Georgia Tech College of Engineering
School of Electrical and Computer Engineering

ECE 4320: State Estimation

Daniel K. Molzahn, Samuel Talkington

March 26, 2025

Logistics

Agenda: next 2 weeks

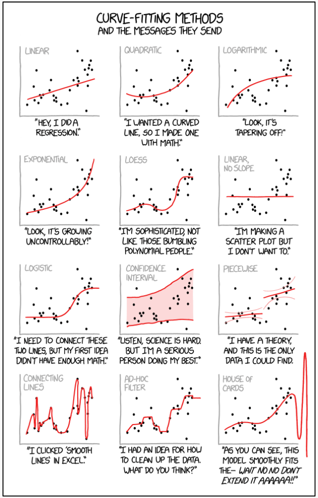
- State estimation (x2)

Office hours w/ me

- Project, coding, support
- Research interest chats

This week: Thurs. 2pm;

Next week: TBD.



Source: xkcd

Recap

Last time:

- We introduced **unit commitment**–an extension of DC Optimal Power Flow (OPF) with **discrete** decision.
- We talked about how to solve Unit Commitment with the "branch-and-bound" tree.

Today:

- We'll talk about *state estimation*–bringing the world of statistics to power systems.
- Time permitting, we'll also talk about how to solve these problems computationally.

Overview

Introduction to State Estimation

Mathematical Formulation

Solution Methods

Linear approximation for state estimation and simple example

Introduction to State Estimation

What is State Estimation?

- Process of estimating the state variables (voltage phasors) of a power system using measurements.
- Essential for real-time monitoring, control, and operation of power systems.
- Objective: Obtain the best estimate of the state of the system given the available measurements.

Mathematical Formulation

System Model

- **State vector**: $\boldsymbol{x} \in \mathbb{R}^n$ representing voltage angles and magnitudes:

$$\boldsymbol{x} = \begin{bmatrix} \overbrace{\theta_2 \quad \theta_3}^{\text{Phase angles}} & \dots & \overbrace{v_1 \quad v_2}^{\text{magnitudes}} & \dots \end{bmatrix}^{\mathsf{T}}$$

- **Measurement function**: $\boldsymbol{h}(\boldsymbol{x})$, a *nonlinear* mapping from states to measurement, consisting of power flows, injections, or bus voltages.

Idea: Estimate x from noisy observations of $h(x)$:

- Measurement vector $\mathbf{z} \in \mathbb{R}^m$:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \underline{\epsilon}$$

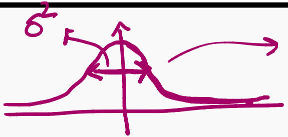
voltage phasors

- ϵ: Measurement noise. We'll model this as a standard Gaussian vector:

$$\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}).$$

additive noise model

$$\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_m \end{bmatrix}$$



normal distribution pdf.

measurements

$x \in \mathbb{R}^n$

this is not the only noise model

$$\underbrace{\mathbf{z} = \begin{bmatrix} z_1 \\ \vdots \\ z_m \end{bmatrix}}_{\text{this what we see}} = \begin{bmatrix} h_1(\mathbf{x}) \\ \vdots \\ h_m(\mathbf{x}) \end{bmatrix} + \underbrace{\begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_m \end{bmatrix}}_{\substack{\text{Corruption} \\ \text{random noise}}}$$

Solution Methods

measurement functions

*LASSO

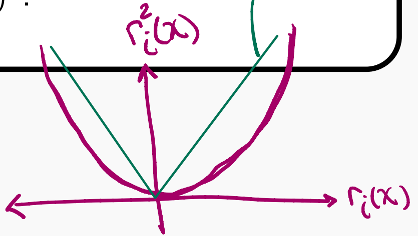
Least squares estimation

- Define the *residuals* (estimation error) as: $\underbrace{z_i - h_i(\boldsymbol{x})}_{r_i(x)}$. $i=1,\dots,m$
- *Least squares* seeks to minimize the sum of the *squared residuals*:

$$\min_{\boldsymbol{x}} \sum_{i=1}^{m} (z_i - h_i(\boldsymbol{x}))^2 \,.$$

$|r_i(x)|$

Q: Why minimize the squares?



Motivation for Weighted Least Squares (WLS) Estimation

- What if we **trust some measurements more** than others?
- **Example:** What if the measurement noise variance is different at each bus?

$$\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}),$$

$\sigma_i^2 \neq \sigma_j^2$

where $\boldsymbol{R}$ is the *covariance martrix*:

$$\boldsymbol{R} = \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n^2 \end{bmatrix}.$$

- The measurements are still: $\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \epsilon$.

Weighted Least Squares

Gauss-Markov Theorem

Idea: Weighted least squares

Minimize the **weighted** sum of the squared residuals:

$$\min_{\boldsymbol{x}} \sum_{i=1}^{m} w_i \, (z_i - h(\boldsymbol{x}))$$

$w_i = \frac{1}{\sigma_i^2}$ for each $i = 1, \ldots, m$

Pair and share:

What should we set the weights w_i to be?

$\hat{x}$ → Our state estimate is the solution to this optimization problem.

$$\min_{x} \underbrace{\sum_{i=1}^{m} \frac{(z_i - h(x))^2}{\sigma_i^2}}_{=C(x)} = \min_{x} \underbrace{(\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}))^{\mathsf{T}} \boldsymbol{R}^{-1} (\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}))}_{\text{Quadratic form}}$$

Cost function

Solving WLS

- **How to solve?**: Iterative methods (e.g., Gauss-Newton, Newton-Raphson).
- Convergence depends on good initial estimates.
- Solve by setting the derivatives of $C(x) = 0$:

gradient vector

$$\nabla_x C = g(x) = \begin{bmatrix} \frac{\partial C}{\partial x_1}(x) \\ \vdots \\ \frac{\partial C}{\partial x_n}(x) \end{bmatrix}$$

contains all of the derivatives of our cost w.r.t. the states $x_1, \ldots, x_n$

Derivation of the Newton-Raphson solution

$$c(x) = \sum_{i=1}^{n} \frac{(z_i - h(x))^2}{\sigma_i^2}$$

$$\frac{\partial c}{\partial x_1} = \sum_{i=1}^{n} 2 \frac{(z_i - h(x))}{\sigma_i^2} \left(-\frac{\partial h}{\partial x_1}(x) \right) = 0$$

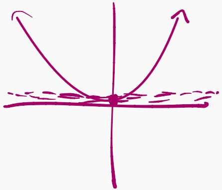
$$\vdots$$

$$\frac{\partial c}{\partial x_n} = \sum_{i=1}^{n} 2 \frac{(z_i - h(x))}{\sigma_i^2} \left(-\frac{\partial h}{\partial x_n}(x) \right) = 0$$

$$\Rightarrow \boxed{g(x) = -H(x)^T R^{-1} (z - h(x))}$$

Gain matrix

$$\boxed{g(x) = -H(x)^T R^{-1}(z - h(x))}$$



$$H(x) = \begin{bmatrix} \frac{\partial h_1}{\partial x_1}(x) & \cdots & \frac{\partial h_1}{\partial x_n}(x) \\ \vdots & & \\ \frac{\partial h_m}{\partial x_1}(x) & \cdots & \frac{\partial h_m}{\partial x_n}(x) \end{bmatrix}$$

"Measurement Jacobian"

Gain matrix $\rightarrow$ Jacobian of the gradient w.r.t. x

$$G(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}(x) & \cdots & \frac{\partial g_1}{\partial x_n}(x) \\ \vdots & & \\ \frac{\partial g_n}{\partial x_1}(x) & \cdots & \frac{\partial g_n}{\partial x_n}(x) \end{bmatrix}$$

$f: \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad \frac{\partial f}{\partial x} \in \mathbb{R}^{m \times n}$

$\rightarrow$ apply Newton raphson to find the state estimate $\hat{x}$ s.t. $g(\hat{x}) \approx \underline{0}$

Alg: Given $\hat{x}_0$,

while not converged

$$\hat{x}^{(k+1)} = \hat{x}^{(k)} - G(\hat{x}^{(k)})^{-1} g(\hat{x}^{(k)})$$

Gain matrix

<u>note:</u>

$$G(\hat{x}^{(k)}) \approx H(\hat{x}^{(\cancel{k})})^T R^{-1} H(\hat{x}^{(\cancel{k})})$$

Approximate $H(\hat{x}^{(k)})$ as nearly constant

$$H(x+\Delta x) \approx H(x)$$

Linear approximation for state estimation and simple example

Linear State Estimation (DC Model)

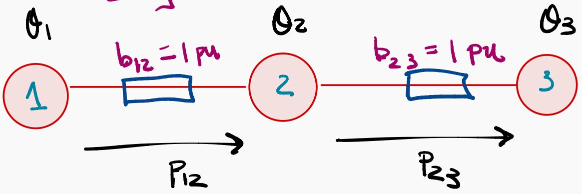
- Simplification of the AC model by linearizing the measurement model.
- The measurement function becomes linear:

$$\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x} + \epsilon$$

3-bus system setup → Use the DC power flow approx.

$P \approx B\theta$

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} \approx B \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \rightarrow P = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$



Q:

Given (noisy) measurements of P_{12}, P_{23}, θ_2

→ What are the "true" phase angles $\theta_1, \theta_2, \theta_3$

→ Still want (denoised) θ_2

State:

$$x \triangleq \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

3-bus system setup

$$z = \begin{bmatrix} P_{12} \\ P_{23} \\ \theta_2 \end{bmatrix} = H x + \varepsilon$$

$$= \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} + \underset{\xi}{\varepsilon}$$

3-bus system setup

Next time

- Observability analysis
- Pseudo-measurements
- New extensions of state estimation