Machine Learning Applications – An Industry Perspective

23PESGM3987

2023 IEEE PES General Meeting
Orlando, FL • July 19, 2023
Marianna Vaiman, V&R Energy

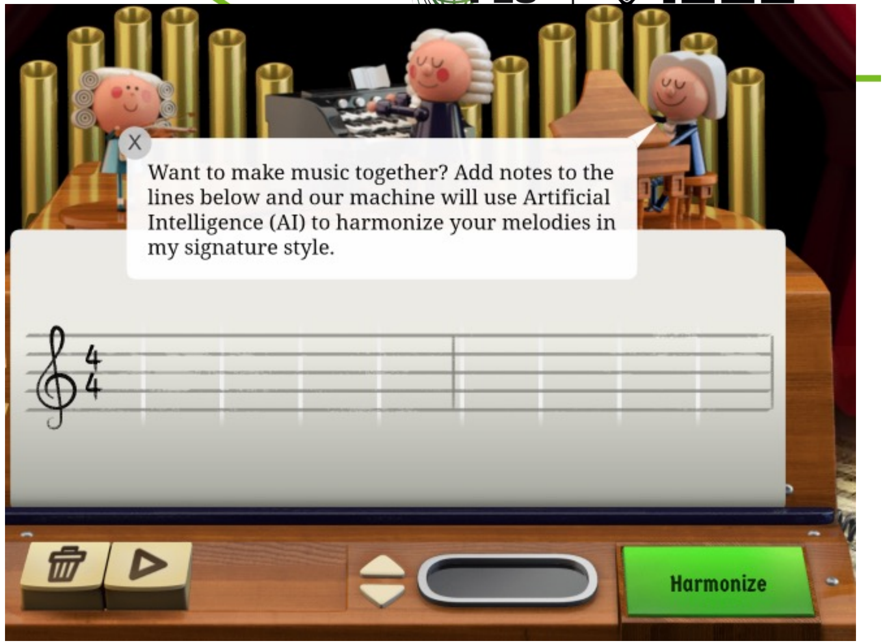
Introduction

3/22/2019

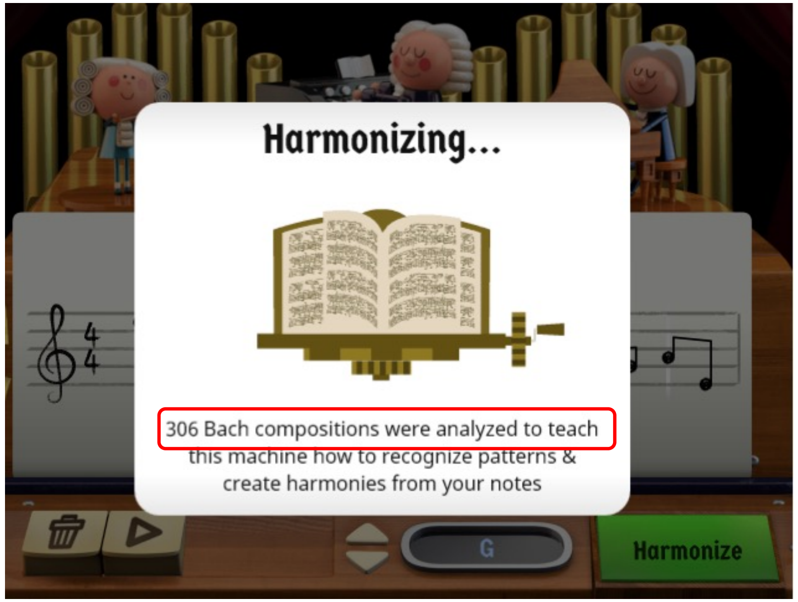
Today we celebrate Johann Sebastian Bach with our first ever **AI-powered** Doodle! The Doodle is an interactive experience encouraging players to compose a **two measure melody** of their choice. With the press of a button, the Doodle then uses **machine learning** to harmonize the custom melody into Bach's signature music style…
Next came our partners at PAIR who used **TensorFlow.js** to allow **machine learning** to happen entirely within the web browser…

1

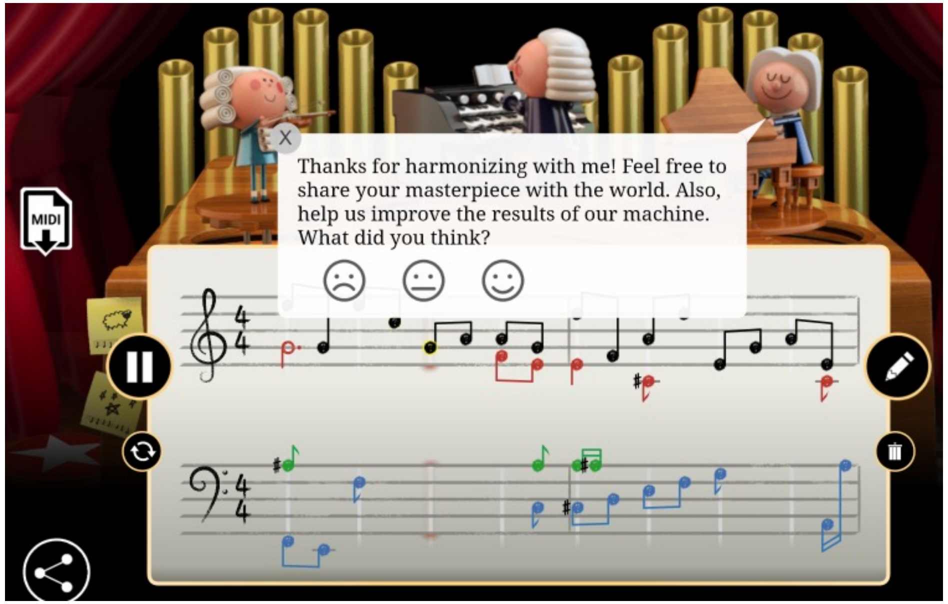
2



3



4



Took 35.42 sec to harmonize my melody

https://www.google.com/doodles/celebrating-johann-sebastian-bach

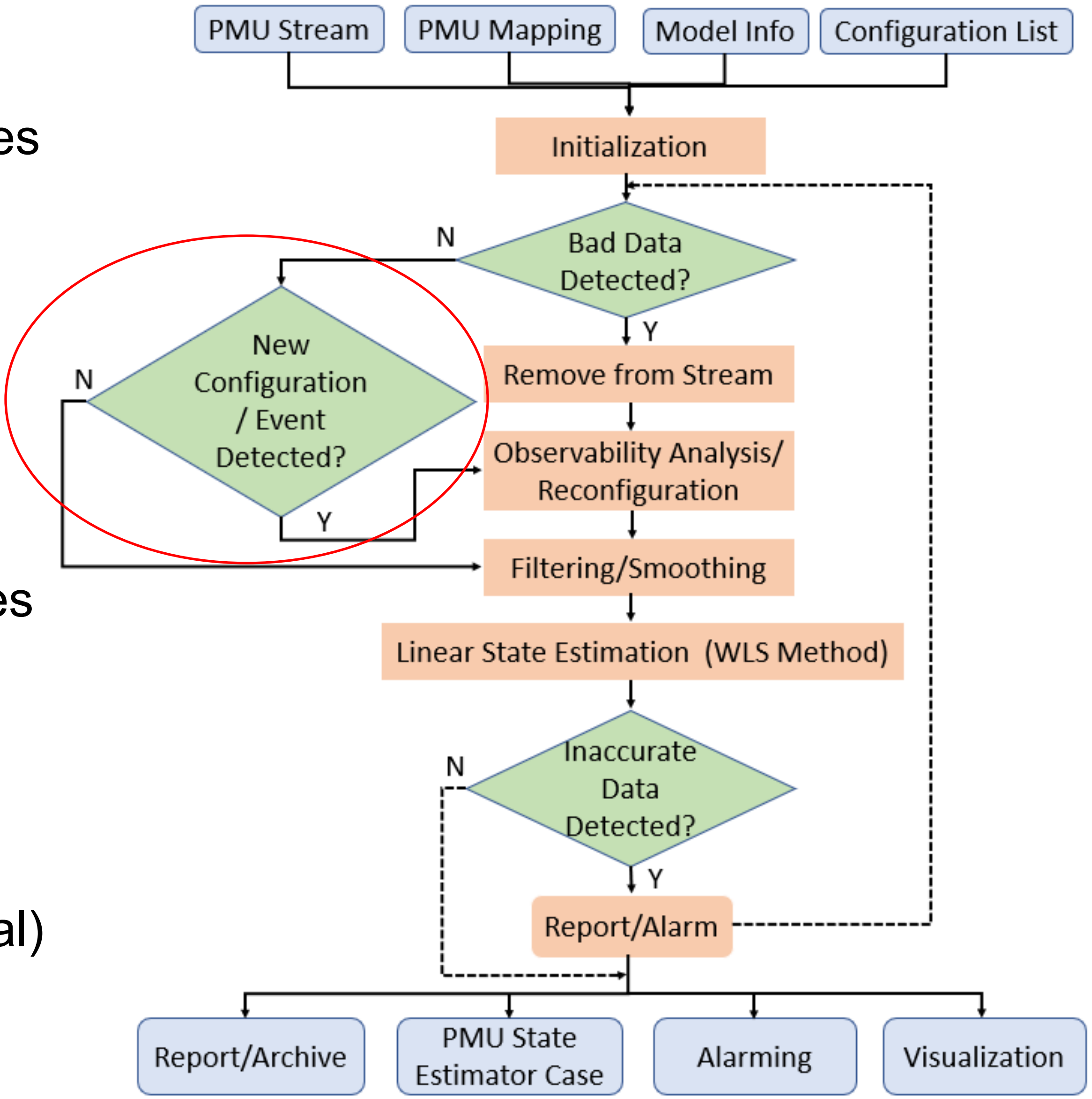
Contents

1. Distribution Linear State Estimator: Machine Learning for Topology Detection
2. Linear State Estimator: Machine Learning for Parameters of Kalman Filter

1. Machine Learning for Topology Detection in D-PMU ROSE (Distribution-PMU ROSE)

Distribution Linear State Estimator (DLSE)

- Three-phase DLSE is based on synchronized phasor measurements of voltage and current phasors, provides a direct, linear solution of system state using time-synchronized synchrophasor data only
- Performs state estimation 60 times per second
- DLSE process in D-PMU ROSE platform consists of:
    - Bad data detection, correction, alarming and reporting
    - Combination of filtering and smoothing techniques
    - Observability analysis
    - Three-phase Distribution Linear State Estimation
    - **Detection of switching events (only based on PMU data)**
    - Real-time system monitoring (voltage and thermal)
    - Alarming, visualization, archiving

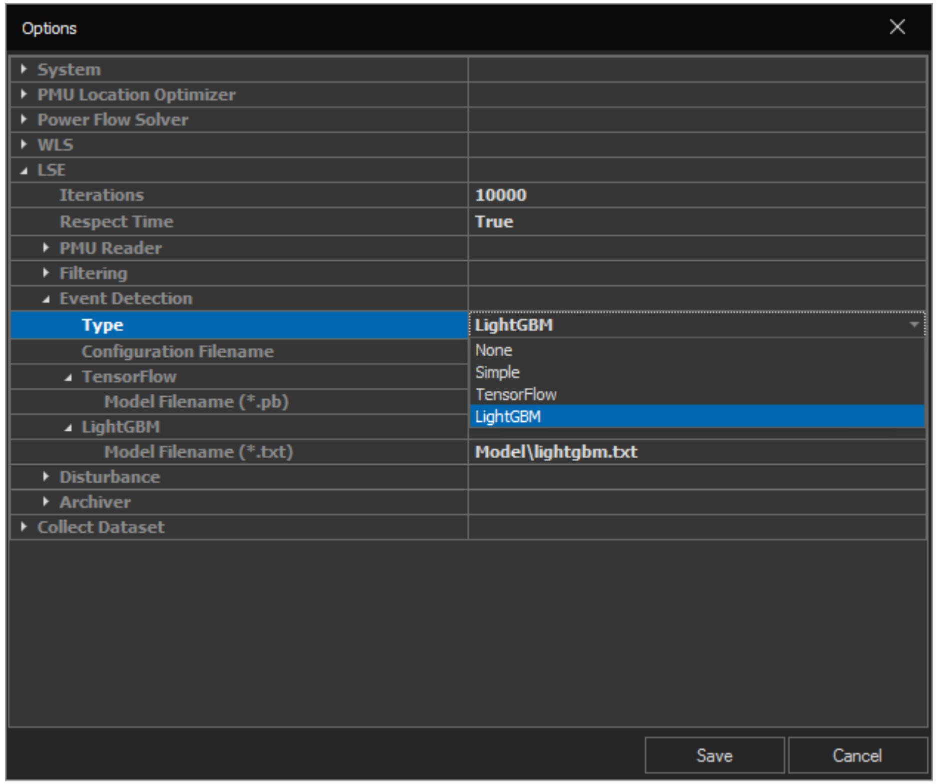


Topology (Switching Event) Detection

- Purpose of the event detection
    - Correctly identify switching events (e.g., topology change detection) and microgrid configuration without supplying switch status to the D-LSE platform
- D-LSE utilizes event detection logic which is based on computing currents using PMU measurements within the circuit
- Machine learning enables more effective/accurate event detection in real-time environment
- One of the main tasks solved by machine learning algorithms is classification of various situations.
- V&R Energy’s machine learning algorithm is used as a part of the platform to identify switching events and classify network configurations **based on PMU measurements of voltage and current only**:
    - Switch/recloser statuses are not provided as a part of PMU data or as additional data

Machine Learning Libraries in D-PMU ROSE

- Tested Microsoft's LightGBM, Google's TensorFlow, and V&R Energy's Simple
- Microsoft's LightGBM:
  - A gradient boosting method (a machine learning technique for regression and classification problems) that uses tree-based learning algorithms:
    - Faster training speed, higher efficiency
    - Lower memory usage
    - Better accuracy
    - Capable of handling large-scale data
- Google's TensorFlow:
  - A symbolic math library, and is also used for machine learning applications such as neural networks, which was developed by Google Brain
  - Uses data flow graphs; graph nodes represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) that flow between them
- All three libraries provided similar results
- Simple was selected as it gave robust solution with very limited available training data set

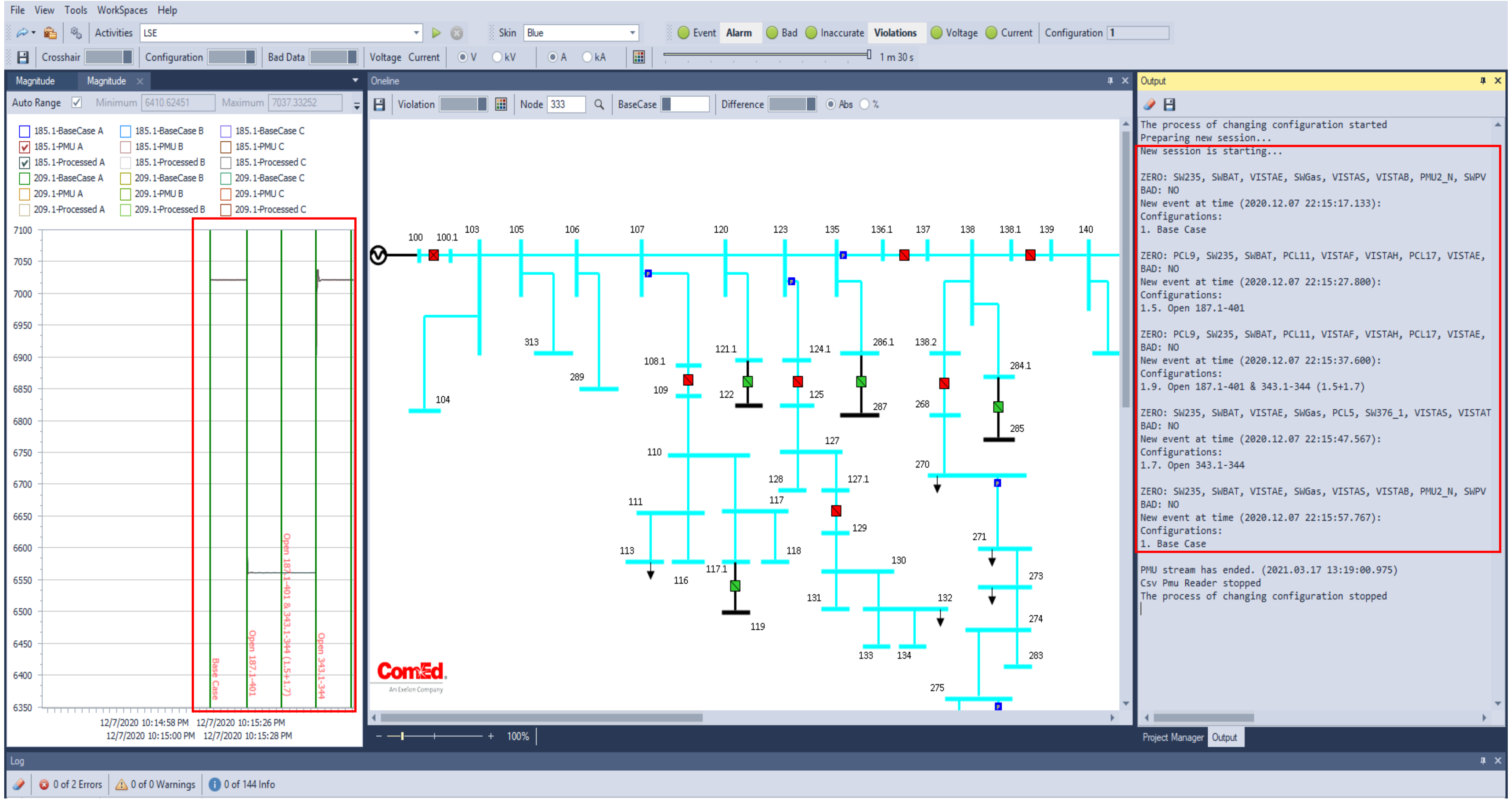


Using Machine Learning with Limited Available Data Sets

- Usually, machine learning approaches require large volume of samples to train the system - tens to hundreds of thousands (or more) of samples
- However, this volume of training data is not available in the industry
- Therefore, we used the approach that allowed us to implement a limited number of training data sets
- Extensive testing in RTDS lab showed a very high accuracy of the used technique:
    - During initial performance testing, 40 network configurations had to be identify and classified based on 10 PMU data sets. All configurations were correctly classified
    - Then, the number of tested configurations was almost doubled, and all of them were also correctly identified

Topology Detection

- Machine learning is used to improve accuracy of event detection in real-time
- Created 75+ different topology cases
- Used RTDS data for development and testing of over 50 different topologies/configurations
- The topology detection accuracy is over 90 % through varying load and system conditions

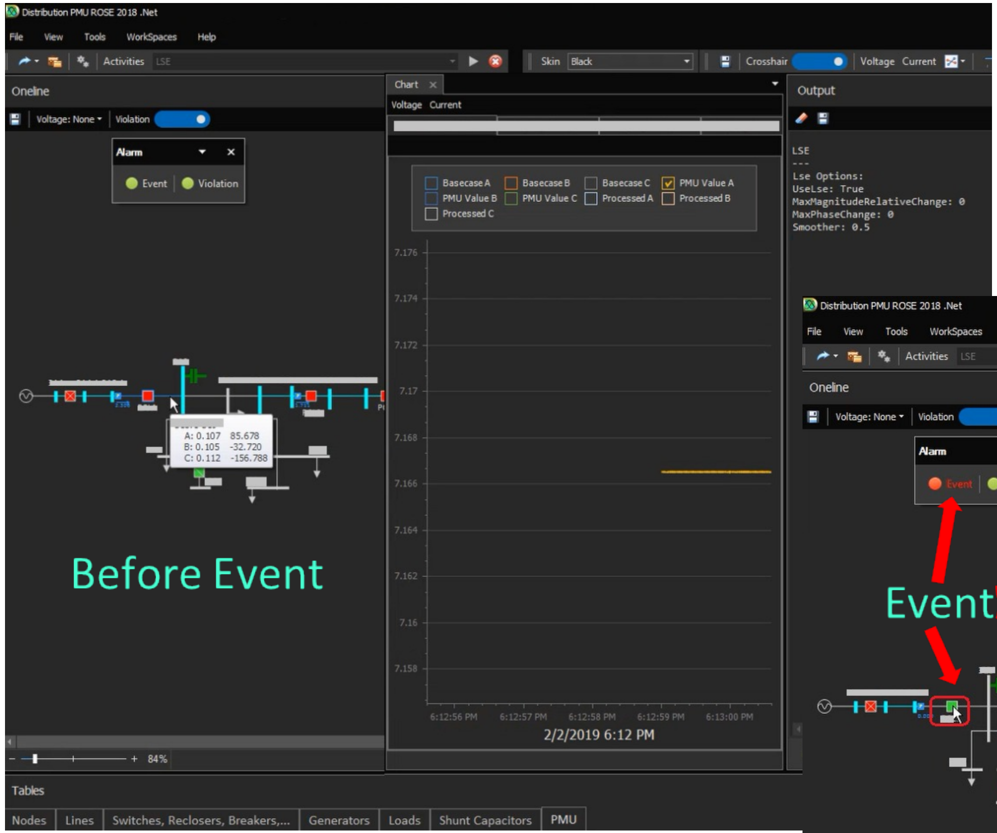


Testing Environment

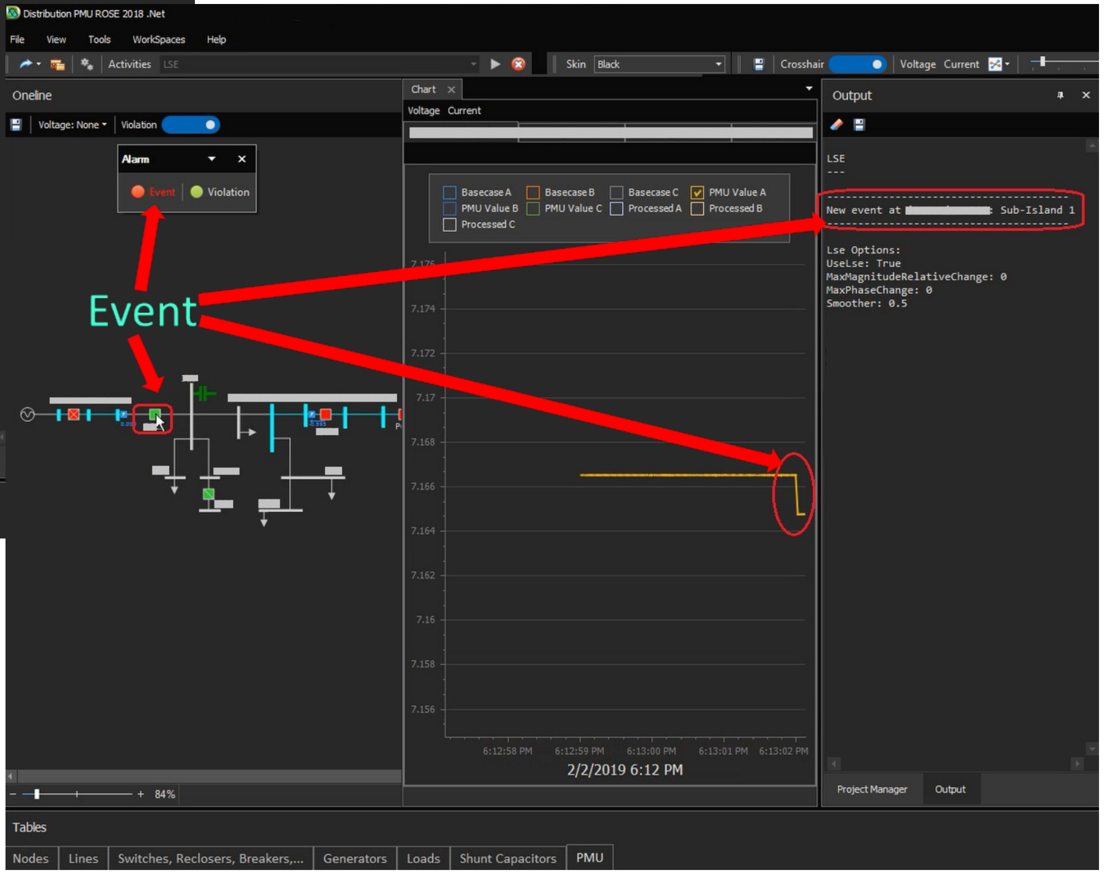
- ComEd performed extensive benchmarking in ComEd’s Grid Integration and Technology (GriT) Lab using real-time digital simulation (RTDS)
- Configurations were created in RTDS
- D-PMU ROSE tool (DLSE) didn’t know which configuration was sent, and based on PMU measurements identified the configuration
- ComEd, then, validated that the configuration identified by the tool was the same as created in RTDS
- The capability was tested and validated for BCM during an extensive period of time under various conditions

Topology Detection

- Topology detection is done using only PMU (three-phase voltages and currents)
- Switch/recloser status is not provided as a part of PMU stream or as additional data

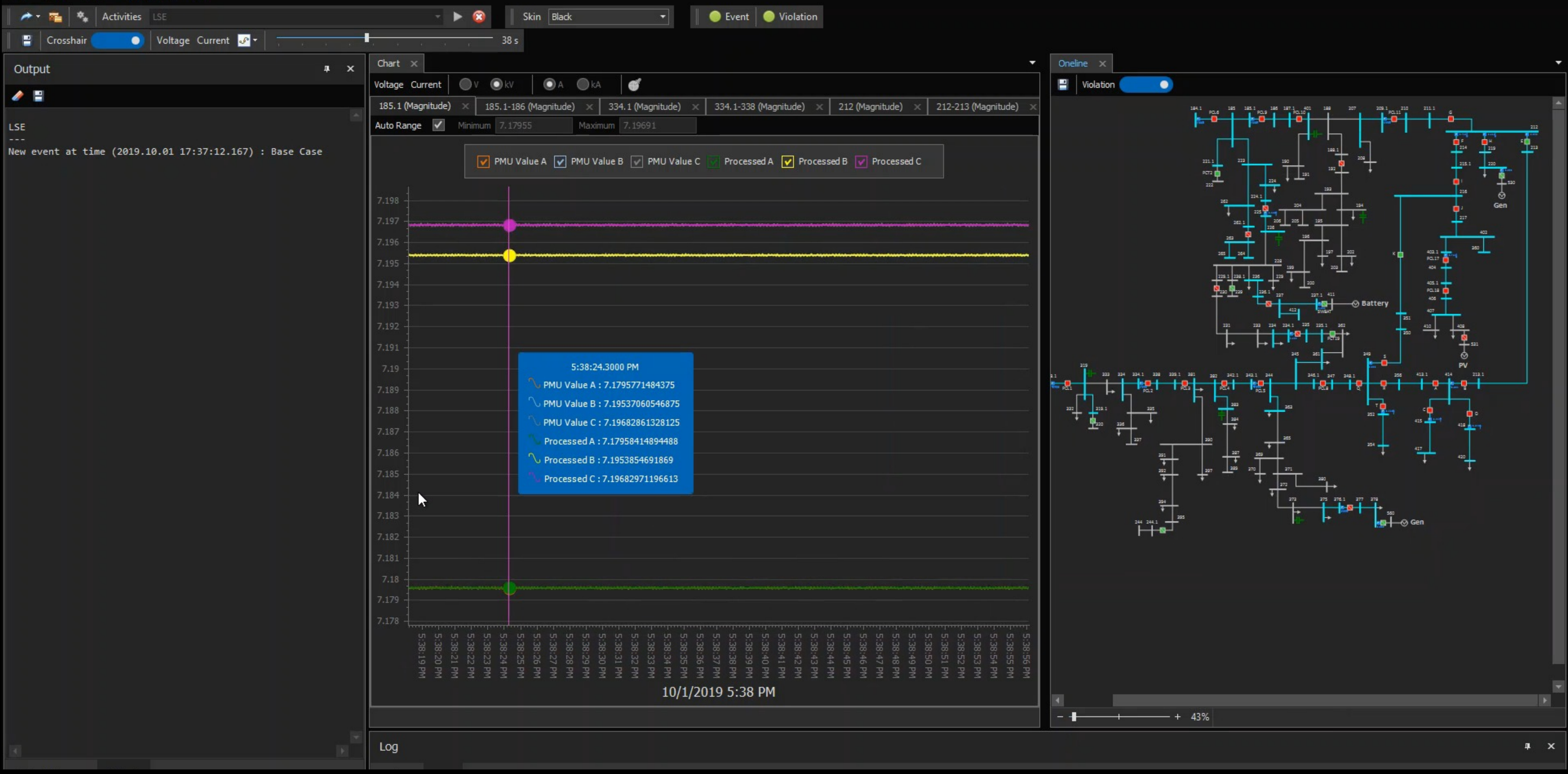


- Observability changes after a switching event has occurred



Video by Heng (Kevin) Chen, formerly with Smart Grid & Technology, ComEd

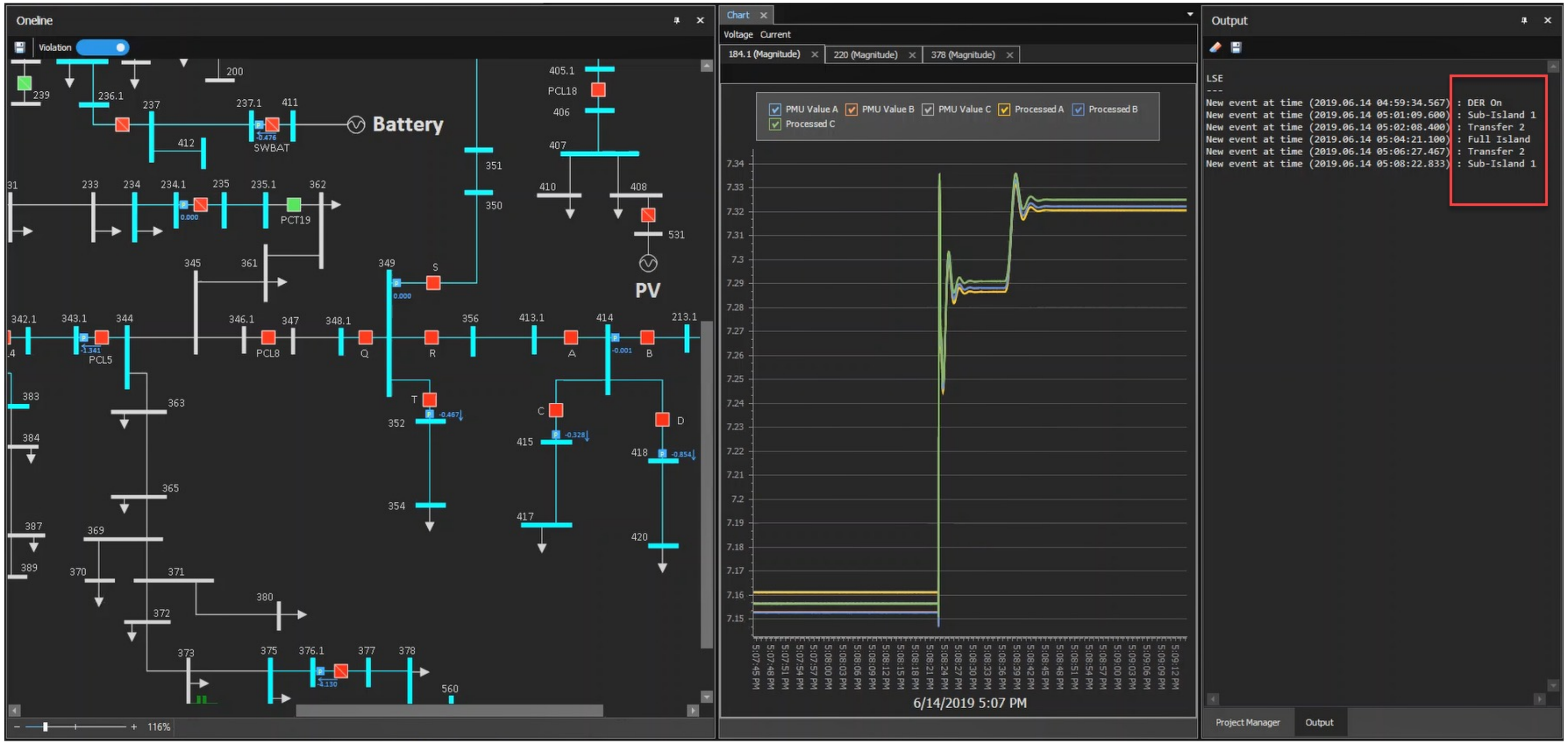
Initially presented at 2020 IEEE PES ISGT NA


Activities
LSE
Skin
Black
Event
Violation
Crosshair
Voltage
Current
38 s
Output
LSE

New event at time (2019.10.01 17:37:12.167) : Base Case
Chart
V
kV
A
kA
185.1 (Magnitude)
185.1-186 (Magnitude)
334.1 (Magnitude)
334.1-338 (Magnitude)
212 (Magnitude)
212-213 (Magnitude)
Auto Range
Minimum
7.17955
Maximum
7.19691
PMU Value A
PMU Value B
PMU Value C
Processed A
Processed B
Processed C
5:38:24.3000 PM
PMU Value A : 7.1795771484375
PMU Value B : 7.19537060546875
PMU Value C : 7.19682861328125
Processed A : 7.17958414894488
Processed B : 7.1953854691869
Processed C : 7.19682971196613
10/1/2019 5:38 PM
Oneline
Battery
Gen
PV
43%
Log

Conclusions

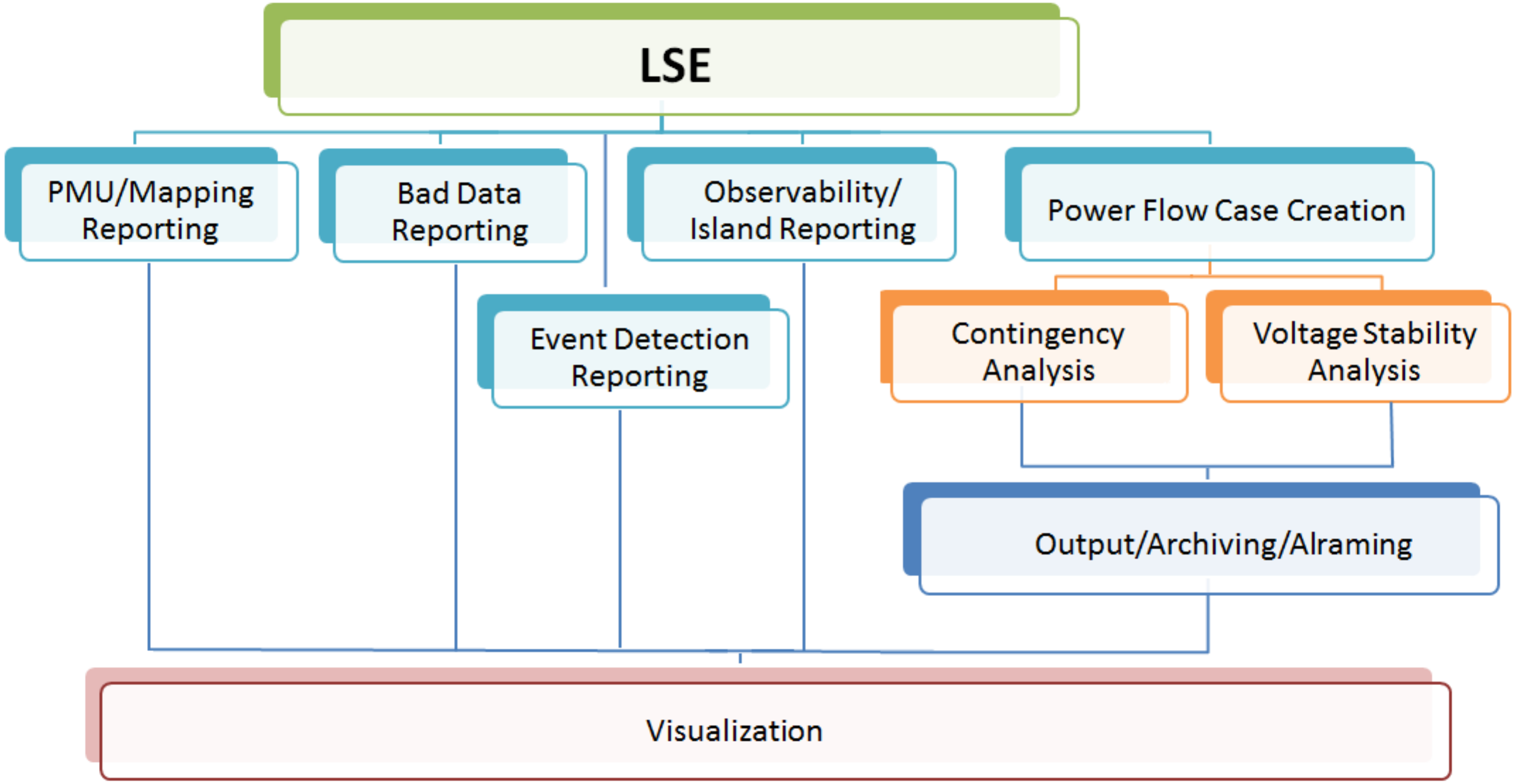
- Estimated value follow raw data during both steady state and transient conditions
- DLSE identifies defined topology changes correctly



2. Machine Learning for Parameters of Kalman Filter in PMU-ROSE

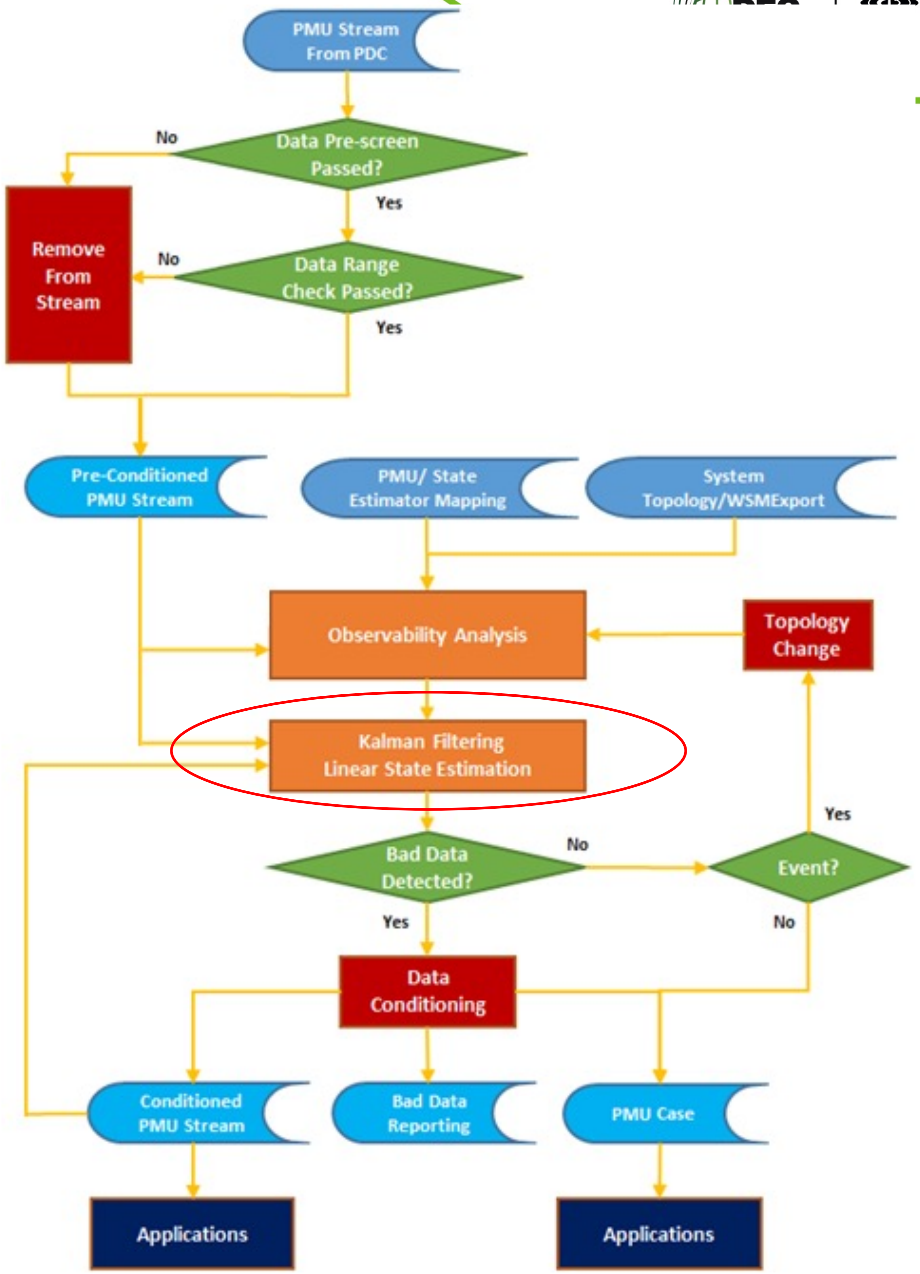
PMU ROSE Real-Time Analysis

- Output of LSE:
  - Conditioned PMU data
  - Bad data reporting and statistics
  - A list of observable islands and their details
  - PMU State Estimator Case



Components of LSE Framework

- Multi-step process:
  1. Several pre-screening techniques
  2. Data range checks
  3. Combination of filtering and smoothing techniques based on Kalman filter
  4. Linear state estimation
  5. **End-to-end machine learning**



Kalman Filtering Algorithm

- An algorithm that:
  - Uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and
  - Produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone

- The Kalman filter approach is a two-stage algorithm:
  - The first stage is **prediction**, which projects the previous time step state forward in time by means of a predefined process model
  - The second stage is **correction/estimation**, which corrects the predicted state by accounting the available measurements and the accuracies of both process model and measurements
  - LSE is used at the second stage

- **PMU-ROSE creates the Kalman filter for each observable island**

Prediction and Estimation using Kalman Filtering

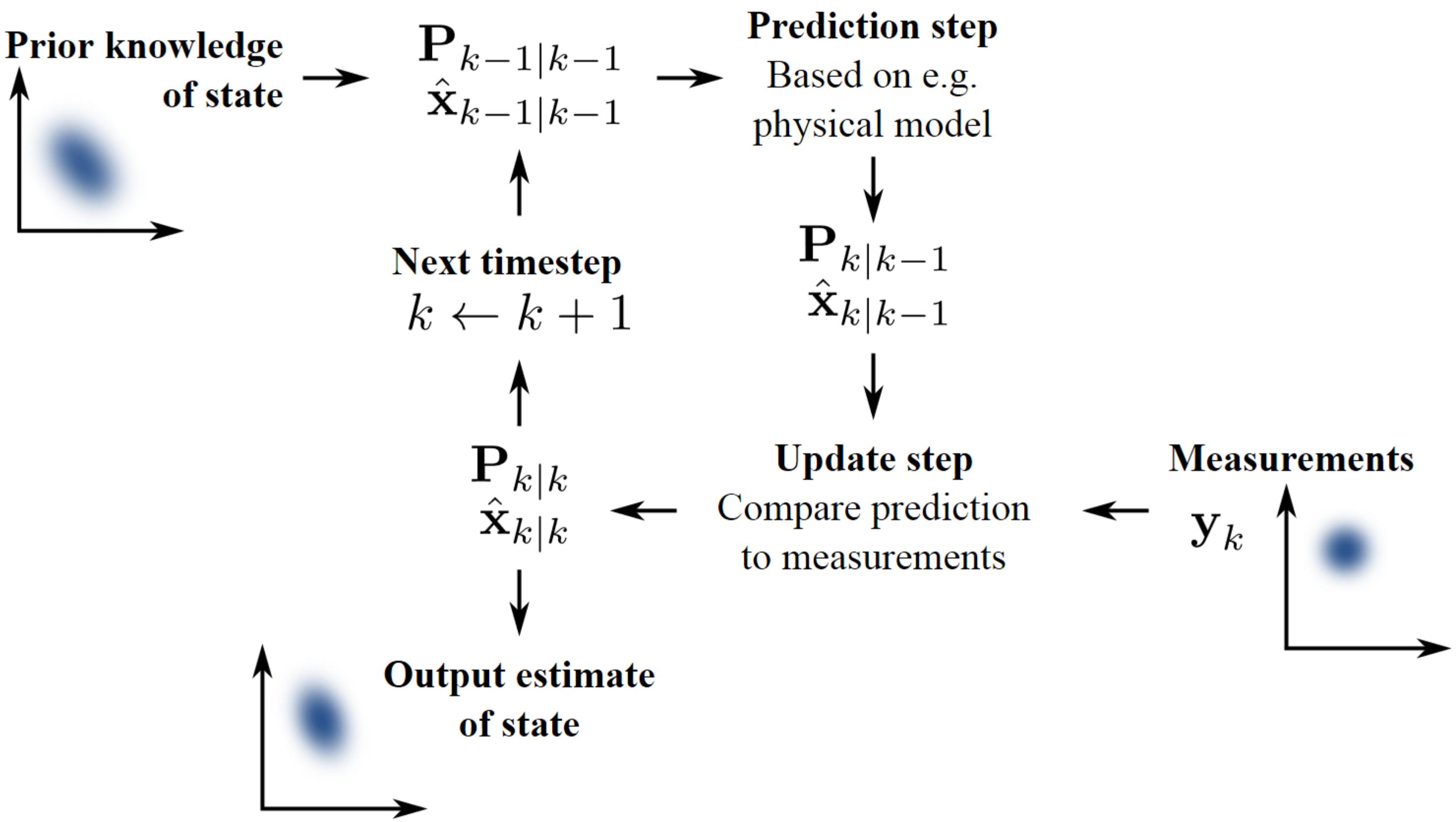
- The prediction equations obtain the a priori state estimate $x_{\bar{k}}$ of the state x_k at time step k given the knowledge of the process prior to time step k, up to and including time step k - 1

- The estimation equations incorporate the new measurements obtained at time-step k into the a priori estimate and are used to derive an improved a posteriori estimate $\hat{x}_k$ of the true state x_k

- Used quadratic prediction based Kalman filter:
    - Uses quadratic relationship between the past, present and future/estimated state

Prediction and Estimation using Kalman Filtering (cont.)

- Stage 1: Prediction
- Stage 2: Correction (Update)



$\hat{x}_{k|k}$ = a posteriori state estimate at time k given observations up to and including at time k

$P_{k|k}$ = a posteriori error covariance matrix (measure of estimated accuracy of the state estimate)

Figure source: https://en.wikipedia.org/wiki/Kalman_filter

Kalman Filtering

- A linear system can be modeled as a pair of linear stochastic process and measurement equations

$$x_k = Ax_{k-1} + w_k$$
$$z_k = Hx_k + v_k$$

where

$x \in R^n$	= the system state vector;
$z \in R^m$	= the measurement vector;
A	= $n \times n$ is the state transition matrix that links the system state at the previous time step $k-1$ to the state at the current time step k;
H	= $m \times n$ matrix that relates the system state and the measurement set z_k;
$w_k \in R^n$	= the process noise at time step k; assumed to be white;
$v_k \in R^m$	= measurement noise; assumed to be white.

Kalman Filtering (cont.)

- The process noise and measurement noise are assumed to be mutually independent random variables with normal probability distributions

$$p(w) \sim N(0, Q_k)$$

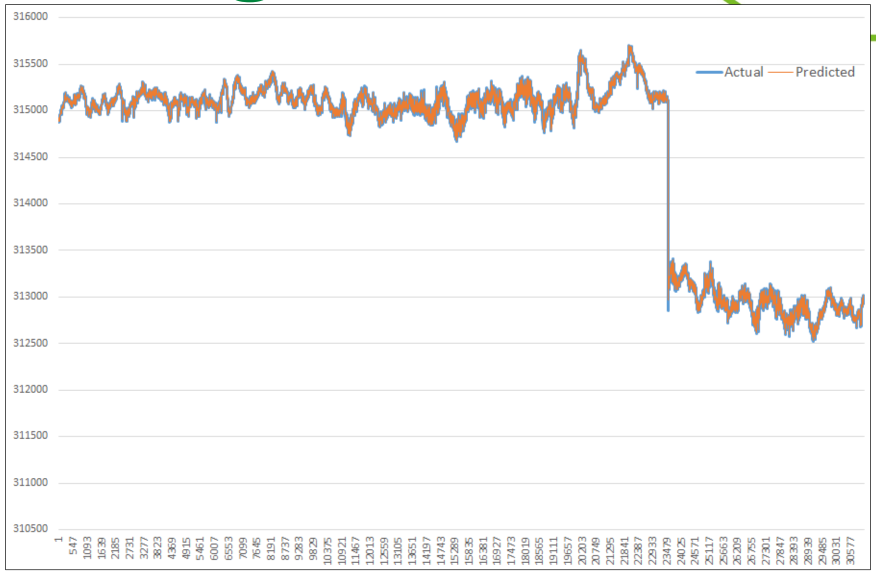
$$p(v) \sim N(0, R_k)$$

where

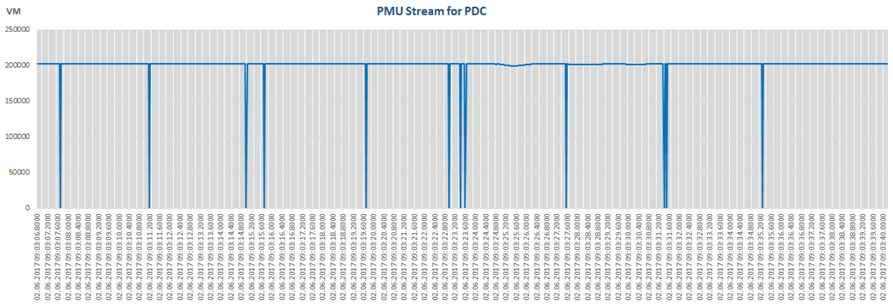
Q_k = the process noise covariance matrix
R_k = the measurement noise covariance matrix

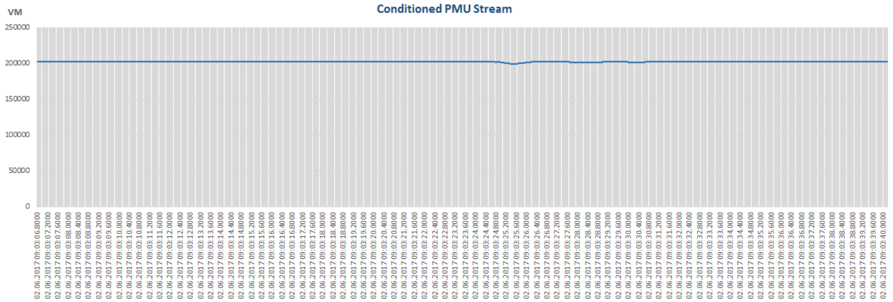
Prediction Using Kalman Filtering

- Stage 1: Prediction



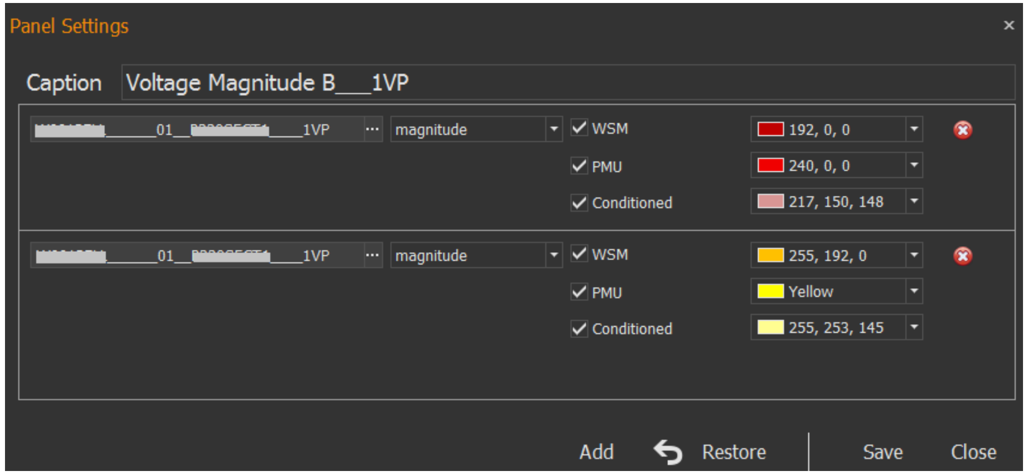
- Stage 2: Correction/LSE
  - Applies both heuristic and statistical methods to identify suspected bad data points and topology changes
  - Considers relationship between signals



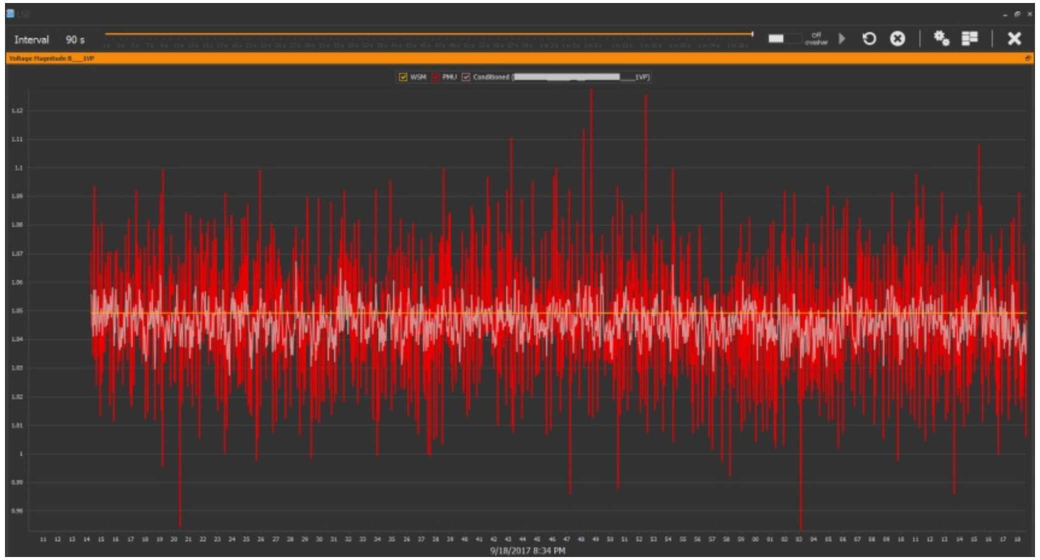


Correction Using Kalman Filtering and LSE

- Displays PMU values and values computed by LSE, compares PMU and LSE values with State Estimator values



- Red graph – real-time PMU measurements;
  Pink graph – the same PMU measurements processed by the LSE (conditioned values);
  Orange line – State Estimator value.



Parameter Learning for Kalman Filter

- During learning we want to learn the best parameters to maximize the likelihood of the input values
- A number of parameters affect filtering:
    - Transition (process) variance
    - Observation (measurement) variance
    - Constraint variance
- At each step, we calculate the likelihood of the new input values and the derivatives of this likelihood by all parameters included in learning
- Make a small gradient step towards likelihood maximization
- After several hundred steps, the parameters are adopted to the data
- Online learning is done continuously

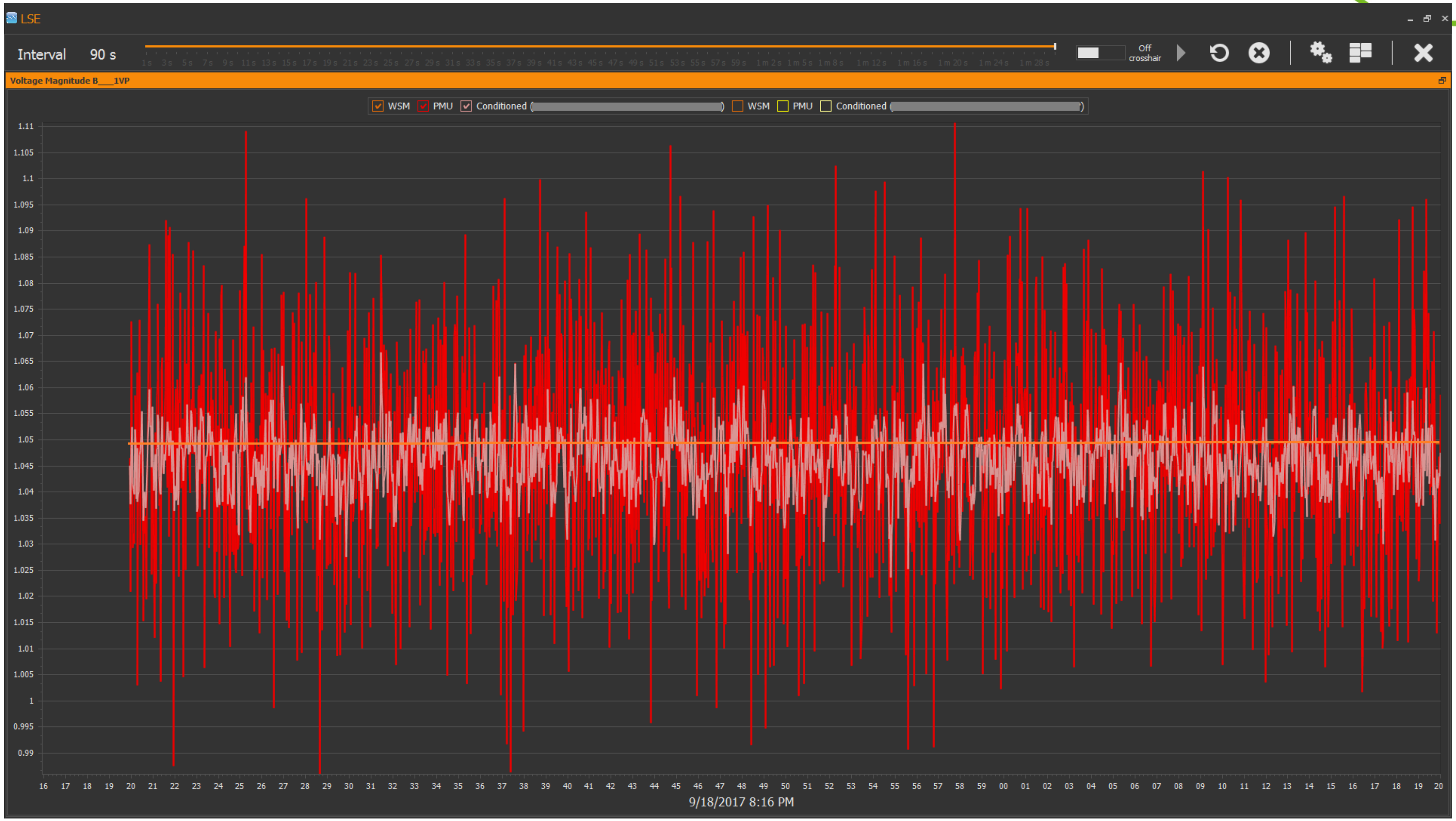
Parameters Included in Learning

- Transition Variance
  - Components of the process noise covariance matrix Q_k
  - Difference between new and previous values (voltages at the next timestamp)
  - The smaller the Transition Variance is, the closer the new values are to the previous ones, and the smoother output signals are

- Measurement Variance:
  - Components of the measurement noise covariance matrix R_k
  - Difference between the measured input values and the predicted ones using the Kalman filter
    - Similar to measurement error
  - The smaller the Measurement Variance is, the closer the output signals are to the input signals

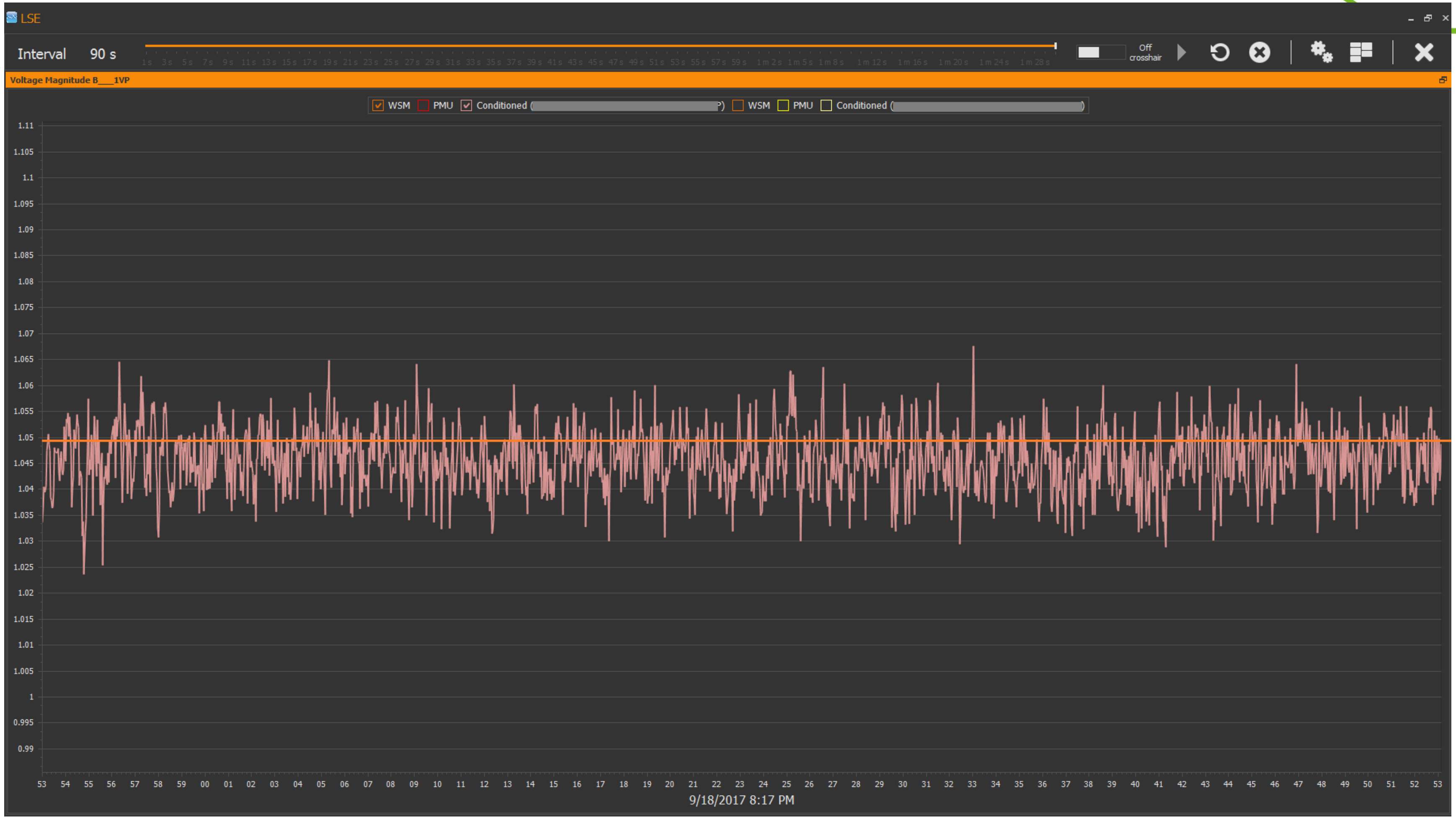
- Constraint Variance:
  - Based on Kirchhoff's law
  - The smaller the Constraint Variance is, the more accurately Kirchhoff's equations are satisfied

Learning is Disabled – 1



Red graph – PMU measurements
Pink graph – LSE values: learning disabled
Orange line – State Estimator value

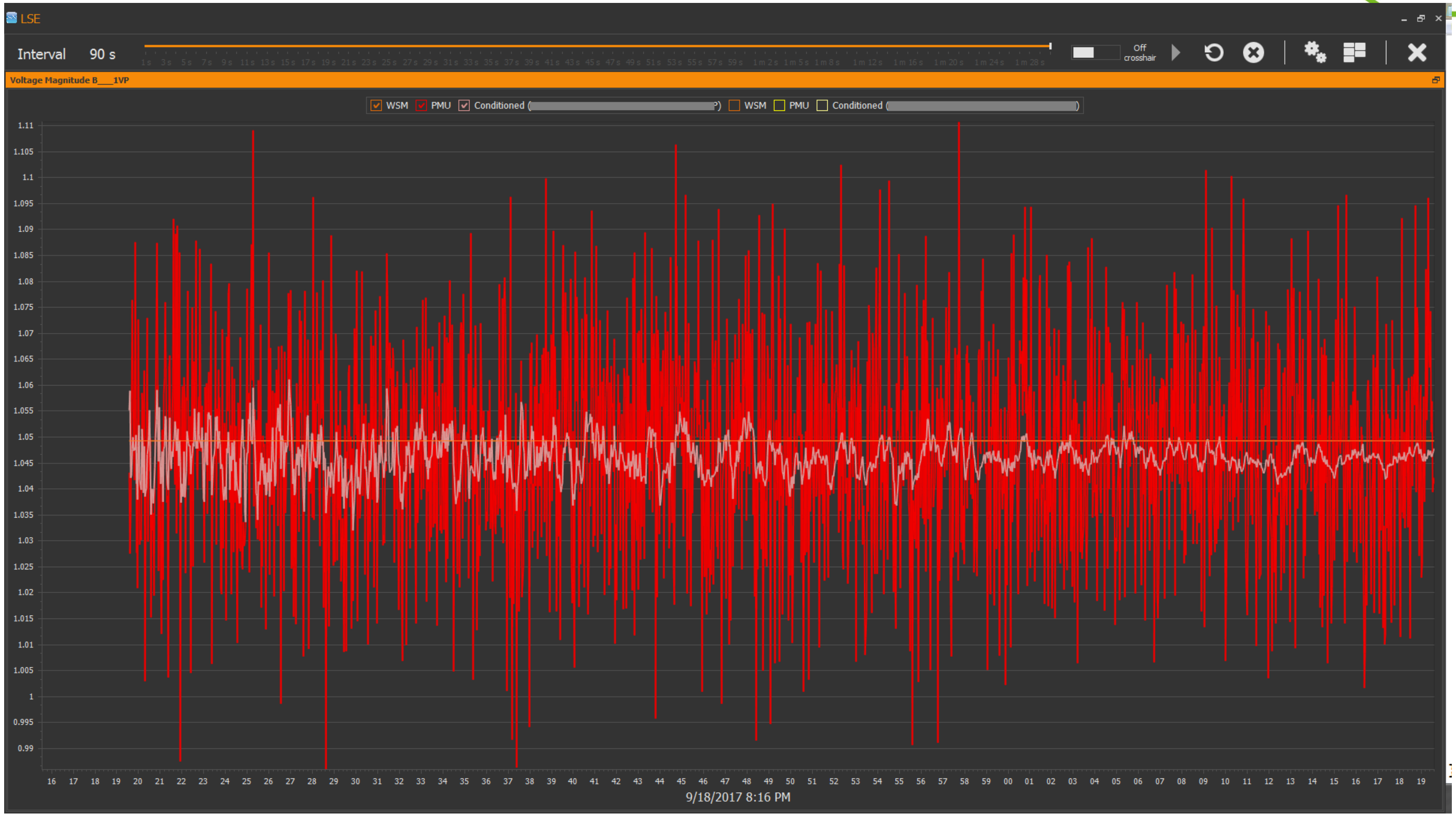
Learning is Disabled – 2



Pink graph – LSE values: learning disabled

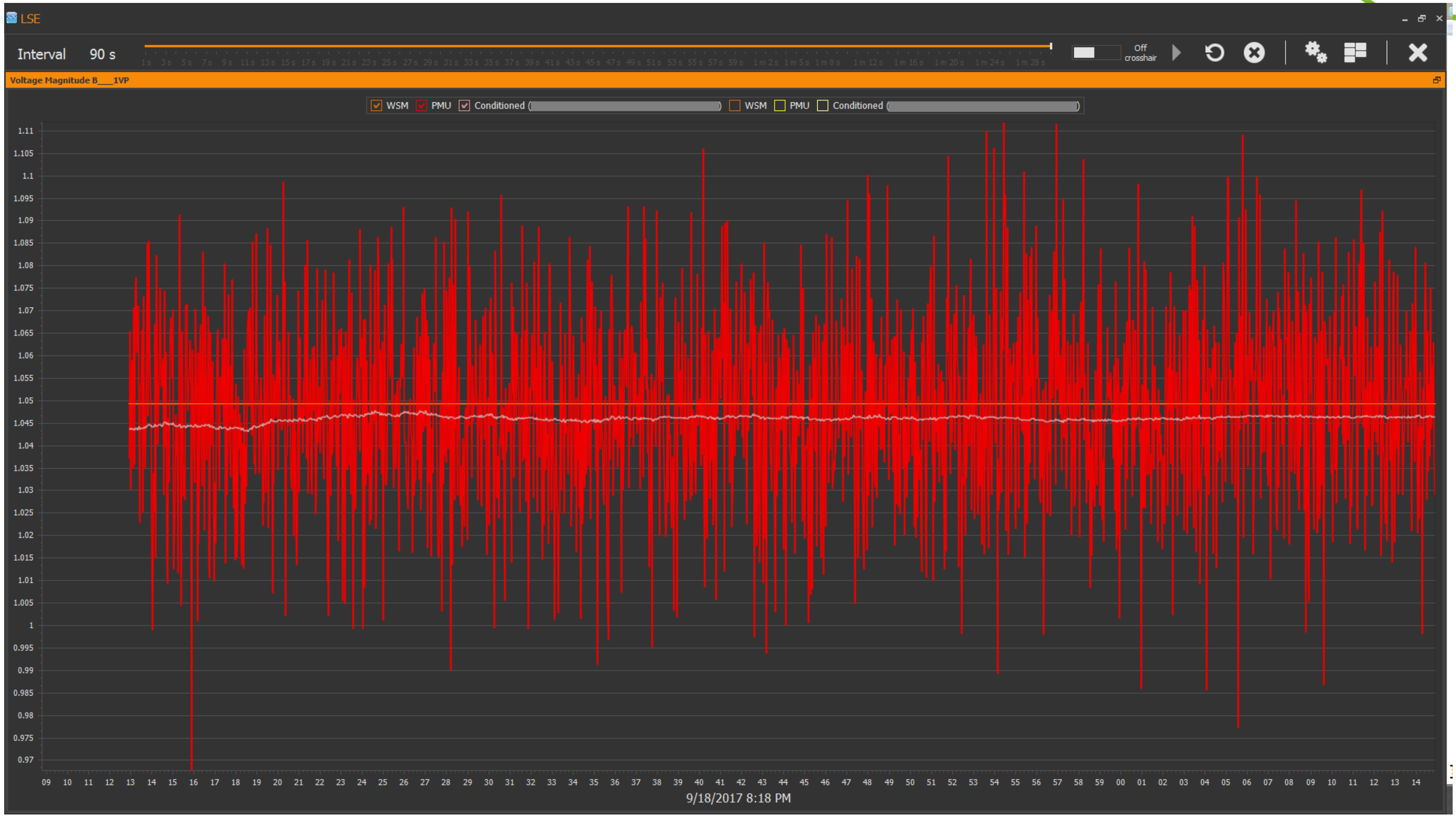
Orange line – State Estimator value

Learning is Enabled, Rate = 0.001



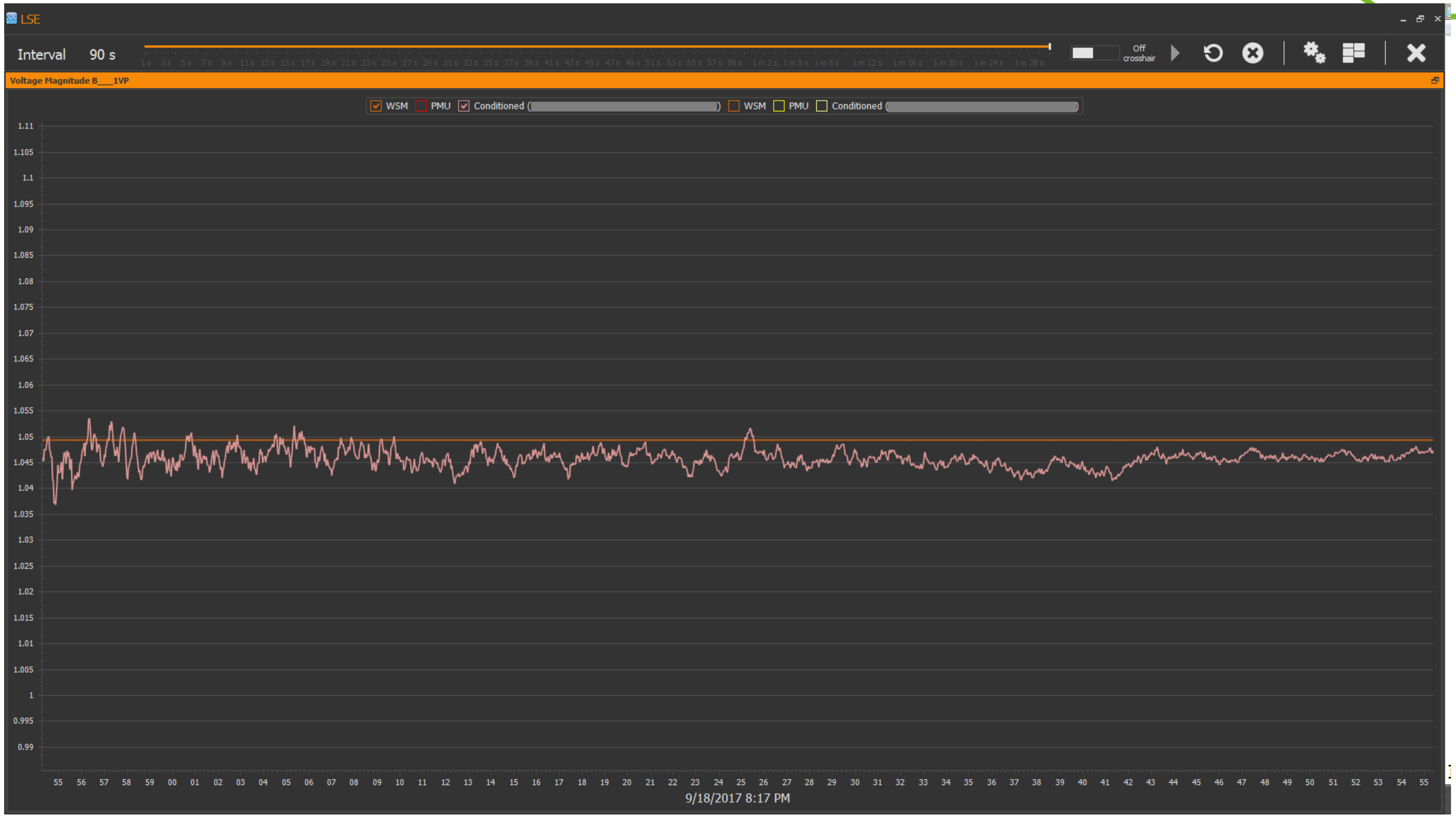
Red graph – PMU measurements
Pink graph – LSE values: learning disabled
Orange line – State Estimator value

Learning is Enabled, Rate = 0.001 (cont.)



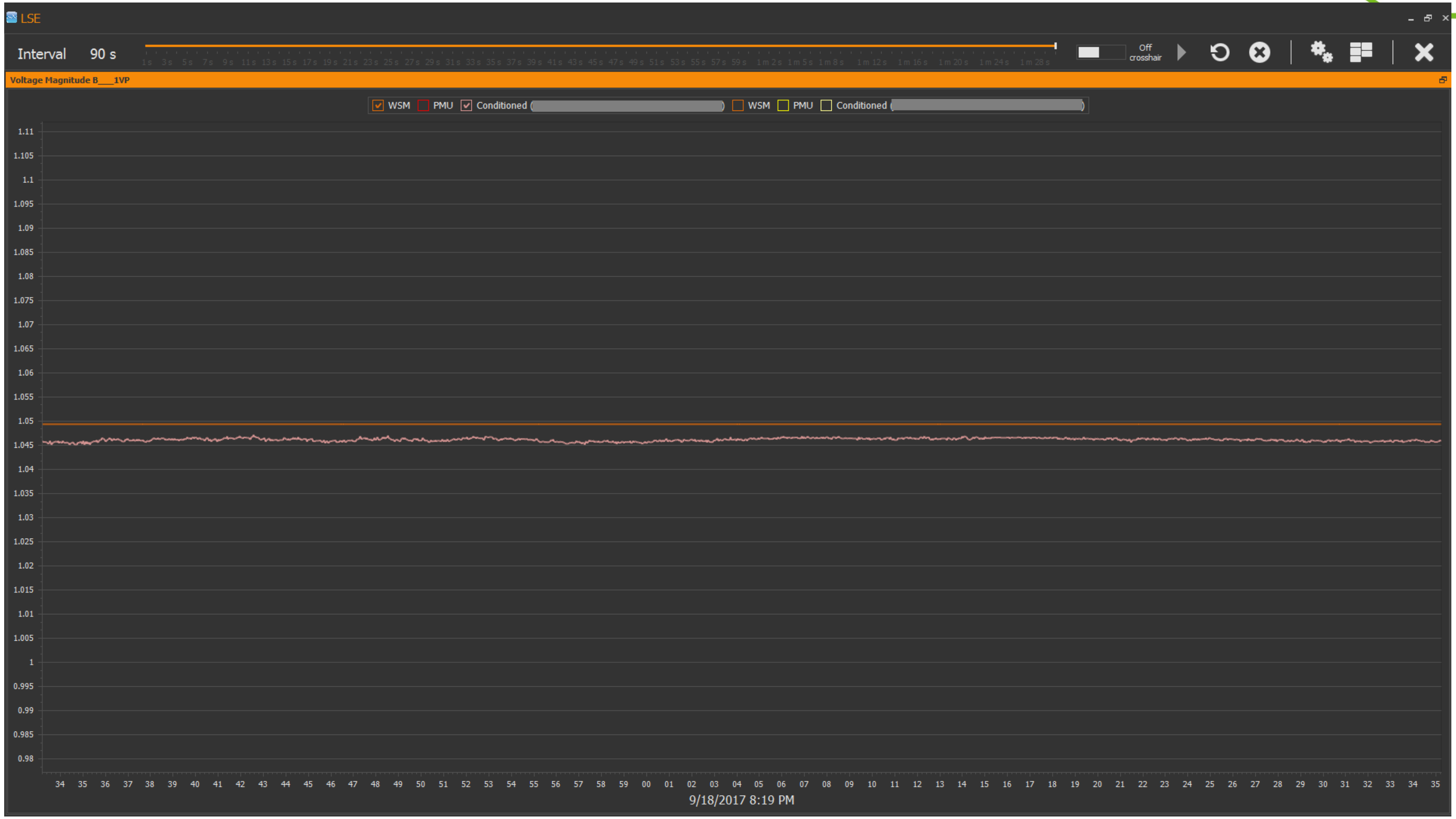
Red graph – PMU measurements
Pink graph – LSE values: learning disabled
Orange line – State Estimator value

Learning is Enabled, Rate = 0.001 (cont.)



Pink graph – LSE values: learning disabled
Orange line – State Estimator value

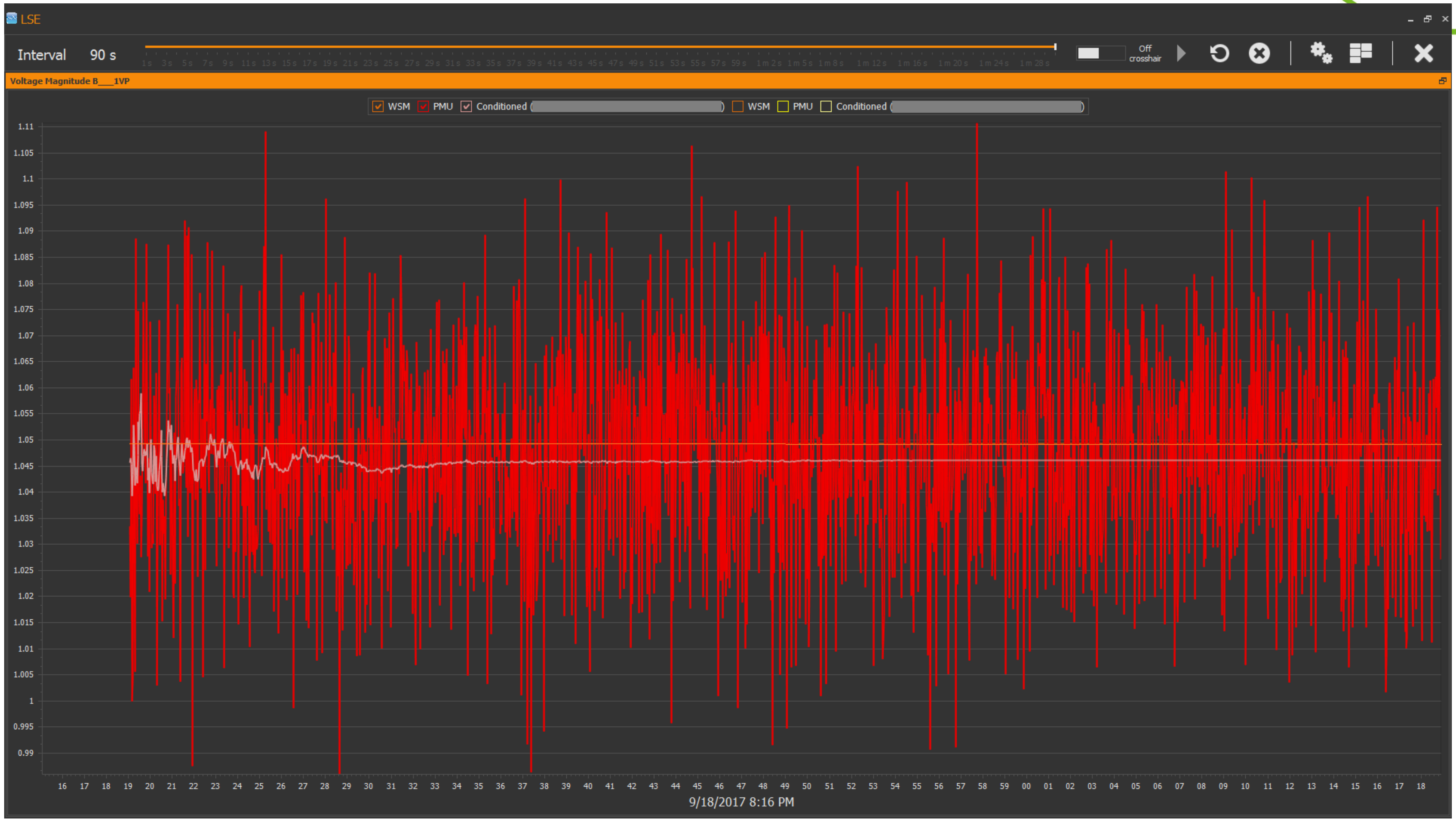
Learning is Enabled, Rate = 0.001 (cont.)



Pink graph – LSE values: learning disabled

Orange line – State Estimator value

Learning is Enabled, Rate = 0.01

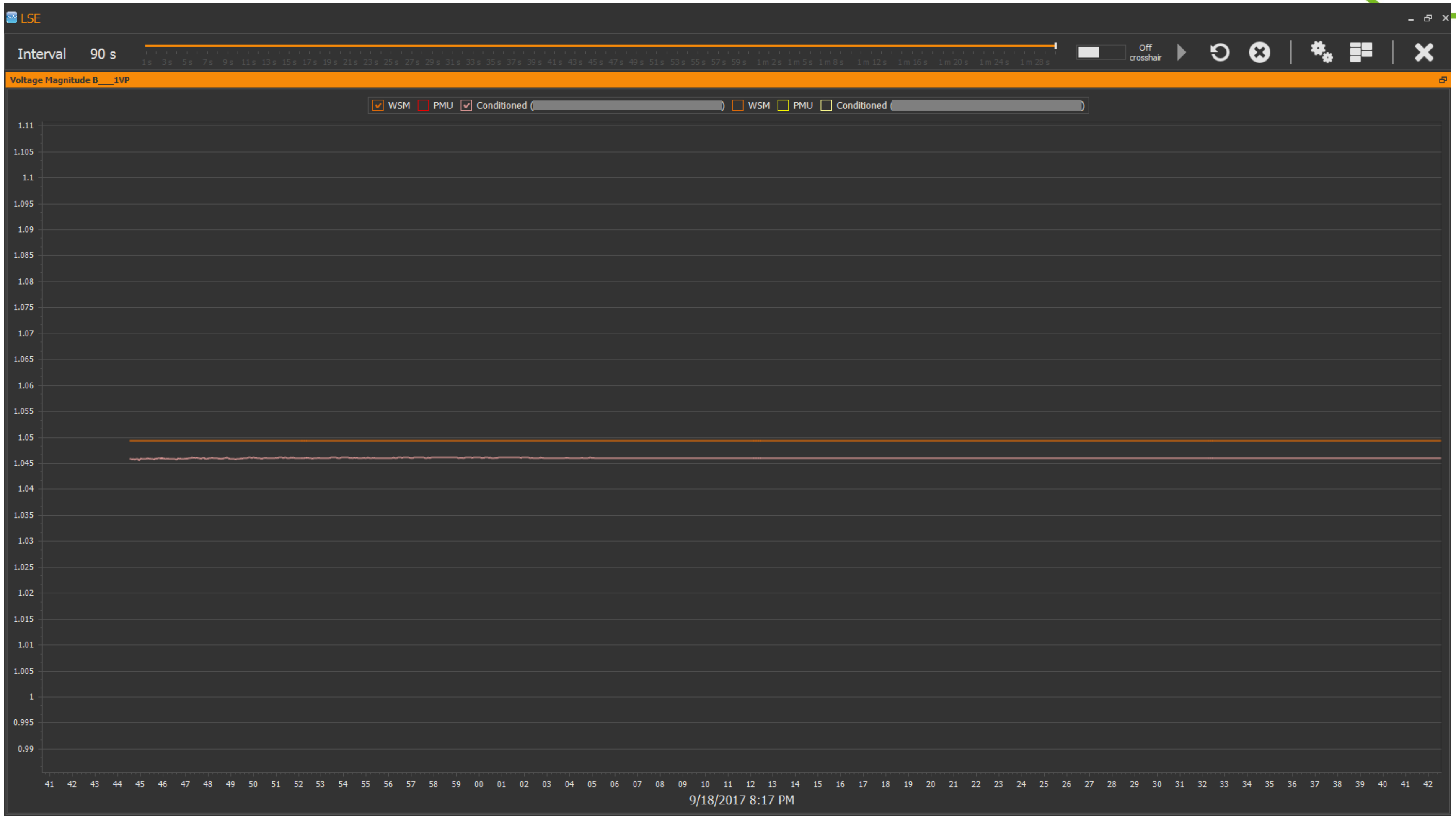


Red graph – PMU measurements

Pink graph – LSE values: learning disabled

Orange line – State Estimator value

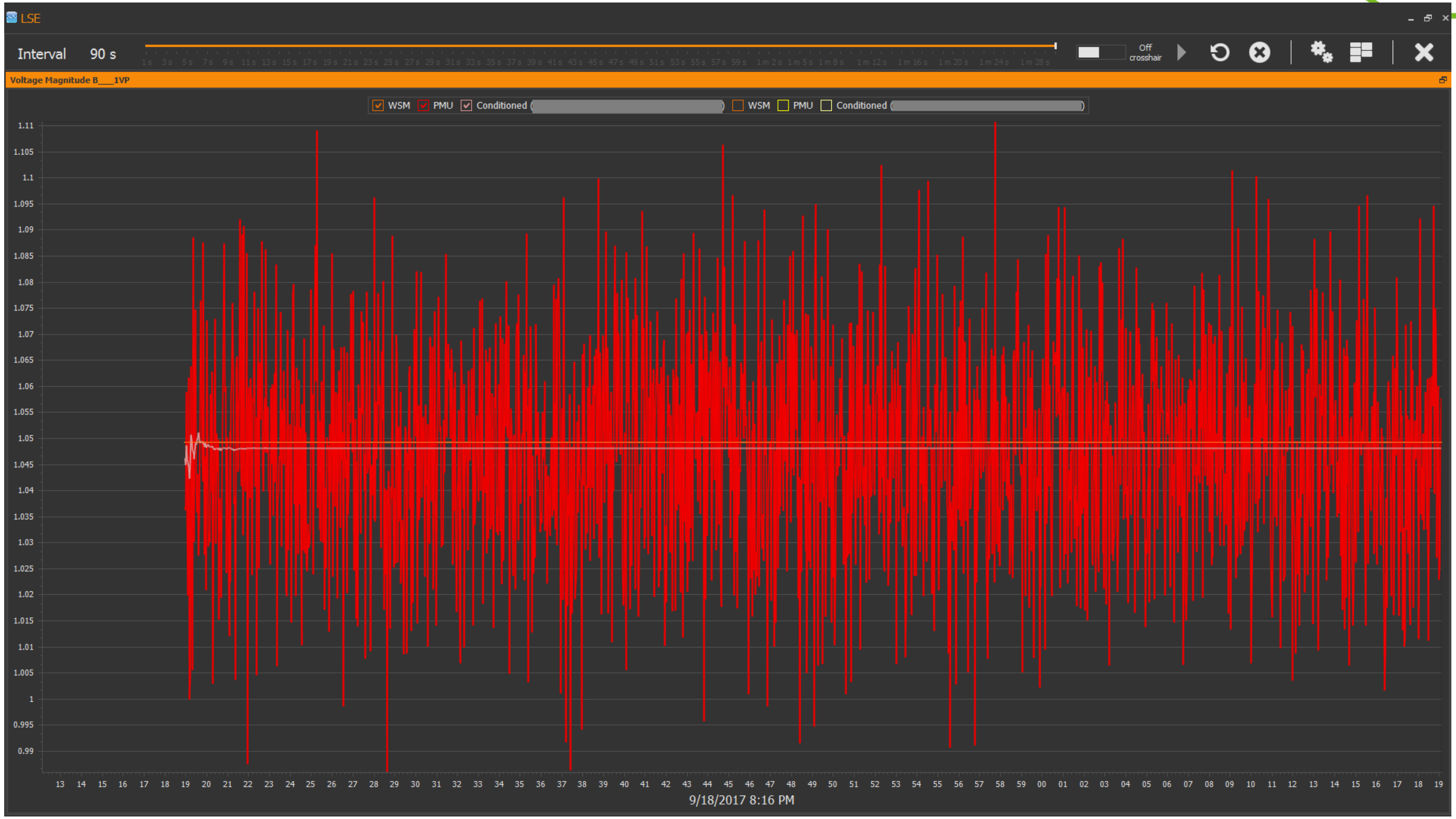
Learning is Enabled, Rate = 0.01 (cont.)



Pink graph – LSE values: learning disabled

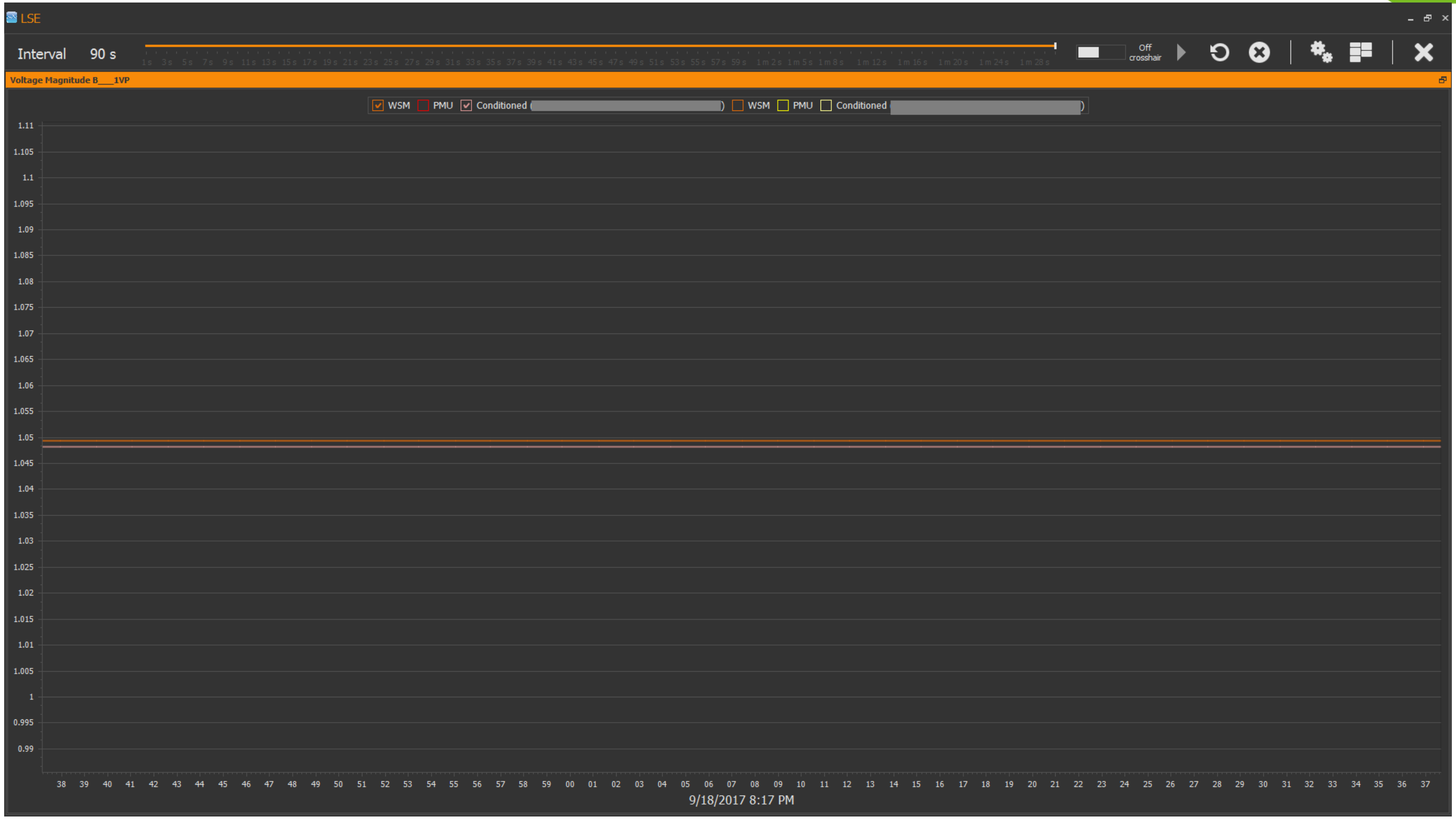
Orange line – State Estimator value

Learning is Enabled, Rate = 0. 1



Red graph – PMU measurements
Pink graph – LSE values: learning disabled
Orange line – State Estimator value

Learning is Enabled, Rate = 0. 1 (cont.)



Pink graph – LSE values: learning disabled

Orange line – State Estimator value

Conclusions

- PMU-ROSE creates the Kalman filter for each observable island
- Use of machine learning improves the accuracy of linear state estimation
- Online learning is done continuously

Thank you!

marvaiman@vrenergy.com