Generator Model Verification Result
Analysis System

ECE 4902, Spring 2020

Cynthia Bissereth, ECE
Andrew Collins, CSE
Jonathan Davis, ECE/CSE
Eric Flower, CSE
Sam Markelon, CSE
Ethan McRae, ECE

ISO-NE
Angela Gaspar
agaspar@iso-ne.com
One Sullivan Road,
Holyoke, MA
# Table of Contents

Abstract 3  
Introduction 3  

**Project Statement** 4  
  Statement of Need 4  
  Preliminary Requirements 5  
  Basic Limitations 5  
  Other Data 5  
  Additional Information 5  

**Approach and Design** 6  
  Theory 6  
  Model 10  
  Constraints 10  
  Standards 11  
  Final Design 11  
  Experimental or Simulation Testing Results 11  

**Project Management** 18  
  RACI Chart 18  
  Part List 18  
  Budget List 18  
  Gantt Chart 18  

**Summary** 19  

**Figures and Diagrams** 20  
  Figure 1: RACI Chart 20  
  Figure 2: Gantt Chart Phase 1 20  
  Figure 3: Gantt Chart Phase 2 21  
  Figure 4: Gantt Chart Phase 3 21  
  Figure 5: Gantt Chart Phase 4 21  
  References 22  

**Appendices** 23  
  Sinusoidal Testing for Signal Similarity Code 23  
  Senior Design Project Checklist 26  

**Glossary** 28
Abstract

In collaboration with ISO New England (ISO-NE) and the Computer Science and Engineering group, we are working towards improving the Power Plant Model Verification Tool (PPMV) that the company uses to evaluate their models for the Bulk Power System (BPS), as well as to keep track of disturbances and outliers within the system. Presently, ISO-NE Seeks to improve their models to allow for more accurate predictions of how the BPS will react to various faults. A scoring engine is therefore needed to determine model accuracy with respect to the actual Bulk Power Systems. This paper will discuss the new and improved methods that the ECE team has implemented in order to improve the metrics to be used within this tool with the intent of assisting the engineers at ISO-NE of determining deficiencies within their models. Our implementation scores models by comparing their similarity to measured data in terms of magnitude, phase angle, and correlation. This paper will also discuss advantages of our system that will allow the engineers at ISO-NE to iteratively refine their models over time.

Introduction

In any engineering context, it is valuable to be able to model and simulate a system to both understand and predict its behavior. System modeling is both significantly cheaper and safer than performing real-world tests, as puts no physical hardware or services at risk should something go wrong. However, the primary disadvantage of modelling is accuracy. Depending on the target system, a model's response can deviate from the actual system response because it is extraordinarily difficult to account for all variables that affect a physical system. Furthermore, it is possible that a model appears to behave well in some contexts, but deviates wildly in others. For example, a model may appear to closely predict the response of a physical system at some frequencies, but become out of phase in others. It is thus desirable that a quantitative means of determining the accuracy of a model is developed. By creating a quantitative scoring system that compares a model's output response to a measured system response, the process of evaluating and ranking models can be automated, saving engineers time and ensuring consistent evaluations across all models. The challenge with this project is determining such a means of providing reasonable scoring of system models. In essence, such a system must be able to distill the complex response of an entire system across a range of frequencies to a single number (score), do so automatically, and provide a sound method of determining said score.
Project Statement

Statement of Need

At ISO New England, there are numerous power system dynamic models that need to accurately represent the actual behavior of the bulk power system (BPS). The importance of these models is to help in planning to identify and mitigate potential criteria violations, determine transfer capability, and develop transmission system reinforcement plans. The models consist of many components and their baseline model structure and parameters are derived from the manufacturer data and field-testing during initial commissioning. North American Electric Reliability Corporation (NERC) requires that a periodic reverification of the generator dynamic models is conducted to ensure that these models reflect a reasonable representation of the equipment in the field.

The Power Plant Model Verification (PPMV) is based on dynamic disturbance recording data from sensors such as phasor measurement units (PMUs) which can serve as a recurring test to ensure model accuracy. A playback simulation is done from the measurements and is compared to real data to determine whether the modeled response to the system events matches the actual response of the power plant. ISO-NE created an online tool, Automatic Power Plant Model Verification (APPMV), that runs as a service 24/7 to automatically perform the task of power plant model verification using real-time power system disturbances. In the past, this tool has accumulated substantial results. The problem at hand is finding a method to manage, analyze, and quantify these results automatically.

The main requirement of this project is to determine a method to manage, analyze, and quantify the results from the APPMV tool. This will need to be done with the development of a database (Task 1) and a user interface and smart engineering analysis engine which can help ISO-NE effectively perform the model verification task (Task 2). With regards to the database and web-based GUI design and implementation, the team needs to understand the overall structure of the ISO-NE developed APPMV tool and its results. With that information, the team will then be able to design and implement a generic event database that stores rich information about each event and make it easily accessible to data mining tools.

With regards to the engineering analysis of power plant model performance, in addition to the information above, the team will need to improve the quantitative metrics that ISO-NE has developed, analyze the previous APPMV results, and then develop a scoring system based on the model performance to identify inaccurate models. If time permits, the team will explore techniques to narrow down the root cause of bad model performance to individual components.
Preliminary Requirements

To complete the electrical engineering portion of this project, we require access to examples of the reports that the APPMV tool generates and sends to ISO-NE engineers. These will be used to develop the system as well as to test the performance of the system. We will also need a specification of what metrics the APPMV already has available in order to develop the scoring engine. Additionally, in order to develop a scoring system based on the model performance to identify inaccurate models, we will need access to reports from a wide array of times. For example, if we had access to fault reports during a few of the major storms and from surrounding days, we could compare and contrast the reported metrics to create an accurate scoring system.

To complete the CSE portion of this project, access to the aforementioned reports is also necessary. Creating the database requires a large sample size of reports to ensure we can categorize the reports effectively. Additionally, coordination between CSE and ECE team members will be crucial in developing the scoring engine. ECE members will be able to design a system on which to score models and provide these requirements to the CSE members to implement. ECE members will also be able to assist in UI design as they will be able to assist in determining important information to be displayed.

Basic Limitations

A major limitation that may hinder progress on the project is access to the APPMV reports. These are kept on a single machine at ISO New England in Holyoke, Massachusetts. In order to use these reports, we would either need to conduct our work in Holyoke or have an engineer send a few samples to us. Additionally, the APPMV tool is an ISO-developed tool. Before learning about the project, we were required to sign a confidentiality agreement.

Other Data

We will be working with two engineers at ISO New England: Qiang Zhang and Xiaochuan Luo. We will be using the data provided to the engineers at ISO New England via the APPMV system. During the first semester we are working on the planning aspect. The second semester will be focused on implementing the system.

Additional Information

For task 1, the database and web-based GUI design and implementation, strong programming skills are required. For task 2, engineering analysis of power plant model performance, good engineering analytical skills is required. In addition, a good knowledge of systems analysis and circuit design is needed.
Approach and Design

Theory

A system, whether it is a model or a physical entity, responds over time to a given input signal. This time-domain response is typically compared using the correlation coefficient of two signals. The correlation coefficient provides a measurement of how similar the shape of two signals are, where a value of unity indicates that two signals are the same (with perhaps an amplification or difference in magnitude), a value of zero indicates that the two signals are entirely uncorrelated, and a value of negative unity means that the signals are exactly the same, but negatives of one another.

\[ \rho = \frac{\text{cov}(x,y)}{\sigma_x \sigma_y} \]

The correlation coefficient is often used to determine signal similarity, but there are other characteristics of a signal that can be analyzed to provide a more holistic approach to comparing signals.

Power systems, like any other system, have an inherent response to a given input that is dependent on both the system's state and the input's frequency and magnitude. In general, this system response can be described as a transfer function, \( H(j\omega) \), where \( j \) is the imaginary unit and \( \omega \) is the frequency of the input and response in radians per second. Namely, the full relationship is as below.

\[ H(j\omega) = \frac{Y(j\omega)}{X(j\omega)} \]

In this equation, \( Y(j\omega) \) corresponds to the Fourier transform of the output response and \( X(j\omega) \) corresponds to the Fourier transform of the input. Therefore, the transfer function \( H(j\omega) \) provides a signal that represents the system's behavior under a given input.

All signals, to include transfer functions, have two defining characteristics. Firstly, it must be noted that as these signals are represented by their Fourier transform, they are functions of frequency, not functions of time. Additionally, it must be noted that the Fourier transform inherently places the output of a given signal in the complex domain. This provides the two characteristics of a signal: its magnitude and its angle. The magnitude of the signal at a particular frequency is simply the result of finding the hypotenuse of a triangle, where one leg is the real part of the signal and the other leg is the imaginary part of the signal. This can be shown below.

\[ |H(j\omega)| = \sqrt{\Re(H(j\omega))^2 + \Im(H(j\omega))^2} \]

The angle of a signal takes a similar trigonometric approach, except we take the inverse tangent of the two legs to find the angle between them.
These two characteristics together are known as the frequency response of a system, and are often shown together in the following configuration, known as a Bode Plot. This representation makes it easy to visualize the characteristics of a given system over a range of frequencies.

While a transfer function is typically used to relate an input to an output, in theory it can be used to find the relationship between any two signals. Should both of the signals be identical, then the transfer function will be unity. This makes intuitive sense, as if both $X(j\omega)$ and $Y(j\omega)$ are identical, then it would follow that the division of the two would result in unity. Additionally, a transfer function $H(j\omega) = 1$ has a magnitude of one and an angle of zero. This also makes sense as if the two signals are identical, then the relationship between them should change neither their magnitude nor their angle. Because the magnitude of a transfer function is multiplied by an input signal, it must be unity to cause no change. In the complex domain, angles are added and subtracted during multiplication, so the angle of zero corresponds to no change between the input and the output. Using these properties, we can determine the similarity between any two signals by finding how close the transfer function between them is to unity.

In order to do so, we must define a distance function, or metric. A metric is a mathematical function of the form

$$d : X \times X \rightarrow [0, \infty)$$

that satisfies the following four conditions for any $x, y, z \in X$.

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \iff x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, y) \leq d(x, z) + d(z, y)$

In other words, as the above function represents a distance between two values, it must always be nonnegative, zero for any identical values, symmetric in that the distance from $x$ to $y$ must be the same as the distance from $y$ to $x$, and it must satisfy the triangle inequality. In our project, we used two semimetrics, proposed by Dr. Rezaei and Dr. Venkatasubramanian in their paper "Quantitative Indicators for Quality of Fit Assessment in Power System Model Validation Problems." The paper was suggested to us by Frankie Zhang, one of the engineers with whom we are working at ISO-NE. A semimetric is a function that satisfies the first three conditions of a metric, but not necessarily the triangle inequality. The two semimetrics used are used to define the distance between the magnitude of a transfer function and unity as well as to define the distance between the angle of a transfer function and zero. They are defined as below.
The general shape of the two equations is shown below.

\[ d_{M,\alpha}(j\omega) = \tanh \left( \frac{20 \log(|H(j\omega)|)}{\alpha} \right) \]

\[ d_{A,\beta}(j\omega) = \tanh \left( \frac{|\Phi(j\omega)|}{2\pi \beta} \right) \]

The general shape of the two equations is shown below.

As can be seen, when the magnitude is unity or the angle is zero, the distance function defines
the distance of the transfer function to be zero, indicating that the two signals relate exactly. As
we move away from these points, the distance rapidly increases toward unity, the maximum
distance. Additionally, both metrics contain their own sensitivity parameter that determines the
tolerance of these two functions. Essentially, the sensitivity parameter tunes the width of the
region that slopes toward unity. This allows the semimetric to score a transfer function more harshly or more leniently with respect to either the magnitude or the angle, independent of one another.

Given that these semimetrics will find a value for a transfer function at one particular frequency, they alone are not sufficient to determine the quality of a model. They must first be applied to the transfer function over a range of frequency values. In order to do so, the average of each distance semimetric is taken over the range of frequency values, using an integral average. This provides us with an overall distance of magnitude and angle over the given frequency range. Both equations are given below.

\[
M_\alpha = \frac{1}{f_H - f_L} \int_{f_L}^{f_H} 1 - d_{M,\alpha}(j\omega) \, d\omega
\]

\[
A_\beta = \frac{1}{f_H - f_L} \int_{f_L}^{f_H} 1 - d_{A,\beta}(j\omega) \, d\omega
\]

After the above analysis, we are left with three individual scores: the correlation coefficient \( \rho \), the magnitude distance, and the angle distance. In order to summarize these scores, we simply take the average of them in order to determine the final score relating the model's output to the measured signal.

\[
S = \frac{|\rho| + M_\alpha + A_\beta}{3}
\]

Note that we take the absolute value of the correlation coefficient, as we are unconcerned with whether or not it is out of phase since the angle metric covers angle similarity. In this instance, a value of unity means that the two signals are identical, and values decreasing toward zero indicate that the model is not accurate. Also worth noting is the effect of sensitivity parameters alpha and beta on the signal similarity score. By altering these sensitivity parameters, the scoring system can be modified to be more or less restrictive on variations in magnitude or phase angle. The benefit of this is that as ISO-NE continues to improve their models’ accuracy, they will be able to tighten restrictions in the scoring system to increase the distance between scores of many highly performant models. This has the effect of translating a cluster of very high scores to a wider spread. This will allow engineers at ISO-NE to provide better relative comparisons to increasingly accurate models.

Another way we are testing these models is by using the normalized root mean square error. This is used as a filter before using the signal similarity metric described above. To calculate the NRMSE, we must first calculate the RMSE (root mean square error). This is done by taking each value of the model and subtracting the actual from the predicted. This value is then squared and added to the sum of the other values. This is then divided by the total number of values and then raised to the \( \frac{1}{2} \) power. The formula is shown here.
To calculate the normalized root mean square error, we divided the root mean square error (RMSE) by the range of the actual (the highest value of the actual - the lowest value of the actual). The formula is shown below.

$$NRMSE = \frac{RMSE}{MAX_{Actual} - MIN_{Actual}}$$

To determine the fit percentage of the models, we take 1 - NRMSE. This gives us an approximate idea of how well the two models match. This is necessary to maximize the accuracy of the estimated transfer function used in the signal similarity algorithm.

**Model**

The models provided to us by ISO-NE are not directly viewable by our team, as it requires proprietary software. However, we had access to the output responses of the models as well as measurements from the phasor measurement units. It was with these outputs that the system was tested and refined in order to determine the required parameters as well as to decide whether additional metrics were necessary to properly discern which models were accurate and which were not.

**Constraints**

Developing the design for the project placed us under a few constraints. One of the primary constraints is the limited data set that we have available to work with. Given the sensitive nature of real data from the New England power grid, only a few small samples were provided to maintain the security of ISO-NE operations. Additionally, the toolset that ISO-NE uses to develop and simulate system models is not available to our team. We are restricted to using previously generated output from their models, as we are unable to develop and generate our own.

Another large constraint for this project was the COVID-19 pandemic. Originally, we met weekly in our designated lab area to discuss any questions that we had among the ECE team. However, with the stay at home order in place, we had to resort to other methods of
communication. We decided on weekly video chat meetings to keep our sponsors up to date with our progress.

Standards

As of this time, we are not aware of any standards that apply to this project. However, we will keep up to date with the sponsor’s guidelines and any university standards.

Final Design

The final design involved two steps. The first step was using the normalized root mean square error as a filter for each model. If the model did not exceed a certain fit threshold (1-NRMSE), set by ISO New England, then the signal similarity metric would not be used with the model. After ensuring the model met the fit threshold criteria, the second step was to use the signal similarity metric to determine the final score. This two step process allowed us to be sure the score produced by the signal similarity metric was as accurate as possible.

Various alternative designs were previously considered. The most rudimentary was a simple correlation coefficient over the entire signal. This, however, did not provide the granularity that we were looking for. The correlation coefficient gives one overall result for the entire signal, but does not necessarily capture the similarities and differences in various frequencies. The same was true for using the normalized root mean square error (NRMSE). While it provided a good measurement of how well a transient signal matched over time, it did not process the characteristics of the frequency response.

Experimental or Simulation Testing Results

In order to ensure that our final design was working properly, a series of tests with sinusoidal signals were performed on both the NRMSE and the signal similarity algorithm. This allowed us to confirm that the program logic was correctly implemented.

Accuracy of Signal Similarity Score through Sinusoidal Testing

The purpose of testing the accuracy of the score produced by the signal similarity function is to catch possible logic errors within the code. Sinusoidal signals have a well-defined structure, as they are a signal of one pure frequency. In order to test the full functionality of the magnitude semi metric and angle semimetrics, test signals were created with varying amplitudes and varying phase angles. Signal amplitudes were varied from 0x to 10x the base signal and the phase offset was varied from -180 deg to +180 deg with respect to the base signal. To ensure that the semimetrics were functioning properly, the scores were plotted with respect to the changes in amplitude and phase angle. This confirmed that the scores were best
when the magnitude was equivalent to the base signal and when the phase angle was equivalent to the base signal.

**Setting Up Signals**

To set up the sinusoidal tests, the following code was first implemented.

\[
t = 0:0.001:10; \\
t = \text{transpose}(t);
\]

This generates a time domain variable with a 1 kHz sample rate over the course of 10 seconds. Transposing this signal was necessary in order to meet the dimensional requirements of some MATLAB functions.

**Magnitude Semimetric: Change in Amplitude**

In order to test the operation of the magnitude semimetric, the amplitude of the base signal was altered. A base sinusoid was generated to serve as the "actual" data that would have been collected from a PMU. The variations represent "model" data that is generated by the TSAT modelling program used at ISO-NE. The code to implement the signals is listed in the appendix. The frequency of all of these signals were chosen to be at 2 Hz because this was found to be similar to the frequency of the system responses for the BPS.

To test the signals, the transfer function estimates were generated with 2 poles, as the signals should represent a simple oscillating, damped system response. The code written to show how the base/model signals were tested against all of the test signals is listed in the appendix. The magnitude sensitivity parameter, alpha, was set to be 10 as was outlined by the paper on which the algorithm was based. The semimetric was then integrated over a frequency range of 0 Hz to 2 Hz, as this is the operating frequency of the signal.

**Results:**

The following shows the results of the test. As mentioned before, in this test we are looking for a score of 1 or very close to one when the base/model signal and one of the test signals with the same amplitude are compared against each other. The base signal in this scenario was given an amplitude of 1 so the test signal with that same amplitude should produce the score of 1 when compared against the base signal. As shown in the figures below, as the amplitude gets closer to 1 the score increases, and as the amplitude increases past 1, the score goes back down. According to the paper, this shows that this part of the semi-metric is functioning properly since according to the paper, as the distance between two models decrease the score will go towards 1 which was the case in this test scenario.
The next test that was performed was comparing the scores derived from the tests and comparing it to the % Normalized Root Mean Square Error of the test signals against the base/model signal. Going into this test, we kept in mind that the more similar the base/model signal is compared to the test signal, the greater the score, and the greater the % NRMSE. As shown in the figure below, this was proven.

<table>
<thead>
<tr>
<th>Amplitude</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.6738</td>
</tr>
<tr>
<td>0.2</td>
<td>0.4524</td>
</tr>
<tr>
<td>0.3</td>
<td>0.4915</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5256</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5678</td>
</tr>
<tr>
<td>0.6</td>
<td>0.8594</td>
</tr>
<tr>
<td>0.7</td>
<td>0.8986</td>
</tr>
<tr>
<td>0.8</td>
<td>0.6824</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9688</td>
</tr>
<tr>
<td>1</td>
<td>0.9994</td>
</tr>
<tr>
<td>2</td>
<td>0.8196</td>
</tr>
<tr>
<td>3</td>
<td>0.4989</td>
</tr>
<tr>
<td>4</td>
<td>0.4681</td>
</tr>
<tr>
<td>5</td>
<td>0.7036</td>
</tr>
<tr>
<td>6</td>
<td>0.6922</td>
</tr>
<tr>
<td>7</td>
<td>0.574</td>
</tr>
<tr>
<td>8</td>
<td>0.6809</td>
</tr>
<tr>
<td>9</td>
<td>0.6806</td>
</tr>
</tbody>
</table>

Angle Semimetric: Change in Phase

In order to test the operation of the angle semimetric, changes in the signal phase were tested. In order to do so, the same base "measured" signal from the magnitude test was used, with variations in phase angle used to represent the "model" signals. The code implementing the signals is listed in the appendix. The frequency of all of these signals were chosen to be kept at 2 Hz for the same reason as in the magnitude test.
Results:

The following shows the results of the test. As mentioned before, in this test we are looking for a score of 1 or very close to 1 when the base/model signal and one of the test signals with the same amplitude are compared against each other. The base signal in this scenario was given a phase of -180 so the test signal with that same phase should produce the score of 1 when compared against the base signal. As shown in the figures below, when the base/model signal was tested against the test signal with the same phase a score very close to 1 was produced. According to the paper, as the angle distance decreases between two signals, the score should produce a value of 1 as well. This appeared to be the case at phase -180 but it seems that when the phase was opposite of -180, so at +180, the score went back up to 0.7463 when we expected a score of 0 or very close to that.

| Score - 2 poles | 0.9995 | 0.9678 | 0.9112 | 0.8601 | 0.8926 | 0.8922 | 0.9048 | 0.9219 | 0.9191 | 0.979 | 0.9359 | 0.9395 | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
|-----------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|--------|-------|-------|-------|
| -180            | 0.9678 | 0.9112 | 0.8601 | 0.8926 | 0.8922 | 0.9048 | 0.9219 | 0.9191 | 0.979 | 0.9359 | 0.9395 | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -170            | 0.9048 | 0.9219 | 0.9191 | 0.979 | 0.9359 | 0.9395 | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -160            | 0.9219 | 0.9191 | 0.979 | 0.9359 | 0.9395 | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -150            | 0.9191 | 0.979 | 0.9359 | 0.9395 | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -140            | 0.979 | 0.9359 | 0.9395 | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -130            | 0.9359 | 0.9395 | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -120            | 0.9395 | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -110            | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -100            | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -90             | 0.979 | 0.9359 | 0.9395 | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -80             | 0.9359 | 0.9395 | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -70             | 0.9395 | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -60             | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -50             | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -40             | 0.9359 | 0.9395 | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -30             | 0.9395 | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -20             | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| -10             | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |
| 0               | 0.9359 | 0.9395 | 0.9524 | 0.9691 | 0.8384 | 0.628 | 0.6513 | 0.9549 | 0.7444 | 0.7463 |

When the score went back up close to a value of 1, I decided to look at the bode plot of the transfer function when the signal with the phase of -180 was compared against the base/model signal. As you can see below, the phase went to 360 degrees, which makes sense since the difference between -180 and +180 is 360, but we want the sinusoidal to in a way override that; instead we want the bode plot to show that its going up to 180 and to recognize that +180 is not a mirror reflection of the signal but still completely different from it though.
To further study why the score wasn’t behaving the way it should, the % NRMSE of the scores was also observed. As you can see below, instead of the plot being linear, it curves up, which supports the finding that the transfer function at that test signal is going back to a degree of 360.

Transfer Function Estimation
The last task that had to be confirmed was the estimation of the transfer function for the model data and PMU data provided to us by ISO-NE. The transfer function between these two signals must be estimated for use in the signal similarity algorithm. However, because the transfer function is not of the system itself, but rather the relation between the model and the measured data, an accurate system analysis cannot be done beforehand, and must be generated dynamically. In order to do so, the tfest() function provided by MATLAB was used. The estimator function provides an NRMSE fit metric to represent the accuracy of the transfer function generated. Since the purpose of this system is to produce a quantitative score between two signals, the estimation of the transfer function must be as accurate as possible. Furthermore, the number of poles used in the transfer function estimation does not need to be consistent, so long as accurate transfer functions are generated for analysis. This is because the transfer functions between different models and different measurement data will differ over time.
Comparison of transfer function estimates with 2, 3, and 4 poles.

Empirical results show that 4 poles was often most accurate at representing the relation between the two signals. However, in certain cases, 3 poles or 5 poles was more accurate. The system first attempts to estimate a transfer function with 4 poles. If the accuracy does not meet a threshold value, then it will attempt to estimate with other poles until a desired transfer function accuracy is reached. This allows the algorithm to adapt to different models for different plants within the BPS.

Further issues resulted from the data itself, as the system response is initially non-linear. Testing shows that estimating the transfer function on only the linear region improves the accuracy significantly, and provides for more consistent scoring of models.
The above results demonstrate that paring the data to contain only the linear region substantially improves the fit of the estimated transfer function.

The last test performed involved the frequency range over which the transfer function semimetrics are integrated and averaged. Although originally we were testing with a static range of 0 to 1 Hz, as outlined in the paper, we then looked for means of generalizing the frequency range to adapt to various system models. This involved taking the FFT of the input data and selecting the first peak as the operating frequency. We then performed the integration over this frequency range.

The results of these tests were then plotted relative to the NRMSE fit of the model. The purpose of this was to confirm that the signal similarity score was at least consistent with the NRMSE fit.
As can be seen from the above figure, original data points with the static frequency range can have a low signal similarity score but a high NRMSE fit. For example, one score for the original Q data is approximately 0.3, despite an NRMSE fit of 70%. By adding this dynamic frequency range, the corresponding score moves closer to match that of the NRMSE fit. The corresponding point from the previous example now has the Q score of approximately 0.75, more closely matched to the NRMSE fit of 70%.

**Project Management**

**RACI Chart**

A RACI Chart helps organize a group project by determining who is responsible for each task. Each person working on the project is included and then assigned a letter: R, A, C, or I, to determine who works on each task. ‘R’ stands for Responsible, ‘A’ for Accountable, ‘C’ for Consulted, ‘I’ for Informed. For most of the tasks, the advisor (Dr. Luh), and the sponsors were consulted or informed. To determine who is responsible or accountable for each task, each individual team member wrote their skills and past experiences. Then we assigned each person a task based on their past experiences. If a person had experiences similar to a task, they were assigned as accountable. The other team members were assigned as responsible. The RACI chart is shown in Figure 1.

**Part List**

There is no part list for this project, as both the ECE team and CSE team implemented the project in software.

**Budget List**

The project is backed by ISO New England. Additionally, since this project is using publicly available software, no purchases were necessary.

**Gantt Chart**

A Gantt Chart helps a project stay up to date and make sure that tasks are completed on time. Using the same breakdown of tasks as the RACI Chart, the Gantt chart for this project was created. The project was broken into four phases: Introduction, Design, Implementation, Verification. The introduction and design phases take place during the first semester. The implementation and verification phases take place during the second semester. The Gantt Chart is included in the Figures Section (both an overview and a broken down view) under Figure 2 through Figure 6.
Summary

The Generator Model Verification Result Analysis System involves both the theoretical side of evaluating various models and the practical side of implementing the analysis in a software system with a web application. Our responsibility as a member of the ECE team assigned to ISO New England's project is to develop the criteria for how each model will be ranked as well as to develop means for improving the models. The scoring engine will rank models by determining the correlation between the models response to an event with respect to the physically measured response to the same event. This will allow models to be compared to one another quantitatively in terms of their accuracy. In addition, by cross-correlating the model's response and the physical response in the time-domain, we can determine where models were adequate in predicting the physical response and thus determine which part of a model is accurate. This analysis will allow us to better determine how to improve the models with respect to the components of a system response. Furthermore, the addition of sensitivity parameters to the semimetrics allows model evaluation to become increasingly strict over time, with the intent of iteratively refining models. This allows a set of highly performant models to be further separated from one another in scoring.

As shown in the sinusoidal testing of the signal similarity score, further research and tests should be performed, particularly on the changes of the phase. Further work must be done to determine a means of generating a test signal pair that creates a transfer function with a phase offset of 180 degrees near 0 Hz. Although a phase offset of 180 degrees in the test signal was expected to produce such a response, the transfer function estimate generated by MATLAB did not reflect this. The root cause was not determined.
Figures and Diagrams

Figure 1: RACI Chart

<table>
<thead>
<tr>
<th>Project task or deliverable</th>
<th>Cynthia Bissereth</th>
<th>Jonathan Davis</th>
<th>Ethan McRae</th>
<th>Qi Zhang / Xiaochuan Luo</th>
<th>Dr. Peter Luh</th>
<th>CSE Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Statement</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>C</td>
<td>C</td>
<td>I</td>
</tr>
<tr>
<td>Project Proposal</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>C</td>
<td>C</td>
<td>I</td>
</tr>
<tr>
<td>Determine Current Qualification Metrics</td>
<td>R</td>
<td>A</td>
<td>R</td>
<td>C</td>
<td>C</td>
<td>I</td>
</tr>
<tr>
<td>Determine New Qualification Metrics</td>
<td>R</td>
<td>A</td>
<td>R</td>
<td>C</td>
<td>C</td>
<td>I</td>
</tr>
<tr>
<td>Determine Grading Metrics</td>
<td>A</td>
<td>R</td>
<td>R</td>
<td>C</td>
<td>C</td>
<td>I</td>
</tr>
<tr>
<td>Design Scoring Engine Based Off Grading Metrics</td>
<td>A</td>
<td>R</td>
<td>R</td>
<td>C</td>
<td>C</td>
<td>I</td>
</tr>
<tr>
<td>Design Basic Database Schema</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>C</td>
<td>C</td>
<td>R</td>
</tr>
<tr>
<td>Complete Basic Software Design</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>C</td>
<td>C</td>
<td>R</td>
</tr>
<tr>
<td>Complete Final Design Report</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>C</td>
<td>C</td>
<td>I</td>
</tr>
<tr>
<td>Implement User Interface</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>C</td>
<td>C</td>
<td>R</td>
</tr>
<tr>
<td>Implement Qualification Metrics</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>C</td>
<td>C</td>
<td>R</td>
</tr>
<tr>
<td>Implement Scoring Engine</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>C</td>
<td>C</td>
<td>R</td>
</tr>
<tr>
<td>Implement Sponsor Suggestions</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>C</td>
<td>C</td>
<td>R</td>
</tr>
<tr>
<td>Design Project Display Board for Demonstration Day</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>C</td>
<td>C</td>
<td>R</td>
</tr>
</tbody>
</table>

Figure 2: Gantt Chart Phase 1

![Gantt Chart Phase 1](image-url)
Figure 3: Gantt Chart Phase 2

<table>
<thead>
<tr>
<th>Task Description</th>
<th>Start</th>
<th>End</th>
<th>% Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provided By ISO New England Complete Project Proposal</td>
<td>Sat 16/10/19</td>
<td>Tue 20/10/19</td>
<td>100%</td>
</tr>
<tr>
<td>Determine Current Qualification Metrics</td>
<td>Wed 23/10/19</td>
<td>Tue 29/10/19</td>
<td>100%</td>
</tr>
<tr>
<td>Determine New Qualification Metrics</td>
<td>Tue 1/11/19</td>
<td>Mon 11/11/19</td>
<td>100%</td>
</tr>
<tr>
<td>Determine Gradining Metrics</td>
<td>Tue 1/11/19</td>
<td>Mon 11/11/19</td>
<td>100%</td>
</tr>
<tr>
<td>Design Scoring Engine Based Off Gradining Metrics</td>
<td>Tue 1/11/19</td>
<td>Mon 11/11/19</td>
<td>100%</td>
</tr>
<tr>
<td>Separate Metrics By ISO New England Specifications</td>
<td>Sun 11/11/19</td>
<td>Fri 11/22/19</td>
<td>100%</td>
</tr>
<tr>
<td>Design Basic User Interface</td>
<td>Tue 13/11/19</td>
<td>Mon 11/11/19</td>
<td>100%</td>
</tr>
<tr>
<td>Create Database Schema</td>
<td>Sun 10/11/19</td>
<td>Sat 10/11/19</td>
<td>100%</td>
</tr>
<tr>
<td>Complete Basic Software Design Meet With ISO New England to Discuss Project Progress</td>
<td>Sun 12/11/19</td>
<td>Fri 12/19/19</td>
<td>100%</td>
</tr>
<tr>
<td>Complete Final Design Report</td>
<td>Tue 10/12/19</td>
<td>Fri 12/06/19</td>
<td>100%</td>
</tr>
</tbody>
</table>

Figure 4: Gantt Chart Phase 3

<table>
<thead>
<tr>
<th>Task Description</th>
<th>Start</th>
<th>End</th>
<th>% Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implement ISO New England’s Suggestions Implement the Qualification Metrics’s</td>
<td>Tue 12/1/20</td>
<td>Fri 2/28/20</td>
<td>100%</td>
</tr>
<tr>
<td>Implement Scoring Engine</td>
<td>Tue 12/1/20</td>
<td>Fri 2/28/20</td>
<td>100%</td>
</tr>
<tr>
<td>Meet With ISO New England to Discuss Project Progress</td>
<td>Fri 2/28/20</td>
<td>Fri 2/28/20</td>
<td>100%</td>
</tr>
</tbody>
</table>

Figure 5: Gantt Chart Phase 4

<table>
<thead>
<tr>
<th>Task Description</th>
<th>Start</th>
<th>End</th>
<th>% Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verification</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Meeting With ISO New England Implement ISO New England Suggestions</td>
<td>Fri 20/10/20</td>
<td>Fri 1/08/20</td>
<td>100%</td>
</tr>
<tr>
<td>Design Documentation</td>
<td>Wed 2/10/20</td>
<td>Wed 4/25/20</td>
<td>100%</td>
</tr>
<tr>
<td>Final Design Review</td>
<td>Wed 3/25/20</td>
<td>Wed 4/25/20</td>
<td>100%</td>
</tr>
<tr>
<td>Present Project</td>
<td>Wed 4/07/20</td>
<td>Wed 4/07/20</td>
<td>100%</td>
</tr>
<tr>
<td>Final Project Report</td>
<td>Wed 4/07/20</td>
<td>Wed 4/07/20</td>
<td>100%</td>
</tr>
</tbody>
</table>
References


GitHub access to the private repository can be provided upon request

Sinusoidal Testing for Signal Similarity Code

Changes in amplitude:

\[ \text{basesignal} = \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \] \text{basesignal with amplitude 1} \]

\[ \begin{align*}
    aa &= 0.1 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 0.1} \\
    bb &= 0.2 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 0.2} \\
    cc &= 0.3 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 0.3} \\
    dd &= 0.4 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 0.4} \\
    ee &= 0.5 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 0.5} \\
    fff &= 0.6 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 0.6} \\
    gg &= 0.7 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 0.7} \\
    hh &= 0.8 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 0.8} \\
    ii &= 0.9 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 0.9} \\
    a &= \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 1} \\
    b &= 2 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 2} \\
    c &= 3 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 3} \\
    d &= 4 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 4} \\
    e &= 5 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 5} \\
    ff &= 6 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 6} \\
    g &= 7 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 7} \\
    h &= 8 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 8} \\
    i &= 9 \cdot \exp(-0.02t) \cdot \sin(2\pi \cdot 2t); \text{signal with amplitude 9} \\
\end{align*} \]

\[ \begin{align*}
    \text{result1} &= \text{signalSimilarity(basesignal, aa, 0.001, 0, 1, 10, 0.5)}; \\
    \text{result2} &= \text{signalSimilarity(basesignal, bb, 0.001, 0, 1, 10, 0.5)}; \\
    \text{result3} &= \text{signalSimilarity(basesignal, cc, 0.001, 0, 1, 10, 0.5)}; \\
    \text{result4} &= \text{signalSimilarity(basesignal, dd, 0.001, 0, 1, 10, 0.5)}; \\
    \text{result5} &= \text{signalSimilarity(basesignal, ee, 0.001, 0, 1, 10, 0.5)}; \\
    \text{result6} &= \text{signalSimilarity(basesignal, fff, 0.001, 0, 1, 10, 0.5)}; \\
    \text{result7} &= \text{signalSimilarity(basesignal, gg, 0.001, 0, 1, 10, 0.5)}; \\
    \text{result8} &= \text{signalSimilarity(basesignal, hh, 0.001, 0, 1, 10, 0.5)}; \\
    \text{result9} &= \text{signalSimilarity(basesignal, ii, 0.001, 0, 1, 10, 0.5)}; \\
    \text{result10} &= \text{signalSimilarity(basesignal, a, 0.001, 0, 1, 10, 0.5)}; \\
    \text{result11} &= \text{signalSimilarity(basesignal, b, 0.001, 0, 1, 10, 0.5)}; \\
    \text{result12} &= \text{signalSimilarity(basesignal, c, 0.001, 0, 1, 10, 0.5)}; \\
\end{align*} \]
result13 = signalSimilarity(basesignal, d, 0.001, 0, 1, 10, 0.5);
result14 = signalSimilarity(basesignal, e, 0.001, 0, 1, 10, 0.5);
result15 = signalSimilarity(basesignal, ff, 0.001, 0, 1, 10, 0.5);
result16 = signalSimilarity(basesignal, g, 0.001, 0, 1, 10, 0.5);
result17 = signalSimilarity(basesignal, h, 0.001, 0, 1, 10, 0.5);
result18 = signalSimilarity(basesignal, i, 0.001, 0, 1, 10, 0.5);

Changes in Phase:

basesignal = exp(-0.02*t).*sin(2*pi*2*t + deg2rad(-180)); %basesignal with phase -180

aa = exp(-.02*t).*sin(2*pi*2*t + deg2rad(-180)); %signal with phase -180
aa1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-170)); %signal with phase -170
bb = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-160)); %signal with phase -160
bb1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-150)); %signal with phase -150
cc = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-140)); %signal with phase -140
cc1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-130)); %signal with phase -130
dd = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-120)); %signal with phase -120
dd1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-110)); %signal with phase -110
ee = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-100)); %signal with phase -100
ee1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-90)); %signal with phase -90
fff = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-80)); %signal with phase -80
fff1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-70)); %signal with phase -70
gg = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-60)); %signal with phase -60
gg1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-50)); %signal with phase -50
hh = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-40)); %signal with phase -40
hh1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-30)); %signal with phase -30
ii = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-20)); %signal with phase -20
ii1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(-10)); %signal with phase -10
a = exp(-.02*t).*sin(2*pi*2*t + deg2rad(0)); %signal with phase 0
a1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(10)); %signal with phase 10
b = exp(-.02*t).*sin(2*pi*11*t + deg2rad(20)); %signal with phase 20
b1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(30)); %signal with phase 30
c = exp(-.02*t).*sin(2*pi*11*t + deg2rad(40)); %signal with phase 40
c1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(50)); %signal with phase 50
d = exp(-.02*t).*sin(2*pi*11*t + deg2rad(60)); %signal with phase 60
d1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(70)); %signal with phase 70
e = exp(-.02*t).*sin(2*pi*11*t + deg2rad(80)); %signal with phase 80
e1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(90)); %signal with phase 90
ff = exp(-.02*t).*sin(2*pi*11*t + deg2rad(100)); %signal with phase 100
ff1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(110)); %signal with phase 110
g = exp(-.02*t).*sin(2*pi*11*t + deg2rad(120)); %signal with phase 120
g1 = exp(-.02*t).*sin(2*pi*11*t + deg2rad(130)); %signal with phase 130
\[ h = \exp(-0.02t) \cdot \sin(2\pi t \cdot 11 + \text{deg2rad}(140)); \ %\text{signal with phase 140} \]
\[ h1 = \exp(-0.02t) \cdot \sin(2\pi t \cdot 11 + \text{deg2rad}(150)); \ %\text{signal with phase 150} \]
\[ i = \exp(-0.02t) \cdot \sin(2\pi t \cdot 11 + \text{deg2rad}(160)); \ %\text{signal with phase 160} \]
\[ i1 = \exp(-0.02t) \cdot \sin(2\pi t \cdot 11 + \text{deg2rad}(170)); \ %\text{signal with phase 170} \]
\[ j = \exp(-0.02t) \cdot \sin(2\pi t \cdot 2 + \text{deg2rad}(180)); \ %\text{signal with phase 180} \]
Senior Design Project Checklist

**Project name:** Generator Model Verification Result Analysis System; Team 2011

**Sponsor:** ISO New England

**Team members (majors/programs):**

Cynthia Bissereth (ECE), Jonathan Davis (ECE, CSE), Ethan McRae (ECE)

**Faculty advisor(s):**

Dr. Peter Luh

**Skills, Constraints, and Standards:** *(Please check (√) all those that apply to your project.)*

<table>
<thead>
<tr>
<th>Skills:</th>
<th>(√)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog circuit design and troubleshooting</td>
<td></td>
</tr>
<tr>
<td>Digital circuit design and troubleshooting</td>
<td></td>
</tr>
<tr>
<td>Software development/programming</td>
<td>√</td>
</tr>
<tr>
<td>Embedded Systems/Microcontrollers</td>
<td></td>
</tr>
<tr>
<td>Web design</td>
<td></td>
</tr>
<tr>
<td>RF/wireless hardware</td>
<td></td>
</tr>
<tr>
<td>Control systems</td>
<td></td>
</tr>
<tr>
<td>Communication systems</td>
<td></td>
</tr>
<tr>
<td>Power systems</td>
<td>√</td>
</tr>
<tr>
<td>Signal processing</td>
<td>√</td>
</tr>
<tr>
<td>Machine shop/mechanical design</td>
<td></td>
</tr>
<tr>
<td>Other (please specify):</td>
<td></td>
</tr>
</tbody>
</table>

**Constraints:**

Economic (budget)
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Health/safety</td>
<td></td>
</tr>
<tr>
<td>Manufacturability</td>
<td></td>
</tr>
<tr>
<td>Environmental (e.g., toxic materials, fossil fuels)</td>
<td></td>
</tr>
<tr>
<td>Social/legal (e.g., privacy)</td>
<td>√</td>
</tr>
</tbody>
</table>

**Standards:**

**List** standards/electric codes that you used (e.g., IEEE 802.11, Bluetooth, RS-232, VHDL, etc.)

If applicable, list the name or # here:

N/A
Glossary

Correlation coefficient — A number between -1 and +1 that describes the correlation between two signals. A coefficient of +1 means the two signals are exactly correlated. A coefficient of -1 means that the two signals are exactly inversely correlated. A coefficient of 0 means there is no correlation between the signals.

Fourier transform — A mathematical transform that decomposes a transient signal into its component frequencies. The Fourier transform of a signal is in the frequency domain, as it is a function of frequency where its output is the magnitude of the given frequency in the original signal.

Frequency domain — The domain of numbers corresponding to frequencies used for frequency analysis. Similarly to how the time domain exhibits how a signal changes over time, the frequency exhibits how a signal changes with respect to frequency.

Metric — A mathematical function that provides the distance between two sets. It must conform to four conditions: non-negativity, identity, symmetry, and the triangle inequality.

Phasor — A complex number capable of representing a sinusoid. It carries information pertaining to the frequency, magnitude, and angle offset of the sinusoid.

Phasor measurement unit — A device capable of estimating the phasor quantity of current or voltage in an electrical system.

Semimetric — A mathematical function that satisfies all of the conditions of a metric except the triangle inequality.

Signal — A signal is a transient function that changes over time, capable of carrying information.

Time domain — The domain of numbers corresponding to moments in time used for transient analysis. A graph of signal strength over time is an example of viewing a signal in the time domain.

Transfer function — A function that relates a system output to a signal input.

BPS - Bulk Power Systems

APPMV - Automatic Power Plant Model Verification