Fast Graphical Learning Method for Parameter Estimation in Large-Scale Distribution Networks

Prof. Nanpeng Yu
Department of Electrical and Computer Engineering
Department of Computer Science (Cooperating Faculty)
Director of Energy, Economics, and Environment Research Center
nyu@ece.ucr.edu
University of California, Riverside

Outline

- Background
- Literature Review and Motivation
- Brief Review of Physics-informed GL Model for Parameter Estimation
- Accelerate Physics-informed GL Model
  - Fast Forward Algorithm
  - Fast Backward Algorithm
- Numerical Study
- Conclusion

Background

- Accurate estimations of distribution network parameters are essential for modeling, monitoring, and control in power distribution systems.
- Distribution network topology and parameters in GIS often contain errors.
- Objective of network parameter estimation
  - Infer the series impedances of distribution lines based on network topology information and data from smart meters, SCADA system and/or micro-PMUs.
- More challenging to estimate the line parameters of distribution network than that of transmission network
  - Single-phase line models of transmission system are insufficient for distribution system.
  - Unbalanced nature of distribution system requires estimation of the elements of the 3×3 phase impedance matrix
  - The number of line parameters of distribution system is much larger than that of transmission system.

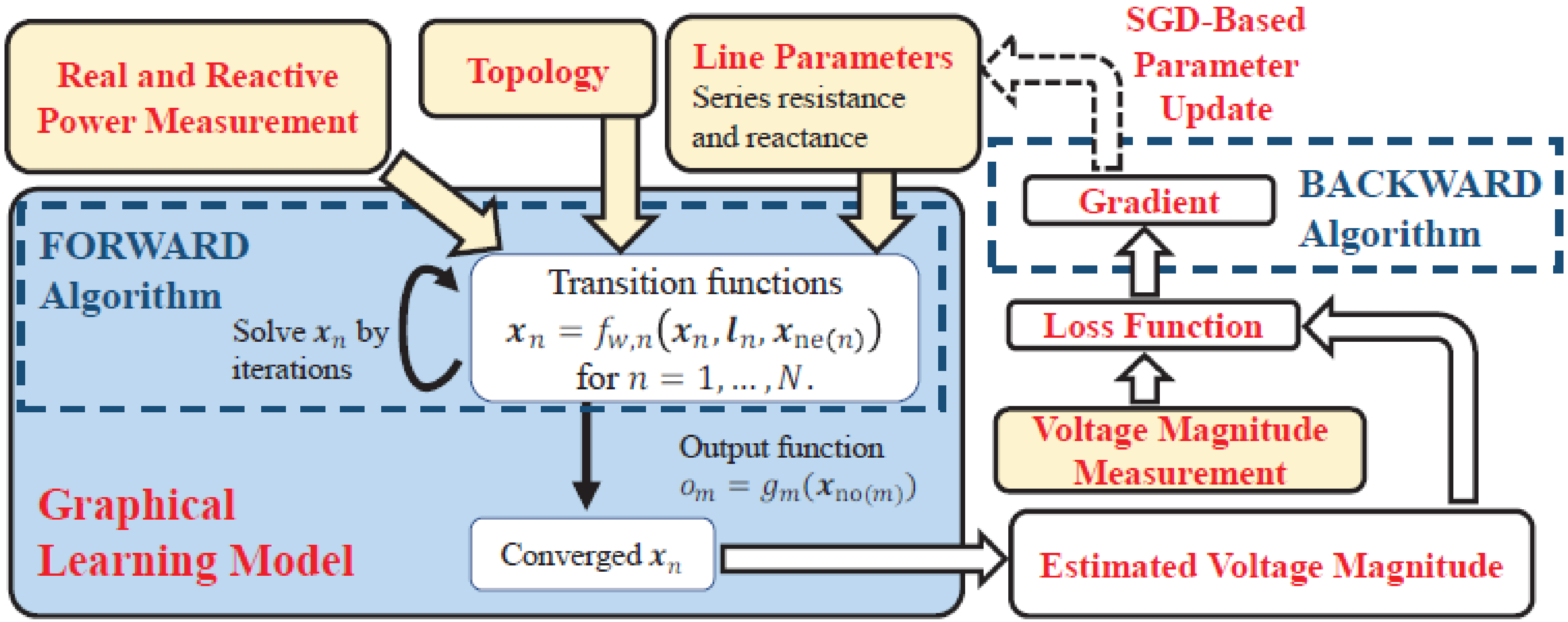
Literature Review and Motivation

- Group 1: SCADA system data are used to estimate transmission network parameters of a single-phase model.
  - Detect and correct parameter errors. Designed for single-phase line models.
    - [Zarco 2000] [Logic 2006] [Castillo 2010] [Lin 2016] [Zhao 2018] [Li 2017]
- Group 2: PMU data are used to estimate line parameters of both transmission and distribution systems.
  - Highly accurate estimation. Expensive and require widespread installations of PMUs.
    - [Ardakanian 2019] [Asprou 2015] [Kumar 2016] [Khandeparkar 2016] [Gajare 2017] [Ren 2017]
- Group 3: Smart meter data are used to estimate distribution line parameters.
  - Single-phase/balanced 3-phase lines. Do not work with delta-connected secondary.
    - [Cunha 2020] [Han 2015] [Lave 2019] [Zhang 2020] [Peppanen 2016] [Wang 2020]
  - Unbalanced 3-phase lines with physics-informed graphical learning (GL). Low computation efficiency
    - [Wang 2021]
- **Motivation**: There is a need to accelerate the computation of physics-informed GL model

Physics-Informed Graphical Learning Model

- Key Idea: **Embed physical equations of power flow in the graphical learning model**
- Inspired by graphical neural network (GNN)
- Difference between physics-informed GL and GNN
  - Leverage three-phase power flow-based physical transition functions to replace the deep neural networks in GNN.
- Key Step: Derive the gradient of voltage magnitude loss function w.r.t. line segment's resistance and reactance parameters with an iterative method.
- Estimate distribution line parameters with SGD considering prior estimates of line parameters and physical constraints.
- Improve computation efficiency with grid partition scheme.

Overall Framework of Graphical Learning Model

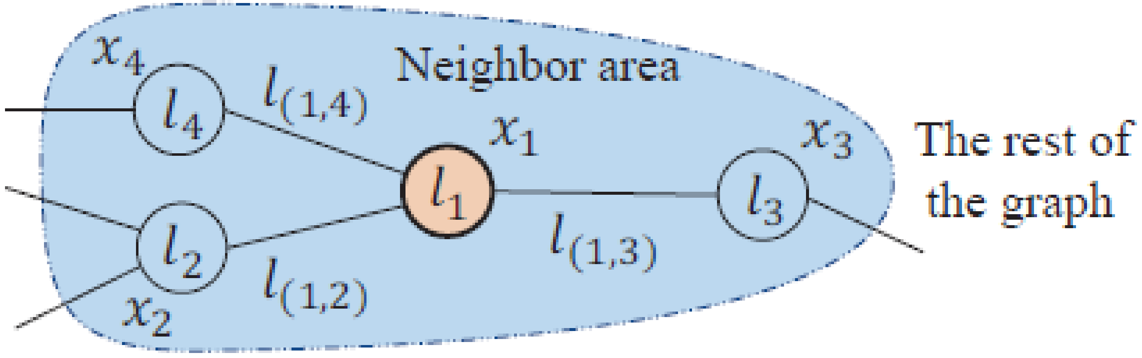


- Inputs to GL engine
- Each node of the GL model corresponds to a physical bus.
- Nodal state: three-phase complex voltage
- Output of GL model
- Loss function
- SGD-based parameter updates

- FORWARD and BACKWARD algorithms are time consuming to execute
- FORWARE function solve the state variables given the line parameters
- BACKWARD function calculates the gradients with respect to line parameters

Review of Graphical Neural Network

- GNN uses a graph's topological relationships between nodes to incorporate the underlying graph-structured information in data.
- Local transition function $\boldsymbol{x}_n = f_{\boldsymbol{\omega},n}\left(\boldsymbol{l}_n, \boldsymbol{l}_{co(n)}, \boldsymbol{x}_{ne(n)}, \boldsymbol{l}_{ne(n)}\right)$
- Output function $o_n = g_{\boldsymbol{\omega},n}(\boldsymbol{x}_n, \boldsymbol{l}_n)$
- Unique solution of the state can be found with $[\boldsymbol{x}]^{\tau+1} = F_{\omega}([\boldsymbol{x}]^{\tau}, [\boldsymbol{l}])$.
  - Sufficient condition for Banach fixed point theorem.
- Parameters ω of global transition and output functions F_{ω} and G_{ω} are updated to minimize a quadratic loss function
- $loss = \sum_{m=1}^{M}(o_m - \breve{o}_m)^2$



Original FORWARD Algorithm

- The original forward algorithm uses transition function to iteratively update the state (complex voltage) of the GL model
- Iterative power flow equation

$$u_n = Y_{nn}^{-1}\left((s_n^* \oslash u_n^*) + \sum_{k\in ne(n)} Y_{nk}u_k\right)$$

- Physics-informed transition function

$$\begin{bmatrix} Re(u_n) \\ Im(u_n) \end{bmatrix} = \langle Z_{nn}\rangle\left(\begin{bmatrix} Re(s_n^* \oslash u_n^*) \\ Im(s_n^* \oslash u_n^*) \end{bmatrix} + \sum_{k\in ne(n)} \langle Y_{nk}\rangle \begin{bmatrix} Re(u_n) \\ Im(u_n) \end{bmatrix}\right)$$

- State vector $x_n \triangleq \begin{bmatrix} Re(u_n) \\ Im(u_n) \end{bmatrix}$
- feature vector $l_n \triangleq \begin{bmatrix} Re(s_n) \\ Im(s_n) \end{bmatrix}$
- Convergence of state vector estimates to the fix point is slow

Algorithm 1 FORWARD(w, t)

Input: Current line parameter w and the time instance t.

Output: Theoretical $[x(t)]$ of the distribution system with line parameter w.

1: Initialize the source nodes' state $x_0(t)$ with the known measurement at the source node. Initialize the other nodes' state $x_n(t)$ as defined in (10) with balanced flat node voltage, i.e. $u_n(t) = [1, e^{-j\frac{2\pi}{3}}, e^{j\frac{2\pi}{3}}]^T$, $(n = 1, ..., N)$.

2: Construct the initial $[x(t)]^0$ by stacking all the initial $x_n(t)$, $(n = 0, ..., N)$. Construct function F_w with w.

3: **repeat**

4: $[x(t)]^{\tau+1} = F_w([x(t)]^\tau, [l(t)])$ and fix $x_0(t)$ to its initial value.

5: $\tau = \tau + 1$

6: **until** $\|[x(t)]^\tau - [x(t)]^{\tau-1}\|^2 < \epsilon_{\text{forward}} \cdot \|[x(t)]^{\tau-1}\|^2$

7: **return** $[x(t)] = [x(t)]^\tau$.

Original BACKWARD Algorithm

- The BACKWARD algorithm calculates the gradient of the loss function of first difference voltage time series w.r.t. line parameters $\boldsymbol{\omega}$.
- The iterative FORWARD function can be represented as a recurrent neural network.
  - $e_{\boldsymbol{\omega}}(\mathfrak{T})$'s gradient is difficult to calculate in the conventional way.
  - Solution: design BACKWARD function following the backpropagation principle of Almeida-Pineda algorithm for RNN.
  - The gradient can be iteratively calculated using an intermediate variable $z(t)$.
- The convergence of intermediate variable to the fixed point is slow.

Algorithm 2 BACKWARD(w, $\mathfrak{T}$)

Input: Current line parameter w and the first difference instance batch index $\mathfrak{T}$.

Output: Gradient $\frac{\partial e_{\boldsymbol{w}}(\mathfrak{T})}{\partial \boldsymbol{w}}$.

$[x(t)]$=FORWARD(w, t), $t \in \mathfrak{T} \cup \tilde{\mathfrak{T}}$.

Construct $[\hat{x}(t)]$ as (20), $t \in \mathfrak{T}$.

Calculate $[\tilde{o}(t)] = \hat{G}([\hat{x}(t)])$, $\hat{A}(t) = \frac{\partial \hat{F}_{\boldsymbol{w}}([\hat{\boldsymbol{x}}(t)],[\hat{\boldsymbol{l}}(t)])}{\partial [\hat{\boldsymbol{x}}(t)]}$, $\hat{b}(t) = \frac{\partial e_{\boldsymbol{w}}(t)}{\partial [\tilde{o}(t)]} \cdot \frac{\partial \hat{G}([\hat{\boldsymbol{x}}(t)])}{\partial [\hat{\boldsymbol{x}}(t)]}$, for $t \in \mathfrak{T}$.

for $t \in \mathfrak{T}$ **do**

  Initialize $z(t)^0 = \mathbb{0}_{1\times 12N}$, $\tau = 0$.

  repeat

    $z(t)^{\tau+1} = z(t)^{\tau} \cdot \hat{A}(t) + \hat{b}(t)$

    $\tau = \tau + 1$

  until $\|z(t)^{\tau} - z(t)^{\tau-1}\|^2 < \epsilon_{\text{backward}} \cdot \|z(t)^{\tau-1}\|^2$

  $\frac{\partial e_{\boldsymbol{w}}(t)}{\partial \boldsymbol{w}} = z(t)^{\tau} \cdot \frac{\partial \hat{F}_{\boldsymbol{w}}([\hat{\boldsymbol{x}}(t)],[\hat{\boldsymbol{l}}(t)])}{\partial \boldsymbol{w}}$, for $t \in \mathfrak{T}$.

end for

$\frac{\partial e_{\boldsymbol{w}}(\mathfrak{T})}{\partial \boldsymbol{w}} = \frac{1}{|\mathfrak{T}|} \sum_{t \in \mathfrak{T}} \frac{\partial e_{\boldsymbol{w}}(t)}{\partial \boldsymbol{w}}$

return $\frac{\partial e_{\boldsymbol{w}}(\mathfrak{T})}{\partial \boldsymbol{w}}$

Fast FORWARD Algorithm

- In the physics-informed graphical learning model, the FORWARD function is only used to compute the state $[\boldsymbol{x}]$, given the parameter set $\boldsymbol{\omega}$.
- Key idea: re-design the FORWARD algorithm without the transition function, as long as it can solve the state $[\boldsymbol{x}]$ given $\boldsymbol{\omega}$.
- Accelerate FORWARD algorithm with two methods
  - Derive a nearly-accurate initial estimate of states using linearized three-phase power flow model
  - Leverage current injection method to accelerate convergence

$$\begin{bmatrix} \check{v} \\ \check{\theta} \end{bmatrix} = \check{A}^{-1} \begin{bmatrix} p \\ \check{q} \end{bmatrix}$$

The left hand side is the deviation of non-substation nodes from the substations.

$\check{\boldsymbol{A}}$ is 6N × 6N matrix derived from topology and line parameters $\boldsymbol{\omega}$.

$$[\Delta I] = J[\Delta V]$$

$[\Delta I]$ is a vector of three-phase real and imaginary parts of nodal current mismatch

J is the Jacobian matrix

The three-phase nodal voltage update $[\Delta V]$ is solved iteratively with $[\Delta I]$ and J, which are calculated from the current state.

Algorithm 1 Fast-FORWARD(w, t)

Input: Parameter w and time index t.

Output: Distribution system state $[x(t)]$ when line parameter is w.

1: Use (9) to calculate $\check{v}$ and $\check{\theta}$. Combine $\check{v}$, $\check{\theta}$, and the source nodes' state $x_0(t)$ to initialize the state $[x(t)]$.

2: **repeat**

3: Update $[\Delta I]$ and J based on current state $[x(t)]$. Solve $[\Delta I]$ from (10) and update $[x(t)]$.

4: **until** The maximum absolute value in $[\Delta I]$ is less than ϵ_{CIM}

5: **return** $[x(t)]$.

Fast BACKWARD Algorithm

- Accelerate BACKWARD algorithm
  - Improve initialization of intermediate variable $z(t)^0$ instead of using $\mathbf{0}_{1\times 12N}$
  - $z(t)^0$ can be found by solving the update equation $z(t) = z(t) \cdot \hat{A}(t) + \hat{b}(t)$
  - If the solution is not feasible (e.g., ill-conditioned matrices), we still use $z(t)^0 = \mathbf{0}_{1\times 12N}$
- Steps 8 - 10 are used to process the converged $z(t)$ to impute the gradient of the loss function with respect to the line parameters.

Algorithm 2 Fast-BACKWARD(w, $\mathfrak{T}$)

Input: Parameter w and batch of time indices $\mathfrak{T}$.

Output: Gradient $\frac{\partial e_w(\mathfrak{T})}{\partial w}$.

1: Calculate $\hat{A}(t)$ and $\hat{b}(t)$ for $t \in \mathfrak{T}$ as in the BACKWARD function in [14].

2: **for** $t \in \mathfrak{T}$ **do**

3: Initialize $z(t)^0 = \hat{b}(t)(I - \hat{A}(t))^{-1}$. If it is not feasible, let $z(t)^0 = \mathbb{0}_{1\times 12N}$. $\tau = 0$.

4: **repeat**

5: $z(t)^{\tau+1} = z(t)^{\tau} \cdot \hat{A}(t) + \hat{b}(t)$

6: $\tau = \tau + 1$

7: **until** $\|z(t)^{\tau} - z(t)^{\tau-1}\|^2 < \epsilon_{\text{backward}} \cdot \|z(t)^{\tau-1}\|^2$

8: Calculate $\frac{\partial \hat{F}_w([\hat{x}(t)],[\hat{l}(t)])}{\partial w}$ as in [14], and $\frac{\partial e_w(t)}{\partial w} = z(t)^{\tau} \cdot \frac{\partial \hat{F}_w([\hat{x}(t)],[\hat{l}(t)])}{\partial w}$, for $t \in \mathfrak{T}$.

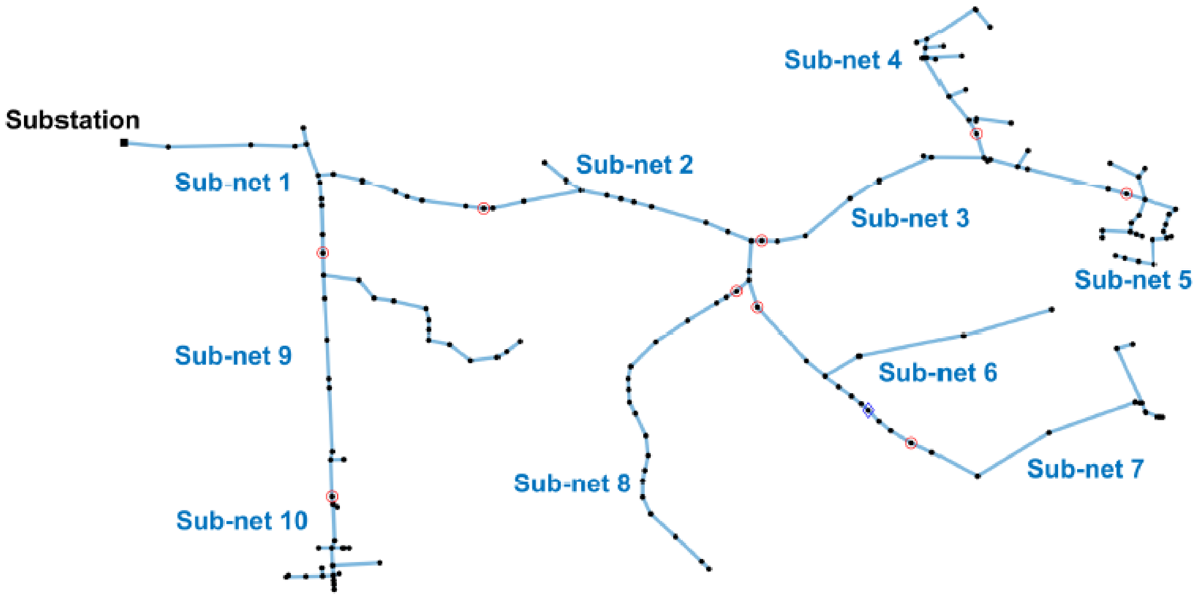
9: **end for**

10: $\frac{\partial e_w(\mathfrak{T})}{\partial w} = \frac{1}{|\mathfrak{T}|} \sum_{t\in\mathfrak{T}} \frac{\partial e_w(t)}{\partial w}$

11: **return** $\frac{\partial e_w(\mathfrak{T})}{\partial w}$

Numerical Study - Setup

- 178-bus feeder modified from a real-world 1922-bus distribution circuit of National Grid
  - Keep three-phase primary lines
  - Single-phase/two-phase loads and renewable generation are reconnected to nearest three-phase buses
  - A series of line sections without smart meters or sensors are combine into one line section



- 13.2 kV with a peak load of 4.16 MW, 3-phase capacitor 900 kVAR
- 177 line sections, 491 loads, 23 solar PV systems
- Different phase connections: AN, BN, CN, and ABC
- Feeder partitioned into 10-subnetworks (parallel estimation)
- 15-minute power measurement data from actual smart meters
- 15-day time window with 1,440 readings
- Actual temperature and irradiance data to simulate solar PV gen.
- Capacitor setting [1.0 p.u., 1.05 p.u.]
- $\omega_{initial}$ within $\pm$ 50% of the correct values.

Baseline Methods and Performance Metric

- Baseline Methods and Ablation Study
  - Linear Maximum Likelihood Estimation (based on linearized power flow)
  - Fast GL (FGL), FGL + Constraints (CON)
  - Hyperparameters for FGL
    - Batch size 10, early stopping patience 10, initial step size 1000, $\alpha = 0.3$, $\beta = 0.5$, and $\epsilon_{stop} = 0.1$
  - Computing Platform
    - Workstation with 16 CPU cores (3.0 GHz) and 192 GB RAM
- Performance Metric
  - Mean absolution deviation ratio (MADR) measures the estimation error of distribution line parameters. $\text{MADR} \triangleq \sum_{i=1}^{12\mathfrak{L}} \left|\omega_i - \omega_i^{\dagger}\right| \div \sum_{i=1}^{12\mathfrak{L}} \left|\omega_i^{\dagger}\right| \times 100\%$
  - Performance of a distribution line parameter estimation algorithm is evaluated by percentage of MADR improvement
  - $\text{MADR imporvement} \triangleq {}^{(\text{MADR}_{initial} - \text{MADR}_{final})}/_{\text{MADR}_{final}} \times 100\%$

Numerical Study – Performance Comparison

MADR Improvement of Parameter Estimation Methods in the Test Feeder (Average / Choose Optimal Value)

Network	LMLE	FGL	FGL+CON
Whole Network	10.8% / 13.5%	20.7% / 25.5%	29.4% / 30.9%
Sub-Net 1	10.3% / 9.1%	20.1% / 21.6%	23.1% / 27.1%
Sub-Net 2	7.3% / 9.2%	13.6% / 20.9%	26.7% / 29.3%
Sub-Net 3	9.9% / 12.2%	34.5% / 40.8%	41.6% / 43.7%
Sub-Net 4	4.5% / 4.7%	5.1% / 5.0%	12.4% / 13.2%
Sub-Net 5	11.6% / 14.8%	20.8% / 22.1%	21.0% / 22.3%
Sub-Net 6	12.0% / 24.3%	37.0% / 62.4%	61.8% / 63.5%
Sub-Net 7	9.3% / 9.3%	16.2% / 18.0%	31.6% / 32.9%
Sub-Net 8	13.9% / 16.5%	22.3% / 23.0%	24.9% / 25.5%
Sub-Net 9	17.3% / 21.3%	30.8% / 34.5%	37.9% / 38.2%
Sub-Net 10	7.6% / 9.6%	2.2% / 8.9%	19.0% / 20.4%

- Average MADR improvement of 20 random tests
- MDAR improvement with subnetwork's estimated parameters from the random test with lowest loss
- FGL has significantly higher MADR improvement than LMLE with an additional 9.9% to 12%.
- The CON improves the performance of FGL further with an additional 5.4% - 8.7%

Average Runtime (Seconds) of Main Functions of Parameter Estimation Methods in Sub-networks of different sizes

Method	Function	7-Bus	14-Bus	22-Bus
FGL	Fast-FORWARD	0.0142	0.0313	0.0519
	Fast-BACKWARD	0.068	0.1046	0.1761
GL	FORWARD	0.3028	1.1603	3.2839
	BACKWARD	0.1057	0.3325	1.0065
LMLE	Gradient Calculation	0.0079	0.1866	0.8531

- Time-consuming functions are the FORWARD and BACKWARD functions in GL and gradient calculation in LMLE.
- The Fast-FORWARD is over 20 times faster than FORWARD in the 7-bus feeder and over 60 times faster in the 22-bus feeder.
- The Fast-BACKWARD is over 1.5 times faster than BACKWARD in the 7-bus feeder and 6 times faster in the 22-bus feeder.
- The average running time of LMLE in a subnetwork is 245.4 mins.
- The average running time of FGL+CON is only 27.5 mins.

Conclusion

- Developed a fast graphical learning algorithm to estimate line parameters of three-phase power distribution networks.
- Wide applicability: only requires readily available smart meter data.
- Accelerate graphical learning's FORWARD algorithm by improving the initialization of state vector and adopting current injection method
- Accelerate graphical learning's BACKWARD algorithm by improve the initialization of intermediate variable
- The fast graphical learning method improves computation efficiency by as much as 60 times while attaining the accuracy of state-of-the-art algorithms on a real-world distribution feeder.

Relevant Publications

- Wenyu Wang and Nanpeng Yu, "Parameter Estimation in Three-Phase Power Distribution Networks Using Smart Meter Data," *the 16th International Conference on Probabilistic Methods Applied to Power Systems*, 2020.
- Wenyu Wang and Nanpeng Yu, "Estimate Three-phase Distribution Line Parameters with Physics-informed Graphical Learning Method," under review, http://arxiv.org/abs/2102.09023, *IEEE Transactions on Power Systems*, 2021, doi: 10.1109/TPWRS.2021.3134952.
- Wenyu Wang, Nanpeng Yu, and Yue Zhao, "Fast Graphical Learning Method for Parameter Estimation in Large-Scale Distribution Networks," under review, 2022.

Contact Information:
Nanpeng Yu, nyu@ece.ucr.edu

Thank You

UCR: Wenyu Wang

SBU: Yue Zhao

National Grid: Brent Fedele, Weixiang Zhao, Brian Yung

NYSERDA: Financially Supported by Agreement Number 159613