Synchrophasor Visualizer

Kenta Kirihara, *Student Member, IEEE*, Karl E. Reinhard, *Student Member, IEEE*,
Yang Liu, *Student Member, IEEE*, Peter W. Sauer, *Fellow, IEEE*
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

***Abstract*—The deployment of synchrophasor technology is a task that many institutions are interested in. The granularity of the information provided by synchrophasors allow for smarter operation of the electricity grid. The Synchro Phasor Visualizer (SPV) is an application that covers the general functionality of the synchrophasor technology. The application aims to introduce a prototype of a functional form of the synchrophasor technology that would be deployed within operation of the electricity grid: visualization, event detection, and data quality issue remediation. In the process of attaining this functional form, SPV identifies the potential challenges and requirements for synchrophasor usage.**

I. Introduction

The latest North American Synchro Phasor Initiative (NASPI) has identified the status of the synchrophasor development in North America. Out of the 13 entities listed in the NASPI Recovery Act Smart Grid Investments Synchrophasor Project Status, WECC is the only entity in which has successfully implemented all of the applications of synchrophasors, both in real-time and study mode application as of Summer 2014 [1]. Many of the projects are still in research stage, and have yet to be implemented. The fundamental challenge with the synchrophasor effort is the lack of designated engineers solely working on the synchrophasor task. Often times, these engineers are expert engineers from both operations and system planning, but the problem lies in the possible time commitment of these engineers. Though Smart Grid Initiative funds their effort in terms of resource, it is still only a fraction of their duties.

To remedy this challenge, a baseline application in which entities can build upon is proposed. Synchrophasor Visualizer (SPV) is a java-based application that covers the essential functionality of the synchrophasor network. SPV utilizes real synchrophasor data for development and is intended for use in all electric companies for synchrophasor implementation.

II. Concept

To understand the functionality of SPV, it is essential to understand the synchrophasor network first. Figure 1 shows the nominal synchrophasor network that is being implemented across the nation. Synchrophasor measurements are collected from Phasor Measurement Units (PMUs) at layer 1, routed through Phasor Data Concentrators (PDC) and Super PDCs at layer 2 and 3, and all the way to control center and data archive at layer 4. SPV is an application that utilizes both the real-time data stream which enters to the control center as well as archive data in a database.

Figure 2 shows the functional block diagram of the SPV. Data is fed in through two sources, either from a live stream or through a database; however, the functionality of SPV does not differ between the two modes of operation: archived data is played back as if the data is streamed. Once the data is secured in the program, the data is utilized in three ways: visualization of raw information, event detection, and synchrophasor data quality issue detection. The three sections work independently and are displayed upon user control.

III. SPV Build

Unlike traditional synchrophasor tools, which were tools for post-event analysis, SPV is a tool that focuses mainly on real-time data. Figure 3 shows the basic structure of SPV which takes this fact into consideration. In SPV, the most essential task is to visualize data continuously. That is, data should be processed in such a way that the continuous stream of information is uninterrupted. To accomplish this, the program is divided into four different processes: data

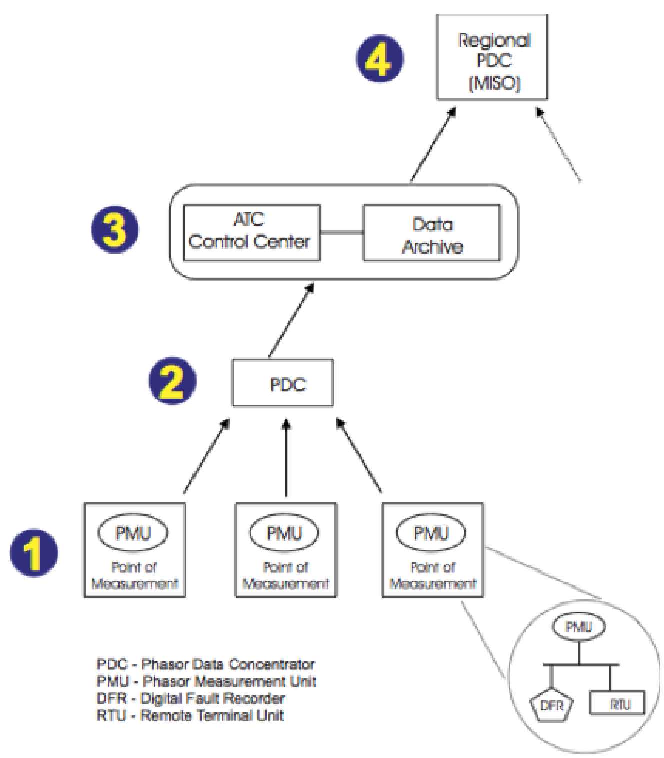

Figure 1. Synchrophasor Network [2]

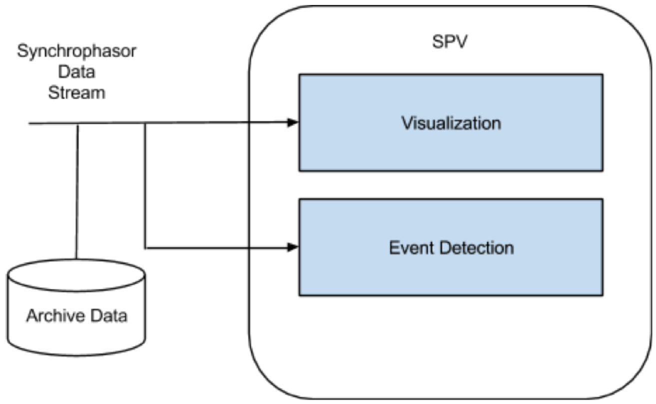

Figure 2. SPV Functional Diagram

retrieval, data processing, data visualization, and user control. This concept is elementary in modular programming. Creating independent sections allows for multiple tasks to be processed independently of each other. However, increase in the number of parallel processing brings about challenges. A core in a processor is only able to process at maximum two parallel threads as of 2015. For a truly independent operation, it would be ideal to be able to dedicate a CPU thread to each process. Of course, many processes take up more than a thread. When programs such as remote access is open, where data streams flow bidirectionally, many threads must be active: thread to access the server, thread to control the GPU, etc. This is done through the use of context switching in which threads are sequentially processed by switching between threads in a very quick manner. As a reference, a commercial computer with 4 available CPU threads often contain around 1000 threads upon normal use as in Figure4.

In the process of building SPV, it is essential to understand the nature of threads in general. Any task that requires a query will benefit from having more threads and opposed to computation to require less threads. SPV is a combination of these two types of operations and thus utilizes both a task requiring more threads and another that requires less threads.

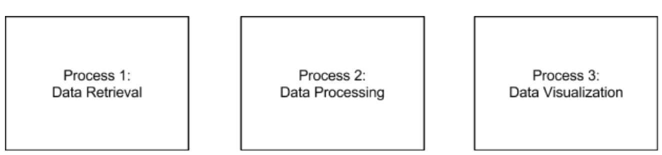

Figure 3. SPV Structure

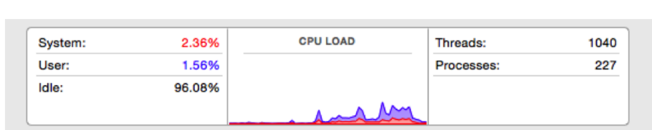

Figure 4. Example Thread Usage

A. Data Retrieval

As with any program, data retrieval is essential in bringing data into the system. When doing so, there are two modes of operation. One method is to have a constant stream of data entering SPV through a streaming connection. Another method is to have the data queried from a database where the playback data is stored locally: in the case of data being stored locally, the data is played back at a rate of 30Hz. Regardless, the data retrieval process must be a continuously running process at a constant retrieval rate. In other words, from a core management perspective, it is a process that can utilize context switching: since most of the processing time is dependent on the refresh rate or query time. More threads can be utilized into this process.

B. Data Processing

In this process, computation is essentially executed. The essential task in SPV, with regards to computation, is to detect anomalies in the synchrophasor measurements while omitting data quality issues. There are two separate types of computation that is executed in parallel. One is a Fast Fourier Transform and another being an anomaly detection through a variation of one dimensional moving distribution. Both are tasks that require less thread and more dedicated CPU core.

1) Fast Fourier Transform: Fast Fourier Transform (FFT) is a type of algorithm in which a time series data can be converted into a Fourier domain. The optimal efficiency for such algorithm is:

$$O(N \log N) \tag{1}$$

The algorithm is relatively computationally inexpensive, and is suited for running in real-time. FFT is utilized to capture an oscillation that may occur in the power system: as this is a fairly common method of detecting oscillation, this feature is left in the back-end. Since oscillation is fairly uncommon in a power system, except for the case of wind farms [4], this feature will not be used as heavily.

In SPV, 3000 data points of synchrophasor voltage magnitude for one phase is collected and a FFT is performed after zero-padding. The advantage of using this many points is that it allows a detection of slower frequency oscillation by having many data points to reference. If an oscillation were to be detected, signal is sent to the visualization process.

2) Moving Distribution-based Anomaly Detection: The core of the processing in the synchrophasor data is in a distribution-based anomaly detection. Essentially, this process is a one dimensional version of an outlier detection through the combination of statistical methods. First, the latest 3600 points, or 2 minutes of synchrophasor voltage measurement data is collected. The mean and the variance of the data points are calculated. The calculation is repeated for every new data point that enters the system. If more than three latest data points lie beyond four standard deviation from the mean, then an alarm is triggered. The alarm will continue to trigger until

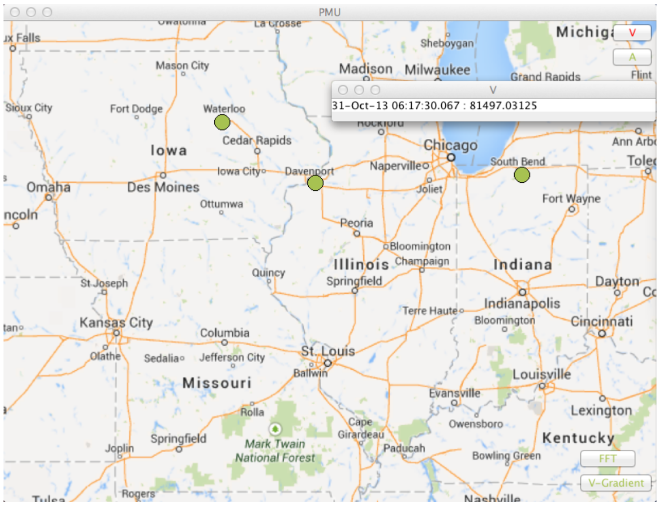

Figure 5. SPV Default

the moving average and deviation completely accommodate to the steady-state reached by the transient.

Instead of having an alarm set when one point is found outside the fourth standard deviation, the alarm is set when multiple consecutive points are outside of the fourth standard deviation from the mean to avoid any potential challenges with synchrophasor data issues. Inherent noise in the synchrophasor data can possibly set a false alarm with a single synchrophasor data point. By requiring three data points to lie outside of the standard deviation, a more careful decision can be made. In three data points, the moving average will move closer to the new steady-state if there exist one. Upon doing so, setting threshold to four standard deviation will reduce the false positive rate as well as the time for which the alarm is enabled.

C. Data Visualization and Control

Data is visualized as shown in Figure 5. Each circle on the map represents a PMU location. Selecting a PMU will reveal the real-time measurements of voltage magnitude and angle. Upon an event detection, an alarm is shown in the same dialog box.

When a more overview-like situation is necessary, the voltage magnitude or voltage angle is mapped out using a gradient map for immediate wide-area system awareness as in Figure 6. Like with the concept of one-line diagrams, the voltage magnitude at each point is normalized to a per-unit convention. By doing such operation, the operator is able to see the relative voltage levels at each section of the map.

The visualization is mapped out and refreshed at a rate of 10Hz per second. The refresh rate is lowered to 10Hz per second since opinions from operators revealed that excess information are non-preferable [3].

IV. SPV Testing and Limitation

SPV was tested using masked archive data from American Transmission Company. The machine used to test SPV featured an Intel Core 2 Quad CPU Q8400 @2.66 Ghz and 16GB RAM. The SPV was ran with the data sets from two distinct PMU locations. Instead of querying these two data sets from a database, or even simulating streaming connection, the data was replayed internally by retrieving the segments of data from two files at 30 lines per second per location.

To prepare the test case, SitAAR from Pacific Northwest National Laboratory was utilized. Figure 7 shows the signature found using this method. Upon confirmation with ATC, the data set captures a double capacitor switching phenomenon. The data sets for two PMU locations, one with the signature and one without, was processed through SPV for testing. The data were pre-queried or left inside the directory and played back.

The results were as expected. SPV was capable of replaying the data set and was also capable of capturing the event. The

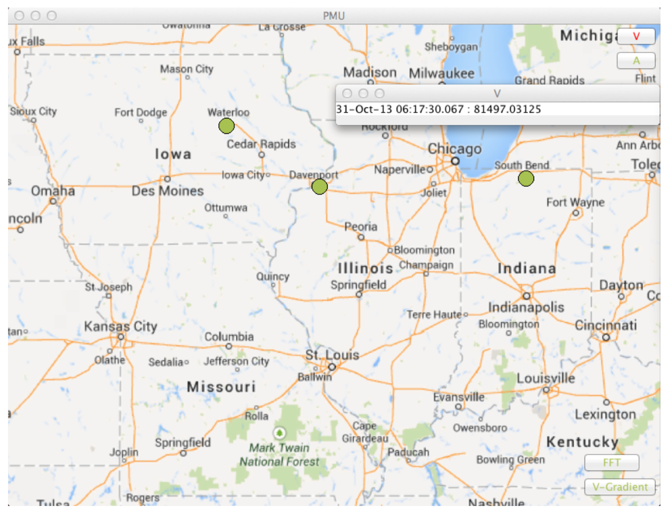

Figure 6. SPV Gradient Mapping: Green indicates normal voltage (1.0 p.u); red indicating low voltage (.95 p.u).

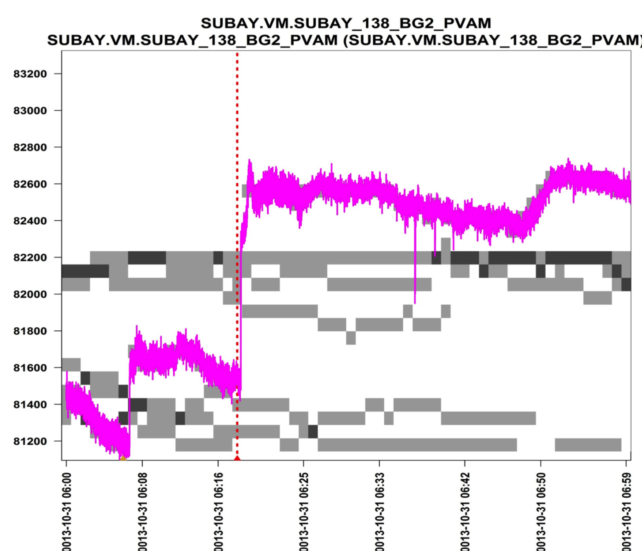

Figure 7. Double Capacitor Switching

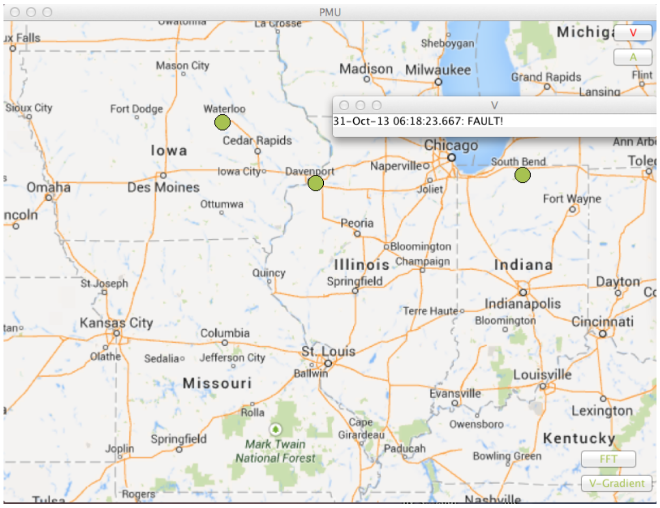

Figure 8. Event Detected

event caught is as shown in Figure 8.

Though SPV was able to perform, there were also drawbacks in the functionality of SPV. First, the test case is minimal in terms of scaling; that is, two PMU locations being processed is far less computationally intensive than one hundred PMUs. If SPV were to accommodate for more PMUs, perhaps it would not be suffice to run this task solely on one computer. Even with this test case, running computation in the background seems to have used a good portion of computational resources.

Also, another drawback that is inherent is the event detection portion. The event detection method, does indeed detect an event, however, still cannot differentiate between different types of events. A capacitor bank switching and a line trip will be caught, since the PMU reporting rate is high enough to catch these transients, but this simple method only is insufficient for categorization of faults. To do so, other methods such as sliding window polynomial regression may be more suitable, but bring about question in computational resources. The FFT oscillation detection was never tested since a test case was never found.

V. Conclusion

Through the construction of SPV, one fundamental question is raised: should synchrophasor visualization and event detection be executed on a local machine.

As mentioned in the final section, SPV running on a local machine was suffice to complete an elementary task of visualizing two points and detecting an event. However, as more PMU locations are considered and categorization of faults are wanted, the task becomes much more heavy on the user-end.

One method may be simply search for a better algorithm to compute the events and to visualize. The more plausible method, at this point, is to utilize clusters to divide up the complex task. Since each event detection scheme is independent from other locations, the computation can be parallel processed in different nodes of a cluster, and only sends an alarm when necessary. By having a dedicated machine for each event detection stream, it may be computationally plausible and suitable for running complex algorithms to categorize events. The ties between synchrophasor data utilization and cloud computing is the next topic that should be considered.

References

[1] North American Synchro Phasor Initiative (2014), ”Recovery Act Smart Grid Investments SYNCHROPHASOR PROJECT STATUS” [Online]. Available: https://www.smartgrid.gov/sites/default/files/synchrophasor_project_status_061014-1.pdf

[2] Kirihara, K.; Reinhard, K.E.; Yoon, A.K.; Sauer, P.W., ”Investigating Synchrophasor Data Quality issues,” Power and Energy Conference at Illinois (PECI), 2014 , vol., no., pp.1,4, Feb. 28 2014-March 1 2014 doi: 10.1109/PECI.2014.6804563 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6804563&isnumber=6804537

[3] Obradovich, J.;”NASPI Phasor Tools Visualization Workshop a quick summary”, Atlanta, 2012. Internet:http://www.nerc.com/pa/rrm/hp/2012%20Human%20Performance%20Conference/NASPI%20Viz%20Workshop%20Summary_NERC%20HP_JHObradovich.pdf

[4] Y. Wan, ’Synchronized Phasor Data for Analyzing Wind Power Plant Dynamic Behavior and Model Validation’, U.S. Department of Energy, Oakridge, TN, 2013. [Online]. Available:http://www.nrel.gov/docs/fy13osti/57342.pdf