Iowa State University

Distribution System State Estimation and Smart Meter Analysis

Instructor: Dr. Zhaoyu Wang
GRA: Dingwei Wang
Department of Electrical and Computer Engineering
Iowa State University

Contents

The Concept of SE

What is the state in the power system

- In general, power system has normal, emergency, and restorative states.
- To monitor system states, different measurements from all parts of the system need to be utilized.
- State estimation is a data processing algorithm for converting redundant meter readings into an estimate of the state of an electric power system.

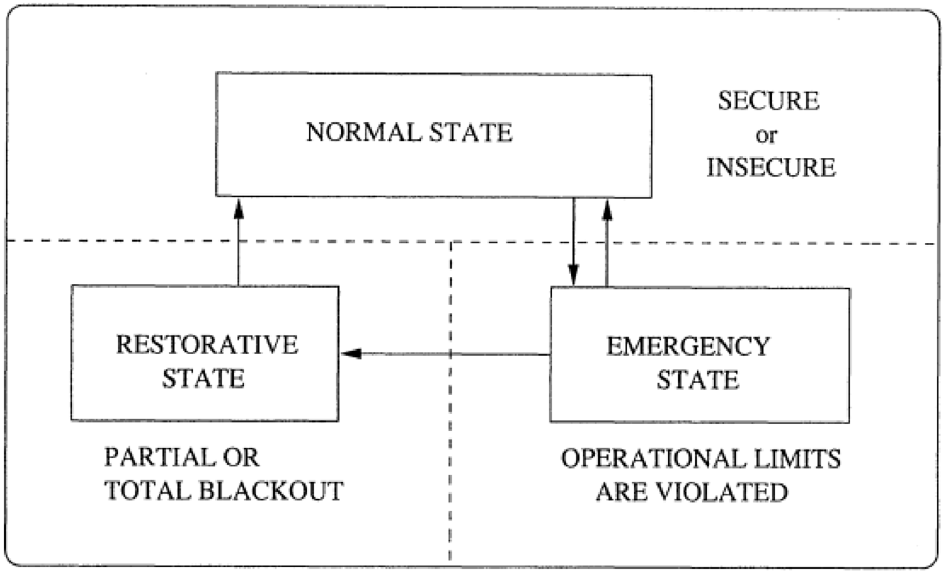


Fig. 1 State Diagram for Power System Operation [1]

[1] Gomez-Exposito A, Abur A. Power system state estimation: theory and implementation[M]. CRC press, 2004.

The Concept of SE

SE is a widely-used tool in transmission systems. In the transmission system SE, voltage magnitudes and phase angles are considered the states of systems.

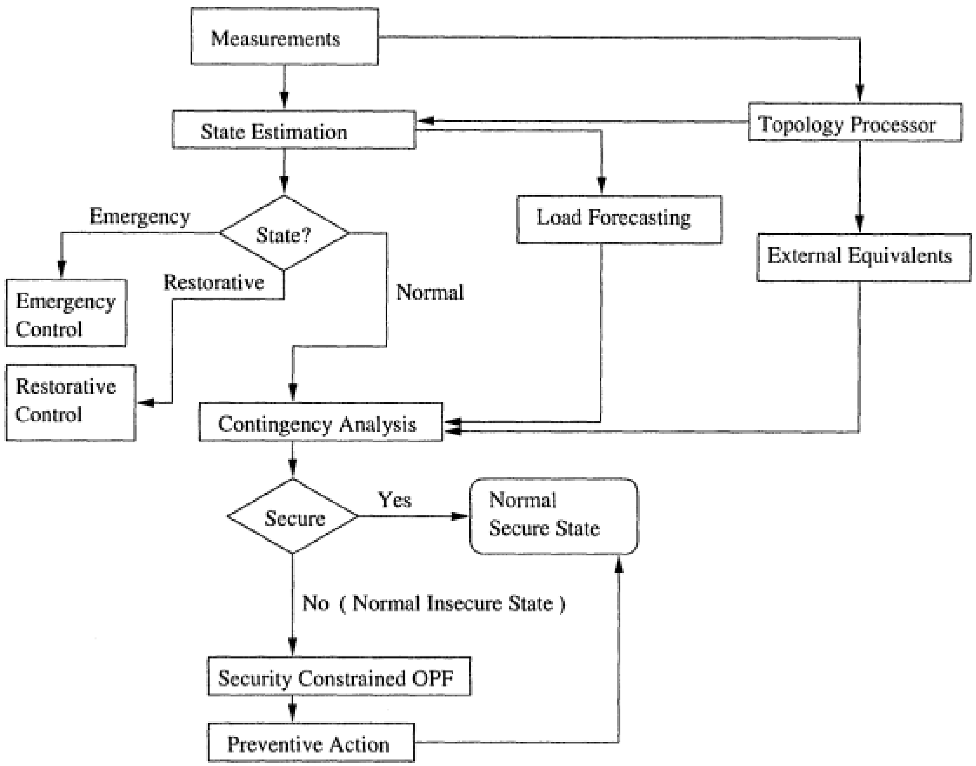


Fig. 2 On-line Static Security Assessment: Functional Diagram [1]

The Concept of SE

Why is it important to use SE in the power system-

Various constraints make it impossible to have a good picture of the power system [2]:

- Because of economical constraints, measurement devices can not be installed everywhere, so the data is incomplete.
- Because of the meter malfunction and the communication problem, the measurements are subject to error or loss, so the data is inaccurate, unreliable, and delayed.

[2] H. Wang and N. N. Schulz, "A revised branch current-based distribution system state estimation algorithm and meter placement impact," IEEE Trans. Power Syst., vol. 19, no. 1, pp. 207–213, Feb. 2004

The Concept of SE

Types of Measurement Errors

- **Random errors** – depend on the class of precision of the measurements (phasor measurement unit, smart meter, etc.).
- **Intermittent errors** – large noise or temporary failures due to communication or meter malfunction.
- **Systematic errors –** deterioration of measurements due to age, temperature, weather, and other environmental effects [3].

[3] Zhong, Shan, and Ali Abur, "Combined state estimation and measurement calibration," IEEE Trans. Power Syst., vol. 20, no. 1, pp. 458–465, Feb., 2005

Conventional SE Method

Traditionally, bus voltage magnitudes and phase angles have been used as state variables in transmission systems. The basic equation of SE can be written by:

$$z = \begin{bmatrix} z_1 \\ \vdots \\ z_m \end{bmatrix} = \begin{bmatrix} h_1(x_1, \dots, x_n) \\ \vdots \\ h_m(x_1, \dots, x_n) \end{bmatrix} + \begin{bmatrix} e_1 \\ \vdots \\ e_m \end{bmatrix} = h(x) + e$$

Where x_n is the **state variable** of bus *n*, z_m is the m-th **real measurements**, h_m is the nonlinear **measurement function** to connect *x* and *z*, and *e* is the **measurement error**.

Conventional SE Method

The problem can be formulated as a Weighted Least Square (WLS) optimization method [1]:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} (\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}))^T \boldsymbol{W} (\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}))$$

where $\hat{\boldsymbol{x}}$ is the **estimated state vector**, T is the **matrix transposition operation**, and $\boldsymbol{W}$ denotes the **weight matrix** that represents the user's confidence in the measured data. A widely-used choice for the weight matrix is $W = diag\{\sigma_1^{-2}, \ldots, \sigma_m^{-2}\}$, where σ_j^{-2} represents the variance of the measurement error corresponding to the j^{th} element of measurement z.

Conventional SE Method

Conventionally, Gauss-Newton method has been applied to iteratively solve the WLS problem [4]. The basic idea is to find a solution to the equation $\nabla J = 0$:

$$\boldsymbol{H}(\boldsymbol{x}(k)) = \frac{\partial J}{\partial \boldsymbol{x}(k)}$$

$$\boldsymbol{G}(k) = \boldsymbol{H}(\boldsymbol{x}(k))^T \boldsymbol{W} \boldsymbol{H}(\boldsymbol{x}(k))$$

$$\Delta \boldsymbol{x}(k) = \boldsymbol{G}(k)^{-1} \boldsymbol{H}(\boldsymbol{x}(k))^T \boldsymbol{W}(\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}(k)))$$

$$\boldsymbol{x}(k+1) = \boldsymbol{x}(k) + \Delta \boldsymbol{x}(k)$$

Where ***H*** is the Jacobian matrix with respect to the state variables and real measurements. ***J*** denotes the objective function of the WLS problem. ***G(k)*** is the system gain matrix.

[4] F. F. Wu, "Power system state estimation: a survey," International Journal of Electrical Power & Energy Systems, vol. 12, no. 2, pp. 80–87, Apr. 1990.

Conventional SE Method

State Variables

$$x = [V, \theta]$$

Measurement Variables

$$z = [I, V, P_b, Q_b, P_L, Q_L, P_L^s, Q_L^s]^{*}$$

Jacobian Matrix of the State Equations

$$H(x) = \begin{bmatrix} \frac{\partial P_b}{\partial V} & \cdots & \frac{\partial P_L}{\partial V} \\ \frac{\partial Q_b}{\partial \theta} & \cdots & \frac{\partial P_L}{\partial \theta} \end{bmatrix}$$

Weight matrix

$$W_{ii} = \begin{cases} 1 & \text{For the pseudo load} \\ 10 & \text{For the actual measurements} \end{cases}$$

I: Line current measurements
V: Voltage magnitudes
θ: Voltage angles
P_b: Branch real power
Q_b: Branch reactive power
P_L: Injection real power
Q_L: Injection reactive power
P_L^s: Pseudo injection real power
Q_L^s: Pseudo injection reactive power

* The voltage here can be used for both prediction features and verifications.

Contents

Transmission Grid vs. Distribution Grid

Differences between transmission system and distribution system

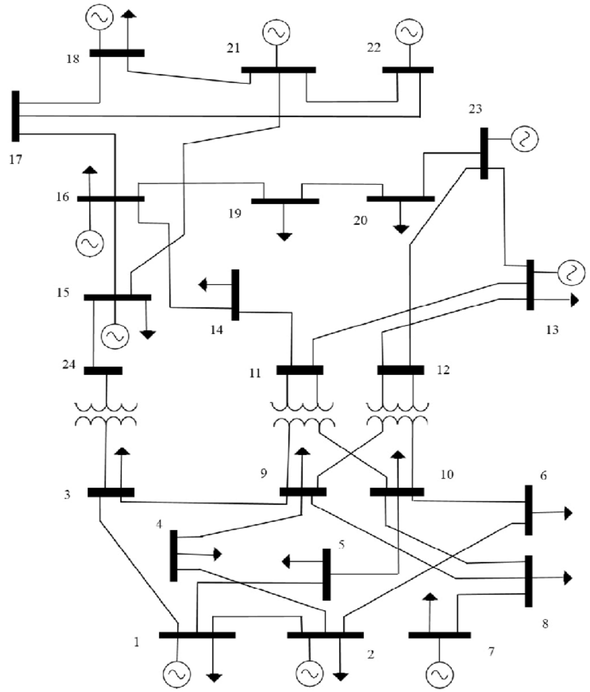


Fig. 3 IEEE 24 Bus Test System.

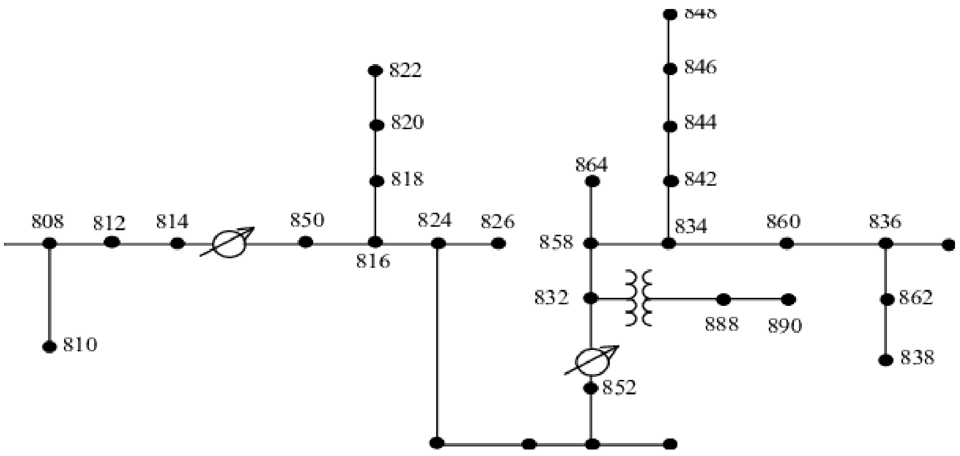


Fig. 4 IEEE 34 Bus Test System.

Meshed topology
Uni-directional power flows
Balanced lines and loads
Single phase analysis

Radial topology
Bi-directional power flows
Unbalanced lines and loads
Three phase analysis

Transmission Grid vs. Distribution Grid

Differences between transmission system and distribution system

Items	Transmission Network	Distribution Network
Topology	The general network topology is mesh-shaped and needs to be analyzed as a whole.	The power supply situation is regional power supply, the network topology in the region is radial, the closed-loop design between the regions, and the open-loop operation.
Network imbalance	The imbalance of the network is small and can basically be ignored. It can be considered that the three-phase line parameter balance and three-phase load balance can be analyzed in single-phase or positive sequence.	The three-phase line parameters are unbalanced, the R/X ratio fluctuates greatly, the three-phase load is unbalanced, and there are single-phase and two-phase loads, which cannot be analyzed independently.
SCADA measuring device	There are a large number of real-time measurement devices and a small number of pseudo-measurements, and the measurement redundancy is high.	A small amount of real-time measurement, a large number of load pseudo-measurement, from the perspective of real-time measurement, the measurement redundancy is low, and the network value is generally unobservable.
Network scale	A typical network generally contains hundreds of buses to one or two thousand buses.	A typical network generally contains 10,000 to 100,000 nodes.
Existing power plant	Generally, thermal power, large-scale hydropower and nuclear power generation, the output power is basically stable and adjustable with the load.	Mostly distributed DG distributed in the feeder of the distribution network, the output power fluctuates greatly, and has certain controllability.

The Concept of DSSE

Why do we need to perform DSSE-

With complex interactions in distribution networks and rapid growth of distributed energy resources (DER), electric vehicles, SCADA, and advanced metering infrastructure (AMI), DSSE is expected to become a significant function in monitoring and power management of smart grids by estimating the high accurate system states [5]-[6].

[5] "FERC staff report: Assessment of demand response and advanced metering - Dec. 2017." [Online]. Available: https://www.ferc.gov/legal/staff-reports/2017/DR-AM-Report2017.pdf.
[6] A. Primadianto and C. N. Lu, "A review on distribution system state estimation," IEEE Trans. Power Syst., vol. 32, no. 5, pp. 3875–3883, Sep. 2017.

Transmission System SE vs Distribution System SE

- System Analysis Process:
  - Transmission: Single phase analysis and one-line diagram of the system is configured.
  - Distribution: Unbalanced three-phase analysis and power flow constraints are necessary.
- Data Availability:
  - Transmission: Data is over-determined (the number of available measurements is more than the number of estimations).
  - Distribution: As the number of meter points is much lower in the distribution network, most of the measurement data used in DSSE are pseudo measurements data.

Transmission System SE vs Distribution System SE

- Measurement Jacobian Matrix:

Transmission:

$$H = \begin{bmatrix} \frac{\partial(Pinj)}{\partial\theta} & \frac{\partial(Pinj)}{\partial V} \\ \frac{\partial(Pfl)}{\partial\theta} & \frac{\partial(Pfl)}{\partial V} \\ \frac{\partial(Qinj)}{\partial\theta} & \frac{\partial(Qinj)}{\partial V} \\ \frac{\partial(Qfl)}{\partial\theta} & \frac{\partial(Qfl)}{\partial V} \\ \frac{\partial(Vmag)}{\partial\theta} & \frac{\partial(Vmag)}{\partial V} \\ & . \end{bmatrix}$$

Distribution:

$$\mathrm{H} = \begin{bmatrix} \frac{\partial((Iinj)eq\ of\ Pinj)}{\partial Ir} & \frac{\partial((Iinj)eq\ of\ Pinj)}{\partial Im} \\ \frac{\partial((Ifl)eq\ of\ Pfl)}{\partial Ir} & \frac{\partial((Ifl)eq\ of\ Pfl)}{\partial Im} \\ \frac{\partial((Iinj)eq\ of\ Qinj)}{\partial Ir} & \frac{\partial((Iinj)eq\ of\ Qinj)}{\partial Im} \\ \frac{\partial((Ifl)eq\ of\ Qfl)}{\partial Ir} & \frac{\partial((Ifl)eq\ of\ Qfl)}{\partial Im} \\ \frac{\partial(Vmag)}{\partial Ir} & \frac{\partial(Vmag)}{\partial Im} \\ \frac{\partial(Ifl)}{\partial Ir} & \frac{\partial(Ifl)}{\partial Im} \end{bmatrix}$$

Reference: Sarada Devi, M. S. N. G., and G. Yesuratnam. "Comparison of State Estimation Process on Transmission and Distribution Systems." Advances in Decision Sciences, Image Processing, Security and Computer Vision. Springer, Cham, 2020. 414-423.

Distribution System Real-time Measurements

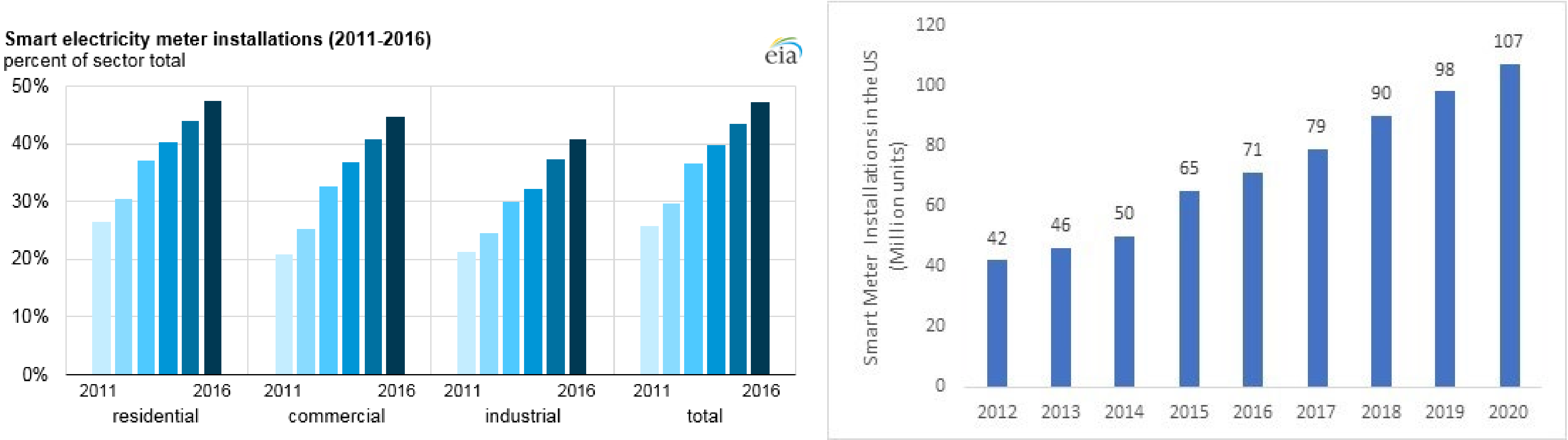


Fig. 5 Percent and numbers of Smart meter installations [7]

[7] Energy Information Administration. (2017) Annual Electric Power Industry Report. [Online]. Available:https://www.eia.gov/electricity/data/eia861/

The Process of DSSE

Like transmission system SE, DSSE is the process of inferring the values of the distribution system's state variables using a limited number of measured data at certain locations in the system [8].

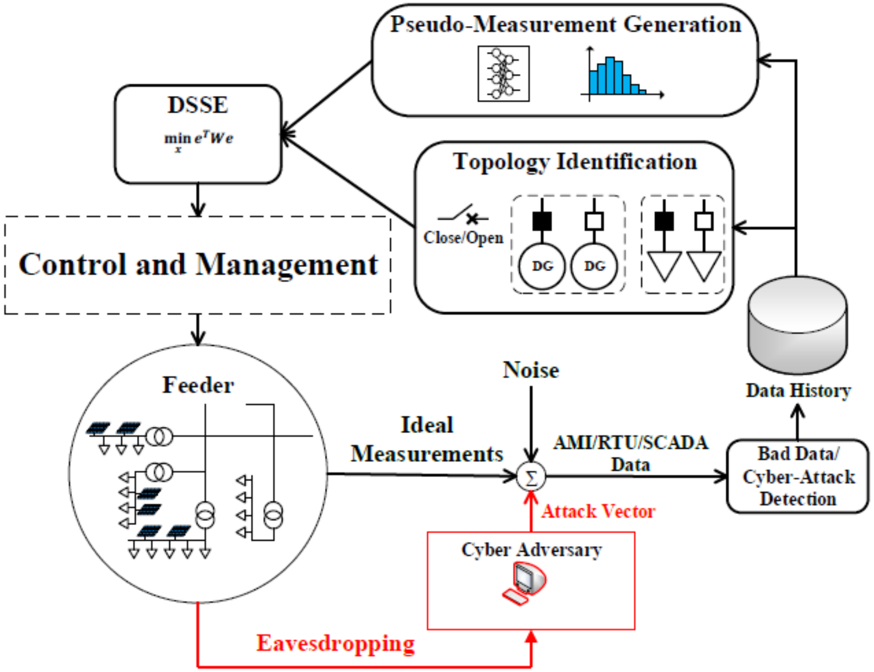


Fig. 6 DSSE function in smart grid environment [9].

[8] A. Monticelli, State estimation in electric power systems: a generalized approach. Springer Science & Business Media, 1999.
[9] K. Dehghanpour, Z. Wang, J. Wang, Y. Yuan and F. Bu, "A Survey on State Estimation Techniques and Challenges in Smart Distribution Systems," in IEEE Transactions on Smart Grid, vol. 10, no. 2, pp. 2312-2322, March 2019.

The Method of DSSE

The selection of state variables in DSSE is separated into two categories:

1. Voltage-Based DSSE [10]-[12]

2. Branch Current-Based SE (BCSE) [13]-[15]

[10] M. E. Baran and A. W. Kelley, “State estimation for real-time monitoring of distribution systems,” IEEE Trans. Power Syst., vol. 9, no. 3, pp. 1601–1609, Aug. 1994
[11] D. A. Haughton and G. T. Heydt, “A linear state estimation formulation for smart distribution systems,” IEEE Trans. Power Syst., vol. 28, no. 2, pp. 1187–1195, May 2013.
[12] C. N. Lu, J. H. Tang, and W. H. E. Liu, “Distribution system state estimation,” IEEE Trans. Power Syst., vol. 10, no. 1, pp. 229–240, Feb. 1995.
[13] M. E. Baran and A. W. Kelley, “A branch-current-based state estimation method for distribution systems,” IEEE Trans. Power Syst., vol. 10, no. 1, pp. 483–491, Feb. 1995.
[14] M. Pau, P. A. Pegoraro, and S. Sulis, “Efficient branch-current-based distribution system state estimation including synchronized measurements,” IEEE Trans. Instrum. Meas., vol. 62, no. 9, pp. 2419–2429, Sep. 2013.
[15] M. E. Baran, J. Jung, and T. E. McDermott, “Including voltage measurements in branch current state estimation for distribution systems,” In IEEE Power & Energy Society General Meeting, pp. 1–5, Jul. 2009.

The Method of DSSE

1. Voltage-Based DSSE [10]-[12]
   - State variables: node voltage magnitude and angle.
   - Equations: WLS approach with PF, similar to the conventional SE method.
   - Shortage: High computational complexity, mainly used for meshed networks, sensitive to line parameters. May not work satisfactorily for networks with a high R/X ratio [16].
2. Branch Current-Based SE (BCSE) [13]-[15]
   - State variables: branch current and angle.
   - Equations: WLS approach and can be expressed using the rectangular or polar form.
   - BCSE is more insensitive to line parameters than the conventional node-voltage-based SE methods [9] and has better computation speed and memory usage [2].

[16] Mohamed Ben Ahmed and Anouar Abdelhakim Boudhir. 2018. Innovations in Smart Cities and Applications: Proceedings of the 2nd Mediterranean Symposium on Smart City Applications (1st ed.). Springer Publishing Company, Incorporated.

Branch Current-Based SE Algorithm

The three-phase branch current, also known as the state variables of the system, x can be expressed as:

$$x = [I_1^{ph,real} \ldots I_l^{ph,real} \ldots I_N^{ph,real}, I_1^{ph,im} \ldots I_l^{ph,im} \ldots I_N^{ph,im}]^T$$

Where $I_l^{ph,real}$ and $I_l^{ph,im}$ represent the real and imaginary parts of the three-phase branch current at branch l, respectively and N represents the number of branches. The compact form can be expanded as:

$$I_l^{ph,real} = \begin{bmatrix} I_l^{a,real} \\ I_l^{b,real} \\ I_l^{c,real} \end{bmatrix} \qquad I_l^{ph,im} = \begin{bmatrix} I_l^{a,im} \\ I_l^{b,im} \\ I_l^{c,im} \end{bmatrix}$$

Branch Current-Based SE Algorithm

The system measurements considered are power, current magnitude, and voltage magnitude measurements. They are derived as follows:

(1) Power flow Measurements:

$$I_l^{ph} = \left(\frac{P_l^{ph} + jQ_l^{ph}}{V_i^{ph,k}} \right)^* = I_l^{ph,real} + jI_l^{ph,im}$$

where $V_i^{ph,k}$ is the estimated node voltage at the *k*-th iteration. The reason for converting all power measurements is to have linear relationships between the equivalent currents and the state variables as it can be observed from this equation.

Branch Current-Based SE Algorithm

(2) Current Magnitude Measurements:

$$\left|I_l^{ph}\right| = \sqrt{(I_l^{ph,real})^2 + (I_l^{ph,im})^2}$$

(3) Voltage Magnitude Measurements:

$$\left|V_j^{ph}\right| = \left|V_s^{ph} - \sum_{l=1}^{N} Z_l^{ph} I_l^{ph}\right|$$

where V_j^{ph} is the substation voltage.

Branch Current-Based SE Algorithm

Detailed Algorithm:

-Step 1 Initialization:

- Set the initial value of voltage at every node, such as 1 pu.
- **Backward Step**: Using the injected power at every node, the values of state variables (branch current magnitudes and phase angles) are computed starting from the end of networks.

-Step 2 WLS

- Using the WLS method, the state variable increments are obtained.
- Update the value of state variables.

Branch Current-Based SE Algorithm

Detailed Algorithm:

-Step 3 Forward Step:

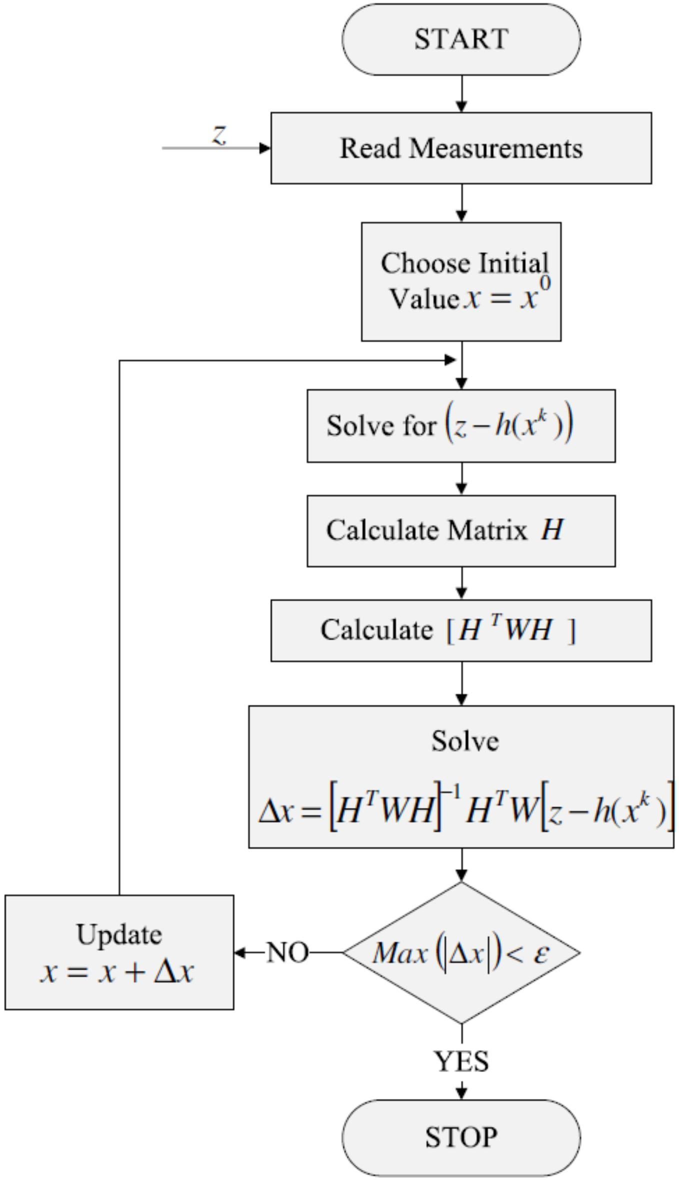
- Using the new values of state variables, the values of nodal voltages are calculated starting from the substation.

-Step 4 Convergence Analysis

- If the increments are smaller than the tolerance: stop. If not, return to step 2.

Block Diagram for solving WLS problem:

- Start by sitting the iteration $k = 0$.
- Initialize the node voltages.
- Find the gain matrix $G(x)$.
- After obtaining the branch current estimate, update the node voltages using the forward sweeping approach.
- Check for convergence, If no, update k = k + 1. Else, stop.



Contents

- Introduction to State Estimation (SE)
    - The concept of SE
    - Conventional SE method
- Introduction to Distribution System State Estimation (DSSE)
    - Transmission and distribution grids
    - The concept of DSSE
    - Conventional DSSE method
- **DSSE Research Topics**
    - **The challenges of DSSE**
        - **Observability problem**
        - **Metering device placement**
        - **Unbalanced problem**
        - **Topology configuration**
        - **Renewable integration**
        - **Robust DSSE methods**
- Smart meter data analysis
- Conclusion

The Challenges of DSSE

Compared to transmission system SE, DSSE is facing some unique challenges due to the characteristics of the system[2]. These challenges include:

- Low observability due to the lack of measurement device placements
- Higher R/X ratio
- Three-phase unbalanced system
- Communication issues and network topology identification problem
- Renewable energy and EV integration
- Cyber-security issues

Observability Problem

- Unlike transmission systems with a high data redundancy level, the distribution systems are generally undetermined with low observability.
- “Observability” refers to the system operator’s ability to solve the SE problem. That depends on the number and location of metering devices.
- Observability problem is one of the main challenges in applying transmission SE techniques to distribution systems directly [2].
- In the traditional WLS-based SE method, the number of measurements must be larger than the estimated states.
- The bad/missing measurement data also causes the observability problem.

Observability Problem

Distribution systems can be divided into three groups according to observability: fully observable systems, partially observable systems, and fully unobservable systems.

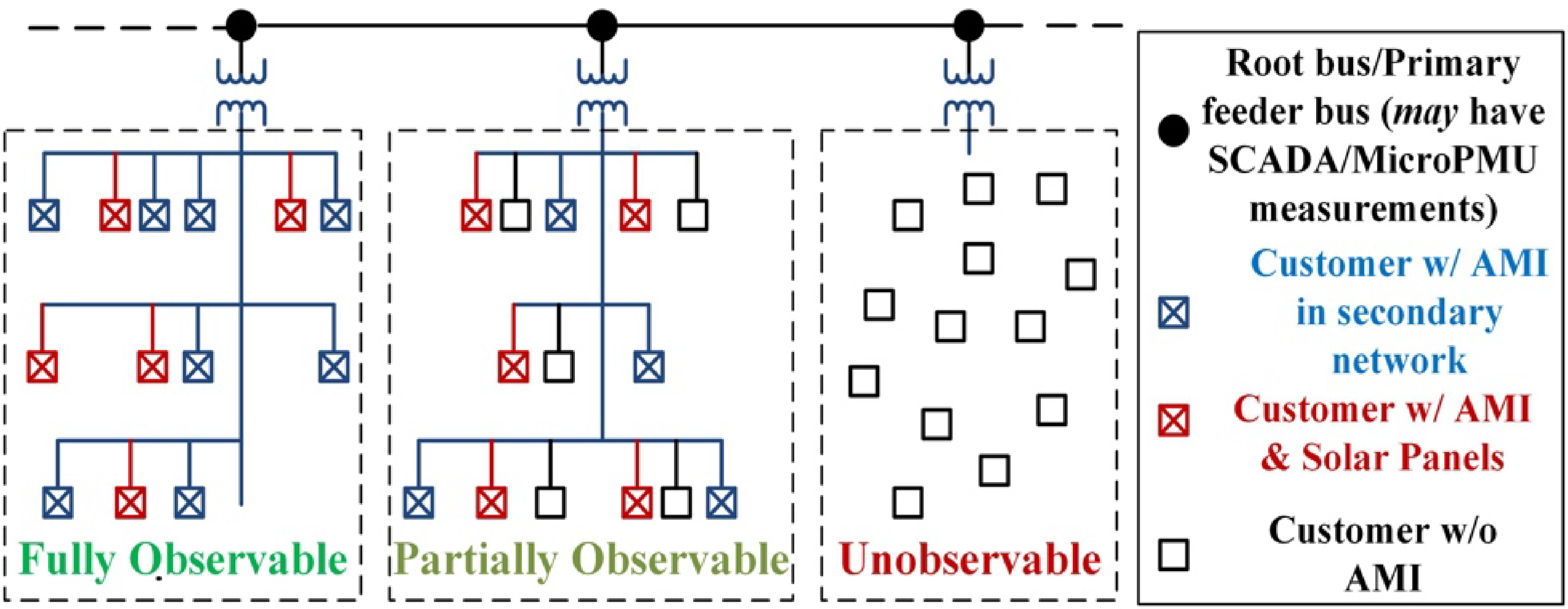


Fig. 9 Distribution Systems with different observability.

Distribution System Real-time Measurements

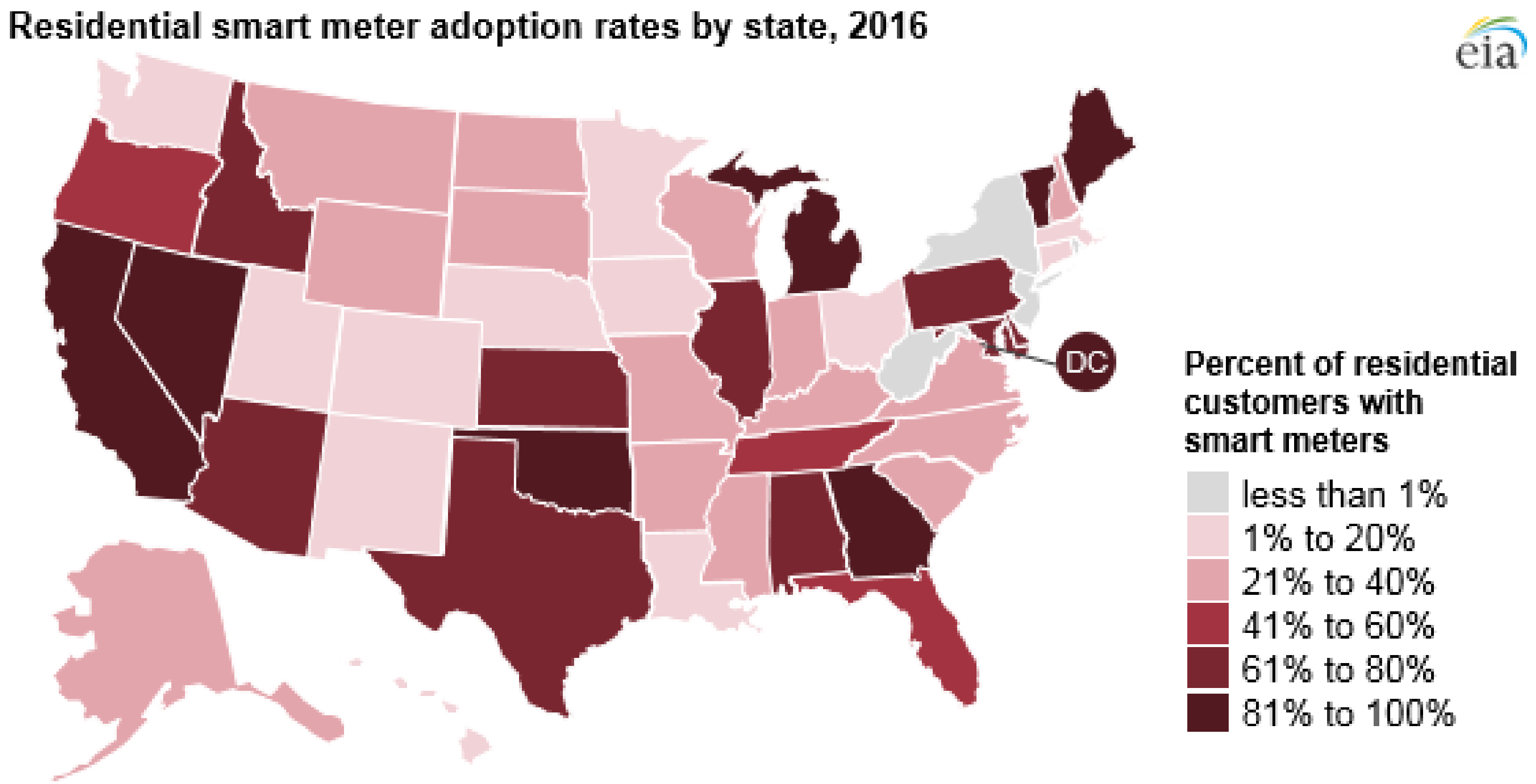


Fig. 10 Percent of Residential Smart meter installation rate by state, 2016 [7].

Observability Problem

How to solve the observability problem-

- Observability problem is addressed by generating pseudo-measurements when real measurements are unavailable [17].
- Pseudo-measurements are artificially-generated data points based on the data history of the distribution systems.
- Use of poor accurate pseudo-measurements will introduce high variance levels in the weight matrix, which could even lead to ill-conditioning of the DSSE problem [17].

[17] A. Angioni, T. Schlosser, F. Ponci, and A. Monti, "Impact of pseudo-measurements from new power profiles on state estimation in low-voltage grids," IEEE Trans. Instrum. Meas., vol. 65, no. 1, pp. 70–77, Jan. 2016.

Observability Problem

The existing data-driven pseudo-measurement methods can be roughly separated into two categories,

- Probabilistic and Statistical Approaches:

These methods employ spatial/temporal correlation and historic probability distribution data to generate reasonable pseudo-measurements and assess their uncertainty [18]-[21].

- Encouraged to read:

[18] C. Muscas, M. Pau, P. A. Pegoraro, and S. Sulis, “Effects of measurements and pseudo-measurements correlation in distribution system state estimation,” IEEE Trans. Instrum. Meas., vol. 63, no. 12, pp. 2813–2823, Dec. 2014.
[19] A. K. Ghosh, D. L. Lubkeman, M. J. Downey, and R. H. Jones, “Distribution circuit state estimation using a probabilistic approach,” IEEE Trans. Power Syst., vol. 12, no. 1, pp. 45–51, Feb. 1997.
[20] R. Singh, B. C. Pal, and R. A. Jabr, “Statistical representation of distribution system loads using Gaussian mixture model,” IEEE Trans. Power Syst., vol. 25, no. 1, pp. 29–37, Feb. 2010.
[21] R. Singh, B. C. Pal, and R. A. Jabr, “Distribution system state estimation through Gaussian mixture model of the load as pseudo-measurement,” IET Gener. Transm. Distrib., vol. 4, no. 1, pp. 50–59, Jan. 2009.

Observability Problem

Existing data-driven pseudo-measurement method can be roughly separated into two categories:

- Learning-Based Approaches:

Multiple machine learning algorithms have also been utilized to generate active/reactive power pseudo-measurement and uncertainty assessment [22]-[26].

- Encouraged to read:

[22] B. P. Hayes, J. K. Gruber, and M. Prodanovic, “A closed-loop state estimation tool for MV network monitoring and operation,” IEEE Trans. Smart Grid, vol. 6, no. 4, pp. 2116–2125, Jul. 2015.
[23] D. Gerbec, S. Gasperic, I. Smon, and F. Gubina, “Allocation of the load profiles to consumers using probabilistic neural networks,” IEEE Trans. Power Syst., vol. 20, no. 2, pp. 548–555, May 2005.
[24] E. Manitsas, R. Singh, B. C. Pal, and G. Strbac, “Distribution system state estimation using an artificial neural network approach for pseudo measurement modeling,” IEEE Trans. Power Syst., vol. 27, no. 4, pp. 1888–1896, Nov. 2012.
[25] Y. Yuan, K. Dehghanpour, F. Bu, and Z. Wang, "A Multi-Timescale Data-Driven Approach to Enhance Distribution System Observability," IEEE Transactions on Power Systems, vol. 34, no. 4, pp. 3168-3177, July 2019.
[26] K. Dehghanpour, Y. Yuan, Z. Wang and F. Bu, "A Game-Theoretic Data-Driven Approach for Pseudo-Measurement Generation in Distribution System State Estimation," in *IEEE Transactions on Smart Grid*.

Metering Device Placement

Optimizing the location of meters in distribution systems is a significant subject for research, given the size of the system and potentially limited financial resources [9].

Objective Function	Constraints	Solution Algorithm
Meter cost [27]	Estimation accuracy	Heuristic search
Estimation accuracy [28]	Meter number	Mixed integer semidefinite optimization
Network observability [29]	NA	Heuristic search
Meter cost & estimation accuracy [30]	Estimation accuracy	Multi-Objective evolutionary

[27] M. E. Baran, J. Zhu, and A. W. Kelley, “Meter placement for real-time monitoring of distribution feeders,” IEEE Trans. Power Syst., vol. 11, no. 1, pp. 332–337, Feb. 1996.
[28] T. C. Xygkis, G. N. Korres, and N. M. Manousakis, “Fisher information based meter placement in distribution grids via the d-optimal experimental design,” IEEE Trans. Smart Grid, vol. 9, no. 2, pp. 1452–1461, Mar. 2018.
[29] B. Brinkmann and M. Negnevitsky, “A probabilistic approach to observability of distribution networks,” IEEE Trans. Power Syst., vol. 32, no. 2, pp. 1169–1178, Mar. 2017.
[30] S. Prasad and D. M. V. Kumar, “Trade-offs in PMU and IED deployment for active distribution state estimation using multi-objective evolutionary algorithm,” IEEE Trans. Instrum. Meas., vol. 67, no. 6, pp. 1298–1307, Jun 2018.

Three Phase unbalanced Problem

In distribution systems, loads can be three-phase, two-phase, or single-phase. Hence it is desirable to use a three-phase model in DSSE [14]. The basic WLS SE method was adapted for three-phase analysis to address the phase unbalanced problem [31].

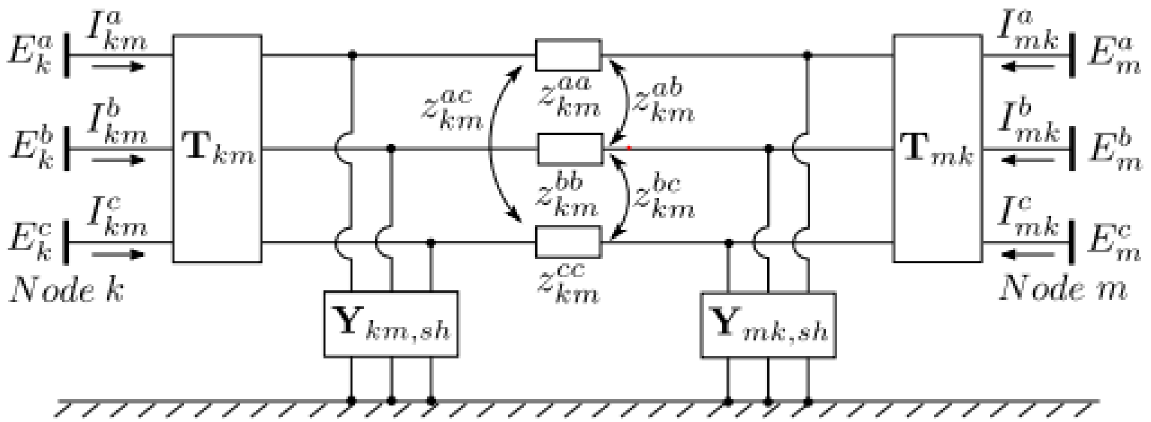


Fig. 11 Three-phase branch model [30].

[31] U. Kuhar, M. Pantos, G. Kosec, and A. Svigelj, “The impact of model and measurement uncertainties on a state estimation in three-phase distribution networks,” to appear in IEEE Trans. Smart Grid.

Three Phase unbalanced Problem - Solution

- Functions that relate measurements to the vector of state variables are developed from a three-phase branch model.
- The BCSE method that was demonstrated earlier is modeled in three phases.
- However, according to [30], the measurement uncertainties will impact estimation accuracy for different estimators, such as LAV, WLS, LMS, and SHGM.
- Examples using different system models to achieve three-phase distribution state estimation can be found in [32-34]

[32] Langner, Andre L., and Ali Abur. "Formulation of three-phase state estimation problem using a virtual reference." IEEE Transactions on Power Systems 36.1 (2020): 214-223.
[33] A. Majumdar and B. C. Pal, "A three-phase state estimation in unbalanced distribution networks with switch modelling," 2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI), 2016, pp. 474-478, doi: 10.1109/CMI.2016.7413793.
[34] F. Magnago, L. Zhang and R. Nagarkar, "Three phase distribution state estimation utilizing common information model," *2015 IEEE Eindhoven PowerTech*, 2015, pp. 1-6, doi: 10.1109/PTC.2015.7232515.

High R/X Ratio Problem

- Distribution line with a high R/X ratio is another challenge for SE [6].
- Recall the Jacobian Matrix ***H***: $H(x) = \begin{bmatrix} \frac{\partial P_b}{\partial V} & \cdots & \frac{\partial P_L}{\partial V} \\ \frac{\partial Q_b}{\partial \theta} & \cdots & \frac{\partial P_L}{\partial \theta} \end{bmatrix}$
- The high R/X ratio results in the ill-conditioning of ***H*** matrix. In the transmission system, the off-diagonal elements $\partial P/\partial V$ and $\partial Q/\partial \theta$ are neglected, that is $\partial P/\partial V \approx \partial Q/\partial \theta \approx 0$, because of weak coupling. With a high R/X ratio and strong coupling, the off-diagonal elements cannot be discarded in the distribution system.

High R/X Ratio Problem -Solution

- To avoid the ill-conditioning of ***H*** matrix, the ***P*** and ***Q*** measurements are handled by converting into equivalent current measurement ***I*** and making the ***H*** matrix independent of state variables equation.

- The equivalent current measurement can be expressed with voltage or a current variable for every iteration k.

$$I_{phase\ a,b,c}^{k} = \left(\frac{P + jQ}{V^{k}}\right)^{*}$$

- In this case, to address this challenge, branch current (BCSE) have been adopted as state variables, which turns out to be a more natural way of DSSE formulation [9].

[35] Bhattar, Poornachandratejasvi Laxman, Naran M. Pindoriya, and Anurag Sharma. "A combined survey on distribution system state estimation and false data injection in cyber-physical power distribution networks." *IET Cyber-Physical Systems: Theory & Applications* 6.2 (2021): 41-62.

Topology Configuration

DSSE relies on the basic assumption that we know the exact network model so that we can write the measurement functions ***h(x).*** Hence, it is necessary to perform topology configuration process to identify the current topology.

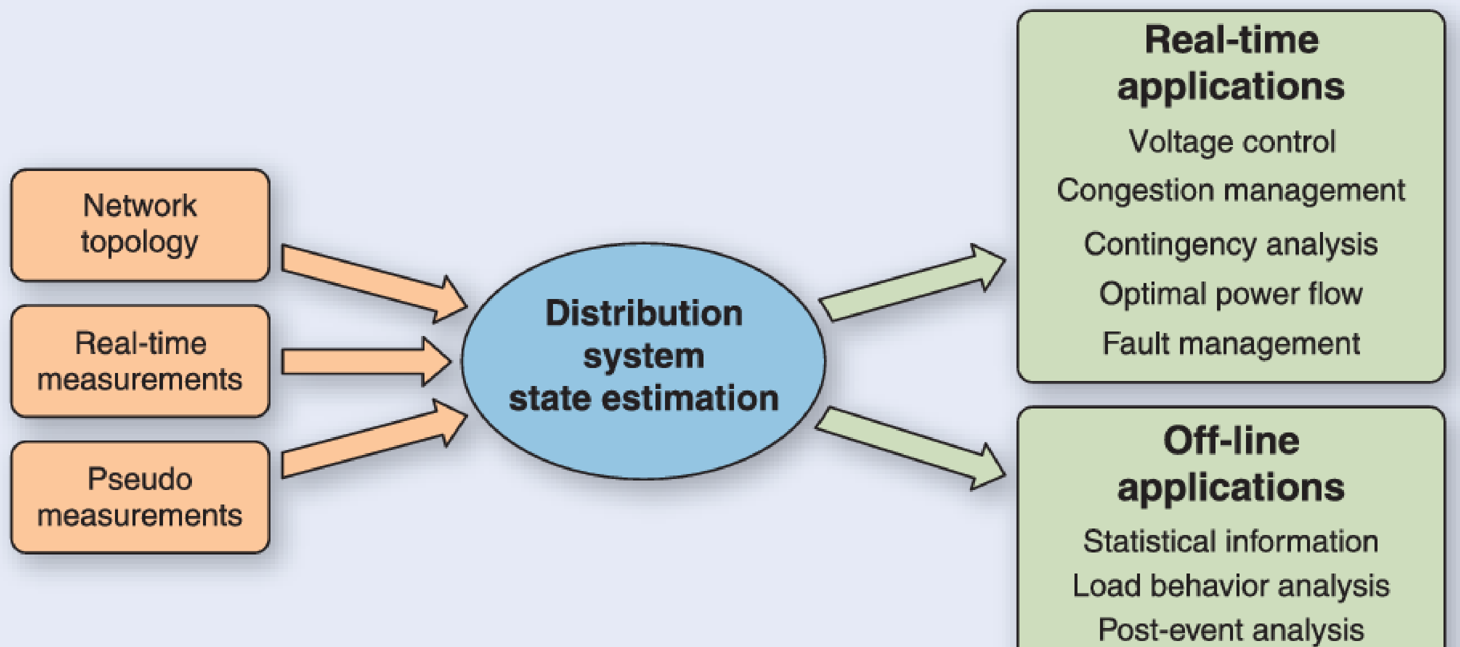


Fig. 30 DSSE and possible associated applications into a DMS.

Topology Configuration

The existing topology configuration method can be roughly separated into two categories:

- System Identification Approaches:

These methods assume the basic topology of the network is known to the system operator. However, due to local events, such as faults, line disconnections, switching events, etc, the basic topology will undergo local changes over time. [36]-[40].

[36] G. N. Korres and N. M. Manousakis, “A state estimation algorithm for monitoring topology changes in distribution systems,” in Proc. IEEE Power Energy Soc. Gen. Meeting, San Diego, CA, USA, Jul. 2012, pp. 1–8.
[37] M. E. Baran, J. Jung, and T. E. McDermott, “Topology error identification using branch current state estimation for distribution systems,” In IEEE Transmission & Distribution Conference & Exposition: Asia and Pacific, pp. 1–4, Oct. 2009.
[38] D. Singh, J. P. Pandey, and D. S. Chauhan, “Topology identification, bad data processing, and state estimation using fuzzy pattern matching,” IEEE Trans. Power Syst., vol. 20, no. 3, pp. 1570–1579, Aug. 2005.
[39] G. Cavraro and R. Arghandeh, “Power distribution network topology detection with time-series signature verification method,” IEEE Trans. Power Syst., vol. 33, no. 4, pp. 3500–3509, Jul. 2018.
[40] W. Luan, J. Peng, M. Maras, J. Lo, and B. Harapnuk, “Smart meter data analytics for distribution network connectivity verification,” IEEE Trans. Smart Grid, vol. 6, no. 4, pp. 1964–1971, Jul. 2015.

Topology Configuration

The existing topology configuration method can be roughly separated into two categories:

- Topology learning Approaches:

These methods assume that the system operator has very limited or no knowledge of the basic topology of the network. Hence, the objective is to learn the network's topology using nodal and branch measurements [41]-[44].

[41] M. Babakmehr, M. G. Simões, M. B. Wakin, and F. Harirchi, "Compressive sensing-based topology identification for smart grids," IEEE Trans. Ind. Informat., vol. 12, no. 2, pp. 532–543, Apr. 2016.
[42] Y. Weng, Y. Liao, and R. Rajagopal, "Distributed energy resources topology identification via graphical modeling," IEEE Trans. Power Syst., vol. 32, no. 4, pp. 2682–2694, Jul. 2017.
[43] S. J. Pappu, N. Bhatt, R. Pasumarthy, and A. Rajeswaran, "Identifying topology of low voltage distribution networks based on smart meter data," IEEE Trans. Smart Grid, vol. 9, no. 5, pp. 5113–5122, Sep. 2018.
[44] J. Yu, Y. Weng, and R. Rajagopal, "PaToPa: A data-driven parameter and topology joint estimation framework in distribution grids," IEEE Trans. Power Syst., vol. 33, no. 4, pp. 4335–4347, Jul. 2018.

Renewable Energy Integration

- The higher penetration of renewable power resources introduces a higher level of uncertainty in DSSE.
- With the integration of DER and EV charging station, typical load patterns are becoming unreliable and uncertain.
- The non-Gaussian distribution of renewable generation would adversely affect WLS-based DSSE methods [9].
- Fast changes in system states can result in unreasonable errors of the WLS-based DSSE [45].

[45] Y. Weng, R. Negi, C. Faloutsos, and M. D. Ilic, "Robust data-driven state estimation for smart grid," IEEE Trans. Smart Grid, vol. 8, no. 4, pp. 1956–1967, Jul. 2017.

Renewable Energy Integration

- Probabilistic methods represent the major group of techniques for modeling the impacts of renewable uncertainty on DSSE [9].
  - Use GMM technique to obtain the non-Gaussian distribution of renewable power [46].
  - Use Beta distribution function to generate renewable pseudo-measurement [47].

[46] G. Valverde, A. T. Saric, and V. Terzija, “Stochastic monitoring of distribution networks including correlated input variables,” IEEE Trans. Power Syst., vol. 28, pp. 246–255, Feb. 2013.
[47] A. Angioni, T. Schlosser, F. Ponci, and A. Monti, “Impact of pseudo measurements from new power profiles on state estimation in low voltage grids,” IEEE Trans. Instrum. Meas., vol. 65, no. 1, pp. 70–77, Jan. 2016.

Cyber Security

Due to the vulnerability of the power system against cyber-attacks has been observed in practice, several common types of cyber-attack related to SE have been modeled in the literature:

- False data injection [48]-[51]
- Topology attacks [52]-[53]
- Data privacy attacks [54]

[48] Q. Yang et al., "On false data-injection attacks against power system state estimation: Modeling and countermeasures," IEEE Trans. Parallel Distrib. Syst, vol. 25, no. 3, pp. 717–729, Mar. 2014.
[49] S. Li, Y. Yilmaz, and X. Wang, "Quickest detection of false data injection attack in wide-area smart grids," IEEE Trans. Smart Grid, vol. 6, no. 6, pp. 2725–2735, Nov. 2015.
[50] J. Liang, L. Sankar, and O. Kosut, "Vulnerability analysis and consequences of false data injection attack on power system state estimation," IEEE Trans. Power Syst., vol. 31, no. 5, pp. 3864–3872, Sep. 2016.
[51] Y. Chakhchoukh and H. Ishii, "Enhancing robustness to cyber-attacks in power systems through multiple least trimmed squares state estimations," IEEE Trans. Power Syst., vol. 31, no. 6, pp. 4395–4405, Nov. 2016.
[52] Y. Chakhchoukh and H. Ishii, "Coordinated cyber-attacks on the measurement function in hybrid state estimation," IEEE Trans. Power Syst., vol. 30, no. 5, pp. 2487–2497, Sep. 2015.
[53] J. Zhang and L. Sankar, "Physical system consequences of unobservable state-and-topology cyber-physical attacks," IEEE Trans. Smart Grid, vol. 7, no. 4, pp. 2016–2025, Jul. 2016.
[54] H. Li, L. Lai, and W. Zhang, "Communication requirement for reliable and secure state estimation and control in smart grid," IEEE Trans. Smart Grid, vol. 2, no. 3, pp. 476–486, Sep. 2011.

Robust DSSE Methods

- WLS estimator is a quadratic form of the maximum likelihood estimator and can be stated as the minimization of the weighted sum of squares. It is a fast and widely-used mathematical formulation. However, WLS is sensitive to bad data.

- To handle the uncertainty of measurement data, alternative mathematical formulations have been proposed to increase the robustness of the state estimator.

Robust DSSE Methods

- *Least Absolute Value estimator (LAV)*: LAV estimator is based on the minimization of the L_1 norm of weighted measurement residual [55].
- *Schweppe Huber Generalised M* (SHGM): SHGM estimator is an estimator that combines both WLS and WLAV [56].
- *Matrix completion:* MC Utilizes a direct connection between the row and column elements within the matrix. The method can estimate missing values in the smart meter measurement matrices caused by low observability [57-58].

[55] R. Jabr, B. Pal, and R. Singh, "Choice of estimator for distribution system state estimation," IET Generation, Transmission & Distribution, vol. 3, no. 7, pp. 666–678, Jul. 2009.
[56] Donti, Priya L., et al. "Matrix completion for low-observability voltage estimation." *IEEE Transactions on Smart Grid* 11.3, 2019, pp. 2520-2530.
[57] Zhang, Yingchen, et al. "State estimation in low-observable distribution systems using matrix completion." *National Renewable Energy Lab. (NREL)*, 2019.
[58] Liu, Bo, et al. "Robust matrix completion state estimation in distribution systems." *2019 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2019.

Smart Meter Data Analytics

Why is it important to perform SM data analytics-

- The widespread popularity of SMs enables an immense amount of fine-grained electricity consumption data to be collected [59].
- High-resolution data from SM provide rich information to understand the consumption behaviors of the consumers.
- SM data provides a unique opportunity to develop a data-enabled modernized power system [60].

[59] Y. Wang, Q. Chen, T. Hong and C. Kang, "Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges," in IEEE Transactions on Smart Grid, vol. 10, no. 3, pp. 3125-3148, May 2019.
[60] H. Sun, Z. Wang, J. Wang, Z. Huang, N. Carrington and J. Liao, "Data-Driven Power Outage Detection by Social Sensors," in IEEE Transactions on Smart Grid, vol. 7, no. 5, pp. 2516-2524, Sept. 2016.

Smart Meter Data Analytics

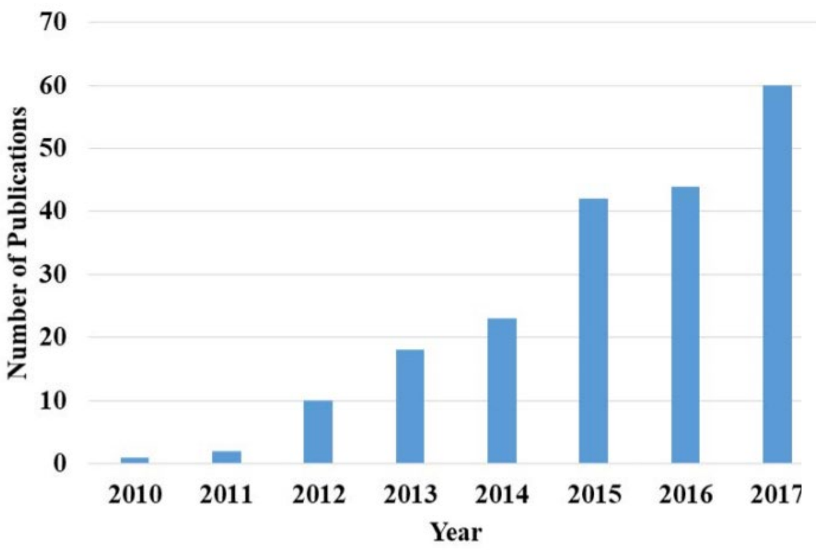


Fig. 13 Number of publication about SM data analytics。

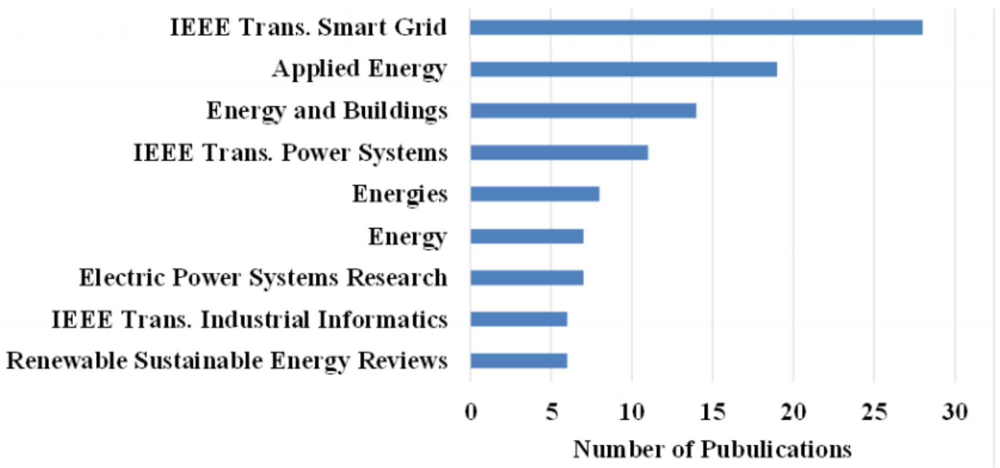


Fig. 14 Number of publications in most popular journals.

- In 2010, the number of SM data analytics publications was at a low level.
- The number of publications increased rapidly from 2012.

Smart Meter Data Analytics

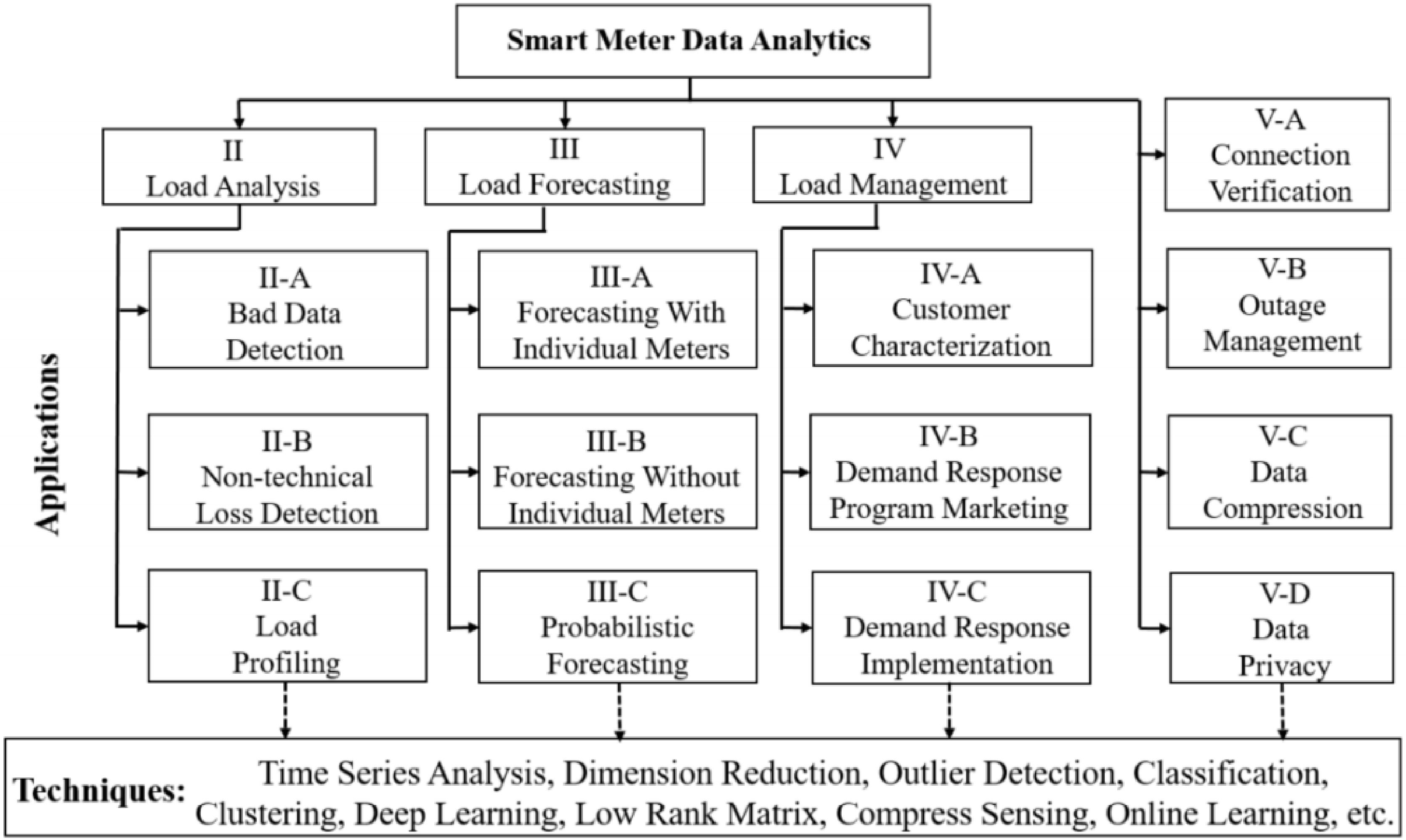


Fig. 15 Taxonomy of SM data analytics。

Conclusions

- The goal of DSSE is to infer the values of the distribution system's state variables using a limited number of measured data. The DSSE is a numerical process to map data measurements to system state variables.

- Technically, conventional transmission level SE approaches cannot be directly applied to the DSSE due to various challenges.

- To address these challenges, different methods are proposed for DSSE. Most recent works are concentrated on using SM data analytics-based approaches to improve the conventional DSSE.

- How to take advantage of massive SM data to enhance the efficiency of the demand side has become an important topic.

Appendix I: Matlab Example of Solving WLS Problem

IOWA STATE UNIVERSITY

```
A=importdata('H_data.m'); % import bus and line data with 5 columns, this is saved in another file
fbus=A(:,1); %From bus
tbus=A(:,2); %To bus
x=A(:,3); % Reactance in pu
Zm=A(:,4);% Impedance in pu
sig=A(:,5);% Susceptance in pu
buses=input('Enter the number of buses : \n');
B=input('Enter the reference bus : ');
for i=1:buses
        H(i,fbus(i))=1/x(i);
        H(i,tbus(i))=-1/x(i);
end
H(:,B)=[];    % Compute H matrix
for i=1:buses
    R(i,i)=sig(i)^2; % Compute R Matrix
end
[nm,ns]=size(H);
if ns<nm
    xest=[H'*R^-1*H]^-1*H'*R^-1*Zm;  %Compute Gain matrix and forward sweep
else if ns==nm
        xest=H^-1*Zm;
    else if ns>nm
            H'*[H*H']^-1*Zm;
        end
    end
end
Zt=H*xest; %True Value
e=Zm-Zt; %estimation error
fprintf(' The H Matrix \n');
disp(H);
fprintf(' The R inverse Matrix \n');
disp(R^-1);
fprintf('Estimated Value \n');
disp(xest);
fprintf('\n Measured value \n');
disp(Zm);
fprintf('\n True Value \n');
disp(Zt);
fprintf('\nError in equipment \n');
disp(e);
```

Display Results:

For 3 bus test case, use bus 1 as reference bus:

The H Matrix

-5.0000	0
0	-2.5000
-4.0000	4.0000

The R inverse Matrix

10000	0	0
0	10000	0
0	0	10000

Measured value

0.6000
0.4000
0.4050

True Value

0.7076
0.1848
0.2705

Error in equipment

-0.1076
0.2152
0.1345

Appendix II: VBSE Matlab Code Using WLS Method

IOWA STATE UNIVERSITY

- The IEEE 14 bus system is used in this example.

Bus data:

```
%           |Bus | Type | Vsp | theta | PGi | QGi | PLi | QLi |  Qmin | Qmax |
busdata14 = [ 1      1     1.060    0         0       0       0       0         0         0;
              2      2     1.045    0        40    42.4    21.7    12.7       -40        50;
              3      2     1.010    0         0    23.4    94.2    19.0         0        40;
              4      3     1.0      0         0       0    47.8    -3.9         0         0;
              5      3     1.0      0         0       0     7.6     1.6         0         0;
              6      2     1.070    0         0    12.2    11.2     7.5        -6        24;
              7      3     1.0      0         0       0     0.0     0.0         0         0;
              8      2     1.090    0         0    17.4     0.0     0.0        -6        24;
              9      3     1.0      0         0       0    29.5    16.6         0         0;
              10     3     1.0      0         0       0     9.0     5.8         0         0;
              11     3     1.0      0         0       0     3.5     1.8         0         0;
              12     3     1.0      0         0       0     6.1     1.6         0         0;
              13     3     1.0      0         0       0    13.5     5.8         0         0;
              14     3     1.0      0         0       0    14.9     5.0         0         0;];
```

Line Data:

```
%           |  From |  To    |   R      |   X      |     B/2    |  X'mer   |
%           |  Bus  | Bus    |  pu      |  pu      |     pu     | TAP (a)  |
linedata14 =[1        2        0.01938    0.05917    0.0264          1
             1        5        0.05403    0.22304    0.0246          1
             2        3        0.04699    0.19797    0.0219          1
             2        4        0.05811    0.17632    0.0170          1
             2        5        0.05695    0.17388    0.0173          1
             3        4        0.06701    0.17103    0.0064          1
             4        5        0.01335    0.04211    0.0             1
             4        7        0.0        0.20912    0.0         0.978
             4        9        0.0        0.55618    0.0         0.969
             5        6        0.0        0.25202    0.0         0.932
             6       11        0.09498    0.19890    0.0             1
             6       12        0.12291    0.25581    0.0             1
             6       13        0.06615    0.13027    0.0             1
             7        8        0.0        0.17615    0.0             1
             7        9        0.0        0.11001    0.0             1
             9       10        0.03181    0.08450    0.0             1
             9       14        0.12711    0.27038    0.0             1
            10       11        0.08205    0.19207    0.0             1
            12       13        0.22092    0.19988    0.0             1
            13       14        0.17093    0.34802    0.0             1 ];
```

Measurement Data:

```
% Vi - 1, Pi - 2, Qi - 3, Pij - 4, Qij - 5, Iij - 6;

%          |Msnt |Type | Value | From | To | Rii |
zdata14    = [ %---- Voltage Magnitude ------------%
             1      1     1.06       1         0    9e-4;
             %-----------------------------------%
             %---- Real Power Injection ---------%
             2      2     0.1830     2         0    1e-4;
             3      2    -0.9420     3         0    1e-4;
             4      2     0.00       7         0    1e-4;
             5      2     0.00       8         0    1e-4;
             6      2    -0.0900    10         0    1e-4;
             7      2    -0.0350    11         0    1e-4;
             8      2    -0.0610    12         0    1e-4;
             9      2    -0.1490    14         0    1e-4;
            %-------------------------------------%
            %---- Reactive Power Injection ------%
            10      3     0.3523     2         0    1e-4;
            11      3     0.0876     3         0    1e-4;
            12      3     0.00       7         0    1e-4;
            13      3     0.2103     8         0    1e-4;
            14      3    -0.0580    10         0    1e-4;
            15      3    -0.0180    11         0    1e-4;
            16      3    -0.0160    12         0    1e-4;
            17      3    -0.0500    14         0    1e-4;
            %-------------------------------------%
            %------ Real Power Flow ------------- %
            18      4     1.5708     1         2    64e-6;
            19      4     0.7340     2         3    64e-6;
            20      4    -0.5427     4         2    64e-6;
            21      4     0.2707     4         7    64e-6;
            22      4     0.1546     4         9    64e-6;
            23      4    -0.4081     5         2    64e-6;
            24      4     0.6006     5         4    64e-6;
            25      4     0.4589     5         6    64e-6;
            26      4     0.1834     6        13    64e-6;
            27      4     0.2707     7         9    64e-6;
            28      4    -0.0816    11         6    64e-6;
            29      4     0.0188    12        13    64e-6;
            %-------------------------------------%
            %------ Real Power Flow ------------- %
            30      5    -0.1748     1         2    64e-6;
            31      5     0.0594     2         3    64e-6;
            32      5     0.0213     4         2    64e-6;
            33      5    -0.1540     4         7    64e-6;
            34      5    -0.0264     4         9    64e-6;
            35      5    -0.0193     5         2    64e-6;
            36      5    -0.1006     5         4    64e-6;
            37      5    -0.2084     5         6    64e-6;
            38      5     0.0998     6        13    64e-6;
            39      5     0.1480     7         9    64e-6;
            40      5    -0.0864    11         6    64e-6;
            41      5     0.0141    12        13    64e-6;];
            %-------------------------------------%
```

State Estimation using Weighted Least Square Method:

Parameter Settings:

```
num = 14; % IEEE - 14 bus system
ybus = ybusppg(num); % Get YBus
zdata = zdatas(num); % Get Measurement data
bpq = bbusppg(num); % Get B data
nbus = max(max(zdata(:,4)),max(zdata(:,5))); % Get number of buses
type = zdata(:,2); % Type of measurement, Vi - 1, Pi - 2, Qi - 3, Pij - 4, Qij - 5, Iij - 6
z = zdata(:,3); % Measuement values
fbus = zdata(:,4); % From bus
tbus = zdata(:,5); % To bus
Ri = diag(zdata(:,6)); % Measurement Error
V = ones(nbus,1); % Initialize the bus voltages
del = zeros(nbus,1); % Initialize the bus angles
E = [del(2:end); V];    % State Vector
G = real(ybus);
B = imag(ybus);

vi = find(type == 1); % Index of voltage magnitude measurements
ppi = find(type == 2); % Index of real power injection measurements
qi = find(type == 3); % Index of reactive power injection measurements
pf = find(type == 4); % Index of real powerflow measurements
qf = find(type == 5); % Index of reactive powerflow measurements

nvi = length(vi); % Number of Voltage measurements
npi = length(ppi); % Number of Real Power Injection measurements
nqi = length(qi); % Number of Reactive Power Injection measurements
npf = length(pf); % Number of Real Power Flow measurements
nqf = length(qf); % Number of Reactive Power Flow measurements

iter = 1;
tol = 5;
```

```
while(tol > 1e-4)

    %Measurement Function, h
    h1 = V(fbus(vi),1);
    h2 = zeros(npi,1);
    h3 = zeros(nqi,1);
    h4 = zeros(npf,1);
    h5 = zeros(nqf,1);

    for i = 1:npi
        m = fbus(ppi(i));
        for k = 1:nbus
            h2(i) = h2(i) + V(m)*V(k)*(G(m,k)*cos(del(m)-del(k)) + B(m,k)*sin(del(m)-del(k)));
        end
    end

    for i = 1:nqi
        m = fbus(qi(i));
        for k = 1:nbus
            h3(i) = h3(i) + V(m)*V(k)*(G(m,k)*sin(del(m)-del(k)) - B(m,k)*cos(del(m)-del(k)));
        end
    end

    for i = 1:npf
        m = fbus(pf(i));
        n = tbus(pf(i));
        h4(i) = -V(m)^2*G(m,n) - V(m)*V(n)*(-G(m,n)*cos(del(m)-del(n)) - B(m,n)*sin(del(m)-del(n)));
    end

    for i = 1:nqf
        m = fbus(qf(i));
        n = tbus(qf(i));
        h5(i) = -V(m)^2*(-B(m,n)+bpq(m,n)) - V(m)*V(n)*(-G(m,n)*sin(del(m)-del(n)) + B(m,n)*cos(del(m)-del(n)));
    end

    h = [h1; h2; h3; h4; h5];

    % Residue
    r = z - h;
```

```
% Jacobian
    % H11 - Derivative of V with respect to angles, All Zeros
    H11 = zeros(nvi,nbus-1);

    % H12 - Derivative of V with respect to V
    H12 = zeros(nvi,nbus);
    for k = 1:nvi
        for n = 1:nbus
            if n == k
                H12(k,n) = 1;
            end
        end
    end

    % H21 - Derivative of Real Power Injections with Angles
    H21 = zeros(npi,nbus-1);
    for i = 1:npi
        m = fbus(ppi(i));
        for k = 1:(nbus-1)
            if k+1 == m
                for n = 1:nbus
                    H21(i,k) = H21(i,k) + V(m)* V(n)*(-G(m,n)*sin(del(m)-del(n)) + B(m,n)*cos(del(m)-del(n)));
                end
                H21(i,k) = H21(i,k) - V(m)^2*B(m,m);
            else
                H21(i,k) = V(m)* V(k+1)*(G(m,k+1)*sin(del(m)-del(k+1)) - B(m,k+1)*cos(del(m)-del(k+1)));
            end
        end
    end
```

```matlab
% H22 - Derivative of Real Power Injections with V
    H22 = zeros(npi,nbus);
    for i = 1:npi
        m = fbus(ppi(i));
        for k = 1:(nbus)
            if k == m
                for n = 1:nbus
                    H22(i,k) = H22(i,k) + V(n)*(G(m,n)*cos(del(m)-del(n)) + B(m,n)*sin(del(m)-del(n)));
                end
                H22(i,k) = H22(i,k) + V(m)*G(m,m);
            else
                H22(i,k) = V(m)*(G(m,k)*cos(del(m)-del(k)) + B(m,k)*sin(del(m)-del(k)));
            end
        end
    end

    % H31 - Derivative of Reactive Power Injections with Angles
    H31 = zeros(nqi,nbus-1);
    for i = 1:nqi
        m = fbus(qi(i));
        for k = 1:(nbus-1)
            if k+1 == m
                for n = 1:nbus
                    H31(i,k) = H31(i,k) + V(m)* V(n)*(G(m,n)*cos(del(m)-del(n)) + B(m,n)*sin(del(m)-del(n)));
                end
                H31(i,k) = H31(i,k) - V(m)^2*G(m,m);
            else
                H31(i,k) = V(m)* V(k+1)*(-G(m,k+1)*cos(del(m)-del(k+1)) - B(m,k+1)*sin(del(m)-del(k+1)));
            end
        end
    end
```

```
% H32 - Derivative of Reactive Power Injections with V
    H32 = zeros(nqi,nbus);
    for i = 1:nqi
        m = fbus(qi(i));
        for k = 1:(nbus)
            if k == m
                for n = 1:nbus
                    H32(i,k) = H32(i,k) + V(n)*(G(m,n)*sin(del(m)-del(n)) - B(m,n)*cos(del(m)-del(n)));
                end
                H32(i,k) = H32(i,k) - V(m)*B(m,m);
            else
                H32(i,k) = V(m)*(G(m,k)*sin(del(m)-del(k)) - B(m,k)*cos(del(m)-del(k)));
            end
        end
    end

    % H41 - Derivative of Real Power Flows with Angles
    H41 = zeros(npf,nbus-1);
    for i = 1:npf
        m = fbus(pf(i));
        n = tbus(pf(i));
        for k = 1:(nbus-1)
            if k+1 == m
                H41(i,k) = V(m)* V(n)*(-G(m,n)*sin(del(m)-del(n)) + B(m,n)*cos(del(m)-del(n)));
            else if k+1 == n
                H41(i,k) = -V(m)* V(n)*(-G(m,n)*sin(del(m)-del(n)) + B(m,n)*cos(del(m)-del(n)));
                else
                    H41(i,k) = 0;
                end
            end
        end
    end
```

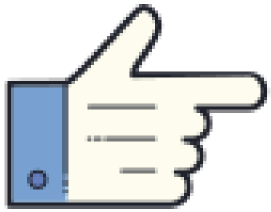
```
% H42 - Derivative of Real Power Flows with V
    H42 = zeros(npf,nbus);
    for i = 1:npf
        m = fbus(pf(i));
        n = tbus(pf(i));
        for k = 1:nbus
            if k == m
                H42(i,k) = -V(n)*(-G(m,n)*cos(del(m)-del(n)) - B(m,n)*sin(del(m)-del(n))) - 2*G(m,n)*V(m);
            else if k == n
                H42(i,k) = -V(m)*(-G(m,n)*cos(del(m)-del(n)) - B(m,n)*sin(del(m)-del(n)));
                else
                    H42(i,k) = 0;
                end
            end
        end
    end

    % H51 - Derivative of Reactive Power Flows with Angles
    H51 = zeros(nqf,nbus-1);
    for i = 1:nqf
        m = fbus(qf(i));
        n = tbus(qf(i));
        for k = 1:(nbus-1)
            if k+1 == m
                H51(i,k) = -V(m)* V(n)*(-G(m,n)*cos(del(m)-del(n)) - B(m,n)*sin(del(m)-del(n)));
            else if k+1 == n
                H51(i,k) = V(m)* V(n)*(-G(m,n)*cos(del(m)-del(n)) - B(m,n)*sin(del(m)-del(n)));
                else
                    H51(i,k) = 0;
                end
            end
        end
    end
```

```
    % H52 - Derivative of Reactive Power Flows with V..
      H52 = zeros(nqf,nbus);
      for i = 1:nqf
          m = fbus(qf(i));
          n = tbus(qf(i));
          for k = 1:nbus
              if k == m
                  H52(i,k) = -V(n)*(-G(m,n)*sin(del(m)-del(n)) + B(m,n)*cos(del(m)-del(n))) - 2*V(m)*(-
  B(m,n)+ bpq(m,n));
              else if k == n
                  H52(i,k) = -V(m)*(-G(m,n)*sin(del(m)-del(n)) + B(m,n)*cos(del(m)-del(n)));
                  else
                      H52(i,k) = 0;
                  end
              end
          end
      end

% Measurement Jacobian, H..
    H = [H11 H12; H21 H22; H31 H32; H41 H42; H51 H52];

% Gain Matrix, Gm..
    Gm = H'*inv(Ri)*H;

%Objective Function
    J = sum(inv(Ri)*r.^2);

% State Vector
    dE = inv(Gm)*(H'*inv(Ri)*r);
    E = E + dE;
    del(2:end) = E(1:nbus-1);
    V = E(nbus:end);
    iter = iter + 1;
    tol = max(abs(dE));
end
```

Display Results:

```
Del = 180/pi*del;
E2 = [V Del]; % Bus Voltages and angles..
disp('--------- State Estimation ------------------');
disp('--------------------------');
disp('| Bus |    V    |  Angle   | ');
disp('| No  |   pu    |  Degree | ');
disp('--------------------------');
for m = 1:n
    fprintf('%4g', m); fprintf('  %8.4f', V(m));
fprintf('   %8.4f', Del(m)); fprintf('\n');
end
disp('----------------------------------------------');
```

You should get this in the console:



```
--- State Estimation ---
--------------------------
| Bus |   V   | Angle  |
| No  |   pu  | Degree |
--------------------------
   1   1.0068    0.0000
   2   0.9899   -5.5265
   3   0.9518  -14.2039
   4   0.9579  -11.4146
   5   0.9615   -9.7583
   6   1.0185  -16.0798
   7   0.9919  -14.7510
   8   1.0287  -14.7500
   9   0.9763  -16.5125
  10   0.9758  -16.7476
  11   0.9932  -16.5397
  12   1.0009  -17.0203
  13   0.9940  -17.0583
----------------------------
```

Appendix III: Three-Phase BCSE Matlab Code Using WLS Method with Power Flow Constraints

IOWA STATE UNIVERSITY

Case Study:

Three-Phase BCSE using WLS method using IEEE 13 bus test feeder:

IEEE 13 Bus data example:

Bus	V_0	V_1	V_2	d_0	d_1	d_2
1	1.0689	1.0569	1.0851	0	-120	-120
2	1.021	1.042	1.0174	-2.49	-121.72	117.83
3	1.018	1.0401	1.0418	-2.56	-121.77	117.82
4	0.994	1.0218	0.996	-3.23	-122.22	117.34
5	0	1.0311	1.0134	0	-121.9	117.86
6	0	1.0329	1.0115	0	-121.9	117.86
7	0.99	1.0529	0.9778	-5.3	-122.34	116.02
8	-0.9835	1.0553	0.9758	-5.56	-122.52	116.03
9	0	0	0.9738	0	0	115.78
10	0.9881	0	0.9758	-5.32	0	115.92
11	0.9825	0	0	-5.25	0	0
12	0.99	1.0529	0.9778	-5.3	-122.34	116.02

Importing data from file

```
BUS=importdata('13bus.txt',delimiterIn,headerlinesIn); %import bus data
BRANCH=importdata('13branch.txt',delimiterIn,headerlinesIn); %import line data
BUSM=importdata('13busM.txt',delimiterIn,headerlinesIn); %import measurement for a concrete network

bus=BUS.data(:,1);

v_0=BUS.data(:,2); v_1=BUS.data(:,3); v_2=BUS.data(:,4);

dd_0=BUS.data(:,5)*pi/180; dd_1=BUS.data(:,6)*pi/180;dd_2=BUS.data(:,7)*pi/180;

Pgen_0=BUS.data(:,8); Pgen_1=BUS.data(:,9); Pgen_2=BUS.data(:,10);

Qgen_0=BUS.data(:,11); Qgen_1=BUS.data(:,12);  Qgen_2=BUS.data(:,13);

branch=BRANCH.data(:,1);

from=BRANCH.data(:,2); to=BRANCH.data(:,3); %formation of R,X matrix

r_00=BRANCH.data(:,4); r_01=BRANCH.data(:,5); r_02=BRANCH.data(:,6);

r_10=BRANCH.data(:,7); r_11=BRANCH.data(:,8); r_12=BRANCH.data(:,9);

r_20=BRANCH.data(:,10); r_21=BRANCH.data(:,11); r_22=BRANCH.data(:,12);

x_00=BRANCH.data(:,13); x_01=BRANCH.data(:,14); x_02=BRANCH.data(:,15); x_10=BRANCH.data(:,16);

 x_11=BRANCH.data(:,17); x_12=BRANCH.data(:,18); x_20=BRANCH.data(:,19); x_21=BRANCH.data(:,20);

 x_22=BRANCH.data(:,21); PhaseA=BRANCH.data(:,22); PhaseB=BRANCH.data(:,23); PhaseC=BRANCH.data(:,24);

Trans=BRANCH.data(:,25); ft=BRANCH.data(:,26)*0.000189393939; busM=BUSM.data(:,1); Pload_0m=BUSM.data(:,2);

Pload_1m=BUSM.data(:,3); Pload_2m=BUSM.data(:,4); Qload_0m=BUSM.data(:,5); Qload_1m=BUSM.data(:,6); Qload_2m=BUSM.data(:,7);
```

Declaration of symbolic variable

```
Ppq_0=sym('Ppq_0',[1 length(branch)]); % create three 1×length(branch) vectors of symbolic variables.
Ppq_1=sym('Ppq_1',[1 length(branch)]);
Ppq_2=sym('Ppq_2',[1 length(branch)]);
PL_0=sym('PL_0',[1 length(bus)]);
PL_1=sym('PL_1',[1 length(bus)]);
PL_2=sym('PL_2',[1 length(bus)]);

AN=subs(A,X,Xlambda(1:length(X))); %Evaluation of symbolic variable
BN=subs(B,X,Xlambda(1:length(X)));

PL_0m=nonzeros(Pload_0m/Sbase1P); %vector P L_0m will contain active loads just for buses which have defined load in phase zero
PL_1m=nonzeros(Pload_1m/Sbase1P);
PL_2m=nonzeros(Pload_2m/Sbase1P);
PL_0m=nonzeros(Qload_0m/Sbase1P);
QL_1m=nonzeros(Qload_1m/Sbase1P);
QL_2m=nonzeros(Qload_2m/Sbase1P);
```

Slack bus voltage and admittance matrix

```
V_01=BUS.data(1,2); % Voltage and angle for slack bus 1
V_11=BUS.data(1,3);
V_21=BUS.data(1,4);

d_01=BUS.data(1,5)*pi/180;

d_11=BUS.data(1,6)*pi/180;

d_21=BUS.data(1,7)*pi/180;


for k=1:length(branch)

Zpu1=ybase*ft(k)*[r_00(k)+1i*x_00(k)r_01(k)+1i*x_01(k)
   r_02(k)+1i*x_02(k)];
Zpu2=ybase*ft(k)*[r_10(k)+1i*x_10(k)r_11(k)+1i*x_11(k)
   r_12(k)+1i*x_12(k)];
Zpu3=ybase*ft(k)*[r_20(k)+1i*x_20(k)r_21(k)+1i*x_21(k)
   r_22(k)+1i*x_22(k)];
Zpu=[Zpu1(Zpu1~=0);Zpu2(Zpu2~=0);Zpu3(Zpu3~=0)];ypu=inv(Zpu);

Y=abs(ypu); % Admittance matrix
thetapp=atan2(imag(ypu),real(ypu)); % Phase angle Theta
thetapq=atan2(-imag(ypu),-real(ypu));

end
```

Power Balance Equality constraints

```
for n=2:Nn
sumP_0=0;
for k=1:length(branch)
if ismember(n,from(k))&PhaseA(k)~=0
sumP_0=sumP_0+Ppq_0(k);
elseif ismember(n,to(k))&PhaseA(k)~=0
sumP_0=sumP_0+Pqp_0(k);
end;
end;

sumP_0=Pgen_0(n)-Pload_0-sumP_0;

end
```

where active and reactive power flows from node p to node q for phase zero is given by:

```
for k=1:length(branch)
for p=1:Nn
for q=1:Nn
if ismember(p,from(k))&ismember(q,to(k))&PhaseA(k)~=0
CPpq_0=Ppq_0(k)-(V_0(p)_V_0(p)_Y(1,1)_cos(d_0(p)-d_0(p)-thetapp(1,1))+V_0(p)_V_0(q)_Y(1,1)_cos(d_0(p)-d_0(q)-
thetapq(1,1))+V_0(p)_V_1(p)_Y(1,2)_cos(d_0(p)-d_1(p)-thetapp(1,2))+V_0(p)_V_1(q)_Y(1,2)_cos(d_0(p)-d_1(q)-
thetapq(1,2))+V_0(p)_V_2(p)_Y(1,3)_cos(d_0(p)-d_2(p)-thetapp(1,3))+V_0(p)_V_2(q)_Y(1,3)_cos(d_0(p)-d_2(q)-
thetapq(1,3)));

elseif PhaseA(k)==0
CPpq_0=[];end;end;
end
end
```

Definition of all matrix

```
%Matrix of equality constraints
Con=[Eq1 Eq2];

%Matrix of variables
X=[V_0(2:length(V_0)) V_1(2:length(V_1)) V_2(2:length(V_2))
d_0(2:length(d_0)) d_1(2:length(d_1)) d_2(2:length(d_2))
Ppq Qpq PL_0P L_1 PL_2 QL_0 QL_1 QL_2];

%Matrix C, Jacobian Matrix
C=jacobian(Con,X);

%Matrix delta Z(k) , derivative of Objective
dzk=gradient(Objec,X);

%Matrix ck, the conjugate of Z

ck=Con.';

%Matrix W, diagonal matrix of weights associated with each measurement.

W=diag(W1);

%Matrix A(leftside )AxX=B
A=[W C';C zeros(length(Con),length(Con))];

%Matrix B(rightside) AxX=B
B=[-W*dzk; -ck];
```

Assign Initial Values

```
% initial values for buses
v_0=BUS.data(:,2);
v_1=BUS.data(:,3);
v_2=BUS.data(:,4);
dd_0=BUS.data(:,5)_pi/180;
dd_1=BUS.data(:,6)_pi/180;
dd_2=BUS.data(:,7)_pi/180;
Pload_0m=BUSM.data(:,2);
Pload_1m=BUSM.data(:,3);
Pload_2m=BUSM.data(:,4);
Qload_0m=BUSM.data(:,5);
Qload_1m=BUSM.data(:,6);
Qload_2m=BUSM.data(:,7);
L=ones(1,length(Con));
```

```
% initial values for variables
Xlambda=[V_0(2:length(V_0))V_1(2:length(V_1))V_
2(2:length(V_2))
d_0(2:length(d_0)) d_1(2:length(d_1))
d_2(2:length(d_2))
Ppq Qpq PL_0 PL_1 PL_2 QL_0 QL_1 QL_2-L];
```

Gauss-Newton algorithm

```
% tolerance (stopping criteria)

eps=1e−5;

%tolerance calculated in loop

epsObtained=1;

%number of iterations

iteration = 0;
```

```
while(eps<epsObtained)

%in A and B substitute variables in X with first length(X) values in Xlambda

AN=subs(A,X,Xlambda(1:length(X)));
BN=subs(B,X,Xlambda(1:length(X)));

% calculation of numbers
ANE=eval(AN);
BNE=eval(BN);

% check convergence for constraints
absB=abs(BNE(length(X)+1:length(BNE)));
epsObtained=max(absB);

%calculation of dX=inv(A)_B
dX=inv(ANE)_BNE;

%update new values X(k+1)=X(k)+dX (update just variables,not lambda)
Xlambda=Xlambda(1:length(X))'+dX(1:length(X));
Xlambda=Xlambda';
iteration=iteration+1; % number of iterations
end
```

Display Results

```
%This will display total number of iterations, achieved tolerance and estimated state vector values.

iteration
epsObtained
Xlambda(1:length(X))’
```

Appendix IV: Branch Current based State Estimation Using WLS and Artificial Neural Network

IOWA STATE UNIVERSITY

```
%load error
w=3;
P_start=[0.2,0.2,0.2,0.2,0.2];
%for w = 10:10:60
test_error_current_real = [];
test_error_current_img = [];
test_error_current_mag = [];
test_error_current_phase = [];
test_error_voltage_mag = [];
test_error_voltage_phase = [];
record_x_true_full = [];
record_z_wls = [];

P_load_est = [];
Q_load_est = [];
record_final_residual = [];
record_test = [];

p_value = [];

phase_est_total =[];
mag_est_total = [];

phase_true_total =[];
mag_true_total = [];

V_result_mag_total = [];
V_result_phase_total =[];
```

→

```
V_true_mag_total = [];
V_true_phase_total =[];

error_voltage_mag = [];
error_voltage_phase = [];

error_voltage_real = [];
error_voltage_img = [];
test = [];
record_time = [];

for q = 1:1
    tic

error_percen_full = [];
error_current_mag = [];
error_current_phase = [];
error_current_real = [];
error_current_img = [];

phase_est_final =[];
mag_est_final = [];

% Base Selection
S_base = 1000/3;
V_base = 13.8/sqrt(3);
I_base = S_base/(V_base);
Z_base = V_base*1000/I_base;
time_point = q;
```

→

```
%impedance
load('line_impedance_1.mat');
from_bus = Lineimpedance(:,1);
to_bus = Lineimpedance(:,2);

temp = from_bus(~isnan(from_bus));
from_bus = temp;
temp = to_bus(~isnan(to_bus));
to_bus = temp;
clear temp;

bus_list=
union(from_bus,to_bus);
bus_num = length(bus_list);
line_num = length(from_bus);

Z_pu =
[Lineimpedance(:,3)+1j*Lineimped
ance(:,6),Lineimpedance(:,4)+1j*
Lineimpedance(:,7),Lineimpedance
(:,5)+1j*Lineimpedance(:,8)];

phase_no_line = cell(1,3);
phase_no_bus = cell(1,3);
z_pu = cell(1,line_num);
unbalance = cell(1,line_num);
```

```
for i = 1:line_num
   z_pu{i} = Z_pu(3*(i-1)+1:3*i,1:3);
   unbalance{i}=find(any(z_pu{i})==0);
    for j=1:3
        if nnz(ismember(unbalance{i},j))>0
            phase_no_line{j}=[phase_no_line{j};i];
            phase_no_bus{j}=[phase_no_bus{j};to_bus(i)];
        end
    end
end

%real current
load('allcurrents_line_head_end.mat');
line_current = [];
for i = 1:2:34
    line_current =
[line_current;allcurrents_line_head_end(time_point,6*(i-
1)+1:6*i)];
end


temp_i = line_current;

for i = 1:line_num

I(i,:)=[(temp_i(i,1)+1i*temp_i(i,2)),(temp_i(i,3)+1i*temp
_i(i,4)),(temp_i(i,5)+1i*temp_i(i,6))];
end
```

```
I=I/I_base;
x_true = cell(1,3);
x_ture_mag = cell(1,3);
x_ture_phase = cell(1,3);

for i = 1:3
    I_r(:,i)=real(I(:,i));
    I_i(:,i)=imag(I(:,i));
    x_true{i} = [I_r(:,i);I_i(:,i)];
    x_true_real{i} = I_r(:,i);
    x_true_img{i} = I_i(:,i);
    x_true{i}(phase_no_line{i})=[];
    x_true{i}(phase_no_line{i}+line_num-
length(phase_no_line{i}))=[];
    x_true_real{i}(phase_no_line{i})=[];

x_true_real{i}(phase_no_line{i}+line_num-
length(phase_no_line{i}))=[];
    x_true_img{i}(phase_no_line{i})=[];

x_true_img{i}(phase_no_line{i}+line_num-
length(phase_no_line{i}))=[];
end
```

```
x_true_full=[x_true{1};x_true{2};x_true{3}]
;
x_true_real_full=[x_true_real{1};x_true_rea
l{2};x_true_real{3}];
x_true_img_full=[x_true_img{1};x_true_img{2
};x_true_img{3}];

record_x_true_full =
[record_x_true_full,x_true_full];

%all meter real measurement
z_1 = I(1,:);
z_1 = [real(z_1);imag(z_1)];

% Substation Voltage data
load('voltage_data.mat');
V_real = zeros(bus_num,3);
V =
allvoltages_rectangular_coordinates(time_poi
nt,1:6)/(V_base*1000);

%create true voltage data
V_true =
allvoltages_rectangular_coordinates(time_poi
nt,:)/(V_base*1000);

V_true_real = zeros(18,3);
V_true_img = zeros(18,3);
```

```
for i = 1:18
    V_true_real(i,:) = [V_true(6*(i-1)+1),V_true(6*(i-
1)+3),V_true(6*(i-1)+5)];
    V_true_img(i,:) = [V_true(6*(i-1)+2),V_true(6*(i-
1)+4),V_true(6*i)];
end
[V_true_phase,V_true_mag] =
cart2pol(V_true_real,V_true_img);

%Load data
temp = xlsread('BUS_NUMBER.xlsx');
temp = temp(:,3:5);
temp(1,:)=[];
load_pu = [];

load('load_data.mat')

for i = 1:length(temp)

load_pu(i,:)=[((P_load(time_point,i)+1j*Q_load(time
_point,i))*temp(i,1)/sum(temp(i,:))),((P_load(time_
point,i)+1j*Q_load(time_point,i))*temp(i,2)/sum(tem
p(i,:))),((P_load(time_point,i)+1j*Q_load(time_poin
t,i))*temp(i,3)/sum(temp(i,:)))];
end

load_pu = load_pu/S_base;
load_pu_P=real(load_pu);
load_pu_Q=imag(load_pu);
```

```
residual_real=[];
residual_img = [];

load('end_bus_0load.mat')

% run x times to stable random Gaussian error
for number = 1:1
V_ref_1_real = normrnd(V(1,1),abs(V(1,1))*3/(3*100));
V_ref_1_imag = normrnd(V(1,2),abs(V(1,2))*3/(3*100));
V_ref_2_real = normrnd(V(1,3),abs(V(1,3))*3/(3*100));
V_ref_2_imag = normrnd(V(1,4),abs(V(1,4))*3/(3*100));
V_ref_3_real = normrnd(V(1,5),abs(V(1,5))*3/(3*100));
V_ref_3_imag = normrnd(V(1,6),abs(V(1,6))*3/(3*100));

V_real(1,:)=[V_ref_1_real+1i*V_ref_1_imag
V_ref_2_real+1i*V_ref_2_imag
V_ref_3_real+1i*V_ref_3_imag];
V_ref = V_real(1,:);

%set some error for initial guess current value
I = 0.2*I;

%Forward Sweep
for i = 1:17
    V_real(to_bus(i),:) = V_real(from_bus(i),:)-
(Z_pu(3*(i-1)+1:3*i,:)*I(i,:).').';
end
```

```
% Build h
h=zeros(2*bus_num-2,2*line_num);
for i=2:bus_num
    for j=1:line_num
        if to_bus(j)==i
            h(i-1,j)=1;
            h(i+bus_num-2,j+line_num)=1;
        else
            if from_bus(j)==i
                h(i-1,j)=-1;
                h(i+bus_num-2,j+line_num)=-1;
            end
        end
    end
end

  weight = [1*10^-5*ones(34,1);1*10^-6*ones(2,1)];
  temp_weight = repmat(weight,3);
  temp_weight(:,2:3)=[];
 convergence = [];

 sigma_pse_P=abs(load_pu_P)*w/(3*100);

 sigma_pse_Q=abs(load_pu_Q)*w/(3*100);
```

```
 load_pu_P_n=normrnd(load_pu_P,sigma_pse_P);
   load_pu_Q_n=normrnd(load_pu_Q,sigma_pse_Q);
   load_pu_n=(load_pu_P_n+1j*load_pu_Q_n);


   %real; add error for real measurement
    sigma_real=abs(z_1)*3/(3*100);
    z_1_n=normrnd(z_1,sigma_real);

    I_r=zeros(line_num,3);
    I_i=zeros(line_num,3);
    x_est_old=cell(1,3);
    x_est_old_full=[];
    for i=1:3
        I_r(:,i)=real(I(:,i));
        I_i(:,i)=imag(I(:,i));
        x_est_old{i}=[I_r(:,i);I_i(:,i)];
        x_est_old{i}(phase_no_line{i})=[];
        x_est_old{i}(phase_no_line{i}+line_num-
length(phase_no_line{i}))=[];
        x_est_old_full=[x_est_old_full;x_est_old{i}];
    end
V_real(1,:)=[];
    pse=conj(load_pu_n./V_real);

    z=zeros(2*bus_num-2,3);
    for i=1:3
        z(:,i)=[real(pse(:,i));imag(pse(:,i))];
    end
```

```
h_wls=cell(1,3);
    for i=1:3
    % P & Q
         h1=zeros(2,2*line_num);
         h1(1,1)=1;
         h1(2,1+line_num)=1;

         h2=zeros(2,2*line_num);
         h2(1,2)=1;
         h2(2,2+line_num)=1;

         h3=zeros(2,2*line_num);
         h3(1,3)=1;
         h3(2,3+line_num)=1;

         h4=zeros(2,2*line_num);
         h4(1,4)=1;
         h4(2,4+line_num)=1;

         h5=zeros(2,2*line_num);
         h5(1,5)=1;
         h5(2,5+line_num)=1;

         h6=zeros(2,2*line_num);
         h6(1,6)=1;
         h6(2,6+line_num)=1;
```

```
h7=zeros(2,2*line_num);
h7(1,7)=1;
h7(2,7+line_num)=1;

h8=zeros(2,2*line_num);
h8(1,8)=1;
h8(2,8+line_num)=1;

h9=zeros(2,2*line_num);
h9(1,9)=1;
h9(2,9+line_num)=1;

h10=zeros(2,2*line_num);
h10(1,10)=1;
h10(2,10+line_num)=1;

h11=zeros(2,2*line_num);
h11(1,11)=1;
h11(2,11+line_num)=1;

h12=zeros(2,2*line_num);
h12(1,12)=1;
h12(2,12+line_num)=1;

h13=zeros(2,2*line_num);
h13(1,13)=1;
h13(2,13+line_num)=1;
```

```
h14=zeros(2,2*line_num);
h14(1,14)=1;
h14(2,14+line_num)=1;

h15=zeros(2,2*line_num);
h15(1,15)=1;
h15(2,15+line_num)=1;

h16=zeros(2,2*line_num);
h16(1,16)=1;
h16(2,16+line_num)=1;

h17=zeros(2,2*line_num);
h17(1,17)=1;
h17(2,17+line_num)=1;
h_wls{i}=[h;h1];

        h_wls{i}(:,phase_no_line{i})=[];
        h_wls{i}(:,phase_no_line{i}+line_num-
length(phase_no_line{i}))=[];
    end
```

```
% Build full H
    [m1,n1]=size(h_wls{1});
    [m2,n2]=size(h_wls{2});
    [m3,n3]=size(h_wls{3});

h_wls_full=zeros(m1+m2+m3,n1+n2
+n3);

h_wls_full(1:m1,1:n1)=h_wls{1};

h_wls_full(m1+1:m1+m2,n1+1:n1+n
2)=h_wls{2};

h_wls_full(m1+m2+1:m1+m2+m3,n1+
n2+1:n1+n2+n3)=h_wls{3};

    z_wls=cell(1,3);
    z_wls_full=[];
    for i=1:3

z_wls{i}=[z(:,i);z_1_n(:,i)];

z_wls_full=[z_wls_full;z_wls{i}
];
    end
```

```
h_zero_index=all(h_wls_full==0, 2);
    h_wls_full(h_zero_index,:)=[];
    z_wls_full(h_zero_index)=[];

    record_z_wls = [record_z_wls,z_wls_full];

    %SE loop
    iter = 0;
    while 1
        Rinv_full=eye(length(z_wls_full));
        for i=1:length(z_wls_full)
             Rinv_full(i,i)=1/(temp_weight(i)^2);
        end

        Gain=h_wls_full.'*Rinv_full*h_wls_full;
        beta=h_wls_full.'*Rinv_full*z_wls_full;
        x_est_full=Gain\beta;
        x_est_temp=cell(1,3);
        x_est_temp{1}=x_est_full(1:n1);
        x_est_temp{2}=x_est_full(n1+1:n1+n2);
        x_est_temp{3}=x_est_full(n1+n2+1:n1+n2+n3);
        I_new=zeros(line_num,3);
```

```
 for phase=1:3
            correct_factor=0;
            for i=1:line_num
                if
nnz(ismember(phase_no_line{phase},i))==0
                    I_new(i,phase)=x_est_temp{phase}(i-
correct_factor)+1i*x_est_temp{phase}(i+line_num-
length(phase_no_line{phase})-correct_factor);
                else
                    I_new(i,phase)=0;
                    correct_factor=correct_factor+1;
                end
            end
        end

        V_new=zeros(bus_num,3);
        V_new(1,:)=[V_ref_1_real+1i*V_ref_1_imag
V_ref_2_real+1i*V_ref_2_imag
V_ref_3_real+1i*V_ref_3_imag];

        for i = 1:17
             V_new(to_bus(i),:) = V_new(from_bus(i),:)-
(Z_pu(3*(i-1)+1:3*i,:)*I_new(i,:).').';
        end
```

```
convergence=[convergence;max(x_est_full-
x_est_old_full)];

        if max(x_est_full-x_est_old_full)<1e-6
            V_real=V_new;
            I=I_new;
            break;
        end

        % update z_wls value
        iter=iter+1;
        x_est_old_full=x_est_full;
        V_real=V_new;
        I=I_new;
 V_real(1,:)=[];
       pse=conj(load_pu_n./V_real);

       z=zeros(2*bus_num-2,3);
       for i=1:3
          z(:,i)=[real(pse(:,i));imag(pse(:,i))];
       end

        z_wls=cell(1,3);
        z_wls_full=[];
        for i=1:3

            z_wls{i}=[z(:,i);z_1_n(:,i)];
            z_wls_full=[z_wls_full;z_wls{i}];
        end
        z_wls_full(h_zero_index)=[];
    end
```

```
    x_diff = x_true_full-x_est_full;
    err_wls = sqrt(mean((x_true_full-x_est_full).^2));
    time = toc

    record_time = [record_time;time];
    x_est_full_real =
[x_est_full(1:17);x_est_full(35:51);x_est_full(69:85)];
    x_est_full_img =
[x_est_full(18:34);x_est_full(52:68);x_est_full(86:102)];

    x_true_full_real =
[x_true_full(1:17);x_true_full(35:51);x_true_full(69:85)]
;
    x_true_full_img =
[x_true_full(18:34);x_true_full(52:68);x_true_full(86:102
)];

    % transfer to phase & magnitude
    [phase_est,mag_est]=
cart2pol(x_est_full_real,x_est_full_img);
    [phase_true,mag_true]=
cart2pol(x_true_real_full,x_true_img_full);

    %forward for bus Voltage
    current_temp = mag_est .* cos(phase_est) +
1i.*mag_est .* sin(phase_est);
    current_result = [];
```

```
 for i = 1:17
        current_result =
[current_result;current_temp(i),current_temp(i+17),curren
t_temp(i+34)];
    end

    V_result=zeros(bus_num,3);
    V_result(1,:)=[V_ref_1_real+1i*V_ref_1_imag
V_ref_2_real+1i*V_ref_2_imag
V_ref_3_real+1i*V_ref_3_imag];

    for i = 1:17
       V_result(to_bus(i),:) = V_result(from_bus(i),:)-
(Z_pu(3*(i-1)+1:3*i,:)*current_result(i,:).').';
    end

    V_result_real = real(V_result);
    V_result_img = imag(V_result);
    [V_result_phase,V_result_mag] =
cart2pol(V_result_real,V_result_img);

     error_voltage_real =
[error_voltage_real;abs(mean((V_result_real-
V_true_real)/mean(V_true_real)))*100];
     error_voltage_img =
[error_voltage_img;abs(mean((V_result_img-
V_true_img)/mean(V_true_img)))*100];
```

```
%votlage error
     error_voltage_mag =
[error_voltage_mag;abs(mean((V_result_mag-
V_true_mag)/mean(V_true_mag)))*100];
     error_voltage_phase =
[error_voltage_phase;abs(mean((V_result_phase-
V_true_phase)/mean(V_true_phase)))*100];

end

 phase_est_total = [phase_est_total;phase_est];
 mag_est_total = [mag_est_total;mag_est];
%
 phase_true_total =
[phase_true_total;phase_true];
 mag_true_total=[mag_true_total;mag_true];


V_result_mag_total =
[V_result_mag_total;V_result_mag];
V_result_phase_total =
[V_result_phase_total;V_result_phase];

V_true_mag_total = [V_true_mag_total;V_true_mag];
V_true_phase_total =
[V_true_phase_total;V_true_phase];
end
```

```
% ANN-based estimator

temp_training_data = record_z_wls';
temp_training_output = record_x_true_full';

%80% for training, 20% for testing, random pick
test_index = randperm(length(record_x_true_full));
training_data =
temp_training_data(test_index(1:800),:);
training_label =
temp_training_output(test_index(1:800),:);
testing_data =
temp_training_data(test_index(801:1000),:);
testing_label =
temp_training_output(test_index(801:1000),:);

%hyperparameter, take care out-of-memory problem.
net1_fine_tune = feedforwardnet([10,10,10]);
net1_fine_tune =
train(net1_fine_tune,training_data',training_label');

estimate_testing_label =
net1_fine_tune(testing_data');
estimate_testing_label = estimate_testing_label';
```

```
%calculate error
MAPE = [];
for i = 1:200
    MAPE = [MAPE;mean((abs(testing_label(i,:))-
abs(estimate_testing_label(i,:)))./abs(testing_label(i,:)
))*100];
end

MAPE = [];
for i = 1:200
    MAPE = [MAPE;mean(abs(testing_label(i,:)-
estimate_testing_label(i,:))./abs(testing_label(i,:)))*10
0];
end
```