oscillation. The reason it was labeled as a load shedding is due to the fact that the first swings of the oscillations were upward, bearing some similarity to the frequency behavior of a load shedding. The attributes of the event and the frequency and angle signatures prove it is improper for the event to be detected as a loading shedding case.

The two events enumerated in this section strongly verify the need for an oscillation detection module to detect oscillations and to filter out the oscillation elements from normal frequency disturbances. FNET situational awareness functions can be greatly enhanced if the oscillations in the monitored areas are correctly and precisely spotted, and the frequencies and angles collected during the oscillation detection time frame can be the data source for FNET oscillation analysis applications that will be developed later on. The next sections of this chapter will switch to the details of the implementation of the oscillation detection module. Real cases generated from the online testing of the module will be given as concrete evaluations of its performance.

6.2 The Implementation of the FNET Oscillation Detection

Oscillations manifest themselves very clearly in the frequency plots and angle plots provided by FNET database when confirmed by other applications or utilities. The obviousness of the oscillatory elements in the oscillation events observed in FNET demonstrates the feasibility of detecting real-time oscillations by utilizing online FNET data. There are many approaches to attack this problem, among which are some common methods, such as frequency-based method, angle-based method and voltage-based method. Those electrical parameters in a power system are direct indicators of an oscillation. As for the application of FNET data to the oscillation detection, the data processing and the specific algorithm need to be customized according to FNET data structure and the attributes of different data.

The voltage of a bus in a substation or a power plant can be a direct index of the healthiness of the operation of the power system it belongs to, and therefore can be used to detect oscillations in the power system if it is installed at generating system or transmission level. However, since FDRs are installed at the distribution level, the voltages measured by them are more or less polluted by certain electrical phenomena endemic to the distribution level such as: phase imbalance, greater fluctuation and higher percentage noise. Consequently, it is not suitable for the FDR voltage to be employed as the detection variable for power system oscillations.

Frequency data and angle data of FDRs all exhibit a better representation of the corresponding electrical parameters of the nearby bus and the status of the power system in general. The comparison between the dynamic performance of the FDR and that of the PMU in laboratory evaluation experiments also attests to the sufficiency of the measurement accuracy of the FDRs when it is applied to the oscillation detection development. We can also notice a smooth sinusoidal curve of the frequency or angle plot of an FDR when an oscillation is confirmed to be present, confirming the utility of FNET frequency or angle measurements for the purpose of detecting oscillations. The comparison between the two images in Figure 6.4 shows an appreciably better accuracy of the phase angle measurement over that of the frequency measurement, which substantiates the superiority of the phase angle data when it comes to oscillations.

This chapter will embark on the particulars of the development of the oscillation detection program by the analysis of choosing angle as the sole parameter for the detection. Data conditioning, angle unwrapping and real-time handling mechanism will be explained in detail so as to clarify the procedures of preparing the data source of the detection module. Lastly, the algorithm that uses the processed angle data to continuously determine the occurrence of an oscillation is discussed. This chapter concludes with a flowchart of the detection module.

6.2.1 Analysis of the Superiority of Angle-based Oscillation Detection

Since frequency data and angle data can both be utilized to accomplish the detection of the oscillations in power systems, it is foremost important to decide which measurements are more qualified for the oscillation detection application. We can get some hint from the figures in the previous section that angles appear to be more accurate in the point view of visualizing the oscillation with the historical data. There are also indications from recent detected oscillation cases by the new angle-based oscillation detection module that the oscillatory trajectory in angle plots is much clearer than that in frequency plots.

Figure 6.5 presents an oscillation event that can evince the better quality of the angles by the comparison of the two plottings. The disturbance occurred at 09:13:20 on 10/06/09. Relatively clear oscillations are observable from the angle plots in the left image of Figure 6.5, and no appreciable noise is present to affect the visibility of the oscillations. Some oscillation parameters like oscillation frequency, oscillation magnitude and damping ratio can even be roughly estimated purely from the angle plots. On the other hand, although there are some sine-shape curves of frequency seemingly indicative of oscillation expressions, the frequency plots on the whole are chaotic and do not convey the oscillation information realistically. The disparate oscillation resolutions of the frequency data and the angle data dictate the usage of angles in FNET oscillation detection module.

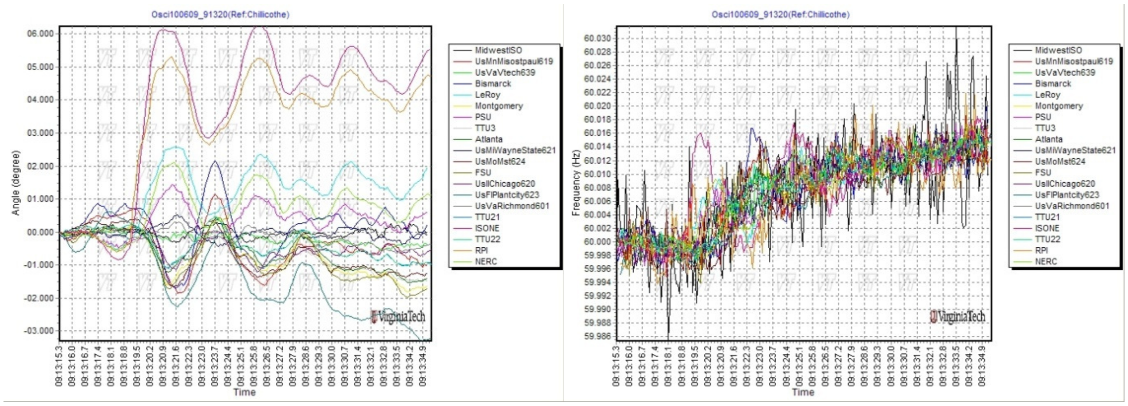


Fig. 6.5 Plots of the 10/06/2009 Florida Event

The discrepancy of clarity of oscillation in the frequency signature and in the angle signature results from their different signal noise ratios (SNRs). As can be clearly observed, the trace of oscillation in the frequency plots is overwhelmed by the noise and it is difficult to differentiate oscillations from noise. The reason why the SNR of angle measurements are much better than that of frequency measurements is that the long period of a low-frequency oscillation of a small magnitude in frequency can cause the angle to accumulate and reach such a level that the oscillation magnitude in angle, which is the integral of the frequency deviation, can squelch the effect of the noise. The frequency of the inter-area oscillation is usually as low as 0.1 Hz and the period correspondingly can be as high as 10 seconds. The angle change can reach a sizable amount over the long period of the oscillation even if the oscillation magnitude in frequency is fairly small.

Assume an oscillation with the amplitude in frequency being A_{freq} and the oscillation period being T_{osci} as shown in Figure 6.6.

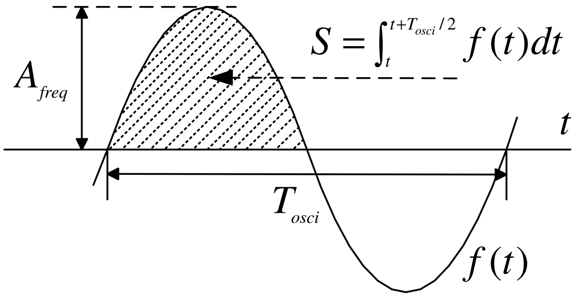


Fig. 6.6 Diagram of the Oscillation Magnitude in Angle

Since the frequency is the derivative of the angle, the relative angle change over a time period is proportional to the integral of the frequency deviation from the reference. The integral of the frequency deviation over a time period is the number of revolutions over the time period. Hence, the maximum angle change, which is also the peak-to-peak oscillation magnitude in angle, is the product of the coefficient of 360 degrees per revolution and the shade area in Figure 6.6. The result of the integration operation in Figure 6.6, the area of the shade, is $\frac{A_{freq}}{\pi}T_{osci}$. Therefore, we can get the correlation between the oscillation magnitude in angle A_{ang} and the oscillation magnitude in frequency A_{freq} in (6-1):

$$A_{ang} = 360\frac{A_{freq}}{\pi}T_{osci} \qquad (6\text{-}1)$$

If the oscillation magnitude in frequency is 10 mHz, and the oscillation frequency is 0.1Hz, the relative angle change will be 11.46 degrees according to 6-1, which is a marked change in phase angle in power system operations.

The frequency accuracy of an FDR is around 2 mHz, and the angle accuracy is approximately 0.1 degree. Most PMUs have the similar frequency and angle accuracies in their specifications. That being said, the SNR of frequency measurement, which can be calculated as the ratio between the oscillation magnitude in frequency and the frequency resolution of an FDR, is 10 mHz/ 2 mHz = 5 for the parameters given above. However, the SNR of the angle measurement if calculated with the same parameters is 11.46 degrees/ 0.1 degrees = 114.6, which is far greater than that of the SNR of the frequency measurement. The immense disparity between the SNRs of the frequency and the angle cause the oscillation to be clear-cut in angle measurements whereas obscure in frequency measurements. This justifies the choice of the angle data to detect oscillations.

6.2.2 Treatment of the Sawtooth Error

The data accuracy and integrity is the foundation of the success of any power system application. As discussed in Chapter 4, some generation I FDRs use old firmware and therefore have sawtooth errors on angle measurements that can weaken the efficiency and correctness of the oscillation detection and make the angle signature chaotic.

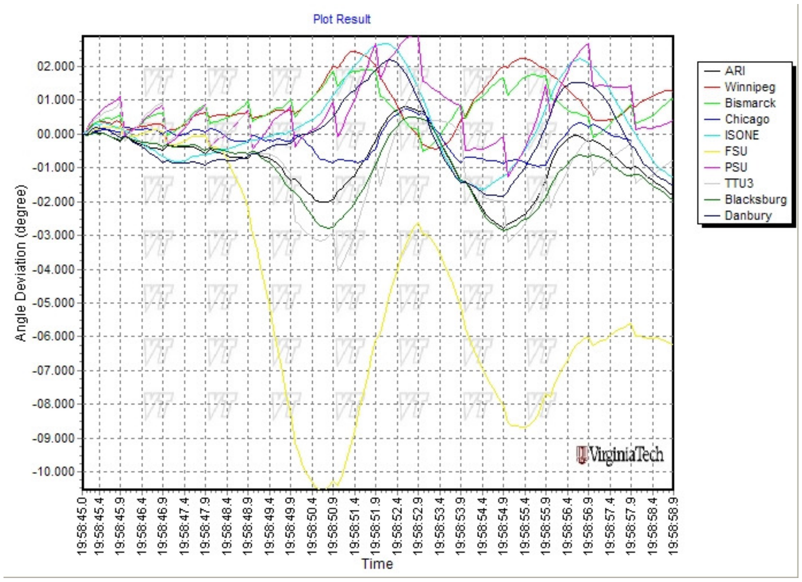


Fig. 6.7 Effect of Sawteeth on the Oscillation Angle Curve

The angle plots in Figure 6.7 are the signatures of the angle data of an oscillation with some FDRs having sawtooth errors in angle measurements. The new FDRs without sawtooth error register clear decaying sinusoidal curves, whereas the old FDRs with sawtooth error display curves cluttered by the sawteeth with abrupt turns. Considering the size of the sawtooth is almost 1.5 degrees (PSU, Bismark units) and the magnitude of the oscillation is around 2.5 degrees, it is highly possible that the angle curves can be considerably deformed by the sawtooth errors. In order to correctly conduct the oscillation detection, the sawtooth errors must be rectified.

Because the sawtooth errors are linear with the data point number, which is the number of the tenth of a second, we can compensate for the errors by subtracting the erroneous amount corresponding to the linearity of the errors. Note that a constant frequency deviation also causes the angle measurements to change linearly. Therefore, the frequency deviation effect on the linearity of the errors must be taken into account in order for the problem to be properly resolved.

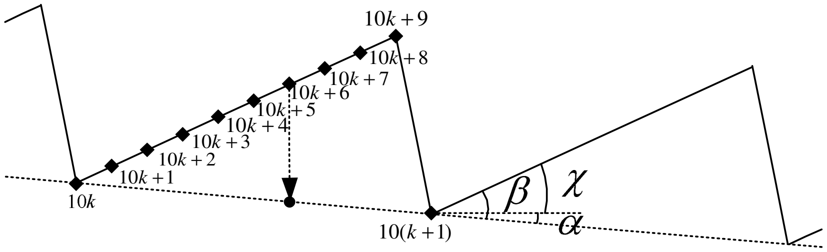


Fig. 6.8 Solution of the Sawtooth Error

The illustration of the sawtooth error in the presence of a constant frequency deviation is drawn in Figure 6.8. The solid curve is the angle measurement, and the broken straight line is the right angle value. The goal is to bring the measurement from solid line to the broken straight line, for example, to move the diamond 10k+6 to the dot on the broken line as directed by the arrow. More specifically, the slope between the solid straight line and the broken straight line β in Figure 6.8, is the coefficient that the compensation procedure needs to amend the measurements with. However, slope β cannot be directly obtained. The two quantities that can be directly obtained are the slope of the solid straight line and the slope of the broken straight line, χ and α, in Figure 6.8 respectively. The coefficient β can be obtained by the adding χ to α.

From the knowledge of Chapter 4, β is the inherent attribute of a FDR and is consistent throughout its entire lifespan. Therefore, β can be computed upon the startup of the server program, and be kept as a quasi-constant in the program. Keep in mind that it is assumed the frequency deviation in Figure 6.8 is constant and therefore the computation should be performed when the system is in a steady state. The steady state condition can be confirmed by the fact that the angle difference between two consecutive seconds, for instance point 10k+1 and point 10(k+1)+1 in Figure 6.8, is less than a small threshold.

As a summary, the procedures of rectifying the sawtooth errors are:

(1) Calculate $|Angle[10(k+1)+1]-Angle[10k+1]|$ and wait until the value is less than 0.5 degree. Half a degree is the threshold to determine the system is running at a steady state during measurement time period.

(2) Calculate $\frac{Angle[10k]-Angle[10(k+1)]}{10}$ and write down the quotient as α.

(3) Calculate $\frac{Angle[10k]-Angle[10(k+1)]}{10}$ and write down the quotient as χ.

(4) Adding α to χ. Assign the sum to β and save β as the compensation coefficient.

(5) After the coefficient β is obtained, for the k^{th} sample in a second, subtract $k\beta$ from the original value to obtain the compensated angle.

6.2.3 Angle Unwrapping, Reference Angle Issues, and Real-time Condition Handling

A) Angle unwrapping

The value range of the angle measurements are from 0 to 2π, which causes sudden angle jumps when the measurement crosses the boundary of the value range. It is necessary for the angles to be made monotonic through the detection time frame. However, the detection module runs continuously once it is turned on, and therefore the angle value might increase in one direction and eventually exceed the definition limit of the variable if the unwrapping routine runs without resetting. The solution then is to reset the unwrapping parameter periodically (every minute in the routine) so that the angle value is always within reasonable bounds.

Figure 6.9 shows the flowchart of the angle unwrapping routine. The algorithm is to keep track of the angle jumps, and modify the unwrapping parameter m_fUnwrap according to the jumps. If it is a negative jump, which means the angle changes from near 2π to just over 0, 2π needs to be added to fUnwrap to make the angle monotonic. If it is a positive jump, which means the angle changes from just over 0 to near 2π, 2π needs to be subtracted from fUnwrap to make the angle monotonic. If there is no angle jump, no modification needs to be made to fUnwrap. The sum of the unwrapping parameter fUnwrap and the original angle value fAng is then the unwrapped angle data.

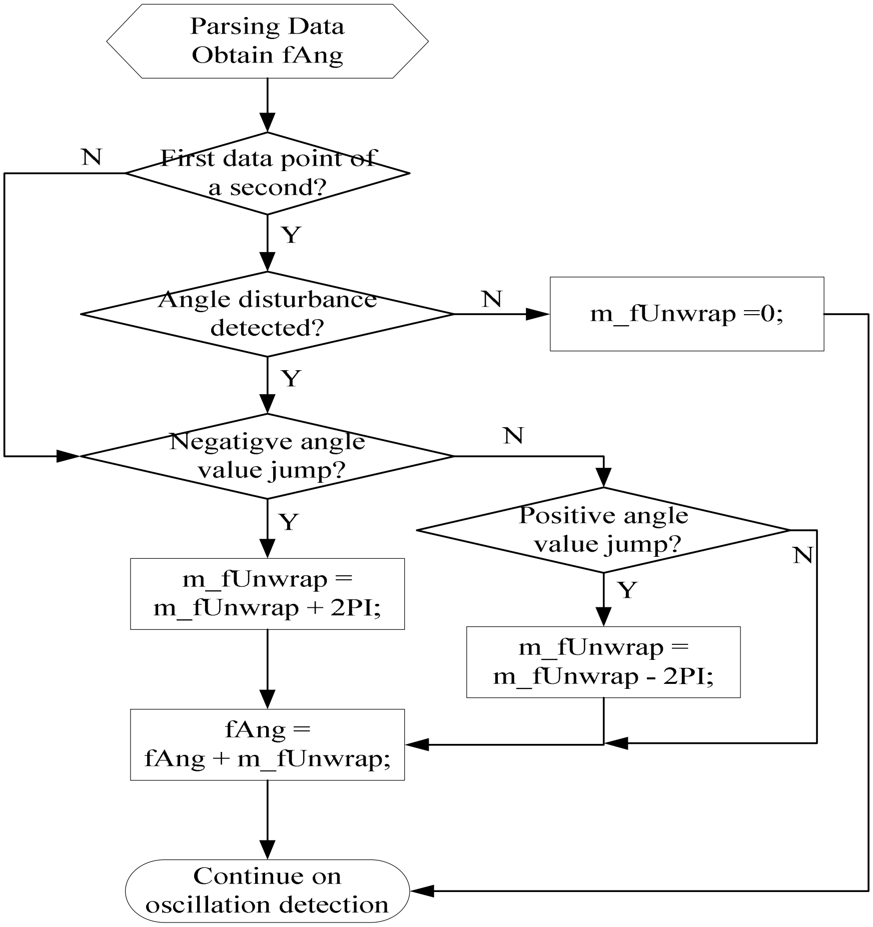

Parsing Data
Obtain fAng
First data point of a second?
N
Y
Angle disturbance detected?
N
m_fUnwrap =0;
Y
Negatigve angle value jump?
N
Y
Positive angle value jump?
N
Y
m_fUnwrap = m_fUnwrap + 2PI;
m_fUnwrap = m_fUnwrap - 2PI;
fAng = fAng + m_fUnwrap;
Continue on oscillation detection

Fig. 6.9 Angle Unwrapping Flowchart

One complication of the unwrapping procedure is resetting m_fUnwrap to prevent it from overflow as discussed previously. In Figure 6.9, m_fUnwrap is reset at the first data point of every minute. After parsing the data that contain a whole data set of a sample point, including time stamp, frequency, phase angle, and voltage magnitude, the routine checks if the current data point is the first one in a minute and then branches to two routes. If the data point is not the first one of a minute or there is an angle disturbance detected underway when the data point is the first one, the routine braches to the unwrapping part at the bottom of Figure 6.9. If the data point is the first one of a minute and there is no angle disturbance underway, the angle unwrapping routine resets the unwrapping parameter. This is to avoid interrupting the data processing for an ongoing oscillation and ensure the data continuity of the angle data.

B) Reference angle

The absolute phase angle is a dynamic angle that rotates at 3600 RPM in a 60 Hz nominal frequency power grid. The change of absolute phase angle in reaction to an oscillation would be completely inundated by the fast fundamental wave rotation of the absolute phase angle. Moreover, the state of a power system is decided by the relative power angles of the buses instead of the absolute power angles because the power system operates at a synchronized frequency. Hence, only the relative angle of a FDR with respect to the reference FDR is of interest to us. The oscillation detection application has specific means to deal with the particulars of the relative angle calculation and the angle data of the reference FDR itself.

First and foremost, choosing the reference FDR needs to consider the geographic distribution of the FDRs in an electric power interconnection. The reference FDR is best to be placed in a location that can evenly divide the changes of relative angle of all the FDRs during an event. For the Eastern Interconnection, the optimal locations are in from TVA to southern Ohio area. Conventionally, many power system applications use Browns Ferry power plant in TVA as the reference bus for the Eastern Interconnection.

A flag variable m_bOscilRef is used to indicate the FDR that the current socket is associated with is the angle reference. When the FDR is identified as the reference, the new angle value will be filled into the reference angle array g_fOscilBuffer in accordance with the sequence number (variable ‘ind’) of the data point in the current minute, and time keeping variable g_uCurrentRefTime that corresponds to the unsigned integer counting the data points from the beginning of a day will be renewed. If the FDR is not the reference, the new relative angle will be calculated by subtracting the reference angle retrieved from the reference angle array g_fOscilBuffer with the sequence number of the time stamp modulo 600. The new relative angle value will be saved to the oscillation data buffer m_fOscilBuffer. Variable g_uCurrentRefTime is used to align the socket data packet of a non-reference FDR with the right reference angle.

C) Real-time condition handling

Owing to the complexities brought by the data transmission via the Internet, the sequence of the data arrival times of the same FDR can be disarranged and the sequence between any 2 FDRs are irregular and constantly changing. Besides, there are missing data almost for all the FDRs, and the length of missing data varies. Therefore, the module needs to handle those real-time related conditions in order to prevent them from affecting the logic of the algorithm of the detection module.

The main task the real-time handling part deals with is to coordinate the paces of the different data arrival times with respect to the reference of the FDRs. Because of various data arrival time delays, a data buffer is needed to cushion the inequalities of the current data times of FDRs. A first-in-first-out 10 element buffer angleFIFO is used to save up to 10 old data points that can tackle 1 second data transmission delay. The element of the buffer is a structure that contains time stamp, current time index, relative angle, and frequency.

The module keeps comparing the most current time g_uCurrentRefTime of the reference with the time stamp of the oldest element in the FIFO and process the data of the oldest element while its time stamp is older than the current time of the reference. If they come in before the data of the reference, the more recent data of an FDR will be saved to the FIFO until the FIFO depth index m_nFIFOPnt exceeds 10, the limit that indicates the reference is probably out of synchronism with the other FDRs. In that case, the data alarm flag g_bRefDataAlarm is set and the module suspends the detection even it is underway until the reference resume synchronized. If there are backup references and one of them is checked to be in synchronism, the backup one will be designated as the incumbent reference FDR and the detection will be resumed at the beginning of the next minute.

Because the module is a real-time application, if the oldest element is too lagging behind, which is preset to be 6 sample points, the module also will set the data alarm flag m_uDataStatus of that specific FDR and block its data to get involved in the computation until the it succeeds in clearing the flag at the turn of every minute.

6.2.4 Oscillation Detection Algorithm

The relative angle of an FDR remains stable without much deviation when the system is in normal operation and it changes abruptly during events because of the power angle redistribution associated with the event. Oscillations may be brought about by the characteristics of the disturbance and the operation mode of the system during the dynamic time period of the event. The angles will form a wavy curve due to the oscillation. The basic idea of the algorithm is to continuously monitor the angles and to identify oscillations by the appearance of those two phenomena. Hence, there are two steps in the process of detecting an event.

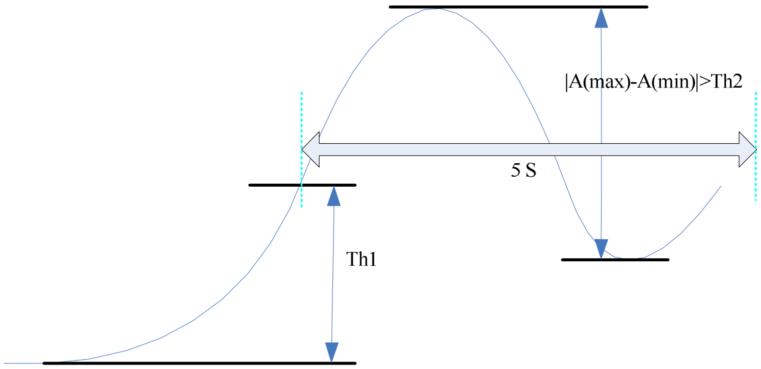


Fig. 6.10 Diagram of the Angle-based Oscillation Detection

A schematic to illustrate the algorithm is shown in Figure 6.10. The first step is to detect a disturbance by sensing a steep angle rise or drop. The module keeps comparing the current relative angle with the relative angle 3 seconds before and the module will proceed to step 2 if the absolute value of the difference exceeds a threshold (Th1 in Figure 6.10). The algorithm starts step 2, continuously checking the maximum point for the next 5 second, upon the detection of exceeding the threshold. After the maximum is determined, the algorithm continues to locate the minimum point between the maximum point and the end of the first 5 seconds period after Th1 is exceeded. If the difference between the maximum and minimum is over a threshold (Th2 in Figure 6.10), an oscillation will be confirmed and reported.

Figure 6.10 uses an example of an upward swing angle oscillation. If it is a downward swing angle oscillation, the maximum and minimum need to be permuted to accommodate the difference. The maximum first found is probably a local maximum and will be overwritten by a following local maximum if the latter one is greater. Then the minimum location process will be restarted until the global maximum of the full 5 seconds period is found. Five seconds is an empirical value derived from the fact that the frequency of the inter-area oscillation ranges normally from 0.1 Hz to 1.0 Hz.

In the routine that implements the algorithm, the data buffer m_fOscilBuffer plays a significant role in orchestrating the smooth performance of the algorithm. The element of the buffer is a data structure that contains time stamp, frequency and phase angle. The structure of m_fOscilBuffer is shown in Figure 6.11. The length of the buffer is 2*g_nPreOsciLen+g_nPostOsciLen, where g_nPreOsciLen is the data length before the oscillation detection time and g_nPostOsciLen is the data length after the oscillation detection time. They are preset to be 50 and 150 respectively and can be customized easily according to the requirement by the applications. The buffer can be divided into three sections, section I, II, III in Figure 6.11, and their lengths are g_nPreOsciLen, g_nPreOsciLen, and g_nPostOsciLen respectively.

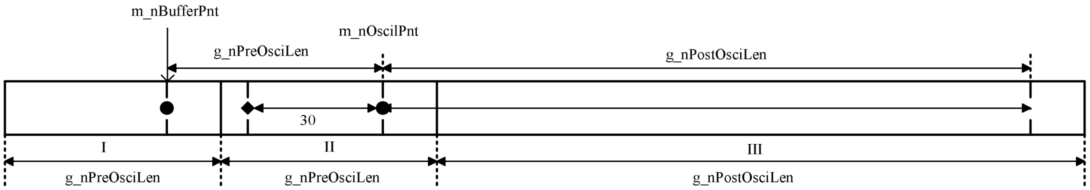


Fig. 6.11 Structure of the Buffer m_fOscilBuffer

Before a disturbance is detected, variable m_nBufferPnt functions as the pointer to the current data point shown in the first dot. The pointer increments from 0 to g_nPreOsciLen and is reset to 0 when it exceeds g_nPreOsciLen. A copy of the current data point is saved to the element at the index g_nPreOsciLen+g_nPostOsciLen shown as the second dot in the structure diagram, which makes section I and section II completely identical. By doing that, the section of the data from the index m_nBufferPnt+1 to m_nBufferPnt+g_nPreOsciLen will be the full continuous 5 seconds worth of the most recent data. To compare the current data point with the one 3 seconds before, the routine only needs to use the data point at m_nBufferPnt+g_nPreOsciLen (the second dot in Figure 6.11) and the one at m_nBufferPnt+g_nPreOsciLen-30 (the diamond in Figure 6.11) without going beyond the limits of the array definition. When a disturbance is detected, the value of m_nBufferPnt is assigned to variable m_nOscilPnt, which is used as the pointer to the oscillation detection data point. The module determines if there was an oscillation when m_nOscilPnt reaches the value of m_nOscilPnt+g_nOsciMaxPrd, where g_nOsciMaxPrd equals to 50. It finishes the detection process and saves the oscillation data to text files when m_nOscilPnt reaches the value of m_nOscilPnt+g_nPostOsciLen. At the finish point of the detection, the section of the buffer from index m_nOscilPnt-m_nPreOsciLen to m_nOscilPnt+g_nPostOsciLen is filled with continuous data from m_nPreOsciLen points before the oscillation and g_nPostOsciLen after the oscillation.

6.2.5 Flowchart of the Oscillation Detection

Summarizing the algorithm description and implementation particulars expounded in the previous section and examining the code of the server program, a flowchart of FNET oscillation detection module is presented in Figure 6.12 and Figure 6.13.

The detection module is in function OscilTrigger() of the object of a socket. When it perceives a new data point of its corresponding FDR coming and finishes preprocessing the data, the socket object invoke OscilTrigger() to start the detection process.

After entry, OscilTrigger() first parses the new data and converts it to a customized data form. The following data validity checking includes the timestamp checking and data correctness checking. If the current time stamp is not immediate after the time stamp of the last data, the program reports the data error and set the flag variable m_uDataStatus to 0. If the angle different between the current data point and the previous one exceeds 3.0 degrees, notwithstanding the correctness of the time stamps, the program also reports data error and set the flag variable m_uDataStatus to 0, due to the impossible angle change rate at 30 degrees per second in power systems. When data error status is reported, the program resets related flag variables if a disturbance is detected to be in process and suspend the oscillation detection on the related FDR until m_uDataStatus increments to 2 again after 2 consecutive minutes without error.

After the news data is confirmed correct, the program has two branches, depending on whether or not the current FDR is the reference. If the reference branch is taken, the module will just update the reference buffer g_fOscilBuffer and the reference time g_uCurrentRefTime, and then exit OscilTrigger(). If the non-reference branch is taken, the module first push the new data into the FIFO buffer and then detects oscillations using the new data.

The non-reference branch examines the working condition of the reference by checking if the FIFO buffer of the current FDR overflows, which suggests the time stamp of the reference is too lagging behind. The program will report reference error and disable the detection until the reference comes back normal. Also, it checks the status of the backup references in case the current one has a permanent malfunction.

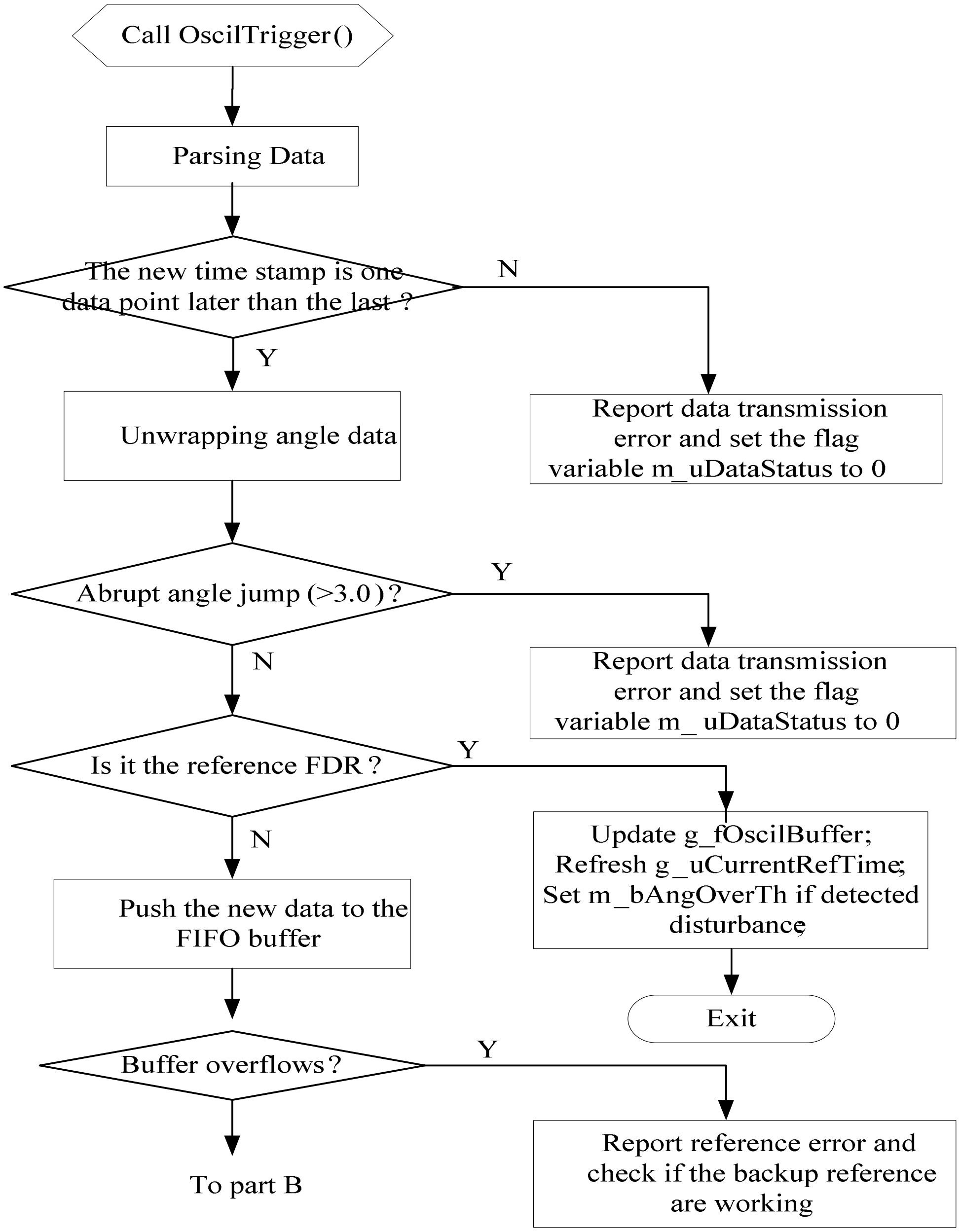

Call OscilTrigger()
Parsing Data
The new time stamp is one data point later than the last ?
N
Y
Unwrapping angle data
Report data transmission error and set the flag variable m_uDataStatus to 0
Abrupt angle jump (>3.0)?
Y
N
Report data transmission error and set the flag variable m_ uDataStatus to 0
Is it the reference FDR?
Y
N
Update g_fOscilBuffer; Refresh g_uCurrentRefTime; Set m_bAngOverTh if detected disturbance;
Push the new data to the FIFO buffer
Exit
Buffer overflows?
Y
To part B
Report reference error and check if the backup reference are working

Fig. 6.12 Part A of the Flowchart of the Angle-based Oscillation Detection

If the reference works properly, the program proceeds to find out if there is backlog in the FIFO buffer and to process those backlogged data until no newer data than the current reference time stamp g_uCurrentRefTime can be retrieved.

After aligning the data of an FDR with the reference, the program subtracts the angle value of the reference from that of the current FDR and generates the relative angle. The new relative angle is then employed to detect oscillations in accordance with the procedures shown in bottom part of Figure 6.13.

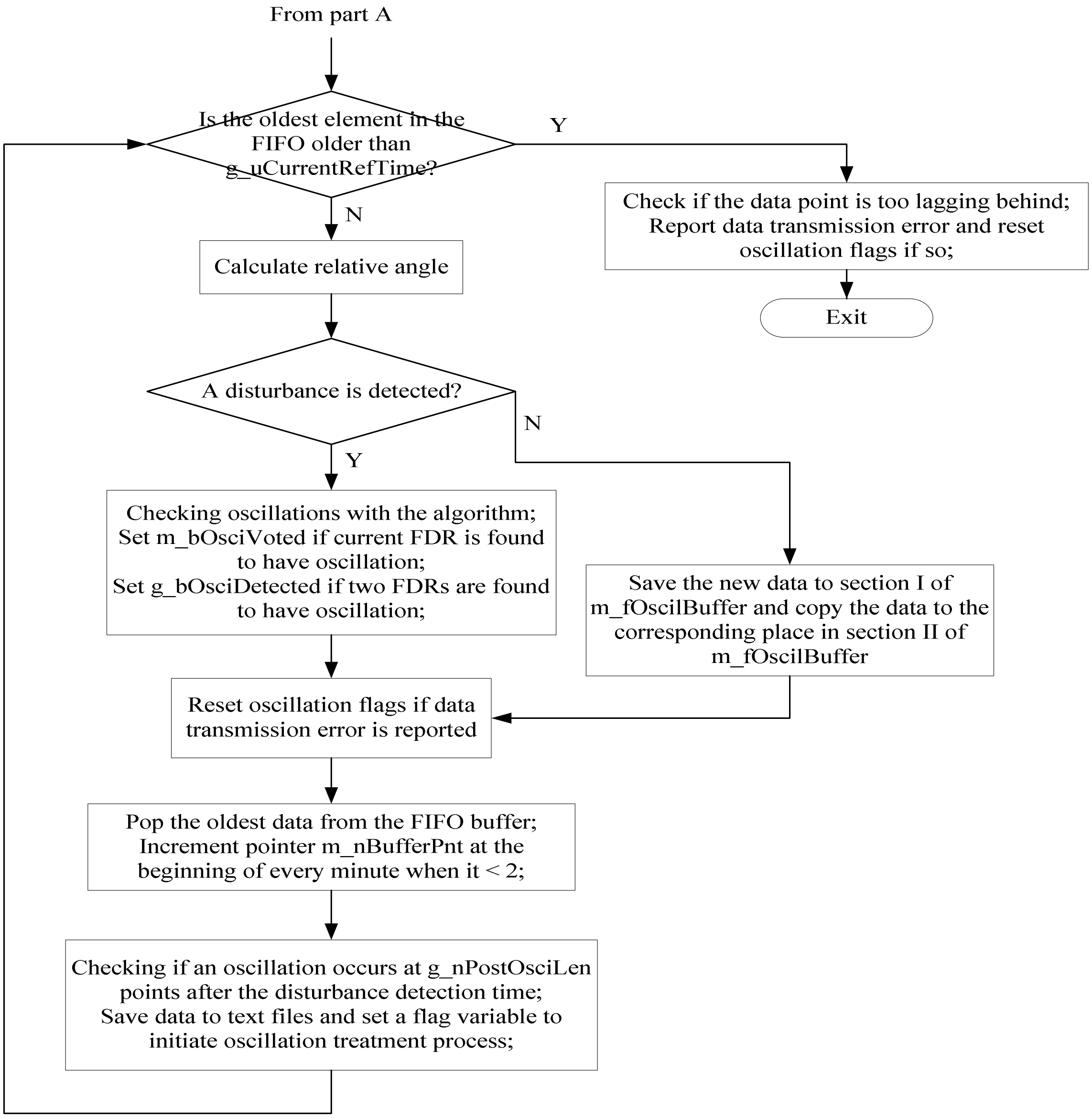

From part A
Is the oldest element in the FIFO older than g_uCurrentRefTime?
Y
N
Check if the data point is too lagging behind; Report data transmission error and reset oscillation flags if so;
Exit
Calculate relative angle
A disturbance is detected?
N
Y
Checking oscillations with the algorithm; Set m_bOsciVoted if current FDR is found to have oscillation; Set g_bOsciDetected if two FDRs are found to have oscillation;
Save the new data to section I of m_fOscilBuffer and copy the data to the corresponding place in section II of m_fOscilBuffer
Reset oscillation flags if data transmission error is reported
Pop the oldest data from the FIFO buffer; Increment pointer m_nBufferPnt at the beginning of every minute when it < 2;
Checking if an oscillation occurs at g_nPostOsciLen points after the disturbance detection time; Save data to text files and set a flag variable to initiate oscillation treatment process;

Fig. 6.13 Part B of the Flowchart of the Angle-based Oscillation Detection

The first step is to check is if there is a disturbance by the criterion of exceeding the threshold value of relative angle change over 3 seconds time period. If no disturbance is found, the second step will be saving the new data to section I and section II of the data buffer m_fOscilBuffer; whereas if a disturbance is detected, the second step will be checking if the greatest angle return after the global extremum within the next 5 seconds to the disturbance detection time exceeds the preset threshold.

There is a voting mechanism in determining the authenticity of the oscillation. Every FDR checks oscillation except the reference, and if the data of the FDR show a definite sign of oscillation in the sense of exceeding the two threshold, the flag m_bOsciVoted is set and g_uOsciNumber, the number of FDRs that detected oscillations, increments to vote for the confirmation of oscillation. If g_uOsciNumber reaches 2, a global oscillation confirmation flag g_bOsciDetected is set and an oscillation is detected. The purpose of this voting mechanism is to prevent false positives that may result from the bad data of one FDR.

The following steps are resetting the oscillation flags if data transmission error is reported, popping out the oldest data from the FIFO buffer and incrementing pointer m_nBufferPnt at the beginning of every second if it is less than 2. The final step in the loop is to check if an oscillation occurs using g_bOsciDetected at g_nPostOsciLen points after the disturbance detection time and to save data to text files. A flag variable is set to initiate oscillation treatment process if an oscillation is confirmed.

When all the data older than the current reference time are popped out from the FIFO buffer, the program goes to check if the data point is too lagging behind from the reference and reset any flag for ongoing oscillation detection usage if so. After all those steps are taken, the program exit the function OscilTrigger().

6.3 Case Analyses

The angle-based oscillation detection module has been put online for research evaluation purpose in FNET laboratory and it has reported 1 to 2 oscillation events on average per day. As a matter of fact, the power system is a dynamic system with disturbances of different sizes occurring from time to time, and as a result oscillations occur on a very frequent basis if the thresholds are set to be as low as just to avoid noise and measurement errors. Therefore, appropriate thresholds need to be set in order for the detection to run without being overwhelmed by small size oscillations.

The thresholds are then set to be 4 degrees for Th1 and 3 degrees for Th2 in Figure 6.10 to distinguish the oscillations that are triggered by large system disturbances and can be detrimental to the safety and security of the system from the oscillations that are small in size and are brought about by normal system operations which do not pose potential hazards to the system.

Three cases detected in FNET in 2009 that are representative of the different event scenarios are presented below.

The first case is an oscillation that occurred at 08:54:36 on 09/08/2009. Clear-cut oscillation signatures can be observed and the event appears to be a line trip followed by the oscillations it brought about. The maximum relative angle change is -17 degrees and the peak-to-peak oscillatory relative angle change is approximately -4.9 degrees of Plantcity, FL. The sharp and long spike in the frequency plots on the right suggests a line trip occurred. After the line trip, the frequency became unstable and finally began to oscillate with a large frequency magnitude about 4 seconds later possibly as a result of a small size generation trip near Plantcity. A spike can also be observed in the Plantcity curve in the angle plots on the left. The time of the spike in the angle plots corresponds to that in the frequency signature.

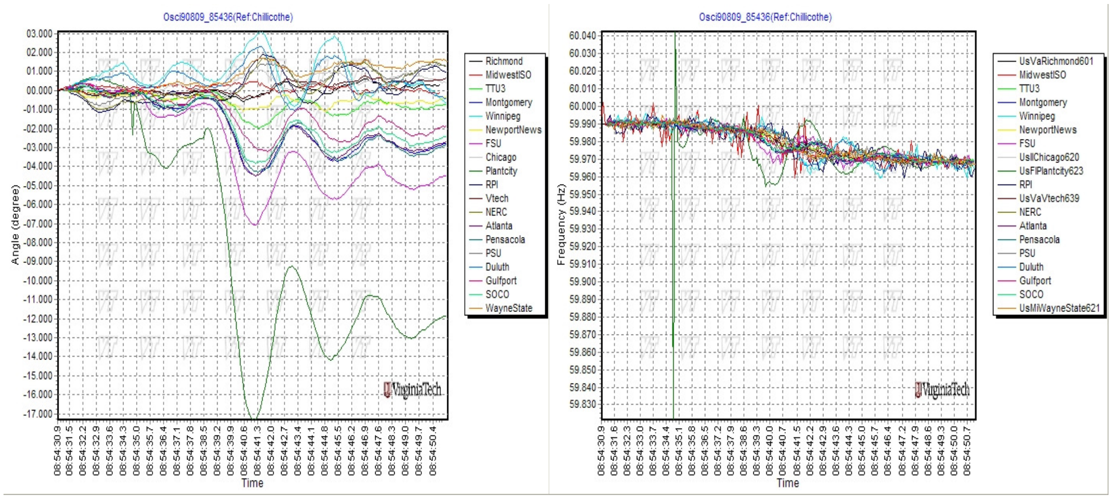


Fig. 6.14 Frequency and Angle Plots of the 09/08/2009 Event

An oscillation magnitude distribution map is generated by the oscillation magnitudes and geographic locations of the FDRs. The height of the bars is proportional to the peak-to-peak oscillatory relative angle change which can be roughly deemed as the oscillation magnitude in general. An approximate oscillation frequency is also estimated by a data analyzing program, which in this case is 0.248 Hz. From the map, we can clearly notice that the Northwest corner and the Northeast corner of the Eastern Interconnection are oscillating against the Southern part of the Eastern Interconnection.

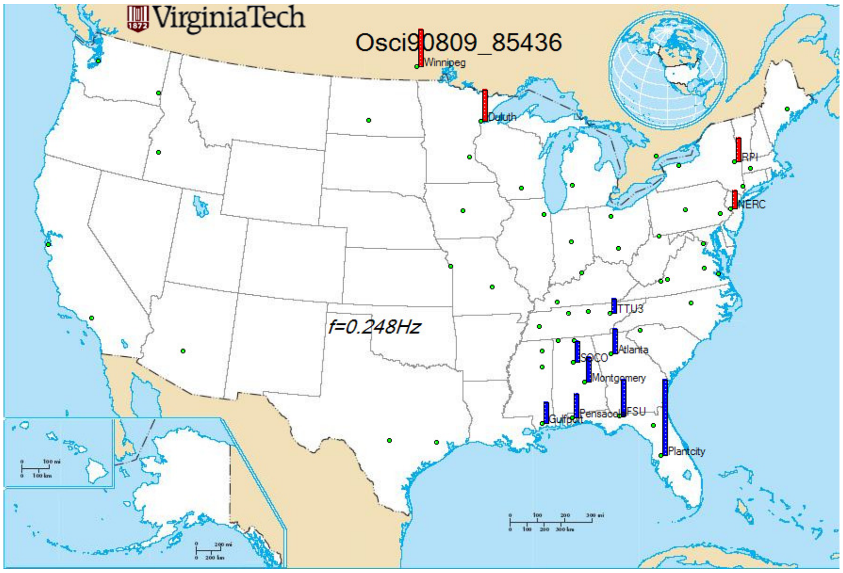


Fig. 6.15 Oscillation Magnitude Distribution Map of the 09/08/2009 Event

The second example event occurred at 16:16:08 on 09/29/2009. What distinguishes this case is the fact that almost all the FDRs are oscillating after a generation trip disturbance touched off the oscillation.

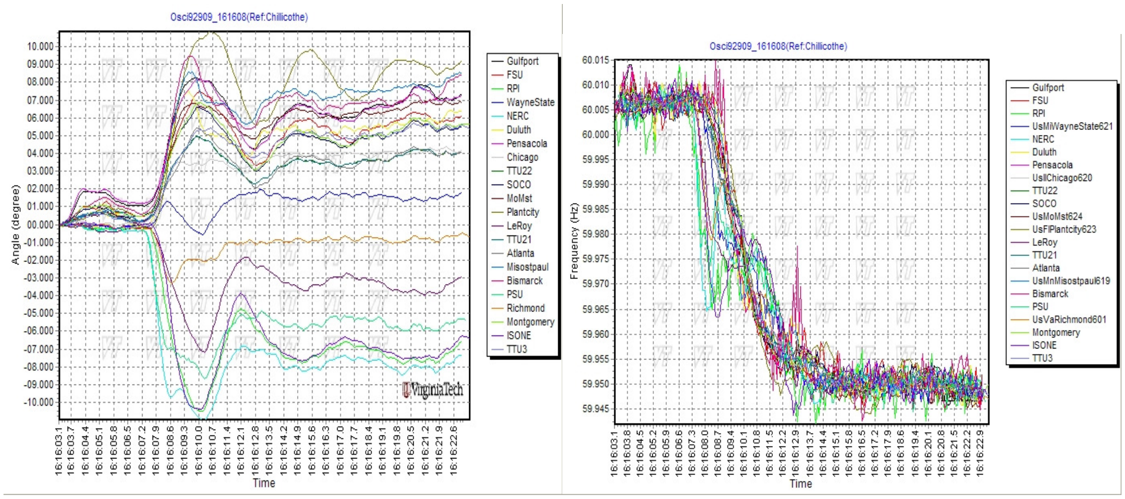


Fig. 6.16 Frequency and Angle Plots of the 09/29/2009 Event

It is apparent that the oscillations in this case are more widespread than the previous case, from the perspectives of both the number of FDRs that oscillate and the magnitudes of the oscillations. The western part of the Eastern Interconnection system is oscillating against the Northeast part of the system at the estimated frequency of 0.191 Hz.

From the two oscillation magnitude distribution maps above, it can be generalized that large oscillatory FDRs are in most cases located along the border of an Interconnection. The larger the size of the oscillation, the more remote the FDR.

The last example event occurred at 10:11:29 on 10/01/2009. The special phenomenon conspicuous in the event is that the oscillations of some FDRs sustain without much decay in magnitude, which is incongruent with the common belief that the Eastern Interconnection is a well damped power grid. The FDRs that have prominent oscillations are Plantcity, Winnipeg, Bismark, and Duluth. The signatures of relative angle change of Bismark and Dululth are overlapping each other, and in phase with that of Winnipeg FDR but still oscillating against it at a considerable magnitude. The geographic distribution of the three FDRs is geometrically close to an isosceles triangle even with the distance between Bismark and Duluth being the longest side. The reason why Bismark and Duluth act in that nature is there is a direct transmission line between the two places, which contributes to the complete synchronism in phase angle between the two cities. The oscillation of angle appears to be going on without any sign of damping in the time frame of the plots, which is definitely far beyond the damping factor limit in the Eastern Interconnection. Large under-damped oscillations in frequency are also discernable in the frequency plots. There is no drastic angle redistribution of the rest FDRs from the beginning of the plots and the end of the plots, and the system frequency also does not change much during the time frame, which means there is no significant generation-load imbalance during the oscillation time period.

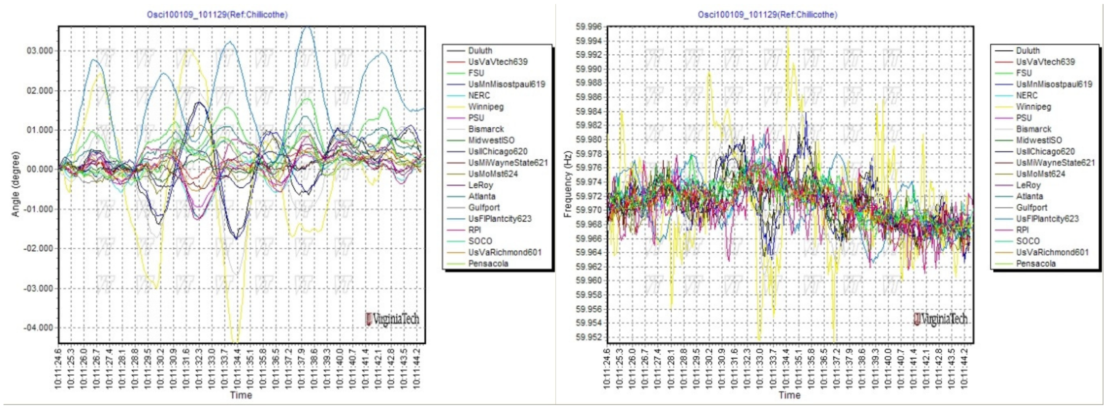


Fig. 6.17 Oscillation Magnitude Distribution Map of the 10/01/2009 Event

Chapter 7: Matrix Pencil Modal Analysis of FNET Phase Angle Oscillation Data

With the growing deployment of FDRs across the North America power grid, it is now possible to observe and analyze system wide dynamic phenomena in real-time based on the synchrophasor data gathered by the FNET server. The previous chapter presented the framework of an automatic real-time oscillation monitoring module that can detect an oscillatory event when it emerges in a real power system. As the FNET sever detects an oscillation, certain measurements prior to and immediately after the detection time are simultaneously saved to event data files for recording and analysis purposes. Undoubtedly, proper analysis proceeding from the oscillation data can enhance the situational awareness feature of FNET and add to the content of service provided to its users. Moreover, since it takes a minimum of four to five swings before growing oscillations become large enough to cause an oscillatory instability event, there does exist sufficient time for an automatic controller to take corrective actions before the oscillation becomes critical if operator alerts or control triggers can be issued to the controller promptly. The alerts and triggers are usually generated according to the parameters extracted from the oscillation data, especially the modal information of the event. Insofar as the data are ready for analytical computation upon the detection of an oscillatory event, there is definitely a possibility for the FNET server to take advantage of the real-time measurements and extract the modal information, which then can be sent to the users for situational awareness and can further be utilized for control after the technique is fully developed.

The first step towards on-line analysis of the detected oscillations is to process the data that have been accumulated heretofore with a signal processing engine specific to power system oscillation phenomena, in hopes of examining its effectiveness and the consistency of the results as well as fine tuning its settings. As is commonly observed and accepted, the power system oscillation manifest itself in the forms of the exponential sinusoidal waves of the electric quantities, such as real power, voltage, frequency and phase angle. Mathematically, the electric oscillatory waves of that nature can be decomposed into a sum of complex exponentials. The approximation of a function by a sum of complex exponentials is a problem that is at least two centuries old. There are generally three signal processing engines to implement the approximation, namely, Prony's method, the Matrix Pencil method and the Hankel Total Least Squares (HTLS) method. Fundamentally, all the three said techniques use the same sequence of data samples and vary only, but importantly, in how those samples are used in achieving the parameter estimation.

This chapter first briefly introduces the power system oscillation problem and the common approaches to interpret and analyze the data. Subsequently, the details of the three analytical engines are accounted for and their characteristics are discussed. Lastly, the angle data of the Eastern Interconnection oscillations detected in FNET during the months of September and October of year 2009 are processed with the Matrix Pencil method. Various settings of the Matrix Pencil engine are experimented and the optimal values are put forward based on the results. The performance of the method is evaluated by a Matrix Pencil modal analysis instance of a typical oscillatory event.

7.1 Introduction of Common Power System Modal Analysis Tools

Small signal stability problem of power systems has remained a concern for power engineers for past several decades. In modern day power systems, the power-flows across distant portions of the transmission network have been growing steadily to accommodate growing consumer demands. Moreover, owing to deregulation, the power transfers have also become somewhat unpredictable as dictated by market price fluctuations[35]. Those new practices results in the system operation being close to or outside the secure operating limits under severe contingencies. Therefore, large power exchanges over long transmission lines are deemed as major contributing factors for instability phenomena in power systems including oscillations[36][37]. When a power system lacks sufficient ability to damp out oscillations, it becomes prone to poorly damped or even growing oscillations under small disturbances. Even low magnitude oscillations if present over extended periods of time can be harmful to individual devices such as generator shafts. Additionally, growing inter-area oscillations from negatively damped inter-area modes can bring about widespread blackouts such as the August 10, 1996 western blackout[38].

Recent occurrences of large-scale blackouts like that all over the world reiterate the significance of developing safety net type control mechanisms, which are specifically designed to handle such unforeseen operating conditions with the ability to detect these poorly damped or growing oscillations in the early stages of the disturbance and initiate appropriate controls to damp out the oscillations before they become critical[39][40]. The controller should be motivated towards preventing oscillatory instability events such as the afore-mentioned August 10, 1996 western blackout that was caused by the negative damping of the 0.25 Hz western interarea mode[41][42].

Power systems are typically modeled as large, complex, nonlinear systems. Although power systems are inherently nonlinear, in some instances, they may respond to well-tuned linear controls[43]. In order to implement linear feedback control, the system designer must have an accurate model of sufficiently low order from which to design the control. Frequently, however, the original system is too complex or the parameters are not known with enough accuracy to produce an adequate reduced-order model. In these cases, it is desirable to extract the modal information directly from the system response to a perturbation. Using this approach, it may be possible to replace the actual dynamic model with an estimated linear model that is derived from the system output waveform. The time-varying dynamic response of a power system to a disturbance may be composed of numerous modes that must be identified.

On the other hand, however, for small disturbances, we can linearize the system around its operation point (or the equilibrium point). The linearized system can be simplified into the following form:

$$\begin{array}{l}\Delta\dot{x} = A\Delta x + b\Delta u \\ \Delta y_i = c_i \Delta x\end{array} \quad , \qquad i = 1,2,\ldots,m \qquad (7\text{-}1)$$

where Δx are the state vector, b and c are the input and output vectors respectively, Δu is the input and Δy_i is the output. As shown in (7-1), after some assumptions, the transfer functions between the input and output can have the following form,

$$G_i(s) = \frac{\Delta y_i(s)}{\Delta u(s)} = \sum_{i=1}^{n} \frac{R_i}{s - \lambda_i} \qquad (7\text{-}2)$$

where $R_i = c_i \phi_i \psi_i b$, ϕ_i and ψ_i are right eigenvector and left eigenvector corresponding to λ_i respectively.

If we apply an impulse as input to the system, the m outputs are

$$y_j(t) = \sum_{i=1}^{n} R_i \exp(\lambda_i t),\ i = 1,2,...,m \qquad (7\text{-}3)$$

If the input is not an impulse, such as a step input, the linearized system response will still be a sum of exponential terms. This is the form that our methods can be applied to for modal estimation. When $y_j(t)$ is sampled at a constant sampling period Δt, we get the following discrete form.

$$y(k) = \sum_{i=1}^{n} R_i z_i^k \qquad (7\text{-}4)$$

where $z_i = \exp(\lambda_i \Delta t)$, $\lambda_i = \sigma_i + j\omega_i$. n is called the model order, which is not known for real power system measurements.

All over the world, efforts are underway to use synchrophasor measurement-based information networks to bring the PMU synchronized phasor measurements into the control center from different parts of a large power system to determine the current state of the system[43]. The synchrophasor networks in North America usually include Phasor Measurement Units (PMU) throughout the system and Phasor Data Concentrators (PDC) usually located in control centers. The PDCs gather data from many PMUs through digital communication channels, and from the real-time wide-area PMU data, we can extract dynamic information of the system such as the presence of oscillatory modal responses.

The above being said, the problem may be posed such that given a set of measurements that vary with time, it is desired to fit a time-varying waveform of a pre-specified form to the actual waveform:

$$x_i(t) = \sum_{k=1}^{n} R_k e^{\lambda_k t} \qquad (7\text{-}5)$$

where x_i is one of the n states. The complex parameters R_k are derived from the influence of the initial conditions, whereas λ_k are the eigenvalues of A. The problem then becomes one of minimizing the error between the actual time-varying function and the proposed function by estimating the magnitude, phase, and damping parameters of the fitting function. In the problem of estimating a nonlinear waveform by a series of functions, the minimization function is given by

$$\min f = \sum_{i=1}^{N}\left[\sum_{k=1}^{n}[R_k e^{(\lambda_k t_i)}] - y_i\right]^2 \qquad (7\text{-}6)$$

where n is the number of desired modes of the approximating waveform, N is the number of data samples, and y_i is the sampled waveform.

Three algorithms are used for oscillation analysis by approximating the desired modes of the waveform of synchronized wide-area measurements: 1) multi-Prony analysis[44][45], 2) Matrix Pencil algorithm[46] and 3) the Hankel Total Least Square method (HTLS)[47]. All three algorithms decompose signals into sums of exponentially damped sinusoidal responses, and hence, are the three common tools that try to fit a sum of exponential terms to the uniformly sampled data and are useful in modal analysis of real-time wide-area data, such as FNET angle measurements. Each method has its own advantages versus weaknesses.

Prony analysis, which was originally developed by Baron de Prony in 1795 to explain the expansion of various gases, has now been successfully applied for offline signal and model identification in power systems[48][49], and a wide variety of fields[50][51][52]. It has been shown that comparing with Fourier transform based techniques, the Prony's method is more powerful when it works[53]. Issues as system's nonlinearities, noise, low signal to noise ratios (SNR) and defining the time window or data length has limited its use in real-time applications[54][55][56][57][58].

The Matrix-Pencil technique[57] is relatively new, even though its roots go back to the pencil-of-functions approach, which has been in use for some time[59][60][61][62][63][64][65][66][67]. The basic difference between the pencil-of-functions method[63] and the Matrix-Pencil (MP) approach (often termed GPOF, or generalized pencil-of-function)[68] is that even though both of them start with the same philosophy, the Matrix-Pencil method is computationally more efficient than the pencil-of-functions method. The original pencil-of-functions method was a variation of the polynomial method, and hence had the same computational bottlenecks as associated with Prony's method. The Matrix-Pencil approach is discussed in the next section. It is shown to be a one-step process, as opposed to the pencil-of-functions method that provides the solution in two distinct steps, like Prony's method. The Matrix Pencil method now is used in areas like system identification and spectrum estimation[69][70]. The singular value decomposition (SVD) step in Matrix Pencil method acts as a filtering process, thus this method can extract dynamic information from noisy data.

The Hankel Total Least Squares (HTLS) method is a special case of ESPRIT algorithm[46][47]. It is also able to extract signal poles from multi-channel noisy measurements.

7.2 Theoretical Summary of the Prony, Matrix Pencil, and HTLS Methods

7.2.1 Prony's Method

Prony's Method tries to fit a sum of exponential terms to the measured data. It has been discussed in many papers[71][72], and the main steps are summarized below. First, the sample vector can be written in the following form in (7-7).

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_n \\ \cdots & \cdots & \cdots & \cdots \\ z_1^{N-1} & z_2^{N-1} & \cdots & z_3^{N-1} \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{bmatrix} \qquad (7\text{-}7)$$

The z_i are the roots of a n th-order polynomial with unknown coefficients, and thus satisfy

$$z^n - (a_1 z^{n-1} + a_2 z^{n-2} + \cdots + a_n z^0) = 0 \qquad (7\text{-}8)$$

Left-multiplying N dimensional row vector $[-a_n, -a_{n-1}, \cdots, -a_1, 1, 0, \cdots, 0]$ to both sides of (7-7), we get the following (7-9) by using (7-8).

$$[-a_n, -a_{n-1}, \cdots, -a_1, 1, 0, \cdots, 0] \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = 0 \qquad (7\text{-}9)$$

which can be re-written as (7-10) after $N-n$ times repetitions,

$$\begin{bmatrix} y(n) \\ y(n+1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} y(n-1) & y(n-2) & \cdots & 0 \\ y(n) & y(n-1) & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots \\ y(N-2) & y(N-3) & \cdots & y(N-n-1) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \qquad (7\text{-}10)$$

Now, the procedures of Prony's Method can be summarized in the following three steps.

Step 1, Solve (7-10) and obtain the coefficients a_i.

Step 2, Calculate the roots z_i of equation (7-8).

Step 3, Solve (7-7) and find residues R_i.

Prony's Method can be extended to analyze multiple signals simultaneously. In step 1, we stack equations for each signal in (7-10). Consider a set of m signals $y_j(t), j = 1,2,\cdots,m$. Then we have a total of $(N-n)\times m$ equations with n unknown coefficients a_i in (7-10). The coefficients a_i are solved in the least-square sense. After calculating roots z_i, Step 3 becomes solving the following equations.

$$\begin{bmatrix} y_1(0) & y_2(0) & \cdots & y_m(0) \\ y_1(1) & y_2(1) & \cdots & y_m(1) \\ \cdots & \cdots & \cdots & \cdots \\ y_1(N-1) & y_2(N-1) & \cdots & y_m(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_n \\ \cdots & \cdots & \cdots & \cdots \\ z_1^{N-1} & z_2^{N-1} & \cdots & z_3^{N-1} \end{bmatrix} \begin{bmatrix} R_1^1 & R_1^2 & \cdots & R_1^m \\ R_2^1 & R_2^2 & \cdots & R_2^m \\ \cdots & \cdots & \cdots & \cdots \\ R_n^1 & R_n^2 & \cdots & R_n^m \end{bmatrix} \quad (7\text{-}11)$$

A practical issue in Prony's Method is to determine the model order n. The most common approach is to fit a high order model to the data, causing the modes corresponding to the noise to have small residue magnitudes that can be filtered out from the result.

7.2.2 Matrix Pencil Method

The main steps of Matrix Pencil method are shown below. The interested readers may refer to for more details of this algorithm. First define two matrices as follows.

$$[Y_1] = \begin{bmatrix} x(0) & x(1) & \cdots & X(L-1) \\ x(1) & x(2) & \cdots & x(L) \\ \cdots & \cdots & \cdots & \cdots \\ x(N-L-1) & x(N-L) & \cdots & x(N-2) \end{bmatrix} \quad (7\text{-}12)$$

$$[Y_2] = \begin{bmatrix} x(1) & x(2) & \cdots & X(L) \\ x(2) & x(3) & \cdots & x(L+1) \\ \cdots & \cdots & \cdots & \cdots \\ x(N-L) & x(N-L+1) & \cdots & x(N-1) \end{bmatrix} \quad (7\text{-}13)$$

where $x(k)$ are the noise-free data points. L is the pencil parameter.

Using (7-4), we can write (7-12) and (7-13) as

$$[Y_2] = [Z_1][R][Z_0][Z_2] \qquad (7\text{-}14)$$

$$[Y_1] = [Z_1][R][Z_2] \qquad (7\text{-}15)$$

where $[Z_2] = \begin{bmatrix} 1 & z_1 & \cdots & z_1^{L-1} \\ 1 & z_2 & \cdots & z_2^{L-1} \\ \cdots & \cdots & \cdots & \cdots \\ 1 & z_n & \cdots & z_n^{L-1} \end{bmatrix}_{n \times L}$

$$[Z_1] = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_n \\ \cdots & \cdots & \cdots & \cdots \\ z_1^{N-L-1} & z_2^{N-L-1} & \cdots & z_n^{N-L-1} \end{bmatrix}_{(N-L) \times n}$$

$$[Z_0] = diag\left[z_1, z_2, \cdots, z_n\right]$$

$$R = diag\left[R_1, R_2, \cdots, R_n\right].$$

$diag[]$ represents a diagonal matrix.

The matrix pencil is defined as follows

$$[Y_2] - \lambda[Y_1] = [Z_1][R]\{[Z_0] - \lambda[I]\}[Z_2] \qquad (7\text{-}16)$$

where $[I]$ is identity matrix.

When $n \le L \le N - n$, the rank of $\{[Y_2] - \lambda[Y_1]\}$ is n if $\lambda \ne z_i$[73]. However, if $\lambda = z_i$, then the i-th row of becomes zero, and the rank of $\{[Y_2] - \lambda[Y_1]\}$ is reduced by one. Hence, the parameters z_i are the generalized eigenvalues of the matrix pair $\{[Y_2];[Y_1]\}$. Or, equivalently, we can solve the ordinary eigenvalues of $\{[Y_1]^+[Y_2] - \lambda[I]\}$ to get the parameters z_i, where $[Y_1]^+$ is pseudo-inverse of $[Y_1]$. After finding parameters z_i, we solve (7-5) and obtain residues R_i.

For the actual measured data, define a new matrix $[Y]$ containing the noisy data as follows.

$$[Y]=\begin{bmatrix} y(0) & y(1) & \cdots & y(L) \\ y(1) & y(2) & \cdots & y(L+1) \\ \cdots & \cdots & \cdots & \cdots \\ y(N-L-1) & y(N-L) & \cdots & y(N-1) \end{bmatrix} \qquad (7\text{-}17)$$

Comparing (7-17) with (7-12) and (7-13), we can observe that $[Y_1]$ is obtained by deleting the last column of $[Y]$, and $[Y_2]$ is obtained by deleting the first column of $[Y]$.

Next, apply SVD to $[Y]$ as follows.

$$[Y]=U\Sigma V^H \qquad (7\text{-}18)$$

where U and V are unitary matrices. Σ is a diagonal matrix containing the singular values of $[Y]$ with descending order. The superscript H denotes conjugate transpose.

If the data were noise free, the matrix $[Y]$ has n nonzero singular values. However, when noise is present, the zero singular values are perturbed and become nonzero. Now, the Singular Value Decomposition provides an effective way of noise filtering. The singular values below some specified threshold are considered to be caused by noise and need to be set as zero.

Then we use the n significant singular values to reconstruct the original data matrix. Now, we have

$$\left[V^{'}\right]=\left[v_1, v_2, \cdots, v_n\right] \qquad (7\text{-}19)$$
$$\left[Y_1\right]=U\Sigma^{'}[V_1^{'}]^H \qquad (7\text{-}20)$$
$$\left[Y_2\right]=U\Sigma^{'}[V_2^{'}]^H \qquad (7\text{-}21)$$

where v_i are column vectors of V corresponding to the n dominant singular values. $\Sigma^{'}$ is the first n columns of Σ. $\left[V_1^{'}\right]$ is obtained from $\left[V^{'}\right]$ with the last row of $[V^{'}]$ deleted. $[V_2^{'}]$ is obtained from $\left[V^{'}\right]$ with the first row of $\left[V^{'}\right]$ deleted.

Now, we have $\left[Y_1\right]^+\left[Y_2\right]=\left[V_1^{'}*V_1^{'H}\right]^{-1}*V_1^{'}*V_2^{'H}=\left[V_1^{'H}\right]^+V_2^{'H}$. The nonzero eigenvalues of $\left[V_1^{'H}\right]^+V_2^{'H}$ are the same as $V_2^{'H}\left[V_1^{'H}\right]^+$, where $V_2^{'H}\left[V_1^{'H}\right]^+$ is an $n\times n$ matrix. Calculating the eigenvalues of an $n \times n$ matrix is computationally inexpensive since n is usually a small number. After the eigenvalues are obtained, solve (7-5) for the residues.

7.2.3 Hankel Total Least Square (HTLS) Method

HTLS method also fits an exponential decay model onto a waveform. It is proposed in papers[46][47], and the main procedures are summarized as follows.

First, form the Hankel matrix as in (7-22).

$$H = \begin{bmatrix} y_0 & y_1 & \cdots & y_{M-1} \\ y_1 & y_2 & \cdots & y_M \\ \cdots & \cdots & \cdots & \cdots \\ y_{L-1} & y_L & \cdots & y_N \end{bmatrix} \qquad (7\text{-}22)$$

where L is a parameter chosen to be larger than n, $M = N+1-L$, N is the number of measurements.

If there is no noise, i.e. $y(k) = \sum_{i=1}^{n} R_i z_i^k$, then H can be factorized as follows.

$$H = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1^1 & z_2^1 & \cdots & z_n^1 \\ \cdots & \cdots & \cdots & \cdots \\ z_1^L & z_2^L & \cdots & z_n^L \end{bmatrix} \begin{bmatrix} R_1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & R_n \end{bmatrix} \begin{bmatrix} 1 & z_1^1 & \cdots & z_1^{M-1} \\ 1 & z_2^1 & \cdots & z_2^{M-1} \\ \cdots & \cdots & \cdots & \cdots \\ 1 & z_n^1 & \cdots & z_n^{M-1} \end{bmatrix} \qquad (7\text{-}23)$$
$$= SRT^T$$

where both S and T are Vandermonde matrices.

The matrix S is shift-invariant, that is $S_{\downarrow} Z = S_{\uparrow}$, where the up (down) arrow placed behind a matrix stands for deleting the top (bottom) row of the matrix and Z is the $n \times n$ diagonal matrix whose diagonal entries are $z_1, z_2, \cdots, z_n$.

The Hankel matrix can also be decomposed by SVD as follows

$$H = U \Sigma V^H = \begin{bmatrix} \hat{U} & U_0 \end{bmatrix} \begin{bmatrix} \hat{\Sigma} & \\ & \Sigma_0 \end{bmatrix} \begin{bmatrix} \hat{V} \\ V_0 \end{bmatrix}^H \qquad (7\text{-}24)$$

where U and V and unitary matrix, i.e. $UU^H = I$, $VV^H = I$, I is identity matrix, Superscript H denotes complex conjugate transpose. Σ is a diagonal matrix with singular values on the diagonal in decreasing order. $\hat{\Sigma}$ is the sub-matrix containing first n singular values. If there is no noise in the signal, the sub-matrix Σ_0 is null. Otherwise, Σ_0 is a full matrix with small singular values on the diagonal. When noise is present, the usual procedure is to set a threshold and those singular values below the threshold are considered to be generated by noise and can be discarded.

In the noise-free case, $H = \hat{U}\hat{\Sigma}\hat{V}^H$. Comparing this equation with (7-23) and (7-24), we have the following relationship.

$$\hat{U} = SQ \tag{7-25}$$

where Q is an $n \times n$ nonsingular matrix. Deleting the first row in the above equation gives $\hat{U}_\downarrow = S_\downarrow Q$, while deleting the last row gives $\hat{U}_\uparrow = S_\uparrow Q$.

Thus $\hat{U}_\downarrow$ and $\hat{U}_\uparrow$ are related by the following equation.

$$\hat{U}_\uparrow = S_\downarrow ZQ = \hat{U}_\downarrow Q^{-1} ZQ = \hat{U}_\downarrow \tilde{Z} \tag{7-26}$$

where $\tilde{Z} = Q^{-1}ZQ$, which has the same eigenvalues as Z, i.e. $z_1, z_2, \cdots, z_n$.

In a noisy case, (7-25) does not hold exactly. In that situation, we solve $\tilde{Z}$ by total least square method. After calculating $\tilde{Z}$, the signal poles are calculated as the eigenvalues of $\tilde{Z}$.

When analyzing multiple signals simultaneously, it suffices to simply replace H in (7-22) by the horizontally-stacked Hankel matrices as follows.

$$H_{series} = [H_1 \quad H_2 \quad \cdots \quad H_m] \tag{7-27}$$

where m is the number of signals.

The last step, calculating residues R_i, is the same as that of Prony’s method.

7.3 Modal Analysis with Matrix Pencil

Because of its robustness to noise and relatively higher speed among the three algorithms, the Matrix Pencil modal analysis engine is more widely used in power system applications. This section will use Matrix Pencil tool written in MATLAB to analyze a typical inter-area oscillation that was detected during September and October of 2009, and proceed to make some observations based on the analysis results.

7.3.1 Parameter Fine Tuning

Before any research on modal analysis is conducted, some of the parameters of the Matrix Pencil engine need to be determined to accommodate the characteristics of the FNET angle data. The customization of the parameters is more of a fine tuning process by the comparison the consistency of the results obtained from different reasonable parameter settings. The parameters involved here are the number of measurements in (7-17) and the modal analysis start time.

Pencil Parameter L is also an important factor that can affect the performance of the algorithm, but it is empirically set to be 0.4~0.6 of the number of measurements in many of the Matrix Pencil engine implementation practices. For FNET Matrix Pencil modal analysis instances, the value is pre-configured at 0.6, making the number of measurements the sole parameter that determines the dimension of matrix $[Y]$in (7-17). This way to deal with the Pencil Parameter L causes the number of measurements more mission-critical in terms of the performance of the Matrix Pencil analysis engine. Since the data rate of FNET system is fixed at 10 points per second, the number of measurements is equivalent to the window size of the data fed into the engine.

The modal analysis start time is equally significant in that it decides which segment of the data sequence the engine uses. Different segments of the data sequence have distinct data qualities and thus may result in various levels of performance of the engine. For example, the nonlinearities in the first few swings from factors such as saturations and exciter limiters can result in an underestimate (or overestimate) of damping ratio for the inter-area mode. The rationale for this phenomenon is that the concepts of eignevalues and modal analysis, from linear system theory, are only applicable for small disturbances away from an equilibrium point. It is therefore imperative that the results of modal analysis be ignored when the system responses are excessively far away from the equilibrium point.

Another factor that many cause inaccurate estimation of the mode information of an oscillation is that as the oscillation decays, the Signal to Noise Ratio (SNR) decreases and may enter a very fuzzy zone that may exceed the noise tolerance of the modal analysis engine. The end of the data window is the area that has the smallest SNR and thus is prone to make the engine generate an incorrect result. The approaches to avoid the fuzzy zone are, 1) to decrease the window size, or 2) if the data window size is invariable, the modal analysis start time must be no later than a certain point. This poses a difficulty for determining the modal analysis start time because there is definitely a balance to strike between avoiding the incipient stage of an oscillation and steering clear of the fuzzy zone of too small SNR. The incipient stage of the data contains dynamics and nonlinearities that may result in misestimating the oscillation modal parameters, whereas the fuzzy zone may also bring about inaccurate computation result due to high content of noise.

A fairly representative oscillation instance that was detected in FNET at 06:59:30 on 10/18/09, shown in Figure 7.1, is used for the Matrix Pencil parameter fine tuning.

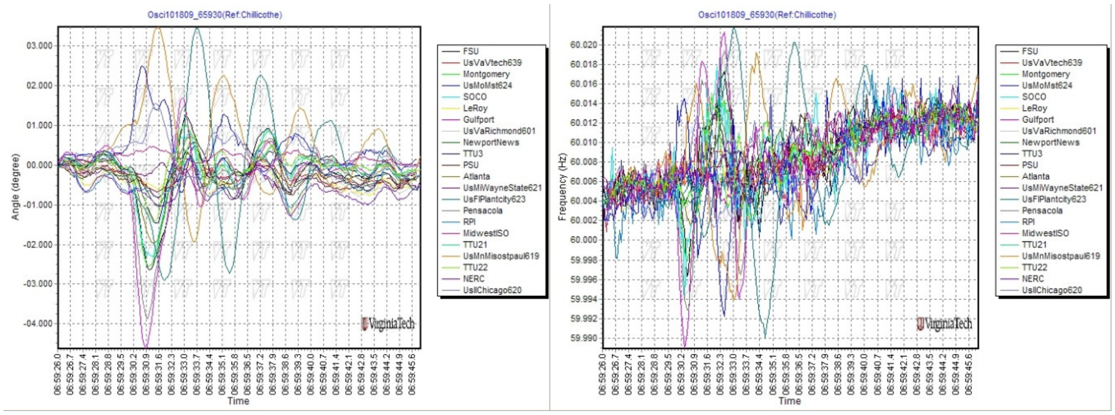


Fig. 7.1 10/18/09 06:59:30 Event

Modal analyses using 5 second sliding window and 10 second sliding window with the analysis start time moving from the event detection time and 15.0 seconds afterwards are performed to customize of the parameters for MATLAB Matrix Pencil engine. The sliding window moves at the pace of 0.5 second per increment from 0.0 to 15.0 seconds after the oscillation detection time. The analysis start time is identical to the data window start time, and the number of measurements is 50 data points and 100 data points for the 5 second sliding window and 10 second sliding window respectively under the current 10 points per second FNET data rate. An optimal start time and the better length of the sliding window are obtained from the modal analysis results of the 62 Matrix Pencil computations. Sixty two is 31 increments from 0 to 15.0 seconds at 0.5 second per increment multiplied by the two window size options, namely, 5 seconds and 10 seconds.

In Figure 7.1, there are so many FDRs that do not exhibit strong oscillations with clear-cut exponentially decaying sinusoidal curve whose sinusoidal content can suppress the effect of noise. However, the signatures of PlantCity and StPaul units are conspicuously outlined with a decaying sinusoidal curve that is eminently expressive of the modal information of the oscillation. Therefore, only the data of the two units are utilized in the process of obtaining the optimal setting.

First, the angle data together with the GPS time stamps of the two particular units are extracted from the text files and converted into MATLAB format. Then, the MATLAB data are fed to the Matrix Pencil engine and processed by the single sequence Matrix Pencil algorithm. The results of the mode extraction process include the oscillation frequency, the damping ratio and the mode shape angle difference between the two units. Table 7.1 and Table 7.2 are the lists of the oscillation frequency, damping ratio and relative mode shape angle of the two selected units of the dominant mode for 5-second sliding window and 10-second sliding window respectively.

Freq, DR and Ang stand for the frequency, damping ratio, and mode shape angle difference respectively. PC and StP in the parentheses represent the results of the PlantCity and StPaul units. The dimensions of the frequency and angle are Hz and degree. The ‘Start Time’ is the parameter of the modal analysis start time, and is time interval in seconds counting from the oscillation detection time to the measurement time of the first data point of the data window.

Start Time	Freq(PC)	DR(PC)	Ang(PC)	Freq(StP)	DR(StP)	Ang(StP)
0.0	0.2744	-0.0856	174.457	0.2701	-0.0106	23.174
0.5	0.2792	-0.005	-145.69	0.1775	0.25538	132.38
1.0	0.2909	0.04418	-108.07	0.2661	0.22528	122.78
1.5	0.2908	0.0338	-54.494	0.2282	0.02615	-158.8
2.0	0.285	0.05829	0.26498	0.2486	0.25977	-141.2
2.5	0.2842	0.08846	50.3499	0.2175	0.32935	-71.69
3.0	0.2889	0.10517	96.2359	0.2096	0.24492	-22.58
3.5	0.2763	0.10029	157.767	0.2007	0.05697	39
4.0	0.2749	0.14307	-153.41	0.2079	-0.0043	72.419
4.5	0.2672	0.08205	-93.887	0.2273	-0.0096	98.398

5.0	0.2736	0.10576	-50.269	0.2688	-0.0121	106.14
5.5	0.2765	0.06606	-1.9977	0.2844	-0.0639	144.32
6.0	0.2733	0.11233	46.7991	0.2651	0.02649	-157.7
6.5	0.2781	0.11456	92.4744	0.264	0.15496	-119.4
7.0	0.2828	0.10143	140.111	0.2596	0.16486	-68.05
7.5	0.2731	0.14944	-165.85	0.2378	0.11574	-1.584
8.0	0.273	0.19253	-118.52	0.2118	0.17035	61.801
8.5	0.2735	0.13996	-118.52	0.2078	0.10583	110.99
9.0	0.28	0.00793	-7.0069	0.2411	0.07573	129.18
9.5	0.2586	0.17035	36.7577	0.2539	-0.0891	168.55
10.0	0.3293	0.01465	56.3192	0.2797	-0.0676	-149.3
10.5	0.3312	0.04084	111.962	0.2583	-0.0459	-101.1
11.0	0.329	0.05083	170.677	0.3062	0.00007	-95.8
11.5	0.3308	0.10753	-132.05	0.3505	-0.0698	-62.91
12.0	0.3206	0.08378	-61.283	0.3152	0.04321	5.4699
12.5	0.31	0.14615	-3.6939	0.3382	0.09437	43.082
13.0	0.2583	0.44724	69.4776	0.3042	0.06655	126.23
13.5	0.2012	0.33652	154.717	0.2878	0.10373	-168.1
14.0	0.2111	0.20298	-171.18	0.2998	0.10726	-123.6
14.5	0.2541	0.22047	-157.61	0.2602	-0.0858	-30.78
15.0	0.246	-0.29	-134.55	0.2772	-0.0834	1.21

Tab. 7.1 Modal Analysis Results of the 5 Second Sliding Data Window

Start Time	Freq(PC)	DR(PC)	Ang(PC)	Freq(StP)	DR(StP)	Ang(StP)
0.0	0.2825	0.04701	174.457	0.2512	0.095	156.68
0.5	0.281	0.06149	-149.91	0.2506	0.09636	84.051
1.0	0.2827	0.07372	-103.31	0.2464	0.07306	135.4
1.5	0.2793	0.07534	-46.944	0.2441	0.07201	-176.9
2.0	0.2785	0.08073	3.57912	0.2481	0.06854	-140.5
2.5	0.2766	0.08657	56.7554	0.2467	0.06428	-94.41
3.0	0.2755	0.09559	108.592	0.2447	0.08114	-45.14
3.5	0.2755	0.10542	157.96	0.2458	0.07447	-3.143
4.0	0.2749	0.10913	-151.6	0.2491	0.07035	34.237
4.5	0.2742	0.1096	-99.397	0.2495	0.05748	78.959
5.0	0.2767	0.10761	-51.148	0.2465	0.05556	129.5
5.5	0.276	0.11227	-1.3471	0.2518	0.05029	168.04
6.0	0.2857	0.08879	45.3872	0.2534	0.05648	-148.9
6.5	0.2977	0.08037	79.6954	0.2552	0.05992	-108.8
7.0	0.2269	0.1464	128.634	0.2544	0.08994	-62.57
7.5	0.24	0.18224	-148.25	0.2511	0.13637	-9.033
8.0	0.3076	0.03413	-132.45	0.2497	0.11982	37.754
8.5	0.301	0.1215	-81.132	0.2619	0.06511	68.832
9.0	0.2916	0.14336	-18.436	0.2811	0.03042	95.152
9.5	0.2847	0.11471	43.2628	0.0888	0.43706	-41.72
10.0	0.2439	0.11137	143.469	0.2926	0.01092	176.79
10.5	0.3781	0.06511	69.5122	0.2918	0.05454	-132.1
11.0	0.4003	0.05849	107.341	0.2932	0.03325	-86.3
11.5	0.3993	0.07891	176.342	0.2877	0.01735	-25.76

12.0	0.4072	0.06818	-120.81	0.2843	0.00853	34.958
12.5	0.4075	0.10549	-45.561	0.3474	0.14129	20.248
13.0	0.2164	0.10113	107.147	0.2892	0.13654	121.16
13.5	0.3252	0.10453	150.247	0.2704	0.03032	-154.7
14.0	0.2587	0.24796	-176.47	0.2582	0.0371	-93.49
14.5	0.1029	0.39884	-32.04	0.3059	0.12213	-109
15.0	0.1298	0.17347	-23.136	0.3218	0.1165	-66.54

Tab. 7.2 Modal Analysis Results of the 10 Second Sliding Data Window

Lastly, the optimal length of sliding window and start time are determined after comparing and weighing the numerical results of the 62 modal analyses.

The determination of the optimal parameters can be better illustrated by plotting the results and comparing the shapes of the curves for different FDRs and settings. The curve or the group of contiguous points on the curve that demonstrate the best consistency and linearity should be identified as the optimal parameters.

Figure 7.2 is the oscillation frequency plots of the dominant mode of the PlantCity and StPaul units when 5 second sliding data window and 10 second sliding data window are applied. The red, blue, green and magenta plots are for PlantCity with 5-second sliding window, PlantCity with 10-second sliding window, StPaul 5-second sliding window and StPaul with 10-second sliding window respectively.

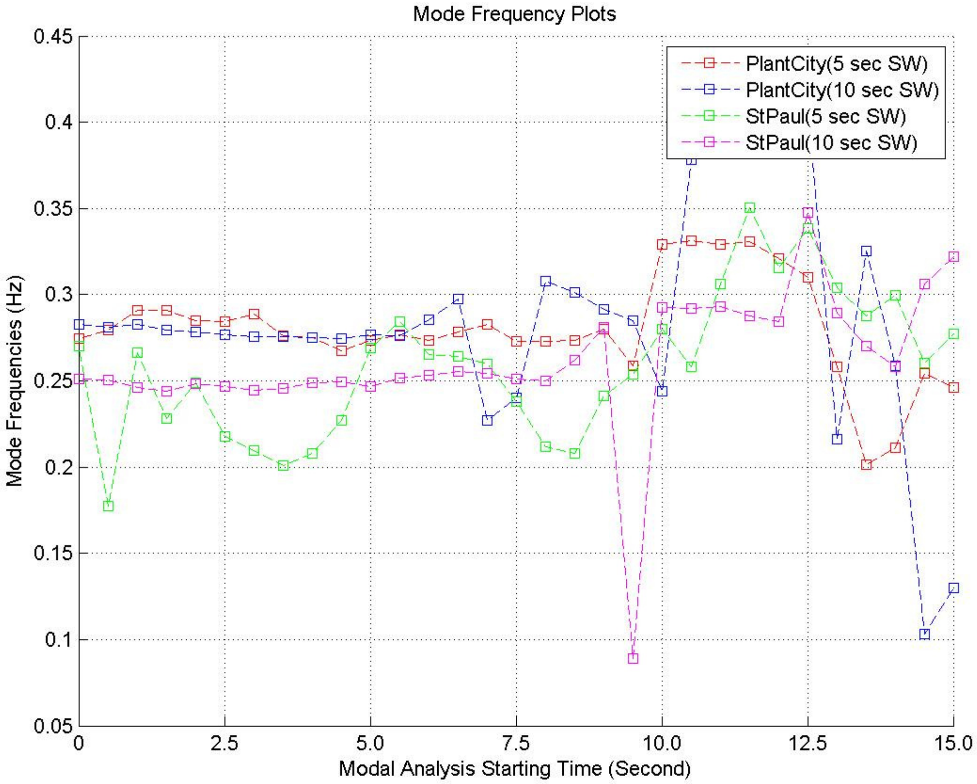


Fig. 7.2 The Frequencies of the Dominant Mode

As can be observed from the plots, the four lines after the start time of about 7.0 seconds are fairly irregular and hard to be identified with any recognizable patterns. This phenomenon is theoretically caused by the fact that the angle data enter a fuzzy zone with a low Signal to Noise Ratio (SNR) as the oscillation decays. The content of the noise exceeds the tolerance limit of the Matrix Pencil algorithm and therefore upsets the performance of the modal analysis engine. As a result, the oscillation frequencies after the start time of 7.0 seconds drift haphazardly and appear to be in consistent with those before the start time.

The blue line, the plots of the frequencies of the PlantCity unit with 10-second sliding window fluctuate much less than the red line, the plots of the frequencies of the PlantCity unit with 5-second sliding window, when the start time ranges from the detection time to 5.5 seconds afterwards. The measurements on the blue line of that range maintain a far better linearity than those on the red line, which is substantiated by the fact that there is less deviation, and the measurements hold more to a constant frequency value. Even wider discrepancy between 10-second sliding window and 5-second sliding window is present in the plots of the frequency measurements of the StPaul unit in which the green line and the magenta line are for the 5-second window size and 10-second window size respectively. The segment of the green line of the 5-second sliding window before the start time of 5.0 seconds jumps up and down quite extravagantly, whereas the magenta line of the 10-second sliding window keeps at the value of 0.25 Hz rather precisely. The stark contrast of the two data window size settings implies that there is a noticeable impact of the nonlinearity of data upon the result of the modal analysis computation during the first one swing or two of an oscillation. It also attests to the fact the 5 seconds worth of the angle data is insufficient for successfully implementing the Matrix Pencil analysis of the FNET angle data, while 10 seconds worth of data can greatly cancel the nonlinearity effect and yield a decent outcome.

Another meaningful observation of the oscillation frequency plots is that the green line stays fairly close to the magenta line from the start time of 5.0 seconds to the time when the SNR enters the fuzzy zone. This can be accounted for by the assumption that the nonlinearity effect fades to such a negligible degree that it does not affect the Matrix Pencil engine with a 5-second sliding window.

From the results in the tables and the plots in Figure 7.2, it can be concluded that 10-second sliding window demonstrates far better performance in terms of the oscillation frequency estimation of the Matrix Pencil engine than 5-second sliding window and hence should be the better choice for the optimal length of data window. It can also be inferred that when it comes to the determination of the optimal start time of the modal analysis, there exists a balance to be stricken between avoiding the nonlinearity of the early stage of an oscillation and steering clear of the low SNR data of the late stage as the oscillation decays. The time at about 5.0 seconds after the detection time is on the whole shown to be a proper choice for the optimal modal analysis start time.

Figure 7.3 is the damping ratio plots of the dominant mode of the PlantCity and StPaul units for both 5-second sliding data window and 10-second sliding data window. The red, blue, green and magenta lines represent the same FDR names and window size settings as in Figure 7.2.

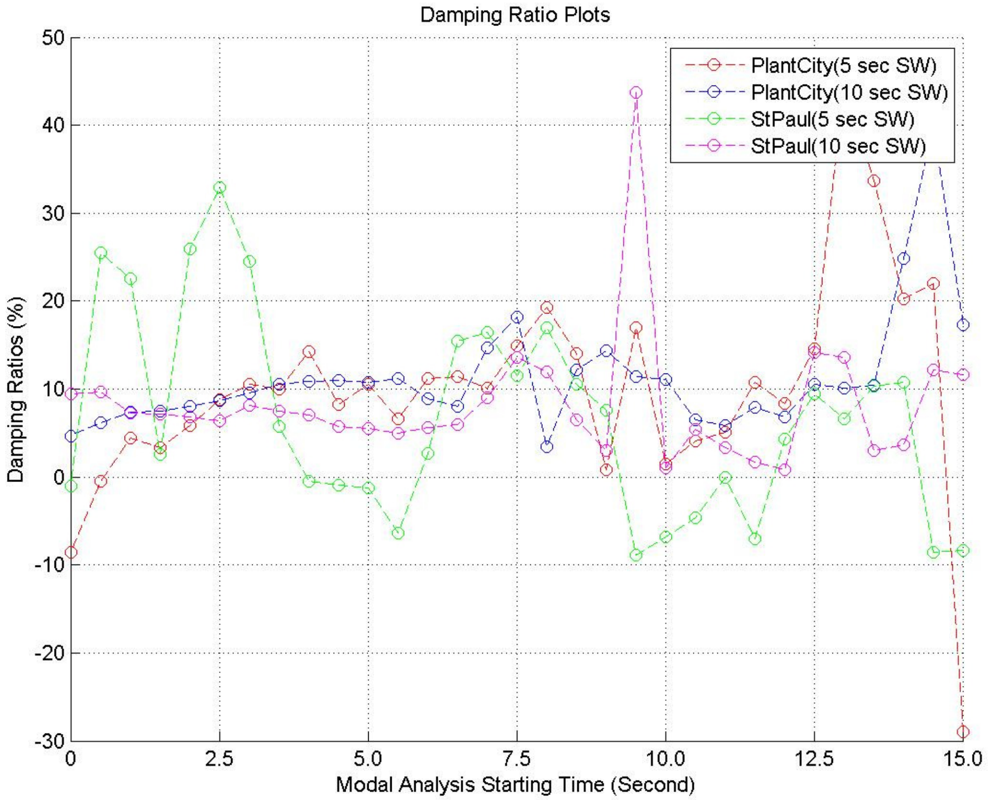


Fig. 7.3 The Damping Ratios of the Dominant Mode

The same phenomenon can be observed in Figure 7.3 that the four lines after the start time of about 7.0 seconds deviate haphazardly and appear to be irregular and inconsistent with those before that start time, which substantiates the fact that the angle data enter a fuzzy zone with a low Signal to Noise Ratio (SNR) as the oscillation decays and the content of the noise exceeds the tolerance limit of the Matrix Pencil algorithm. The blue line and magenta line are much more consistent and much less jittery than the red line and the green line when the start time is prior to 7.0 seconds, connoting the superiority of the 10-second window size over the 5-second window size. The nonlinearity effect of the first one swing or two of the oscillation can also be perceived from the fact that the first three seconds of the four lines are not as in horizontal alignment as the segments around 5.0 seconds. However, after the first few swings, the analysis engine gives consistent and accurate damping ratio estimates before it enters the fuzzy zone. This shows the importance of avoiding the adverse effect of nonlinearities and accentuates the necessity that the swings immediately after detection be ignored in the Matrix Pencil engine. The consistency of the damping ratios in Figure 7.3 on the whole is worse than the frequencies in Figure 7.2, which is a common problem for all modal analysis techniques[74].

Generally speaking, Figure 7.3 also corroborates that the 10-second sliding window and the start time of about 5.0 seconds after detection are a better choice for the optimal modal analysis start time and the optimal length of data window.

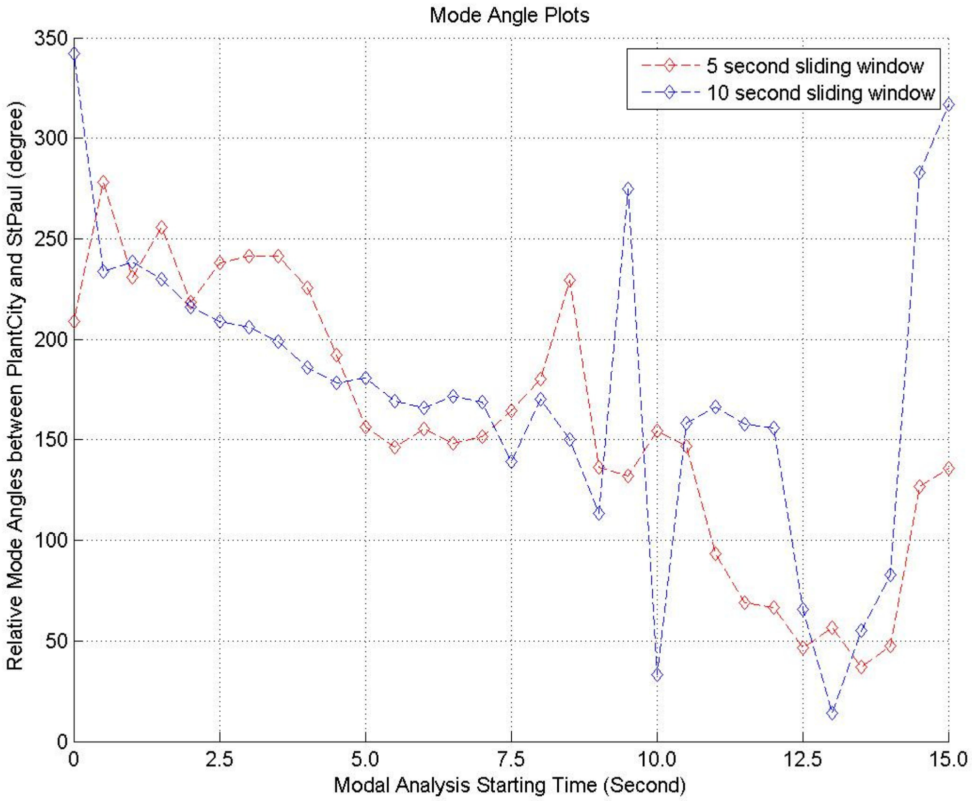


Fig. 7.4 The Mode Shape Angle Differences of the Dominant Mode

Figure 7.4 is the mode shape angle difference plots of the dominant mode. The mode shape angle difference denotes the difference of the mode shape angles of the PlantCity and StPaul units. The red line and the blue line are the relative mode shape angles for 5-second sliding window and 10-second sliding window respectively.

As can be seen from Figure 7.4, the segment approximately from 1.0 second to 6.0 seconds of the blue line is the only part of the two lines that is roughly a straight line, which is what the mode angle differences theoretically are supposed to be like, and thus can be regarded as correct computations of the mode information. This observation of the two plots in Figure 7.4 can also be interpreted from a similar perspective to the previous two figures. Insufficient number of measurements for 5-second sliding window causes its plots to be very cluttered. Low SNR data after the start time of 6.0 seconds make the two lines after that point look chaotic. The presence of nonlinearity in the early portion of the data renders the first two relative mode angle estimates adrift from the following ones. Although the two lines are somewhat chaotic and do not assume a clear form, there is still a discernible descending trend in them, especially the straight part of the blue line. The difference of the oscillation frequencies of dominant modes of the two units shown in Figure 7.2 causes a change of relative mode angle at a constant velocity, i.e. a descending trend.

The plots in the three figures above collectively point to the result of the parameter fine tuning: 5.0 seconds after detection time for the modal analysis start time, and 10-second window size, equivalently 100 data points, for the number of measurements. It should be noted that by the end of the sliding window, approximately at 15.0 seconds after the detection time of the example oscillation if the two proposed optimal values are used, the peak-to-peak oscillation magnitude decays to about 1.0 degree. Since the fuzzy zone with low SNR cannot be strictly defined, the 1.0 degree can be employed as an expedient to determine if the end of the sliding window enters the low SNR zone.

7.3.2 Case Study

Since the oscillation detection module described in Chapter 6 was put online for research evaluation in August 2009, a great number of oscillatory events have been detected and corresponding event files have been saved on the server. Aiming at demonstrating the Matrix Pencil tool on the analysis of the FNET angle data, a rather representative oscillation is pre-processed and then analyzed by the MATLAB Matrix Pencil engine using the customized settings that were determined previously.

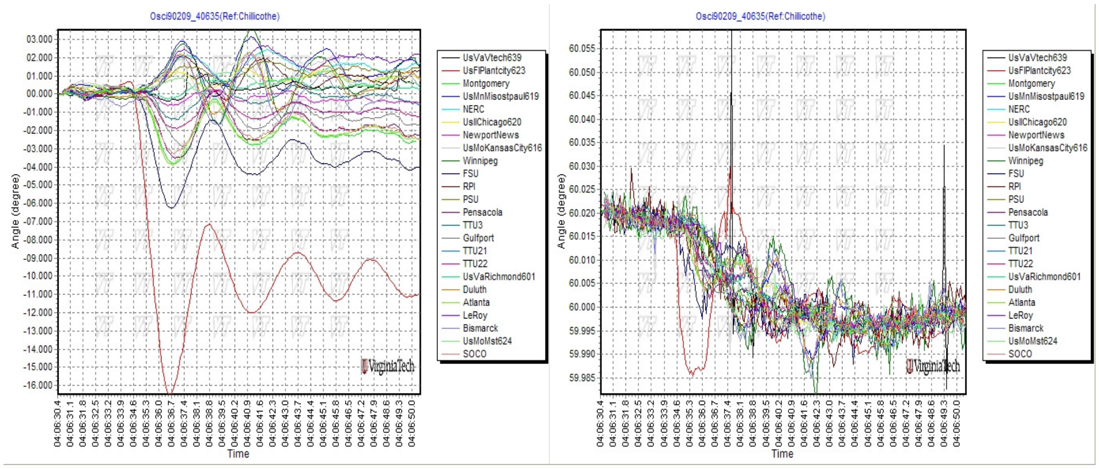


Fig. 7.5 Plots of the 09/02/09 04:06:35 Oscillation

The oscillation occurs at 04:06:35 UTC on 09/02/09 in the Eastern Interconnection. Figure 7.5 shows its relative angle plots on the left and its frequency plots on the right. The angle reference in the relative angle plots is Chillicothe, Ohio. Prominent frequency oscillations can be observed on the PlantCity unit and the Winnipeg unit, which are the red curve and the green curve in the frequency plots on the right respectively. In the angle plots on the left, all the curves are clustered together except the PlantCity unit, corresponding to the predominant frequency profile of the PlantCity unit on the right. It should be noted that the outlier condition of the PlantCity unit is not caused by its oscillation magnitude being larger, but rather by the manner of the frequency drops of the FDRs followed by a generation trip and their different amounts of the post-disturbance power angle redistributions associated with the frequency reactions. Weak interconnection between Florida peninsular and the rest of the Eastern Interconnection grid plays a major role in making the PlantCity plots deviating far from the rest of the curves.

The original text oscillation data saved on the server upon the detection of the oscillation are converted into the MATLAB format that the Matrix Pencil modal analysis engine requires. Then single sequence modal analyses are conducted using the relative angle data. The results are given in Table 7.3.

09/02/09 04:06:35	Magnitude(degree)	Frequency(Hz)	Damping Ratio	Angle(degree)
RPI	0.782166	0.2044	20.79%	-26.19
Winnipeg	1.556642	0.2573	14.29%	17.62
StPaul	0.786987	0.2369	7.60%	31.82
Bismarck	0.950296	0.2382	6.14%	37.58
Duluth	0.830511	0.2337	8.28%	46.71
Plantcity	0.690500	0.2368	2.05%	168.72
FSU	0.738963	0.2277	12.19%	187.92
Pensacola	0.948911	0.2182	25.42%	188.30
Montgomery	0.636261	0.2208	18.17%	190.02
SOCO	0.630226	0.2158	20.48%	192.14
Gulfport	0.990400	0.1958	31.98%	198.06

Tab. 7.3 Modal Analysis Results of the 09/02/09 04:06:35 Event

The table only lists the results of the units with significant oscillation element whose RMS oscillation magnitudes exceed 0.5 degree. The oscillation frequency estimates in the table are quite uniform, centering around 0.226 Hz with small deviations. The sizes of the oscillation in the first column of the table are also in the same order of magnitude. However, the damping ratios of the units vary from 2.05% to 31.98%, an exceedingly wide range for an oscillation event in the same electric power grid. The widespread damping ratio values are embodied in the angle plots in which the oscillations of the other FDRs decay much faster than that of the PlantCity one. Since 3% or 5% is practically the threshold for an under-damped power system oscillation that is hazardous to the operation of the power system and may even jeopardize the its safety, the excessive low damping ratio of the Plantcity area is a concern that entails due attention. The damping of the other FDRs is found to be sufficient in that the damping ratios are well above 5% and the corresponding angle oscillations in the plots all die down in 3~4 oscillatory cycles. The anomaly of PlantCity damping ratio can to some extent be attributed to the relatively weak linkage between the Florida peninsular where the PlantCity FDR is deployed and the rest of the Eastern Interconnection grid considering the fact that there are only two 500 kV level transmission lines transferring large amounts of electric power from the heavily loaded Orlando and Miami area to the rest of the EI system.

As the most pronounced oscillations are the PlantCity and Winnipeg FDRs, the modal analysis overviews of the two FDRs are selected to be presented in Figure 7.6 and Figure 7.7. The oscillation frequencies of the dominant mode of the two units are fairly close in Table 7.3, whereas the corresponding mode phases are almost opposite. This means the PlantCity area is oscillating against the Winnipeg area in that oscillation mode. The difference in the damping ratios of the two FDRs also manifests itself in the distinctive decay speeds of the dominant mode. The magnitude of the Winnipeg unit dwindles to almost 1/10 in only two swings, while that of the PlantCity unit remains nearly unchanged. The content of the other modes is observed to be only a fraction of that of the dominant mode, and the sinusoidal waveform of the dominant mode resembles the curve of the input signal very well. The figures and the table concerning the modal analysis all prove the results of the Matrix Pencil engine in this case are quite successful.

The mode shape angles of the dominant mode in the fifth column of Table 7.3 demonstrate that the FDRs with significant oscillations are grouped into two categories that are approximately 180 degrees out of phase with each other. The mode shape angles of one category range from -26.19 degrees to 46.71 degrees, while those of the other category range from 168.72 degrees to 198.06 degrees. This phenomenon indicates that one group of FDRs is oscillating against another.

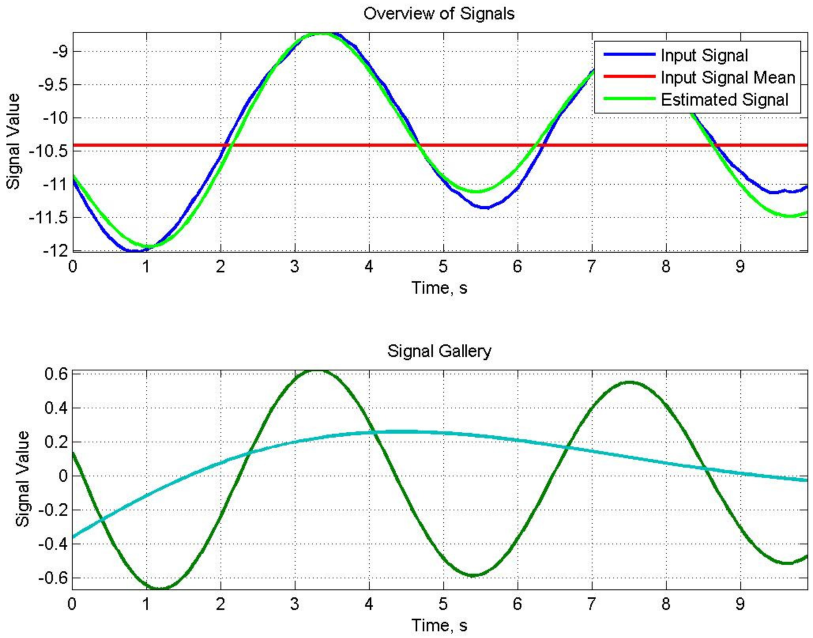


Fig. 7.6 Modal Analysis Overview of the PlantCity FDR

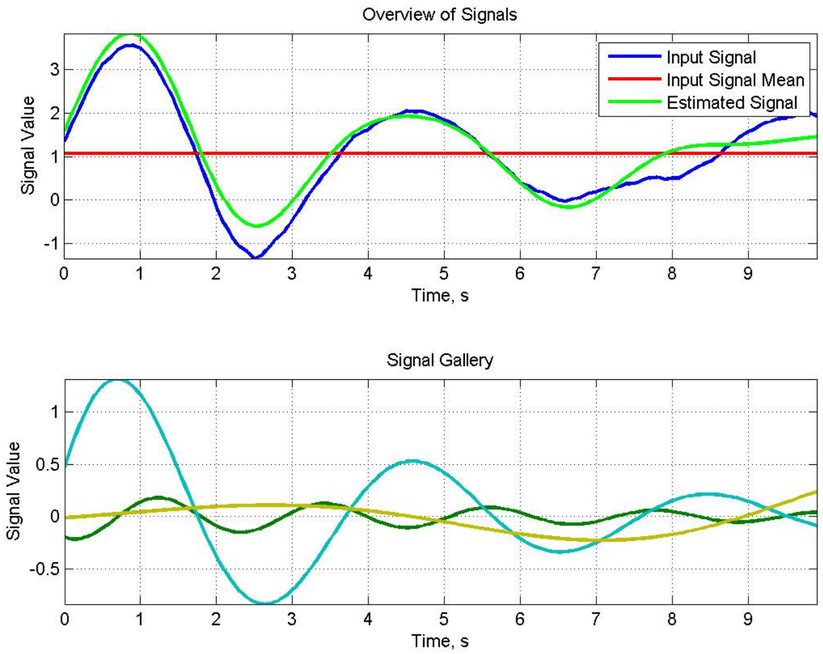


Fig. 7.7 Modal Analysis Overview of the Winnipeg FDR

Figure 7.8 is the mode shape distribution map of the oscillation. The length of the bars denotes the oscillation magnitude and the two colors stand for the two angle categories. The red color is for the category whose mode shape angles are around 0 and the blue color for the category whose mode shape angles are around 180 degrees. This map vividly shows that the

Southern part of the EI system is in oscillation with the Northeast corner and Northwest corner of the system, which is a rather reasonable oscillation distribution and is also the most observed EI oscillation mode in FNET.

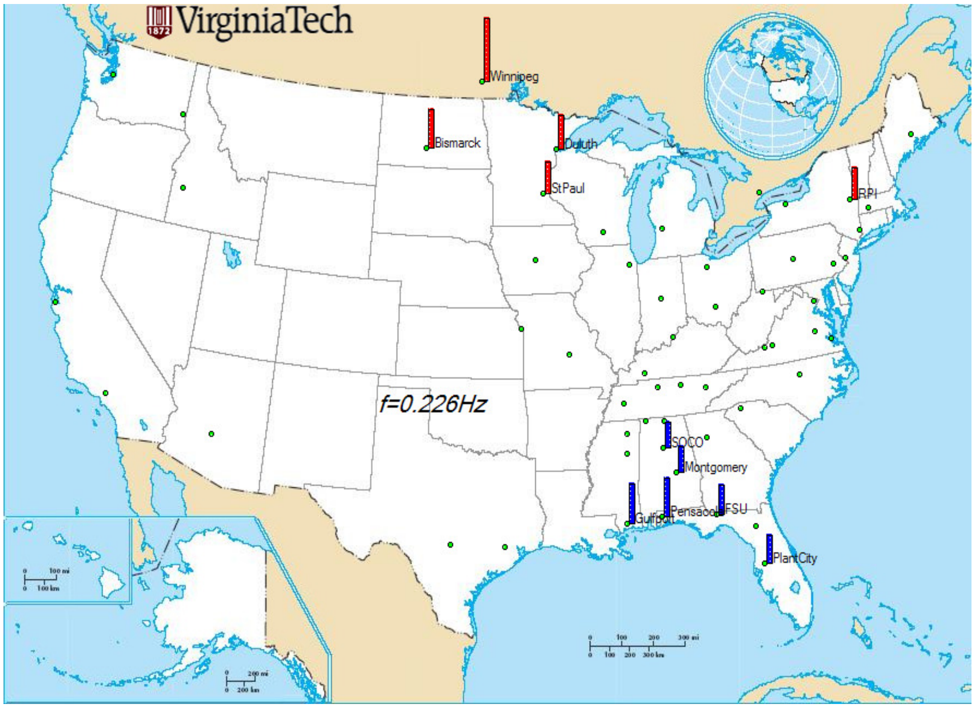


Fig. 7.8 Mode Shape Distribution Map

Chapter 8: Conclusions and Future Work

8.1 Conclusions

In this dissertation, several aspects of FNET are touched upon. First, this report briefly introduces the genesis and the architecture of the FNET system, and then elaborates upon its concrete implementations, including FDR hardware/code, FNET server data communication, FNET server database systems, and FNET website structure, which can be used as a detailed documentation for future reference. Next, FNET frequency estimation algorithm and phase angle estimation algorithm are discussed at length. Mathematical examination and simulation results concerning the frequency algorithm reveal the effect of harmonics and noises on and the dynamic performance of the algorithm. The solution of the residue problem, coupled with the sinusoidal error reducing method and the fixed offset balancing method are presented to exhibit the new methodologies to eliminate the computation errors of the angle algorithm.

After that, the dissertation discusses the algorithms developed at FNET to detect the frequency disturbance and estimate the disturbance location by the triangulation procedure using real-time frequency data and geographic topology of the FNET units in the power grid where the disturbance occurs. Techniques such as the adaptive threshold method and the oscillation-related method are put forward as complements to the location estimation algorithm. Lastly, the dissertation proceeds to look at the FNET angle-based power system oscillation detection and presents some research about the Matrix Pencil Modal Analysis of FNET phase angle oscillation data.

First and foremost, this dissertation touches on almost every aspect of FNET, hardware issues, software problems, algorithm research, server code enhancement, oscillation detection implementation, angle-based modal analysis, etc. It is virtually an exhaustive documentation of the FNET system and a complete manual of the research the author has conducted in his Ph.D program.

The solutions of many practical problems, such as memory leak and on-line trigger loop-out, have made the FNET system a very stable and reliable research platform. The newly-developed data forwarding feature offers enormous flexibility and mobility to FNET applications. The work regarding the frequency and angle algorithms depicts a clear picture of the mechanisms and performance of those algorithms. It is especially worth mentioning that the enhancements to the angle computation, including sawtooth error correction, quasi-positive-sequence DFT and fixed offset balancing, greatly improve the precision of measurement of FNET devices, which facilitate the usage of FNET measurements and brings the system to a more refined level.

This dissertation also presents an event location estimation technique together with two supplementary methods that has been proven to be more accurate and robust than the prototype triangulation module. It was shown from the triangulation accuracy statistics that said event location estimation algorithm is fairly effective in locating frequency disturbances.

This dissertation lays the groundwork for future angle-based oscillation analysis using FNET data. It has provided a method to detect power system oscillations and extract relative angle data from FDR measurements. It also has given us the first look at the behavior of the relative voltage angles during an oscillatory event in the EI. Matrix Pencil modal analysis with respect to relative angles broaches an interesting topic for future FNET members to embark upon. The techniques and results presented in this work could be used as a launch pad for future research.

8.2 Publications

- Tao Xia, Yilu Liu, 'Single-phase Phase Angle Measurements in Electric Power Systems', *IEEE Transactions on Power Systems, scheduled to publish in May 2010 issue*
- Tao Xia, Hengxu Zhang, Robert Gardner, *et al.*, 'Wide-area Frequency Based Event Location Estimation', *IEEE PES general meeting 2007*
- Tao Xia, Robert Gardner, Yilu Liu, 'FNET Observations on the Impact of Super Bowl XLII on the Power Grid Frequency', *IEEE PES general meeting 2009*
- Tao Xia, Yilu Liu, Robert.M.Orndorff, 'Frequency Monitoring Network (FNET) and its Aspects on Disturbance Triggering and Disturbance Location Estimation', *The Georgia Tech fault and disturbance analysis conference 2009*
- Jingyuan Dong, Tao Xia, Yilu Liu, *et al.*, 'Monitoring the North American Interconnections at Distribution Level', *IEEE PES general meeting 2009*
- Wei Li, Tao Xia, Yilu Liu, *et al.*, 'Wide Area Synchronized Measurements and Inter-Area Oscillation Study', *IEEE Power Systems Conference and Exposition, 2009*
- Jingyuan Dong, Tao Xia, Yilu Liu, *et al.*, 'Wide-Area Measurements of Three North America Interconnections at Distribution Level', *IEEE PES general meeting 2009*
- Jingyuan Dong, Tao Xia, James S. Thorp, *et al.*, 'Monitoring Power System Disturbances at the Distribution Level', *IEEE PES general meeting 2009*
- Qingsheng Zhao, Tao Xia, Yilu Liu, *et al.*, 'Detection of the Start of Frequency Excursions in Wide-Area Measurements', *IEEE PES general meeting 2008*
- Jing Ma, Tao Xia, Yilu Liu, *et al.*, 'Wave-Front Arrival Time Analysis Using Wide-Area

Frequency Measurements', *IEEE PES general meeting 2009*

8.3 Future Work

Potential future research may include:

- Research on the frequency algorithm to improve its accuracy especially when various noises in distribution level are present
- Precise determination of the event time, which could be conducive to enhancing the frequency-based event location estimation algorithm
- Further refine the angle-based oscillation detection algorithm
- Research on angle-based event location estimation algorithm
- Angle-based situation awareness development
- More extensive modal analysis of FNET phase angle data

Reference

[1] B. Qiu, L. Chen, V.A. Centeno, X. Dong, Y. Liu, “Internet Based Frequency Monitoring Network (FNET)”, IEEE Power Engineering Society Winter Meeting, 28 Jan.-1 Feb. 2001, Vol. 3, pp 1166 – 1171.

[2] Z. Zhong, C.C. Xu, B.J. Billian, L. Zhang, S.-J.S. Tsai, R.W. Conners, V.A. Centeno, A.G. Phadke,; Y. Liu, “Power system frequency monitoring network (FNET) implementation”, IEEE Transactions on Power Systems, vol. 20, no. 4, pp. 1914 – 1921, Nov. 2005.

[3] A. G. Phadke, “Synchronized Phasor Measurements in Power Systems”, IEEE Computer Applications in Power, Vol. 6, No. 2, April, 1993, pp. 10-15.

[4] T. Xia, H. Zhang, R.M. Gardner, J. Bank, J. Dong, J. Zuo, Y. Liu, L. Beard, P. Hirsch, G. Zhang, R. Dong, “Wide-area Frequency Based Event Location Estimation”, IEEE Power Engineering Society General Meeting, 24-28 June 2007.

[5] N. Jaleeli, D. N. Ewart, Understanding automatic generation control: IEEE Transactions on Power Systems, Vol. 7, No.3, Aug. 1992, pp1106-1122.

[6] J.R. Smith, D. A. Pierre, A supplementary adaptive VAR unit controller for power system dumpling.

[7] Miroslav M. Begovic, Petar M. Djuric, Sean Dunlap, Arun G. Phadke, Frequency Tracking of Power Networks in the Presence of Harmonics: IEEE Transactions on Power Delivery, Vol. 8, No.2, Apr. 1993, pp480-486.

[8] W. A. Mittelstadt, P. E. Krause, The DOE wide area measurement system (WAMS) project – Demonstration of dynamic information technology for the future power system, EPRI conference on the future of power delivery, April 1996.

[9] Report of a Panel Discussion, “Power System Disturbance Monitoring Utility Experiences”, IEEE Transactions on Power Systems, Vol. 3, No. 1, Feb. 1988, pp 134--147.

[10] J. F. Hauer, R. L. Cresap, “Measurement and Modeling of Pacific Inter-tie Response to Random Load Switching”, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-100, No. 1, Jan. 1981, pp 353 – 357.

[11] J. Chen, Accurate frequency estimation with phasor angles, Virginia Tech M.S. thesis, April, 1994.

[12] Gabriel Benmouyal, “An Adaptive Sampling-interval Generator for Digital Relaying”, IEEE Transactions on Power Delivery, Vol. 4, No. 3, July, 1989, pp1602-1609.

[13] P. Bonanomi, “Phase Angle Measurements with Synchronized Clocks - Principle and Applications”, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-100, No. 12 December 1981.

[14] A. G. Phadke, “Synchronized Phasor Measurements in Power Systems”, IEEE Computer

Applications in Power, Vol. 6, No. 2, April, 1993, pp. 10-15.

[15] R. O. Burnett Jr., M. M. Butts, B.J. Billian, T. W. Cease, V.A. Centeno, G. Michel, R. J. Murphy, A.G. Phadke, “Synchronized Phasor Measurements of a Power System Event”, IEEE Transactions on Power Systems. Vol. 9, No. 3, August 1994.

[16] Z. Zhong, C. Xu, B. J. Billian, L. Zhang, S. S. Tsai, R. W. Conners, V. A. Centeno, A. G. Phadke, and Y. Liu, “Power System Frequency Monitoring Network (FNET) Implementation,” IEEE Transactions on Power Systems, Vol. 20, No. 4, November 2005.

[17] IEEE Standard for Synchrophasors for Power System, C37.118-2005, prepared by the IEEE Power System Relaying Committee of the Power Engineering Society.

[18] D. Fan, A. Centeno, “Phasor-Based Synchronized Frequency Measurement in Power Systems”, IEEE Transactions on Power Delivery, Vol. 22, No. 4, October, 2007.

[19] J. Chen, “Accurate frequency estimation with phasor angles,” M.Sc. dissertation, Bradley Dept. Elect. Comput. Eng., Virginia Polytechnic Inst. State Univ., Blacksburg, VA, 1994.

[20] A. G. Phadke and J. S. Thorp, Computer Relaying for Power Systems. New York: Research Studies Press, 1988.

[21] J. S. Thorp, C. E. Seyler, and A. G. Phadke, "IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications," Electromechanical wave propagation in large electric power system, vol. 45, pp. 614 - 622, June 1998.

[22] M. Parashar, 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, vol. 51, pp. 1848 - 1858, Sept. 2004.

[23] A. J. Arana, J. N. Bank, R. M. Gardner, and Y. Liu, "Estimating Speed of Frequency Disturbance Propagation Through Transmission and Distribution Systems," presented at Power Systems Conference and Exposition, 2006. PSCE '06. 2006 IEEE PES Oct. 29 2006-Nov. 1 2006.

[24] D. Wang, X. Wang, and J. S. Thorp, "Study on electromechanical wave continuum model for power systems in mechanics," presented at Power Engineering Society General Meeting, 2006. IEEE, 18-22 June 2006.

[25] R. J. Murphy, "Power system disturbance monitoring," presented at Western Protective Relay Conference, Spokane,WA, Oct 17-20, 1994.

[26] L. Huang, M. Parashar, A. G. Phake, and J. S. Thorp, "Impact of electromechanical wave propagation on power-system reliability," presented at Proc. 39th CIGRE Conf., Paris, France, Aug. 25-30,2002.

[27] A. Semlyen, "Analysis of disturbance propagation in power systems based on a homogeneous dynamic model," IEEE Trans. on PAS, vol. PAS-93, pp. 676-684, Mar. 1974.

[28] "History of GPS," http://usinfo.state.gov/.

[29] Jingyuan Dong, Tao Xia, Yingchen Zhang, Lei Wang, Yilu Liu, Lisa Beard, Terry Bilke, “Wide-area measurements of three North America interconnections at distribution level”,

Power Systems Conference and Exposition, 15-18 March 2009

[30] North America Electric Reliability Council, “Understand and calculate frequency response”. Available at: http://www.nerc.com/~oc/ce/resources.html

[31] A. Bykhovsky and J. H. Chow, “Power system disturbance identification from recorded dynamic data at the Northfield substation,” International Journal Electrical Power & Energy Systems, vol. 25, pp. 787-795, 2003.

[32] N. Jaleeli, L. S. VanSlysk, “NERC’s new control performance standards,” IEEE Trans. Power Systems, vol. 14, pp. 1092-1099, 1999.

[33] Huang, L. and M. Parashar. Impact of Electromechanical Wave Propagation on Power System Protection and Control. in CIGRE conference. 2001.

[34] Parashar, M., J.S. Thorp, and C.E. Seyler, Continuum modeling of electromechanical dynamics in large-scale power systems. [Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on], 2004. 51(9): p.1848.

[35] Jaime Quintero, Guoping Liu and Vaithianathan “Mani” Venkatasubramanian; An Oscillation Monitoring System for Real-time Detection of Small-Signal Instability in Large Electric Power Systems.

[36] P. Kundur, Power System Stability and Control. McGraw-Hill, Inc., New York, 1994.

[37] M. Klein, G. J. Rogers, and P. Kundur, “A Fundamental Study of Inter-Area Oscillations in Power Systems,” IEEE Trans. Power Systems, vol. 6, pp. 914-921, Aug. 1991.

[38] U.S.–Canada Power System Outage Task Force, “Causes of the August 14th Blackout in the United States and Canada,” Interim Rep., Nov. 2003, ch. 6.

[39] C.W. Taylor, D.C.Erickson, K.E.Martin, R.E.Wilson, and V.Venkatasubramanian, “WACS-Wide-area stability and voltage control system: R&D and online demonstration”, Proceedings of the IEEE, Vol.93, No.5, pp. 892-906, May 2005.

[40] K. Tomsovic, D. Bakken, V. Venkatasubramanian, A. Bose,“Designing the Next Generation of Real-Time Control, Communication and Computations for Large Power Systems,” Proceedings of IEEE, vol. 93, no. 5, May 2005.

[41] D. N. Kosterev, C. W. Taylor, and W. A .Mittelstadt “Model validation for the August 10, 1996 WECC system outage”, IEEE Trans. Power Systems, Vol.14, No.3, pp.967-979, August 1999.

[42] V. Venkatasubramanian and Y. Li, “Analysis of 1996 western American electric blackouts”, Proc. Bulk Power System Phenomena-Stability and Control, Venice, Italy, 2004.

[43] J.Y. Cai; Zhenyu Huang; J. Hauer; K. Martin. Current Status and Experience of WAMS Implementation in North America. Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES. 2005 Page(s):1-7.

[44] G. R. B. Prony, “Essai experimental et analytique sur les lois de la dilatalrlite de fluids elastiques et sur cells de la vapeur de l’alcool, à différents tempoeratures,” Journal de l’Ecole

Polytechnique (París), vol. 1, pp. 24-76, 1795.

[45] D. J. Trudnowski, J. R. Smith, T. A. Short, and D. A. Pierre, “An application of Prony methods in PSS design for multi-machine systems,” IEEE Trans. Power Systems, vol. 6, pp. 118-126, Feb. 1991.

[46] M.L.Crow, and A.Singh, “The matrix pencil for power system modal extraction”, IEEE Trans. Power Systems, Vol. 20, Issue 1, Feb 2005, pp. 501-502.

[47] J.-M. Papy, L.D. Lathauwer, and S.V. Huffel, “Common pole estimation in multi-channel exponential data modeling”, Signal Processing, 86, 2006, pp. 846-858.

[48] P. S. Dolan, J. R. Smith, and W. A. Mittelstadt, “Prony analysis and modeling of a TCSC under modulation control,” in Proc. 4th IEEE Conference on Control Applications, 1995, Sept. 1995, pp. 239 – 245.

[49] M. Amono, M. Watanabe, M. Banjo, “Self-testing and selftuning of power system stabilizers using Prony analysis,” in Proc. IEEE Winter Power Meeting, 1999, pp. 655-660.

[50] S. W. Chen, “A two-stage discrimination of cardiac arrhythmias using a total least squares-based Prony modeling algorithm,” IEEE Trans. Biomedical Engineering, vol. 47, pp. 1317-1327, Oct. 2000.

[51] F. M. El-Hefnawi, “Use of Prony’s method for extracting the poles and zeros yielding a wideband window type response of circular antenna arrays,” in Proc. IEEE Radio and Wireless Conference, 1998, pp. 201-204.

[52] K. F. Sabett, L. P. B. Katehi, and K. Sarabandi, “Waveletbased CAD modeling of microstrip discontinuities using least square Prony's method,” in Proc. IEEE MTT-S International, 1997, pp. 1799-1802.

[53] M. Meunier, F. Brouaye, “Fourier transform, wavelets, Prony analysis: tools for harmonics and quality of power,” in Proc. 8th International Conf. Harmonics and Quality of Power, 1998, pp. 71-76.

[54] M. A. Andrade, A. R. Messina, C. A. Rivera, and D. Olguin, “Identification of Instantaneous Attributes of Torsional Shaft Signals using the Hilbert Transform,” IEEE Trans. Power Systems, to be published.

[55] D. Ruiz-Vega, A. R. Messina, and M. Pavella, “Online Assessment and Control of Transient Oscillations Camping,” IEEE Trans. Power Systems, vol. 19, pp. 1038-1047, May. 2004.

[56] J. Xiao, X. Xie, J. Han, and J. Wu, “Dynamic Tracking of Low-frequency Oscillations with Improved Prony Method in Wide-Area Measurement System,” presented at the IEEE Power Meeting, Denver, CO, June 6-10, 2004.

[57] R. Zivanovic, P. Schegner, “Pre-filtering improves prony analysis of disturbance records,” in Proc. IEE Int. Conf. Developments in Power System Protection, 2004, pp. 780-783.

[58] M. A. Johnson, I. P. Zarafonitis, M. Calligaris, “Prony analysis and power system stability-some recent theoretical and applications research,” in Proc. IEEE Power summer

Meeting, 2000, pp. 1918-1923.

[59] V K Jain and R D Gupta, “Identification of linear systems through a Grammian technique,” International Journal of Control, 12, 3, 1970, pp 421-431.

[60] V. K. Jain, “Filter analysis by grammian method,” IEEE Transactions on Audio Electrocoustics, AU-21, 1973, pp. 120-123.

[61] V. K. Jain, “Filter analysis by use of pencil-of-functions: Part I,” IEEE Transactions on Circuits and Systems, CAS-21, September 1974, pp. 574-579.

[62] V. K. Jain, “Filter analysis by use of pencil-of-functions: Part II,” IEEE Transactions on Circuits and Systems, CAS-2 1, September 1974, p 580-583.

[63] T. K. Sarkar, J. N. Nebat, D. D. Weiner, and V. K. Jain, “Suboptimal approximation/identification of transient waveforms from electromagnetic systems by pencil-of-function method,” IEEE Transactions on Antennas and Propagation, AP-37, 2, February 19, 1979, pp. 229-234.

[64] V. K. Jain, T. K. Sarkar, and D. D. Weiner, “Rational modeling by the pencil-of-functions method,” IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-31, June 1983, pp. 564-573.

[65] J. A. Mackay and A. McCowen, “An improved pencil-of-functions method and comparisons with traditional methods of pole extraction,” IEEE Transactions Antennas and Propagation, AP-35, April 1987, pp 435-441.

[66] V. K. Jain and T. K. Sarkar, “High performance signal modeling by pencil-of-function method,” IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-34, August 1986, pp 997- 1000.

[67] V. K. Jain and T. K. Sarkar, “Multivariable system identification by pencil-of-functions method,” IEEE Transactions on Instruments and Measurements, IM-34, December 1985, pp 550-557.

[68] Y. Hua, “On techniques for estimating parameters of exponentially damped/undamped sinusoids in noise,” PhD dissertation, Syracuse University, Syracuse, NY, August 1988.

[69] J. Quintero, A real-time wide-area control for mitigating small-signal instability in large electric power systems, Ph.D. dissertation, School of EECS, Washington State University, Pullman, WA, May 2005.

[70] Sarkar, T.K.; Pereira, O. Using the matrix pencil method to estimate the parameters of a sum of complex exponentials. Antennas and Propagation Magazine, IEEE. Volume 37, Issue 1, Feb. 1995 Page(s):48 – 55.

[71] D. J. Trudnowski. Making Prony Analysis More Accurate using Multiple Signals. IEEE Transactions on Power System: 1999,14.

[72] J. F. Hauer, C. J. Demeure, and L. L. Scharf, “Initial Results in Prony Analysis of Power System Response Signals,” IEEE Trans. Power Systems, vol. 5, pp. 80-89, Feb. 1990.

[73] Hua, Y.; Sarkar, T.K. Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise. Acoustics, Speech, and Signal Processing, IEEE Transactions on. Volume 38, Issue 5, May 1990 Page(s):814 – 824.

[74] Guoping Liu, Jaime Quintero, and Vaithianathan "Mani" Venkatasubramanian, Oscillation Monitoring System Based on Wide Area Synchrophasors in Power Systems, 2007 IREP Symposium- Bulk Power System Dynamics and Control - VII, Revitalizing Operational Reliability August 19-24, 2007, Charleston, SC, USA.

Appendix

Detailed code for sawtooth error correction in the angle computation algorithm

```
void interrupt capint()
{
  int i;
  int temp;

  if ((*EVAIFRC & 0x0004) == 0x0004)           //if cap3 int happen
  {
    //for(i = 0; i < 5; i++);  //empty other 3 channels data
    //temp = ADC_VAL;

      for(i = 0; i < 5; i++);
      temp = ADC_VAL;             //read out the sample data
    temp = temp>>2 ;            //right aglian

    Sample_Data[Sam_Data_Ptr] = temp;
      Sam_Data_Ptr++;

      temp=(int)(Sam_Data_Ptr*Overhead+0.5);//for counter compensation
      if(temp>Cmpst_Ptr)
      {
        *T1PER = 30000000/NUM_PER_SEC+2;
        Cmpst_Ptr=temp;
      }
      else *T1PER = 30000000/NUM_PER_SEC+1;//for counter compensation
      if(Sam_Data_Ptr == NUM_PER_SEC)
      {
        Sam_Data_Ptr = 0;
        Cmpst_Ptr = 0;
        *T1CON = (*T1CON) & 0xFFBF;
        *T1CNT = *T1CMP - 0x04;
    }

    for(i = 0; i < 5; i++);  //empty other 3 channels data
    temp = ADC_VAL;
    for(i = 0; i < 5; i++);  //empty other 3 channels data
    temp = ADC_VAL;
    for(i = 0; i < 5; i++);  //empty other 3 channels data
    temp = ADC_VAL;

    *EVAIFRC |= 0x0004;       // Resets flag
  }

  if((*EVAIFRC & 0x0001) == 0x0001)
  {
    *T1CON = (*T1CON) | 0x0040;
    *EVAIFRC |= 0x0001;
  }

  enable();
  return;
}
```