8-2016

Data Analytics and Application Developments Based on Synchrophasor Measurements

Jiahui Guo
University of Tennessee, Knoxville, jguo7@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Jiahui Guo entitled "Data Analytics and Application Developments Based on Synchrophasor Measurements." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

Yilu Liu, Major Professor

We have read this dissertation and recommend its acceptance:

Fangxing Li, Qing Cao, Chien-fei Chen

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Data Analytics and Application Developments Based on Synchrophasor Measurements

A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Jiahui Guo
August 2016

Acknowledgements

First of all, I would like to express my sincere appreciation to my advisor Dr. Yilu Liu for continuous support, patient guidance and constant inspiration for both of my academic research and personal life.

Next, I would like to thank all the rest of the committees: Dr. Fangxing Li, Dr. Qing Cao and Dr. Chien-fei Chen. I greatly appreciate their comments and suggestions. My sincere thanks also goes to all my colleges: Dr. Yanzhu Ye, Dr. Ye Zhang, Dr. Yin Lei, Dr. Lingwei Zhan, Dr. Jidong Chai, Dr. Gefei Kou, Dr. Wenpeng Yu, Dr. Zhiyong Yuan, Dr. Yong Liu, Siqi Wang, Shutang You, Dao Zhou, Hesen Liu, Ling Wu, Wenxuan Yao, Jicheng Zhao, and Micah Till. Also, I would like to thank Dr. Matthew Gardner, Dr. Tao Xia, Dr. Rui Sun in Dominion Virginia Power for their guidance. I also like to thank two special staff members in the EECS department, Ms. Dana Bryson and Mr. Markus Iturriaga for their generous help in many ways.

Last but not least, I would like to express my deepest appreciation to my family for their love and support.

Abstract

Frequency Monitoring Network (FNET) is an Internet-based, wide-area phasor measurement system that collects power system data using Frequency Disturbance Recorders (FDRs) that are installed at the distribution level. These synchrophasor measurements enable the monitoring of bulk power systems, and provides wide-area situational awareness and disturbance analysis for understanding power system disturbances and system operations. Various data analytics and applications are built based on these valuable measurements.

Real-time situational awareness tools are of critical importance to power system operators. Knowledge of the scope and extent of facilities impacted, as well as the duration of their dependence on backup power, enables emergency response officials to plan for contingencies and provide a better overall response. Based on the measurement data acquired by the FDRs deployed in the North American power grids, an off-grid detection method is proposed and implemented, which monitors the critical electrical loads and detects the transition of these loads from an on-grid operation to an off-grid operation. And two visualization tools are developed to display the real-time system operation condition in an intuitive manor.

Moreover, electromechanical oscillation is an inherent property of power systems which cannot be fully eliminated, and poorly damped low-frequency oscillations would hazard the system stability, reliable real-time mode estimation of low frequency oscillation and visualization is necessary, such that appropriate actions can be taken to maintain the stability of the power system and furthermore to provide guidelines to tune the parameters in PSS. With the aid of synchrophasor technology, pure measurement-based approach is proposed to estimate the oscillation frequency and damping regardless of ring-down or ambient condition in real-time environment.

In large interconnected power systems, changes in load, generation, topology, and control may initiate disturbances, e.g. generation trip, line trip, load shedding, oscillation, etc., that could be dangerous for system security and reliability. During these disturbances, frequency, phase angle and voltage magnitude will change simultaneously correspondingly. Some of these recorded events are analyzed for to identify relative characteristic to understand the dynamic behaviors from a wide area measurement perspective.

Table of Contents

List of Tables

List of Figures

Chapter 1. Overview of FNET System

1.1 Frequency Monitoring Network (FNET/GridEye)

Frequency Monitoring Network (FNET/GridEye) is an internet-based, wide-area phasor measurement system that collects power system data utilizing low-cost, high-accuracy Frequency Disturbance Recorder (FDR), which is a single phase phasor measurement unit (PMU). Since 2004, FNET/GridEye has been providing researchers with valuable synchrophasor data collected by FDRs in North American and worldwide power grids. With the precise timing information provided by the Global Positioning System (GPS), FDR is capable of measuring voltage phase angle, magnitude, and frequency from ordinary 120V electrical outlets accurately at 100ms intervals. These measurements are then timestamped and transmitted to a Phasor Data Concentrator (PDC), where they are synchronized, analyzed, and archived. More detailed information about FNET/GridEye can be found in [1]–[3]. Currently, the FNET/GridEye system consists of around 200 FDRs installed in the three North American interconnections: Eastern Interconnection (EI), Western Electricity Coordinating Council (WECC), and Electric Reliability Council of Texas (ERCOT) in United States. A variety of applications based on this collected data have been developed, such as real-time event and oscillation detection [4]–[10], event location estimation [1], [11], animated event visualization [12], [13], voltage stability monitoring [14]–[16], model reduction [17], forensic authentication of digital evidence [18], [19] and statistical analysis [20]–[25].

Figure 1.1 shows the locations of all the FDRs deployed in North America and Figure 1.2 shows the world wide deployment.

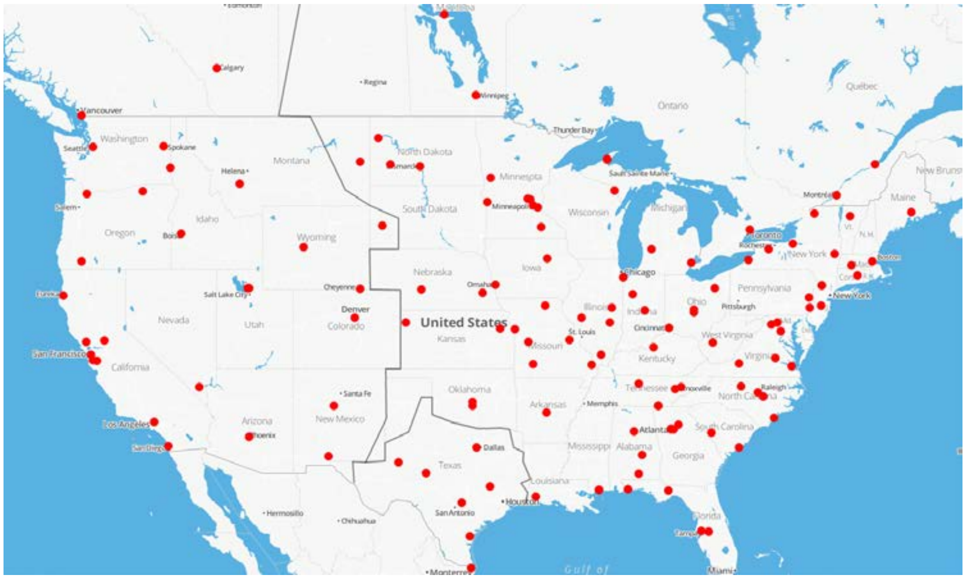


Figure 1.1 Map of FDR locations in North America

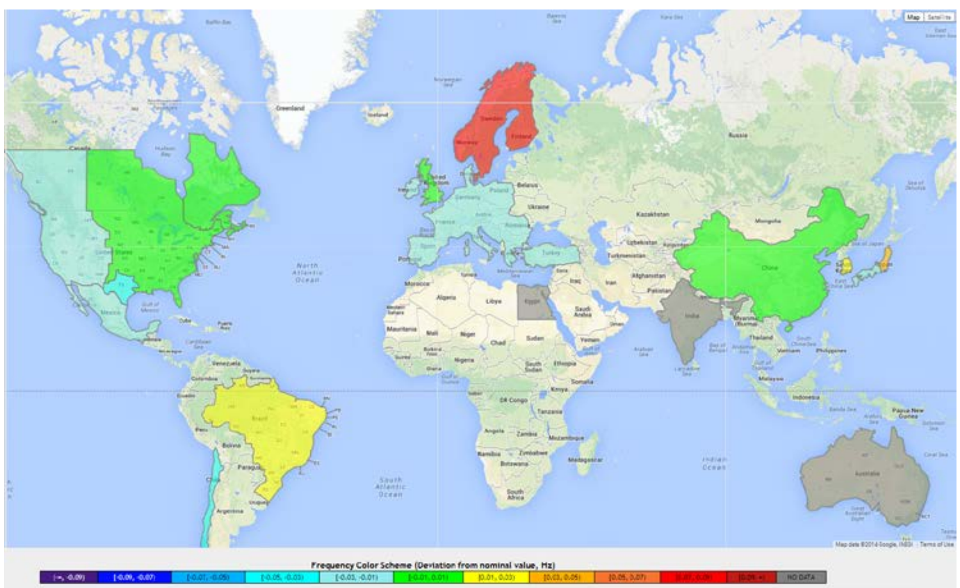


Figure 1.2 World map of FDRs

1.2 Frequency Disturbance Recorders

The first generation of FDRs was built in 2003, and FDRs are now in its second generation. The FDR framework is shown in Figure 1.3. An FDR consists of a current transducer, a low pass filter, an analog to digital converter (ADC), a digital signal processor (DSP), a microcontroller unit (MCU), a GPS receiver, and a network module.

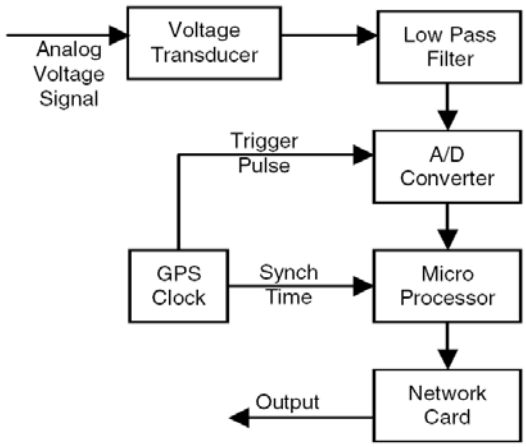


Figure 1.3 Framework of Frequency Disturbance Recorders

The current transducer converts the input voltage signal to current signal, and then converts the current signal to low voltage signal by an operational amplifier. The low pass filter has 200 Hz cutoff frequency, and filters the high frequency noise and harmonics. The ADC is started by the pulse per second (PPS) signal from GPS receiver once at the beginning of every second, then ADC samples and converts the analog signal into digital signal. DSP is used to compute the magnitude, angle and frequency of the input signal, and the measurements are time stamped by GPS. The measurement data are transmitted to MCU and sent out to FNET servers through the network module by TCP/IP protocol. In the hardware design, a GPS receiver is integrated into FDRs to

provide PPS to control the conversion of ADC and timestamp the measurement data. Generally the PPS accuracy of the GPS receiver is less than 500 ns, which will cause less than 0.0108 degree angle error for 60Hz power grid, and 0.009 degree angle error for 50Hz power grid. The DSP of the FDR is the core of the device, which is used to compute the magnitude, angle, and frequency of the inputs signals at distribution level. In order to easily develop and maintain the firmware, the C language is adopted in FDRs, and the code can be easily transplanted to the DSPs of later generation FDRs. Discrete Fourier Transform (DFT) is one of the common algorithms used in synchrophasor measurement area, and it is adopted as the basic framework of the FDR algorithm. The angle and frequency error of FDRs is less than 0.02 degree and 0.0005Hz respectively under ideal 60Hz signal condition [26], [27].

1.3 FNET/GridEye Framework

Figure 1.4 demonstrates the FNET system framework. Phasor measurements from the North American power grids are collected by widely installed FDRs, and are transmitted via the Internet to the FNET data center for processing and long-term data storage. The data center plays a critical role in the entire FNET system framework since the FDR measurement data are concentrated and processed here. Physically, the FNET data center is composed of several dedicated servers including a data server, application server, web server, and backup server. Functionally, the data center can be treated as a multi-layer data management system as shown in Figure 1.5. The top layer is the FNET data concentrator, which can be seen as the intermediate interface between the FDRs and other layers in the FNET system.

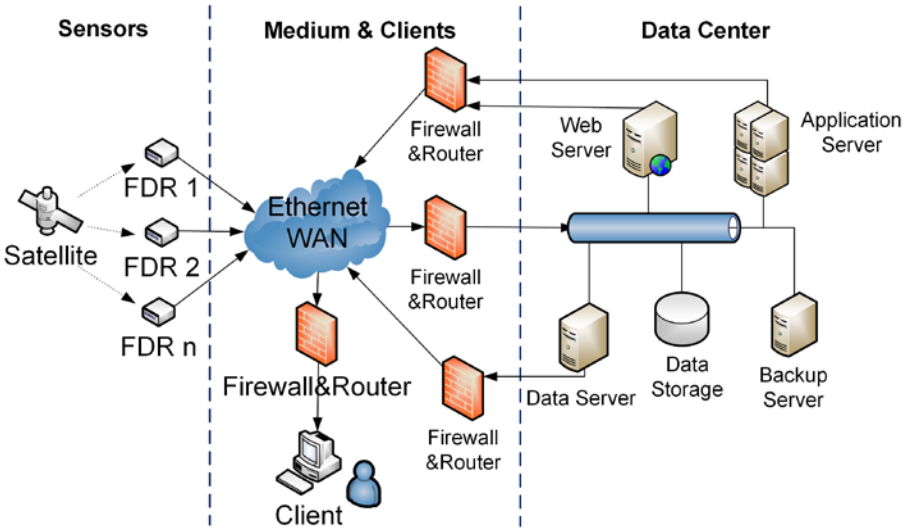


Figure 1.4 The Frequency Monitoring Network (FNET) Framework

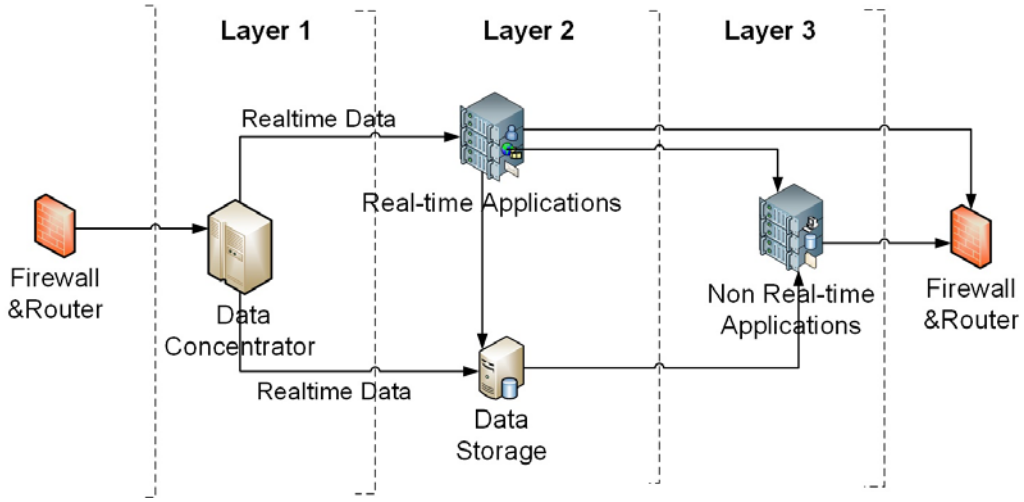


Figure 1.5 Hierarchical Structure of FNET data center

In the data concentrator, FDR measurements are extracted from the network data package, interpreted, error-checked, time-aligned, and then streamed into different layers for real-time applications and data storage. The second layer of the data center hierarchy is composed of the real-time application agent and the data storage agent. The real-time application agent is composed of various real-time applications, e.g. the frequency monitoring interface, event detection, and oscillation detection, etc. The function of the data storage agent is to simply receive and archive the FDR records for later use.

The third layer is the non-real-time application agent. Applications implemented in this layer operate on previously stored data rather than real-time streaming data. Examples of applications in this layer include event visualization, event analysis, oscillation modal analysis, and several different kinds of web services.

1.4 Organization of Study

An islanding / Off-grid detection system, as a crucial WAMS situational awareness tool, is proposed in Chapter 2, the detailed algorithm is illustrated, lab test cases and real test cases are elaborated to verify the effectiveness of the proposed algorithm. Chapter 3 focuses on another important WAMS application, which utilizes synchrophasor measurements to estimate real time low frequency and damping ration to help indicate the stability of the power grid. Chapter 4 addresses the real world power system events observed by synchrophasor measurement in North America and world-wide countries. Chapter 5 summaries the enterprise data historian project using OSIsoft PI system, participated in Dominion Virginia Power Company. Chapter 6 spotlights an enhancement for FNET server, which adds a data base for metadata and configuration management,

which heavily reduces the maintenance labor. Chapter 7 proposes a data driven statistical model to predict the solar power output at day ahead interval. Chapter 8 introduces a big data analytics platform to enable the capability to process over Terabyte historical FNET data to investigate deeper value.

Chapter 2. Islanding / Off-grid Detection

2.1 Motivation

Real-time situational awareness tools are of critical importance to power system operators, especially during emergencies. The availability of electric power has become a linchpin of most post-disaster response efforts, because public and private sector services depend upon it. Knowledge of the scope and extent of facilities impacted, as well as the duration of their dependence on backup power, enables emergency response officials to plan for contingencies and provide a better overall response. Based on the measurement data acquired by the Frequency Disturbance Recorders (FDRs) deployed in the North American power grids, an off-grid detection method is proposed and implemented. This method monitors the critical electrical loads and detects the transition of these loads from an on-grid operation to an off-grid operation, during which the loads are fed by a backup generation system or microgrid [28]. The details of the detection algorithm are presented, and some case studies and off-grid detection scenarios are also provided to verify the effectiveness and robustness. This paper also presents the real-time implementation of this method and several effectively detected off-grid situations. Moreover, two visualization tools are developed to display the real-time system operation condition in an intuitive manor. Based on the wide deployment of the FDRs, the FNET/GridEye system is implemented to capture power system dynamics and perform situational awareness applications. Utilizing the time-synchronized phasor and frequency measurements, power system events, such as generator trips, line trips, load shedding, off-grid operations, and oscillations, can be detected in real-time. Among these power system events, an off-grid

operation is a situation during which a part of the grid or a number of individual loads becomes electrically isolated from the remainder of the power system. Generally, the isolated loads are powered by UPS or backup generation. The off-grid operation can be initiated by a natural disaster. An example of this is when the New Orleans metropolitan area and an area between New Orleans and Baton Rouge became isolated from the rest of the power grid during Hurricane Gustav [29]. Some off-grid operations may be intentional, e.g. a microgrid which is designed to remain online when separated from the electric grid. Some other off-grid operations may be unintentional, e.g. a hospital or residence switches to dedicated UPS or emergency backup generator during blackouts. The duration of the off-grid condition and the nature of the back-up power sources are key indicators for situational awareness. Although it is easy to visually distinguish whether the system is operated on grid or off grid by eyeballing the frequency or angle trajectories, computer-aided situational awareness systems must have certain algorithms embedded to enable real-time detection, alarming, and event analysis. Off-grid operation manifests the same phenomenon as islanding operation in terms of frequency and voltage angles. The only difference is that islanding indicates that operation is maintained as a small grid, while an individual residence or hospital using a backup generator is not a complete grid. Even given this distinction, it is possible to use the same detection methods in both cases.

2.2 Literature Review

Islanding detection methods commonly used over the past several decades can be divided into two main categories: local methods and remote methods [30], [31]. The local

method can be further classified as passive methods, active methods, and hybrid methods. In the passive method category, over/under voltage, over/under frequency, rate of change of frequency (ROCOF) and voltage (ROCOV), phase and voltage harmonic monitoring methods are commonly used [32], [33]. Active methods contain active frequency drift [34], phase drift [35], and impedance detection methods. Hybrid methods have the advantages of both passive and active methods. In [36] the author combined the average rate of voltage change and real power shift to achieve a smaller Non-Detection Zone (NDZ) without perturbing the system and in [37] the author proposed a method using proportional power spectral density. In the remote methods, the power-line-signaling-based scheme [38] and Supervisory Control and Data Acquisition (SCADA) or Phasor Measurement Units (PMUs) depends methods are proposed in [39], [40]. Intelligent-based approaches are also investigated. An intelligent decision tree algorithm is proposed in [40], and data-mining technology is used in [41], [42]. Among these methods, remote detection methods have the advantage of theoretically no NDZ and increased reliability, but are expensive to implement and require utility involvement [30]. This work is a successor of previous research [43] and describes an algorithm using the combination of the frequency difference (FD) and integration of frequency deviation (IOFD) algorithm. This algorithm is implemented in the FNET/GridEye system as a low cost and highly reliable off-grid operation detection tool.

2.3 Proposed Algorithm

Power system frequency and voltage angles are the primary measurements used to determine the status of a power grid. Under normal operating conditions, multiple

frequency measurements across a single interconnection should be identical. However, after a local blackout, UPS or backup generators are not electrically synchronized with the interconnection. This results in a sustained frequency deviation between the isolated system and the bulk power system. If the loads connected with an FDR are operated on UPS or backup generators, the deviation between the FDR's reporting frequency and the frequency of the interconnection can be relatively large. This situation persists until the isolated system returns to synchronism with the grid.

This work proposes an algorithm using the combination of frequency difference (FD) and integration of frequency deviation (IOFD). At each timestamp t, the frequency deviation for the *i-th* FDR is given by

$$\underset{0<i\le N}{FD_i}(\mathrm{t}) = \left| f_i(t) - f_{ref}(\mathrm{t}) \right| \tag{2-1}$$

where N is the number of monitored FDRs, $f_i(t)$ is the measured frequency value of the *i-th* FDR at timestamp t and $f_{ref}(t)$ is the reference frequency, which is defined as the median value of all the monitored FDRs in the same interconnection. The equation for $f_{ref}(t)$ is given by

$$f_{ref}(t) = \mathrm{median}(f_1(t), f_2(t), ..., f_i(t), ..., f_N(t)) \tag{2-2}$$

The integration of frequency deviation is defined as the accumulation of the FD over a certain time period, which is given by

$$\underset{t_1\le t\le t_2}{IOFD_i} = \sum_{t=t_1}^{t_2} FD_i(\mathrm{t}) \tag{2-3}$$

where t_1 and t_2 are the start and end times, respectively, for this integration time period.

With these parameters introduced, the detection algorithm can be expressed in 3 phases.

1) *Phase I: Trigger early warning*

The $FD_i(t)$ is continuously calculated with (2-1). Once the frequency deviation of any FDR surpasses the first threshold as shown in (4), a trigger early warning is initiated and the detection procedure goes to Phase II.

$$FD_i(t) \geq F_{th1} \tag{2-4}$$

2) *Phase II: Trigger confirmation*

Once the detection procedure enters Phase II, the $IOFD_i$(s) for the early warned FDR(s) will be calculated over time period t_1 to t_2. If the *IOFD* value for any FDR is above a certain threshold, given by (2-5), it can be determined that the system monitored by the *i-th* FDR is in off-grid operation and the detection procedure goes to Phase III. Otherwise the procedure returns to Phase I.

$$\underset{t_1 \leq t \leq t_2}{IOFD_i} > F_{th2} \tag{2-5}$$

3) *Phase III: Return to normal indication*

After an off-grid trigger is confirmed, the $IOFD_i$ is continuously calculated, until it falls below the third threshold, as given in (6), which indicates that the system has rejoined the power grid, then the detection procedure goes back to Phase I.

$$\underset{t_3 \leq t \leq t_4}{IOFD_i} < F_{th3} \tag{2-6}$$

The flow chart of this detection method is shown in Figure 2.1.

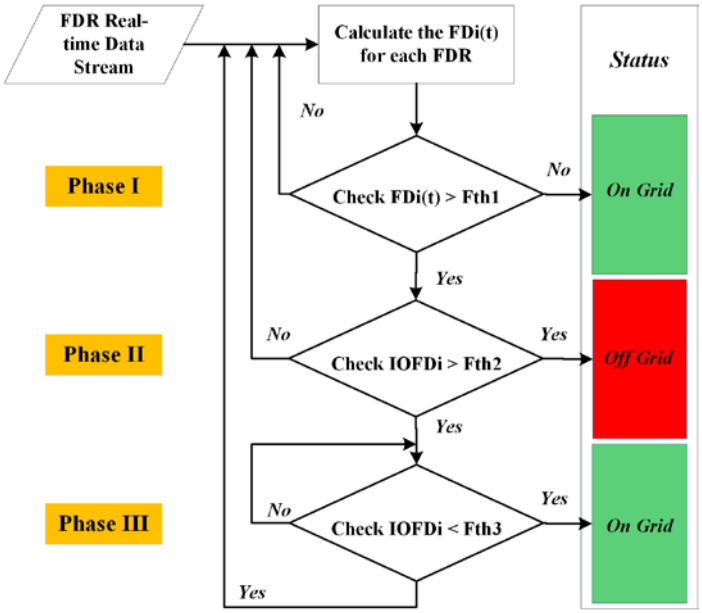


Figure 2.1 Flow chart of the off-grid operation detection module.

2.4 Offline Experiments

2.4.1 UPS Detection Experiments

Case 1: UPS

In this experiment, 5 FDRs are used with a commercially available UPS rated at 1000 VA and 120 V. In order to obtain real-time system frequency data, four FDRs (Unit812, Unit851, Unit865, and Unit873) were directly connected to the power grid and the other FDR (Unit874) was connected to the power grid through the UPS. In each situation, if the UPS is switched to “ON”, this FDR will measure the frequency of the UPS, while if the UPS is turned to “OFF”, the FDR will measure the frequency of the power grid. For this experiment, the UPS was switched to “ON” for approximately 1.5

minutes and then switched back to “OFF.” By doing this, the FDR connected to the UPS acquired the data needed to test the UPS detection algorithm. The real-time data were sent to the FNET/GridEye server and stored in a MySQL database and Microsoft Access files for offline analysis. Figure 2.2 shows the frequency plot over a 3 minute time window. Through several similar experiments, the thresholds are determined as F_{th1} = 15 millihertz (mHz), F_{th2} = 800 mHz·s and F_{th3} = 300 mHz·s, respectively.

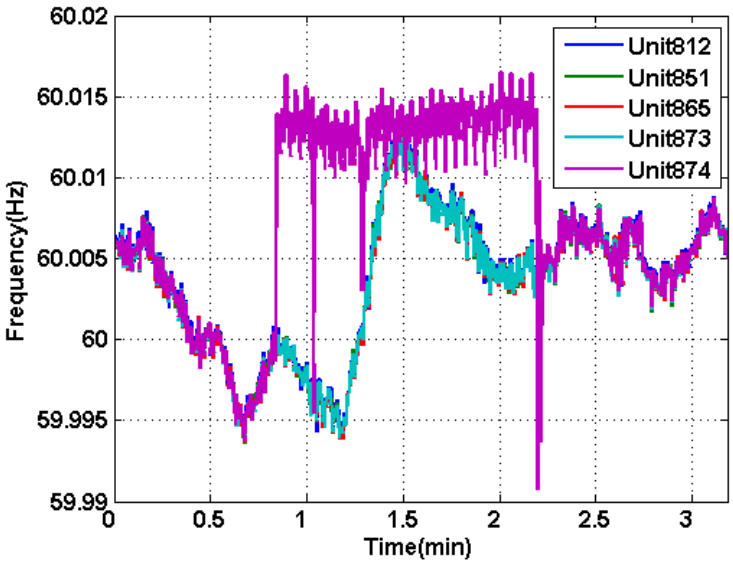


Figure 2.2 UPS case

2.4.2 Validation of Detection Algorithm Robustness

The detection algorithm for off-grid operation should not be falsely triggered during other power system events. The following cases are selected from FNET/GridEye historically detected events and are used to verify the robustness of the detection algorithm to ensure that the algorithm does not give a false positive during other power system events.

Case 2: Generation Trip

Figure 2.3 shows a generation trip detected by FNET/GridEye on November 14, 2013 at 17:18 (UTC) with an estimated 1200MW of generation tripped offline. For this event, the frequency deviation detected was 20.20 mHz, which surpassed the first threshold while the integration of the deviated frequency was 597.35 mHz·s, which was below the second threshold. Accordingly, the detection process ended without triggering.

Case 3: Line Trip

Figure 2.4 shows a line trip case detected on September 24, 2013, at 11:55:15 (UTC), which is representative of the typical line trip scenario. The frequency deviation detected in this case was 74 mHz, and the integration of the frequency deviation was 314.02 mHz·s. Thus, it was not triggered.

Case 4: Load Shedding

Figure 2.5 shows a load shedding case detected on November 24, 2013 at 09:45:29 (UTC) near Fresno, CA. The frequency deviation detected in this case was 8.45 mHz and since even the first threshold was not reached, the proposed algorithm did not trigger in this case.

Case 5: Oscillation

Figure 2.6 shows an oscillation case caused by an incident on the 500kV network detected on March 14, 2013 at 11:15:33 (UTC). The frequency deviation detected in this case was 21.1 mHz, the integration of the frequency deviation was 571.58 mHz·s. Thus, the proposed algorithm did not trigger in this case.

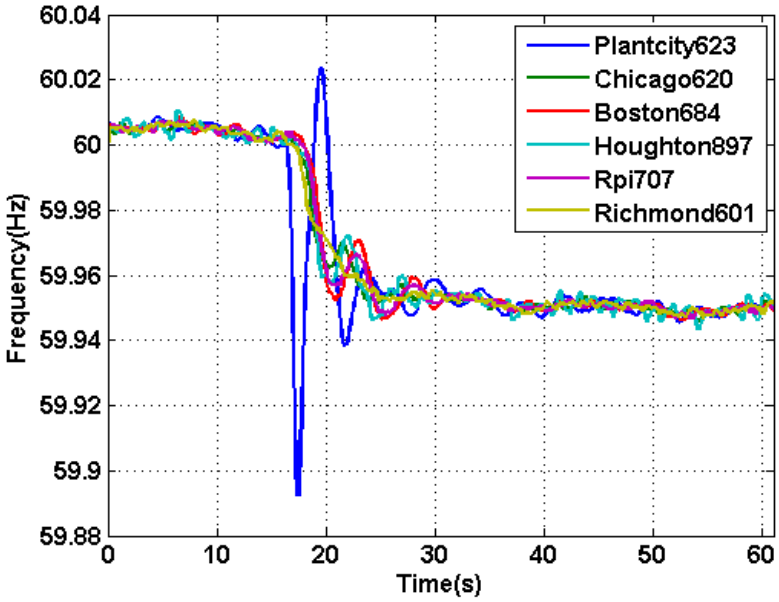


Figure 2.3 Generation trip case

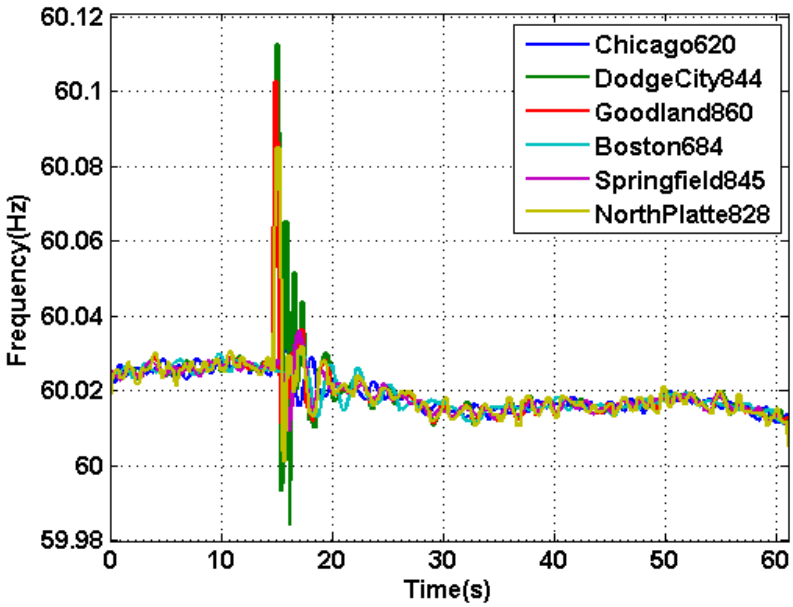


Figure 2.4 Line trip case

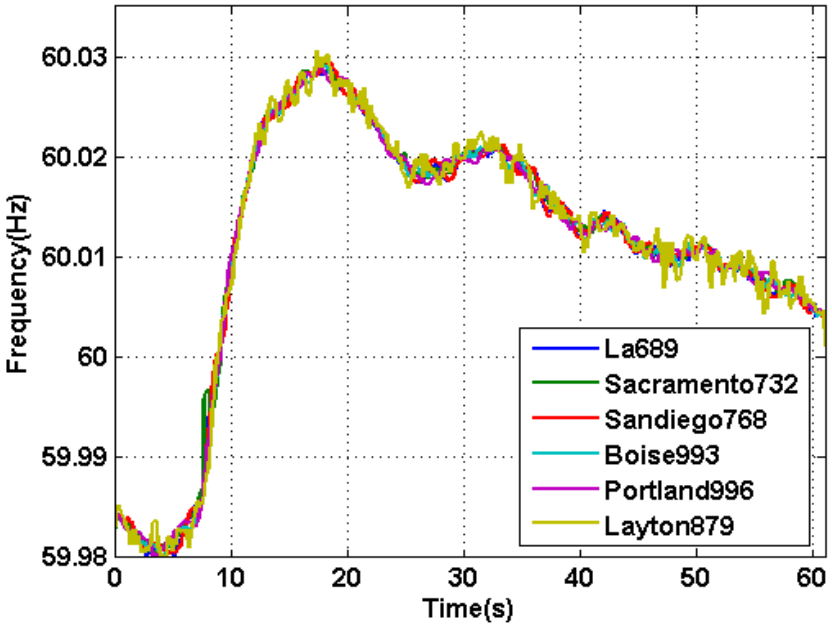


Figure 2.5 Load shedding case

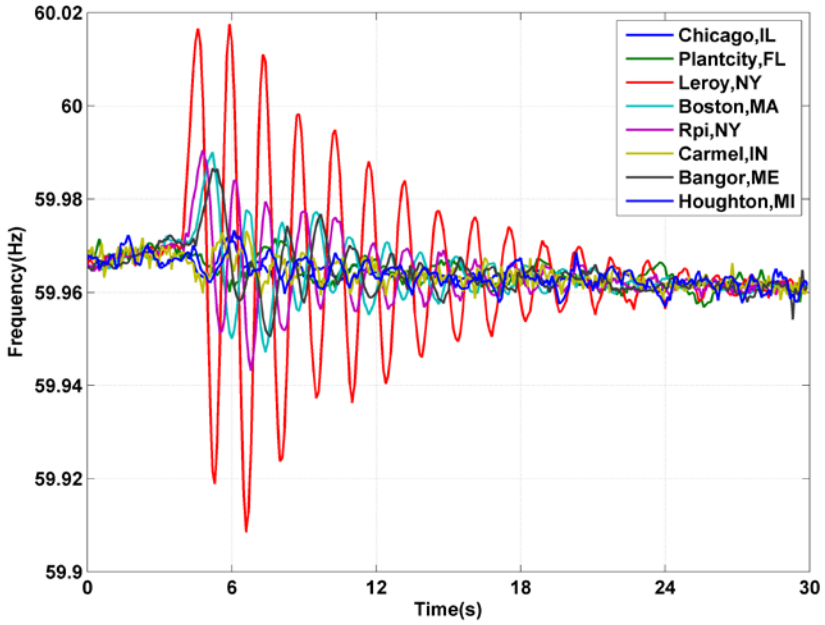


Figure 2.6 Oscillation case

The results of these offline case studies based on historical data indicate that the detection algorithm is robust enough to be practically applied for off-grid operation detection in the FNET/GridEye situational awareness system.

2.5 Real-time Application Implementation

2.5.1 The Overview of Real-Time Implementation Infrastructure

The off-grid operation detection algorithm is implemented in the FNET/GridEye system for real-time detection. Figure 2.7 shows the infrastructure of the complete real-time detection.

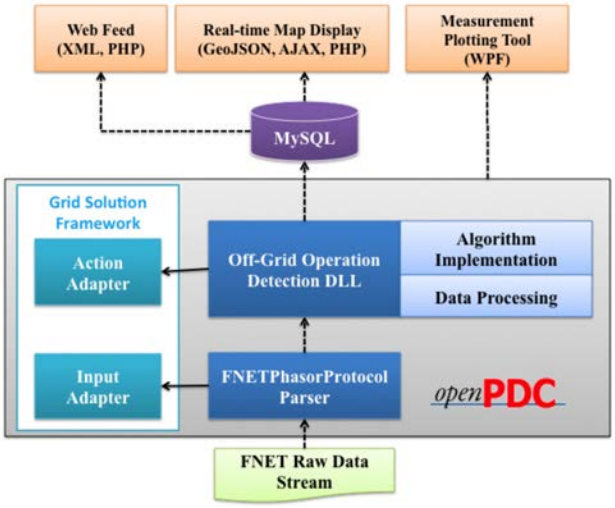


Figure 2.7 Real-time detection tool implementation structure

The real-time off-grid operation detection algorithm is implemented in OpenPDC [44] by customizing the libraries of the GSF [45]. The FNETPhasorProtocol Parser interprets the raw data stream from the FDRs using the FNET protocol. Then, the data is handed over to the off-grid operation detection program compiled in the form of the

dynamic-link library (DLL). The real-time detection program filters and sorts measurements by time, and applies the detection algorithm on a synchronized set. The detection algorithm runs every 0.1 seconds to identify the status of each FDR. The status is classified as 3 types: NODATA, ON GRID, and ON BACKUP. It then stores the detection results in the MySQL database.

A web service integrates the detection results with the geographic information of the devices to generate an XML feed for the Department of Energy (DOE) for off-grid operation reporting. Two other tools are also developed to visualize the off-grid operation status and the corresponding system condition in an intuitive manor.

2.5.2 Multi-threaded Implementation

Compared to post-event analysis, the real-time detection program requires significantly higher computational speed. In order to perform real-time detection on hundreds of high sampling rate measurement devices, the program should be implemented using multi-threaded computing to take advantage of powerful multi-core processors. The challenge associated with multi-threaded computing is guaranteeing the thread safety of the program. Thread safety is a computer programming concept applicable specifically in the context of multi-threaded programs. A piece of code is thread-safe if it only manipulates shared data structures in a manner that guarantees safe execution by multiple threads at the same time. Otherwise, two or more threads could access the same resource simultaneously without awareness of each other's actions. The result of such a scenario is unpredictable data corruption. The program uses thread-safe data structures in any possible situation. For situations that require non thread-safe data

structures, the program uses the lock mechanism provided by C# to safely perform multi-threading.

2.5.3 Real-time Data Processing

The most important issue for any real-time application is data processing. Various real-time data conditions need to be properly handled to guarantee the accuracy and stability of real-time applications [24].

1) Network latency

Since internet speed varies with physical locations, the measurements of multiple FDRs sampled at the same timestamp may arrive at the data concentrator server at different times. Slower applications (such as visualization) can tolerate higher data latency and a data archive can tolerate the highest latencies. Real-time applications typically require very low data latency. If the data from a certain device arrives later than “real-time”, the data may be considered too stale to contribute to the real-time applications decision making process. The definition of “real-time” differs depending on the types of applications and the requirements of organizations. Three parameters are most important to customize the latency of a real-time application: LeadTime, LagTime, UseLocalClock-AsRealTime. UseLocalClockAsRealTime indicates whether to use the local clocks as real-time or to instead use the timestamp of the latest received measurements. This should only be set to true if the local system clock time is derived by GPS or otherwise very accurately synchronized to real-time. The accuracy of the local clock time relative to GPS-time determines the needed value for the LeadTime setting. LeadTime defines the maximum time that a real-time application will tolerate for

measurements that arrive with a future timestamp as compared to "real-time" - a relative term based on the value of UseLocalClockAsRealTime. The value must be greater than zero, but it can be less than one for sub-second tolerances. If the LeadTime is set too short (relative to the accuracy of the local clock), measurements may be unnecessarily discarded. However, if the local clock is very accurate, and accordingly UseLocalClockAsRealTime is set true, this number should be very small, e.g., 0.1. LagTime defines the maximum time that a real-time application will wait for measurements to arrive after the “real-time”. If the measurements arrive later than LagTime, they will be discarded from the computation process of the application. The value must be greater than zero, but it can be less than one for sub-second tolerances.

The measurements arrive within the tolerance of both LeadTime and LagTime are considered to be “real-time” measurements and will be further processed in the off-grid operation detection algorithm.

2) Time alignment of measurement data

The off-grid operation detection algorithm needs to obtain the median frequency of all the frequency measurements at each timestamp. Therefore, it is necessary for the program to align all the real-time measurements based on their sampling timestamps. All the measurements with the same timestamp will be further sorted and the median value will be identified.

3) Missing data handling

Due to hardware failure or internet package loss, some devices may stop sending data for a period of time or introduce latency too long to be considered as “real-time”.

The program monitors the data stream of each device to check two typical conditions: 1) if the data from a specific device has not been coming in for a certain period, 2) if a relatively large number of measurements received from the device are too old to contribute to the real-time triggers decision making process. If any of those conditions is spotted, affected devices will be dismissed from the monitoring list temporarily. Each is reinstated when their data streams are again stable.

4) Bad data filtering

Some devices with hardware issues may send out problematic data. This implies that the detection program can still receive continuous data streams but the data does not reflect real system conditions. Figure 2.8 shows an example of such situation. This kind of data needs to be detected and excluded to avoid misleading the detection program and triggering false-alarms [46]. Thus, the detection program checks abnormal data conditions and temporarily dismisses any problematic devices from the monitoring list.

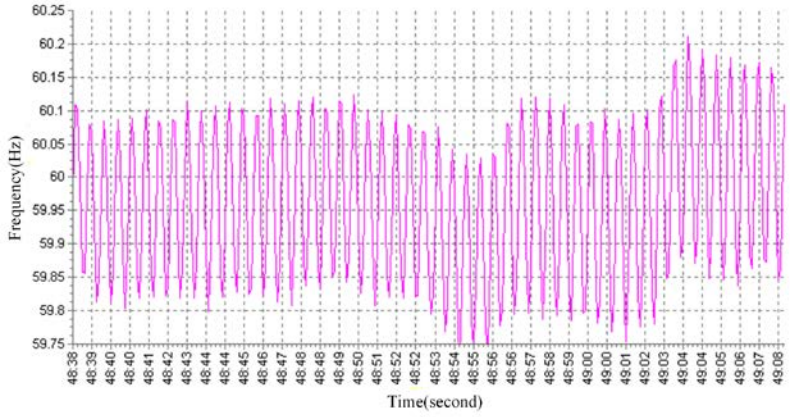


Figure 2.8 Bad data stream received at PDC

2.5.4 Algorithm Implementation

After data processing, the detection program analyzes the time-aligned measurements to identify any off-grid operation behavior using the proposed method. Instead of coding the threshold values in the detection program source code, it is more maintainable and flexible to define the thresholds as configurable parameters. This enables to easily customize thresholds for different power grids using the openPDC interface. All the parameters of the detection program are stored as one connection string in the MySQL database of openPDC. Table 2.1 lists several key configuration parameters for the off-grid operation detection program with their description.

Table 2.1 Key parameters description

Parameters	***Description***
Check Frequency	Number of checks per second to identify the device status
PreIslanding Threshold(mHz)	Threshold to compare with frequency difference from the median value, to detect suspicious off-grid operation
Consecutive Detections	If the frequency differences of one device have been over PreIslandingThreshold consecutively, start the integration to confirm the off-grid operation status
Islanding Confirm WindowSize (s)	Time window to integrate the frequency difference to confirm off-grid operation status of a device
Islanding Confirmation Threshold (mHz·s)	Once the integration over the time of IslandingConfirmWindowSize is completed, compare the integration result with this threshold. If the integration is larger than this value, the islanding status of one device is confirmed.
BackToGrid WindowSize (s)	For any device that is islanding, keep calculating integration of frequency difference over the time window of this value to check whether the device is back to grid.
BackToGrid Threshold (mHz·s)	Once the integration over time window of is completed, compare the integration result with this threshold. If the integration is smaller, the device is confirmed to be back to grid.

2.6 Detected Event Cases

After being implemented in the FNET/GridEye system, this detection tool has captured several UPS/Backup generation cases.

The following cases are presented to verify the effectiveness of the proposed algorithm.

Case 6: Microgrid at ORNL

The Distributed Energy Communication and Controls (DECC) Laboratory is a unique testing facility at ORNL for testing dynamic controls for both rotating and inverter-based distributed energy (DE) resources. The layout of the DECC facility and its connection to the bulk power system is shown in Figure 2.9.

The FNET/GridEye system is able to monitor the status of the microgrid through the use of FDRs installed on the microgrid.

In this case, the microgrid is isolated from the power grid and powered by its own DE. The proposed algorithm detected this off-grid condition, shown in Figure 2.10, with the frequency deviation, integration of frequency deviation being evaluated as 29.75 mHz, 1308.7 mHz·s and 100.23 mHz·s, respectively.

Case 7: Sandy

During Hurricane Sandy in 2012, off-grid status is detected. Figure 2.12 shows the start of the off-grid operation, with a frequency deviation 262.65 mHz and an integration of frequency deviation 18204 mHz·s.

It shows the islanded system synchronised with power grid while the integration of frequency deviation fell to 160.71 mHz·s.

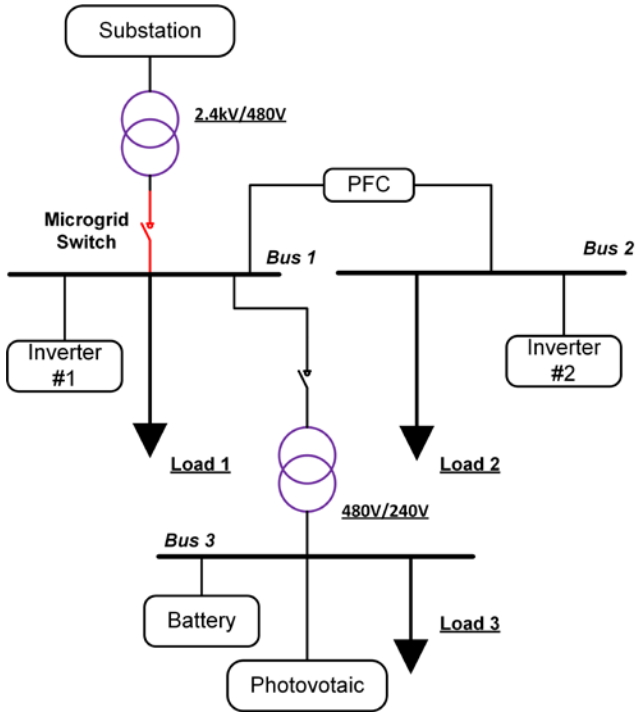


Figure 2.9 DECC Laboratory Interfaced with the ORNL Distribution System.

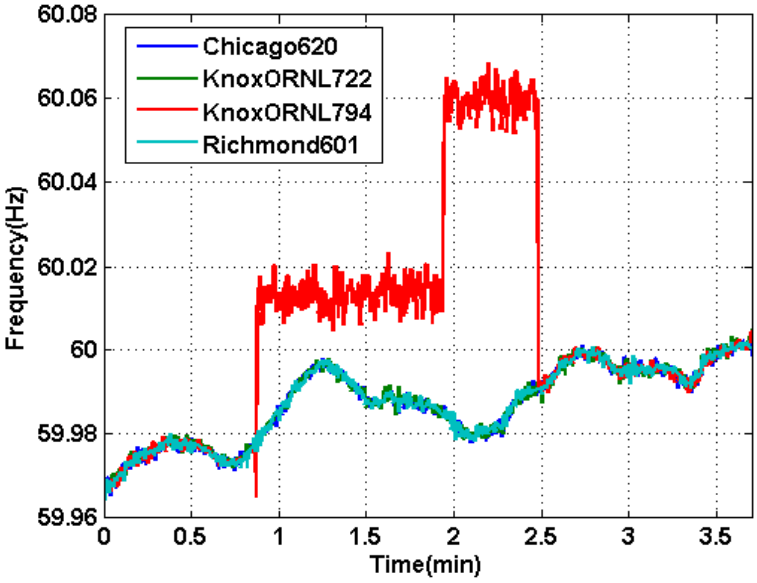


Figure 2.10 Microgrid Case at ORNL

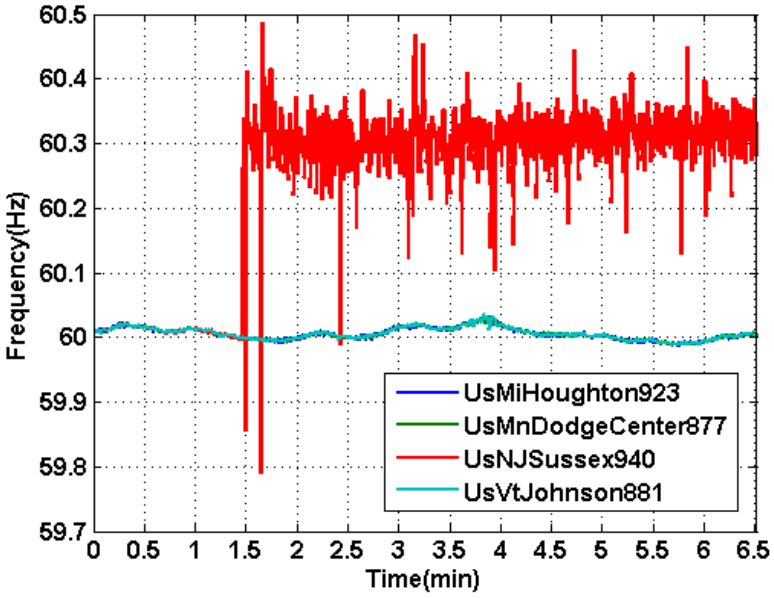


Figure 2.11 (a) Sandy case when starting

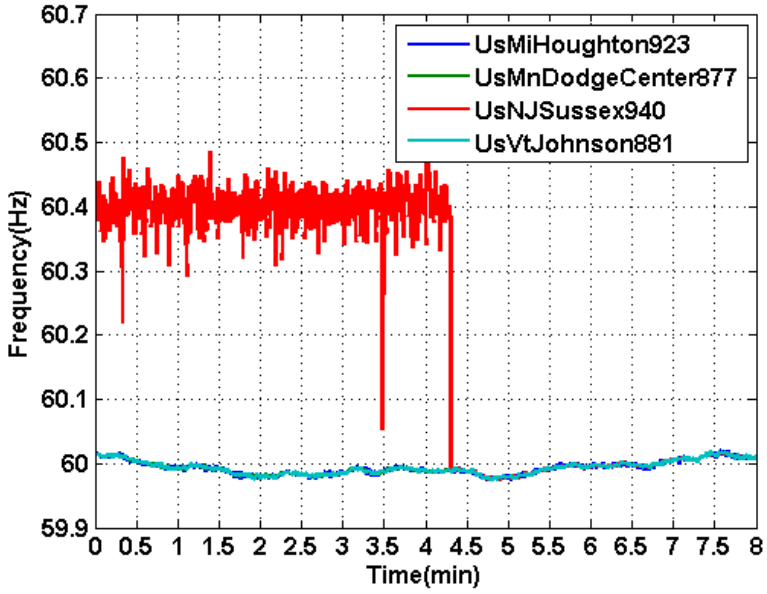


Figure 2.12 (b) Sandy case when ending

Case 8: Backup Generation

Figure 2.13 shows a backup generation case which occurred in Fredericksburg, VA on November 12, 2011. The event occurred from 19:30 to 19:50 (UTC) and was first detected by FDR857 with frequency deviation 639.7 mHz. The integration of frequency deviation was 46564 mHz·s, and was detected as off-grid operation. This event lasted around 20 minutes as shown in Figure 2.13, and when the integration of frequency deviation dropped to 133.98 mHz·s, it was detected that it returned to grid-tied operation.

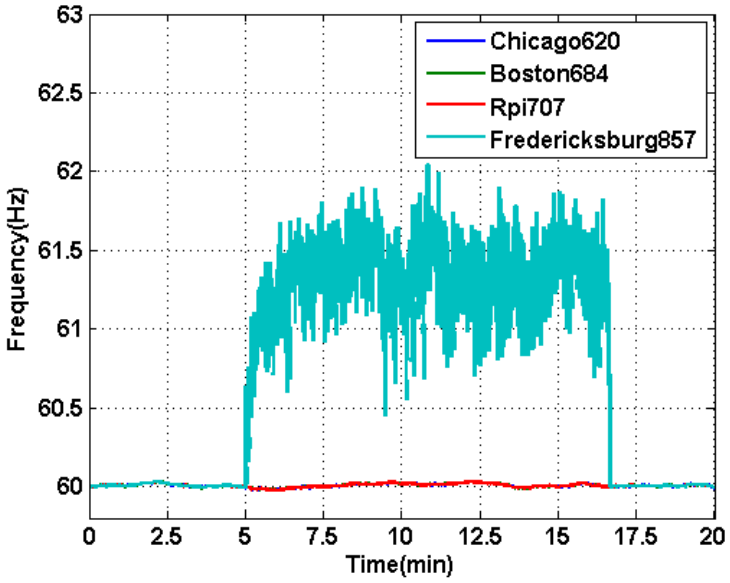


Figure 2.13 Backup generation case

2.7 Visualization

2.7.1 Real-time Map Display

The real-time map display retrieves the status and frequency measurement of each FDR from the database every four seconds and displays them on an interactive map.

Figure 2.14 shows the screenshot of the map display during Case 6. The dots represent the FDRs. Each dot is color-coded based on the deviation between its reporting frequency and the nominal frequency. Grey dots indicate the NODATA status of the devices. Once an off-grid operation is detected, the web interface will highlight the particular FDR with a red cross as an alert. A user can simply click on each FDR to check the device name, city and state, reporting frequency and status.

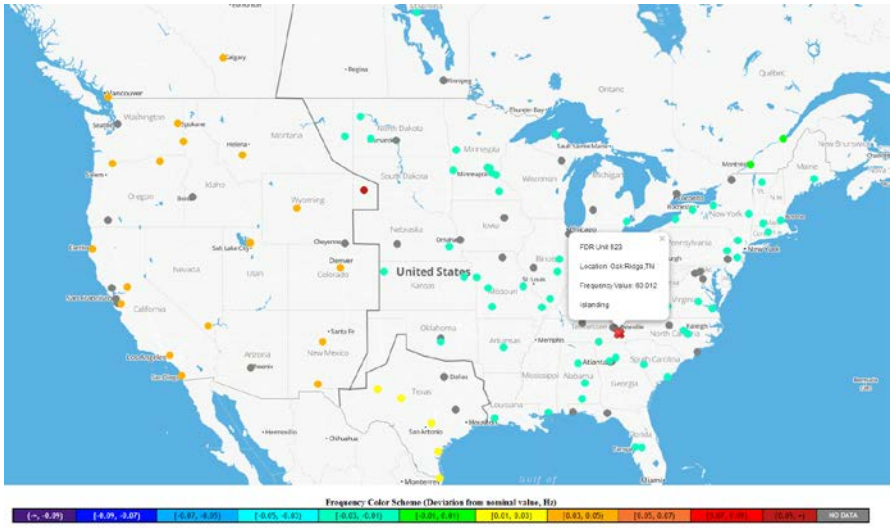


Figure 2.14 Real-time map display

2.7.2 Measurement Plotting Tool

The FNET measurement plotting tool is developed on the Windows Presentation Foundation (WPF) model using the openHistorian API. This tool can plot the real-time graph of frequency, voltage magnitude and relative angle to help visualize any abnormal system behavior. Moreover, this tool can plot historical measurements over any time frame, which is helpful for event replay. By observing the frequency and relative angle

plots of Figure 2.15, it is easy to identify the FDR represented by the purple line is off-grid.

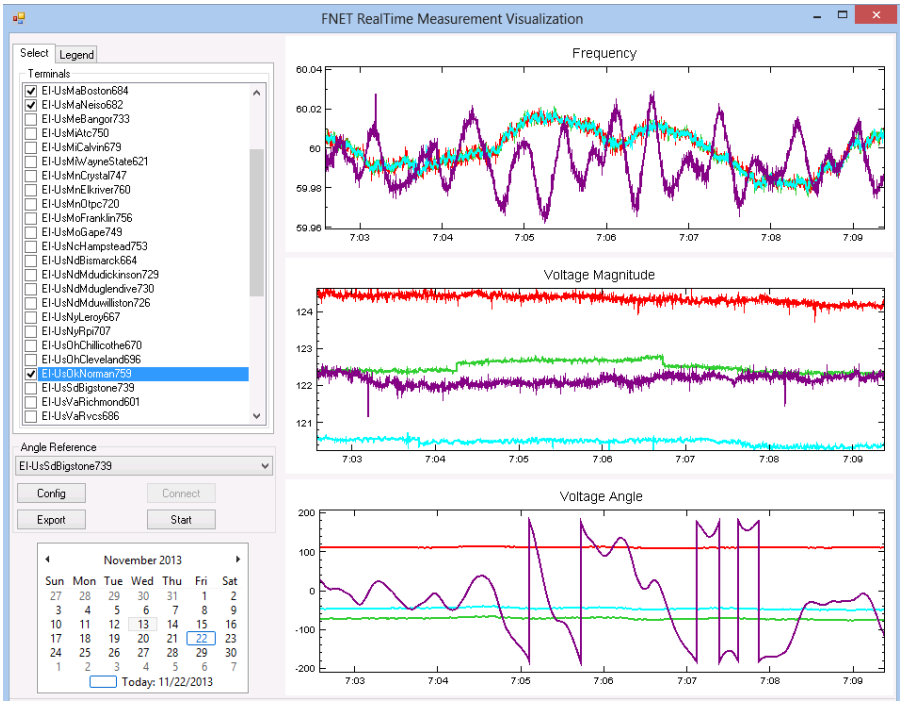


Figure 2.15 Real-time measurement visualization interface and event replay

2.8 Conclusion

The off-grid operation detection algorithm is proposed in this paper, and several case studies have been analyzed to verify the significance and robustness of this trigger algorithm. Moreover, a real-time detection algorithm is implemented in the FNET/GridEye system and has captured several events to confirm the validity of this method. The scalability, robustness, and portability of this system have been demonstrated.

In this work, uniform thresholds are used, but it is possible that the parameters need to be tuned to achieve smaller detection time delay in specific scenarios to boost the performance of the algorithm. Future work will concentrate on the development of a heuristic self-adaptive threshold tuner module to determine the trigger thresholds without sacrificing the fast-detection characteristic. Moreover, email alerts with detailed event information will be implemented as a service in the future.

Chapter 3. Ambient Mode Estimation and Visualization

3.1 Motivation

Electromechanical oscillation is an inherent property of power systems which cannot be fully eliminated, and poorly damped low-frequency oscillations would hazard the system stability. Since the 1996 Western Electricity Coordinating Council (WECC) system blackout [47], lots of research were conducted on monitoring low frequency phenomenon. Though the installation of Power System Stabilizer (PSS) and Flexible AC transmission system (FACTS) could help depress these oscillations, it still remains as a challenge since the frequency and damping of existing modes would change as a system evolves, and new modes may emerge, especially with renewable energy integrations. Therefore, reliable real-time mode estimation of low frequency oscillation and visualization is necessary [48], such that appropriate actions can be taken to maintain the stability of the power system [49] and furthermore to provide guidelines to tune the parameters in PSS.

With the aid of wide area measurement system (WAMS) and synchrophasor technology, measurement-based method is becoming a powerful tool to identify the electromechanical dynamic properties, as an implementation of WAMS at the distribution level at a low cost, FNET/GridEye system is capable of monitoring power system states with high dynamic accuracy. Based on the streaming data from these sensors, various applications have been designed and implemented and a comprehensive review of the system can be found in [1].

3.2 Literature Review

With the aid of synchrophasor measurements, it becomes feasible to use pure measurements to track the oscillatory dynamics [50], and many mode estimators have been developed under either ring-down or ambient conditions [51], [52].

1) Ringtown analysis requires system transient response data that is normally obtained from field tests or sudden disturbances, the representative approaches are Prony method [53], Matrix Pencil (MP) method [54], Hilbert Huang Transformation (HHT) [55], and Multivariate Empirical Mode Decomposition (MEMD) [56], [57], the oscillation modes can be captured with high accuracy.

2) Ambient condition indicates that the power system is operating in a steady state where no significant disturbances happen, and it is the natural response of system to small magnitude disturbances or random load changing. One of the typical methods is subspace-based technology, including subspace state-space system identification (N4SID) method [58], stochastic subspace identification (SSI) method [15], etc. Another frequency domain decomposition (FDD) is proposed in [59]. However, the main drawback of these technology is high computation burden due to the singular value decomposition (SVD) computation of a large-dimensional matrix, some approximation methods to accelerate the computation of SVD have been developed [9], [60]–[62].
Pencil (MP) method [9], and Hilbert Huang Transformation

3.3 Proposed Algorithm

This work introduces an approach to estimate the oscillation frequency and damping regardless of ring-down or ambient condition in real-time environment. It is achieved by utilizing the empirical mode decomposition (EMD) [63] to subtract the trend

from the frequency signal and then the Modified Yuler Walker (MYW) algorithm to derive a predefined auto regressive and moving average (ARMA) model. Moreover, this paper describes the implementation of a real-time mode estimation and visualization method using synchrophasor data obtained through FNET/GridEye.

The hierarchy structure of the mode estimation system is shown in Figure 3.1, it is composed of four modules, namely 1) data collection module, 2) parallel model estimation analysis module, 3) data storage module, and 4) end user monitoring module. The following parts narrates the functionalities of these modules.

3.3.1 Synchrophasor Data Acquisition

The geographically dispersed FDRs calculate phasor data timestamped by Global Positioning System (GPS), and then transmit the data to FNET/GridEye server. Then the server will feed the multi-channel timestamped frequency data streams to the mode estimation analysis modules.

3.3.2 Data Pre-processing

In data pre-processing, since the modes of interest are at frequencies below 2Hz, a low pass filter with a 2.5Hz cut off frequency is first applied to the measured data to exclude extraneous modes, which is noted as anti-aliasing filter. As for measurements-based oscillation analysis, detrending, which is used as a pre-processing step to prepare time series for analysis by methods that assume stationary, is the key for statistical techniques such as computing correlations, as well as spectral analysis techniques. One common constraint on this process is that the detrended signal must be a zero-mean process for the time span considered in the detrending.

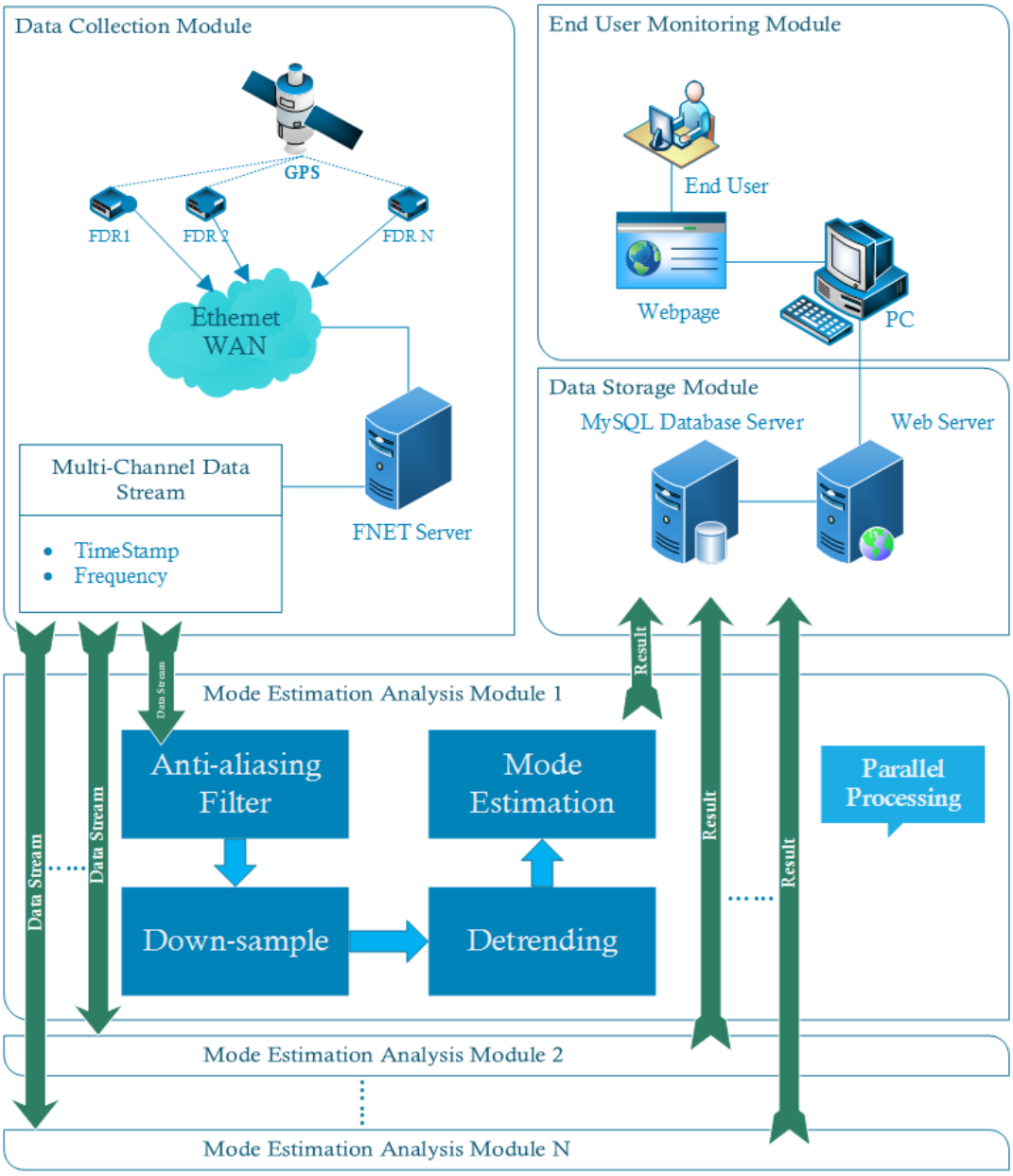


Figure 3.1 Hierarchy structure of the mode estimation system

The most common detrending techniques, such as regression, moving mean, or filtering are often based on stationary and linearity assumptions. Thus, these approaches may be illogical and physically meaningless when dealing with non-linear non-stationary data for real-world applications. If the functional form of the trend is not preselected, the processes of determining the trend have to be adaptive to accommodate data from non-stationary and non-linear processes. To tackle this problem, EMD method is utilized to adaptively identify the trend from the ring-down or ambient frequency signal, the steps to apply the method are listed as follows:

1. Identify all the local extrema of a time series y(t).
2. Formulate an upper envelop $y_{up}(t)$ and a lower envelope $y_{low}(t)$ by connecting all the local extrema using cubic spline.
3. Compute the mean m(t) of the envelope point by point.
4. Extract the details signal, d(t) = y(t)m(t).
5. Check the properties of the details signals.
6. Iterate on the residual m(t).

In practice, an inner loop that iterates steps 1 to 5 upon the detail signal d(t), until this latter can be considered as zero-mean according to some stopping criterion. Once this is achieved, the detail is considered as the effective Intrinsic Mode Function (IMF), the corresponding residual is computed and only then, step 6 applies. Eventually, the original signal y(t) is first decomposed through the main loop as

$$y(t) = d_1(t) + m_1(t) \tag{3-1}$$

and the first residual $m_1(t)$ is itself decomposed as

$$m_1 = d_1(t) + m_2(t) \tag{3-2}$$

Such that

$$\begin{aligned} y(t) &= d_1(t) + m_1(t) \\ &= d_1(t) + d_2(t) + m_1(t) \\ &= \ldots \\ &= \sum_{k=1}^{K} d_k(t) + m_K(t) \end{aligned} \tag{3-3}$$

where $m_k(t)$ is the final residue, $d_j(t)$ are the IMFs and K is the number of IMFs.

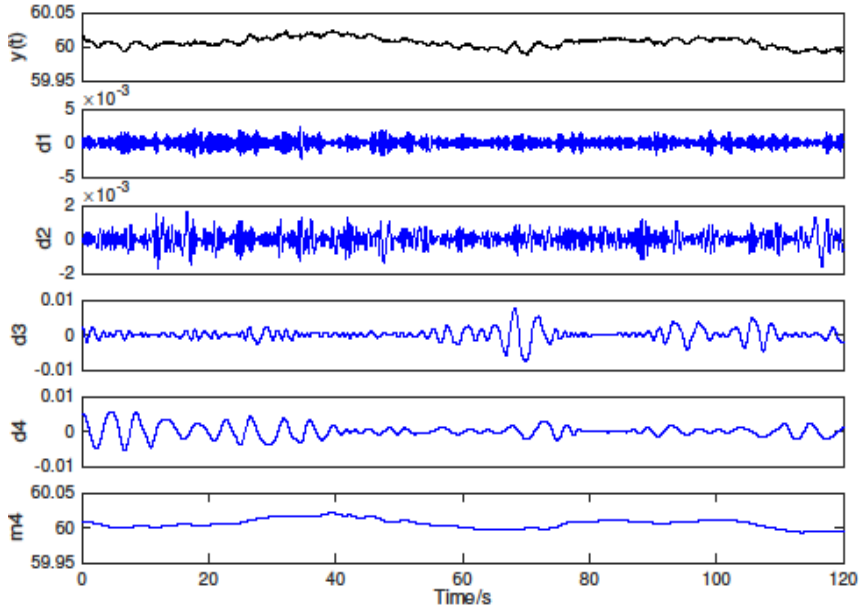


Figure 3.2 The IMFs and residue during the process of EMD

Figure 3.2 shows the results of EMD on 2 minutes of frequency data. The frequency bands contained in each IMF are different and change over time as the time

series y(t) changes. The basis of expansion of the EMD method is therefore adaptive and locally determined.

By construction, the number of extrema is decreased when going from one residual to the next (thus guaranteeing that the complete decomposition is achieved in a finite number of steps), and the corresponding spectral supports are expected to decrease.

The detrending method relies on the property of perfect construction and the spectral interpretation, therefore, the detrended signal is computed by superimposing partial IMFs. The detrended signals captured major oscillations while the slower drift and steady-state responses are removed.

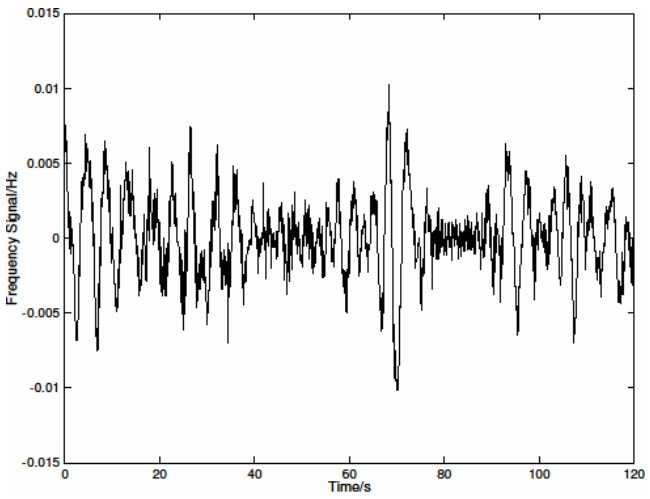


Figure 3.3 Detrended frequency signal under ambient situation

Figure 3.3 is the detrended frequency of original signal in Figure 3.2. The identified trend using EMD method is shown in Figure 3.4.

From the frequencies depicted in Figure 3.3 and Figure 3.4, it can be seen that the trend is well-represented under both ambient and transient conditions, which can be eliminated in the original frequency signal to facilitate the estimation steps.

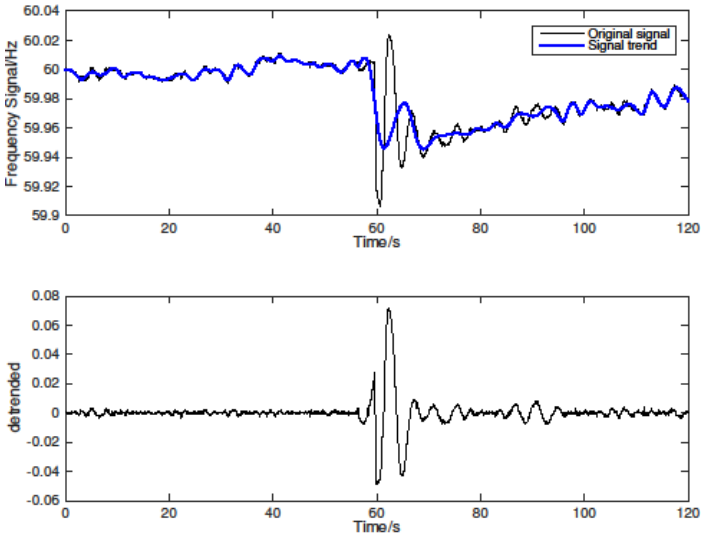


Figure 3.4 Trend identification and detrended frequency during an event

3.3.3 Estimation of Frequency and Damping

A linear ARMA model used here for representing power system data is written in the following form:

$$y(k) = -\sum_{i=1}^{p} a_i y(k-i) + \sum_{j=0}^{q} b_j e(k-j) \tag{3-4}$$

where p and q represent the order of the AR and the MA parts, respectively. According to [64], the output autocorrelation can be defined as

$$r_y(k) = \mathrm{E}[y(p) \cdot y(p-k)] \tag{3-5}$$

Then for the ARMA model, it can be derived that

$$r_y(k) \approx -\sum_{i=1}^{p} a_i r_y(k-i), k > q \tag{3-6}$$

then it results in a MYW equation for AR parameters [65] expressed by

$$\begin{bmatrix} r_y(q+1) \\ r_y(q+2) \\ \vdots \\ r_y(q+Q) \end{bmatrix} = \begin{bmatrix} r_y(q) & r_y(q-1) & \cdots & r_y(q-p+1) \\ r_y(q+1) & r_y(q) & \cdots & r_y(q-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ r_y(q+Q+1) & r_y(q+Q+1) & \cdots & r_y(q-p+Q) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} \tag{3-7}$$

Solving equation

$$s_j = \frac{\ln(z_j)}{T} \tag{3-8}$$

where T is the sample period. z_j (j = 1,…, p) are the roots of the characteristic equation (3-9), comprising the a coefficients of the model.

$$1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p} = 0 \tag{3-9}$$

The roots correspond to the eigenvalues of the system, thus the continuous time mode frequency and damping can be estimated as shown in (3-10).

$$\begin{cases} f_j = \dfrac{\mathrm{Im}(s_j)}{2\pi} \\ \zeta_j = -\dfrac{\mathrm{Re}(s_j)}{|s_j|} \end{cases} \tag{3-10}$$

3.3.4 Scalability

Each mode estimation analysis module processes one single channel of synchrophasor data as described above and these computation of each module are delegated to different threads, or cores. Currently since only limited FDRs are monitored, all the computations are executed on a single application server, with more FDRs been

installed and monitored, this method could be easily ported to distributed computation with highly scalability.

3.4 Real-time Monitoring and Visualization

3.4.1 Implementation Details

In real-time monitoring implementation, rolling window of frequency data streaming is utilized, and after the initialization process, the estimator will run every 4 seconds using the latest 5 minutes block data, and the current timestamp, estimated frequency and damping will be stored in a MySQL database, one table per monitored FDR. A php web server is built to exchange information between the database and outside requests in XML message. To visualize the estimated results, gauges and scatter plots are developed using Google Chart API [66], and Dygraphs Visualization Library [67] to show the instant value and trend of interested mode frequency and damping. The structure is depicted in Figure 3.5.

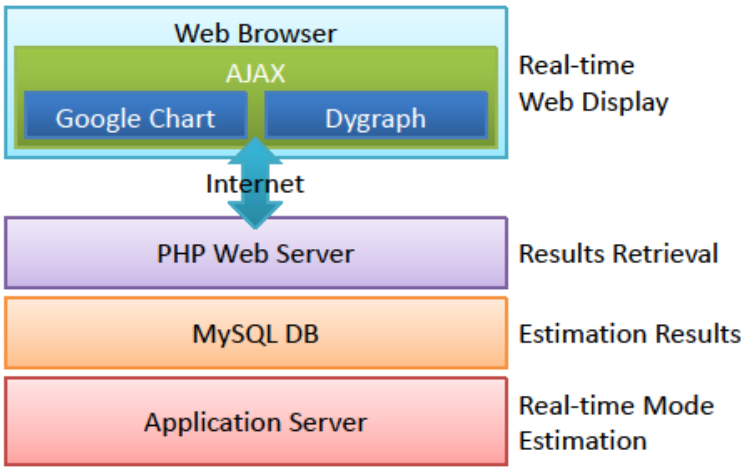


Figure 3.5 Real-time visualization structure

3.4.2 Missing Data and Network Delay

The data quality for synchrophasors are not always good due to GPS lost or communication issues, in this paper, once missing data or network delay are identified using the timestamp information, the estimator will not been executed until a new 5 minutes data block with continuously timestamped frequency data is fulfilled. Under this situation, if the browser sends request for data, the server will response with latest estimated values stored in database.

3.4.3 Demonstration

The webpage is shown in Figure 3.6, the top two drop-down lists are used to select the interconnection and FDR unit, after clicking on the 'Go' button, the webpage will refresh to show the mode estimation result for the specified unit. The four gauges indicate the current estimated frequency and damping, and the four scatter plots display the streaming of these values in real-time, and two interested frequencies range from 0.1-0.3Hz and 0.3-0.5Hz are monitored in this case. To prevent the webpage refresh each time a new value arrive, the asynchronous JavaScript and XML (AJAX) technique is used to initiate request to the server and update the value on-the-fly.

3.5 Conclusion

This work introduces a real time mode estimation system utilizing the synchrophasor data from FNET/GridEye and illustrated the algorithm used to screen the data and estimate the frequency and damping under both ring-down and ambient situation.

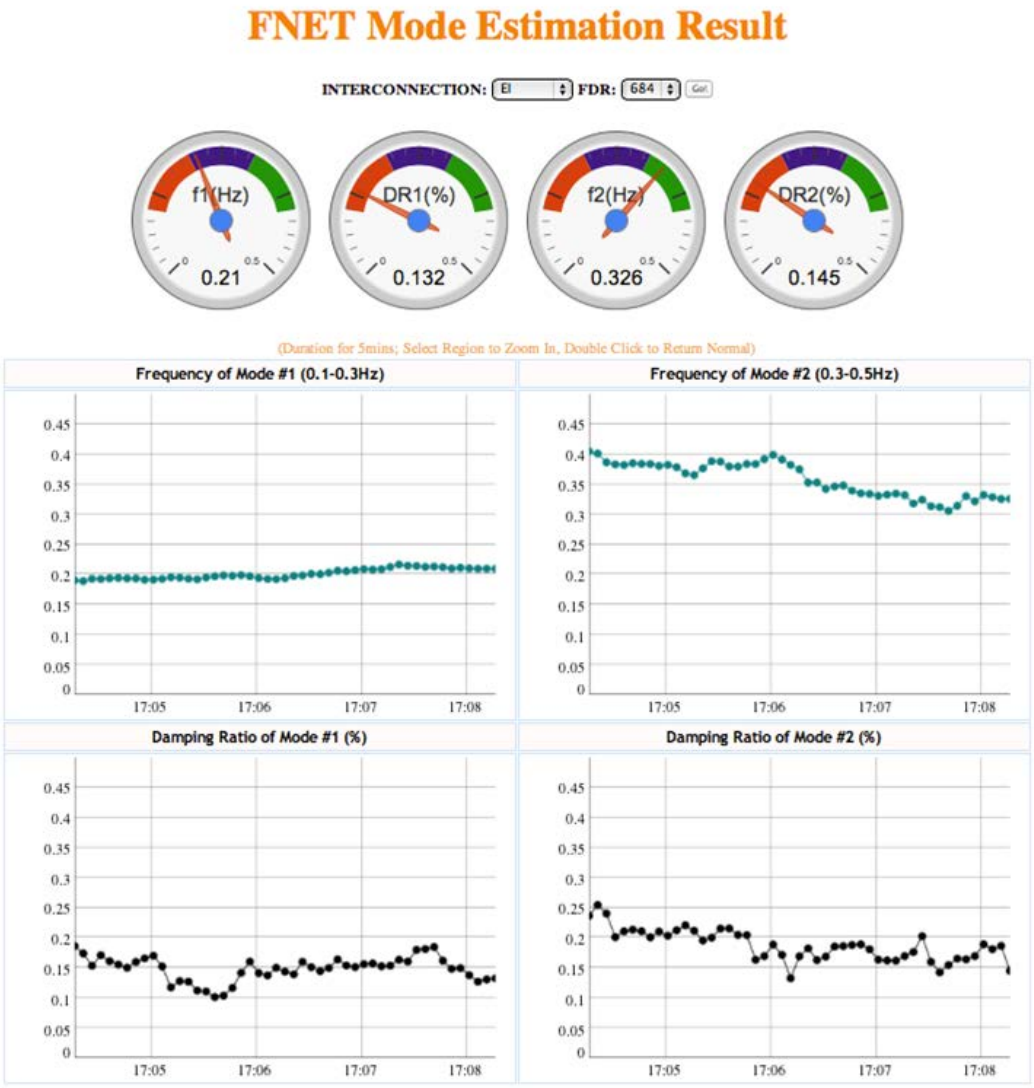


Figure 3.6 Mode estimate result from FDR684 in Eastern Interconnection

Moreover, the visualization technique is elaborated, and the web-based tool is demonstrated to show the ability to monitor the modes using a browser at any locations on a computer or mobile devices. Future work will focus on proposing more robust estimation method to be adapted to both ambient and ring-down situation, and utilizing multi-channel data stream as proposed in [7].

Chapter 4. Enterprise Data Historian Based on OSISoft PI

4.1 Motivation

With the expansion of the scale of power systems and widely used advanced information technologies, the quantities and categories of data and information at various resolutions are increasing dramatically. New phenomena and issues in power systems need to be recognized and analyzed. Therefore, data mining and analytics are critical for the power industry. However, several ubiquitous challenges are many information silos without cross system integration, the lack of global data description and data models and even insufficient common applications and services. They impede efficient data mining, waste valuable information and obstruct advanced data analytics in electric utilities under the big data environment. Meanwhile, these challenges also have a negative effect on the ever-growing business and delicate management of electric utilities. Dominion Virginia Power (DVP), which is one of the nation's largest producers and transporters of electrical energy, has also suffered these problems for a long time.

For the above challenges, academics have suggested some potential solutions [13], [68], [69]. Moreover, some European enterprises have been applying analytic strategies of big data to enhance customer management and operational capability. In addition, IT commercial giants, such as IBM, also have proposed the enterprise level integration solutions based on cloud computing in [70].

However, the majority of them are conceptual and not easy to practice. But the platforms are mainly for applications and services in control centers rather than the entire enterprise [71].

Since the existing solutions are not appropriate for large electric utilities like DVP, the optimal approach, which is exhibited in the paper, is to implement a robust data platform for improving the capability of big data management and analytics. In order to manage data and models flexibly and provide applications easily, the innovative platform may have four features.

1) Scalability: Though the number and the variety of data are increasing continuously, the platform may handle a growing amount of work and to be enlarged to accommodate the growth.

2) Real time: Huge amounts of high resolution real time data are integrated in the platform. Therefore, the platform with real time database uses real-time processing to handle workloads and enable access to historical and current data.

3) Service-oriented: The implementation of encapsulating services for hiding the trivial details is critical for users. In the platform, the numerous adapters and interfaces of third party applications or systems become a common service in the platform through standard protocols. For end users, the platform offers flexible and lightweight tools for data query, visualization, and analytics.

4) High reliability: The platform will be a core part of IT systems within the enterprise and provide services for different departments through the enterprise network. Given that the system can consist of a large number of hardware components, partial failures are unavoidable. Therefore, the platform design is to use cluster servers with load balancers to handle partial failures gracefully without causing service interruption.

4.2 The Methodology of Data Integration

With the rapid growth of both structured and unstructured data from multiple sources, the current IT infrastructure needs to be reorganized to optimize the flow of big data for fulfilling the intensive analytic applications. The implementation is to utilize the OSIsoft PI system [72] to build a highly reliable and flexible common data repository.

4.2.1 Types of Big Data

Big data sets for the enterprise-level data platform are depicted in Figure 4.1.

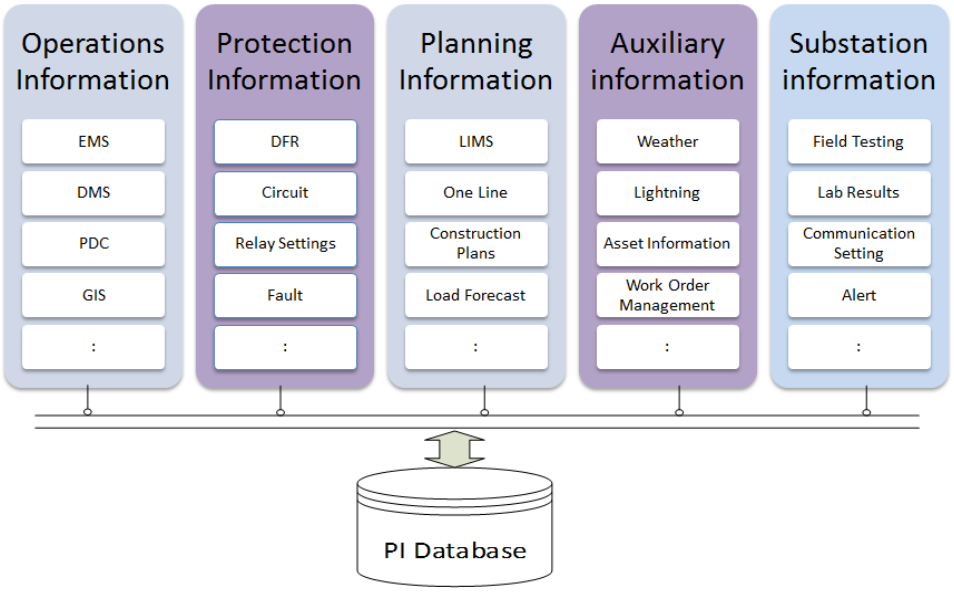


Figure 4.1 Types of big data for the enterprise-level data platform

The majority of real time data still depend on the Supervisory Control and Data Acquisition-Energy Management System (SCADA-EMS) since the deployment of remote terminal units is widely practiced and has provided the operators the ability of monitoring the operation status of the entire system. Meanwhile, the historical data from

SCADA-EMS contain abundant raw information for situational awareness and system planning. On the other hand, with the increasing number of phasor measurement units (PMUs), high resolution PMU data can provide more adequate dynamic responses and instantaneous values with accurate timestamps. In the distribution network, with the introduction of intelligent distribution automation equipment and distributed generation into the grid, the need to monitor, analyze, optimize and control the distribution system in real time is greater than ever, and the data from the Distribution Management System (DMS) play an important role to fulfill the above requirements. For protection technicians, comprehensive information from Digital Fault Record (DFR), relay settings and circuit calculation are critical for detecting and analyzing faults. In addition, traditional planning mainly focuses on the off-line limitation calculation, the design of the substation and network topology. If it is easier to involve more statistical information from the data platform, the planning decision should be smarter than ever. Moreover, many electric utilities maintain and collect the auxiliary information and substation information such as asset information, weather information, field test data, and so on. Such information is able to exert a greater contribution for management and operation in power utilities while it is integrated with data from other sources.

4.2.2 Implementation of Big Data Integration

Big data integration (BDI) is fundamental and critical to implement the vision of big data in terms of modeling, application and analytics. The value of data can be exhibited by data mining only when it is possible for disparate data to link and seamlessly interweave with other data to derive a unified and global representation. A

comprehensive review about power grid data architecture is presented in [73] and in [74], the author mentioned that BDI is different from traditional data integration in several dimensions. In Figure 4.2, the IT infrastructure of big data integration is presented.

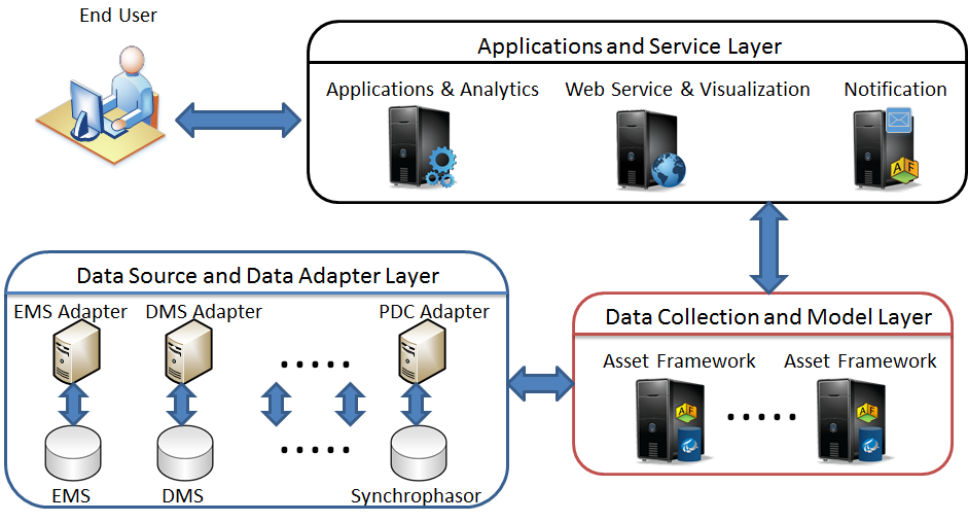


Figure 4.2 High level architecture of big data integration

Meanwhile, several core requirements such as scalability, real time, service-orient service and high reliability are needed to be fulfilled. In order to tackle the challenges, one practical approach is provided below:

1) Integrate the data through unified interface services: Unified interface services support connecting the platform to disparate data sources. Some interfaces enable history recovery, and some simply access the history stored in third-party historians. These data sources are seamlessly interwoven into the platform independent of source, protocol or vendor. Interfaces service enable buffer to multiple servers, intelligent data reduction,

single PI tag definition as well as point by point security. Redundancy and auto point sync are also available on interface service.

2) Archive into the data collective layer and build data driven model: Data is instantly stored in the archive servers of the platform and made available to users in real-time. Meanwhile, the hierarchical data modeling is able to create a consistent representation of assets or processes and associate data in proper context, providing information related to the data itself. The model may provide the easiest way for users to find the information they need.

3) Provide the applications and services to end users: In order to reduce barriers to use data and models, the platform provides the popular tools such as Microsoft Office Excel and mobile phone for end users to make them work on the data easier and implement analytics rather than waste time on data collection.

4) Backup strategy: The platform has redundant design and two groups of systems with same structure serve backup to the other. It can guarantee the reliability during the operation.

4.3 Hierarchical Data Modeling in Data Platform

4.3.1 Common Information Model and its Extension

Hierarchical structure is used in PI Asset Framework (AF) to restore and manage data. In order to represent the global data model in AF, the feasible approach is to utilize the hierarchical structure of the Common Information Model (CIM) in AF with customized extension. It may guarantee the data integration and interoperation from different systems.

CIM defines a common vocabulary and basic ontology for the aspects of power industries. Various CIM packages describe the basic classes and attributes for network, energy management, metering, and outage management and so on. However, since CIM would not contain all classes and attributes of a specified application, CIM needs to be customized and extended based on business requirements. For example, self-contained equipment containers, which are extensible and have a flexible structure, are used to build the hierarchical structure to manage the data at DVP. However, the standard CIM cannot serve the demand of building the hierarchical structure in order to maintain the styles and preferences of the data model at DVP. The existing types of equipment containers, such as Substation, VoltageLevel (VL), and Bay, which are depicted in Figure 4.3, are not self-contained and cannot satisfy the requirement of building a hierarchical structure. Thus, a new class called “FunctionLocation (FL)”, derived from EquipmentContainer (EC), is defined in Figure 4.3. An association is created between FL and EC as well, by which a FL may have sub-level ECs.

4.3.2 AF Implementation Based on Hierarchical structure

The AF implementation consists of two steps: creating the global AF template on the ontology layer and forming the association of data for creating the hierarchical tree.

For the first step, CIM profile, which is a subset model of CIM, needs to be created for AF implementation since not all the packages or classes of CIM are needed. Then an AF adaptor is developed to create AF templates according to the CIM profile based on AF SDK, as shown in Figure 4.4.

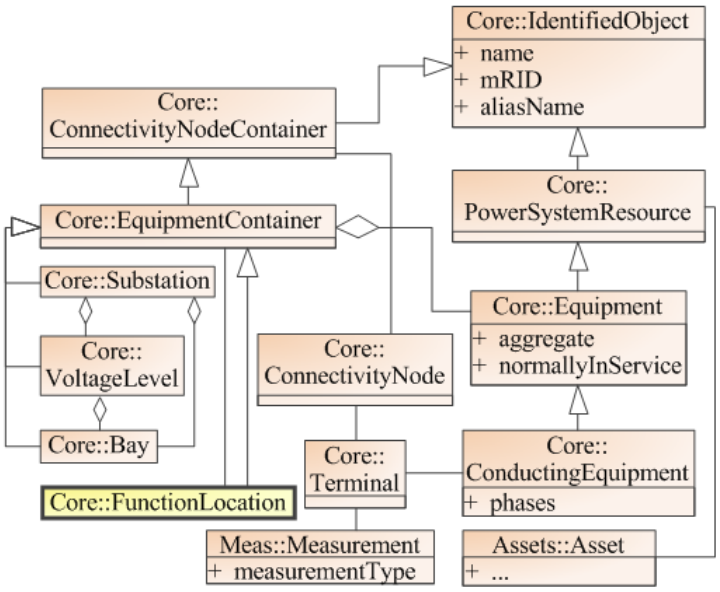


Figure 4.3 Chart for the example of CIM extension

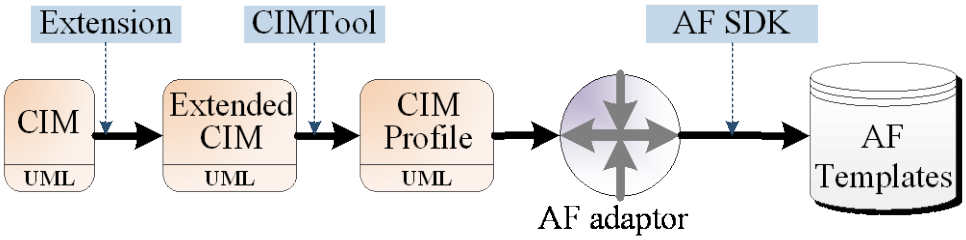


Figure 4.4 Flowchart of generating the AF templates based on CIM

For the second step, links between data objects need to be established. There are two types of links between objects in AF at the data level: Parent-Child and Reference. The ways of creating links in AF based on CIM are listed below: Aggregation between CIM classes is converted to a Parent-Child link between objects of AF.

Figure 4.5 exhibits the aggregation between EC and Equipment. As subclasses of EC, Substation and VL are aggregate of equipment, and Substation consists of VLs. In AF, equipment, such as buses, breakers, are children of VL, while VL are children of Substation, as shown in the right portion of Figure 4.5.

Association between CIM classes is converted to a Parent-Child link or a Reference between elements of AF. Here is an example to exhibit how to create links in Figure 4.6.

ConnectivityNode (CN), ConductingEquipment (CE) and a Terminal are used in CIM to describe the topology of the network. Considering most of CEs have two Terminals with the exception of BusbarSection and TransformerWinding which have only one Terminal, the association between CE and the Terminal is converted to Parent-Child link, while the association between CN and Terminal is converted to reference. Terminals are children of Buses, Breakers or other CEs, each of Terminal has a reference link to a CN, and CNs have several reference links to Terminals.

Once a hierarchical structure is created in AF, application and visualization can be built based on AF and achieved data.

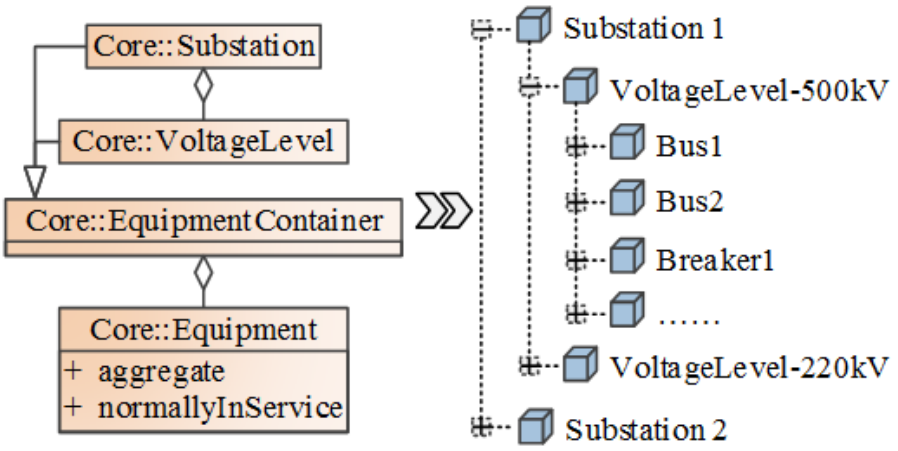


Figure 4.5 Aggregation between equipment and equipment container

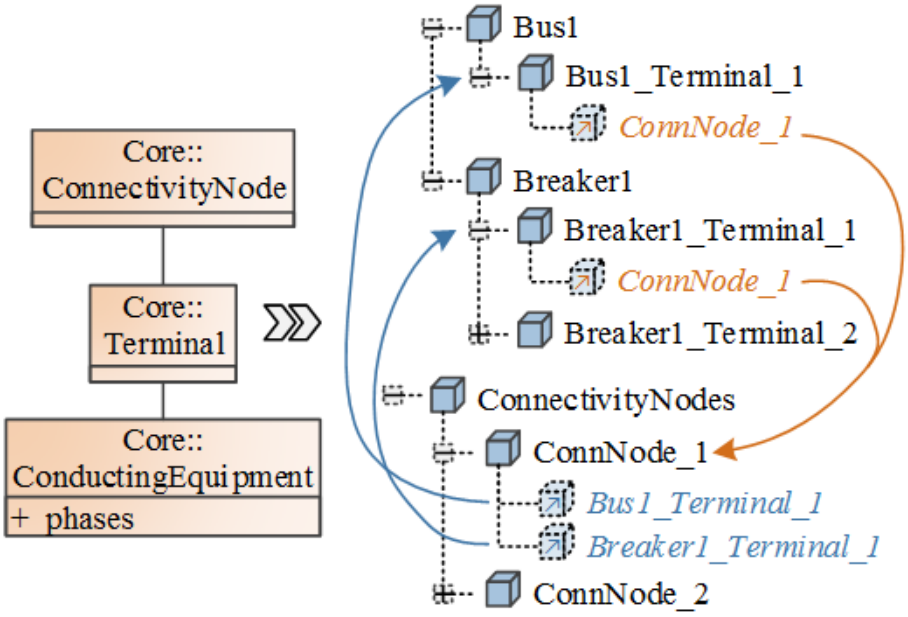


Figure 4.6 Associations between CN, Terminal and CE

4.4 Data Driven Analytics, Applications and Visualizations

4.4.1 Infrastructure of Data Analytics

1) Data Preprocessing

After collecting data from various data sources, the following steps are applied for data preprocessing.

- Exception: Utilizing a simple dead-band algorithm, the exception step uses some statistical values to decide whether to report a certain value to the archive, so as to reduce network traffic and storage memory.
- Compression: This step aims at ensuring that the archive stores just enough data to accurately reproduce the original signal, which is not only intended to save storage space, but to effectively remove noise and may improve system performance for end users without scarifying potential values of data.
- Outlier Screening: Due to unexpected sensors malfunction or communication errors, some outliers are mixed with original signals, this step ensures good data quality feeding to the data archive.

2) Data Mining and Pattern Discovery

With structural historical and real-time data integrated in the common data repository, a unified data source interface is provided. Moreover, the platform provides different levels of analytics tools, which vary from simple calculation using built-in math and statistical libraries, to high-level programming language development interfaces, such as C#, Visual Basic and MATLAB. Leveraging these powerful tools, massive data mining is used to find patterns from historical events as empirical information to support

future predictive asset management, to provide insights from typical scenarios for decision-making, etc.

3) Delivering Business Intelligence

The ultimate goal of this data platform and data driven analytics is to benefit end users, including system operators, circuit analysts, maintenance technicians, etc. With a common data repository, all of the end users could embrace cleaner, easier and smarter data, even build self-customized services and visualizations rather than completely depend on IT professionals to deploy the services. Meanwhile, since the data are provided by the global model, it is feasible to enhance asset management, improve situational awareness, reduce forced outage, extend equipment life, etc. The features of the robust data platform are that the common data repository can provide the data service with high security and end users can manipulate the data without corrupting and tampering the raw data.

4.4.2 Preliminary Applications and Visualizations

1) Assets Conditional-Based Maintenance (CBM)

To improve asset performance, reliability and lifecycle management of asset, it is critical to change the maintenance strategy from previous calendar-based to conditional-based, in order to reduce the unnecessary maintenance cost. The platform enables asset monitoring and analytics, and with predefined thresholds or conditions, it can generate notification, email alert and work order automatically.

Among assets management in a utility, power transformers constitute one of the largest investments, therefore it is a high priority to have an effective diagnostic tool for

condition assessment. Dissolved gas analysis (DGA) of insulating oil is considered the single best indicator of a transformer's overall health condition, with real-time monitoring data streaming, online DGA monitoring is built to visualize the trend of 8 critical gas concentrations for each transformer, as shown in Figure 4.7.

2) Comprehensive Diagram View of EMS

PI ProcessBook could import existing EMS information and automatically sketch one-line diagram, all the measurements, including voltages magnitude, current magnitude, active and reactive power of each line, switch and breaker status are updated using real-time measurements as shown in Figure 4.8.

3) Wide Area Monitoring in Transmission Systems

In a transmission system, frequency and voltage are key indicators of operational condition. The integration of synchrophasor data in this platform enables monitoring the frequency, voltage, power information in real-time, and building situational awareness applications so as to aid system operation. Figure 4.9 shows the frequency contour map during an event.

4) Extensible Visualization Platform

With this common information data model and provided uniform interfaces, it enables the 3rd party software to make communicate to the data repository and retrieve information to achieve innovative visualization. As one example, CIMSpy [75], which is a CIM-based exploratory tool, could be leveraged to visualize the power exchange between utilities as shown in Figure 4.10.

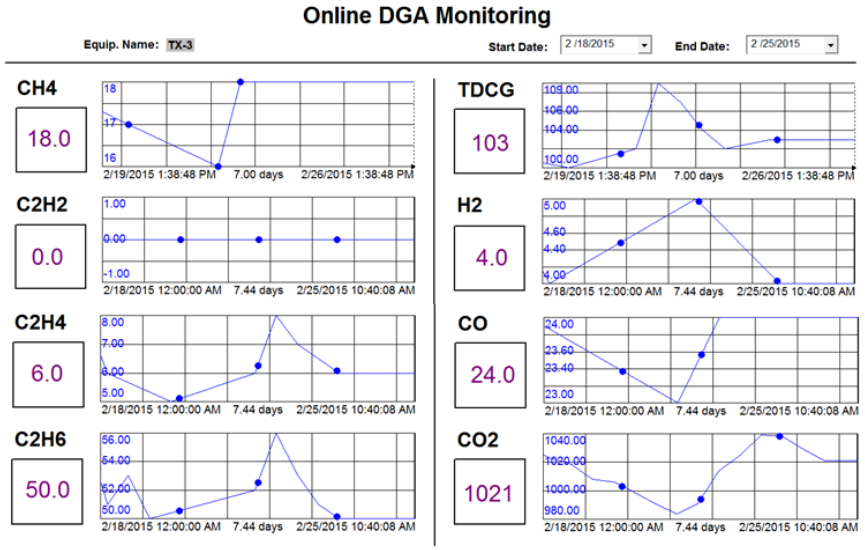


Figure 4.7 DGA visualization dashboard for transformer TX3

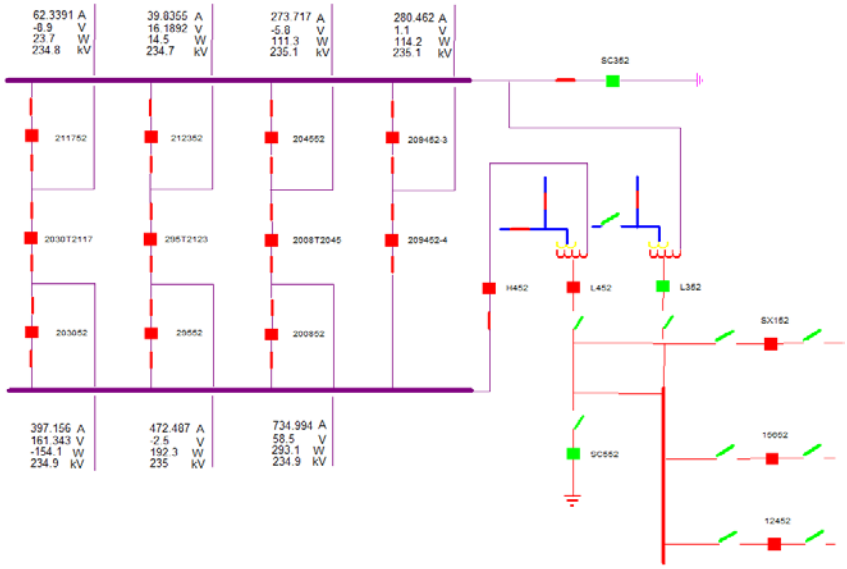


Figure 4.8 EMS one-line diagram in PI ProcessBook

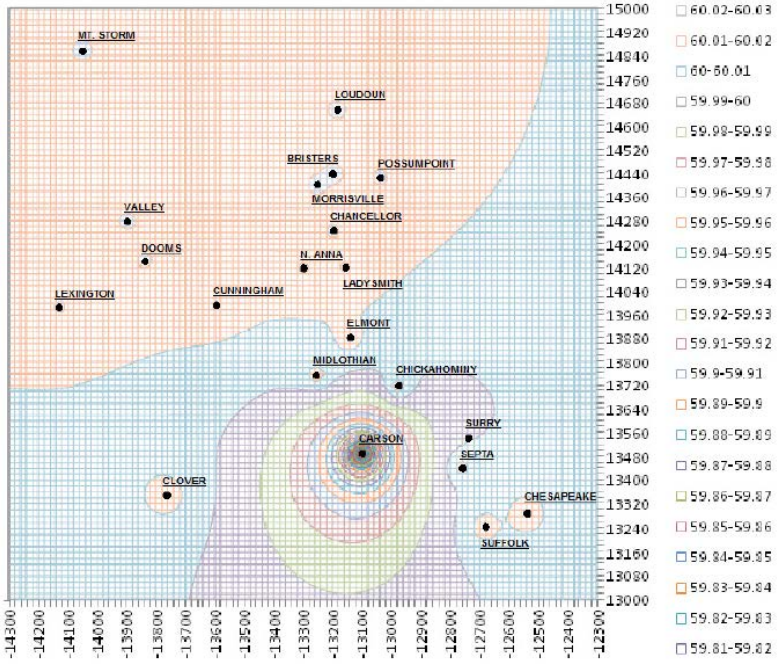


Figure 4.9 Frequency contour map

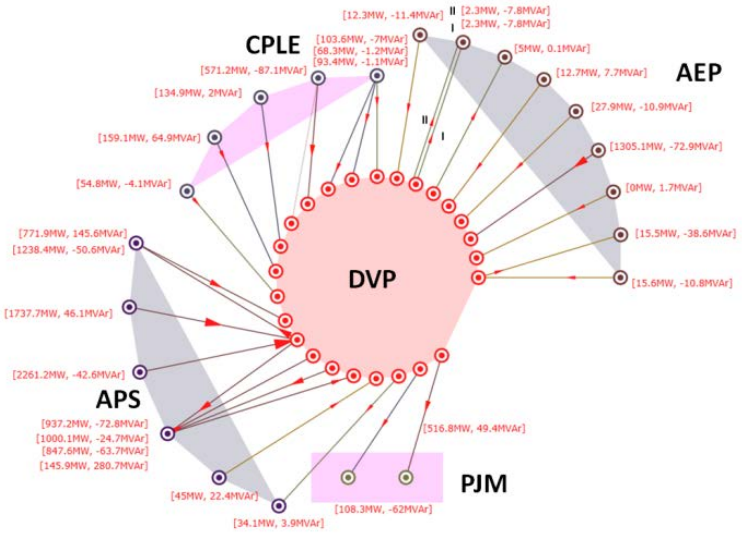


Figure 4.10 CIMSpy visualization for power exchanges

4.5 Data Validation

4.5.1 Purpose

Dominion utilizes a custom built historian, namely FM, to collects and archives distribution management system (DMS) data. Even though OSIsoft PI system has lots of interfaces and connectors, there are some systems that require a customized solution to achieve the required functionality. Partner with The Structure Group, a customized DMS to PI adaptor utilized the OSISoft Software Development Kit (SDK) and IIS Web Services is developed.

However, an evaluation of the performance of the adaptor has to be done before using it in production environment, and this data validation work is a main part of the evaluation task.

4.5.2 Dominion DMS/FM Overview

The DMS currently obtains data from distribution equipment and transmission data through a connection with the Alstom EMS. The data is currently stored in text based files by Dominion's custom built historian. As shown in Figure 4.11.

The data in FM are stored in a Linux machine, and due to security issues, it is not feasible to run script on that machine and no APIs are provided to fetch data directly using a program. However, a web portal can be used to manually export all the DMS data points into excel files with specific time range and data format.

Table 4.1 illustrates the details of the DMS/FM data.

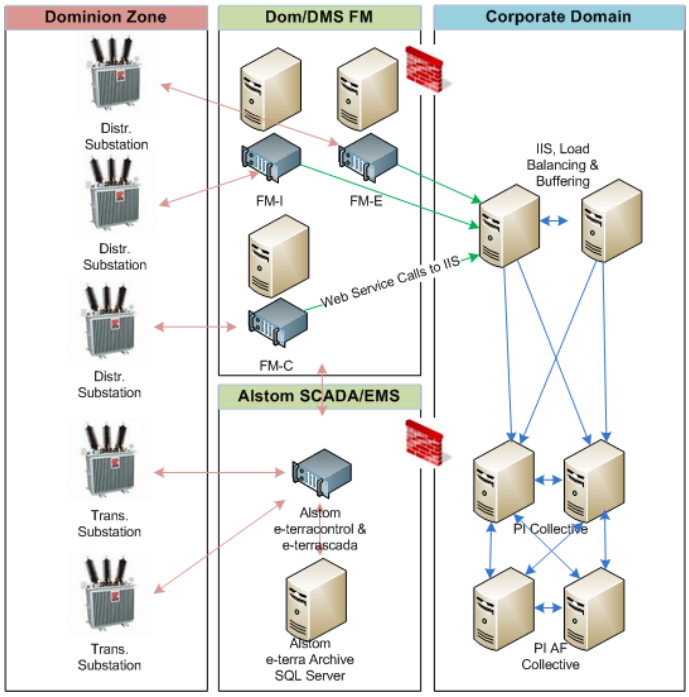


Figure 4.11 Dominion DMS/FM overview

Table 4.1 DMS data format

Data Source	***Element Name***	***Description***	***Sample Format***
DMS	Point Name	The DMS point name. This will map to the PI Instrument tag attribute. This value contains a "~" the PI tag cannot contain "~"	A.Device
DMS	Flag	This is the DMS data quality flag for a particular point/value. Up to 64 quality flags for each status and analog point. Dominion suggested that it might be necessary to define a bit map of those flags which are of interest.	4294967296
DMS	Value	This is the value for a particular DMS point. For string values double quotes will be used to identify values as a status.	-39.990000, "OPEN", "CLOSED"
DMS	Timestamp	Timestamp for a value in Unix seconds.	1023854350

4.5.3 Data Validation Requirements and Strategy

The data validation task mainly focus on determine whether all the information in DMS/FM is integrated intact and relative quality tag is built in the PI system. The following aspects need to be evaluated for a certain point:

- Whether the PI tag is created
- Whether the PI Quality tag is created
- Whether there is a value at the timestamp
- Whether the value is the same precision
- Whether the flag is the same

Due to the limitation of APIs to directly retrieve data from DMS/FM, intermediate files have be used as the DMS/FM data source, after obtaining the source data, the AF-SDK can be utilized to communicate with PI system. Figure 4.12 Data validation strategy shows the validation strategy.

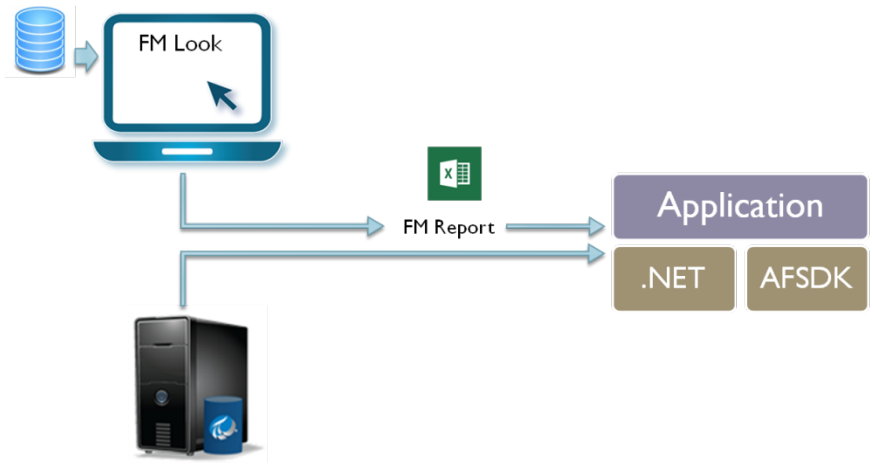


Figure 4.12 Data validation strategy

The validation application is built upon .NET framework and AF-SDK, written in C#, Figure 4.13 shows the detailed design of this application.

4.5.4 Auto Point Sync (APS) Tool

Dominion is using a customized interface, DMS adaptor to transfer the data from its original DMS/FM database to the PI data historian. However, the adaptor lacks the capability to create new PI points once new points are added in the DMS/FM database, which will result in data stream rejection by the PI data historian. To solve this problem, an auto point sync application is developed based on AF SDK to synchronize the newly added points in DMS/FM through creating corresponding PI point following a certain naming convention weekly. The hierarchy of the application is shown in Figure 4.14 and detailed design is shown in Figure 4.15.

4.5.5 PI Tag Management Tool

Due to large amounts of PI tags in PI system, for example, there are already 1,366,356 (statistics obtained at 05/19/2015) points from DMS so far and it keeps growing, it becomes much harder to keep track of the PI points, especially during the implementation of the naming convention (Change the PI point names according to a predefined rules). However, one of the PI point attributes “Instrument Tag” will reserve the original point name, so a PI tag management tool is needed to keep track of the PI points to ensure proper attributes are retained after naming convention, moreover, it can be used to identify missing PI tag in the PI system and create relative PI tag automatically.

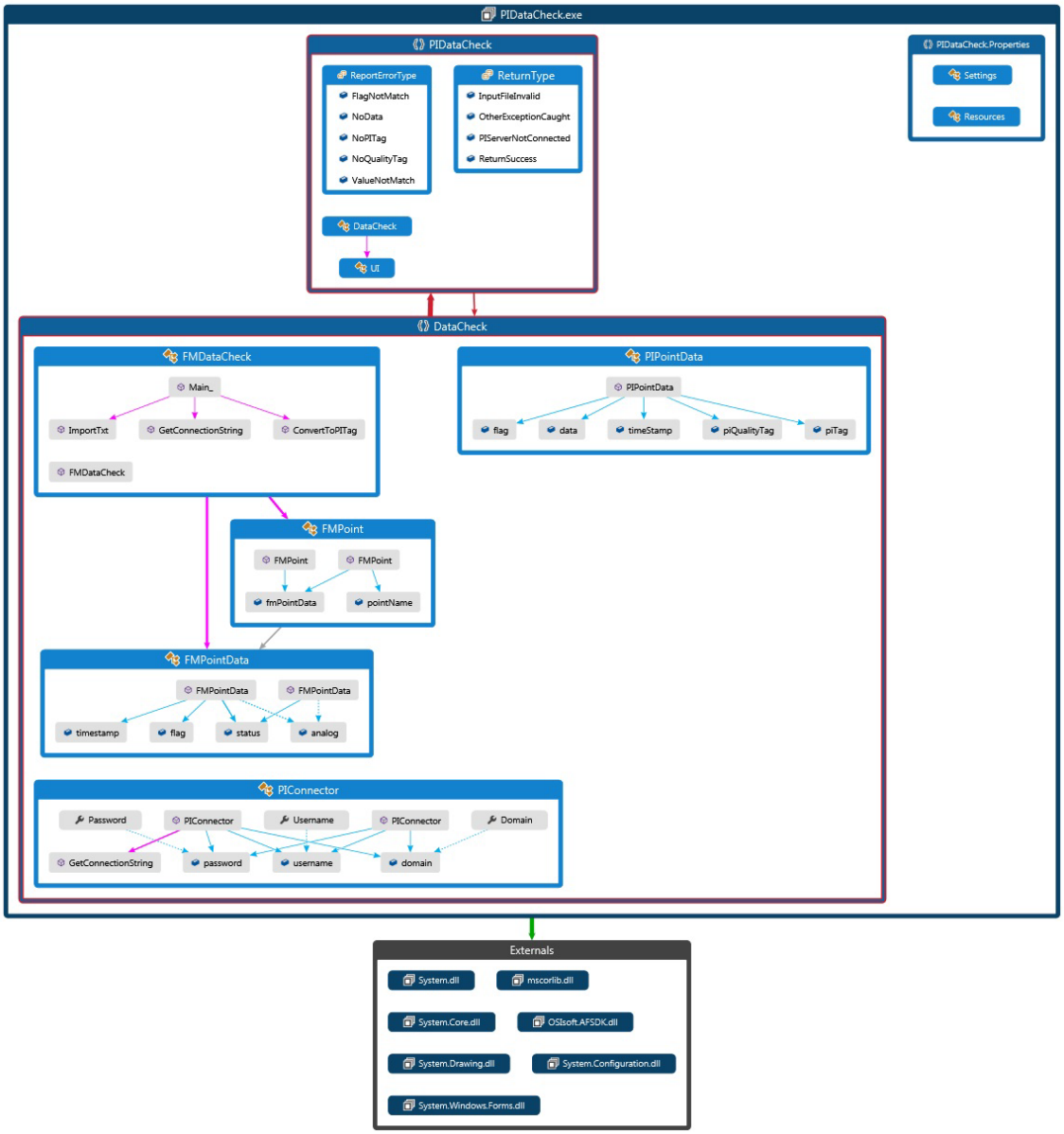


Figure 4.13 Design of the validation application

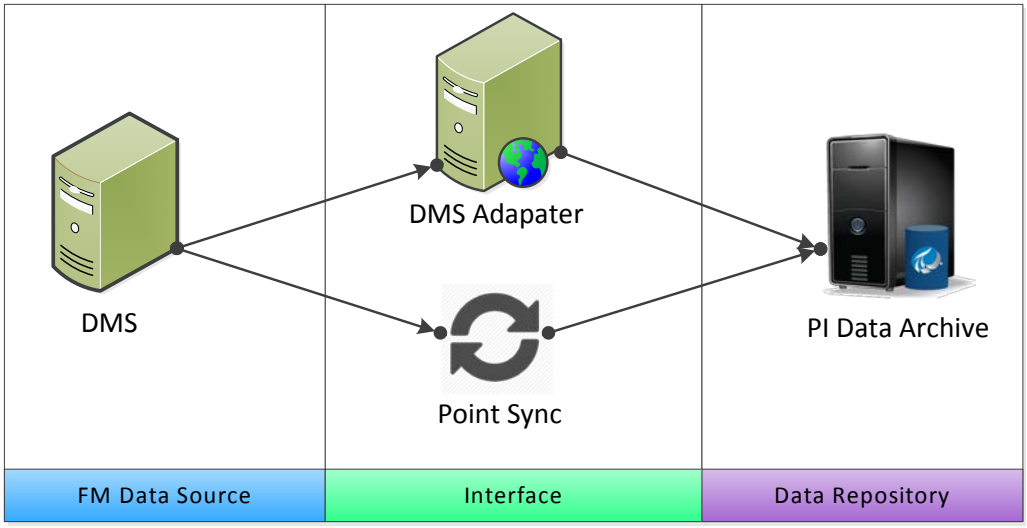


Figure 4.14 Point sync for DMS/FM data and PI

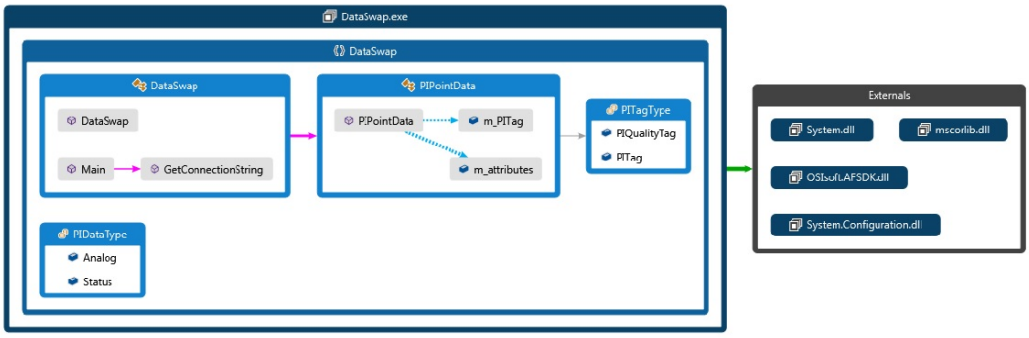


Figure 4.15 Design of auto point sync application

The application is built upon AFSDK using C#, and utilizes the build-in UI controls of the PI system, and the structure of the application is shown in Figure 4.16.

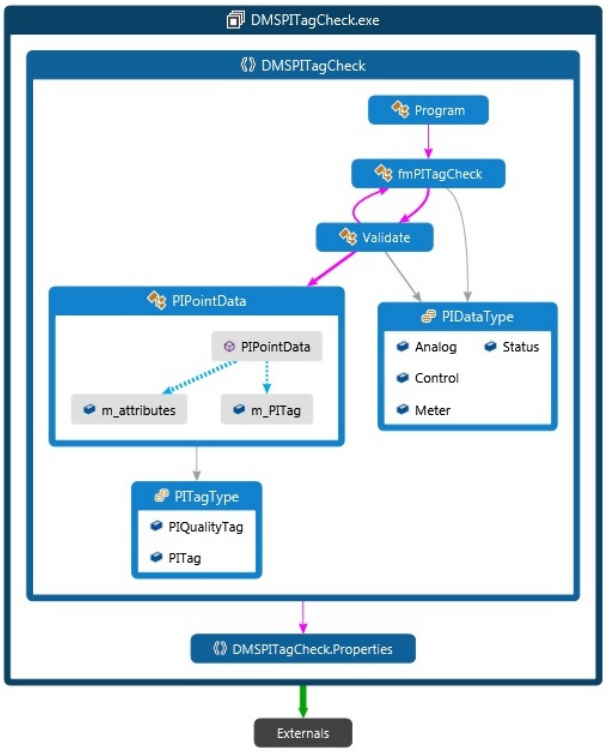


Figure 4.16 Design of PI tag management tool

Figure 4.17 shows the user interface of the management tool.

- Step 1: Choose the destination PI servers from a dropdown list, and connect to the server.
- Step 2: Browse the file system to pick the PI tag list files, generally in .csv file format, and then choose the PI point type by clicking the radio button.
- Step 3: Browse an output file or type an output filename in the text box.
- Step 4: Choose only identifying missing tags or identifying then creating tags tasks. Then click the ‘Start’ button.

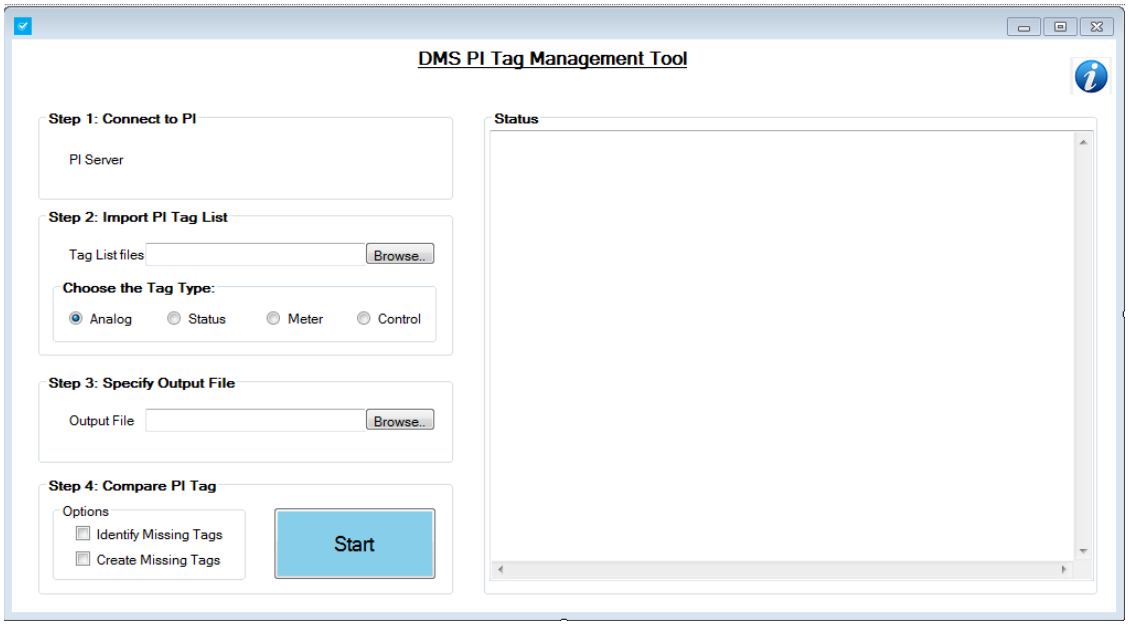


Figure 4.17 PI tag management tool user interface

4.6 SAP Integration

4.6.1 SAP Overview

Dominion is currently in the process of implementing a PI system for the integration, storage, and analytics of transmission and distribution data from various data sources such as SCADA/EMS, SCADA/DMS, field and lab diagnostic, and PMU data. In the future, Dominion desires to leverage PI-AF for analytics and displays which will require that all points included in the model be associated to SAP identifiers, such as equipment and functional location numbers. The high level architecture of the Dominion historian is shown in Figure 4.18.

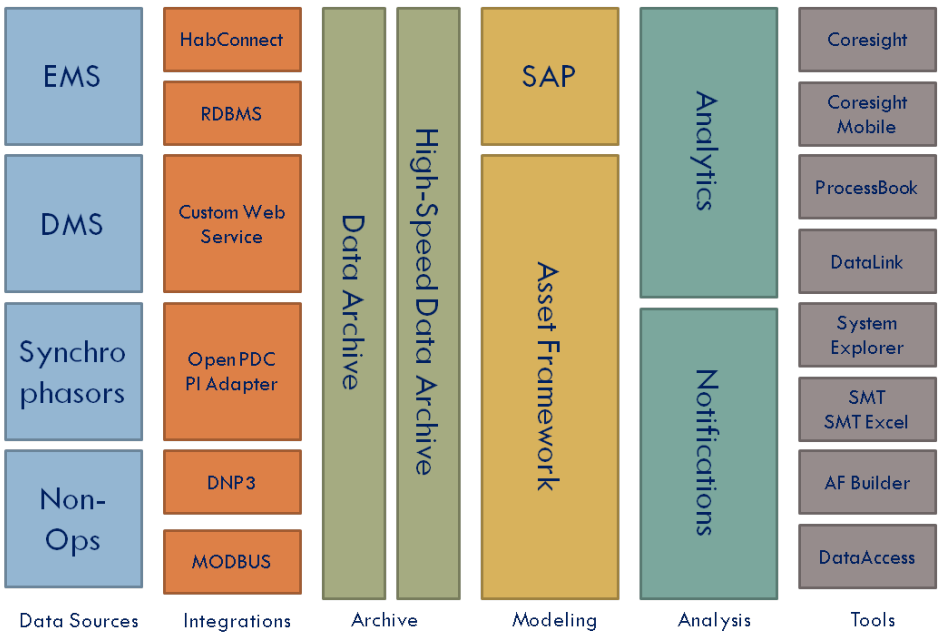


Figure 4.18 High-level architecture

After collecting the data into the PI data archive, the next step is to build data model in Asset Framework, to better leveraging the PI system for further analytics, notification and applications.

4.6.2 Implementation

To implement the SAP PI AF integration, the AF table lookup method is selected since it is a typical and most efficient way to integrate information using SQL queries, and link to the SAP database as the back-end, the following sections illustrate the equipment types, relationship between SAP and websis and SQL queries for different equipment.

4.6.2.1 Equipment Type

In SAP database, the field 'Object Type' is used to identify the type of an equipment, Table 4.2 summarizes the mapping between object type and equipment type, and also provides an example of the pattern of equipment number.

Table 4.2 Equipment type and object type relationship

Equipment # Series	***Object Type***	***Equipment Type***
SW123456	ABS	AIR BREAK SWITCH
A1234567	ARRESTER	ARRESTER
RLE123456	AMS	COMMUNICATIONS PROCESSOR
D123456	BAT	BATTERY
MON12345	BATMON	BATTERY MONITOR
B1234	BKR	BREAKER
MON12345	BKRMON	BREAKER MONITOR
CB123456	CAPBANK	CAPACITOR BANK

Table 4.2 Equipment type and object type relationship continued

CC123456	CCPD	COUPLING CAPACITOR POTENTIAL DEVICE
CHG12345	CHG	CHARGER
CLR123456	CLR	CURRENT LIMITING REACTOR
CS123456	CSW	CIRCUIT SWITCHER
CT123456	CT	CURRENT TRANSFORMER
MON12345	GASMON	GAS MONITOR
X123456	GENERATOR	GENERATOR
MU123456	JUNCTBOX	MAKE-UP BOX
L1234	LOC	LOCATION (SUBSTATION)
PT123456	PT	POTENTIAL TRANSFORMER
X123456	PT/CT	PT/CT COMBO
X123456	PVT	POWER VOLTAGE TRANSFORMER
X123456	RCL	RECLOSER
RLE123456	RELAY	RELAY
RLP123456	RLY_PROT	RLY_PROT
R1234	REG	REGULATOR
RS123456	RSWR	REACTOR SWITCHER
SST12345	SSTX	STATION SERVICE TRANSFORMER
X123456	SWI	SWITCH (VACCUM & OIL)
T1234	TXL	TRANSFORMER (LTC)
MON12345	TXMON	TRANSFORMER MONITOR
T1234	TXN	TRANSFORMER (NON LTC)
UG1234	UGC	UNDERGROUND CABLE
WT123456	WAVETRP	WAVE TRAP

4.6.2.2 Stored procedure

This section provided an example of the stored procedures for the each equipment type information.

1) ABS

Top Tab:

```
Select
e.EqNum,e.Floc,e.Description,e.ObjType,e.YrManf,e.Manf,e.ModelNum,e.Manf
PartNum,e.SerialNum,e.Owner,e.UserStatus,e.SysStatus,e.Profile,e.ValidFrom,sp
.SupremeParent,crew.EID as CrewID,supv.EID as SupvID, mang.EID as
MangID, p.CrewBackup,tech.EID as TechID, spec.EID as SpecID, techsupv.EID
as TechSupvID,specsupv.EID as SpecSupvID,p.Floc as Parent,
p.Crew,p.Supv,p.Mang,p.Spec,p.Tech,p.TechSupv ,p.SpecSupv,
p.Safety_Spec,p.GR_COMP_MONITORING,safetyspec.EID as
SafetySpecId,e.ConstYr,e.SAPPosition,e.Superior,e.Class,e.MainWorkCntr,e.Star
t_Up,e.InvNbr,misc.GR_THA_RISK,scr.SUB_CONST_REGION
From SapTransfer..Equip e Left Outer Join SapTransfer..FLoc c On e.FLoc =
c.FLoc
Left Outer Join SapTransfer..vwEquipSupremeParent sp On sp.EqNum =
e.EqNum
Left Outer Join SapTransfer..FLocParent p On p.FLoc = sp.SupremeParent
Left Outer Join BpSub..Misc misc On misc.eqnum = e.EqNum
Left Outer Join SapTransfer..Emp crew On crew.SapEmpCode = p.Crew
Left Outer Join SapTransfer..Emp supv On supv.SapEmpCode = p.Supv
Left Outer Join SapTransfer..Emp mang On mang.SapEmpCode = p.Mang
Left Outer Join SapTransfer..Emp tech On tech.SapEmpCode = p.Tech
Left Outer Join SapTransfer..Emp spec On spec.SapEmpCode = p.Spec
Left Outer Join SapTransfer..Emp techsupv On techsupv.SapEmpCode =
p.TechSupv
```

```
Left Outer Join SapTransfer..Emp specsupv On specsupv.SapEmpCode =
p.SpecSupv
Left Outer Join SapTransfer..Emp safetySpec On safetySpec.SapEmpCode =
p.Safety_Spec
Left Outer Join SapTransfer..Constr_Region scr On scr.Object =
sp.SupremeParent
WHERE e.objType='ABS'
```

Nameplate:

```
SELECT
[EqNum],[GR_KV_OPERATING],[GR_KV_MAX_DESIGN],[GR_BIL],[GR_C
ONTINUOUS_CURRENT],[GR_STYLE],[GR_WORK_ORDER],[GR_OP_ST
YLE],[GR_DRIVE_PIPE_LENGTH],[GR_DRIVE_PIPE_DIAM],[GR_HEIGHT
_GEAR_BOX_CENTER],[GR_OP_TYPE],[GR_WORM_GEAR_RATIO],[GR_
SW_INTERRUPTER],[GR_SW_INTERRUPTER_TYPE],[GR_SW_INTER_BO
TTLE_MFR],[GR_SUBSTATION_LOC],[TL_MAINT_RESPONSIBILITY],[T
L_INSIDE_STATION_FENCE],[GR_NOM_KV],[GR_WORM_GEAR_SERIA
L_NO]
FROM [SapTransfer].[dbo].[TL_Switch]
```

Motor Operator:

```
Select
EqNum,[GR_MANUFACTURER],[GR_TYPE],[GR_SERIAL_NO],[GR_MOT
OR_OP_VOLTS],[GR_TORQUE],[GR_YEAR_MANUFACTURED],[GR_MA
NF_STYLE_NO],[GR_PO_JOB_NO],
From BpSub..MotorOperator
```

4.7 Asset Framework Building

4.7.1 Asset Framework Overview

Business constantly changes and evolves. To remain current and valuable to users, the presentation of information must also evolve. Included with the PI Server, PI Asset

Framework allows defining a consistent representation of your assets and providing a structure for your information. PI AF allows to identify the components or elements that make up a process, specify relationships between these objects, and organize them in whichever way that makes the most sense to business. Once an element is defined, it can be associated with data as properties of assets. These may be static data, time-series data from PI Server, or information from other data sources such as relational databases.

PI AF supports a wide range of data types and objects including PI Event Frames. With PI Event Frames, one can search and analyze data in terms of meaningful events that have start and end times and other attributes. You define the events and their properties to track the events important to business.

PI Event Frames is flexible enough to model any event you have, such as downtimes, startups, batches, grades, environmental events and more. Furthermore, individual assets in PI AF can be organized and regrouped allowing multiple presentations of data. Since PI AF data is always in context and easy to find, you spend more time using the information and less time hunting for it.

4.7.2 AF Data Model

For the element of company, substation, equipment and relative information, monitoring, a hierarchy structure is utilized to reflect the “parent-child” relationship, in which way it is intuitive for users to understand the whole structure of the AF data model. As shown in Figure 4.19, the left figure shows the conceptual hierarchy structure and the right figure shows the exact data instances organized using this structure. Figure 4.20 shows an example of Transformer in substation Loudoun in DVP.

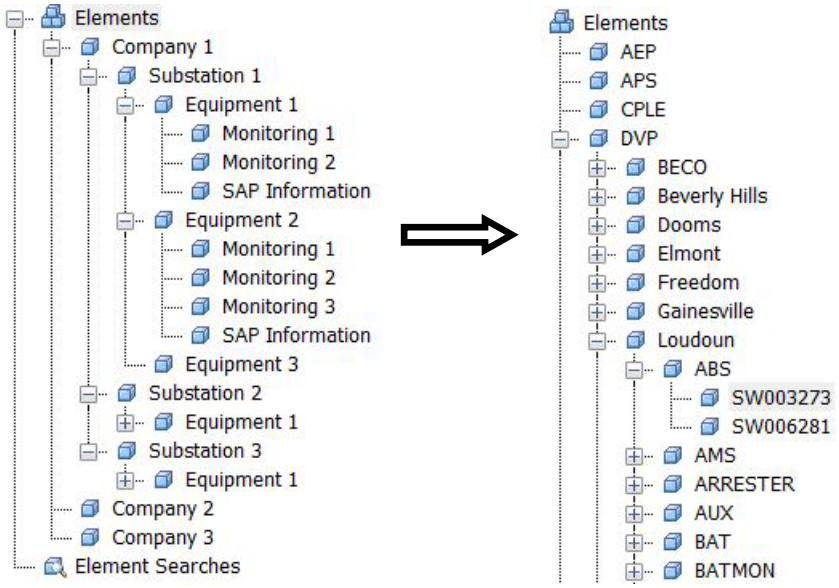


Figure 4.19 Hierarchy AF data model structure and instances

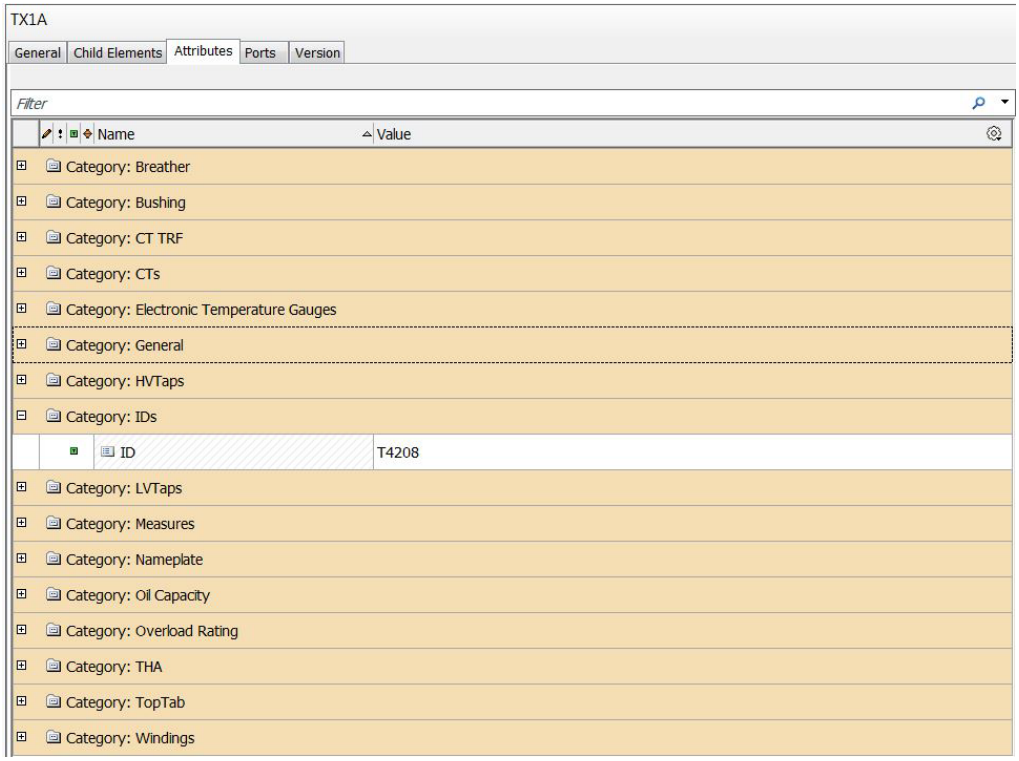


Figure 4.20 Flat category attribute data model

For the attributes of each equipment, a flat structure is utilized, to reserve the structure of the attributes, and “Category” is used to label attributes and groups them to a better way to organization. The reason a hierarchy structure is not used here is to reduce the complexity of the data model as much as possible to further simplify the operations that a user would have to take when using the system for analysis or visualization.

4.8 Synchrophasor Integration

4.8.1 Purpose

Dominion’s central PDC server and database is the central console that collects synchrophasor measurements from 82 PMUs installed in Dominion’s major 500 kV substations, archives the data and applies data processing to provide the ability of exporting the data to visualization and analytic tools. Dominion’s present PMU-PDC network achieves a full coverage of Dominion’s 500 kV backbone network and a portion of 230/115 kV network. OpenPDC is maintained by the same company, openPDC PI adapter is aimed to establish a secure and efficient transmission session between the software and the under-deployment PI common data repository of Dominion.

4.8.2 OpenPDC-PI Adaptor Evaluation

Figure 4.21 shows the architecture of the PMU data integration in PI system. The version of the adapter evaluated in this report is published with OpenPDC v1.5, and written in PI SDK. Figure 4.22 shows the version of PI-SDK utilized in this testing. In terms of the libraries utilized for the PI adapter, COM based PI-SDK is used in the testing.

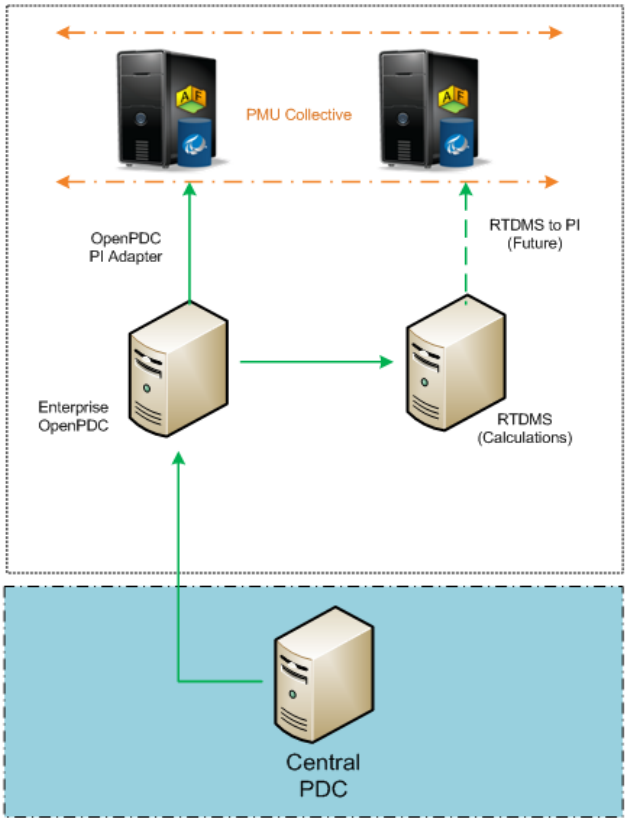


Figure 4.21 PMU integration in PI system

PISDK -Installed/Registered Components (32-bit)

File View Options

File Name	Version	Date	File Path
MDBTree.ocx	1.06.0365	8/15/2014 2:49:18 PM	C:\Program Files (x86)\PIPC\PISDK
pi2srvr.dll	1.4.4.484	8/15/2014 2:43:46 PM	C:\Program Files (x86)\PIPC\PISDK
pisdk.dll	1.4.4.484	8/15/2014 2:43:56 PM	C:\Program Files (x86)\PIPC\PISDK
PISDKCommon.dll	1.4.4.484	8/15/2014 2:44:00 PM	C:\Program Files (x86)\PIPC\PISDK
PISDKCtl.ocx	1.06.0400	8/15/2014 2:49:22 PM	C:\Program Files (x86)\PIPC\PISDK
PISDKDlg.dll	1.08.0420	8/15/2014 2:48:52 PM	C:\Program Files (x86)\PIPC\PISDK
PISDKParse.dll	1.4.4.484	8/15/2014 2:44:04 PM	C:\Program Files (x86)\PIPC\PISDK
PISDKRegistry.dll	1.4.4.484	8/15/2014 2:44:08 PM	C:\Program Files (x86)\PIPC\PISDK
PITimeServer.dll	1.4.4.484	8/15/2014 2:44:12 PM	C:\Program Files (x86)\PIPC\PISDK
sdktrace.dll	1.4.4.484	8/15/2014 2:44:34 PM	C:\Program Files (x86)\PIPC\PISDK

Figure 4.22 PI-SDK versions

4.8.2.1 Testing Environment

The evaluation was conducted in Dominion's RTDS laboratory at Grayland Ave. A Schweitzer SEL-3355 which had OpenPDC v1.5 and PI SDK adapter installed was used as the test bed for the data exchange. The SEL-3355 has received raw synchrophasor data from the central PDC server sitting in Dominion's Enterprise network, composed and delivered PI structure tags to the PI Q/A Server with the appliance of Dominion's naming convention rules. The transferred data then has been analyzed in PI-in-the-box environment.

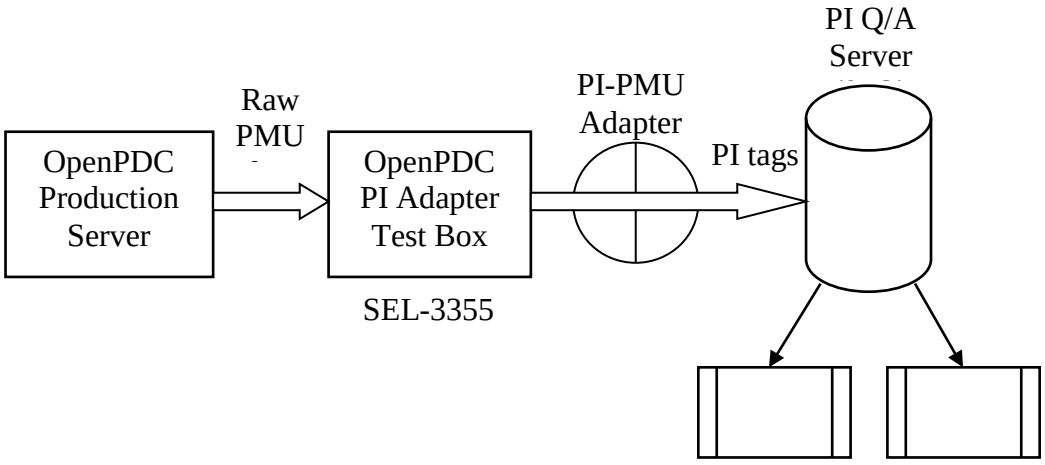


Figure 4.23 PI-PMU adapter testing environment

4.8.2.2 Configuring OpenPDC to Communicate with PI system

The key parameters are:

- Connection String: Used to specify a set of parameters for the adaptor configurations.

- ServerName: Destination IP address, which refers to PI system
- Pipointsource: Indication the data source, here it is *synchrophasor*.
- runMetadatasync: With this parameter set as True, the adaptor will automatically create PI tag using default setting once the adaptor realizes the data stream it is sending doesn't have relative PI tag in PI system.

- Assembly Name:
  - PIAdapters.dll: The assembly of the adaptor, if there are some functionalities change overtime, only need to recompile the assembly.

Figure 4.24 shows the OpenPDC manager interface to configure the PI adaptor to send PMU data stream to PI system.

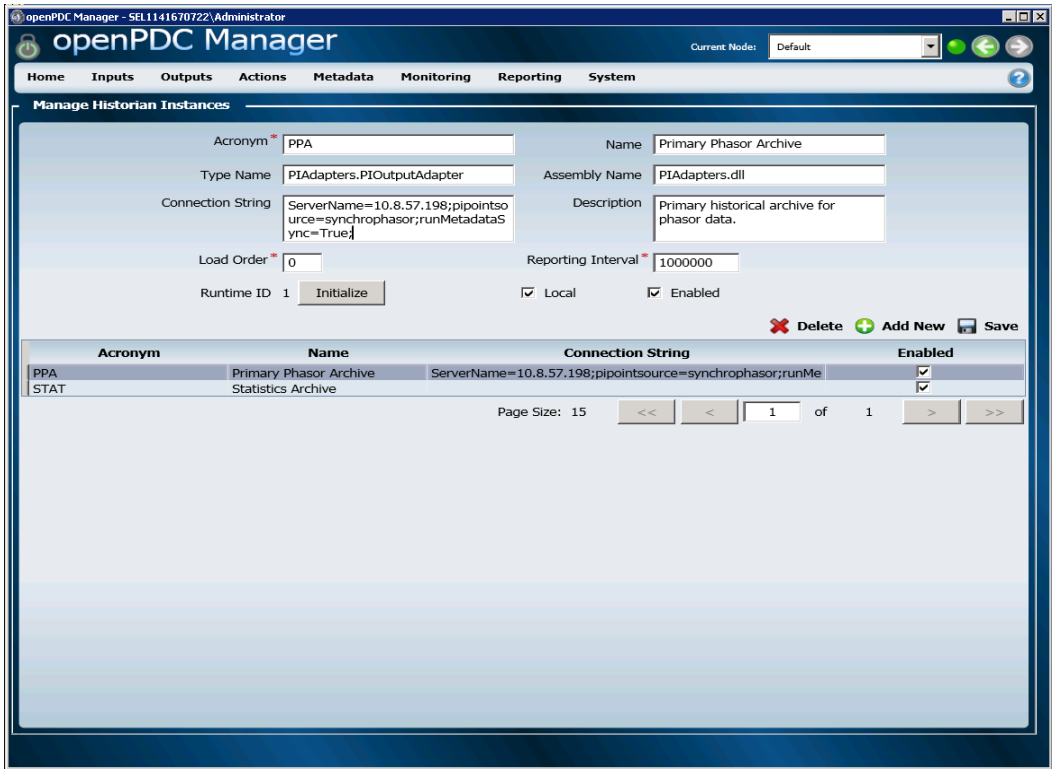


Figure 4.24 OpenPDC manger interface

4.8.2.3 Data formats

Three types of data formats are exported and will be archived in PI:

- Analog: Site frequency, voltage angle and magnitude, current angle and magnitude, real and reactive power
- Digital: Breaker status
- Status: PMU channel condition

4.8.3 Evaluation

4.8.3.1 Sampling rates

Synchrophasors process and archive measurements at a very high sampling rate, which give users a great facility to achieve real-time monitoring and online applications. The raw data at device is archived at 60Hz; while the data archived at central PDC server has been down sampled to 30fps. The data transferring follows IEEE C37.118 protocol for high resolution, high quantity and high volume data exchange. The PI SDK adapter for PI is in compliance with IEEE C37.118.

One essential goal of the test is to validate the adapter's capability to transfer data with 30fps resolution without dropping packets/data. Adapter timeout and communication errors are also under evaluation. Through the testing, it is validated that the adaptor is capable of transfer data with 30fps resolution.

4.8.3.2 Data Consistency

The main task of the adaptor is to transmit the data from OpenPDC to PI data archive and it is necessary to ensure the data consistency. The consistency of data is crucial for further data driven applications, any discrepancy in data may result in a

controversial control action or an unnecessary maintenance, etc. This testing is to evaluate whether there are any data value variations and duplicated values due to the communication error, package loss or data packing issues. During this testing, it is found that the connection sometimes will be aborted due to the server issues, however, the adaptor itself works fine. The following figures show the comparison of the original data and the data in PI data archive.

4.8.3.3 Latency Calculation

Data streaming latency is critical for various real-time or near real-time applications utilizing PMUs data, a small time latency value is the fundamental that many real-time applications using PMU measurements can effectively fulfill their functionality. When the PI system is in place, we could imagine many real-time applications that require minimum data latency to be implemented in Dominion, including RTDMS, PGDA and customized applications using PI analysis and AF-SDK. To name a few: intelligent line reclosing program using PMU measurements at two ends of the line; real time VAR control program, islanding detecting module, linear state estimator. For such applications, 1ms second of delay equals 21.6 degrees of angle variance. These calculations are all taken place in the PI server end in the buffer of the PI system, after the data is transferred by the adapter. So the performance of the adapter is essential for them.

Dominion to PJM's PMU data exchange has a latency of ~160ms, which successfully meets the requirement from PJM to apply data visualization and related calculation. The goal of maximum latency in this evaluation was set to be 120ms. Considering the possible time delays in the above data transmission procedures, this goal

is reasonable and satisfies major PMU functionalities that would be achieved in Dominion.

The latency contributors are listed below:

- Field to substation PDC
- Substation PDC to central PDC
- Central PDC to Test PDC environment (SEL-3355)
- Test PDC environment to PDC Adapter
- PDC Adapter to PI system

Latency 1: Dominion's field PMU devices measure and process synchrophasor data and send the data packets to substation PDC. The data transmission is instantaneous and estimated latency is ~30ms;

Latency 2: Substation PDC is responsible to resample and send the data packets to Dominion's Central PDC which is located in DVP's enterprise network. The latency of this stage mostly depends on the setting of substation PDC on waiting time, the transmitting distance and the network flow.

Latency 3: After gathering data from substation PDCs within the settled waiting time, central PDC and openPDC server will process all data streams and reconstruct the data packets subject to the settled time period. Then the data packets are sent out to OpenPDC PI test box located in RTDS laboratory at Grayland Ave. The total estimated latency for stage 2 and 3 is ~60-70ms.

Latency 4: The C37.118 protocol used for PMU data transmission between OpenPDC and PI is derived from TCP-IP protocol, which requires handshaking and

acknowledgments after each data exchange. The following figure has shown the procedures. The Central PDC server (10.8.52.161) sent 4 packets to achieve a complete delivery, then after receiving the acknowledgment from PI data server (10.8.57.157), the OpenPDC PI test box (10.109.109.247) pushed the complete message to the PI data server, finishing one data exchange procedure. The processing time between the OpenPDC PI test box receiving data and constructing packets is estimated to be ~2.5ms.

Latency 5: The time cost between the test box sending the data and the PI server receiving the data was measured by two methods. Method 1 is to estimate by analyzing the packet exchange using Wireshark. The intervals between each acknowledgment from PI data server were ~4.0-5.0ms. Considering this time has included both the travel-from and –back time, the actual time cost was smaller. The second method is to estimate the maximum network data flow capacity with this PMU data packet size. This is done by using the "PING" command. With 6100 bytes of data, the average round trip time was computed as 4ms.

By observing this, the latency for stage 5 is estimated to be 4ms.

To sum up, the total latency between the field and the PI data server is estimated to be ~96.5-106.5ms. This time satisfies the 120ms goal for maximum latency evaluation. When the architecture of PI-PMU is finalized, the latency is expected to be reduced due to the removal of testing box and better network conditions.

4.8.4 Naming Convention

OpenPDC has the built-in customer naming convention, and the openPDC now supports custom point tag naming conventions, i.e., the format of the point tag names

automatically created by the system for device measurements can now be controlled by an expression. The expression is defined in the openPDC.exe.config file under systemSettings\PointTagNameExpression.

In this evaluation, we have utilized the Dominion naming convention standard (Version 1) to manipulate the PI tag names. The Dominion naming convention standard has defined the PI Naming Frame as:

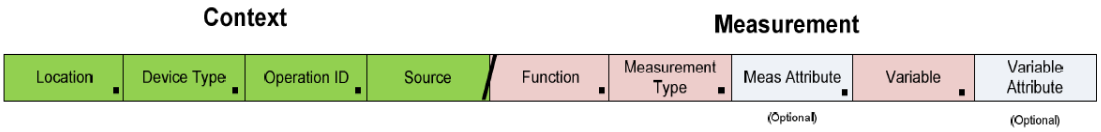


Figure 4.25 Standard PI naming frame

The following translation table is defined to map electrical quantities to standard PI names.

Table 4.3 Translation table

Electrical Quantities	***Recommended PI Names***
V1 (positive sequence voltage)	MSQI1.SeqV:c1.cVal
VA (phase A voltage)	MMXU1.PhV:phsA.cVal
VB (phase B voltage)	MMXU1.PhV:phsB.cVal
VC (phase C voltage)	MMXU1.PhV:phsC.cVal
I1 (positive sequence current)	MSQI1.SeqA:c1.cVal
IA (phase A voltage)	MMXU1.A:phsA.cVal
IB (phase B voltage)	MMXU1.A:phsB.cVal
IC (phase C voltage)	MMXU1.A:phsC.cVal
Frequency	MMXU1.Hz.mag
Rate of Frequency Change	MMXU1.HzRte.mag
Breaker Status	XCBR1.Pos.stVal

4.8.5 Compression Rate

According to OSIsoft, PI data repository with the adapters built to comply with C37.118 synchrophasor standard is capable of continuously receive and write events upon a 30/60 fps refreshing rate through an established session with the PDC server, giving the assumption that both ends run with buffer service and the buffers are capable of processing new synchrophasor events within settled time intervals. A data transmission and archiving with no exception and compression is defined as full pressure mode in this report. Though data can be processed without exception and compression, it is claimed that adding a reasonable exception and compression combination is beneficial for successive operations as visualization and analytics. Utility peers such as PJM, BPA also adopted their own rules. Hence we tested the data reception with BPA's applied rule and aimed to discover the most efficient rule for Dominion through the findings.

4.8.5.1 Verifications

In this testing, before applying the BPA compression rate settings, as shown in Table 4.4, to all the PMU channels in PI, a verification process is conducted. This verification aims to check whether the compressed data could still be good enough to reflect the dynamics of the system.

Table 4.4 BPA compression rate settings

Measurement Type	***Compression (Deviation)***
HZ	0.0001Hz
AMPS	0.100 A
DEG-AMPS	0.100 Deg
DEG-VOLTS	0.001 Deg
VOLTS	10.000 V

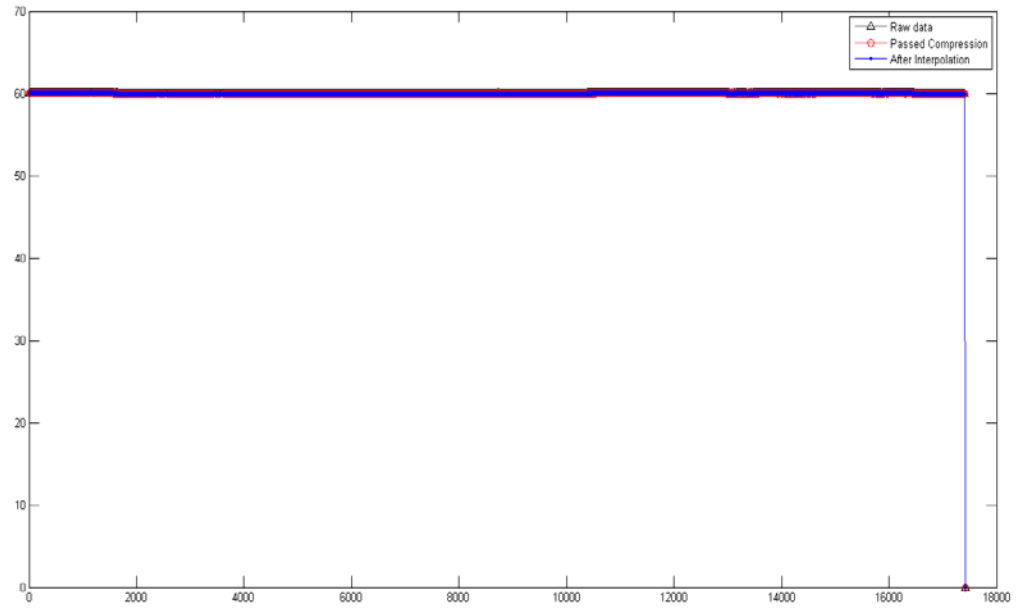


Figure 4.26 Frequency trends

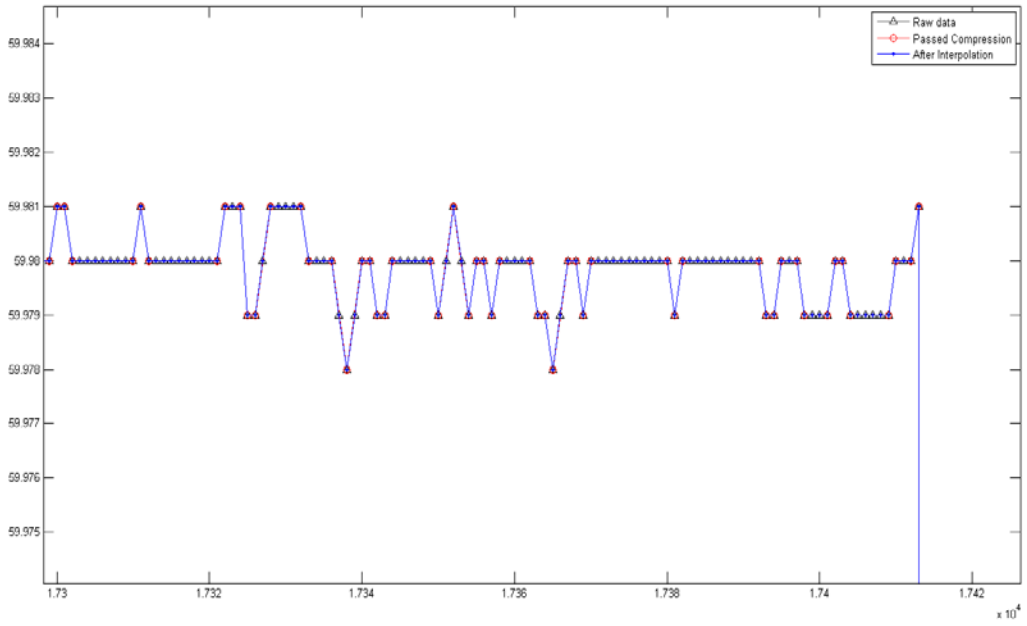


Figure 4.27 Zoomed frequency trends

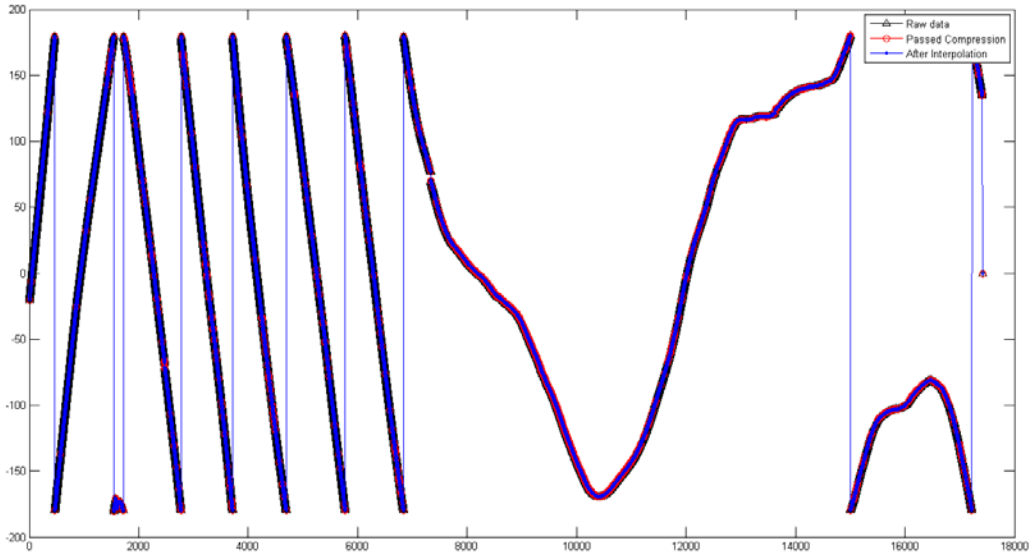


Figure 4.28 Current angle trends

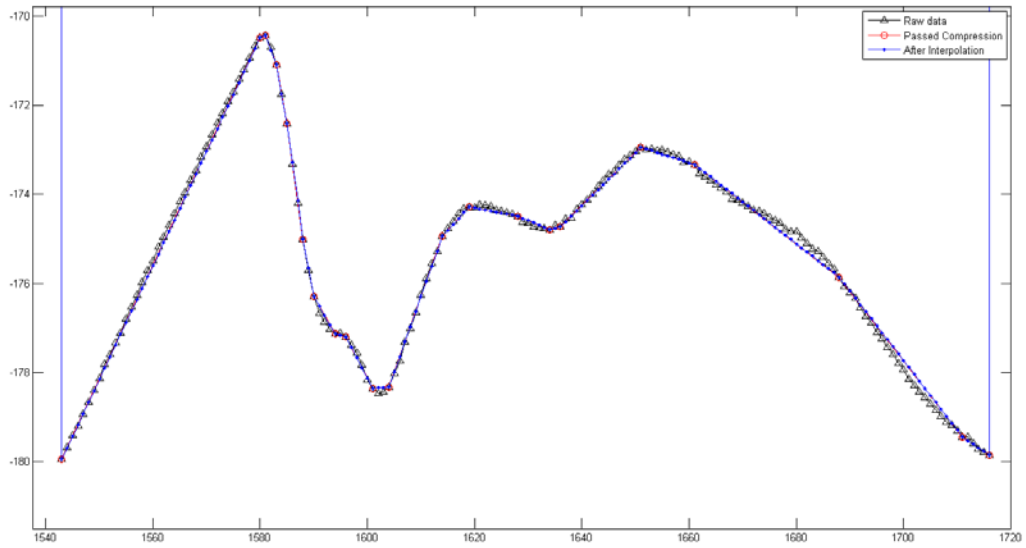


Figure 4.29 Zoomed current angle trends

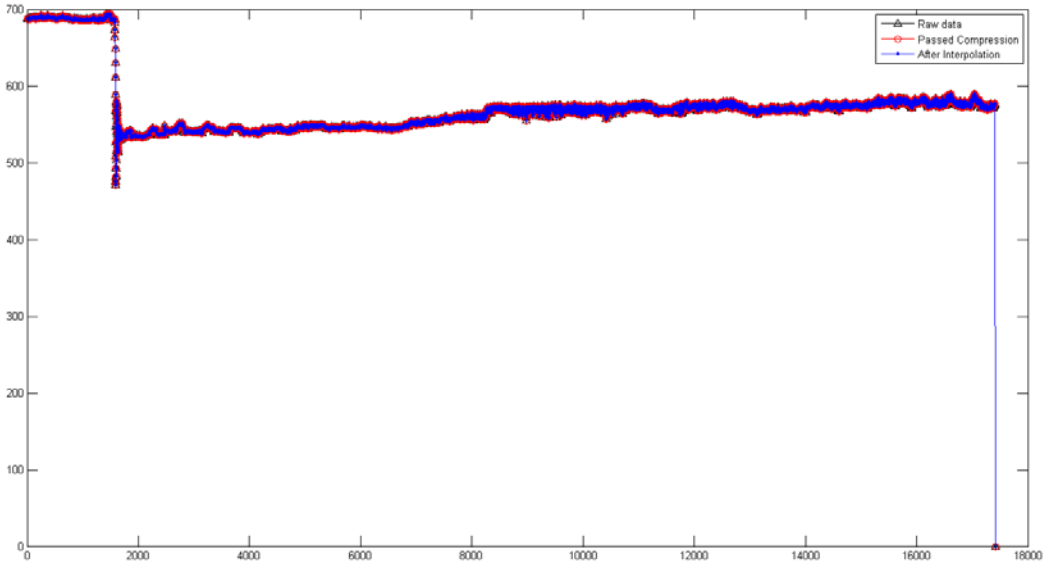


Figure 4.30 Current magnitude trends

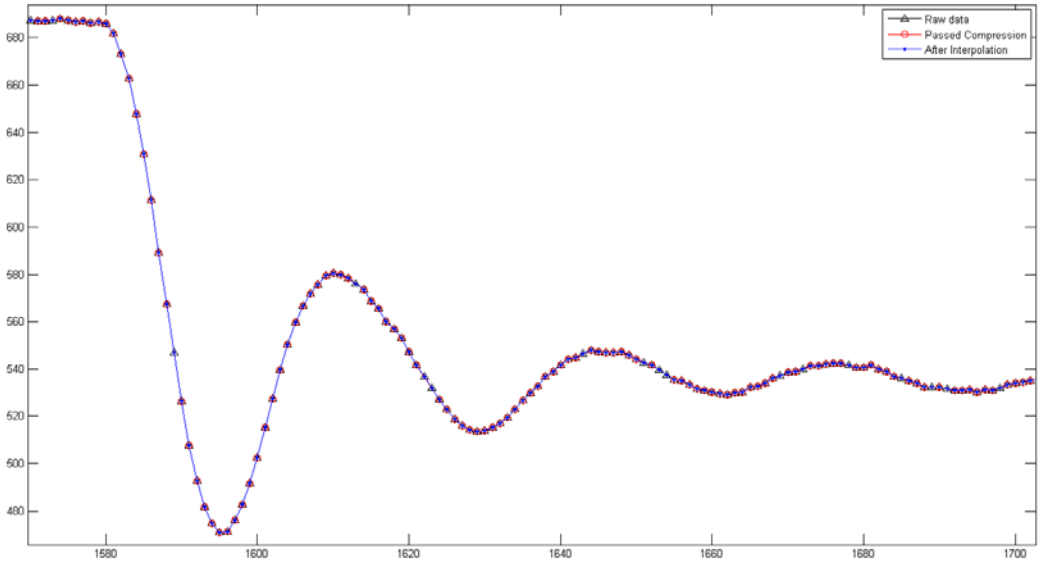


Figure 4.31 Zoomed current magnitude trends

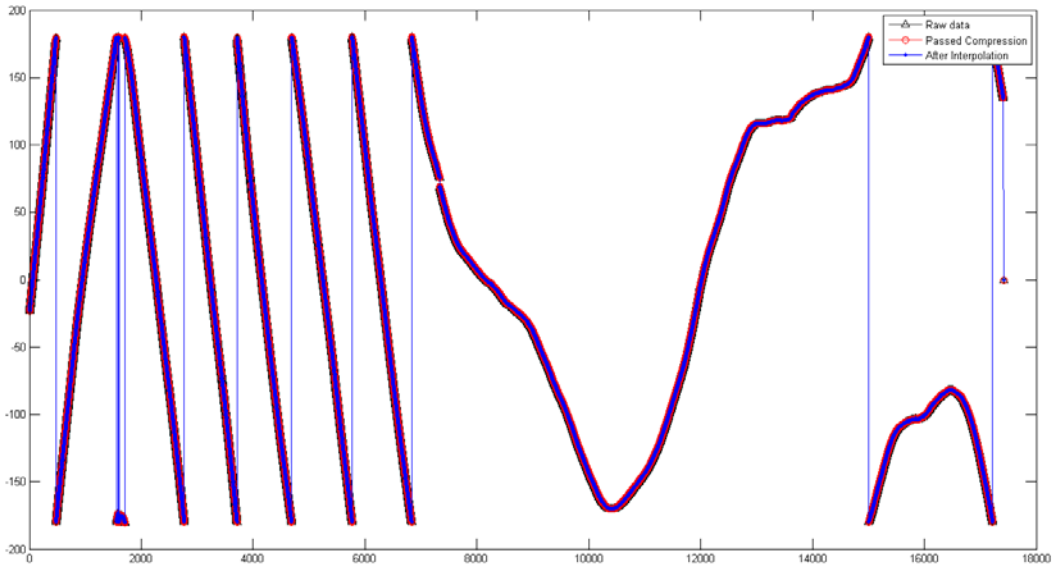


Figure 4.32 Voltage angle trends

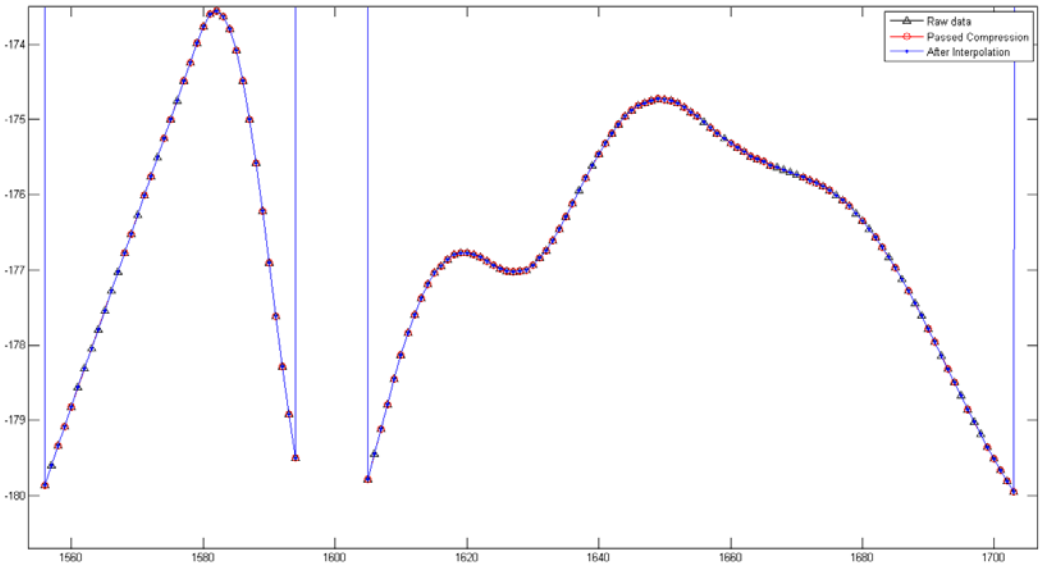


Figure 4.33 Zoomed voltage angle trends

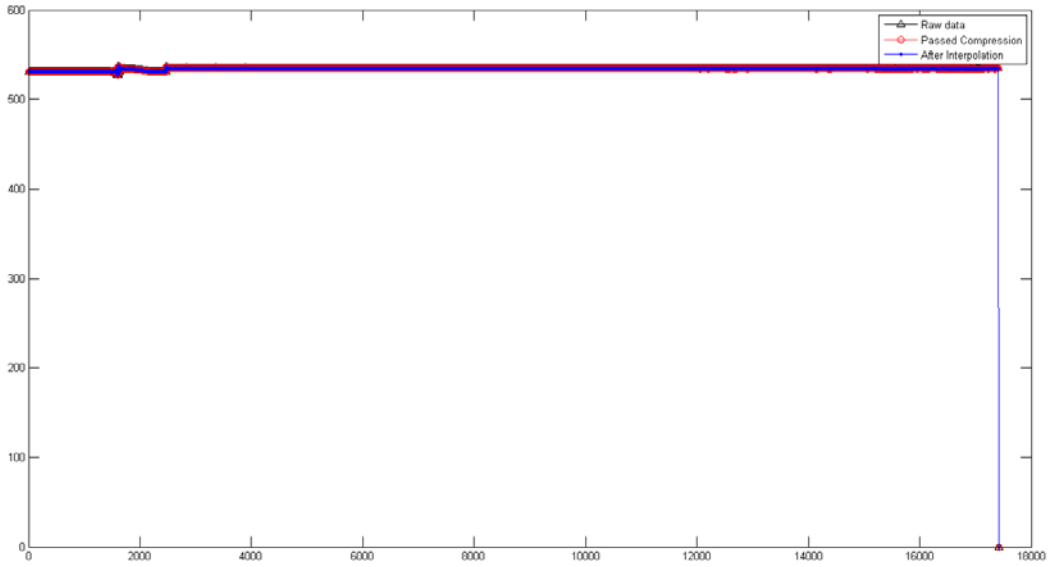


Figure 4.34 Voltage magnitude trends

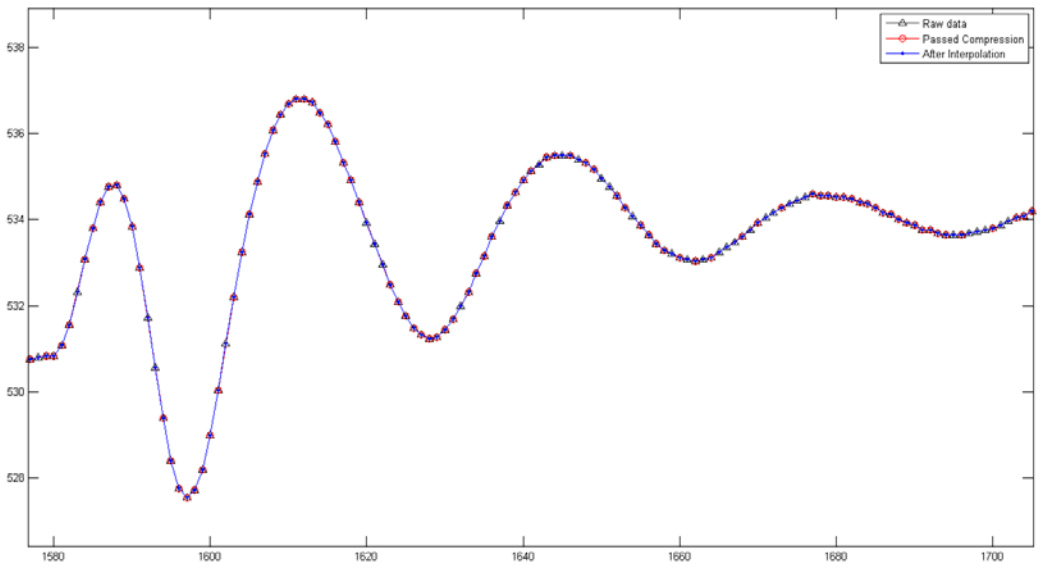


Figure 4.35 Zoomed voltage magnitude trends

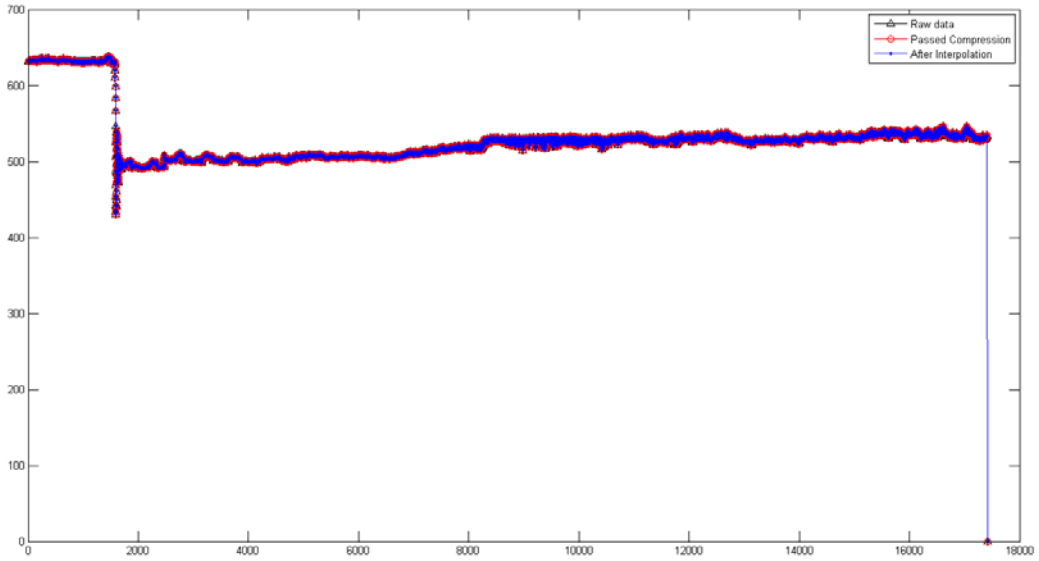


Figure 4.36 Active power trends

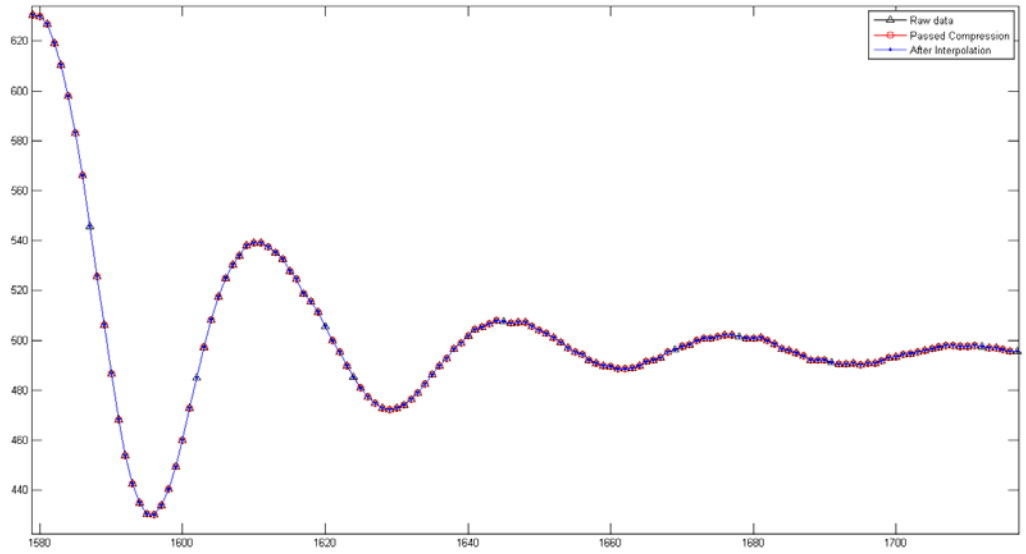


Figure 4.37 Zoomed active power trends

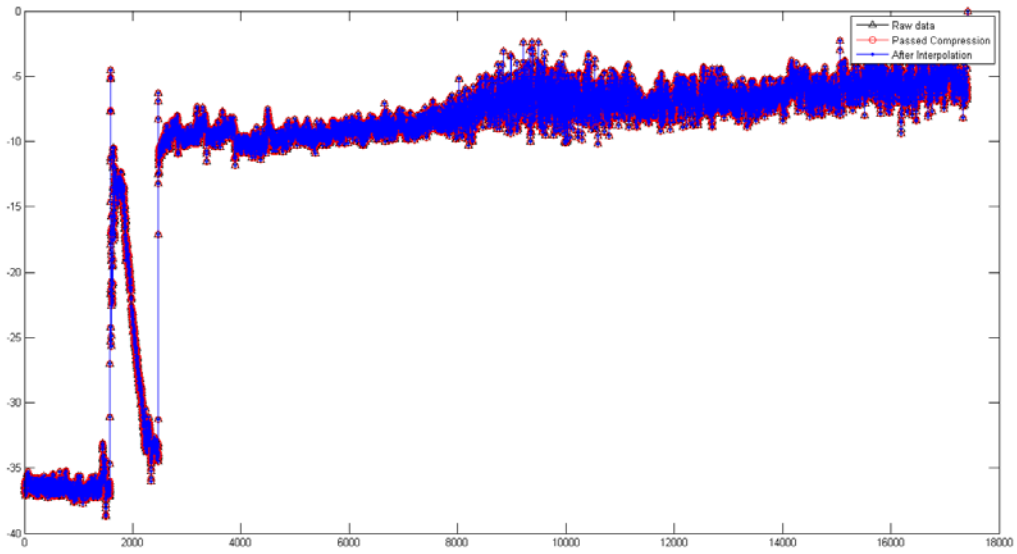


Figure 4.38 Reactive power trends

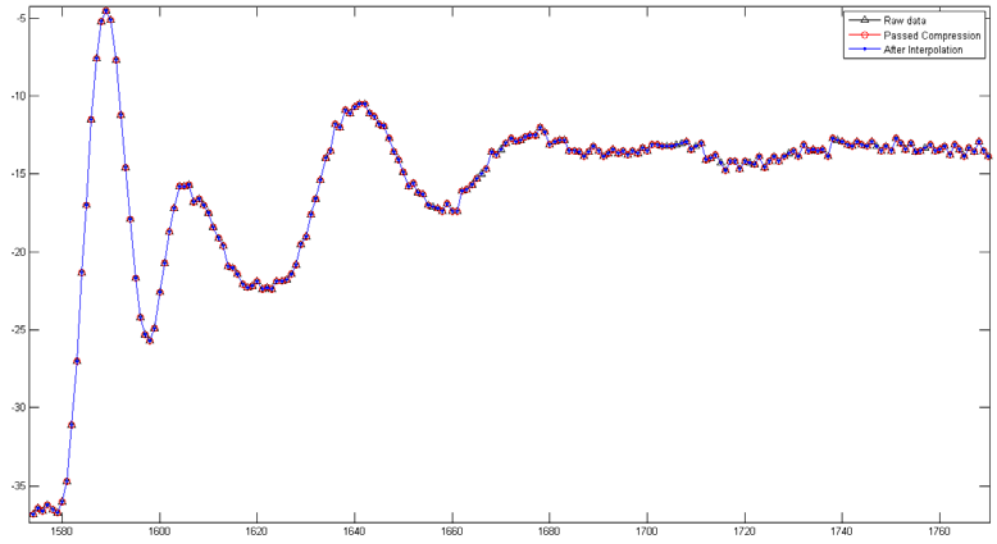


Figure 4.39 Zoomed reactive power trends

The Surry unit2 generation trip event happened at 2014/10/13 is used as a testing case, and the following figures show the frequency and voltage angle data trend before compression, after compression and interpolation. It can be seen that the measurements passed the compressing testing, which means will be archived in the PI system, could reflect the dynamics of the system status, and been easily constructed through simple interpolation.

4.8.5.2 Effectiveness

With the setting above, 1 day PMU data (03/13/2015 – 03/14/2015) from different measurement types are summarized to calculate the compression rate, the original data count is calculated as SampleRate * T = 30 * 86400 = 2592000. And the compression rate is calculated based on the following equation:

$$CompressionRate = \left(1 - \frac{DataCount_{AfterCompressed}}{OriginalDataCount}\right) \times 100\%$$

The following table shows the compression rate for the measurements in Lexington substations, one channel of each type of measurements is shown as a demonstration.

Table 4.5 Compression rate

Measurement Type	***Original Data Count***	***Data Count After Compressed***	***Compression Rate***
HZ	2592000	279	99.989%
AMPS	2592000	1461746	43.605%
DEG-AMPS	2592000	1263435	51.256%
DEG-VOLTS	2592000	1416090	45.367%
VOLTS	2592000	1200925	53.668%

It can be seen that without losing the dynamic information of the system behavior, almost half of the data could be compressed, for frequency measurement, most of the data are compressed since the frequency tends to be constant within a small range under normal operation. Table 4.5 shows clearly the effective of the compressing function in terms of saving data storage.

4.9 Conclusion

Through the ongoing project at DVP, this paper presents the entire solution for integrating, managing and analyzing big data in the power industry. The ultimate target is to build a highly reliable and flexible common data platform with the features of scalability, real time, high reliability and security within the enterprise. It is foreseeable for DVP for the business intelligence and management capability to be improved and reinforced significantly in the future. Meanwhile, it is convenient and efficient for individuals to initialize self-customized applications to discover and analyze the issues with fewer restrictions under permissions. Furthermore, this paper may provide a good instance for peers who intend to enhance the capability of managing and utilizing big data so as to improve business intelligence.

Chapter 5. Solar Output Prediction

5.1 Motivation

Due to its economic and environmental benefits to society and industry, integrating solar panels is continuously growing in many utilities and Independent System Operators (ISO). The intermittent nature of the renewable energy brings new challenges to the system operators, and one key to resolve this problem is to have a ubiquitously efficient solar power forecasting system, so as to help enhance system reliability, improve power quality, achieve better generation scheduling and formulate bidding strategy in market. This paper proposed an ensemble learning method to forecast solar power output, combining the effectiveness of state-of-art machine learning technology, namely, multiple linear regression, neural network and random forest. The performance of the model is evaluated through different metrics, and the numerical results validate the effectiveness of the model.

It is reported that there are now over 22,700 MW of cumulative solar electric capacity operating in the U.S., enough to power more than 4.6 million average American homes [76], and the number is continuously growing due to its economic and environmental benefits to society and industry. Integrating renewable is undergoing in many utilities and Independent System Operators (ISO) [77] on transmission and distribution level [78]. However, the intermittent nature of the renewable energy brings new challenges to the system operators, especially with high penetration level, since there may be a chance for reverse power flow into the distribution system or energy mismatch. There are some research focusing on management and mitigation strategies using energy

storage [79], advanced control method [80], demand response . However, it is still essential to integrate a ubiquitously efficient forecasting technology in order to help enhance system reliability, improve power quality, achieve better generation scheduling and formulate bidding strategy in market.

Solar power output mainly depends on solar irradiation, which heavily relies on cloud cover, water vapor, air pollution, etc., besides, solar power output is also influenced by the panel manufacture, ambient temperature, relative humidity, etc. [80]. With advanced sensors and information technology enabled in smart grid, the weather information can be acquired through temperature sensor, wind sensor, and solar irradiation data can be measured through pyranmeter sensor, and these information, along with historical dataset can be utilized to formulate an efficient forecasting model.

5.2 Literature Review

Numerous algorithms have been applied to forecast solar power output. In [81], a statistical time series model is proposed, [82] assessed a short-term forecast model based on historical similarity. [82] utilizes numerical weather prediction (NWP) information through different statistical learning methods, such as Auto-Regressive Integrated Moving Average (ARIMA), k-nearest neighbors (KNN), and adaptive neuro-fuzzy. [83]–[85] assess the problem from a probabilistic forecasting point of view, utilizing vector auto regression framework (VAR), weighted quantile regression and Markov chain respectively. The proposed method in [86] combines a self-organizing map (SOM), a learning vector quantization (LVQ) network, the SVR method, and the fuzzy inference approach to make 1-day ahead hourly forecasting. Several regression models, e.g.

boosting and random forest (RF) are reviewed in [87]. Overall, Neural network (NN) is a proved effective model for forecasting, [88] presents a back propagation (BP) NN considering aerosol index to improve the forecasting accuracy, [89] develops a hybrid particle swarm optimization (PSO) NN model to outperform the solo BP-NN, and concludes the sunshine duration and air temperature are key factors. Other machine learning algorithms are also well investigated. In [90], a support vector machine (SVM) with firefly algorithm (FFA) is proposed to predict monthly mean horizontal global solar irradiation. Paper [91] evaluates a Coral Reefs Optimization - Extreme Learning Machine (CRO-ELM) algorithm to predict the global solar radiation using different meteorological data. An extended overview for the solar power forecast literature can be found in [92].

5.3 Algorithm

This paper proposed an ensemble forecasting model combining multiple linear regression (MLR), NN and RF, utilizing weather and solar irradiation dataset, and the forecast performance is assessed through different metrics.

5.3.1 Characteristics of Solar Power Output

It is well investigated that the solar power output has strong relationship with weather conditions, such as solar irradiation, temperature, humility, pressure and solar angle. In terms of the influence of temperature and solar irradiation, it can be estimated as shown in (7-1) [86].

$$P = \left[P_{MPP} \times \frac{SI}{1000} \times [1 - \gamma \times (T_j - 25)] \right] \times N_s \times N_p \tag{5-1}$$

where PMPP is the output of PV array at the maximum power point (MPP), N_s is the number of PV arrays in series, N_p is the number of PV arrays in parallel, SI is the solar irradiation at standard test conditions (STC), is a temperature parameter at MPP, T_j is the temperature of solar panel. However, some of the parameters of this mathematical model is tested at STC, and there are other parameters, like humility, have influence on the PV output, thus this model is not adequate for accurately forecasting the PV power in practice. Figure 5.1 shows the characteristic of daily PV power.

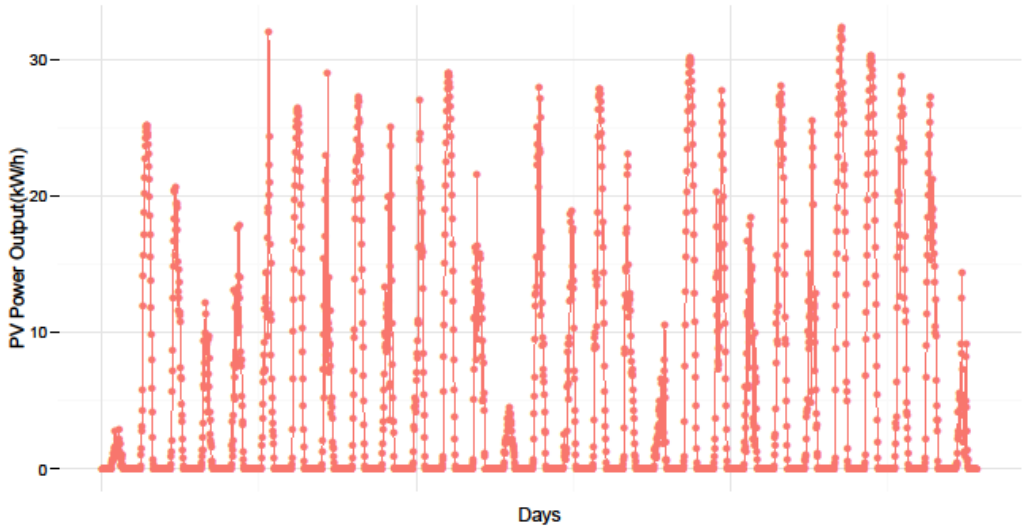


Figure 5.1 Daily solar power output in June, 2012

5.3.2 Ensemble Model

5.3.2.1 Multiple Linear Regression Model

Multiple linear regression (MLR) is an extension of the simple linear regression which is generally used to analyze the relationship of multiple independent variables $x_1, x_2, \cdots, x_k$ with the dependent variable p. It fits the dataset to the following model

$$P = \beta X \tag{5-2}$$

Where

$$p = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix},\ X = \begin{bmatrix} 1 & x_{11} & x_{12} \cdots & x_{1k} \\ 1 & x_{21} & x_{22} \cdots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots \ \vdots \\ 1 & x_{n1} & x_{n2} \cdots & x_{nk} \end{bmatrix},\ \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix},\ \varepsilon = \begin{bmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_k \end{bmatrix} \tag{5-3}$$

And X_i represents the *j*th predictor and β_j quantifies the association between that variable and the response. Then the parameters can be estimated using the least square approach, which chooses β^T to minimize the sum of squared residue. Assuming ε_i are i.i.d with $\mathrm{E}[\varepsilon] = 0$ and $Var[\varepsilon] = \sigma^2$, thus with minimizing

$$RSS = \sum_{i=1}^{n} (p_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_m x_{im})^2 \tag{5-4}$$

The least square estimate could be obtained as

$$\beta = [\beta_0, \beta_1, \cdots, \beta_k]^T = (X^T X)^{-1} X^T p \tag{5-5}$$

Then the prediction can be made by

$$P_{MLR} = \beta X \tag{5-6}$$

5.3.2.2 Neural Networks Model

Neural Network performs a computation simulation of the behavior of neuron in human brain to formulate model based on the learning of training data. It is typically represented by a network diagram, which consists of three layers: input, output, and hidden layer.

The input variable is $X = [x_1, x_2, \cdots, x_n]^T$, hidden layer unit is $H = [h_1, h_2, \cdots, h_m]^T$, and output variable is $P = [p_1, p_2, \cdots, p_n]^T$. The weight connecting input and hidden layer units $V = [v_1, v_2, \cdots, v_n]^T$, the weight connected between hidden and output layer units is $W = [w_1, w_2, \cdots, w_n]^T$. Then it can be formulated that

$$p_k = f(\sum_{j=1}^{m} w_{jk} h_j + b), k = 1, 2, \cdots, n$$
$$h_j = f(\sum_{j=1}^{n} v_{ij} x_j + b), j = 1, 2, \cdots, m \quad (5\text{-}7)$$

The transfer function is chosen as the widely used sigmoid function

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5\text{-}8)$$

in the implementation. And through minimizing the cost function (8), the model can be obtained using gradient decent, and calculating the prediction PNN. The details of derivation can be found in

$$J = \frac{1}{2} \sum_{k=1}^{n} (\hat{p}_k - p_k)^2 \quad (5\text{-}9)$$

5.3.2.3 Random Forest Model

Generally, tree-based methods which grow very deep tends to overfit the training data since they have low bias and high variance. However, random forest builds a large collection of trees, and then average them so as to reduce the variance and boost the performance of the final model. Given a training set $X = [x_{i1}, x_{i2}, \cdots, x_{in}]^T$, and response $P = [p_1, p_2, \cdots, p_n]^T$, the way to construct a random forest is to bag repeatedly B

times, each time to draw a bootstrap sample of size N from the training data. Then a random forest tree T_b can grow to the bootstrapped data. Recursively repeating the following steps for each terminal node of the tree, until the minimum node size $n_{\min}$ is reached [93]. 1. Select m variables at random from the p variables. 2. Pick the best variable among the m. 3. Split the node into two daughter nodes. Then output the ensemble of trees $\{T_b\}_1^B$, to make a prediction

$$P_{RF}^B = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \tag{5-10}$$

With obtaining the prediction above, an ensemble model can be obtained by the weighted sum of the previous models.

$$\hat{p} = \sum_{k=1}^{N} w_k p_k, \sum_{k=1}^{N} w_k = 1 \tag{5-11}$$

Empirically, the ensemble model tends to yield better results through mitigating over-fitting or under-fitting. The structure of the ensemble model and its forecasting strategy is shown in Figure 5.2.

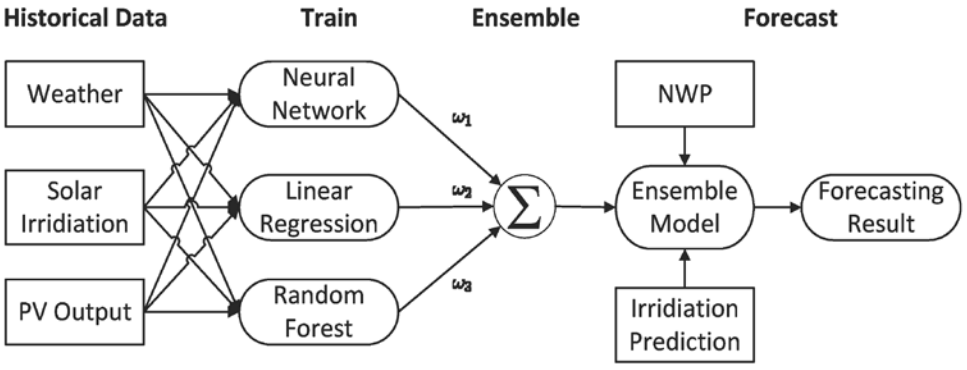


Figure 5.2 Schematic diagram of the ensemble model

The historical solar irradiation data are collected from the NREL solar database for year 2012 [94]. Three types of solar irradiation data are provided, namely Direct Normal Irradiance (DNI) and Diffuse Horizontal Irradiation (DHI) and Global Horizontal Irradiation (GHI). DNI is solar radiation that comes in a straight line from the direction of the sun at its current position in the sky. DHI is solar radiation that does not arrive on a direct path from the sun, but has been scattered by molecules and particles in the atmosphere and comes equally from all directions. GHI is the total amount of short wave radiation received from above by a surface horizontal to the ground. Also, numerical weather information is also included in the data archive, Table 5.1 lists some of the key parameters and respective unit of measure, and a statistically analysis is shown in Figure 5.3 and Figure 5.4.

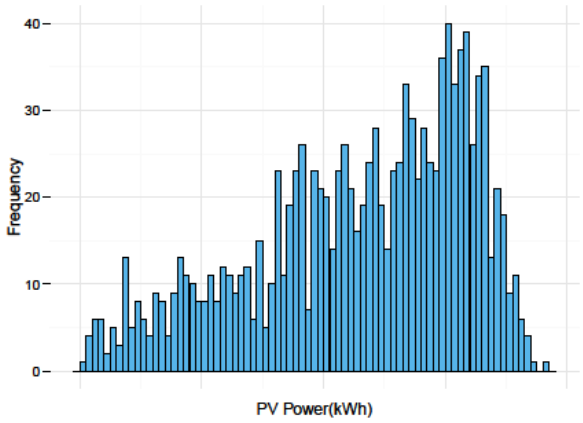


Figure 5.3 Histogram of solar power output during one month

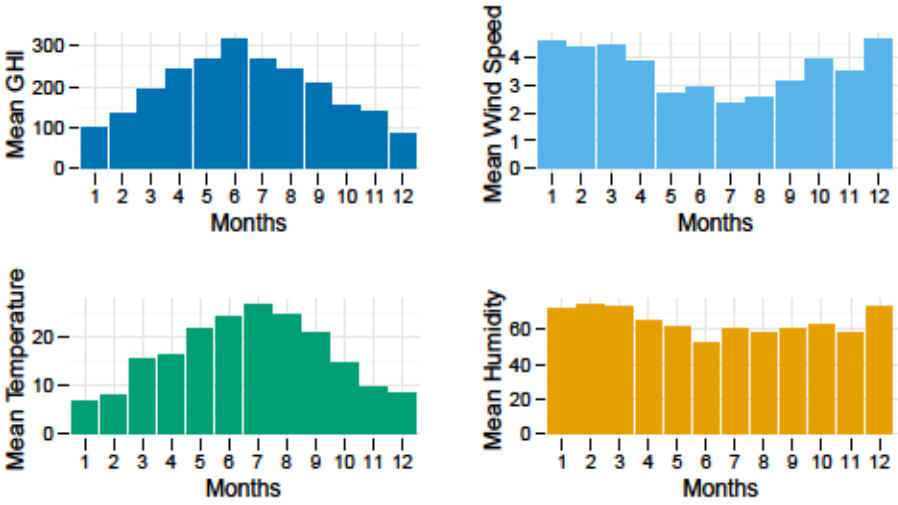


Figure 5.4 Monthly mean value for some parameters

Table 5.1 Parameters description

Parameters	***Description***
DHI	W/m^2
DNI	W/m^2
GHI	W/m^2
Solar Zenith Angle	°
Temperature	°C
Pressure	millibar
Relative Humidity	%
Wind Speed	m/s

The data preprocessing aims to improve predictive accuracy and not allow a particular feature impact the prediction due to large numeric value ranges. Thus, normalization or scaling values under different features can be performed using the following method.

$$\overline{x} = \frac{x_i - x_{\min}}{x_i - x_{\max}} \tag{5-12}$$

where x_{i} is the *i*-th component in the data vector, and $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the parameter. Also, it is necessary to exclude the uncorrelated parameters in the feature selection process. In this work, the MLR is first utilized to judge the significance of all the parameters, and eliminate those with lower significance.

5.3.3 Training and Validation

The training dataset is selected as the first three weeks of each month, and then filtered out the time periods when there is no solar irradiation. The last week data of each month is prepared as testing dataset to evaluate the performance of the model.

In validation process, k-fold cross validation is performed to tune the parameters. The general idea is to divide the training data into k subsets. Each time one of the k subsets is used as the test set and the other $k-1$ subsets are put together to form a training set. Then the average error across all k trials is computed to determine the optimal parameters. k is selected as 5 in this experiment. This validation step could make the model more generalized to new data inputs.

5.3.4 Forecasting and Performance Metrics

Proper performance metrics are necessary to evaluate the forecast accuracy or uncertainty. However, there is still not an industry-wide standard metrics available. Several well accepted metrics are summarized in [77], such as, Pearson's correlation coefficient, root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), mean bias error (MBE), standard deviation, kurtosis, skewness, Kolmogorov–Smirnov test Integral (KSI), and the integrated differences between cumulative distribution functions. All of these metrics could assess the performance of the forecasting model. In this paper, RMSE, MBE and MAE are utilized, and their definitions are illustrated below.

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\hat{p}_t - p_t)^2}$$

$$MBE = \sum_{t=1}^{N}(\frac{\hat{p}_t - p_t}{N}) \tag{5-13}$$

$$MAE = \sum_{t=1}^{N}\left|\frac{\hat{p}_t - p_t}{N}\right|$$

5.3.4.1 Benchmark

The benchmark of the forecasting performance is obtained through a persist model, which forecasts day-ahead solar power output using current day's data. The model can be presented as

$$\hat{p}_t = p_{t-1} \tag{5-14}$$

5.3.4.2 Day-ahead Forecast

The solar power output in the last week of each month is used to evaluate the performance of the proposed model. For the day-ahead forecast, the data in the first day of the week are used, in Table 5.2, the performance metric for the day-ahead forecast is calculated and presented. Figure 5.5 shows the comparison of the predicted power and actual power for a day in September. As shown in Figure 5.5, the proposed ensemble model could capture "averaged" power output through learning from the historical data to provide a reasonable result. If the model wants to handle the ramp situation more accurately, the sample rate of historical measurement data should be increased, so as to provide enough pattern information and transition time for the model to learn. According to the performance metrics, the largest RMSE is 7.98% and 10.27% for day-ahead and week-ahead, while the smallest RMSE is 1.00% and 4.42%, respectively. It indicates that the proposed model could forecast the solar power output at a reasonable accuracy level. The metrics MBE and MAE also validate the effectiveness of the model.

5.4 Conclusion

This chapter proposed an ensemble model which is a weighted sum of multiple linear regression, NN, and RF models, and the model is trained using historical weather and solar irradiation dataset with k-fold cross validation. Day-ahead and week-ahead forecasts are conducted and the performance metrics are calculated to access the accuracy of the model. Future work would focus on integrating time series model, such as auto regressive moving average model with exogenous inputs model (ARMAX) to boost the forecast performance in intra-day or hour-ahead forecast.

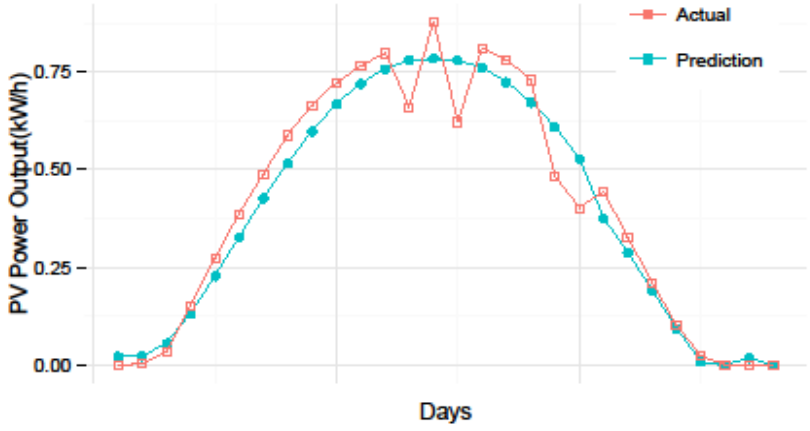


Figure 5.5 Day-ahead forecasting and actual solar power output

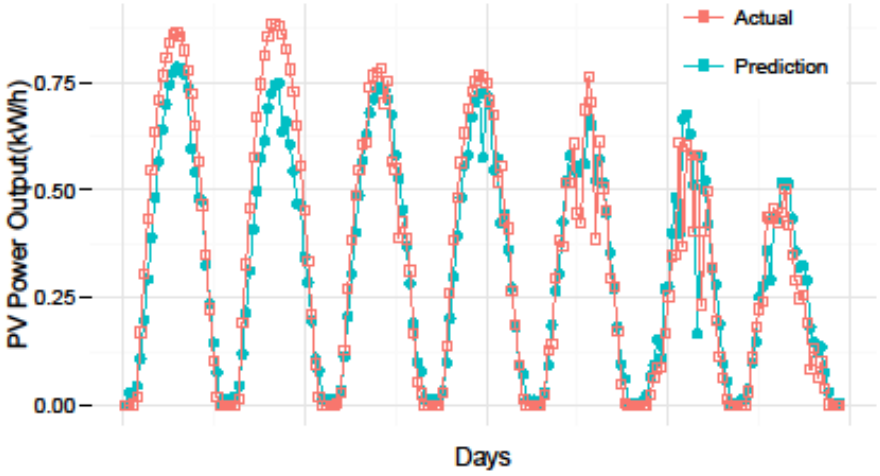


Figure 5.6 Week-ahead forecasting and actual solar power output

Table 5.2 Day ahead forecasting performance metrics

Index	Jan	Feb	Mar	Apr	May	Jun
RMSE	1.82	1.61	7.98	5.33	2.86	2.24
MBE	-0.27	-0.28	-0.40	-0.30	0.23	-0.12
MAE	0.58	0.55	6.23	1.75	0.93	0.60
Index	Jul	Aug	Sep	Oct	Nov	Dec
RMSE	2.80	2.17	4.13	6.93	2.56	1.00
MBE	0.22	0.52	0.63	-0.01	0.23	-0.15
MAE	0.84	0.60	1.17	0.11	0.80	0.33

Table 5.3 Week ahead forecasting performance metrics

Index	Jan	Feb	Mar	Apr	May	Jun
RMSE	4.42	6.31	7.98	10.8	9.10	6.38
MBE	-0.90	0.01	-0.40	-1.15	-0.41	-0.56
MAE	3.14	5.28	6.23	8.88	7.25	3.94
Index	Jul	Aug	Sep	Oct	Nov	Dec
RMSE	9.81	7.53	10.3	7.74	7.21	8.72
MBE	0.04	0.43	-0.30	0.32	1.85	1.50
MAE	7.34	5.34	8.14	6.40	6.18	7.47

Chapter 6. Big Data Analytics Platform

6.1 Motivation

With the increasing deployment of PMUs and the maturity of ICT (information and communication technology), it becomes possible to have insight into the power system dynamics by analyzing the real time data stream and large amount of archived data. FNET/GridEye, as a nationally deployed Wide Area Measurement System (WAMS), has collected Terabytes synchrophasor data and these data hold invaluable potential in situational awareness supporting system operation. Large volume of data are streamed into a central data server in real-time. However, unless the server system is designed with the capability to efficiently process and analysis the data, system operators cannot exploit the information hidden in the data to assist the system operation or control. Thus, how to effectively analyze the historical and real time data becomes a challenge.

6.2 Literature Review

Big data technologies are increasingly boosting the performance to handle data with large volume, high velocity and variety [95]. Introduction of the technologies like Hadoop and Spark could bring the smart grid data analytics to a new era, in which comprehensive analysis algorithms can be applied in real-time, and close-loop data solutions can be implemented for situation awareness, asset management, planning, fault detection and protection [69]. Some works has been done to study the availability using big data technologies for grid data storage [96], data processing [97], and data analysis [71], [98], [99]. However, most of these studies are based on simulation or offline data.

More challenges have to be solved to build an infrastructure for real-time data collection, analysis and visualization [100].

6.3 System Architecture

To solve the challenges brought by the real-time phasor applications, as well as to ensure the system scalability, extensibility, concurrency and robustness, a distributed computing architecture is proposed to host the server system and data analytics platform, as shown in Figure 6.1.

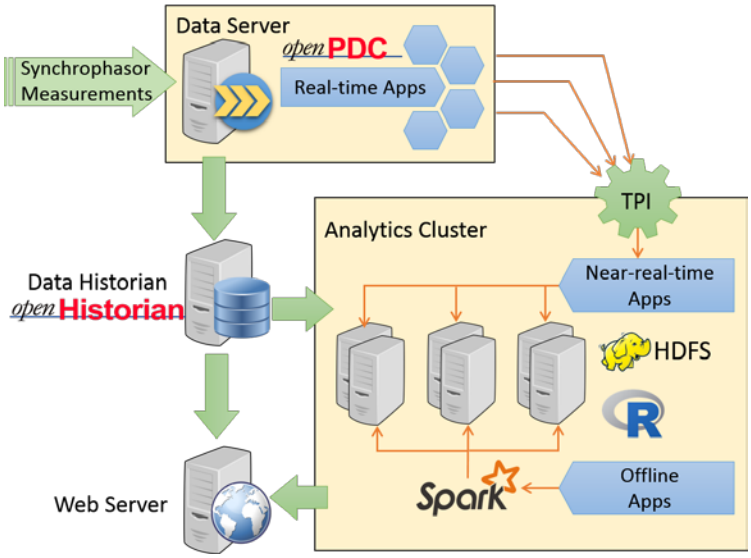


Figure 6.1 Distributed data analytics system architecture

The proposed architecture consists of a data server, a data historian [101], a web server, and a computing cluster for data analytics. The synchrophasor data are collected by the data server through internet. The real-time phasor data are concentrated and aligned with timestamps and dumped to historian. Also, the phasor data are scanned by

the real-time triggers, which detect disturbances and send trigger messages to the analytics clusters.

The analytics cluster contains a series of computation nodes. Whenever the analytics cluster receives the trigger message, the trigger processing interface (TPI) finds its corresponding applications and dispatch them to the computation nodes. Also, the TPI is responsible for loading the data from historian according to the requested time range. The applications are distributed into different nodes based on their computation complexity, so that the computation load is balanced. For offline applications, the analytics cluster can be reused if there is no application being processed or waiting in queue. Multiple software packages, including Spark [102], Hadoop [103] and R [104], are hosted on the cluster to support big data analytics. The web server requests down-sampled data from the historian for real-time display, as well as publish disturbance analysis reports whenever it receives the processing results from analytics cluster.

6.3.1 Data Concentrator

The FNET/GridEye data server plays the role as a data concentrator that collects data from hundreds of FDRs around the world. Synchrophasor data are transferred to the server through TCP connections. Since the FDRs are developed as distributed level PMUs, the FDR data format is designed as a simplified version of IEEE C37.118 standard format [105], which only contains timestamp, frequency, angle, voltage information measured at the distribution level, as well as the GPS coordinates of each FDR unit. The openPDC developed by Grid Protection Alliance (GPA) is tailored to adopt FNET/GridEye data stream format, and serves as the server-side concentrator

software for FNET/GridEye system [44]. Once a connection is built, the data server continuously receives the data packet, parses data to structured format, and verifies data validity.

6.3.2 Data Storage

The capability to write and read large volume of data with high speed is critical for the data historian. After exploration of a several relational databases, including MS Access and MySQL, and a variety of NoSQL databases, including MongoDB [106] and Cassandra [107], the openHistorian 2.0 developed by GPA is selected to fulfill the requirements. The openHistorian 2.0 is a file based storage system designed to efficiently integrate and archive SCADA, synchrophasor, Digital Fault Recorder (DFR) and other process control data to support real-time grid operations and post-disturbance analysis. It supports indexed data retrieval interface and lossless data compression, which largely saves the storage capacity of the historian. The archive files produced by the openHistorian are ACID Compliant, which create a very durable and consistent file structure that is resistant to data corruption. Internally the data structure is based on a B+ Tree that allows out-of-order data insertion. It also provides high-speed APIs that can be customized for visualization of real-time and historical data, web-based data access and remote historian data extraction.

6.3.3 System Prototype in Virtual Machines

The prototype system is setup in FNET data server, which has 20 cores hyper-threading, 128 GB RAM, and 15TB Hard Drive, which is capable of archiving all the historical FNET data dating back to 2004. The system is built upon virtual machines to

emulate a data cluster, and the data are loaded into HDFS with Spark build upon. And could provide various APIs for computation. Figure 6.2 shows the system prototype.

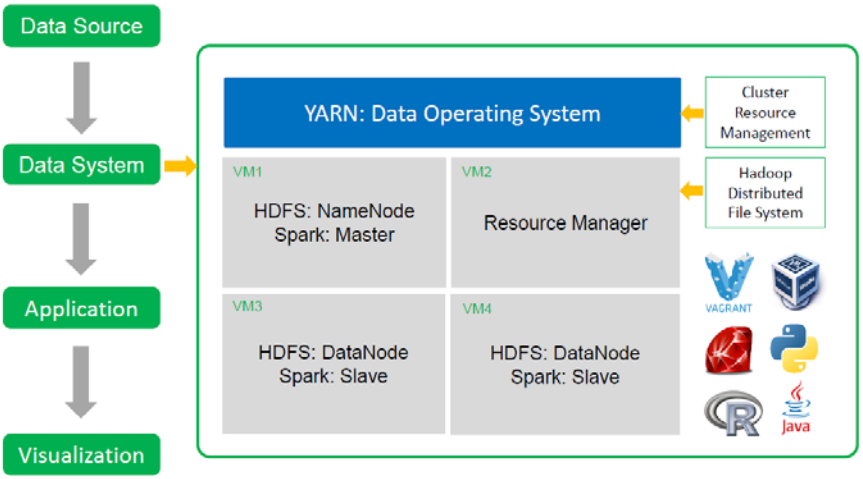


Figure 6.2 System prototype in virtual machines

6.4 Analytics

A variety of real-time and near-real-time analytics applications hosted by the FNET/GridEye system were published since 2006. The offline applications were recently migrated to the cluster analytics platform to fully utilize the big data technologies. A sample statistical analysis using Spark is demonstrated.

The frequency disturbance event trigger detects frequency deviation caused by generation trip or load shedding. All the triggered events are logged with a timestamp and the raw event data are stored in the data historian. North American Electric Reliability Corporation (NERC) utilize a system frequency for frequency response analysis [108]. The system frequency is calculated using the median value at each sampling time from the event data including 1 minute pre-event and 5 minutes post-event frequency. From

2012 to 2015, tens of thousands of events are captured, which makes the data extraction and conversion a challenging job.

Assuming 10000 events are to be analyzed with 100 FDR data, and the sampling frequency of FDRs is 10 per second, the total data size is 10000* 100 FDRs *10 per second * 360 second * 4bytes = 24 Gigabyte. At each time point of 0.1 second, the frequency value of 100 FDRs are to be sorted to obtain the median point.

The distributed data analytics platform with Spark provides a solution to process such amount of data in parallel. Each of the frequency data and is according timestamp are packed as key-value pairs, and loaded as Spark RDDs. The RDD will first map the data partitions to computation nodes for processing, and let each nodes execute the sorting function in parallel. Then the calculated median value are sorted by their timestamps, to form a frequency time series for each event.

Eventually, the time series are grouped by their event id and are aggregated as final result to the master node. Unlike traditional MapReduce method, Spark handles the whole process in memory, which ensures the computation speed to be significantly faster. The test results show that for 100 events, the least computation time can is about 128 millisecond, while without the Spark implementation, the computation time is 2592 millisecond, which is over 20 times slower.

Increasing the number of partitions of the dataset may improve the performance by processing data in a more parallel cluster. However, the communication overhead to map and collect the dataset will also increase.

Figure 6.3 demonstrates the computation time using different partitions of the dataset. In this case, the least computation time is achieved with 8 partitions. Using more than 8 partitions will increase the computation time because of communication.

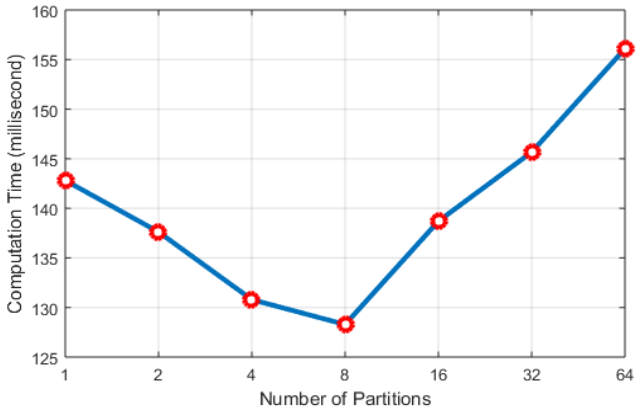


Figure 6.3 Computation time vs. number of partitions

To count the events happened by their hour of the day in each month, the timestamps of events in Eastern Interconnection (EI) from July 2012 to July 2015 are stored in a Python list structure. The timestamps are then converted to a Spark RDD, and grouped by their month and hour segment. The number events with the same hour within each month is counted, and the results is plotted as a colored contour map in Figure 6.4, with the X axis as the months, Y axis as the hours of the day (UTC time), and the color representing the total number of events counted within the time range. Since most of the events are congregated at 7:00 to 14:00 UTC, it is suspected that some scheduled operation is one of the major causes of events. To prove this hypothesis, generation trip events and load shedding events are plotted separately in Figure 8.5.

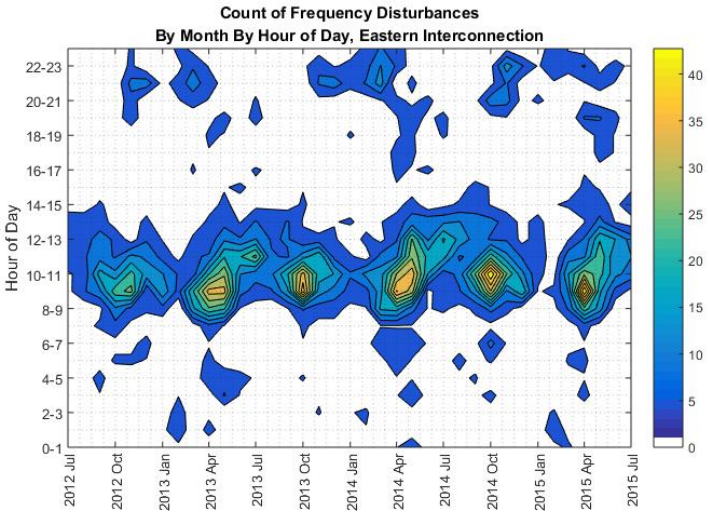
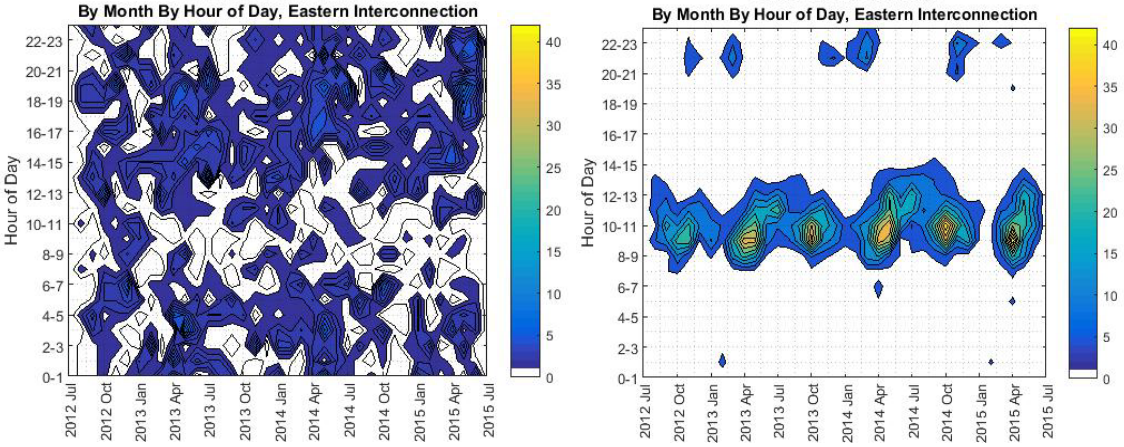


Figure 6.4 Count of EI events by month by hour of the day



a) Count of Generator Trip b) Count of Pump Storage Shutdown

Figure 6.5 Count of EI event by month by hour by day

The generation trip events are random distributed over a day, and there is no obvious pattern over a year. However, the load shedding events are concentrated around 9:00 and 21:00 UTC (4:00 and 16:00 EST). It confirms that most of the events are scheduled pump storage shut down.

To further compare events between Eastern and Western Interconnections, the events data in WECC is analyzed with the same procedure. Similarly, the generation trip events are randomly distributed. However, most the load shedding events in WECC are laid between 10:00 to 17:00 UTC, as shown in Figure 6.6.

It implies that the utilities in WECC may have one scheduled time per day for load shedding or storage shut down. The peak positions of load shedding indicate that these operations are usually 2 to 3 hours later than the load shedding in EI, possibly caused by the local time difference.

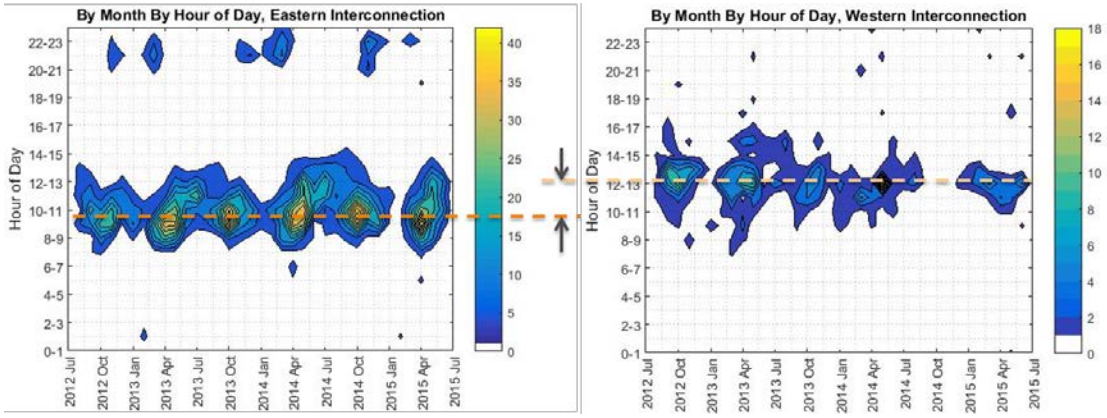


Figure 6.6 Count of pumped storage shutdown in EI and WECC

6.5 Conclusion

The distributed data analytics platform proposed in this paper extends the capability to handle large volume and high velocity synchrophasor data in WAMS. Real-time applications can be implemented on the data server; near-real-time and offline applications are hosted on the analytics cluster. The computation power is enhanced by parallelizing analysis algorithms, distributing the computation load across multiple nodes, and utilizing big data analysis tools like Spark. Large volume historian data are easily accessible to for applications and external clients. Based on this platform, more sophisticated data analytics algorithms and visualizations tools can be easily implemented for power system monitoring, operation and control.

6.6 Future Work

The future work would focus on implement more data analytic application using the data platform and make full advantage of the huge amount of historical data through data mining techniques.

List of References

[1] Y. Liu, L. Zhan, Y. Zhang, P. Markham, D. Zhou, J. Guo, Y. Lei, G. Kou, W. Yao, J. Chai, and others, “Wide-Area Measurement System Development at the Distribution Level: an FNET/GridEye Example,” *Power Deliv. IEEE Trans.*, vol. 31, no. 2, pp. 721–731, 2016.

[2] Y. Zhang, P. Markham, T. Xia, L. Chen, Y. Ye, Z. Wu, Z. Yuan, L. Wang, J. Bank, J. Burgett, R. W. Conners, and Y. Liu, “Wide-Area Frequency Monitoring Network (FNET) Architecture and Applications,” *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 159–167, Sep. 2010.

[3] Z. Zhong, C. Xu, B. J. Billian, L. Zhang, S.-J. S. Tsai, R. W. Conners, V. A. Centeno, A. G. Phadke, and Y. Liu, “Power System Frequency Monitoring Network (FNET) Implementation,” *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 1914–1921, Nov. 2005.

[4] D. Bertagnolli, “Worldwide Power System Oscillations Observed by Distribution Level Phasor Measurements Introduction Overview of FNET / Grideye Oscillation Cases Around The World,” in *CIGRE/EPRI Grid of the Future Symposium*, 2013.

[5] J. Guo, Y. Zhang, M. A. Young, M. J. Till, A. Dimitrovski, and P. Willging, “Design and Implementation of a Real-Time Off-Grid Operation Detection Tool from a Wide-Area Measurements Perspective,” *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 2080–2087, Jul. 2015.

[6] J. Guo, Y. Ye, Y. Zhang, Y. Lei, and Y. Liu, “Events associated power system oscillations observation based on distribution-level phasor measurements,” in *2014 IEEE PES T&D Conference and Exposition*, 2014, pp. 1–5.

[7] S. You, J. Guo, G. Kou, Y. Liu, and Y. Liu, “Oscillation mode identification based on wide-area ambient measurements using multivariate empirical mode decomposition,” *Electr. Power Syst. Res.*, vol. 134, pp. 158–166, 2016.

[8] J. Guo, H. Liu, D. Zhou, J. Chai, Y. Zhang, and Y. Liu, “Real-time Power System Electromechanical Mode Estimation Implementation and Visualization Utilizing Synchrophasor Data,” in *2016 IEEE PES T&D Conference and Exposition*, 2016, pp. 1–5.

[9] H. Liu, L. Zhu, Z. Pan, J. Guo, J. Chai, W. Yu, and Y. Liu, “Comparison of MIMO System Identification Methods for Electromechanical Oscillation Damping Estimation,” in *2016 Power and Energy Society General Meeting*, 2016, pp. 1–5.

[10] S. You, J. Guo, G. Kou, Y. Liu, and Y. Liu, “Oscillation mode identification based on wide-area ambient measurements using multivariate empirical mode decomposition,” *Electr. Power Syst. Res.*, vol. 134, pp. 158–166, 2016.

[11] Y. Liu, Y. Liu, Y. Zhang, J. Guo, and D. Zhou, “Wide Area Monitoring through Synchrophasor Measurement,” in *Smart Grid Handbook, 3 Volume Set*, vol. 1, Wiley, 2016.

[12] Y. Zhang, Y. Liu, L. Chen, J. Guo, and Y. Liu, “Visualization of distribution level voltage magnitude pattern trend in EI system using FNET data,” in *2014 IEEE PES T&D Conference and Exposition*, 2014, pp. 1–5.

[13] D. Zhou, J. Guo, Y. Zhang, J. Chai, H. Liu, Y. Liu, C. Huang, X. Gui, and Y. Liu, “Distributed Data Analytics Platform for Wide-Area Synchrophasor Measurement Systems,” *IEEE Trans. Smart Grid*, pp. 1–9, 2016.

[14] F. Hu, K. Sun, A. Del Rosso, E. Farantatos, and N. Bhatt, "Measurement-Based Real-Time Voltage Stability Monitoring for Load Areas," *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 2787–2798, 2016.

[15] T. Jiang, H. Yuan, H. Jia, N. Zhou, and F. Li, "A Novel SSI-based Approach for Tracking Inter-area Oscillatory Modes in Bulk Power System Utilizing Synchrophasor Measurements."

[16] H. Yuan, T. Jiang, H. Jia, F. Li, Y. Mishra, H. Chen, and G. Li, "Real-time wide-area loading margin sensitivity (WALMS) in power systems," in *2015 IEEE Power & Energy Society General Meeting*, 2015, pp. 1–5.

[17] Y. Lei, G. Kou, J. Guo, Y. Liu, and R. Nuqui, "Eastern Interconnection model reduction based on phasor measurements," in *2014 IEEE PES T&D Conference and Exposition*, 2014, pp. 1–5.

[18] J. Chai, J. Zhao, W. Yao, J. Guo, and Y. Liu, "Application of Wide Area Power System Measurement for Digital Authentication," in *2016 IEEE PES T&D Conference and Exposition*, 2016, pp. 1–5.

[19] J. Chai, Y. Liu, Z. Yuan, R. W. Conners, and Y. Liu, "Tampering Detection of Digital Recordings Using Electric Network Frequency and Phase Angle." Audio Engineering Society, 2013.

[20] P. N. Markham, T. Bilke, and D. Bertagnolli, "Analysis of frequency extrema in the Eastern and Western Interconnections, 2010–2011," in *2012 IEEE Power and Energy Society General Meeting*, 2012, pp. 1–8.

[21] Y. Lei, Y. Zhang, J. Guo, D. Zhou, J. Culliss, P. Irminger, and Y. Liu, "The impact

of synchronized human activities on power system frequency," in *2014 IEEE PES General Meeting | Conference & Exposition*, 2014, pp. 1–5.

[22] J. Chai, Y. Liu, J. Guo, L. Wu, D. Zhou, Y. Liu, T. King, M. Young, and J. R. Gracia, "FNET/GridEye-based Wide-area Measurement Data Analytics," in *2016 IEEE PES Power Systems Computation Conference (PSCC)*, 2016.

[23] L. Wu, Y. Liu, D. Zhou, J. Guo, and Y. Liu, "Observation of Inertial Frequency Response of Main Power Grids Worldwide Using FNET/GridEye," in *2016 Power and Energy Society General Meeting*, 2016, pp. 1–5.

[24] C. Huang, F. Li, J. Guo, Z. Pan, Y. Liu, and Y. Liu, "Data Quality Issues for Synchrophasor Applications Part I: A Review," *J. Mod. Power Syst. Clean Energy*, vol. 4, no. 3, 2016.

[25] L. Liu, J. Chai, H. Qi, and Y. Liu, "Power grid disturbance analysis using frequency information at the distribution level," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2014, pp. 523–528.

[26] L. Zhan, Y. Liu, J. Culliss, J. Zhao, and Y. Liu, "Dynamic Single-Phase Synchronized Phase and Frequency Estimation at the Distribution Level," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 2013–2022, 2015.

[27] L. Zhan, Y. Liu, and Y. Liu, "A Clarke Transformation-Based DFT Phasor and Frequency Algorithm for Wide Frequency Range," *IEEE Trans. Smart Grid*, pp. 1–1, 2016.

[28] M. Buevich, X. Zhang, O. Shih, D. Schnitzer, T. Escalada, A. Jacquiau-Chamski,

J. Thacker, and A. Rowe, “Microgrid Losses: When the Whole Is Greater Than the Sum of Its Parts,” in *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*, 2016, pp. 1–10.

[29] F. Galvan and C. H. Wells, “Detecting and managing the electrical island created in the aftermath of Hurricane Gustav using Phasor Measurement Units (PMUs),” in *IEEE PES T&D 2010*, 2010, pp. 1–5.

[30] A. Timbus, A. Oudalov, and C. N. M. Ho, “Islanding detection in smart grids,” in *2010 IEEE Energy Conversion Congress and Exposition*, 2010, pp. 3631–3637.

[31] L. E. Miller, J. Schoene, R. Kunte, and G. Y. Morris, “Smart grid opportunities in islanding detection,” in *2013 IEEE Power & Energy Society General Meeting*, 2013, pp. 1–4.

[32] F. De Mango, M. Liserre, A. D. Aquila, and A. Pigazo, “Overview of Anti-Islanding Algorithms for PV Systems. Part I: Passive Methods,” in *2006 12th International Power Electronics and Motion Control Conference*, 2006, pp. 1878–1883.

[33] S.-I. Jang and K.-H. Kim, “An Islanding Detection Method for Distributed Generations Using Voltage Unbalance and Total Harmonic Distortion of Current,” *IEEE Trans. Power Deliv.*, vol. 19, no. 2, pp. 745–752, Apr. 2004.

[34] L. A. C. Lopes and H. Sun, “Performance Assessment of Active Frequency Drifting Islanding Detection Methods,” *IEEE Trans. Energy Convers.*, vol. 21, no. 1, pp. 171–180, Mar. 2006.

[35] Guo-Kiang Hung, Chih-Chang Chang, and C.-L. Chen, “Automatic phase-shift

method for islanding detection of grid-connected photovoltaic inverters," *IEEE Trans. Energy Convers.*, vol. 18, no. 1, pp. 169–173, Mar. 2003.

[36] P. Mahat and B. Bak-Jensen, "A Hybrid Islanding Detection Technique Using Average Rate of Voltage Change and Real Power Shift," *IEEE Trans. Power Deliv.*, vol. 24, no. 2, pp. 764–771, Apr. 2009.

[37] J. Yin, C. P. Diduch, and L. Chang, "Islanding Detection Using Proportional Power Spectral Density," *IEEE Trans. Power Deliv.*, vol. 23, no. 2, pp. 776–784, Apr. 2008.

[38] W. Xu, G. Zhang, C. Li, W. Wang, G. Wang, and J. Kliber, "A Power Line Signaling Based Technique for Anti-islanding Protection of Distributed Generators: Part I: Scheme and Analysis," in *2007 IEEE Power Engineering Society General Meeting*, 2007, pp. 1–1.

[39] A. Borghetti, C. A. Nucci, M. Paolone, G. Ciappi, and A. Solari, "Synchronized phasors monitoring during the islanding maneuver of an active distribution network," in *2010 Innovative Smart Grid Technologies (ISGT)*, 2010, pp. 1–8.

[40] R. Sun and V. A. Centeno, "Wide Area System Islanding Contingency Detecting and Warning Scheme," *IEEE Trans. Power Syst.*, vol. 29, no. 6, pp. 2581–2589, Nov. 2014.

[41] K. El-Arroudi and G. Joos, "Data Mining Approach to Threshold Settings of Islanding Relays in Distributed Generation," *IEEE Trans. Power Syst.*, vol. 22, no. 3, pp. 1112–1119, Aug. 2007.

[42] K. El-Arroudi, G. Joos, I. Kamwa, and D. T. McGillis, "Intelligent-Based

Approach to Islanding Detection in Distributed Generation," *IEEE Trans. Power Deliv.*, vol. 22, no. 2, pp. 828–835, Apr. 2007.

[43] Z. Lin, T. Xia, Y. Ye, Y. Zhang, L. Chen, Y. Liu, K. Tomsovic, T. Bilke, and F. Wen, "Application of wide area measurement systems to islanding detection of bulk power systems," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 2006–2015, May 2013.

[44] "The Open Source Phasor Data Concentrator." [Online]. Available: http://openpdc.codeplex.com/. [Accessed: 13-Aug-2013].

[45] "Grid Solutions Framework." [Online]. Available: http://gsf.codeplex.com/. [Accessed: 13-Aug-2013].

[46] J. Prieto and G. Bravo, "Bad Data Detection and Identification in Distribution Power Systems By Means of Principal Component Analysis," no. 0466, pp. 6–9, 2011.

[47] D. N. Kosterev, C. W. Taylor, and W. a. Mittelstadt, "Model validation for the August 10, 1996 WSCC system outage," *IEEE Trans. Power Syst.*, vol. 14, no. 3, pp. 967–979, 1999.

[48] D. J. Trudnowski, "Estimating Electromechanical Mode Shape From Synchrophasor Measurements," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1188–1195, Aug. 2008.

[49] F. Bai, L. Zhu, Y. Liu, X. Wang, K. Sun, Y. Ma, M. Patel, E. Farantatos, and N. Bhatt, "Design and implementation of a measurement-based adaptive wide-area damping controller considering time delays," *Electr. Power Syst. Res.*, vol. 130,

pp. 1–9, Jan. 2016.

[50] J. F. Hauer, D. J. Trudnowski, and J. G. DeSteese, “A Perspective on WAMS Analysis Tools for Tracking of Oscillatory Dynamics,” in *2007 IEEE Power Engineering Society General Meeting*, 2007, pp. 1–10.

[51] D. J. Trudnowski and J. W. Pierre, “Overview of algorithms for estimating swing modes from measured responses,” in *2009 IEEE Power & Energy Society General Meeting*, 2009, pp. 1–8.

[52] L. Dosiek, N. Zhou, J. W. Pierre, Z. Huang, and D. J. Trudnowski, “Mode shape estimation algorithms under ambient conditions: A comparative review,” *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 779–787, May 2013.

[53] J. F. Hauer, C. J. Demeure, and L. L. Scharf, “Initial results in Prony analysis of power system response signals,” *IEEE Trans. Power Syst.*, vol. 5, no. 1, pp. 80–89, 1990.

[54] T. K. Sarkar and O. Pereira, “Using the matrix pencil method to estimate the parameters of a sum of complex exponentials,” *IEEE Antennas Propag. Mag.*, vol. 37, no. 1, pp. 48–55, Feb. 1995.

[55] D. S. Laila, A. R. Messina, and B. C. Pal, “A Refined Hilbert–Huang Transform With Applications to Interarea Oscillation Monitoring,” *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 610–620, May 2009.

[56] S. You, J. Guo, W. Yao, S. Wang, Y. Liu, and Y. Liu, “Multi-Channel Inter-Area Oscillation Mode Identification Using Multivariate Empirical Mode Decomposition,” in *2016 Power and Energy Society General Meeting*, 2016, pp.

1–5.

[57] N. Rehman and D. P. Mandic, “Multivariate empirical mode decomposition,” *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 466, no. 2117, pp. 1291–1302, 2010.

[58] N. Zhou, J. W. Pierre, and J. F. Hauer, “Initial Results in Power System Identification From Injected Probing Signals Using a Subspace Method,” *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1296–1302, Aug. 2006.

[59] V. Venkatasubramanian, “Oscillation monitoring from ambient PMU measurements by Frequency Domain Decomposition,” in *2008 IEEE International Symposium on Circuits and Systems*, 2008, pp. 2821–2824.

[60] Q. Hu, H. Jia, L. Bai, T. Jiang, X. Jin, and F. Li, “Synchrophasor measurement-based correlation approach for dominant mode identification in bulk power systems,” *IET Gener. Transm. Distrib.*, May 2016.

[61] T. Jiang, H. Jia, H. Yuan, N. Zhou, and F. Li, “Projection Pursuit: A General Methodology of Wide-Area Coherency Detection in Bulk Power Grid,” *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 2776–2786, Jul. 2016.

[62] T. Wu, S. A. N. Sarmadi, V. Venkatasubramanian, A. Pothen, and A. Kalyanaraman, “Fast SVD Computations for Synchrophasor Algorithms,” pp. 4–5, 2015.

[63] N. E. Huang, Z. Shen, S. Long, M. Wu, H. Shih, Q. Zheng, N.-C. Yen, C. Tung, and H. Liu, “The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis,” *Proc. R. Soc. London. Ser. A Math. Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, 1998.

[64] M. G. Anderson, N. Zhou, J. W. Pierre, and R. W. Wies, “Bootstrap-Based Confidence Interval Estimates for Electromechanical Modes From Multiple Output Analysis of Measured Ambient Data,” *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 943–950, May 2005.

[65] B. Friedlander and B. Porat, “The Modified Yule-Walker Method of ARMA Spectral Estimation,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-20, no. 2, pp. 158–173, 1984.

[66] “Google Charts API.” [Online]. Available: https://developers.google.com/chart/. [Accessed: 10-Jul-2012].

[67] “Dygraphs JavaScript Visulization Library.” [Online]. Available: http://dygraphs.com/. [Accessed: 01-Jul-2012].

[68] A. Bose, “Smart Transmission Grid Applications and Their Supporting Infrastructure,” *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 11–19, Jun. 2010.

[69] M. Kezunovic, L. Xie, and S. Grijalva, “The role of big data in improving power system operation and protection,” in *2013 IREP Symposium Bulk Power System Dynamics and Control - IX Optimization, Security and Control of the Emerging Power Grid*, 2013, pp. 1–9.

[70] “Managing big data for smart grids and smart meters.” [Online]. Available: http://www-935.ibm.com/services/multimedia/Managing_big_data_for_smart_grids_and_smart_meters.pdf. [Accessed: 14-Jan-2016].

[71] H. Liu, J. Guo, W. Yu, T. Xia, R. Sun, M. Gardner, L. Zhu, and Y. Liu, “Design

and Implementation of Enterprise-level Data Platform and Big Data Driven Applications,” in *2016 IEEE PES T&D Conference and Exposition*, 2016, pp. 1–5.

[72] “Sensor-based data | PI System – OSIsoft.” [Online]. Available: http://www.osisoft.com/. [Accessed: 14-Jan-2016].

[73] S. You, L. Zhu, and Y. Liu, “A Survey on Next-generation Power Grid Data Architecture,” 2015.

[74] X. L. Dong and D. Srivastava, “Big data integration,” in *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, 2013, pp. 1245–1248.

[75] “CIMSpy EE - A CIM-based Model Exploratory Tool.” [Online]. Available: http://www.powerinfo.us/CIMSpyEE.html. [Accessed: 14-Jan-2016].

[76] “Solar Industry Data.” [Online]. Available: http://www.seia.org/research-resources/solar-industry-data. [Accessed: 17-Nov-2015].

[77] K. D. Orwig, M. L. Ahlstrom, V. Banunarayanan, J. Sharp, J. M. Wilczak, J. Freedman, S. E. Haupt, J. Cline, O. Bartholomy, H. F. Hamann, B.-M. Hodge, C. Finley, D. Nakafuji, J. Peterson, D. Maggio, and M. Marquis, “Recent Trends in Variable Generation Forecasting and Its Value to the Power System,” *IEEE Trans. Sustain. Energy*, vol. 6, no. 3, pp. 924–933, Jul. 2015.

[78] X. Fang, F. Li, N. Gao, and Q. Guo, “Available transfer capability of photovoltaic generation incorporated system,” in *2014 IEEE PES General Meeting | Conference & Exposition*, 2014, pp. 1–5.

[79] M. J. E. Alam, K. M. Muttaqi, and D. Sutanto, “Distributed energy storage for mitigation of voltage-rise impact caused by rooftop solar PV,” in *2012 IEEE*

Power and Energy Society General Meeting, 2012, pp. 1–8.

[80] M. Starke, J. Nutaro, T. Kuruganti, and D. Fugate, “Integration of photovoltaics into building energy usage through advanced control of Rooftop Unit,” *International High Performance Buildings Conference*, 2014. [Online]. Available: http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1154&context=ihpbc.

[81] V. Prema and K. U. Rao, “Development of statistical time series models for solar power prediction,” *Renew. Energy*, vol. 83, pp. 100–109, Nov. 2015.

[82] L. A. Fernandez-Jimenez, A. Muñoz-Jimenez, A. Falces, M. Mendoza-Villena, E. Garcia-Garrido, P. M. Lara-Santillan, E. Zorzano-Alba, and P. J. Zorzano-Santamaria, “Short-term power forecasting system for photovoltaic plants,” *Renew. Energy*, vol. 44, pp. 311–317, Aug. 2012.

[83] R. J. Bessa, A. Trindade, C. S. P. Silva, and V. Miranda, “Probabilistic solar power forecasting in smart grids using distributed information,” *Int. J. Electr. Power Energy Syst.*, vol. 72, pp. 16–23, Nov. 2015.

[84] P. Bacher, H. Madsen, and H. A. Nielsen, “Online short-term solar power forecasting,” *Sol. Energy*, vol. 83, no. 10, pp. 1772–1783, Oct. 2009.

[85] J. Song, V. Krishnamurthy, A. Kwasinski, and R. Sharma, “Development of a Markov-Chain-Based Energy Storage Model for Power Supply Availability Assessment of Photovoltaic Generation Plants,” *IEEE Trans. Sustain. Energy*, vol. 4, no. 2, pp. 491–500, Apr. 2013.

[86] H.-T. Yang, C.-M. Huang, Y.-C. Huang, and Y.-S. Pai, “A Weather-Based Hybrid Method for 1-Day Ahead Hourly Forecasting of PV Power Output,” *IEEE Trans.*

Sustain. Energy, vol. 5, no. 3, pp. 917–926, Jul. 2014.

[87] M. Zamo, O. Mestre, P. Arbogast, and O. Pannekoucke, “A benchmark of statistical regression methods for short-term forecasting of photovoltaic electricity production, part I: Deterministic forecast of hourly production,” *Sol. Energy*, vol. 105, pp. 792–803, 2014.

[88] J. Liu, W. Fang, X. Zhang, and C. Yang, “An Improved Photovoltaic Power Forecasting Model With the Assistance of Aerosol Index Data,” *IEEE Trans. Sustain. Energy*, vol. 6, no. 2, pp. 434–442, Apr. 2015.

[89] M. Benghanem, A. Mellit, and S. N. Alamri, “ANN-based modelling and estimation of daily global solar radiation data: A case study,” *Energy Convers. Manag.*, vol. 50, no. 7, pp. 1644–1655, Jul. 2009.

[90] L. Olatomiwa, S. Mekhilef, S. Shamshirband, K. Mohammadi, D. Petković, and C. Sudheer, “A support vector machine–firefly algorithm-based model for global solar radiation prediction,” *Sol. Energy*, vol. 115, pp. 632–644, May 2015.

[91] S. Salcedo-Sanz, C. Casanova-Mateo, A. Pastor-Sánchez, and M. Sánchez-Girón, “Daily global solar radiation prediction based on a hybrid Coral Reefs Optimization – Extreme Learning Machine approach,” *Sol. Energy*, vol. 105, pp. 91–98, Jul. 2014.

[92] A. Tuohy, J. Zack, S. E. Haupt, J. Sharp, M. Ahlstrom, S. Dise, E. Grimit, C. Mohrlen, M. Lange, M. G. Casado, J. Black, M. Marquis, and C. Collier, “Solar Forecasting: Methods, Challenges, and Performance,” *IEEE Power Energy Mag.*, vol. 13, no. 6, pp. 50–59, Nov. 2015.

[93] G. James, D. Witten, T. Hastie, and R. Tibshirani, “Introduction to Statistical Learning,” *Springer*, 2014. [Online]. Available: http://www-bcf.usc.edu/~gareth/ISL/.

[94] “National Solar Radiation Data Base.” [Online]. Available: http://rredc.nrel.gov/solar/old_data/nsrdb/. [Accessed: 17-Jan-2016].

[95] C. E. Otero and A. Peter, “Research Directions for Engineering Big Data Analytics Software,” *IEEE Intell. Syst.*, vol. 30, no. 1, pp. 13–19, Jan. 2015.

[96] D. Wang and L. Xiao, “Storage and query of condition monitoring data in smart grid based on Hadoop,” *Proc. - 4th Int. Conf. Comput. Inf. Sci. ICCIS 2012*, pp. 377–380, 2012.

[97] M. Edwards, A. Rambani, Y. Zhu, and M. Musavi, “Design of Hadoop-based Framework for Analytics of Large Synchrophasor Datasets,” *Procedia Comput. Sci.*, vol. 12, pp. 254–258, 2012.

[98] J. Zheng and A. Dagnino, “An initial study of predictive machine learning analytics on large volumes of historical data for power system applications,” in *2014 IEEE International Conference on Big Data (Big Data)*, 2014, pp. 952–959.

[99] M. Khan, P. M. Ashton, M. Li, G. A. Taylor, I. Pisica, and J. Liu, “Parallel Detrended Fluctuation Analysis for Fast Event Detection on Massive PMU Data,” *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 360–368, Jan. 2015.

[100] K. Martin and C. Sun, “Best Practice Recommendations for Synchrophasor Systems : Administration , Planning and Implementation , and Operation and Maintenance,” 2013.

[101] “The Open Source Time-Series Data Historian.” [Online]. Available: https://openhistorian.codeplex.com.

[102] “Apache Spark,” 2016. [Online]. Available: http://spark.apache.org/.

[103] “Hadoop.” [Online]. Available: https://hadoop/apache.org.

[104] “The R Project for Statistical Computing.” [Online]. Available: https://www.r-project.org/.

[105] IEEE Power and Energy Society, “C37.118.1-2011 - IEEE Standard for Synchrophasor Measurements for Power Systems.”

[106] “MongoDB.” [Online]. Available: https://www.mongodb.com/.

[107] “Cassandra.” [Online]. Available: http://cassandra.apache.org/.

[108] J. Bian, “Generator Frequency Response,” 2009.

Vita

Jiahui Guo was born in Inner Mongolia, China. He got his B.S. degree in Department of Electrical Engineering from Tsinghua University, Beijing, China, in 2011, and the M.S. degree in Department of Electrical Engineering and Computer Science from the University of Tennessee, Knoxville, TN, USA, in 2014. He is working towards his Ph.D. degree in the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, with anticipated graduation in August, 2016. He is a recipient of Min Kao Fellowship in 2011, and a recipient of Extraordinary Professional Promise (Chancellor's Honors) in 2016.

His research interests include wide-area power system monitoring, situational awareness application development and big data analytics.