

Convex Optimization Project Proposal

Eric Lee

1 Abstract

Contingency Identification in the context of Smart Grids is the act of diagnosing contingencies, which are failures such as a downed line or failed generator, in a Power System using time-domain sensor data. We can reformulate this problem as an optimization problem in which, given a *Partial Eigenpair* $\langle \mu_i, H v_i \rangle$ of some matrix \mathcal{J} , one attempts to fit the full eigenpair $\langle \mu_i, v_i \rangle$

2 Overview

We are given a linear differential algebraic equation (DAE) completely characterizing a Power System.

$$Ez' = \mathcal{J}z \tag{1}$$

Then the solution to (1) has the form

$$z(t) = \sum_{i=1}^k c_i e^{\mu_i t} v_i = \sum_{i=1}^k e^{\mu_i t} d_i \tag{2}$$

Where $\langle \mu_i, v_i \rangle$ are the non-infinite *eigenpairs* of the generalized eigenvalue problem $\mathcal{J}v_i = \mu_i E v_i$, 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

$$f(t) = H z(t) = \sum_{i=1}^k e^{\mu_i t} H d_i \tag{3}$$

Note that H simply picks out a subset of $z(t)$, since we have pushed H onto the eigenvectors. So to be more precise, we get the *Partial Eigenpair* $\langle \mu_i, H v_i \rangle$ from the sensors where $H v_i \subset v_i$.

3 Problem Set-Up

Given a dictionary of contingency matrices, which is the set of possible Power System models we are considering

$$\mathcal{D} = \{J_1, J_2, \dots, J_n\}$$

And a set of sensor readings

$$\mathcal{P} = \{\langle \lambda_1, x_1 \rangle, \langle \lambda_1, x_2 \rangle, \dots, \langle \lambda_n, x_n \rangle\}$$

We want to see which $J \in \mathcal{D}$ is most likely responsible for generating \mathcal{P} . More formally, we must lay down some notion of “distance” $T : \mathcal{D} \times \mathcal{P} \rightarrow \mathbb{R}$. The $J \in \mathcal{D}$ with the smallest distance to each element in \mathcal{P} is the contingency we diagnose as the most likely.

3.1 $M(\lambda, x, J)$

We previously defined this notion of distance as

$$M(\lambda, x, J) = \min_{\alpha, y} \left\| \begin{pmatrix} E\lambda - J \\ y \end{pmatrix} \begin{pmatrix} \alpha x \\ y \end{pmatrix} \right\|_2^2 \quad (4)$$

subject to $\alpha^2 + \|y\|_2^2 = 1$

With $\langle \lambda, x \rangle \in \mathcal{P}$ and $J \in \mathcal{D}$. Intuitively, we have our eigenvector subset x , and are attempting to fit the rest of the eigenvector y subject to a norm constraint on the fitting. This is a straightforward matrix norm calculation and led to excellent results initially.

However, our results were incredibly poor with even minimal amounts of noise. We have simple hypothesis for why; Let’s say that

$$\bar{\alpha}, \bar{y} = \arg \min_{\alpha, y} M(\lambda, x, J)$$

i.e. $\bar{\alpha}$ and \bar{y} are solutions to the non-noisy problem. Then we can find a simple upper bound by plugging $\bar{\alpha}$ and \bar{y} into the noisy optimization problem

$$M(\lambda + \delta, x + \epsilon, J) = \min_{\alpha, y} \left\| \begin{pmatrix} E(\lambda + \delta) - J \\ y \end{pmatrix} \begin{pmatrix} \alpha(x + \epsilon) \\ y \end{pmatrix} \right\|_2^2 \quad (5)$$

subject to $\alpha^2 + \|y\|_2^2 = 1$

$$\leq M(\lambda, x, J) + \left\| \delta E \begin{pmatrix} \bar{\alpha}(x + \epsilon) \\ \bar{y} \end{pmatrix} + (E\lambda - J) \begin{pmatrix} \bar{\alpha}\epsilon \\ 0 \end{pmatrix} \right\|_2^2$$

$$\leq M(\lambda, x, J) + \epsilon \lambda_{max}(E\lambda - J)$$

where $\lambda_{max}(A)$ is the largest eigenvector in absolute value of A . Note that this is not necessarily the ideal upper bound; the λ_{max} term gives us some problems.

4 Proposal Details

We have laid out a fitting methods $M(\lambda, x, J)$, which is highly accurate without noise but suffers a massive loss in accuracy when faced with noise. We want to make $M(\lambda, x, J)$ more robust. We plan on doing so in two ways.

1. The first is via standard robust optimization; we add an error term to be fit in the objective function. So for example, we introduce the robust distance

$$M_r(\lambda, x, J) = \min_{\alpha, y, \alpha, \epsilon} \left\| \begin{pmatrix} E\lambda - J \end{pmatrix} \begin{pmatrix} \alpha_r x \\ y_r \end{pmatrix} \right\|_2^2 \quad (6)$$

subject to $\alpha^2 + \|y\|_2^2 = 1$

$\alpha_r = \alpha + \delta$

$y_r = y + \epsilon$

2. The second is via changing the loss function; the L2 loss function is easy to work with, but perhaps not optimal. We plan on looking into the Huber loss function, as we expect our readings to have a number of outliers due to sensing error which we have not taken into account.

For both items, we hope to build up some theoretical guarantees as well as some implementation, and ultimately improve the results of our research.