Report 2/6/2017

Abstract

One important step towards the creation of a ”Smart Grid” — that is, a self-intelligent power grid, requires the development of some key algorithms. This paper introduces a new method for diagnosing line failures and other contingencies using dynamic data gathered from phasor measurements, something not explored deeply in the current literature. We also modify this method to be more computationally tractable for practical use, and demonstrate high levels of accuracy on IEEE test systems.

1 Introduction

Power Grids are a crucial piece of infrastructure and also among the largest physical networks on the planet. One important step towards the creation of a ”Smart Grid” — that is, a self-intelligent power grid, requires the development of some key algorithms. In particular, identifying changes to the grid structure with on-the-fly is not only a very important problem to power engineers but also an interesting problem in its own right.

2 Mathematical Model of Power System

2.1 Nomenclature

$x \in \mathbb{R}^n$ =state variables in differential equations

$y \in \mathbb{R}^m$ = state variables in algebraic equations

$\mathcal{J} \in \mathbb{R}^{(n+m)\times(n+m)}$ = Jacobian Matrix of DAE system

2.2 Power System Dynamics

A power system is given by a system of differential algebraic equations (DAE)

$$\Omega = \begin{cases} x' = f(x, y) \\ 0 = g(x, y) \end{cases} \tag{1}$$

We note that x are the differential variables and y are the algebraic variables. We also have the initial conditions $x(t_0) = x_0$ and $y(t_0) = y_0$. General DAE theroy states that close to steady state, this nonlinear DAE is closely approximated the linear DAE [5]

$$Ez' = \mathcal{J}z \tag{2}$$

Where $E, \mathcal{J}$, and z are defined as

$$z = \begin{bmatrix} x \\ y \end{bmatrix} \quad \mathcal{J} = \begin{bmatrix} F_x & F_y \\ G_x & G_y \end{bmatrix} \quad E = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \tag{3}$$

Where $\mathcal{J}$ is the Jacobian of the DAE System. Then the solution to (2), and thus the approximate solution to (1), has the form

$$z(t) = \sum_{i=1}^{k} c_i e^{\mu_i t} v_i = \sum_{i=1}^{k} e^{\mu_i t} d_i \tag{4}$$

Where $\langle \mu_i, v_i \rangle$ are the non-infinite *eigenpairs* of the generalized eigenvalue problem $\mathcal{J} v_i = \mu_i E v_i$ [3], the v_i are restricted to have unit length, and c_i are some constants determined by the initial condition. We can fold in our constants into our eigenvectors i.e. rewrite our solution into a slightly more compact form with $d_i = c_i v_i$.

If one has a set of sensors, known as Phasor Measurement Units or PMUs, placed around the power system outputting a signal $f(t)$, then $f(t)$ can be explained mathematically as

$$f(t) = Hz(t) = \sum_{i=1}^{k} e^{\mu_i t} H d_i \tag{5}$$

Note that H simply picks out a subset of $z(t)$, since voltage readings are in the set of algebraic variables.

3 Related Works

In [8], the authors use changes in phase angle to diagnose line failures. Enumerating all possible contingency models, they find one most likely to explain changes in phase angle via some quick linear algebra. They demonstrate success both on a 37 bus model as well as a 7000 bus real world example. Their method seems fast and practical; however, they only use steady-state information and do not post full accuracy results.

In both [2] and [9], the problem of line failures is first reformulated as an additional set of power injections rather than changes to the system topology itself. Changes in phase angle are considered. This problem of identifying pairs of injections explaining changes in angle is then reformulated as a mixed-integer programming problem. The former tests on the 118 bus model while the latter tests on the 37 bus model. This formulation is elegant as it does not require the explicit enumeration of all possible contingencies. The former approach is able to deal with the full non-linear model, while the latter approach tackles double line outages. Problematically, these methods do not seem scalable due to the mixed-integer programming problem. Once again, steady-state is assumed.

In [7], the authors take a different approach and perform system identification/state estimation to diagnose a line failure. Steady state is assumed, and they test in the 7 bus system with success, but this method does not seem scalable and as the authors note, is not as robust to noise as other methods. Estimating loads, according to the authors, is of particiular difficulty.

In [6], the authors look at changes in voltage magnitude instead of phase angle. Assuming steady-state, they adapt a quick filtering procedure to identify a number of different contingencies, including line failures, load trips, and substation reconfigurations. They test on the 56 Bus, 118 Bus, and Polish Systems —they are also the only ones to post performance numbers.

All of these related works wait for oscillations to die out before looking purely at the steady-state. While this is convenient, in our opinion, the oscillations contain valuable information about the dynamics of the system itself. These dynamics, containing a greater volume "of data" than the steady-state, thus should be able to more accurately diagnose contingencies than existing methods.

4 Problem Set Up

Then given the mathematical formulation in the previous section, we may boil down the problem of contingency identification into the following: given a PMU signal $f(t)$ and a dictionary of contingency models

$$\mathcal{D} = \{\Omega_1, \Omega_2, \ldots, \Omega_n\}$$

determine which contingency model Ω_c "best" explains $f(t)$. We want to do so in a mathematically rigorous yet efficient manner. At a high level, the method that we have developed can be enumerated in a number of steps.

1. Fit a set of damped sinusoids to $f(t)$. From this fitting, we extract the partial eigenpair $\{\ \langle \lambda_j, Hv_j \rangle\ \}_{j=1}^{k}$
2. For each Ω_i in $\mathcal{D}$, and for all partial eigenpairs $\langle \lambda_j, Hv_j \rangle$, we retrieve the full eigenpairs via least-squares fitting, yielding the eigenpairs $\langle \lambda_j, {v_j}^{(i)} \rangle$.
3. From the residuals defined by $r_{ij} = \|(E\lambda_j - \mathcal{J}_i){v_j}^{(i)}\|_2$, calculate the likelihood score of contingency Ω_i as $\mathcal{L}_i = \sum_j w_j r_{ij}$. Take $\arg\min_i \mathcal{L}_i$ to be the contingency identified

Intuitively speaking, these steps work in the following manner

1. Since our signal should take the form of (5), we must extract this explicit form from $f(t)$.
2. We only perceive a subset of the eigenvector from (5), so we must extract the rest via a least-squares. This nets us the fitted eigenpairs $\langle \lambda_j, {v_j}^{(i)} \rangle$. Note that $v_j^{(i)}$ is now also dependent on i, as it has been fit with contingency Ω_i in mind. We expect the fitted eigenpairs of the correct contingency Ω_c most accurately reflect the true eigenpairs of the system jacobian $\mathcal{J}_i$.
3. We move forward by calculating the residual defined by $r_{ij} = \|(E\lambda_j - \mathcal{J}_i){v_j}^{(i)}\|_2$ to be smallest among all possible contingencies. We expect the residuals associated with Ω_c to be smallest. To quantify this, we define the likelihood score $\mathcal{L}_i$ of contingency Ω_i to be a weighted sum of residual norms i.e. $\mathcal{L}_i = \sum_j w_j r_{ij}$, and takes the contingency generating the minimum likelihood score to be the correct one

For each of our n contingencies, we therefore fit k eigenpairs. Those who prefer a more precise algorithm may refer to Figure 1, although it is a bit terse.

4.1 Step One

The fitting of damped sinusoids to a signal possesses a number of solutions, including Prony's Method, Filter Diagonalization, and System Identification. We use a system identification algorithm, 4SID, to do so.

System Identification transforms a set of inputs and outputs into a system Ω in state-observer form:

$$\Omega = \begin{cases} z' = Az + Bu + Ke \\ f = Cz + Du + e \end{cases}$$

Algorithm 1: Contingency Identification

```
Result: The Correctly Identified Contingency $\Omega_c$
1  $\mathcal{D} = \{\Omega_1, \Omega_2, \ldots, \Omega_m\}$                      ▷ Dictionary of Contigs
2  $\{\ \langle \lambda_j, Hv_j \rangle\ \}_{j=1}^{k} = \text{expfit}(f(t))$      ▷ Damped Exponential Fit
3  for $i$ = 1:$n$ do
4      for $j$ = 1:$k$ do
5          $\langle \lambda_j, {v_j}^{(i)} \rangle = \text{eigfit}(\Omega_i,\ \langle \lambda_j, Hv_j \rangle)$
6          $r_j^{(i)} = (E\lambda_j - \mathcal{J}_i){v_j}^{(i)}$
7          $r_{ij} = \|r_j^{(i)}\|_2$
8      end
9      $\mathcal{L}_i = \sum_{j=1}^{k} w_j r_{ij}$
10 end
11 idx = argmin($\mathcal{L}_1$, $\mathcal{L}_2$, $\ldots \mathcal{L}_n$)               ▷ Set Contig idx
12 $\Omega_c = \mathcal{D}[idx]$
```

Figure 1: The basic contingency identification algorithm first sets up a dictionary of contingency models and extracts partial eigenpairs in lines 1 and 2 respectively. Line 5 fits the full eigenpair $v_j^{(i)}$ from the partial eigenpair Hv_j and then Line 7 calculates the residual. Line 9 calculates the likelihood score and lines 11-12 calculate the minimum among all likelihood scores to identify the contingency.

Note that e is an error term and K its disturbance matrix, both of which we will ignore. Since we don't have any inputs, our B and D matrices will be zero. $f(t)$ will thus look like

$$f(t) = Ce^{At} \tag{6}$$

Where in our case, C is our observer matrix emulating the behavior of the system's PMUs. The solution to this ODE is simply

$$f(t) = \sum_{j=1}^{k} e^{\omega_j t} C u_j \tag{7}$$

where $\langle \omega_i, u_i \rangle$ are eigenpairs of A. Note this is precisely the explicit form in 5 we desire. We may thus conclude that

$$\langle \lambda_j, Hv_j \rangle = (\omega_j, Cu_j) \tag{8}$$

4.2 Step Two

Having properly retrieved the pairs $\langle \lambda_j, Hv_j \rangle$, we now proceed to fit the rest of the eigenvector ${v_j}^{(i)}$ outside the range of H using each Ω_i. Let's first assume without loss of generality that

$$||Hv_j||_2^2 = 1 \tag{9}$$

$$Hv_j = p_j\ ,\ {v_j}^{(i)} = \begin{pmatrix} p_j \\ q_j \end{pmatrix} \tag{10}$$

(one may always apply a normalization and a permutation otherwise). That is to say, the partial eigenvectors Hv_j are normalized to have unit length and sit in the top of the full eigenvector v_j.

These assumptions make the math a bit simpler. Then we formulate the eigenvector fitting as

$$\min_{\alpha,q_j} \left\| \begin{pmatrix} E\lambda_j - \mathcal{J}_i \end{pmatrix} \begin{pmatrix} \alpha p_j \\ q_j \end{pmatrix} \right\|_2^2 \tag{11}$$

s.t.

$$\left\| \begin{pmatrix} \alpha p_j \\ q_j \end{pmatrix} \right\|_2^2 = 1 \tag{12}$$

Intuitively, we want to find the vector of unit length in the null space of $E\lambda_j - \mathcal{J}_i$ given that its first entries contain Hv_i. Manipulating the problem slightly, we get

$$\min_{\alpha,q_j} \left\| \left[\begin{pmatrix} E\lambda_j - \mathcal{J}_i \end{pmatrix} \begin{pmatrix} p_j & 0 \\ 0 & I \end{pmatrix} \right] \begin{pmatrix} \alpha \\ q_j \end{pmatrix} \right\|_2^2 \tag{13}$$

s.t.

$$\left\| \begin{pmatrix} \alpha \\ q_j \end{pmatrix} \right\|_2^2 = 1 \tag{14}$$

Where I is the identity matrix of proper size. Letting

$$G_{ij} = \left[\begin{pmatrix} E\lambda_j - \mathcal{J}_i \end{pmatrix} \begin{pmatrix} p_j & 0 \\ 0 & I \end{pmatrix} \right] \quad z_j = \begin{pmatrix} \alpha \\ q_j \end{pmatrix} \tag{15}$$

We can rewrite the problem in a more compact form as

$$\min_{z_j} \|G_{ij} z_j\|_2^2 \quad s.t. \quad \|z_j\|_2^2 = 1 \tag{16}$$

Standard Linear Algebra results tell us that the solution to (16) is given by λ_j^s, the smallest eigenvalue of $G_{ij}^T G_{ij}$. Then the jth residual for contingency i, r_{ij} is given by λ_j^s

4.3 Step Three

Now that we've calculated all the residuals r_{ij} for $i = 1 \dots m$, $j = 1 \dots k$, we need to use them to diagnose the contingency. As mentioned before, one would expect the correct diagnosis to have small residuals. In this light, we can rank the likelihood of a contingency based on the size of its residuals; we consequently introduce the likelihood score $\mathcal{L}_i$ of contingency Ω_i as

$$\mathcal{L}_i = \sum_{j=1}^{k} w_j r_{ij} \tag{17}$$

That is to say, the likelihood score $\mathcal{L}_i$ is simply the weighted sum of all residual norms associated with contingency Ω_i. Naively speaking, one might choose to set the weights equal to one to equally weight all residuals. However, this does not account for spurious eigenpairs caused by nonlinearities, small disturbances, or noise. As we expect these spurious eigenvectors to be rather small in amplitude when extracted, we thus set the weights to simply be the norm of the associated eigenvector. To be more precise

$$w_j = \|Hv_j\|_2 = \|Cu_j\|_2 \tag{18}$$

Where Cu_j is defined in (7).

4.4 A Few Optimizations

Some optimizations might include

- Throwing out eigenpairs with associated frequencies outside the Shannon-Nyquist bound and associated dampening factors decaying to machine precision within the first second of the signal.
- Throwing out eigenpairs with associated residual weights under a certain bound.
- Leaving G_{ij} implicit when calculating its smallest singular vector via some iterative method.

5 Initial Results

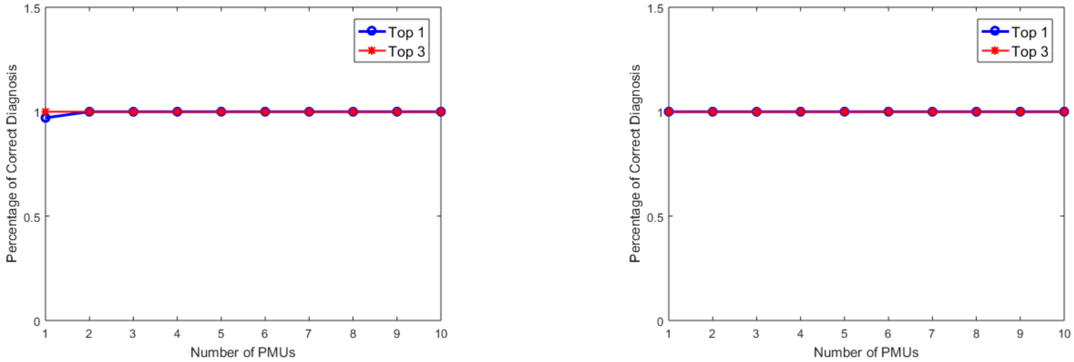

Figure 2: Results for the 39 bus system (left) and 14 bus system (right) with a single PMU configuration on the 16th and 3rd bus respectively. Our results show that we are able to correctly identify the contingency almost every single time with the exception of one contingency on the 39 bus system.

Using Matlab, we tested our method on the IEEE 14 and 39 bus systems. We considered contingencies in the form of line failures, and simulated the dynamics of the contingency via PSAT, a popular, open source power system tool. Our simulations were performed with a fixed timestep of 0.005. We also assumed our PMUs polled at 20 times per second.

Our results show that we are able to correctly identify the contingency almost every single time (with the exception of one contingency) with only a single PMU placed on the system. We are also able to identify the top three most likely contingencies with full accuracy, and once we have two PMUs on the 39 bus system we fix our single misidentification as well.

Further investigation reveals that our single misidentification in the 39 bus system case occurred when one of two sequential lines was disconnected; our method believed the first was disconnected when the second was. The fingerprint scores of misdiagnosed contingency and the correct one were almost identical and much smaller than the rest. Placing another PMU near the area resulted in correct diagnosis. This misdiagnosis suggests not immediately ruling out contingencies with likelihood scores close to the minimum.

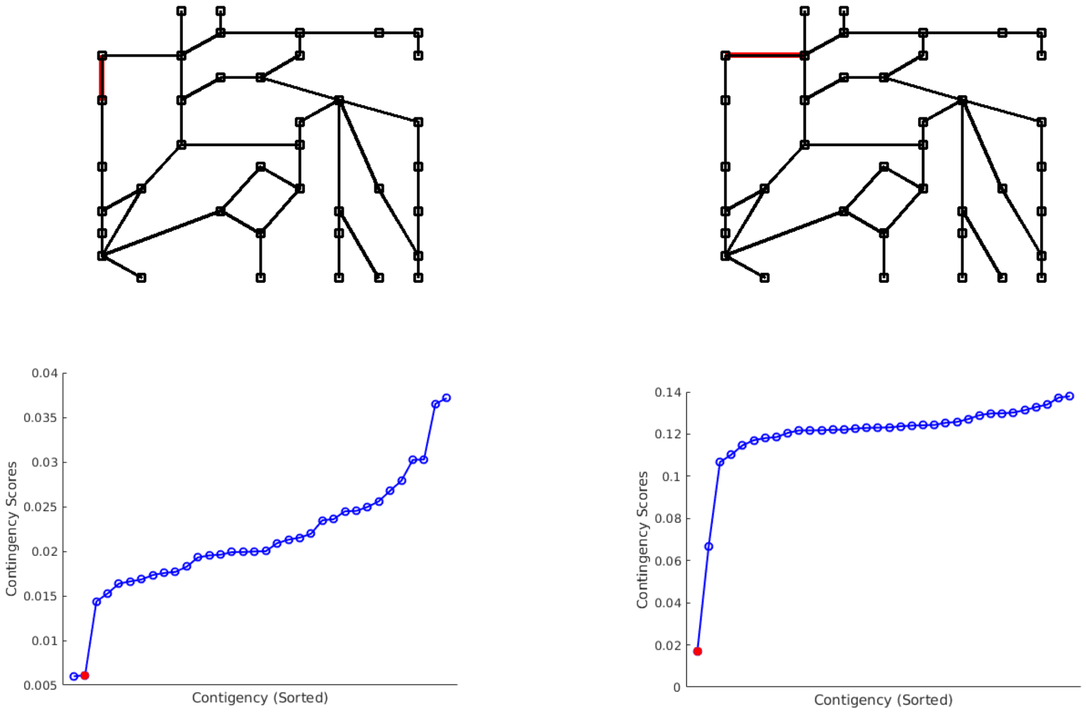

Figure 3: On the top left: a network diagram for the line failure incorrectly identified (highlighted in red) with a single PMU placed in the middle of the network. On the bottom left: the likelihood scores as calculated and sorted, with identified contingency on the far left and the correct contingency marked in red. Note that even though the contingency is incorrectly identified, it is almost identical in score to the correct one. On the top right: a network diagram for the correct line failure, correctly identified after putting a PMU near the line failure itself. On the bottom right: note that the likelihood score gap decreased in variance and increased in mean with the addition of more PMUs, allowing us to correctly identify the contingency

6 Decreasing Computational Cost via Filtering

Because we are looking for the minimum likelihood score among all possible contingencies, we may save the current minimum likelihood score obtained so far. Whenever a contingency Ω's likelihood score exceeds the current minimum's, we may then throw out Ω.

This straightforward idea gives us the basis for an efficient filtering procedure. Recall that each likelihood score is a weighted sum of residuals, with each residual derived from an eigenvector fitting. One the current likelihood score exceeds the min, we stop and throw our the current contingency, which incurs savings by decreasing the number of residual calculations, and thus the number of eigenvector fittings required. We expect this to work well due to the large gap between likehihood scores, as seen in Figure 5

To be more precise, we modify the earlier method. Given a PMU signal $f(t)$ and a dictionary of contingency models.

$$\mathcal{D} = \{\Omega_1, \Omega_2, \dots, \Omega_m\}$$

Algorithm 2: Contingency Identification with Filtering

```
Result: The Correctly Identified Contingency $\Omega_c$
1  $\mathcal{D} = \{\Omega_1, \Omega_2, \ldots, \Omega_m\}$                    ▷ Dictionary of Contigs
2  $\{ \langle \lambda_j, Hv_j \rangle \}_{j=1}^{k} = \text{expfit}(f(t))$        ▷ Damped Exponential Fit
3  $MIN = \infty$
4  for $i$ = 1:n do
5      for $j$ = 1:k do
6          $\langle \lambda_j, {v_j}^{(i)} \rangle = \text{eigfit}(\Omega_i,\ \langle \lambda_j, Hv_j \rangle)$
7          $r_j^{(i)} = (E\lambda_j - \mathcal{J}_i){v_j}^{(i)}$
8          $r_{ij} = \|r_j^{(i)}\|_2$
9          $\mathcal{L}_{ij} = \sum_{j=1}^{k} w_j r_{ij}$
10         if $\mathcal{L}_{ij} > MIN$ then
11             break                         ▷ Break if Likelihood Score exceeds MIN
12         end
13         if $j == k$ then
14             $MIN = \mathcal{L}_{ij}$                    ▷ Update MIN and Contig idx
15             $idx = i$
16         end
17     end
18 end
19 $\Omega_c = \mathcal{D}[idx]$
```

Figure 4: The filtered contingency identification algorithm is quite similar to the first. First it sets up a dictionary of contingency models and extracts partial eigenpairs in lines 1 and 2 respectively. Line 3 sets MIN, the current minimum likelihood score, to be infinity. Line 6 fits the full eigenpair from the partial eigenpair Hv_j and then Line 8 calculates the residual. Line 10 checks if the likelihood score $\mathcal{L}_{ij}$ has exceeded MIN; if it has we break as contingency Ω_i may be discounted. Line 13 updates MIN when with $\mathcal{L}_{ij}$ and records down i as the current most likely contingency.

1. Set the minimum recorded likelihood score as $MIN = \inf$
2. Determine an ordering in which to evaluate contingencies in $\mathcal{D}$. Without loss of generality, we assume $\mathcal{D}$ is already ordered properly for notational convenience
3. Fit a set of damped sinusoids to $f(t)$. From this fitting, we extract the eigenpair "subsets" $\langle \lambda_j, Hv_j \rangle$, for $j = 1 \ldots k$
4. For each Ω_i in $\mathcal{D}$ and all partial eigenpairs $\langle \lambda_j, Hv_j \rangle$, for $j = 1 \ldots k$, we retrieve the full eigenpairs via least-squares fitting, yielding the eigenpairs $\langle \lambda_j, {v_j}^{(i)} \rangle$. From the residuals defined by $r_{ij} = \|(E\lambda_j - \mathcal{J}_i){v_j}^{(i)}\|_2$, calculate the partial likelihood score $\mathcal{L}_{ij} = \sum_0^j w_j r_{ij}$
5. If the $\mathcal{L}_{ij} > MIN$, throw out Ω_i. If the $\mathcal{L}_{ij}$ is fully computed, set $MIN = \mathcal{L}_{ij}$

The precsie algorithm is outlined in figure 2:

This filtering only works well if we can calculate the correct contingency's likelihood score early, as this would allow one to discount more contingencies. Of course, possessing a method guaranteed to calculate the correct contingency's likelihood score earliest is the same as contingency identification itself. However, we can utilize a computationally cheaper method to determine an approximate ordering. This may be done with a faster but less accurate contingency identification method, such as [6], but we opted for a simple and cheap least-squares fit.

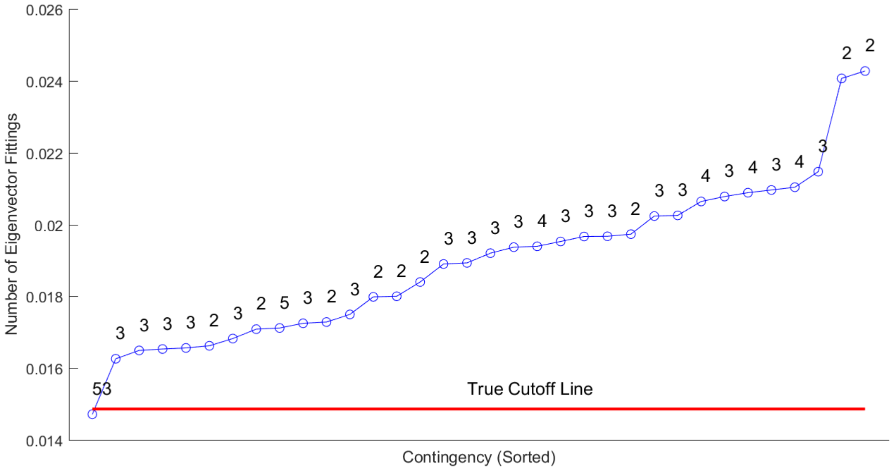

Figure 5: This figure sorts the likelihood scores calculated before the contingency is thrown out where a single PMU is placed on the system. In this case, the correct contingency was evaluated first. The number eigenvector fittings is listed above each contingency, with 53 being the maximum. One may see that the computational savings incurred are significant

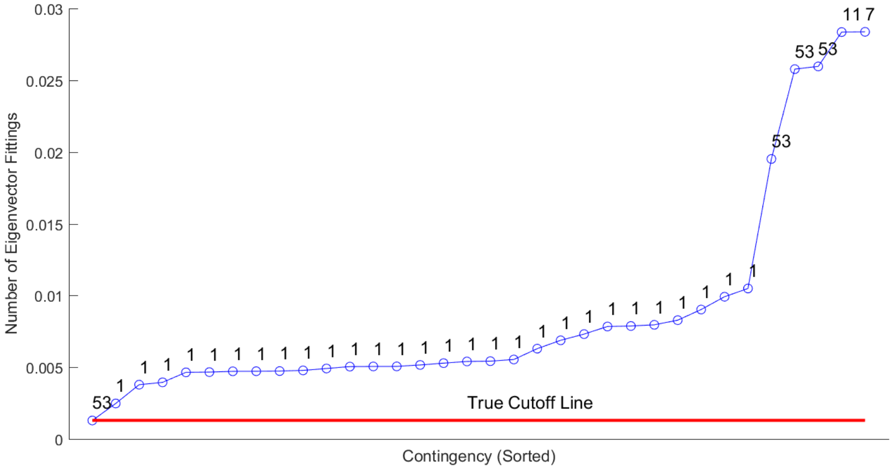

Figure 6: A different situation, in which the correct contingency is evaluated later, but filtering occurs much faster afterwards. Indeed, in this case, it only takes one eigenvector fitting per contingency to properly filter. Note the higher number of fits in the tail end due to a worse contingency evaluation order from the LS fit

We make sure to fit the eigenvectors in an order based on the size of the their respective weights —with heavier weighted eigenvectors fit first. This increases the growth of likelihood scores and thus increases the speed at which contingencies may be discounted.

7 Concerns and Drawbacks

7.1 Numerical Stability

Fitting damped sinusoids to a possible noisy signal is considered an ill-conditioned problem. While this is a topic slightly outside the scope of this report, this concern should still be mentioned.

While N4SID does an admirable job with a non-noisy signal, an important question is how it performs with noise. N4SID does fit a noise parameter, but it makes some assumptions that aren't necessarily true e.g that the noise will be distributed normally. Additionally, adding a minimal amount of norally distributed noise generated a some "spurious" eigenpairs not seen in the un-noisy fitting. Most of the time, these eigenpairs either had associated weight (norm of the fitted eigenvector) small or dampening/frequency outside the window we were looking at.

TO ADD: PERFORMANCE FIGURE

7.2 Model Error and Signal Error

Following from the previous section, how our method will perform with different noise patterns is a bit unclear. We should be clear to separate model error and signal error, as they are fundamentally different. We also note that predicting, modeling, and dealing with both model and signal error noise in power systems are research topics in their own right (for example, see [4], [10] and [1] respectively for examples). Both these sources of error will muddle the accuracy of our contingency identification method.

Model error is derived from an innaccurate picture of the physical system itself. One may be tempted to express as white noise —however, this is simply not true. For example, one large source of modeling error comes from assuming constant loads along buses; the power systems community has long known loads are correlated with not only time of day but also time of year. Approximating loads as roughly piecewise constant is an easy solution, but has the effect of increasing the number of contingencies one might have to store. Modeling loads as dynamic variables is better, but one incurs model error regardless.

Signal error is derived from problems with our sensors on the network. A more catastrophic source of error could be an out-of-sync PMU giving a delayed signal. More subtle is basic signal noise from engineering imperfections in the sensor itself.

There are surely more mathematically concise ways of integrating modeling error into our methodology, but we have not investigated further.

7.3 N-K Contingency Models

In the power grids community, an N-1 contingency denotes a single simultaneous failure. We only considered N-1 contingencies, as our method requires one to enumerate every possible model, before picking the most likely candidate. This scheme is not necessarily the most efficient if one wants to consider N-K contingencies, which denote K simultaneous failures, in which one would have to store combinatorially many contingency models in the dictionary.

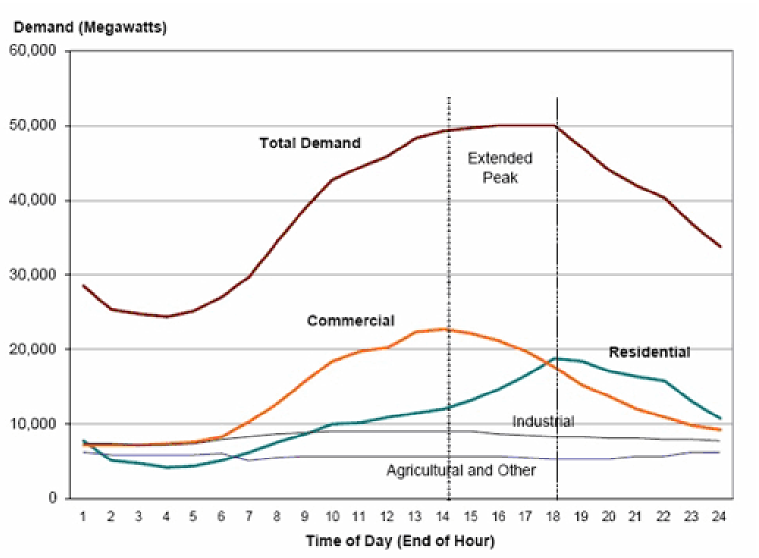

Figure 7: A graph of loads over time of day, separated by market. Image taken from Electropedia

7.4 Scalability

Our method in of itself is efficient and is able to take advantage of sparsity. We have some concerns though, mainly about the scalability of the N4SID routine, which generates large, dense hankel matrices and can get very slow, very quickly, depending both on the size of our signal and the order of the model we are attempting to fit. Indeed, while profiling our code, the majority of computation time was dedicated towards N4SID (with filtering included, otherwise the eigenpair fittings would constitute the large majority). Of course, Matlab's N4SID routine might not be the most optimized and supports a number of features we don't require, thus decreasing overall code performance.

8 TODO

- Improve LS Fit to get a better ordering, as sometimes the ordering not that great.
- Work with more noise, perhaps not normally distributed
- Improve Notation when detailing the numerical methods (notation is hard).
- Extend Power Systems description in Section 2 (working with Robert)

Appendices

Least-Squares Fit Finding an approximate ordering via least-squares is quite straightforward. For each contingency Ω, we have its Jacobian Matrix $\mathcal{J}$. We also know that our signal $f(t)$ should rougly have the form

$$f(t) = \sum_{i=1}^{k} c_i e^{\lambda_i t} H v_i$$

Naively, we may first calculate the non-infinite eigenpairs $\langle \lambda_i, v_i \rangle$ of the generalized eigenvalue problem $\mathcal{J} v_i = \lambda_i E v_i$. We then isolate the subset of the eigenvectors isolated by the PMUs by multiplying by H so obtain the partial eigenpairs $\langle \lambda_i, H v_i \rangle$. For speed, we also throw out eigenpairs with associated frequencies outside the Shannon-Nyquist bound and associated dampening factors decaying to machine precision within the first second of the signal. Assuming that our signal is sampled at times $t_1, t_2, \ldots t_n$, We then fit as follows:

$$\begin{pmatrix} e^{\lambda_1 t_1} H v_1 & e^{\lambda_2 t_1} H v_2 & \ldots & e^{\lambda_k t_1} H v_k \\ e^{\lambda_1 t_2} H v_1 & e^{\lambda_2 t_2} H v_2 & \ldots & e^{\lambda_k t_2} H v_k \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ e^{\lambda_1 t_n} H v_1 & e^{\lambda_2 t_n} H v_2 & \ldots & e^{\lambda_k t_n} H v_k \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{pmatrix} = \begin{pmatrix} f(t_1) \\ f(t_2) \\ \vdots \\ \vdots \\ f(t_n) \end{pmatrix} \tag{19}$$

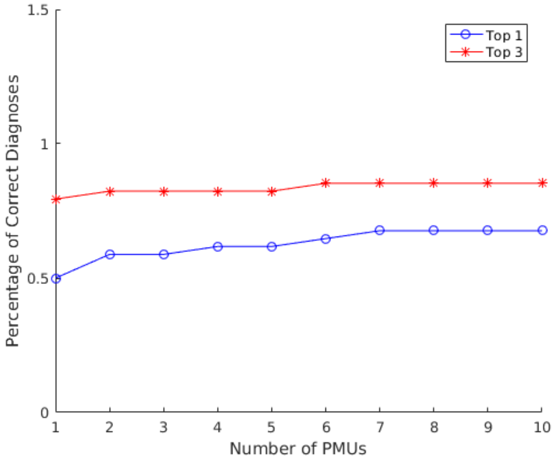

Figure 8: The complete results from our LS fit routine with a varying number of PMUs, randomly placed along the network. Note that while some contingencies are indeed completely diagnosed by the LS fit routine by itself, it lacks the accuracy necessary in other cases. However, it does a very good job approximating an evaluation order.

References

[1] Jian Chen and Ali Abur. Placement of pmus to enable bad data detection in state estimation. *IEEE Transactions on Power Systems*, 21(4):1608–1615, 2006.

[2] Roozbeh Emami and Ali Abur. External system line outage identification using phasor measurement units. *IEEE Transactions on Power Systems*, 28(2):1035–1040, 2013.

[3] Peter Kunkel. *Differential-algebraic equations: analysis and numerical solution.* European Mathematical Society, 2006.

[4] H Meng, Y Ll Guan, and S Chen. Modeling and analysis of noise effects on broadband power-line communications. *IEEE Transactions on Power delivery*, 20(2):630–637, 2005.

[5] Lawrence Perko. *Differential equations and dynamical systems*, volume 7. Springer Science & Business Media, 2013.

[6] C. Ponce and D. Bindel. Flier: Practical topology update detection using sparse pmus. *IEEE Transactions on Power Systems*, 2016.

[7] Katherine M Rogers, Rebecca D Spadoni, and Thomas J Overbye. Identification of power system topology from synchrophasor data. In *Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES*, pages 1–8. IEEE, 2011.

[8] Joseph Euzebe Tate and Thomas J Overbye. Line outage detection using phasor angle measurements. *IEEE Transactions on Power Systems*, 23(4):1644–1652, 2008.

[9] Joseph Euzebe Tate and Thomas J Overbye. Double line outage detection using phasor angle measurements. In *Power & Energy Society General Meeting, 2009. PES'09. IEEE*, pages 1–5. IEEE, 2009.

[10] Manfred Zimmermann and Klaus Dostert. Analysis and modeling of impulsive noise in broad-band powerline communications. *IEEE transactions on Electromagnetic compatibility*, 44(1):249–258, 2002.