



Pacific Northwest
NATIONAL LABORATORY

*Proudly Operated by **Battelle** Since 1965*

Forced Oscillation Detection and Analysis with the DISAT Application

March 2016

JD Follum
FK Tuffner



Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.** Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <<http://www.ntis.gov/about/form.aspx>>
Online ordering: <http://www.ntis.gov>



This document was printed on recycled paper.

(8/2010)

Forced Oscillation Detection and Analysis with the DISAT Application

JD Follum
FK Tuffner

March 2016

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Abstract

In recent years, power system engineers have become increasingly concerned with forced oscillations, which often arise due to the malfunction or misoperation of equipment. To detect these oscillations, which are often too small to be easily observed in time-domain synchrophasor data, several detection algorithms have been developed. Considering the vast quantities of synchrophasor data that system operators are now handling, tools are needed to detect and report on forced oscillations. To address this need, the Data Integrity Situational Awareness Tool (DISAT) was upgraded to perform forced oscillation detection and display results to the user. To perform the detection, two multi-channel algorithms were implemented. Several visualizations were implemented in the tool's application to allow initial investigation of detected oscillations. With the upgrade, DISAT is now able to provide needed detection and analysis of forced oscillations across a large system.

Acknowledgments

The authors would like to acknowledge Bonneville Power Administration (BPA) for providing data and useful feedback to support this project. The authors also would like to thank Phil Overholt with the Department of Energy, Office of Electricity Delivery and Energy Reliability, Transmission Reliability Program for funding this work.

Acronyms and Abbreviations

BPA	Bonneville Power Administration
DISAT	Data Integrity Situational Awareness Tool
DOE	Department of Energy
GMSC	Generalized Magnitude Squared Coherence
PMU	Phasor Measurement Unit
PNNL	Pacific Northwest National Laboratory

Contents

Abstract	iii
Acknowledgments.....	v
Acronyms and Abbreviations	vii
1.0 Introduction	1.1
2.0 Approach	2.1
2.1 Integration with DISAT	2.2
2.2 Data	2.3
2.3 Algorithms.....	2.3
2.4 Application.....	2.4
3.0 Setup	3.5
3.1 Data Processing	3.5
3.2 Mapping	3.7
4.0 Application	4.7
5.0 Conclusion.....	5.21
6.0 References	6.21
Appendix A User-Chosen Parameters in RealTimeRunDetect_Final.R.....	A.1
Appendix B SCmapInfo.R	B.3

Figures

Figure 1: Home screen of the DISAT application. Note the highlighted <i>Forced Oscillations</i> tab, which links to the upgraded forced oscillation detection results.	2.2
Figure 2: Example of the <i>Monthly Overview</i> subtab in the DISAT application.	4.9
Figure 3: Example of the <i>Daily Overview</i> subtab in the DISAT application.....	4.10
Figure 4: Example of the <i>Inspect Oscillation</i> subtab in the DISAT application.....	4.11
Figure 5: Example of the available options in the <i>Time to Plot</i> field.....	4.11
Figure 6: Example of using the <i>Select Minute</i> option to specify a minute for analysis.	4.12
Figure 7: Example of using the <i>Select Range</i> option to specify a range of minutes for analysis. .	4.12
Figure 8: Example map visualization for an oscillation that had strong evidence at a single location.....	4.13
Figure 9: Example map visualization for an oscillation that had strong evidence at multiple points in the system.	4.14
Figure 10: Example test statistic visualization for the periodogram method.	4.15
Figure 11: Example test statistic visualization for the spectral-coherence method.	4.15
Figure 12: Example of the <i>Detections Per PMU</i> visualization for the periodogram method. Compare with Figures 8 and 10.	4.16
Figure 13: Example of the <i>Detections Per PMU</i> visualization for the spectral coherence method. Compare with Figures 9 and 11.	4.17
Figure 14: Example of a time-domain visualization showing a large forced oscillation.	4.18
Figure 15: Example of a time-domain visualization with obvious data quality problems.....	4.18
Figure 16: Example plot of the periodogram method’s test statistic plotted over time with only values corresponding to detected oscillations included.	4.19
Figure 17: Portion of the table on the <i>Inspect Oscillation</i> subtab corresponding to the times included in Figures 16 and 18.	4.20
Figure 18: Example plot of the periodogram method’s test statistic plotted over time with values for all frequencies and times included.	4.20

1.0 Introduction

Over the past several years, networks of phasor measurement units (PMUs) have been installed in power systems. The measurements from these networks are synchronized across power systems and reported at high rates – typically 30 or 60 frames per second. The high reporting rates of PMUs allow for a wide range of applications, but they also lead to large volumes of data that can be extremely difficult to analyze with traditional techniques. In response to this challenge, Pacific Northwest National Laboratory (PNNL), with funding and cooperation from the Department of Energy (DOE) and the Bonneville Power Administration (BPA), developed the Data Integrity Situational Awareness Tool (DISAT). The tool, which is reported on in (Amidan et al. 2014), applied statistical analysis techniques to search for power system events in large volumes recorded data. As discussed separately in (Follum et al. 2015 A), the capability to detect forced oscillations was also included in DISAT.

Industry concern over forced oscillations has grown significantly over the past several years. Discussed infrequently in the literature for several decades, the high reporting rate and availability of PMU data have significantly increased the identification of forced oscillations in recent years. Again in response to the need to analyze vast quantities of PMU data, this time for the presence of forced oscillations, PNNL developed two multi-channel forced oscillation detection algorithms, known as the periodogram method and the spectral-coherence method. These algorithms are discussed in detail in (Follum 2015 B). After the algorithms were developed, they were used to upgrade DISAT's forced oscillation detection capabilities.

Along with upgrading the algorithms behind the oscillation detection, the organization and visualization of results within DISAT were improved. The goal of this report is to guide users through the setup and use of DISAT for forced oscillation detection and analysis. The parameters that impact the detection algorithms will be described, but for detailed descriptions of the algorithms the reader is referred to (Follum 2015 B).

The rest of this report is organized as follows. In Section 2, the general approach to the upgrade is described. Details on setting the application up are described in Section 3. The application, including the overall structure and descriptions of the various plots, is discussed in Section 4.

2.0 Approach

The approach to DISAT's upgrade for forced oscillation detection was guided significantly by the existing structure of DISAT and the PMU data files, as well as general observations about the occurrence of forced oscillations in power systems. In this section, these factors will be discussed to provide the motivation for the details of the upgrade's implementation discussed in Sections 3 and 4. The upgrade's approach will be discussed in terms of the integration with DISAT, data, detection algorithms, and application (user-interface).

2.1 Integration with DISAT

As an update to existing software, it was imperative that the upgrade was designed with the original DISAT implementation in mind. DISAT was implemented using the R software environment (R Core Team 2015). For the user interface, the R package Shiny was used (Chang 2015). To correspond to these initial designs, the upgrades to DISAT's forced oscillation detection capabilities were implemented in R. In the user interface, a new tab was added on the home screen to provide access to the newly available results. This tab is highlighted in Figure 1.

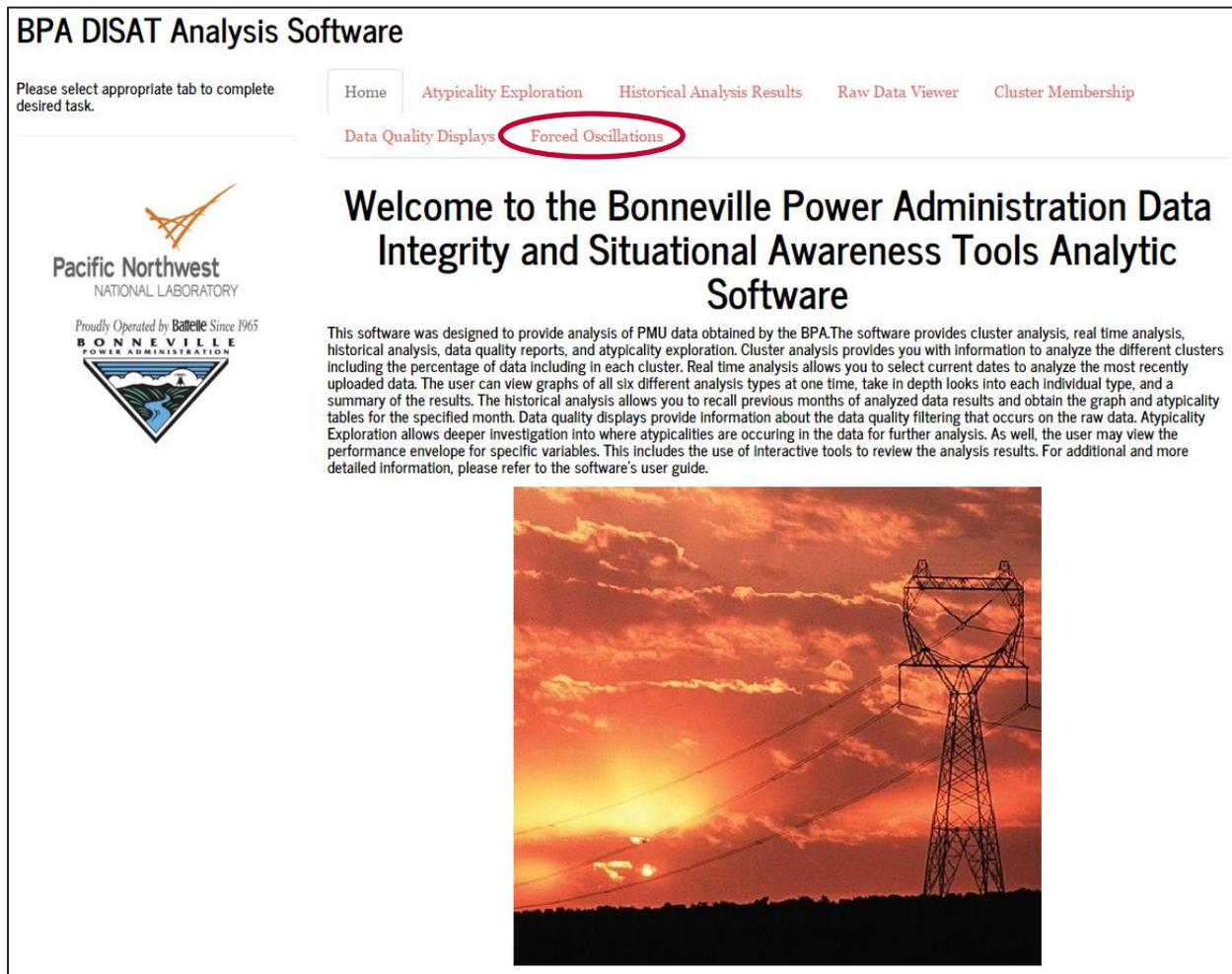


Figure 1: Home screen of the DISAT application. Note the highlighted *Forced Oscillations* tab, which links to the upgraded forced oscillation detection results.

Unlike the user interface, the script of R code that reads and analyzes PMU data was separated to divide the existing data analytics from the updated forced oscillation component. With the new implementation, two instances of R are utilized. The first instance of R implements all the operations for the original DISAT implementation minus forced oscillation detection. The second instance of R implements the operations for the upgraded forced oscillation detection component. This approach was taken to resolve differences in the way the two components process PMU data. Specifically, the big-data analyses in DISAT consider each minute of data separately, but the forced oscillation detection

algorithms benefit from examining multiple minutes of sequential data. This approach impacted the way the upgrade handles PMU data.

2.2 Data

The Bonneville Power Administration stores PMU data in one minute files with the custom PDAT file type. These files adhere to the C37.118 standard for storing synchrophasor measurements (IEEE 2011). To allow the data to be processed in R, DISAT was designed to continuously convert the PDAT files to the Rdata file type native to R. The forced oscillation script also allows for this conversion so that it can be run whether or not the main DISAT script is running.

A concern that arose with the conversion of PDAT files to the Rdata type is storage space. For the installation of DISAT at a BPA testing facility, PDAT files were stored remotely, but Rdata files were necessarily stored on the computer running DISAT. Though significant storage was available, it was not enough to store several years' worth of PMU data. To address this issue, the capability to delete old Rdata files as new ones are produced was added in the upgrade. This capability is discussed further in Section 3.

In analyzing PMU data, a common concern that arises is data quality. Data quality issues can arise for several reasons, including communication failures and problems with the GPS time synchronization. The original DISAT implementation included a data quality filter to remove corrupt samples. Whereas the big-data analyses in DISAT can simply ignore these corrupt samples, the forced oscillation detection algorithms require all samples to be available for the time period being analyzed. The length of this time period varies depending on the detection method and selected parameters. Examples presented in Section 4 were generated with between 1 minute and 10 minutes of data. Rather than attempting to correct the data, which can lead to erroneous detection results, channels with corrupt data were discarded from the analysis. Both detection algorithms, which will be described in the following section, are multi-channel generalizations of their single-channel counterparts, so they can operate on as little as one channel of data. Though detection performance and geographic information improve with the use of multiple channels, removing channels with data quality problems was found to be preferable to analyzing altered data.

2.3 Algorithms

The upgrade of DISAT's forced oscillation detection capabilities included the implementation of two new multi-channel methods. These methods, referred to as the periodogram method and the spectral-coherence method, are described in detail in (Follum 2015 B). Here, brief summaries are provided.

Both methods operate by forming a frequency-dependent test statistic. Forced oscillations are detected at frequencies where the test statistic exceeds a threshold. Other detection methods may not indicate the oscillation's frequency or may limit the frequency to a range, but the periodogram algorithm and the spectral-coherence algorithm share the characteristic of providing a specific frequency estimate for the oscillation. Providing this estimate directly impacted the design approach for the upgrade in several ways, as will be described in Sections 3 and 4. Though similar in this regard, the algorithms also differ significantly.

As suggested by its name, the periodogram method operates by examining the periodograms from each channel of data. Periodograms are frequency dependent, and at oscillation frequencies their value

increases along with the oscillation's amplitude and duration. The periodograms from each channel are combined into a single test statistic to improve detection performance. The periodogram method is very powerful and reliable, meaning that it can detect very small oscillations while very rarely producing false alarms. However, it requires several minutes of data to operate. As a result, it can be difficult to resolve the onset and conclusion of forced oscillations in results from the periodogram method.

The spectral-coherence method operates by combining the self-coherence spectra from several channels of data to form a test statistic. Like periodograms, self-coherence spectra are frequency dependent. They provide a measure of how much a signal is related to a delayed version of itself. A large value at a specific frequency indicates that a sustained oscillation is present in the data. The spectral-coherence method operates best on smaller windows of data. As a result, it is a less powerful detector than the periodogram method, but it provides better resolution of the onset and conclusion of forced oscillations. Because the method is not able to detect very small oscillations, it may be the preferred method if the user is specifically interested in oscillations with large amplitudes.

Each of the detection methods has advantages and disadvantages. By including both in the tool, the user may utilize the strengths of each. Their frequency-dependent test statistics and ability to produce specific frequency estimates for detected oscillations led to specific approaches for designing the DISAT application, as described next.

2.4 Application

One of the observations from initial examination of PMU data was that forced oscillations are quite common. Forced oscillations with distinct frequencies are often present in the system at the same time and it is common for them to start and stop multiple times in a single day. With so many oscillations, it would be easy to overwhelm the user when displaying results in the application. To avoid this situation, several specific design approaches for the application were implemented.

First, the application takes a drill-down approach to displaying detection results. As will be described in more detail in Section 4, the application provides overviews at the month, day, and oscillation levels. The monthly overview helps the user choose specific days warranting further examination. Similarly, the daily overview helps the user choose particular forced oscillations to analyze. Rather than storing each occurrence of a forced oscillation separately, they are grouped by the frequency reported by the detection algorithms. With this approach, the user is able to quickly identify forced oscillations of interest from an entire month's worth of results.

To make this organization approach possible, metrics to rank each occurrence of the forced oscillations were needed. Two metrics were selected: the maximum evidence for the forced oscillation and the duration of the forced oscillation. For the periodogram method, the evidence metric refers to the estimated amplitude of the oscillation at each channel. For the spectral coherence method it refers to the value of the self-coherence at the oscillation's frequency for each channel. This value is often closely related to the oscillation's amplitude. The duration metric refers to the number of sequential minutes that a forced oscillation was detected at a particular frequency. This approach highlights two of the most important aspects of forced oscillations.

The design approaches described in this section were guided by the existing structures of DISAT and data storage, as well as observations about forced oscillations in power systems. In turn, these approaches lead to the specific implementation of the upgrade, which will be described in the following two sections.

3.0 Setup

Though much of the DISAT upgrade's functionality is fixed, there are several user-accessible parameters that determine how data is processed, where results are stored, which PMU channels are analyzed, how the detection methods operate, and more. These parameters are contained within two scripts of R code. These scripts will be discussed in detail in this section. The first script controls everything associated with the processing and analysis of data, while the second very specifically controls a plot within the application.

3.1 Data Processing

When installed at a BPA facility for testing, the main script to implement the DISAT upgrade was named *RealTimeRunDetect_Final.R*. This script includes all of the setup for processing data, commands to read and analyze data, and code to save results in Rdata files. These files are then read by the DISAT application to display results. In this section, the user-accessible parameters in *RealTimeRunDetect_Final.R* will be discussed in the order they appear in the script. The relevant portion of the script is provided in Appendix A for reference. To maintain confidentiality, PMU channel names have been obfuscated.

The first user-accessible parameters in the main script are *NumCPUsPer* and *NumCPUsSC*. Both the periodogram and spectral coherence detection methods are implemented using parallel processing to reduce computation time. The *NumCPUsPer* and *NumCPUsSC* parameters specify how many processing units are used. The user should consider the number of available processors on the computer running the code when selecting values for these parameters.

The next parameters, *DeletePastRdata* and *NumDaysRdataToKeep*, are used to ensure that the computer does not run out of storage space for PMU data. Recall that the original PDAT files are conceptually stored on a server, but once the files are converted to the Rdata format, they are stored locally. If all the Rdata files were kept indefinitely, the computer would eventually run out of storage. To avoid this, the main script is designed to delete an older Rdata file whenever a new one is created. This operation is only implemented if the logical *DeletePastRdata* is set to *TRUE*. When the operation is enabled, *NumDaysRdataToKeep* specifies how far back the Rdata file that is deleted should be. For example, if *NumDaysRdataToKeep* is equal to 365, when the Rdata file for time stamp 2015-04-28 13:45:00 is created, the Rdata file for time stamp 2014-04-28 13:45:00 will be deleted. Though deleting older Rdata files removes DISAT's ability to display the raw data, all detection results are maintained. If the user desires, the Rdata files can be recreated from the raw PDAT files by rerunning the application for the period of interest. The user may also choose to manually archive old Rdata files in a separate location to avoid their deletion.

The reporting rate of the PMU data to be analyzed is specified using the *sample.rate* parameter. The value has units of frames per second. Common frame rates are 30 and 60 frames per second.

The main script is written in such a way that the detector analyzes the PMU data chronologically. Here the file that the script expects to process next is referred to as the *current* file. As the detector is processing data, the current file may not be available, either because the detector is operating in near-real time and it has yet to be written, or because it was never written and is missing. To control the way the detector handles these situations, the parameters *sleepTimeNoDataAvail*, *numSleepsNoDataAvail*, *sleepTimeDataAvail*, and *numSleepsDataAvail* are specified. The former two parameters are used to control the case where data beyond the current minute are not available, while the latter two parameters control the case where data beyond the current minute is available. The case where “future” data is not available occurs when the detector is operating in near-real time and the next file has not yet been written to the server. In both cases, the main script behaves the same. If the current minute is not available, the script will wait *sleepTimeNoDataAvail* (*sleepTimeDataAvail*) seconds before checking for the file again. The script will check *numSleepsNoDataAvail* (*numSleepsDataAvail*) times before concluding that the file will not be made available. If future data is available, the script will jump to the next available minute of data and continue processing. If future data is not available, the script will stop running so that the issue can be addressed.

The next two user-accessible parameters specify which data files should be analyzed. The files are specified in terms of their time stamps. The *StartTime* parameter specifies the first file to be analyzed. If it is set to *NULL*, the script will process data starting one hour before the current system time. If *StartTime* is specified as a string with the time format YYYY-MM-DD hh:mm:ss, then the script will begin processing at the file with the corresponding time stamp. Similarly, if *EndTime* is set to *NULL* the script will process data indefinitely. If it is assigned a string value with the previously specified format, the script will process data until the file with the corresponding time stamp is reached.

The next set of user-accessible parameters specify file paths. Note that in the R software environment the syntax for file paths uses forward slashes (/) or double backslashes (\\) in place of the more common single backslash (\). The first parameter, *FilePathBase*, specifies the file path where Rdata files should be stored. The next parameter, *FilePathBaseRaw*, specifies the path where PDAT files are stored. Finally, *SavePathBase* specifies where detection results should be stored in Rdata files. Examples of these file paths are included in Appendix A. Note that the main script is written to accommodate the file structures used by BPA when testing the detector. Changes to this structure or installation of the software at another site would require adaptations to the main script.

Due to the multi-channel nature of the detection algorithms implemented in the DISAT upgrade, it is necessary to specify which channels should be analyzed. The main script is written specifically for real and reactive power measurements. All channels to be analyzed are specified in the list *MultiChanNamesInit*. In the naming convention utilized by BPA, real power channels are specified with “MW” as the last two characters in the channel name and reactive power channels are specified with “MV” as the last two characters in the channel name. If corresponding channels of real and reactive power are both specified, the main script automatically combines these channels to be analyzed as apparent power instead. In this case, the “MW” and “MV” in the new channel name is replaced with an “S” in the tool’s user interface.

The remaining parameters in the main script are all related to the detection algorithms. Changes to these parameters should be made with caution and with a firm understanding of the detection algorithms discussed in (Follum 2015 B), which also describes the various parameters. Here, only a few comments on the most relevant parameters are provided. The parameters *fmin* and *fmax* specify the limits for the

frequency range examined for forced oscillations. This range should account for the settings of PMUs providing data. The performance of the periodogram method can be adjusted with the *Pfa* parameter, which sets the theoretical probability of false alarm. Small values will provide few false detections, but forced oscillations will generally need to be larger or longer lasting to be detected. Large values will have the converse effect. Similarly, *gam_SC* impacts the performance of the spectral coherence algorithm. Large values will lead to fewer false detections and a decrease in the detector's ability to detect small oscillations. Finally, *NumDelay* specifies the number of delayed windows to examine. For values greater than two, parallel processing is used to compute the Generalized Magnitude Squared Coherence (GMSC), rather than the spectral coherence. Generally, detection performance will improve with more delays at the cost of an increased detection delay and a significant increase in computational burden.

3.2 Mapping

Inside the DISAT application, maps are used to display information about detected forced oscillations. These maps will be described in more detail in Section 4. Here, only the script used to set up the maps is described. This script is named *SCmapInfo.R* and is included in Appendix B for reference. To maintain confidentiality, sensitive information has been obfuscated.

The first user-accessible parameter is *MapInfoPath*. This parameter specifies the folder containing *WECCmap.RData*, which is a data file containing the information for a map oriented to the Pacific Northwest. The script contains comments explaining how to generate this data file, but there should be no need as it is installed along with the rest of the DISAT upgrade.

The next three parameters are related to the display of PMUs on the map. First, *SubNames* is a vector of strings containing the four letter abbreviations for PMUs used in the PDAT format. Though PMUs not included in this list can be used for analysis, they will not be displayed on the map in the application. The location of each PMU is specified by latitude and longitude in *SubLats* and *SubLons*, respectively.

If data from a PMU becomes available, *SCmapInfo.R* can be run to add it to the maps. The PMU's abbreviation, latitude, and longitude must simply be added to the lists and the script re-run. The remainder of the script will save the necessary files to provide the DISAT application access to the updated PMU information.

4.0 Application

To display results of the forced oscillation detection analysis, an addition was made to the DISAT application. The application organizes all of the detected forced oscillations and provides a variety of visualizations to aid in the verification of the oscillation's existence and analysis of its behavior. In this section, each of the tables and visualizations will be described with real-world examples from BPA data. To protect confidentiality, channel names have been obfuscated in several of the figures.

After clicking on the *Forced Oscillations* tab within the main DISAT viewer (see Figure 1), the side panel and subtabs depicted in Figure 2 become available. To begin, the options in the side panel, which are available for all three subtabs, will be described. The *Select algorithm* dropdown has two options: 1) Periodogram and 2) Spectral Coherence. Results from the two detection methods are saved and displayed

separately, but this dropdown makes it simple to transition between the methods. The next dropdown allows the user to choose between the evidence and duration metrics for ranking forced oscillations described in Section 2.4. The tables in the monthly and daily overview subtabs, which will be described shortly, are organized based on this selection. Note, though, that the *Select Significance Criterion* field impacts more than organization; it determines which forced oscillations are significant enough to appear in the table. The final two fields allow the user to specify the year and month for which results should be displayed. The *Select month* dropdown is automatically populated based on available results. Once the user has specified all the fields in the side panel, clicking the *Update* button will populate the monthly overview table in the main panel.

As described in Section 2.4, the application was designed using a drill-down approach. The highest level of analysis is the month level. The table in the *Monthly Overview* subtab has an entry for each day in the month for which data was analyzed. The entry for each day corresponds to the most significant forced oscillation detected during the day, depending on the selection in the *Select Significance Criterion* dropdown. The most significant events from each day are again ranked in the monthly overview table. With this approach, the user can quickly differentiate between days containing forced oscillations that require further investigation and those that do not. The table can also be reorganized based on any value in the table by clicking on the arrows next to the names of table headings. When the forced oscillations detected during a day require further investigation, the *Daily Overview* subtab can be used to drill down further.

After selecting the *Daily Overview* subtab, the *Select day* field is added to the application's side panel. This dropdown is automatically populated with days containing results for the specified month. To bring up the daily overview table, click the *Update* button in the side panel. An example of the application's appearance when the *Daily Overview* subtab is selected is presented in Figure 3. In this table, the forced oscillations are grouped by frequency. Throughout each day, the analysis tracks each forced oscillation as it appears and disappears in the system based on its frequency. Once again, the metric used to rank significance determines which oscillations are displayed in the table as well as how the table is sorted. The user can quickly scan from the top of the table to identify which forced oscillations warrant further investigation. These oscillations can then be examined in the *Inspect Oscillation* subtab.

BPA DISAT Analysis Software

[Home](#)
[Atypicality Exploration](#)
[Historical Analysis Results](#)
[Raw Data Viewer](#)
[Cluster Membership](#)

[Data Quality Reports](#)
[Forced Oscillations](#)

[Monthly Overview](#)
[Daily Overview](#)
[Inspect Oscillation](#)

Show entries Search:

Date	Peak Amplitude (MVA)	Duration (min)	Frequency (Hz)	Time of Peak (hh:mm)	Channel with Peak
2014-04-04	3.545	14	0.31	07:47	
2014-04-15	3.396	18	0.31	12:17	
2014-04-02	3.009	13	0.31	02:40	
2014-04-18	1.105	71	0.50	10:56	
2014-04-05	1.073	2	0.27	09:18	
2014-04-19	1.000	38	0.50	10:59	
2014-04-20	0.990	66	0.50	11:15	
2014-04-01	0.950	125	16.67	22:06	
2014-04-21	0.936	96	0.50	07:26	
2014-04-06	0.901	42	0.50	06:38	
2014-04-17	0.868	67	0.50	09:39	
2014-04-16	0.858	40	0.49	09:46	
2014-04-22	0.819	75	0.50	18:55	
2014-04-03	0.577	4	1.04	22:44	

Showing 1 to 14 of 14 entries Previous **1** Next



Pacific Northwest NATIONAL LABORATORY
 Proudly Operated by **Battelle** Since 1965
BONNEVILLE POWER ADMINISTRATION

Figure 2: Example of the *Monthly Overview* subtab in the DISAT application.

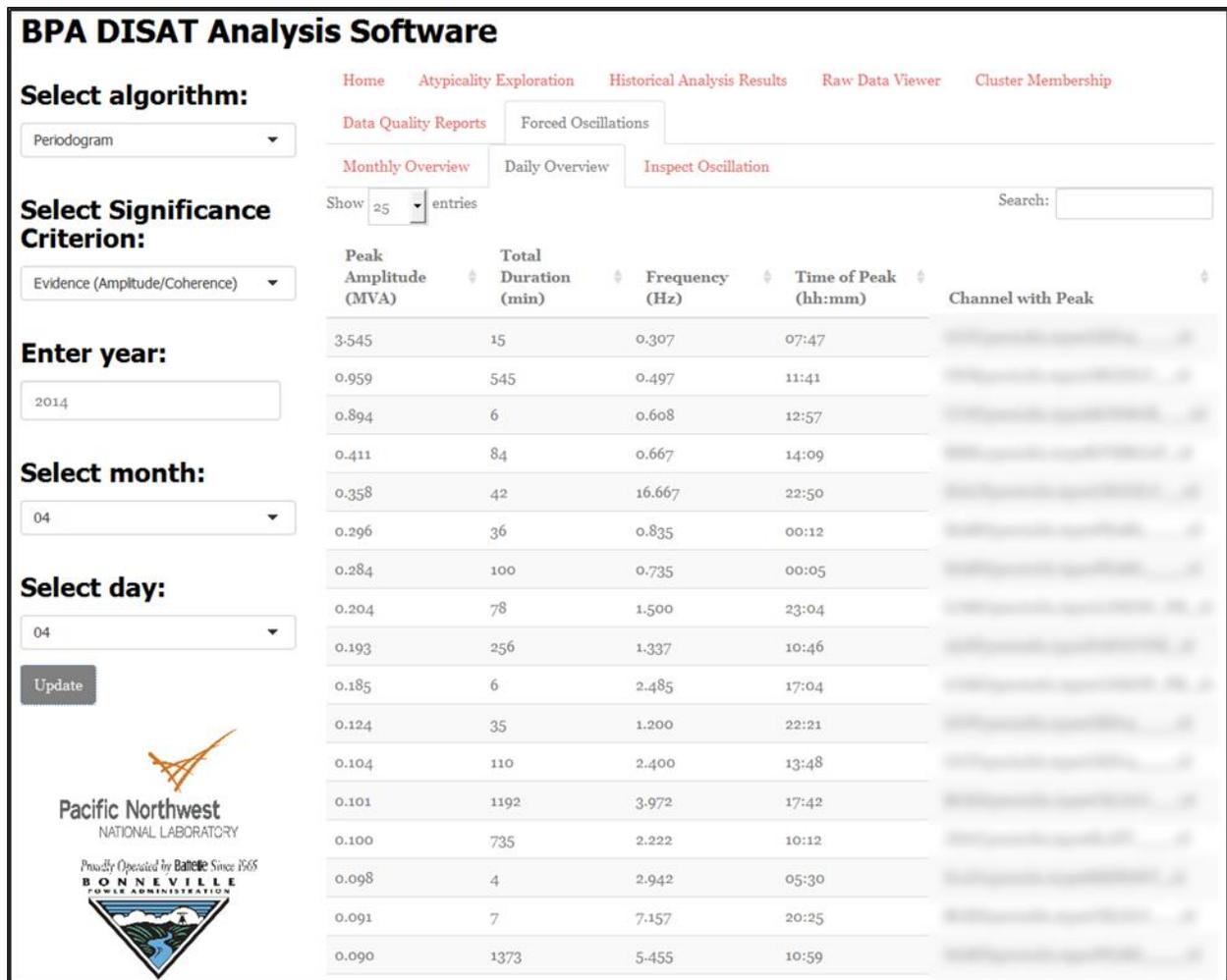


Figure 3: Example of the *Daily Overview* subtab in the DISAT application.

Upon selecting the *Inspect Oscillation* subtab, the *Enter Frequency* field appears in the side panel, as can be seen in Figure 4. By entering the frequency of a forced oscillation that occurred on the specified day and clicking the *Update* button, the user brings up a table such as the one displayed in Figure 4. This table specifies the start and end times for each occurrence of the forced oscillation throughout the day. This table is organized chronologically.

Below the table, several visualizations are available. The *Time to Plot* field allows the user to specify which occurrence of the forced oscillation to examine. For the table in Figure 4, clicking on the *Time to Plot* dropdown lists the options in Figure 5. Note that the list is automatically populated with the times that the forced oscillation amplitude or spectral coherence peaked during each of the oscillation's occurrences. This approach allows the user to quickly examine the consistency of results for each of the oscillation's occurrences, but results from other minutes can also provide insight. To specify any time of the user's choosing, the *Time to Plot* field's *Select Minute* option may be selected, which will result in the extra field and buttons displayed in Figure 6. After the user enters a valid time, the visualizations will automatically appear. To "walk" through results chronologically, the *Previous Minute* and *Next Minute* buttons can be used to update the selected time. Finally, the user can also view a special visualization for a range of minutes by choosing the *Select Range* option in the *Time to Plot* dropdown. Once this option is

selected, new fields to enter the start time and end time of the analysis are provided, as shown in Figure 7. Populating these fields with valid times will automatically produce the visualization.

BPA DISAT Analysis Software

Home Atypicality Exploration Historical Analysis Results Raw Data Viewer Cluster Membership

Data Quality Reports Forced Oscillations

Monthly Overview Daily Overview Inspect Oscillation

Show 25 entries Search:

Select algorithm:
Periodogram

Select Significance Criterion:
Evidence (Amplitude/Coherence)

Enter year:
2014

Select month:
04

Select day:
04

Enter Frequency:
16.667
Update

Onset Time (hh:mm)	End Time (hh:mm)	Time of Peak (hh:mm)	Duration (min)	Peak Amplitude (MVA)	Frequency (Hz)	Channel with Peak
21:57	22:14	22:07	18	0.221	16.668	
22:19	22:29	22:23	11	0.205	16.668	
22:33	22:33	22:33	1	0.068	16.667	
22:43	22:54	22:50	12	0.358	16.667	

Showing 1 to 4 of 4 entries

Time to Plot: 22:07

Map
 Test Statistic
 Detections Per PMU
 Time Domain

Figure 4: Example of the *Inspect Oscillation* subtab in the DISAT application.

Time to Plot:

22:07

- 22:07
- 22:23
- 22:33
- 22:50
- Select Minute
- Select Range

Figure 5: Example of the available options in the *Time to Plot* field.

Time to Plot:

Select Minute ▼

Enter time as hh:mm

Previous Minute 22:00 Next Minute

Figure 6: Example of using the *Select Minute* option to specify a minute for analysis.

Time to Plot:

Select Range ▼

Start time as hh:mm End time as hh:mm

21:50 23:00

Figure 7: Example of using the *Select Range* option to specify a range of minutes for analysis.

The DISAT application includes five visualizations to analyze detected forced oscillations. Four of these visualizations are for the analysis of individual minutes and one is used to display results over a range of minutes. These visualizations will now be described, starting with the four visualizations for results from single minutes.

The checkboxes below the *Time to Plot* field in Figure 4 are used to select which of the four visualizations are displayed. When the *Map* checkbox is selected, maps, such as the examples in Figures 8 and 9, are displayed. Each dot on the maps indicates the location of a PMU specified in the *SCmapInfo.R* setup script. Black dots indicate PMUs whose data could not be used because of data quality issues. The size of each dot corresponds to the amount of evidence, either amplitude or spectral coherence value depending on the selected detection algorithm, for the detected forced oscillation in the PMU's data. Grey dots are used to indicate PMUs where the oscillation was too small for its amplitude to be estimated using the periodogram method. The colors of the dots provide a rough indication of the oscillation's phasing across the system. The empirical transfer function method (Follum et al. 2011) was used to generate the estimates. Though these maps can be useful in identifying areas of interest, they are not intended to pinpoint the source of an oscillation. For example, the PMU identified by the large red dot in Figure 8 is likely a good place to start an investigation, but due to the system's dynamics, this map does not provide proof that the oscillation is originating at the PMU's location. As another example, the forced oscillation in Figure 9 is known to have originated near a PMU in the middle of the system, rather than close to the PMUs at the southern end of the system where the estimated amplitude was largest.

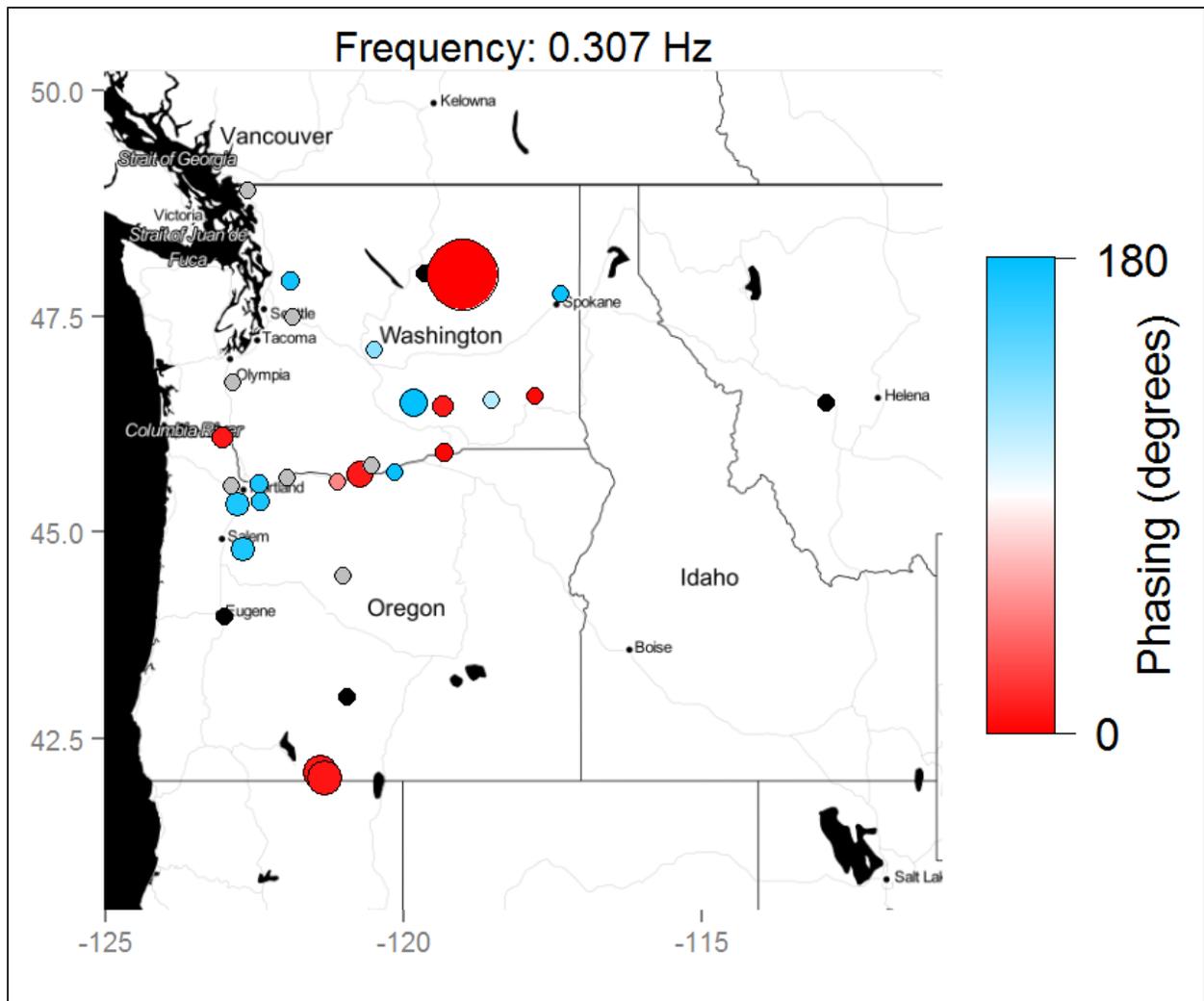


Figure 8: Example map visualization for an oscillation that had strong evidence at a single location.

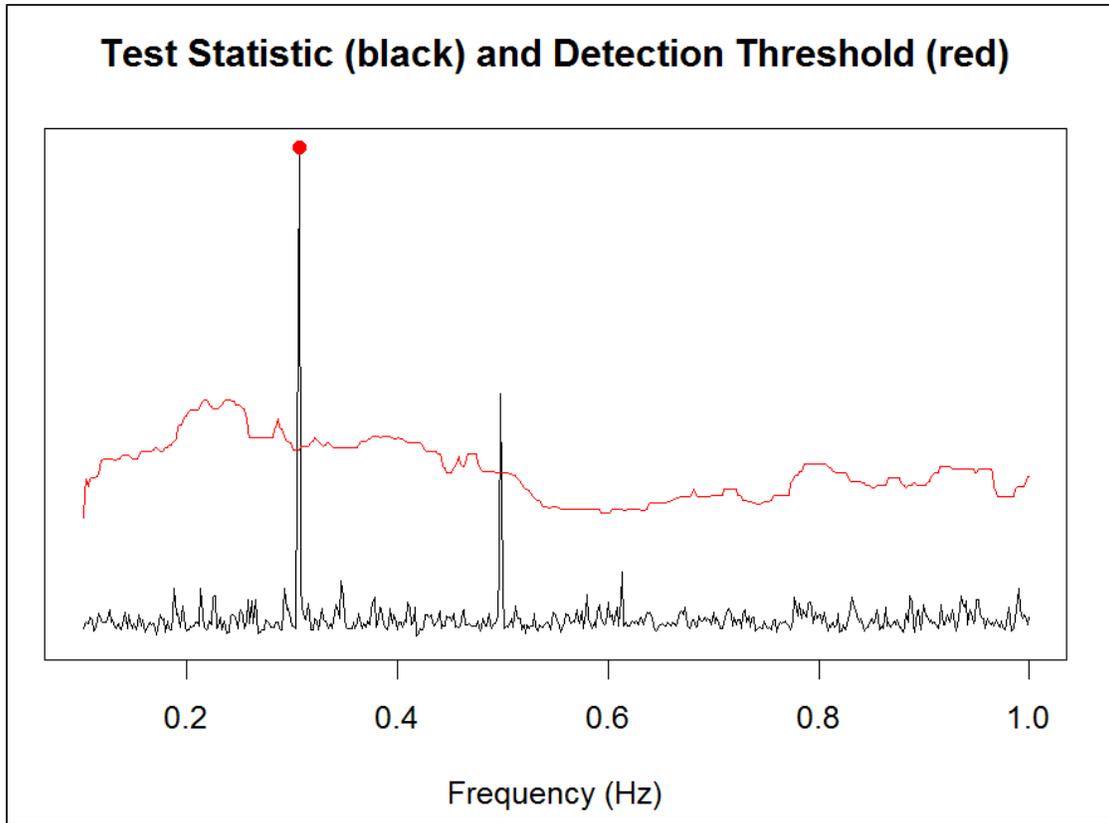


Figure 10: Example test statistic visualization for the periodogram method.

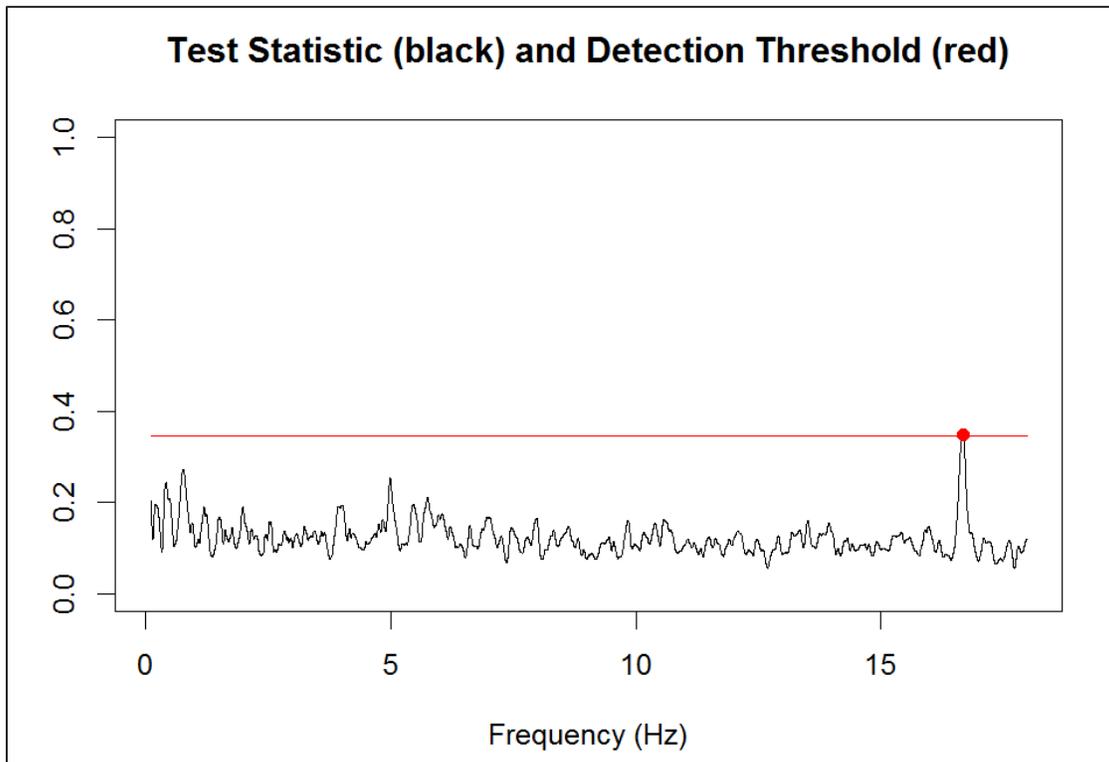


Figure 11: Example test statistic visualization for the spectral-coherence method.

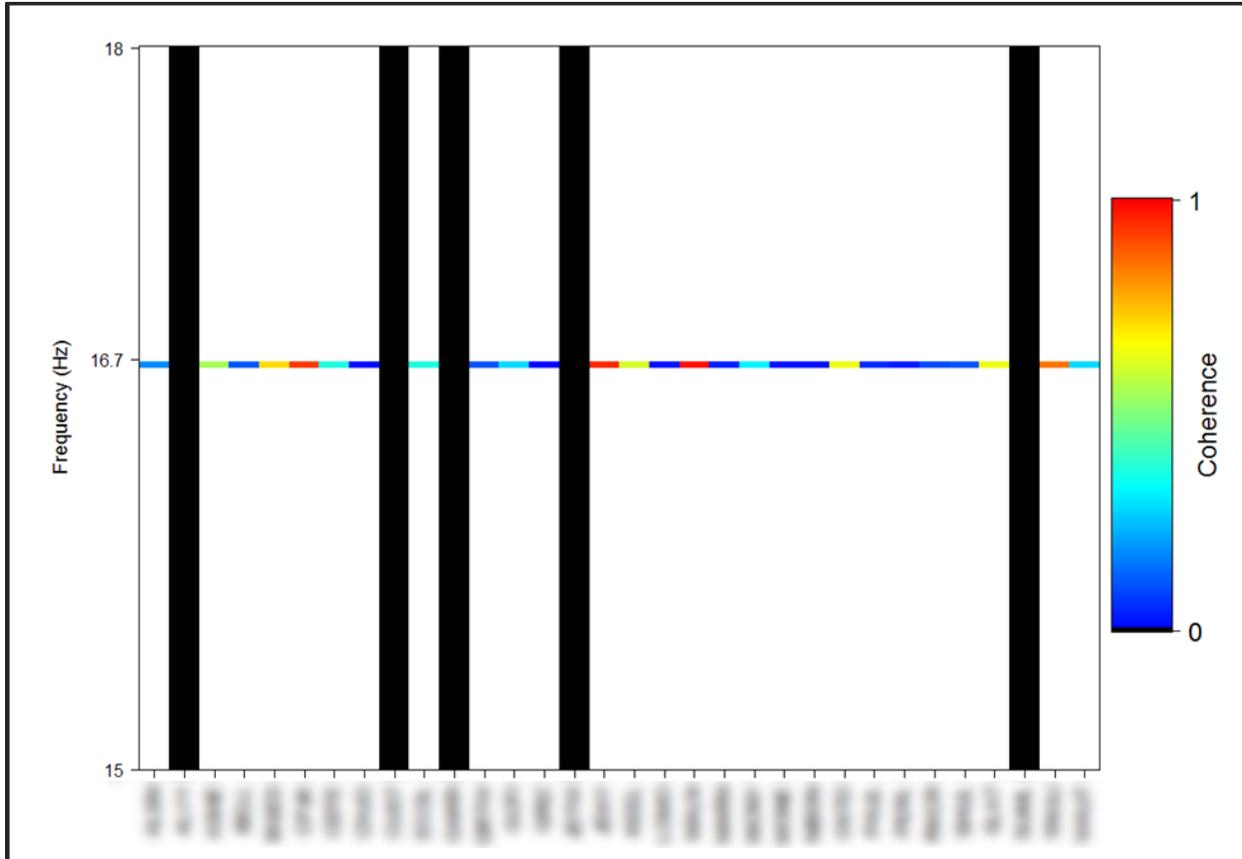


Figure 13: Example of the *Detections Per PMU* visualization for the spectral coherence method. Compare with Figures 9 and 11.

The final visualization for single minute results is a time-domain plot. Many forced oscillations are too small to be apparent in time-domain data above the ambient power system noise, but not always. For example, consider the obvious oscillation in Figure 14, which presents one minute of measurements corresponding to the results in Figures 8, 10, and 12. Though now obfuscated, the channel selected in the dropdown box corresponds to the PMU with the largest amplitude in Figure 12. While in the case of Figure 14 the plot helps to validate the detection, the time-domain visualization can also help identify false detections due to data quality problems. For example, consider Figure 15, which makes the two sets of drop-outs during a minute that led to a false detection apparent. A data quality filter is included in DISAT to remove such unreliable data, but in this case the filter failed. By quickly examining this plot, the user can determine that the “detected” oscillation is not an oscillation at all and warrants no further investigation. Similarly, the time-domain plots can be used to identify false alarms resulting from major system events such as the tripping of a generator.

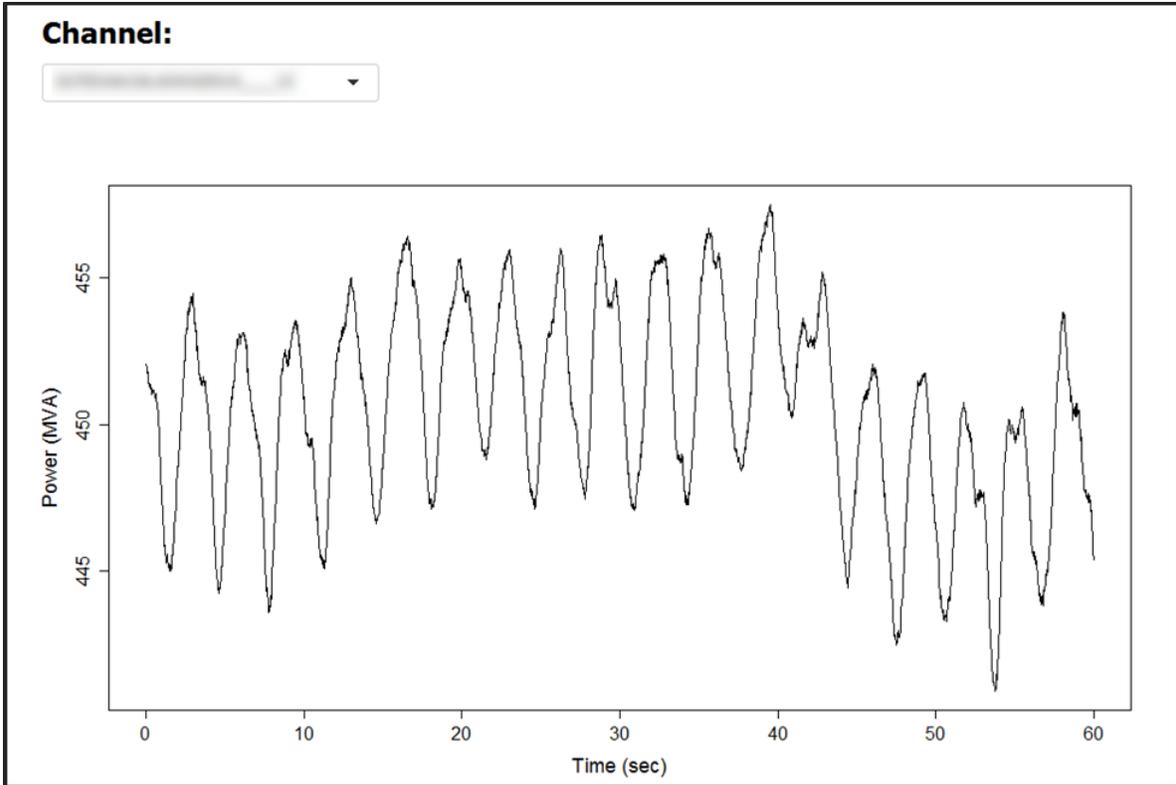


Figure 14: Example of a time-domain visualization showing a large forced oscillation.

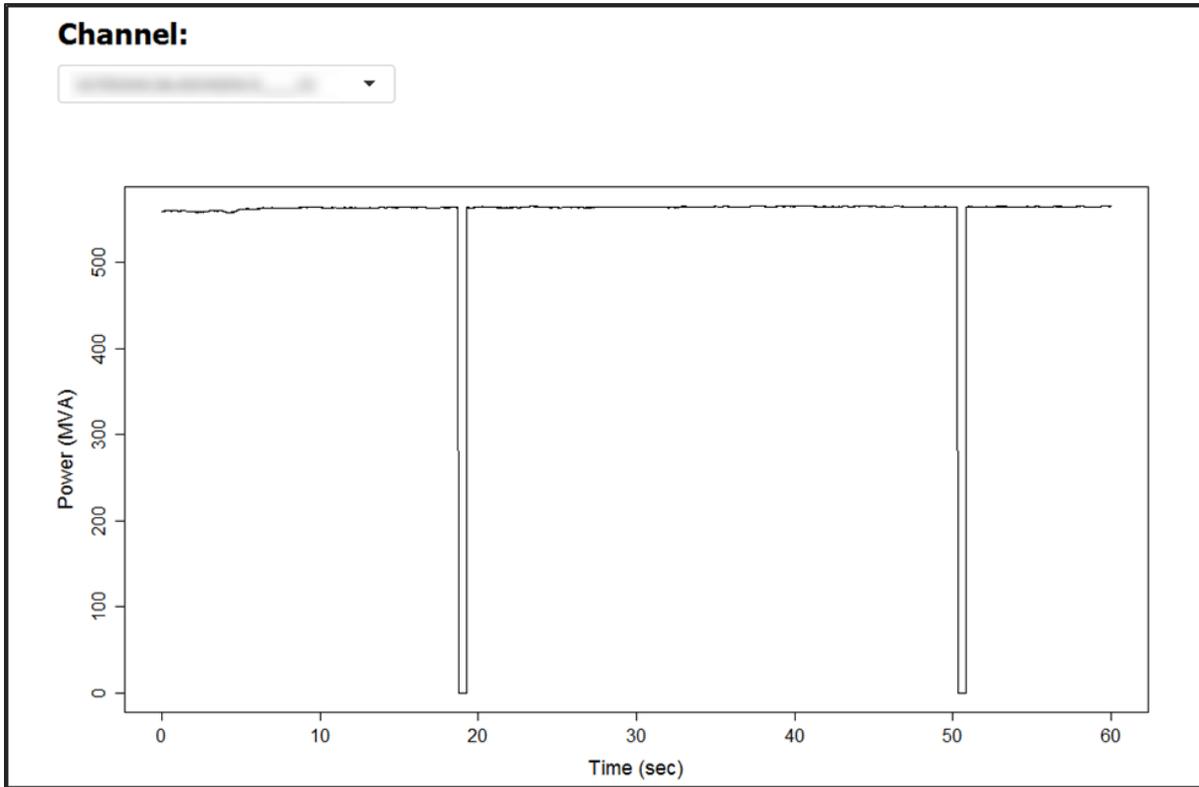


Figure 15: Example of a time-domain visualization with obvious data quality problems.

The final visualization in DISAT for the analysis of forced oscillations allows a range of minutes to be analyzed. Recall that after choosing the *Select Range* input in the *Time to Plot* dropdown, fields for the start and end time of the analysis become available (see Figures 4 and 7). In most cases, the selection of this range should be guided by the table in the upper portion of the main panel (again, see Figure 4). When a forced oscillation is detected on separate occasions, it may be useful to know what occurred between detections and more precisely when the oscillation began. For these purposes, a frequency- and time-dependent visualization of each method's test statistic was included in the DISAT upgrade. An example is provided in Figure 16 with an accompanying table in Figure 17. Note that the y-axis in Figure 16 corresponds to frequency and can be scaled using the slider in the upper-left corner of the plot. The checkbox just below the slider lets the user choose whether or not to only display values where a forced oscillation was detected. In Figure 16, the five occurrences of the forced oscillation listed in Figure 17 are apparent. With this plot, though, it is not possible to determine if the oscillation is still present between detections. Deselecting the checkbox leads to a plot that includes values for all frequencies and times, such as the one presented in Figure 18. In this plot, it is clear that the forced oscillation does not altogether stop. Rather, the peak in the test statistic is not large enough to cross the detection threshold. This information may be valuable when investigating the oscillation further. Also of note in Figure 18 is the presence of a frequency component just below 0.6 Hz that was too small to be detected during this one-hour period. The two versions of this visualization are intended to be complementary and aid the user in understanding the oscillation's behavior.

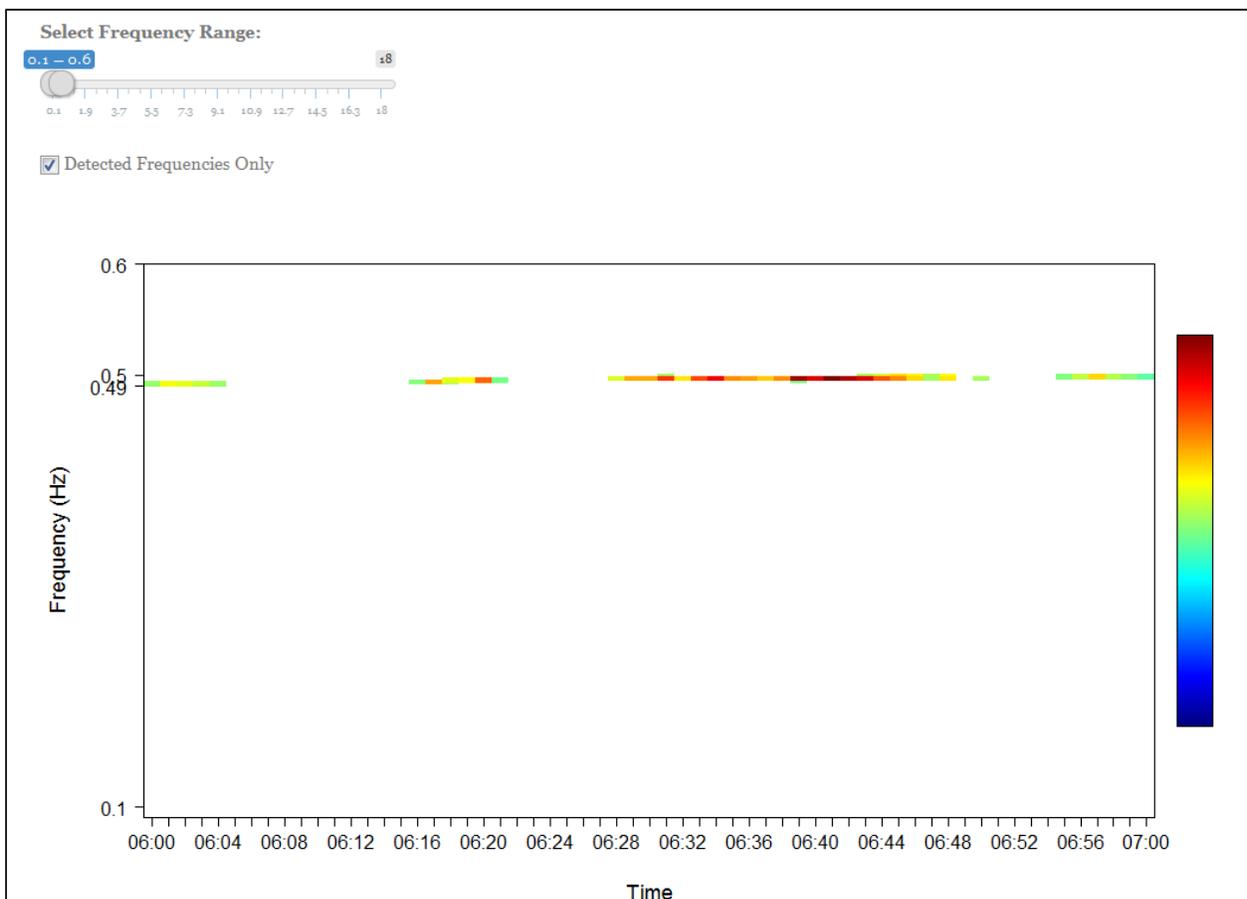


Figure 16: Example plot of the periodogram method's test statistic plotted over time with only values corresponding to detected oscillations included.

Onset Time (hh:mm)	End Time (hh:mm)	Time of Peak (hh:mm)	Duration (min)	Peak Amplitude (MVA)	Frequency (Hz)	Channel with Peak
06:00	06:04	06:01	5	0.546	0.492	
06:16	06:21	06:20	6	0.611	0.495	
06:28	06:48	06:48	21	0.747	0.497	
06:50	06:50	06:50	1	0.724	0.497	
06:55	07:11	06:57	17	0.728	0.500	

Figure 17: Portion of the table on the *Inspect Oscillation* subtab corresponding to the times included in Figures 16 and 18.

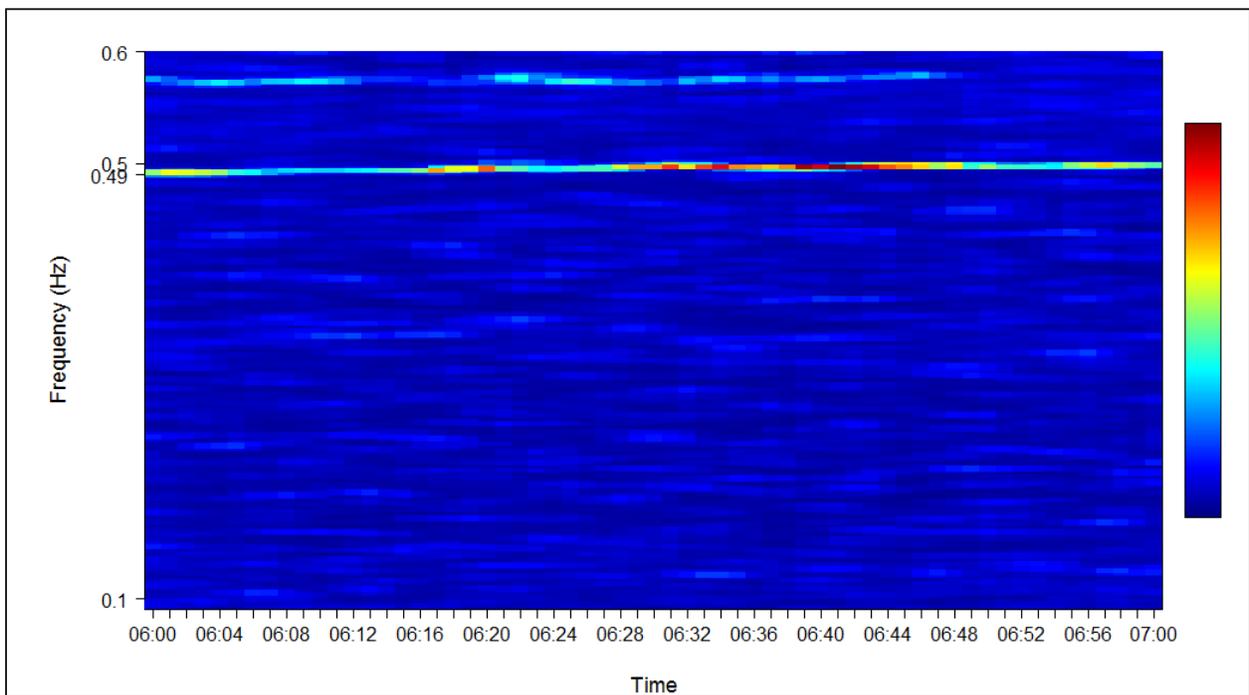


Figure 18: Example plot of the periodogram method's test statistic plotted over time with values for all frequencies and times included.

Together, the tables and visualizations in the DISAT application allow the user to quickly identify forced oscillations of interest and to complete an initial examination. Important parameters including amplitude, frequency, phase, and geographic location are all represented. Test statistics in the frequency domain and raw measurements in the time domain are included to aid in verifying the accuracy of the detection. These visualizations are designed to provide several pieces of important information individually while still complementing each other. With their coordinated use, a great deal can be learned about those forced oscillations that warrant further investigation.

5.0 Conclusion

Over the past several years, there has been an increasing interest in forced oscillations in power systems. The recognition that these oscillations often arise due to equipment malfunction or misoperation, along with concern over how they may lead to improper operator responses, has led system operators to seek out methods of analyzing forced oscillations with synchrophasor data. While several detection and analysis algorithms have been developed, there remains a need for tools that can provide near-real time results in an accessible manner. Still in the prototype stage of development, the forced oscillation components of DISAT are a first step towards the production of such a tool.

The DISAT tool applies two separate multi-channel algorithms to PMU data and is able to operate in near-real time. The use of multi-channel algorithms provides improved detection performance and simplifies the organization of detection results. After the oscillation is detected, important information is gathered for each available channel of PMU data. Detection results are then organized and ranked to allow the user to quickly identify those oscillations requiring further investigation. Initial steps to this investigation are supported by the array of visualizations included in the DISAT application.

Future work on the forced oscillation components of DISAT will focus on improving computational speed, adjusting existing visualizations, and adding new visualizations to provide the user with further information. Because the source of the oscillation is of such high interest to power system engineers, methods for identifying the source will also be pursued. By building on the existing algorithms, visualization, and organization structure, a useful tool can be developed and deployed for practical use.

6.0 References

- (Amidan et al. 2014) Amidan BG, JD Follum, KA Freeman, JE Dagle. 2014. *An Initial Application of DISAT (Data Integrity Situational Awareness Tool, Final Report PNNL-23981*. Pacific Northwest National Laboratory, Richland, WA.
- (Follum et al. 2015 A) Follum JD, FK Tuffner, and BG Amidan. 2015. *Integration of a Self-Coherence Algorithm into DISAT for Forced Oscillation Detection, Final Report PNNL-24127*. Pacific Northwest National Laboratory, Richland, WA.
- (Follum 2015 B) Follum JD. 2015. *Detection of Forced Oscillations in Power Systems with Multichannel Methods, Final Report PNNL-24681*. Pacific Northwest National Laboratory, Richland, WA.
- (Follum et al. 2011) Follum J, L Dosiek, and JW Pierre. 2011. *The Empirical Transfer Function Estimate Method for Determining Mode Shape*. North American Power Symposium (NAPS), Boston, MA.
- (R Core Team 2015) R Core Team. 2015. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

- (Chang 2015) Chang W, J Cheng, JJ Allaire, Y Xie, and J McPherson. 2015. shiny: Web Application Framework for R. R package version 0.12.2. <http://CRAN.R-project.org/package=shiny>.
- (IEEE 2011) IEEE. 2011. *IEEE Standard for Synchrophasor Data Transfer for Power Systems, IEEE Std C37.118.2-2011 (Revision of IEEE Std C37.118-2005)*.

Appendix A

User-Chosen Parameters in RealTimeRunDetect_Final.R

```
12 ### User chosen parameters
13
14 # Number of CPUs to use in parallel processing
15 NumCPUSPer = 4
16 NumCPUSC = 4
17
18 # To keep the amount of Rdata that is stored on the computer reasonable, this script can delete past files.
19 # Past R data files are only deleted if DeletePastrdata == TRUE
20 # NumDaysRdataToKeep specifies how much Rdata to store. For example, if DeletePastrdata is TRUE and
21 # NumDaysRdataToKeep=365, whenever an Rdata file is loaded the Rdata file from one year previous
22 # will be deleted.
23 DeletePastrdata = TRUE
24 NumDaysRdataToKeep = 18*30
25
26 sample.rate=60
27
28 # How long to sleep and number of times to sleep when no data beyond the current minute is available
29 # If the number of sleeps is exceeded, all processing will stop
30 sleepTimeNoDataAvail = 60
31 numSleepsNoDataAvail = 60*16.5
32
33 # How long to sleep and number of times to sleep when data beyond the current minute is available
34 # If the number of sleeps is exceeded, the algorithm will move beyond the current minute to the next available minute of data
35 sleepTimeDataAvail = 60
36 numSleepsDataAvail = 60*16.5
37
38 # The start and end times (inclusive) to perform the analysis
39 #
40 # Make sure to take into account how much data needs to be stored before the
41 # algorithms will start processing, i.e., if N=10*60*sample.rate (defined
42 # later in the initialization) and StartTime is "2016-01-29 00:00:00" then the
43 # first detection results will be recorded for "2016-01-29 00:09:00".
44 #
45 # If StartTime is NULL, the script will process data as it becomes available
46 # starting one hour before the current system time.
47 # If Endtime is NULL, the code will continue to run until stopped manually (hit
48 # escape in the R Console window).
49 #
50 StartTime = NULL
51 EndTime = NULL
52 # StartTime = "2016-01-29 19:10:00"
53 # EndTime = "2016-01-29 19:24:00"
54
55 # Filepath to where Rdata files converted from PDAT are
56 FilePathBase = "D:/DISAT_PDAT/Rdata_PDAT/"
57
58 # Filepath to where raw pdat files are
59 FilePathBaseRaw = "J:/"
60
61 # Filepath to save outputs from test
62 SavePathBase = "D:/DISAT_PDAT/output_FOdet/"
63
64 # These are the names of the MW and MVAR channels to be analyzed
65 # Multiple channels from a single PMU can be used, but plots will suffer. Also keep the computational burden in mind.
66 # If MW and MVAR (abbreviated MV in PMU naming convention) for the same signal are included, they will automatically
67 # be converted to apparent power and denoted with an S in place of the MW and MV in the channel name.
68 MultichanNamesInit = c(
69     ,
70     ,
71     ,
72     ,
73     ,
74     ,
75     ,
76     ,
77     ,
```

```

78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137 # The following parameters are for the detection algorithms.
138 # They are described in detail in the user's manual.
139 # Changes to these values should be made with caution.
140
141 N = 10*60*sample.rate # Amount of data stored. Needs to be an integer multiple of 60*sample.rate, the length of each PDAT file
142
143 # Parameters for both algorithms
144 K0 = N
145 FreqTol = 0.05
146
147 # minimum and maximum frequencies to look for forced oscillations
148 fmin = 0.1
149 fmax = 18
150
151 # Frequency bin stuff
152 f = sample.rate*(0:(K0-1))/K0
153 omegaB = which((f>fmin) & (f<fmax))
154 f = f[omegaB]
155 B = length(f)
156
157 # Periodogram parameters
158 Nper = N
159 Pfa = rep(0.000001,B)
160 win_Per = hamming(Nper)
161 win_GMSC = hamming(Nper/3)
162 Noverlap_GMSC = length(win_GMSC)/2
163 Nmed = 37
164
165 # SC parameters
166 Nsc = 50*sample.rate
167 D = 10*sample.rate
168 NumDelay = 3
169 gam_SC = 3
170 win_sc = hamming(Nsc/5)
171 Noverlap_sc = length(win_sc)/2
172
173
174 ### END user choices

```

Appendix B

SCmapInfo.R

```
1 MapInfoPath = "C:/Users/fo11154/Documents/Spectral Coherence FY15/ApplicationToPMUdata/Output/Maps/"
2
3 subNames = c("ALSN", "ALVY", "ASHE",
4             "BELL", "BGED", "BIGE",
5             "CPJK", "CEFE", "CHJO", "CUST",
6             "DOJN", "DOOL",
7             "ECOL",
8             "GARR", "GRIZ", "GCFI", "GCTW", "GRTW",
9             "JDAY", "JDTW",
10            "KEEL", "KNIT",
11            "LOMO",
12            "MALN", "MARN", "MCNY", "MONE",
13            "NBON",
14            "OSTD",
15            "PAUL", "PERL",
16            "RKCR",
17            "SHUL", "SLAT", "SUML",
18            "TROU", "TAFT",
19            "WAUT")
20
21 subLats = c(
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39 subLons = c(
40
41
42
43
44
45
46
47
48
49
50
51
```

```
52
53
54
55
56
57 # The following 3 lines get the map from the internet. Because an internet connection
58 # may not be available, the map is saved to an RData file for retrieval. To change
59 # the map, connect to the internet and run the code below, which should normally be
60 # commented out. Alternatively, run these lines on a separate computer that is connected
61 # to the internet and move the resulting RData file to the computer running DISAT.
62 #
63 # library(ggmap)
64 # WECCmap <- get_map(location = c(lon = -118, lat = 45.5),mptype="toner", zoom = 6)
65 # save(WECCmap=WECCmap, file = paste(MapInfoPath,"WECCmap.RData",sep=""))
66 #
67 # This line simply loads the map that is saved in the above lines of code. Saving the map
68 # and loading it here makes it so the computer running DISAT does not need an internet
69 # connection to retrieve WECCmap every time this script is run.
70 #
71 load(paste(MapInfoPath,"WECCmap.RData",sep=""))
72
73
74 save(SubNames=SubNames, SubLats=SubLats, SubLons=SubLons, WECCmap=WECCmap,
75      file = paste(MapInfoPath,"SCmapInfo.RData",sep=""))
```

Distribution

<u>No. of Copies</u>	<u>No. of Copies</u>
<p>1 Philip Overholt U.S. Department of Energy 1000 Independence Ave. SW OE-40 Washington, DC 20585 (PDF)</p> <p>1 Joseph Eto 1 Cyclotron Road MS 90R4000 Berkeley, CA 94720 (PDF)</p> <p>1 Anthony Faris Bonneville Power Administration P O Box 3621 Portland, OR 97208-3621 (PDF)</p>	<p>5 Local Distribution Pacific Northwest National Laboratory Eric Andersen (PDF) Jeff Dagle (PDF) James Follum (PDF) Brett Amidan (PDF) Francis Tuffner (PDF)</p>



Pacific Northwest
NATIONAL LABORATORY

*Proudly Operated by **Battelle** Since 1965*

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)

U.S. DEPARTMENT OF
ENERGY

www.pnnl.gov