

Iterative Matrix Pencil Method for Power System Modal Analysis

Wei Trinh, Komal S Shetye, Ikponmwosa Idehen, Thomas J Overbye
 Department of Electrical and Computer Engineering
 Texas A& M University, College Station, TX, USA
 {weit1, shetye, iidehen, overbye}@tamu.edu

Abstract

This paper introduces a modal analysis approach termed as the Iterative Matrix Pencil method. It uses the Matrix Pencil Method as the primary tool for mode identification, and adds to it by utilizing the concept of a cost function in order to reduce the number of signals needed to identify the modes for a large system. The method is tested for a variety of large synthetic power grids in this paper, with the cost function being reported to measure accuracy. A sensitivity analysis is also considered, showing how this new method behaves when adjusting the two primary user-based inputs; the number of iterations, and the SVD threshold.

1. Introduction

Modal analysis has been used in power systems to assess small signal stability (the ability of a system to withstand small disturbances), in addition to determining the characteristics of electromechanical oscillations [1, 2]. This involves calculating the eigenvalues which yield the damping and natural frequency, and eigenvectors (i.e. oscillation mode shapes) for a system. Modal analysis methods can be broadly categorized into two types. First, model-based methods involve linearizing the system at a particular operating point. A detailed dynamic representation of the system is needed, including all its components such as generators, exciters, governors, loads, etc. This can be challenging for large systems. The second type are the signal-based methods which make use of measurements from the grid, such as phasor measurement unit (PMU) data. This can be done totally independent of the model. With the proliferation of PMU's, signal-based methods are gaining more traction, in addition to their advantage of working with data compared to models [3, 4, 5].

The goal of signal-based modal analysis techniques

is to approximate a set of signals, $y(t)$, where:

$$y(t) \approx \sum_{i=1}^M R_i e^{z_i t} + n(t), \quad 0 \leq t \leq T \quad (1)$$

where M is the number of modes associated with the signal. Each mode z_i is defined by its' damping b_i and angular frequency ω_i . Each signal is also associated with its' residue, R_i , defined by its' magnitude $|R_i|$ and phase θ_i . Finally, all signals are determined to have some amount of noise, $n(t)$, associated with them.

There are a myriad of signal-based techniques used, each with its own merits. Prony analysis was established as one of the fundamental polynomial analysis techniques [6], and has been used for decades. However, due to constraints such as its lack of resiliency against noise, new methods were developed to handle cases where Prony Analysis could not accurately identify modes [7]. One such matrix-based method is the Matrix Pencil method [8], which estimates the modes of a system using eigenvalues and matrices that are constructed from system measurements [7]. This method will be used in this paper as the main tool for modal analysis.

While the Matrix Pencil method is more robust against noise, the modes are used to identify the mode shapes using a linear-least squares fit [9]. Due to the highly non-linear properties of power systems, techniques such as the Variable Projection Method (VPM) project the linear mode shapes into non-linear modes, thereby solving for a non-linear optimization problem [10, 11]. Other techniques that have been proven to handle large, non-linear systems include Dynamic Mode Decomposition (DMD), which reports the modes of a large system in a computationally quick manner [12, 13, 14].

Iterative approaches to modal analysis also exist, which initially identify a set of modes, and begin to improve on the guess with further iterations. The aforementioned VPM is one of these methods, which solves for the non-linear modes α , and iteratively

improves on its guesses of α by utilizing the gradient $\nabla\alpha$ [10]. Other iterative techniques in power systems include selective modal analysis (SMA), which iteratively identifies modes by adding more information to the underlying data set that determines the modes, until the modes converge [15].

Many of these methods use a variety of metrics to judge the quality of the estimated modes. One common method is to employ the signal-to-noise ratio (SNR), given in (2), which measures the degree to which a signal is affected by noise [7].

$$SNR = 20 \log_{10} \frac{\|y(t)\|}{\|y(t) - \hat{y}(t)\|} \quad (2)$$

This metric is primarily used in linear modal analysis techniques such as Prony Analysis and the Matrix Pencil method. Methods such as VPM utilize a different metric, known as the cost function, defined as an optimization problem in (3). Here, $\hat{y}(\alpha)$ is the reproduced data given an initial data set y , while the cost function is the normalization of the residuals $r(\alpha)$.

$$\min_{\alpha} \frac{1}{2} \|r(\alpha)\|_2^2 = \min_{\alpha} \frac{1}{2} \|y - \hat{y}(\alpha)\|_2^2 \quad (3)$$

The cost function will be used as the metric of choice for considering the accuracy of the modes. Finally, SMA utilizes a different metric from all other methods, known as the participation factor ρ [16], defined as

$$\rho = \left| \frac{w_r^T v_r}{w_z^T v_z} \right| \quad (4)$$

where v and w are the left and right eigenvectors of the matrix A , which contains data in regards to relevant variables r and less relevant variables z [16].

While a majority of the methods previously discussed allow for accurate identification of modes for single signals and smaller systems, there remains a need for more modal analysis techniques that accurately identify modes in a timely fashion for large systems. Concerns with both the stability of the system after events and the ability to control the system as events occur demand that the identification of the modes happens as fast as possible [14]. As a result of these concerns, the Iterative Matrix Pencil method was created, which allows for the handling of significantly large systems by only considering a small subset of signals to most accurately identify the modes.

The remainder of the paper is structured as follows. Section 2 breaks down the theory behind the Iterative Matrix Pencil (IMP) method, discussing in detail how

the modes and mode shapes are calculated, along with how they are used to generate cost functions and consider further iterations of the IMP method. Section 3 applies the method to a variety of large bus cases. Section 4 discusses the sensitivity of the method to a variety of inputs, giving insight on how to best utilize the Iterative Matrix Pencil method for numerous cases. Finally, Section 5 concludes the paper.

2. Iterative Matrix Pencil Method

The IMP method is an iterative approach to approximating (1) for a large number of signals, using the Matrix Pencil method for mode estimation and cost functions as criterion for signal inclusion. IMP begins with a single signal, and calculates the modes following which the mode shapes and cost functions for all signals in the system may be calculated.

The signals are sorted by their cost function, and the signal of interest for an iteration is the one with the highest cost function. The highest cost function represents the worst fit given the existing modes, which indicates that the signal with the highest cost function contains information pertinent to the oscillations that were not accurately represented in the current set of modes. As a result, the IMP method will select the signal with the highest cost function, and add it into the set of signals being considered for the Matrix Pencil method. The procedure is repeated m times, where m is a user-defined amount of iterations.

Before any analysis, however, we must first detrend the signals to ensure accurate mode estimates. Without detrending, data sets may look as if there are fewer to no oscillations that occur after an event, due to the scale of the oscillation relative to the scale of the entire data set [10]. As a result, performing modal analysis on a data set that is not detrended may result in identifying modes that are not due to the oscillation, but due to other components of the signal. While constant detrending is useful in situations where oscillations have a low magnitude, the oscillations in the large systems being considered do not all have a low magnitude. Therefore, each case will have linear detrending applied to emphasize the oscillations that occur.

2.1. Matrix Pencil Method

We approximate a signal $y(t)$ using the following equation

$$\hat{y} = d(t) + \sum_{i=1}^M |R_i| e^{b_i t} \cos(\omega_i t + \theta_i) \quad (5)$$

The first term, $d(t)$, is the detrend function, which varies case by case. The remaining variables are similar in definition as stated in (1), where the modes z_i are defined by their damping b_i and angular frequency ω_i , and the residues, also known as mode shapes [17], R_i , are defined by their amplitude $|R_i|$ and phase θ_i .

The Matrix Pencil method begins with calculating the modes z_i through the use of a Hankel Matrix. The Hankel matrix, which we denote using the variable $[Y]$, is constructed using the data for $y(t)$ as seen in (6). Here, L is known as the pencil parameter, and has been shown to remove some of the effects of noise in the data being considered [18]. $[Y]$ is an $(N - L) \times L$ matrix, and is used to determine both the eigenvalues and eigenvectors of the signal.

$$[Y] = \begin{bmatrix} y(0) & y(1) & \dots & y(L) \\ y(1) & y(2) & \dots & y(L+1) \\ \vdots & \vdots & \ddots & \vdots \\ y(N-L-1) & y(N-L) & \dots & y(N-1) \end{bmatrix} \quad (6)$$

The pencil parameter L is assigned to the value $N/2$, where N is the number of sampled data points. With this substitution, the Matrix Pencil method has been shown to reach the Cramer-Rao bound, which is a measure of the best result that may be achieved when considering noisy data [7].

In general, the Matrix Pencil method utilizes the Hankel matrix in (6) by creating another matrix whose eigenvalues correspond to the poles z_i . The resulting matrix used by the Matrix Pencil method is known as the pencil, and is seen in (7).

$$[Y_2] - \lambda[Y_1] \quad (7)$$

It should be noted that for noiseless data, $n(t) = 0$ for (1). As a result of this, $[Y_1]$ and $[Y_2]$ are defined as

$$[Y_1] = \begin{bmatrix} y(0) & y(1) & \dots & y(L-1) \\ y(1) & y(2) & \dots & y(L) \\ \vdots & \vdots & \ddots & \vdots \\ y(N-L-1) & y(N-L) & \dots & y(N-2) \end{bmatrix} \quad (8)$$

$$[Y_2] = \begin{bmatrix} y(1) & y(2) & \dots & y(L) \\ y(2) & y(3) & \dots & y(L+1) \\ \vdots & \vdots & \ddots & \vdots \\ y(N-L) & y(N-L+1) & \dots & y(N-1) \end{bmatrix} \quad (9)$$

However, in the presence of noise, we reformulate $[Y_1]$ and $[Y_2]$ using singular value decomposition (SVD) [18]. Singular value decomposition is a

technique that is primarily used for reducing a data set to a lower dimension. This lower dimension matrix is still representative of the original, but is easier to computationally handle. The result of SVD is three matrices, $[U]$, $[S]$, and $[V]$. While $[U]$ and $[V]$ are unitary matrices that contain eigenvectors of $[Y][Y]^T$ and $[Y]^T[Y]$, respectively, $[S]$ is an $M \times N$ diagonal matrix that contains the square root of the eigenvalues of $[U]$ and $[V]$. In general, we also implement a user defined threshold to remove singular values that fall below a desired threshold. This is to remove any variations in the data that are more likely linked to minor noise variations rather than actual oscillations of the system.

Here, we define the threshold as a variable q , and consider the ratio of all singular values compared to the largest singular value. All ratios that fall below the threshold q are removed from consideration. The right singular matrix $[V]$ is used in conjunction with q to define two new matrices, $[V_1]$ and $[V_2]$, in (10).

$$[V_1] = [v_1 \ v_2 \ v_3 \ \dots \ v_{q-1}]$$

$$[V_2] = [v_2 \ v_3 \ v_4 \ \dots \ v_q] \quad (10)$$

These definitions are then used to define a new $[Y_1]$ and $[Y_2]$ in (11). The eigenvalues of the pair of matrices $\{[Y_2], [Y_1]\}$ are give us the poles of the system, which in turn allow us to calculate the damping and angular frequency of each mode.

$$[Y_1] = [V_1]^T[V_1] \quad [Y_2] = [V_2]^T[V_1] \quad (11)$$

The eigenvalues of the poles z_i are calculated using

$$z_i = e^{\lambda_i \Delta t} \implies \lambda_i = \frac{\ln(z_i)}{\Delta t} \quad (12)$$

where Δt is the spacing between the points in the data set. The poles are also used to calculate the mode shapes R_i , through (13), which can be written as the simple matrix equation $ZR = Y$. Due to the fact that R is a $M \times N$ matrix, (13) must be solved using a least squares method [7].

$$\begin{bmatrix} z_1^0 & z_2^0 & \dots & z_M^0 \\ z_1^1 & z_2^1 & \dots & z_M^1 \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{N-1} & z_2^{N-1} & \dots & z_M^{N-1} \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_M \end{bmatrix} = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} \quad (13)$$

2.2. Considering Multiple Signals

The Matrix Pencil method described in 2.1 is primarily written as a process for a single signal.

However, this method may be extended to utilize multiple signals and calculate a set of modes that best represent the set of signals. If we consider any number of signals $y_k(t)$, where $k \in [1, n]$, there is a set of modes $(b_{k,i}, \omega_{k,i})$ which best fit all of the signals and their respective mode shapes $(|R_{k,i}|, \theta_{k,i})$.

The Matrix Pencil method for k signals $y_k(t)$ begins with a construction of Hankel matrices for each signal.

$$[Y'_k] = \begin{bmatrix} y_k(0) & y_k(1) & \dots & y_k(L) \\ y_k(1) & y_k(2) & \dots & y_k(L+1) \\ \vdots & \vdots & \ddots & \vdots \\ y_k(N-L-1) & y_k(N-L) & \dots & y_k(N-1) \end{bmatrix} \quad (14)$$

The matrices are then vertically concatenated into one matrix [10], which we denote as $[Y']$.

$$[Y'] = \begin{bmatrix} [Y'_1] \\ [Y'_2] \\ \vdots \\ [Y'_n] \end{bmatrix} \quad (15)$$

From here, the procedure follows the same methodology as the original Matrix Pencil method, where a SVD is applied to our large Hankel matrix, and the resulting matrices are all used to identify the poles z_i and their respective eigenvalues λ_i . These poles are then used to individually calculate the mode shapes of each signal approximation $\hat{y}_k(t)$, through usage of an equation very similar to (13).

$$\begin{bmatrix} z_{1,k}^0 & z_{2,k}^0 & \dots & z_{M,k}^0 \\ z_{1,k}^1 & z_{2,k}^1 & \dots & z_{M,k}^1 \\ \vdots & \vdots & \ddots & \vdots \\ z_{1,k}^{N-1} & z_{2,k}^{N-1} & \dots & z_{M,k}^{N-1} \end{bmatrix} \begin{bmatrix} R_{1,k} \\ R_{2,k} \\ \vdots \\ R_{M,k} \end{bmatrix} = \begin{bmatrix} y_k(0) \\ y_k(1) \\ \vdots \\ y_k(N-1) \end{bmatrix} \quad (16)$$

2.3. Cost Function

The cost function defined in (3) is a minimization problem, seeking to use the cost function as a metric to minimize the guess of the modes, α . In the IMP method, however, the cost function is used as a metric to determine which signal will be added for consideration in the next iteration. As a result, we reformulate (3), defining the cost function as $c(t)$.

$$c(t) = \|r(t)\|_N = \|y(t) - \hat{y}(t)\|_N \quad (17)$$

The primary difference between (17) and (3) is that the cost function is no longer a minimization problem,

but simply a mathematical definition dependent not on an initial guess α , but on the time t . We have also normalized the residual to the number of points N , instead of simply applying a Euclidean norm. Because we are looking to measure how accurate our reproduced signal is, the cost function is chosen due to its direct comparison between the reproduced data and the actual data. It should also be noted that while applying a Euclidean norm would imply that the residuals may be described using linear relationship, invoking a p-norm would more accurately help to consider the non-linear nature of the residuals.

Through this definition of the cost function, and its use in the IMP method, computation times may be considerably shortened. By quickly calculating the cost functions for all signals, the IMP method selects a significantly smaller subset of signals that are used to calculate the modes and the mode shapes.

3. Case Studies

We apply the IMP method to four different case studies. These are performed on synthetic power networks, which are test systems representing properties of actual large-scale power systems, without revealing confidential information [19, 20, 21, 22]. Because the IMP method was designed to accurately determine modes for large systems, we start with a 1200 bus system, which is a subset of a larger 2000 bus system based on the footprint of Texas. The second set of cases contain 5000 and 10,000 buses, respectively. Again, the 5000 bus case is a subset of the larger 10,000 bus case representing the western United States. The number of buses per area is again dependent on the size of the area. Because the IMP method is a measurement based modal analysis technique, non-linearities are embedded in the data gathered from the synthetic cases.

The statistics of the cost function may be used to measure the accuracy of our reproduced signals. The average cost function measures the overall fit of the reproduced data, while the maximum and minimum cost functions measure the best and worst fits, respectively. All measured values are gathered through the use of PowerWorld Simulator, which has a commercially available implementation of the IMP method.

3.1. 1200 Bus Case

In the 1200 bus simulation, the contingency considered is a generator being opened, one second into the simulation. The frequency at each bus is recorded for 20s, at a sampling rate of 5 Hz beginning at 1s. Each frequency is linearly detrended and scaled by its respective standard deviation. The SVD threshold which

Table 1. 1200 Bus Case Cost Function

	# of Iterations		
	1	10	40
Average $c(t)$	0.00672	0.00225	0.00220
Maximum $c(t)$	0.01363	0.00326	0.00550
Minimum $c(t)$	0.00260	0.00107	0.00095
Time Taken (s)	2.93	9.29	56.1

Table 2. 2000 Bus Case Cost Function

	# of Iterations		
	1	10	40
Average $c(t)$	0.00684	0.00229	0.00226
Maximum $c(t)$	0.01386	0.00425	0.00555
Minimum $c(t)$	0.00260	0.00106	0.00095
Time Taken (s)	4.39	12.7	62.22

we use to determine the number of singular values is set to be 0.025.

Table 1 presents the metrics of interest for three different sets of iterations. When there is only one iteration, we are attempting to reproduce all 1200 signals depending only on the first available signal in the data set. With 10 and 40 iterations, we reproduce the signals with a larger set of sampled signals.

From 1 iteration to 10 iterations, the average cost function decreases 66.5%, indicating an overall better fit at 10 iterations. Likewise, the maximum and minimum cost functions see a decrease of 76.1% and 58.9%, respectively. However, the time taken for the IMP method to determine the modes increases from 2.93s to 9.29s.

When increasing the iteration count from 10 to 40, the average and minimum cost function continue to decrease, although at a lower rate. The maximum cost function, on the other hand, sees an increase of 68.7%. This indicates that there is an upper limit to the number of iterations that should be used, as there is an increase in computation time, from 9.29s to 56.1s.

3.2. 2000 Bus Case

The frequencies for all 2000 buses are again measured for a period of 20s, at a sampling rate of 5 Hz. The frequencies are linearly detrended and scaled by their standard deviation, similar to the 1200 bus case. The SVD threshold is set to be 0.025 as before.

The results of the IMP method are seen in Table 2. We see similar trends as the 1200 bus case, with a 66.5% decrease in the average cost function, a 69.3% decrease in the maximum cost function, and a 59.2% decrease in the minimum cost function. Also like the 1200 bus case, there is an indication that an upper limit for the number of iterations exists, as the maximum

Table 3. 5000 Bus Case Cost Function

	# of Iterations		
	1	10	40
Average $c(t)$	0.0124	0.0027	0.0026
Maximum $c(t)$	0.0324	0.0041	0.0044
Minimum $c(t)$	0.0013	0.0011	0.0010
Time Taken (s)	10	27.89	102.98

Table 4. 10,000 Bus Case Cost Function

	# of Iterations		
	1	10	40
Average $c(t)$	0.01194	0.00279	0.00238
Maximum $c(t)$	0.03244	0.00499	0.00930
Minimum $c(t)$	0.00121	0.00179	0.00098
Time Taken (s)	19.82	58.72	170.33

cost function increases when considering 10 iterations versus 40 iterations. The need for an upper limit on the number of iterations is exacerbated by the increase in computation time, from 12.7s to 62.22s.

3.3. 5000 Bus Case

The 5000 bus case is a subset of a larger, 10,000 bus synthetic power grid, which models the Western US. The contingency considered is again the opening of a generator at time $t = 1s$. The measurement parameters remain the same as in the previous two scenarios, so does the detrending and SVD threshold. The results are shown in Table 3, which confirm the same trends as seen in the 1200 and 2000 bus case.

It is important to note that as we continue to increase the number of buses, the difference in computation time between the iterations begins to increase at a rate higher than previously considered cases. While the 2000 bus case had a 49.52s increase in computation time from 10 to 40 iterations, resulting in a 1.3% decrease in the average cost function, the 5000 bus case has a decrease in the average cost function of 4.1%, but an increase in computation time of 75.09s.

3.4. 10,000 Bus Case

The final case considered is the synthetic 10,000 bus power grid. The bus frequencies are measured for 20s, and sampled at a sampling rate of 5 Hz. The data is linearly detrended and scaled by the standard deviation before implementing the IMP method. The SVD threshold for this case is also set to 0.025. The results of the IMP method are recorded in Table 4.

The cost functions may further be visualized through the use of a contour [23], seen in Figure 1. Here, a color scale has been set on the range [0,0.005], for the

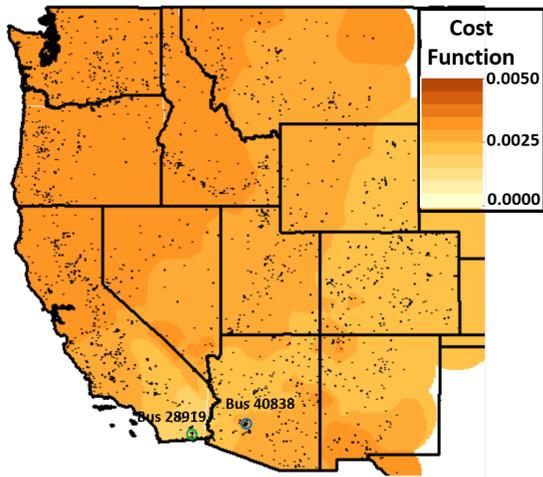


Figure 1. Contour of Cost Functions for 10,000 Bus Case

10 iteration case, to capture the extreme values of the cost functions observed at two generators; the minimum cost function of .00179 being at Bus 28919, and the maximum cost function of .00499 being at Bus 40838. These buses are enclosed by the green and blue circles, respectively. In addition to the small nodal cost function values, the narrow range of wide-area cost function variation across the grid is indicative of the strong signal matching ability of the proposed IMP method.

The most notable behavior of the cost function for the 10,000 bus case is the increase of the maximum cost function when transitioning from 10 iterations to 40 iterations. Further analysis into this reveals that the increase in the maximum cost function of 86.4% is due to a single signal whose fit is negatively impacted by the number of iterations, while all other signals see trends similar to those of smaller cases. As such, we explore why this trend exists further in Section 4.1.

4. Sensitivity Considerations

The results of the IMP method can be highly dependent on two primary user-based inputs; the number of iterations, and the SVD threshold. The behavior of the cost function as the number of iterations increases was briefly explored in Section 3, and it was found that there exists an appropriate range for the number of iterations which minimizes the cost function and the time taken.

The same analysis may be applied to the SVD threshold. The SVD threshold q is also user-dependent, and as such, different thresholds may produced different results. While the SVD threshold was maintained to be 0.025 in all case studies for consistency, it is important

to explore how the different metrics are affected by changes in the SVD threshold.

4.1. Number of Iterations

Figure 2 shows how the average cost function of each of the four case studies changes with the number of iterations. The average cost function decreases exponentially until 10 iterations, after which it begins to stabilize. In the 1200 bus case, after 10 iterations, the average cost function has an average value of 0.00221, with a standard deviation of 3.2×10^{-5} , or about 1.5% of the average. This indicates that in the smallest case considered, the average cost function does not fluctuate much after 10 iterations. It is also important to note that the average cost function drops 63.91% in the 1200 bus case by 5 iterations, with similar decreases occurring for other cases considered.

A similar trend is observed in the 2000 bus case, where the measured values begin to converge to an approximate value of 0.00225 with a standard deviation of 3.2×10^{-5} . The 5000 bus and 10,000 bus cases behave similarly, with a rapid decrease in the average cost function as we approach 10 iterations. However, the behavior after 10 iterations is not entirely consistent with that of the smaller cases. In the 5000 bus case, we see that the average cost function actually begins to steadily increase again after 10 iterations, and only decreases after 34 iterations. In the 10,000bus case, however, the average cost sees a decrease after 17 iterations, which is when the cost function averages out to a value of .00241, with a standard deviation of 1.01×10^{-5} , or about 4.2%.

The results of the average plot indicate that there is not a substantial difference in the average cost function when considering 10 iterations versus 40 iterations.

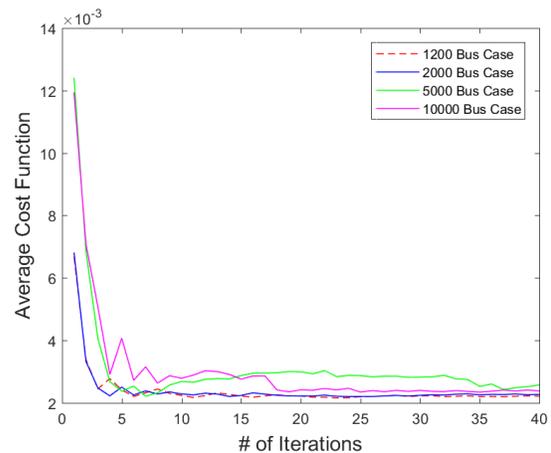


Figure 2. Average $c(t)$ vs # of Iterations

As a direct result of this, we may state that for

large bus cases, having a minimum of 10 iterations is a benchmark which would provide accurate results. However, while the average cost function gives an indication of the lower limit of the number of iterations, it does not provide enough insight on the upper limit.

Figure 3 provides insight on how the maximum cost function varies with the number of iterations. We see the same trend as seen in the average cost function; after 10 iterations, the maximum cost functions for the 1200, 2000, and 5000 bus case do not vary as much as they did before 10 iterations. The 10,000 bus case, however, sees an increasing maximum cost function as the number of iterations increases. Tracking the maximum cost function shows that the same signal is associated with it after 10 iterations. As a result, while the average cost function continues to maintain a value within a small range, the maximum cost function increasing is attributed to a single signal which sees an increase in the maximum cost function, while all other signals obey a similar trend to smaller bus cases. When looking at the physical location of the signal, we see that it is located close to the location of the contingency; this would imply that while adding more signals improves the overall quality of the approximation, signals with a high number of oscillations that were already being considered may have a worse representation as we increase the number of iterations.

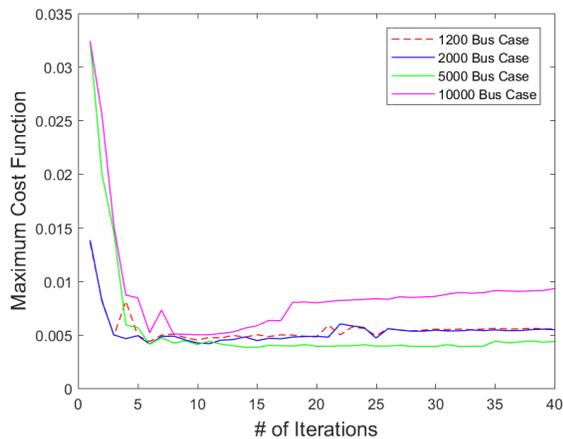


Figure 3. Maximum $c(t)$ vs # of Iterations

While Figure 3 indicates that large changes can occur when considering a varying number of iterations for the maximum cost function in large cases, Figure 2 still shows an unsubstantial change in the overall quality of fit. As a consequence of this, we look to Figure 4 helps to provide insight into another consideration for modal analysis; the time taken to determine the modes. When considering the time for smaller bus cases such as

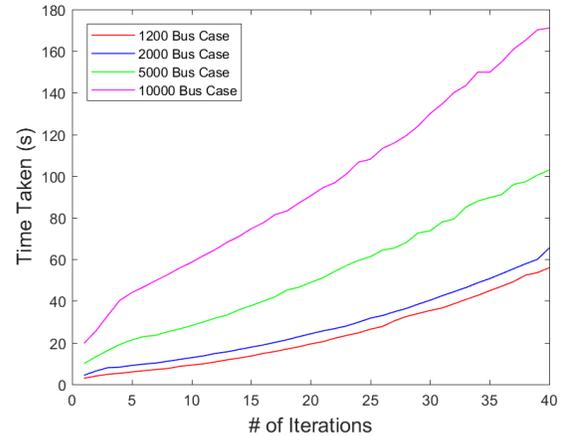


Figure 4. Time Taken (s) vs # of Iterations

the 1200 bus and 2000 bus case, the time taken increases quickly as the number of iterations increases, while large cases see a slower increase between the number of iterations and the time taken to determine the modes.

When considering time and the cost function simultaneously as in Figure 4, several conclusions may be drawn about a recommended number of iterations that minimize both the cost function and the amount of time that the IMP method requires. For the 1200 and 2000 bus case, 10 iterations suffice, as there are no significant improvements in the average or maximum cost function beyond this point, while the time increases. When increasing the number of buses to 5000, the recommended number of iterations increases to 13, as after this point, the maximum cost function achieves a minimum and remains within a small range of values. Finally, for the 10,000 bus case, while 18 iterations increases the maximum cost function by 21.2% from 17 iterations, the overall fit sees an improvement of 15.6%. This point may be further emphasized by visualizing the reproduced signals of the 10,000 bus case.

Figures 5 and 6 show the reproduced data from the IMP method, and compare it to the original data for the minimum and maximum cost function, respectively. Here, the conditions of the IMP method are the same as previously stated, but the iteration count is set to 18. We see that the minimum cost function sees almost no discrepancies between the reproduced data and the actual data, indicating a strong fit. Even in the worst case scenario shown in Figure 6, the discrepancies are more pronounced, but the overall fit still shares a majority of the points with the actual data.

The number of iterations is noted to have minor impacts on the overall fit for cases of roughly equivalent size to the 2000 bus case, but creates much more impact

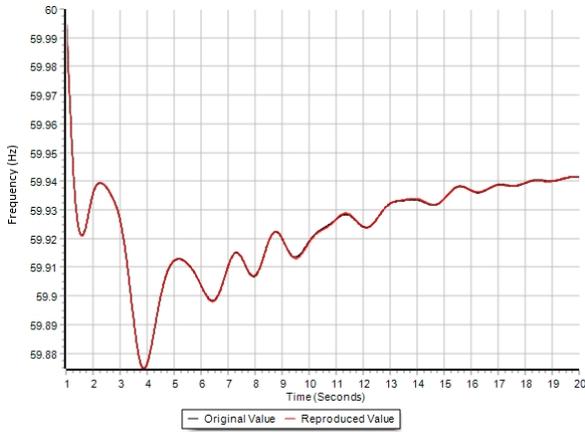


Figure 5. Reproduced Data and Original Data of Minimum $c(t)$ for # of Iterations = 18

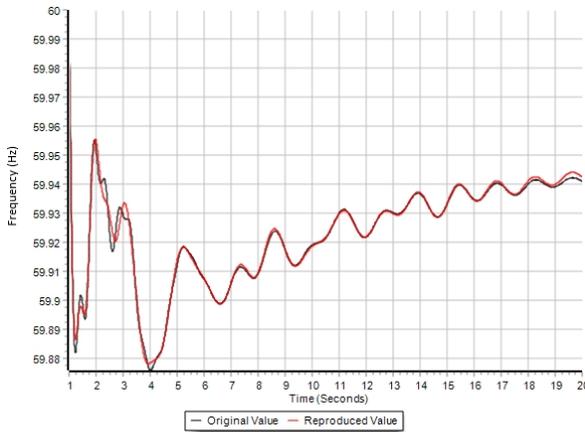


Figure 6. Reproduced Data and Original Data of Maximum $c(t)$ for # of Iterations = 18

when considering how long the IMP takes. In larger systems, the number of iterations plays a vital part in maintaining the balance between accurate results and time minimization. The sensitivities of the cost function to the number of iterations indicate that if one does not carefully choose the number of iterations, results may take longer to achieve, and may not even show any tangible benefits. This is especially important in real time analysis, where the most accurate results must be achieved in the shortest amount of time.

4.2. SVD Threshold

When discussing the sensitivity of the average and maximum cost function to the SVD threshold, an initial threshold range from 0.025 to 1 shows increases in the cost function without an appreciable difference in the time saved after a SVD threshold of 0.2. As a result,

we consider a narrower range of 0.025 to 0.2 for a more thorough discussion on the SVD threshold.

Figure 7 shows how the average cost function evolves as we change the SVD threshold. Here, we see that as we increase the value of the SVD threshold, the average cost function begins to increase, with a large amount of oscillations.

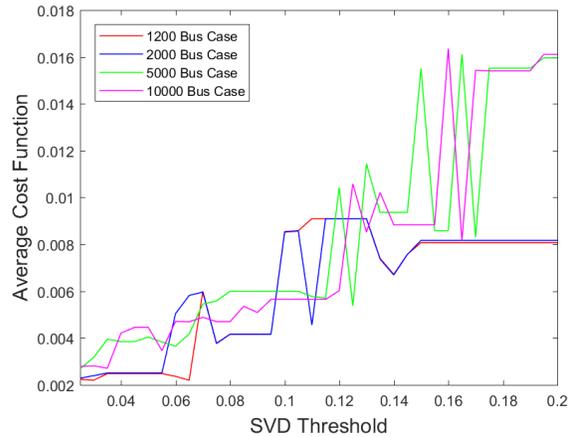


Figure 7. Average Cost Function vs Reduced SVD Threshold

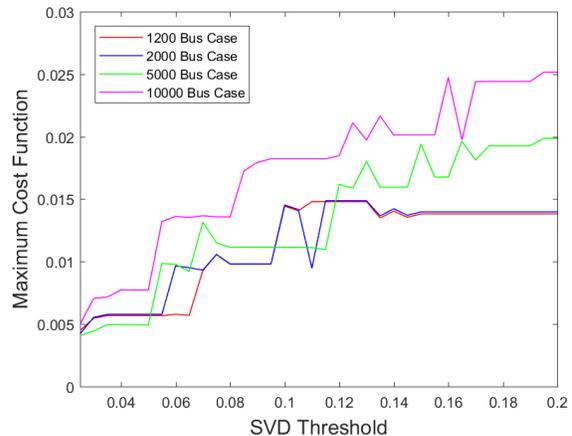


Figure 8. Maximum Cost Function vs Reduced SVD Threshold

These oscillations are also apparent in Figure 8, which shows how the maximum cost function behaves on the same reduced scale. As such, in order to determine an appropriate SVD threshold for each case study, we must weight the magnitude of the oscillations versus the amount of time saved by increasing the SVD threshold, seen in Figure 9.

For the 1200 and 2000 bus case, we see a similar trend persist. While the cost functions continue to rise

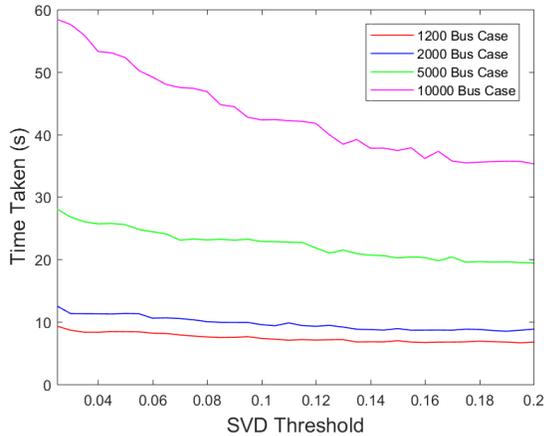


Figure 9. Time Taken vs Reduced SVD Threshold

as we increase the SVD threshold, the time needed to determine the modes does not present a notable change. In the 1200 bus case, the average cost function has more than tripled in magnitude, increasing from 0.00224 to 0.00807, while the time difference between the two thresholds is only 2.56s. Likewise for the 2000 bus case, the average cost function increases from 0.00229 to 0.00817, while the time only decreases by 3.68s.

As Figure 9 shows, the 5000 and 10,000 bus cases see much more significant reductions in time, relative to the prior cases. However, there still exists a noticeable amount of oscillations in the cost function curve. Setting the SVD threshold to 0.105 in the 5000 bus case results in an average cost function of 0.00599. While this average cost function is larger than the average at a SVD threshold of 0.025, there is a time difference of 5.2s. Because contingencies and their effects on a power grid can happen in very short amounts of time (e.g. ms), this reduction in time can be of vital importance in helping to determine how to best control and stabilize a large power grid. The same effect is observed in the 10,000 bus case, where increasing the SVD threshold to 0.115 increases the average cost function from 0.00279 to 0.00565, but saves 16.33 seconds, a reduction in computation time of 27.9%.

These trends may again be visualized for clarity through Figures 10 and 11, where we consider the best and worst reproduced data for the 10,000 bus case. The number of iterations was maintained at 10, but the SVD threshold was set to be 0.06. In this instance, we see that the best case scenario still has a close fit to the original data, even with an increase in the SVD threshold. Likewise, while the worst case scenario in Figure 11 has noticeable differences between the reproduced data and the actual data, there is still a close fit for a majority of

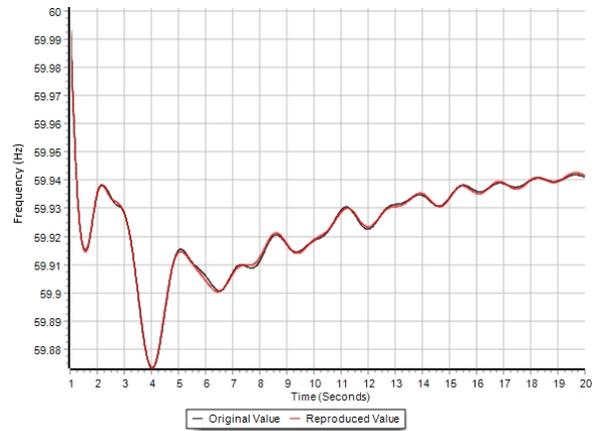


Figure 10. Reproduced Data and Original Data of Minimum $c(t)$ for SVD Threshold = 0.06

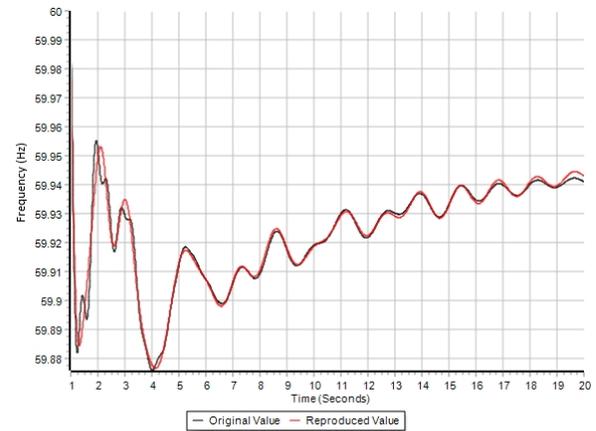


Figure 11. Reproduced Data and Original Data of Maximum $c(t)$ for SVD Threshold = 0.06

the data points, indicating an accurate reproduction of the original data.

While adjusting the SVD threshold has been shown to have overall negative effects on the average and maximum cost function, the value of reducing computation time during contingencies may sometimes outweigh said negative effects. In smaller bus cases such as the 1200 and 2000 bus case, increasing the SVD threshold holds very little merit; decreases in computation time are far outweighed by the effects on the overall quality of the reproduced data. However, when handling significantly larger power grids, adjusting the SVD threshold may potentially have more pronounced effects on the overall fit, but careful consideration may lead to minimized negative impacts, while improving the speed at which modes are determined.

5. Conclusion

The paper introduced a modal analysis method, which builds on existing techniques, termed as the Iterative Matrix Pencil method. The significance of this method lies in the quick calculation of modes for a large system involving multiple signals. The accuracy of the method was tested through comparisons between signals reproduced from the modes and their associated original signals, showing a strong match between the two.

The IMP method possesses user-defined inputs, and as such, the sensitivity of the method to both the number of iterations and the SVD threshold must be considered. Maintaining the number of iterations in the tens has shown to minimize the average cost function for cases in which there are thousands of buses. Likewise, maintaining the SVD threshold at 0.025 also minimizes the average cost function, but increasing it to 0.06 greatly improves the time taken for the IMP method. Visualizations of the cost function across large grids and its evolution provide a clearer understanding of how the IMP method works. Future work seeks to test the IMP method in scenarios of poor data quality, along with considering how the IMP method performs compared to other modal analysis techniques designed to handle a large number of signals, such as DMD.

References

- [1] S. Okubo, H. Suzuki, and K. Uemura, "Modal analysis for power system dynamic stability," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-97, pp. 1313–1318, July 1978.
- [2] M. Klein, G. J. Rogers, and P. Kundur, "A fundamental study of inter-area oscillations in power systems," *IEEE Transactions on Power Systems*, vol. 6, pp. 914–921, Aug 1991.
- [3] Z. Tashman and V. M. Venkatasubramanian, "Modal energy trending for ringdown analysis in power systems using synchrophasors," in *2014 47th Hawaii International Conference on System Sciences*, pp. 2475–2482, Jan 2014.
- [4] Z. Huang, N. Zhou, F. Tuffner, and D. J. Trudnowski, "Use of modal sensitivity to operating conditions for damping control in power systems," *2011 44th Hawaii International Conference on System Sciences*, pp. 1–9, 2011.
- [5] G. P. K and J. W. Pierre, "New robust time series model for estimation of power system inter-area oscillations," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, vol. 00, pp. 2472–2480, Jan. 2016.
- [6] J. F. Hauer, C. J. Demeure, and L. L. Scharf, "Initial results in prony analysis of power system response signals," *IEEE Transactions on Power Systems*, vol. 5, pp. 80–89, Feb 1990.
- [7] L. L. Grant and M. L. Crow, "Comparison of matrix pencil and prony methods for power system modal analysis of noisy signals," in *2011 North American Power Symposium*, pp. 1–7, Aug 2011.
- [8] M. L. Crow and A. Singh, "The matrix pencil for power system modal extraction," *IEEE Transactions on Power Systems*, vol. 20, pp. 501–502, Feb 2005.
- [9] Y. Hua and T. Sarkar, "Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise," *IEEE Transactions on Signal Processing*, vol. 38, pp. 814–824, 5 1990.
- [10] A. R. Borden, B. C. Lesieutre, and J. Gronquist, "Power system modal analysis tool developed for industry use," in *2013 North American Power Symposium (NAPS)*, pp. 1–6, Sept 2013.
- [11] A. R. Borden and B. C. Lesieutre, "Variable projection method for power system modal identification," *IEEE Transactions on Power Systems*, vol. 29, pp. 2613–2620, Nov 2014.
- [12] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, p. 528, 2010.
- [13] E. Barocio, B. Pal, N. Thornhill, and A. Messina, "A dynamic mode decomposition framework for global power system oscillation analysis," vol. 1, 12 2014.
- [14] S. Mohapatra and T. J. Overbye, "Fast modal identification, monitoring, and visualization for large-scale power systems using dynamic mode decomposition," in *2016 Power Systems Computation Conference (PSCC)*, pp. 1–7, June 2016.
- [15] B. P. Wiseman, Y. Chen, L. Xie, and P. R. Kumar, "Pmu-based reduced-order modeling of power system dynamics via selective modal analysis," in *2016 IEEE/PES Transmission and Distribution Conference and Exposition (TD)*, pp. 1–5, May 2016.
- [16] J. Barquin, L. Rouco, and H. R. Vargas, "Generalized selective modal analysis [power system control simulation]," in *2002 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No.02CH37309)*, vol. 2, pp. 1194–1199 vol.2, 2002.
- [17] A. B. Birchfield and T. J. Overbye, "Convergence characteristics of the variable projection method for mode extraction," in *2017 IEEE Texas Power and Energy Conference (TPEC)*, pp. 1–6, Feb 2017.
- [18] T. K. Sarkar and O. Pereira, "Using the matrix pencil method to estimate the parameters of a sum of complex exponentials," *IEEE Antennas and Propagation Magazine*, vol. 37, pp. 48–55, Feb 1995.
- [19] <https://electricgrids.engr.tamu.edu/electric-grid-test-cases/>.
- [20] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Transactions on Power Systems*, vol. 32, pp. 3258–3265, July 2017.
- [21] T. Xu, A. B. Birchfield, K. S. Shetye, and T. J. Overbye, "Creation of synthetic electric grid models for transient stability studies," in *The 10th Bulk Power Systems Dynamics and Control Symposium (IREP 2017)*, 2017.
- [22] T. Xu, A. B. Birchfield, and T. J. Overbye, "Modeling, tuning and validating system dynamics in synthetic electric grids," *IEEE Transactions on Power Systems*, pp. 1–1, 2018.
- [23] T. J. Overbye and J. D. Weber, "Visualization of power system data," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pp. 7 pp.–, Jan 2000.