Three Phase Linear Tracking State Estimator

Development & Implementation

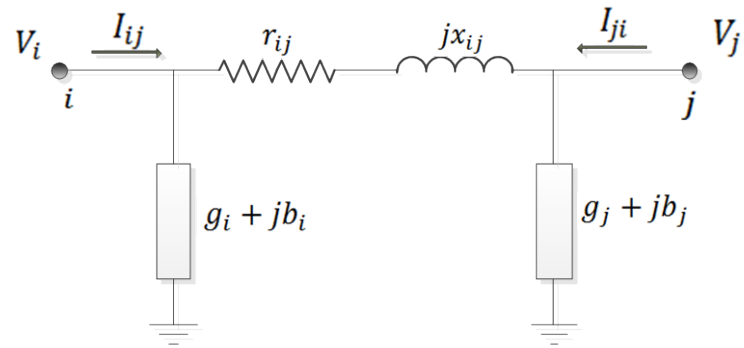
Kevin D. Jones
kdjones@vt.edu

Outline

- Phase 1
  - Linear State Estimation & Three Phase LSE
  - Topology Processing
  - Matlab Implementation
- Phase 2
  - Migration to C#/openPDC
  - Application Implementation

Linear State Estimation

- Uses PMU measurements exclusively
- Measurement set is bus voltage phasors and line current phasors
- State vector is complex
- Natural evolution of state estimation
- Eliminates the possibility of divergence, scan times



$$\begin{bmatrix} V_i \\ V_j \\ I_{ij} \\ I_{ji} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ y_{ij} + y_{i0} & -y_{ij} \\ -y_{ij} & y_{ij} + y_{j0} \end{bmatrix} \begin{bmatrix} V_i \\ V_j \end{bmatrix}$$

$$[\boldsymbol{z}] = \begin{bmatrix} \boldsymbol{E} \\ \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} \boldsymbol{II} \\ \boldsymbol{yA} + \boldsymbol{y_s} \end{bmatrix} [\boldsymbol{x}] + [\boldsymbol{e}]$$

$$[\boldsymbol{x}] = [(\boldsymbol{B^T W^{-1} B})^{-1} \boldsymbol{B^T W^{-1}}][\boldsymbol{z}] = [\boldsymbol{H}][\boldsymbol{z}]$$

Three Phase Linear State Estimation

- Small differences from positive sequence
  - Three phase impedances
  - Matrix formulation

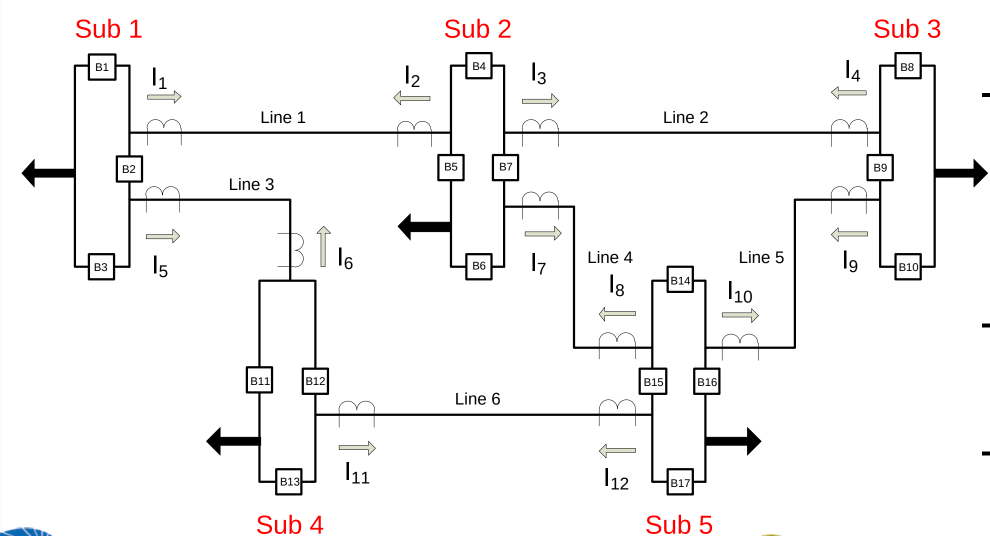
$$[\boldsymbol{Y}] = \begin{bmatrix} y_{1a} & y_{1b} & y_{1c} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ y_{1b} & y_{1d} & y_{1e} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ y_{1c} & y_{1e} & y_{1f} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & y_{3a} & y_{3b} & y_{3c} & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & y_{3b} & y_{3d} & y_{3e} & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & y_{3c} & y_{3e} & y_{3f} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & y_{6a} & y_{6b} & y_{6c} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & y_{6b} & y_{6d} & y_{6e} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & y_{6c} & y_{6e} & y_{6f} \end{bmatrix}$$

$$Z_{abc} = \begin{bmatrix} Z_{aa} & Z_{ab} & Z_{ac} \\ Z_{ab} & Z_{bb} & Z_{bc} \\ Z_{ac} & Z_{bc} & Z_{cc} \end{bmatrix}$$

$$[\boldsymbol{I}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad [\boldsymbol{0}] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Topology Processing

- SCADA information is too slow to use
- Instead, use current flow phasors; bring in breaker statuses in phasor data stream



- Outage Criteria
  - Apply logical filter to current measurement vector (flow? No flow?
  - No flow indicates potential outage
  - Outage is confirmed by breaker statuses & lookup table
  - 100% consistency required
  - System Matrix is updated

Dominion

U.S. DEPARTMENT OF ENERGY

VirginiaTech Invent the Future

Topology Processing

- Updating the System Matrix after a Contingency
  - Repopulation of system matrix & pseudo-inverse can be cumbersome
  - Method to update pseudo-inverse after line outage

$$[\hat{z}] = \left[H\left(H^T H\right)^{-1} H^T\right][z] = [Z][z]$$

$$K(3l-2:3l, 3b_l-2:3b_l) = [I]$$

$$K(3m-2:3m, 3b_m-2:3b_m) = [I]$$

$$[S] = [K^T Z K] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & S_1 & 0 & S_3 \\ 0 & 0 & 0 & 0 \\ 0 & S_3 & 0 & S_2 \end{bmatrix}$$

$$[T] = [I-S]^{-1} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & U_1 & 0 & -U_3 \\ 0 & 0 & I & 0 \\ 0 & -U_3 & 0 & U_2 \end{bmatrix}^{-1}$$

$$T_3 = \left(U_3 - U_2 U_3^{-1} U_1\right)^{-1}$$

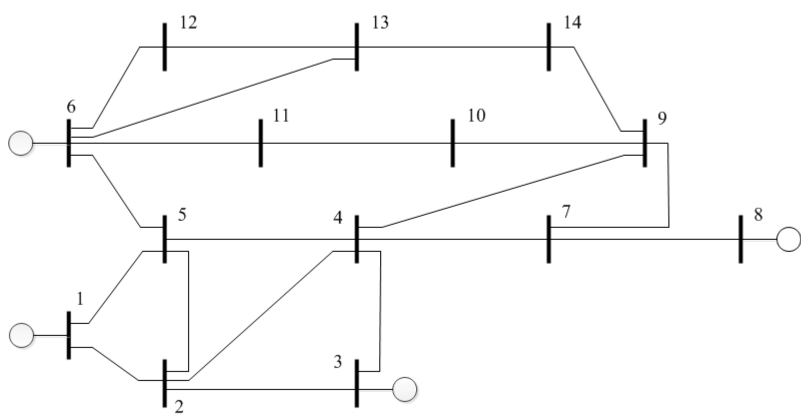
$$T_1 = T_3 U_2 (U_3)^{-1}$$

$$T_2 = T_3 U_1 (U_3)^{-1}$$

$$[M_1] = [M][I - KTK^T Z][I - KK^T]$$

Matlab Implementation

- Represents work completed during initial phase of Dominion/DOE project
- For proof-of-concept & initial testing
- How to generate three phase data?
- NDA - IEEE 14 Bus System



$$[I_{unbalanced}] = [Y_{unbalanced}][V_{balanced}]$$

$$[I_{balanced}] = [Y_{balanced}][V_{balanced}]$$

$$[V_{unbalanced}] = [Z_{unbalanced}][I_{balanced}]$$

$$[V_{final}] = \frac{([V_{balanced}] + [V_{unbalanced}])}{2}$$

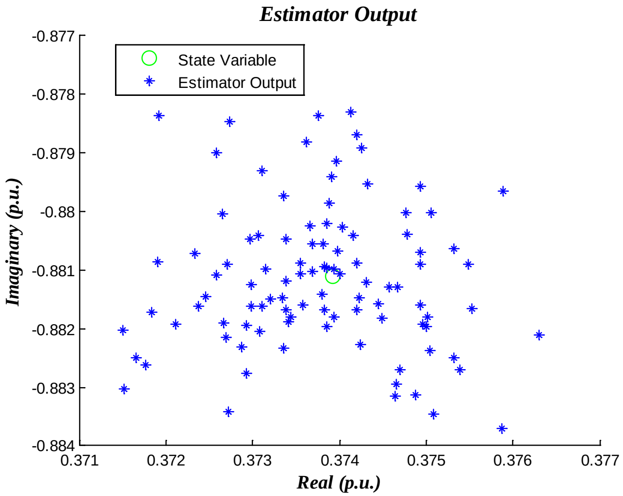
$$[I_{final}] = \frac{([I_{balanced}] + [I_{unbalanced}])}{2}$$

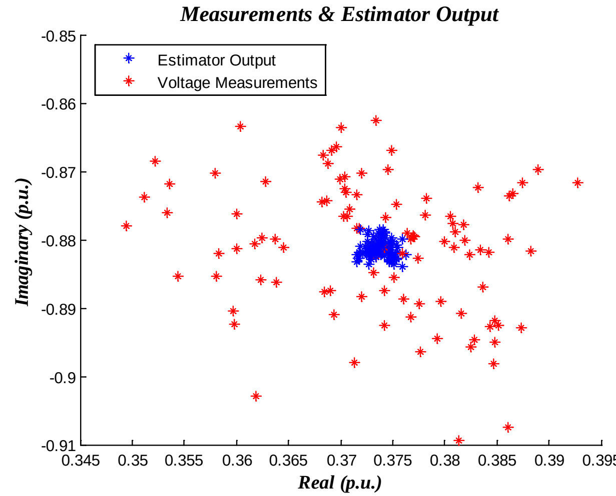
Dominion

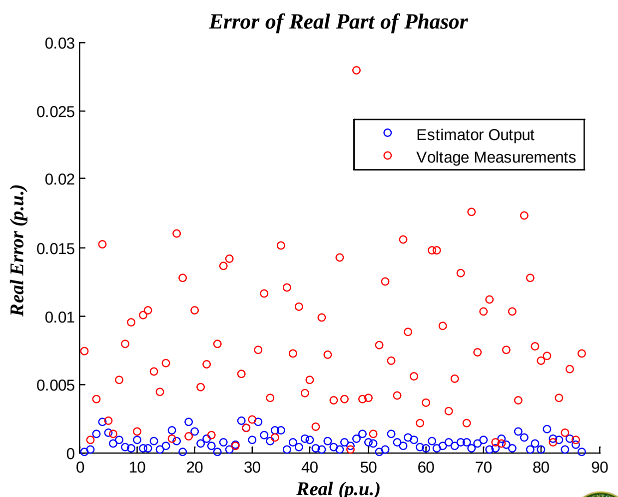
U.S. DEPARTMENT OF ENERGY

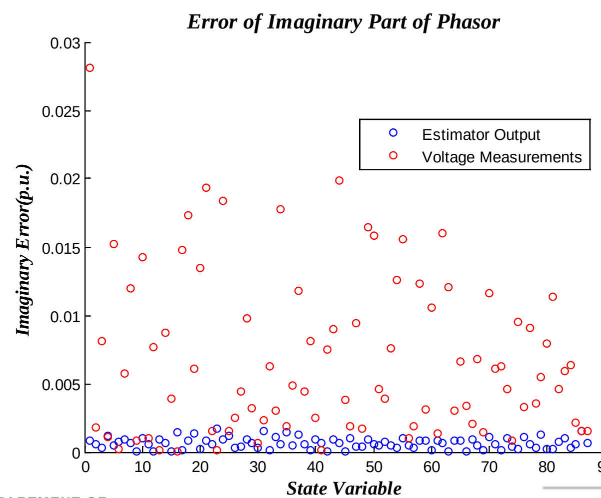
VirginiaTech Invent the Future

Matlab Results (SE)







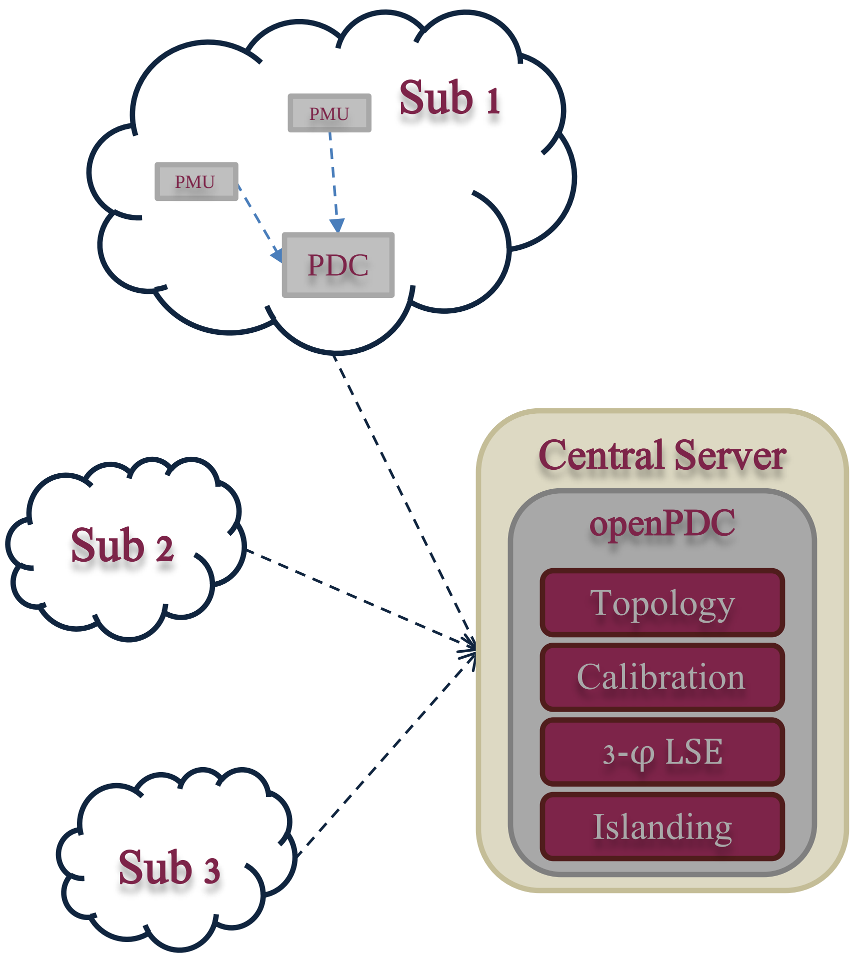


Dominion

U.S. DEPARTMENT OF ENERGY

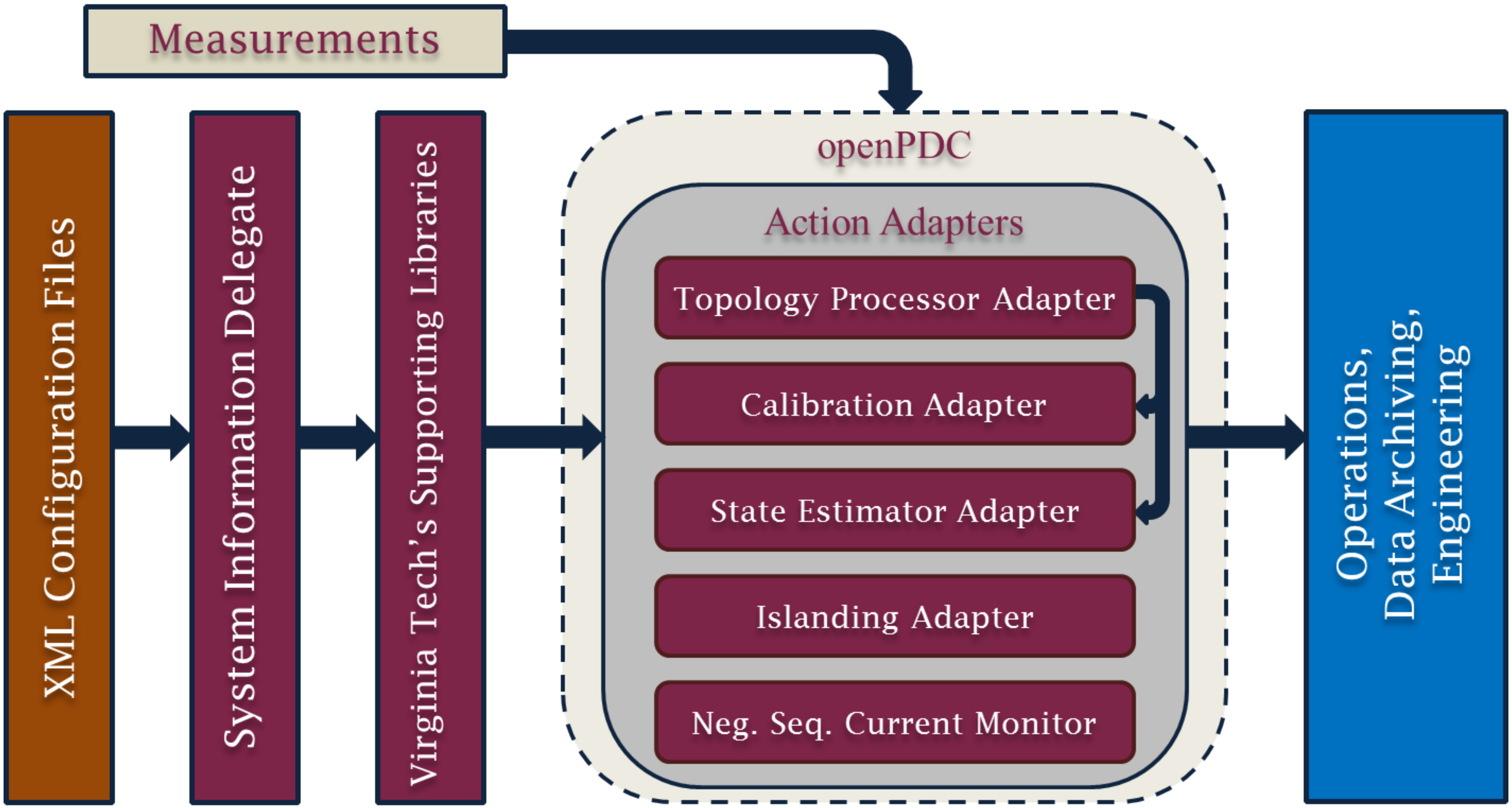
VirginiaTech Invent the Future

High Level Architecture



- Open source software PDC
- Allows for development of custom phasor concentration code
- Applications called 'Adapters', C# libraries
- Extreme Optimizations library for Linear Algebra computations
- 3-φ LSE, Topology Processor, Meter Calibration, Islanding Detection, Unbalanced Condition Monitor

Application Implementation



Dominion

U.S. DEPARTMENT OF ENERGY

VirginiaTech Invent the Future

Conclusion & Future Work

- Individual applications successfully migrated to openPDC
- Configuration files simplify changes/maintenance

- Applications to be integrated with each other and tested on realistic data set
- Process repeated on Dominion's development server

References

[1] A.G. Phadke and J. S. Thorp, *Synchronized Phasor Measurements and Their Applications*, Springer Science + Business Media, 2008.

[2] A. Abur and A. G. Exposito, *Power System State Estimation- Theory and Implementation*: CRC, 2004.

[3] A. G. Phadke, "Synchronized Phasor Measurements – A Historical Overview," *Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES*, vol. 1, pp. 476-479, 2002.

[4] A. G. Phadke, J. S. Thorp, and K. J. Karimi, "State EstimationWith Phasor Measurements," *Power Systems, IEEE Transactions on*, vol. PWRS-1, pp. 233-238, 1986.

[5] A. G. Phadke, J. S. Thorp, R. F. Nuqui, & M. Zhou,"Recent Developments in State Estimation with Phasor Measurements," *Power Systems Conference and Exposition IEEE/PES*, pp. 1-7, 2009.

[6] Synchrophasor Based Tracking Three Phase State Estimator and It's Applications, A.G. Phadke Virginia Tech, Blacksburg, VA. DOE 2010 Transmission Reliability Program Peer Review, October 19-20, 2010 at the Westin Alexandria

[7] The Open Source Phasor Data Concentrator. 2011. Grid Protection Alliance. April 2011. http://openpdc.codeplex.com