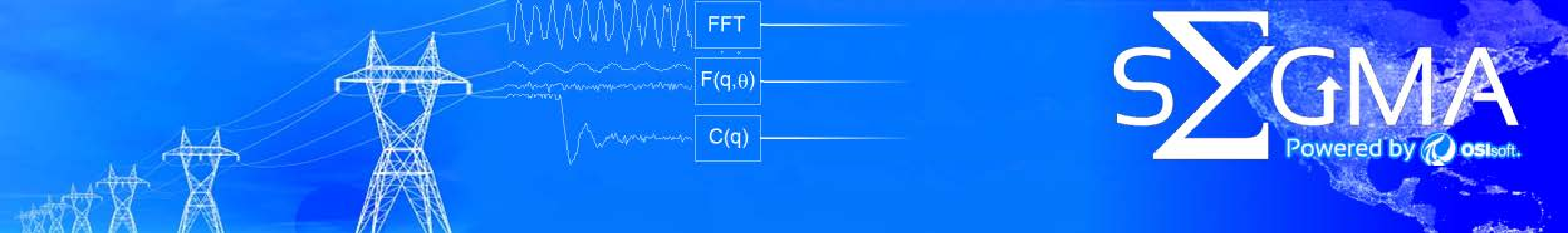


Electric Grid Monitoring using Synchrophasor Data

Sai Akhil Reddy Konakalla
Prof. Raymond de Callafon

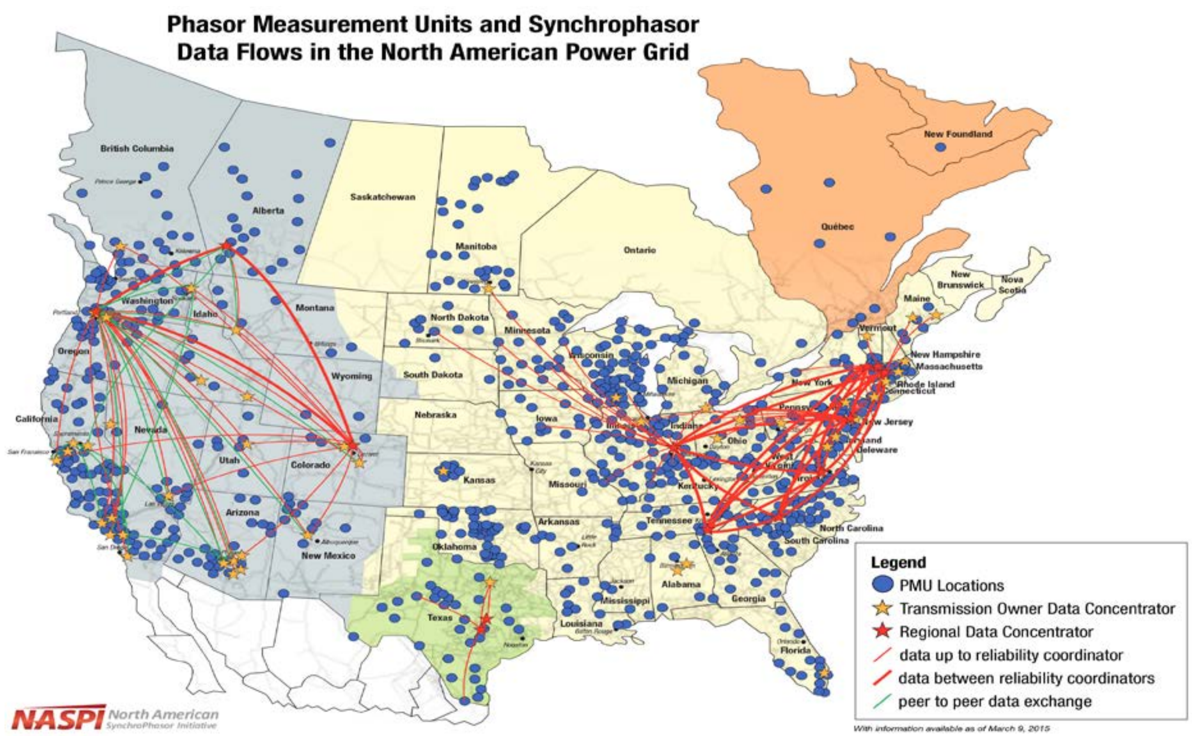
University of California, San Diego

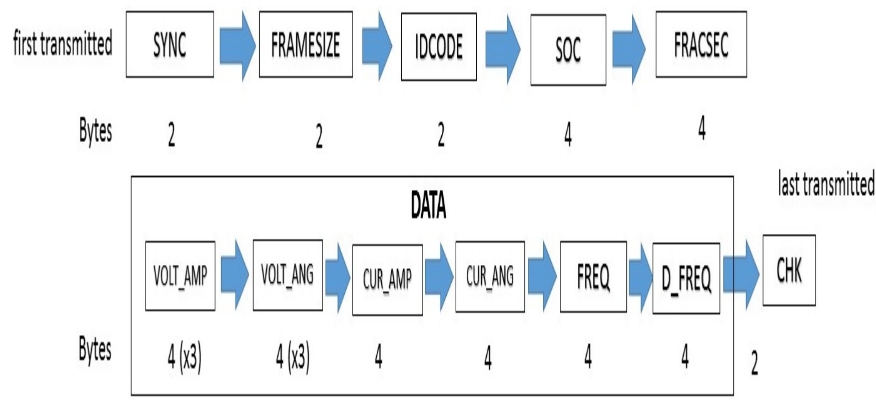
UC San Diego

Email: skonakal@ucsd.edu

Synchrophasors

- Three phase signals sampled at 512x samples/cycle
- Produces 3 phase voltage and current phasors & frequency @ 60Hz
- Data streamed in real-time using transport layer (TCP/IP, UDP)



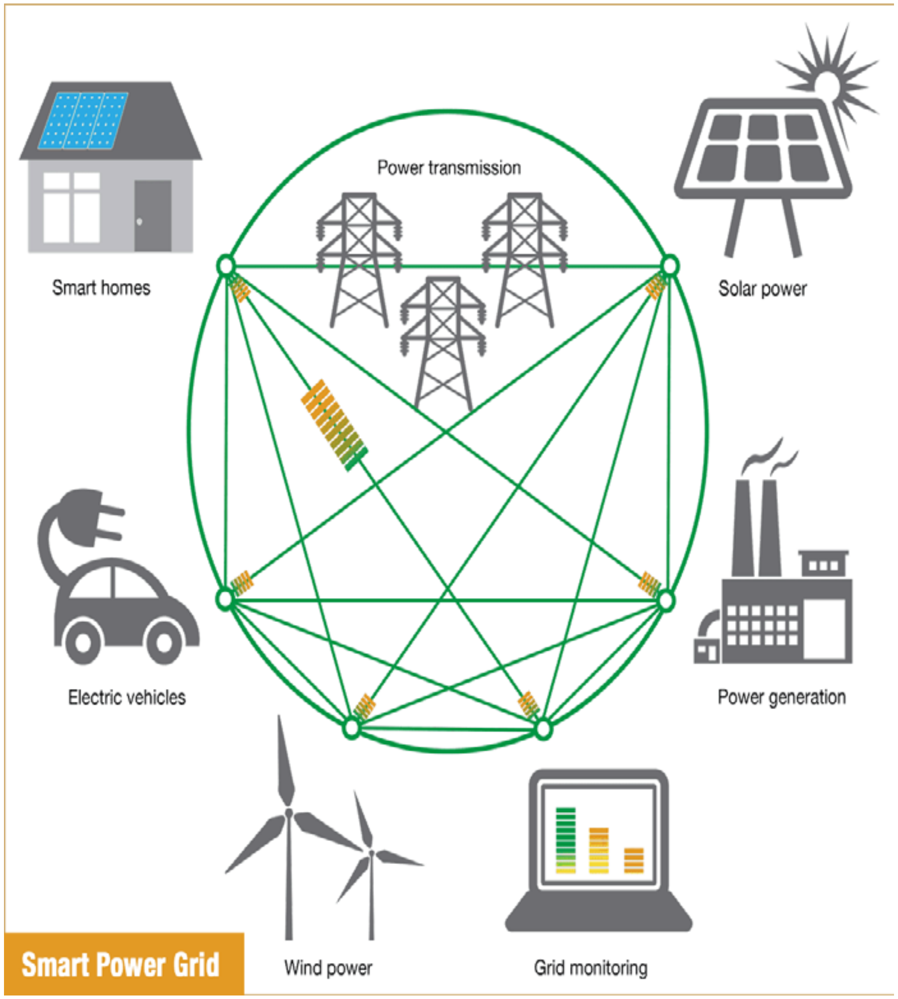


Cons:

- Missed data
- Latency
- Filtering
- Noisy data
- Cybersecurity

Introduction

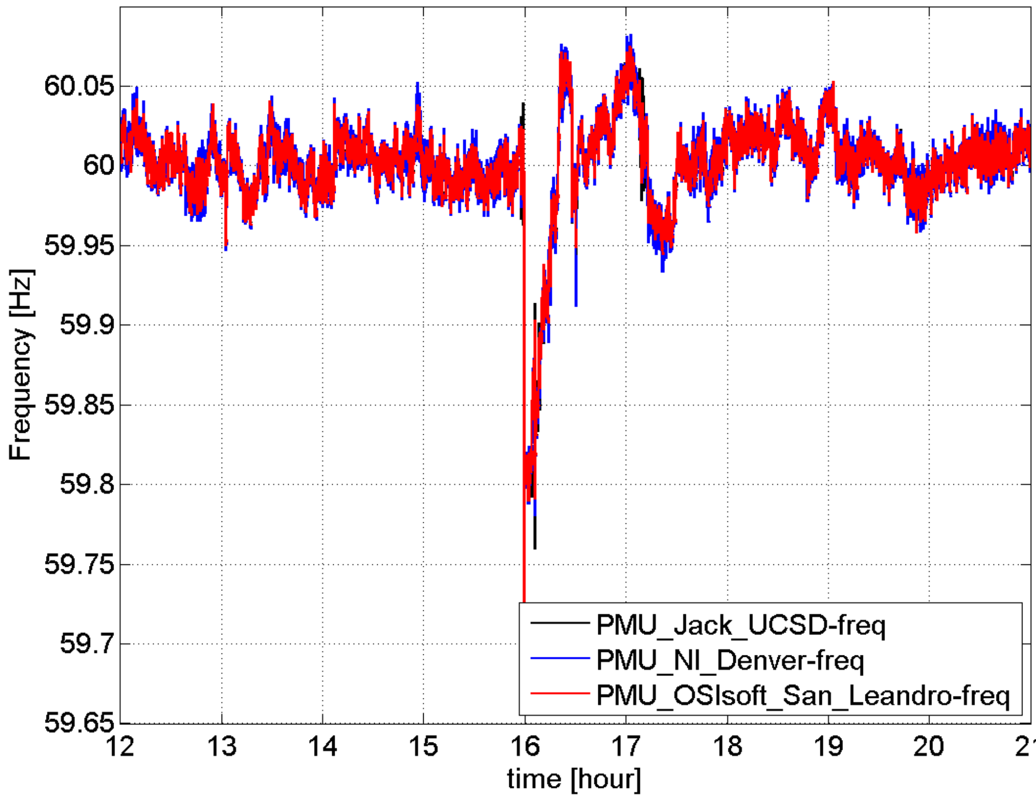
- Intensification of distributed renewable energy resources, storage systems
- Rising need to monitor power flow and quality more accurately, rapidly in the electric grid
- Phasor Measurement Units (PMU):
  - GPS time synchronized
  - 3 phase AC electric signal
  - AC frequency
- Automate data analysis and use it for real-time distributed control



Motivation

"Smart" PMU: *local* signal processing and detect/store events *centrally*

- How to implement "local" signal processing?
- Can "local" processing used to detect individual events?
- Can event detection be distributed on each PMU?



May 30 data: 972000 data points (30Hz sampling noon-9pm)

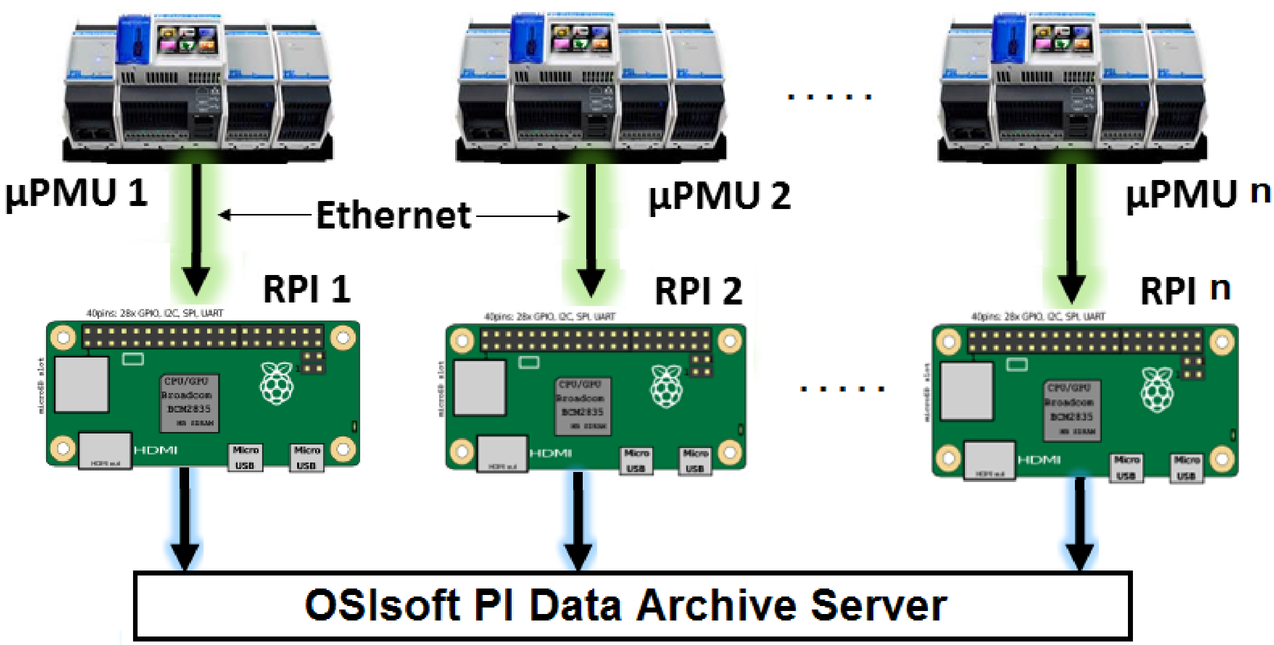
Content

- Grid Event Detection
- Grid Event Classification
- Grid Event Localization

PMU based Grid Event Detection

Infrastructure based on real-time local processing of PMU data:

- Decode the IEEE C37.118 data frames from microPMUs
- Filter phasor data to obtain Filtered Rate of Change (FRoC) signal
- Formulate event detection based on the obtained FRoC signal
- Store finite batch size of PMU data in case of an event only

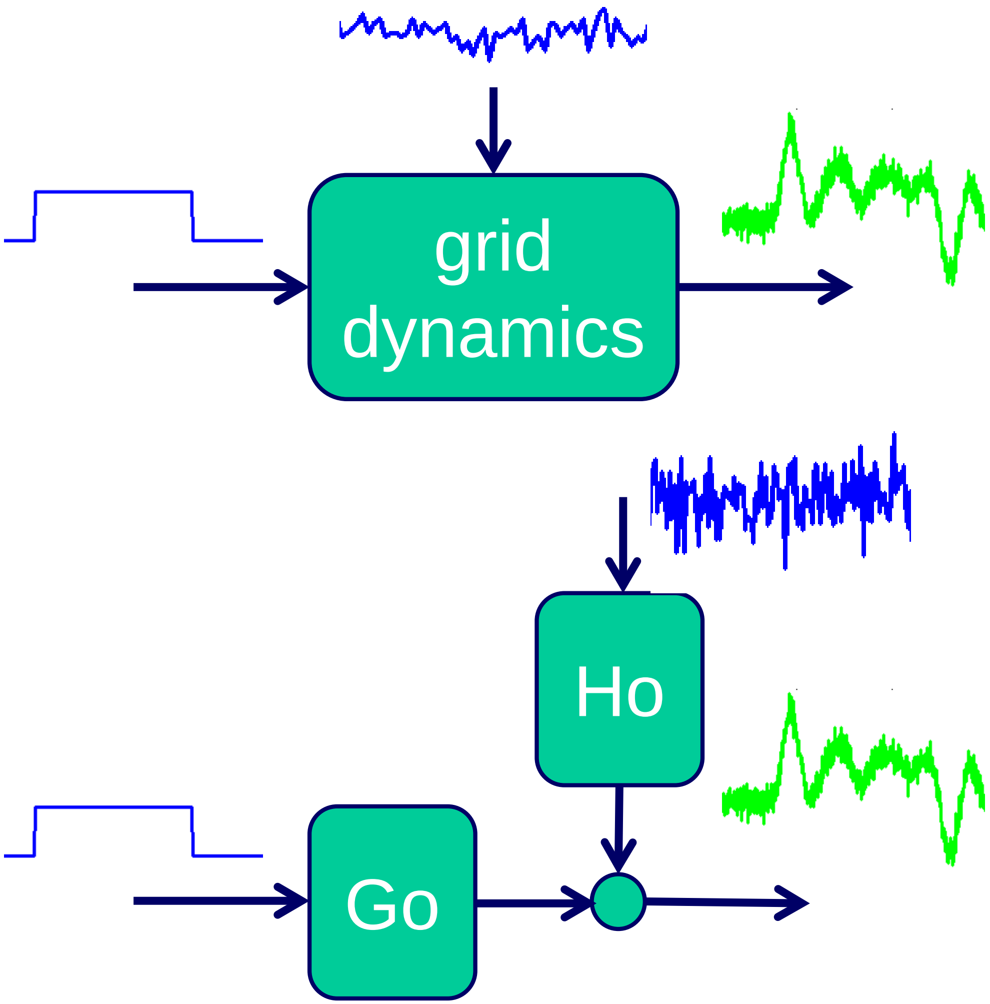


- Data ported on a wired Ethernet over TCP/IP
- TCP sockets to capture PMU data in python
- Real time decoding and filtering of C37.118 data
- Data available for event detection

Filtering the Phasor Data

Approach is based on dynamic and statistical analysis of PMU data

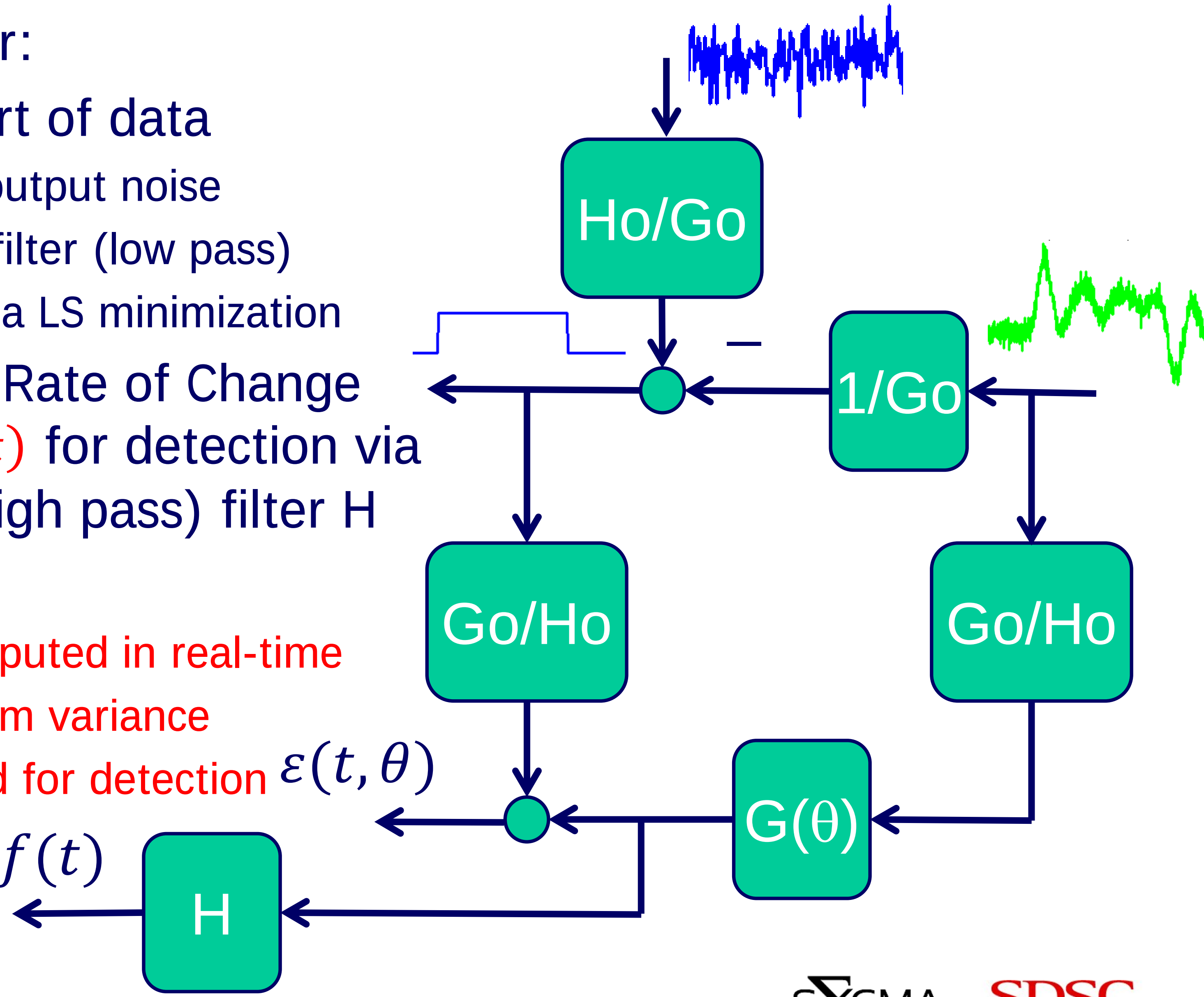
- Assume PMU observation is linear combination of:
  - Main event signal filtered by grid dynamics
  - Small/random events filtered by grid dynamics
- What's new here:
  - Use knowledge on main modes (grid frequency and damping)
  - Compute optimal detection signal by reconstruction of (filtered) main event signal



Filtering the Phasor Data

Computation of filter:

- Select “small” part of data
  - Model noise as output noise
  - Add fixed noise filter (low pass)
  - Compute filter via LS minimization
- Define a Filtered Rate of Change (FRoC) signal $f(t)$ for detection via differentiation (high pass) filter H
- End Result:
  - $f(t)$ can be computed in real-time
  - $f(t)$ has minimum variance
  - $f(t)$ can be used for detection



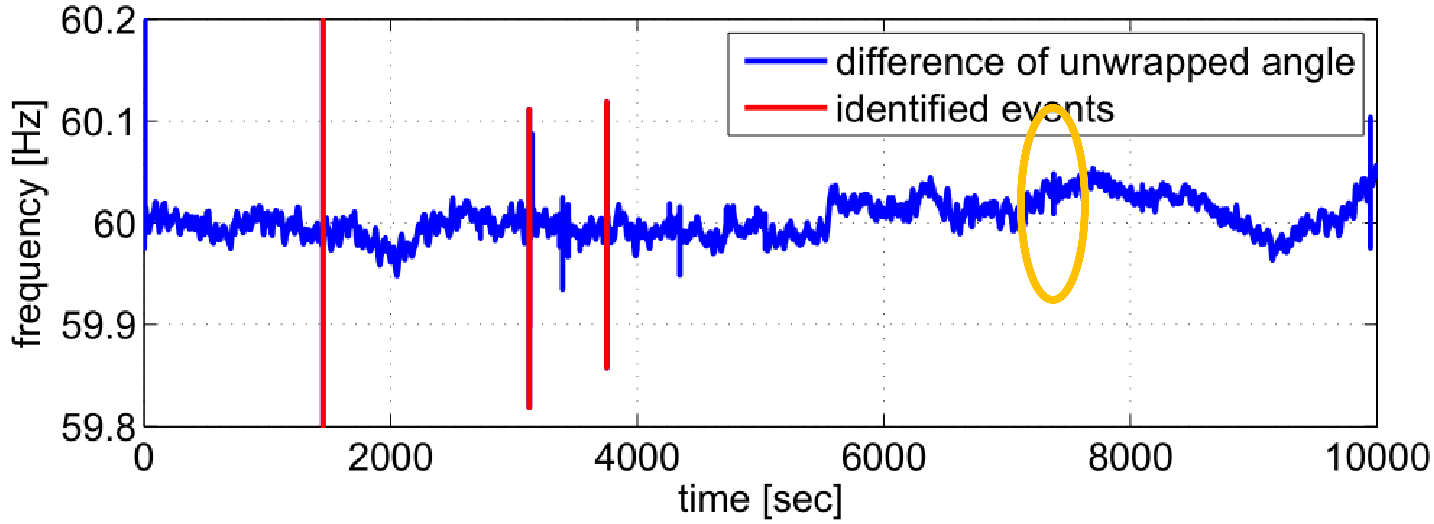
Local/Edge Event Detection

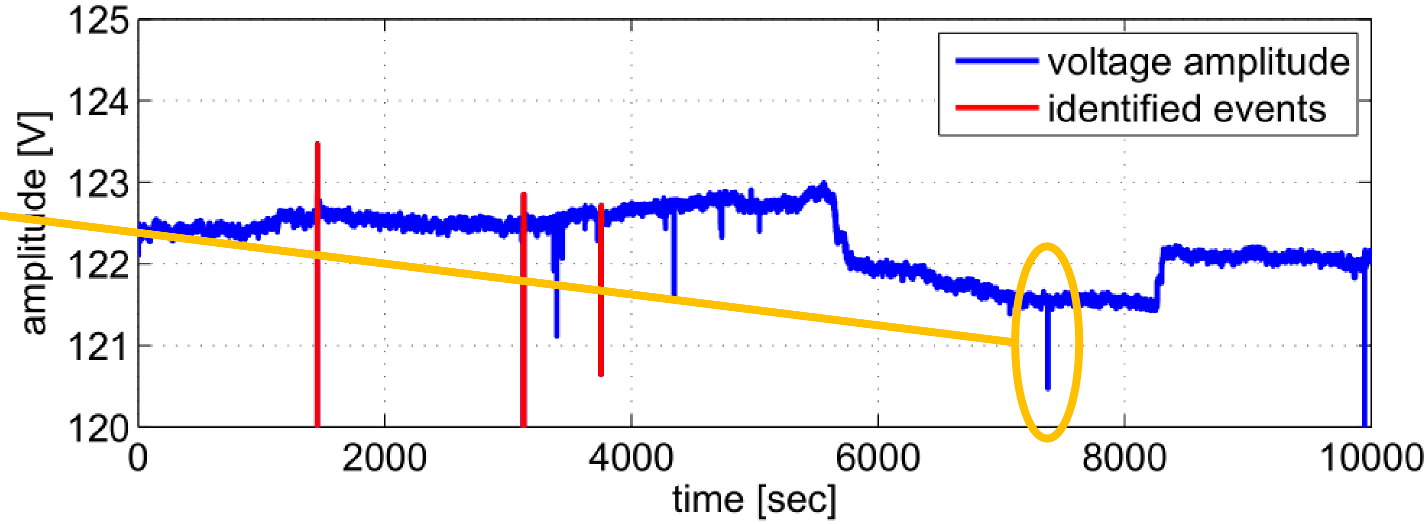
Automatically:

- **Detect events** (via threshold on Filtered Rate of Change signal)
- **Store event** data
- **Notification**/email

Note:

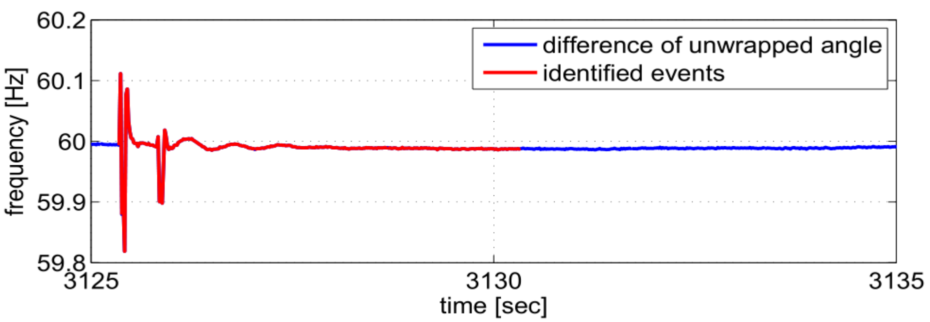
- Not every disturbances is an "event"!
- Hard to see, but clear with edge processing

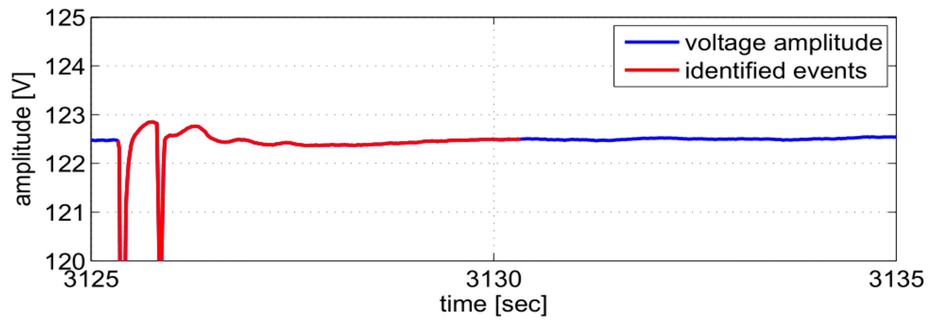




Feature Extraction

- Large interconnected grid (e.g. WECC) -> High complexity
- Highly coupled non-linear dynamic behaviour
- Derivation of swing equations becomes an unwiedly task
- Solution: Deploy System identification techniques using event data for model estimation





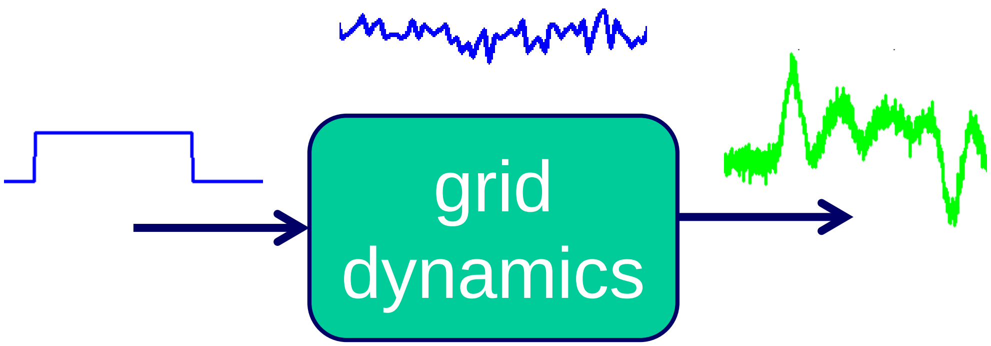
$$M \begin{cases} x(k+1) = A\,x(k) + B\,u(k) \\ y(k) = C\,x(k) + D\,u(k) \end{cases}$$

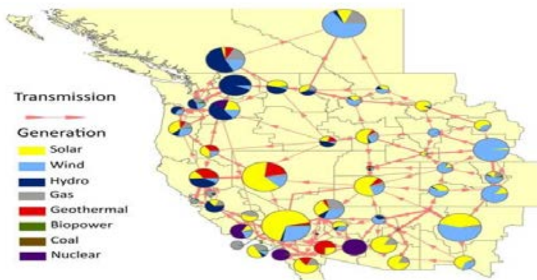
- {A,B,C,D} computed using Realization algorithm

Mathematical tools used:

- Singular Value Decomposition
- Hankel Decomposition

Feature Extraction





Extracted features for classification include:

1. Oscillation frequency (f_i)
2. Damping ratio (ζ_i)
3. Participation Factor (P_i)
4. Post-event frequency deviation (ΔF_i)

Feature set estimation based on known system parameter "*A*":

$$f_i = \frac{|s_i|}{2\pi};\ \zeta_i = \frac{-a_i}{2\pi f_i};$$
$$P_i = |\phi_i \psi_i|$$

$$\Delta \mathrm{F} = \frac{1}{N_1}\sum_{k=N_1-k_0}^{k_0-1} F(k) - \frac{1}{N_2}\sum_{k=k_0+N}^{k_0-N+N_2-1} F(k)$$

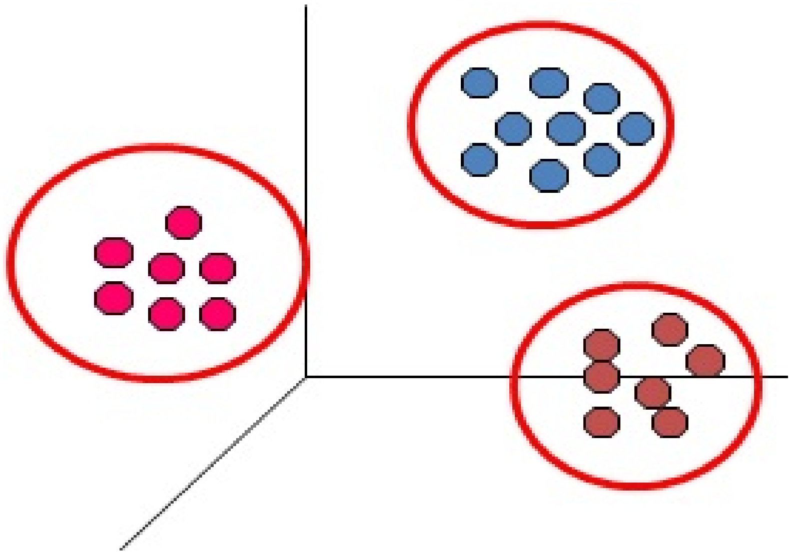
$$s_i = f_s * ln(\lambda_i) = a_i \pm j\, b_i$$

$$[A - \lambda_i I]\ \phi_i = 0$$
$$\psi_i\ [A - \lambda_i I] = 0$$

where ψ_i , ϕ_i are left and right eigenvectors.

Clustering based Classification

- Unsupervised classification technique
- *k*-means: city block, Euclidean etc.
- *p*-dimensional subspace method
- $J(c, C) = \sum_{j=1}^{k} \sum_{x_i \in C_j} |x_i - c_j|$
- Data grouped into clusters for event classification using feature sets



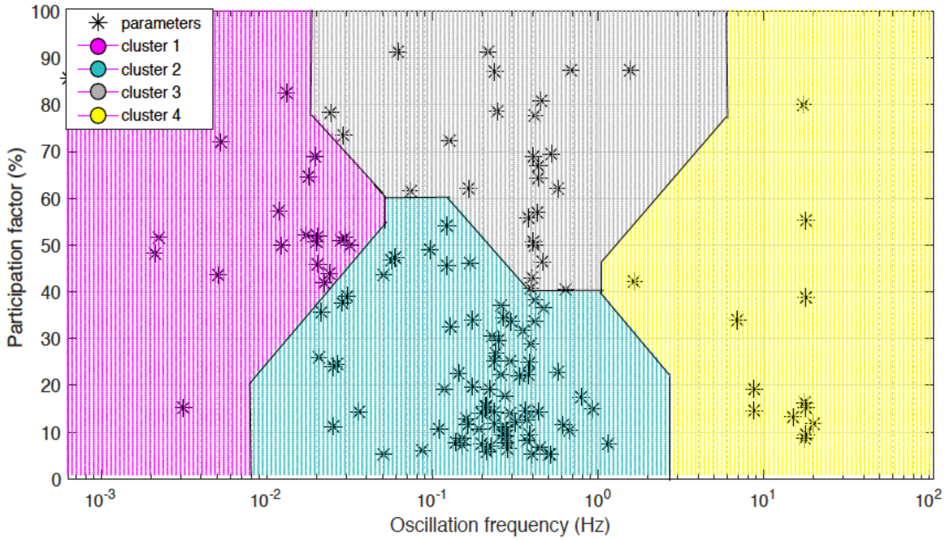
Event Classification

- Event detection based on ***V*** & ***f*** PMU data.

- Feature set extraction using Realization algorithm.

- Features for classification include:

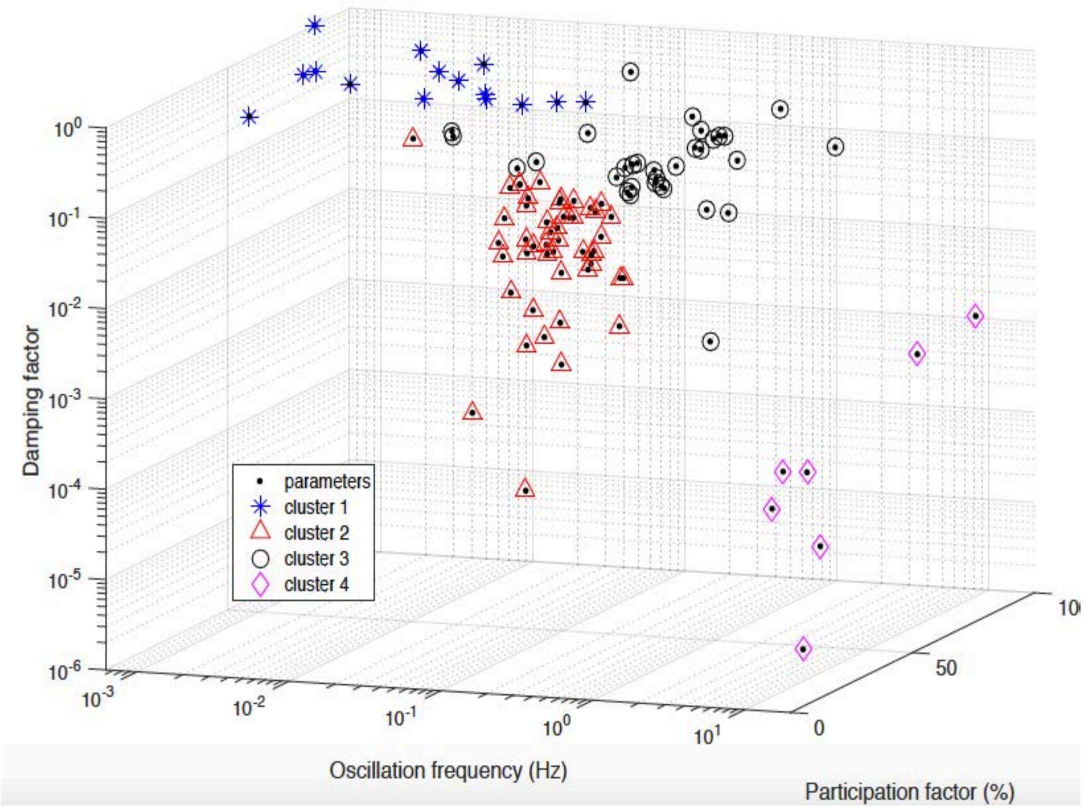
1. Oscillation frequency (f_i)
2. Damping ratio (ζ_i)
3. Participation Factor (P_i)
4. Post-event frequency deviation (ΔF_i)



- *k-means* clustering for unsupervised classification, *P*-dimensional clustering of the feature set.

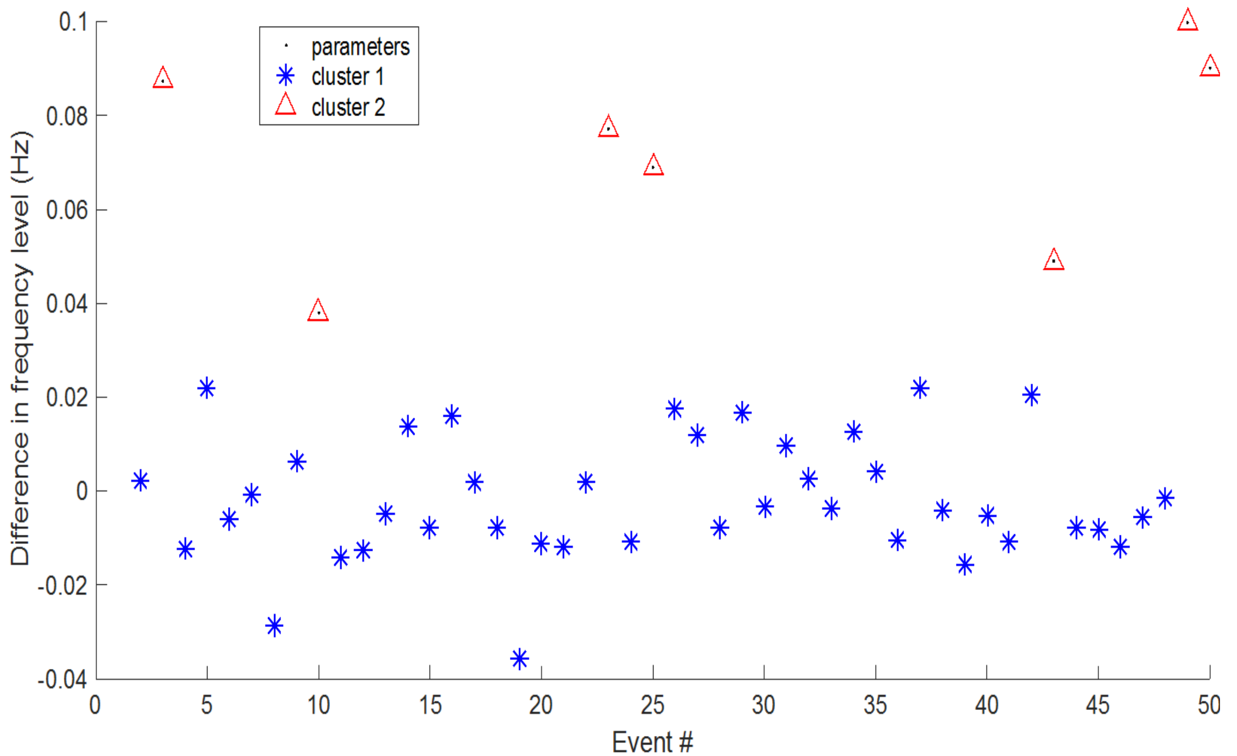
3-D Clustering Analysis

- Oscillations grouped again into 4 clusters based also on damping ratio (ζ_i)
- Clear distinction of clusters with damping levels: local vs non-local events
- Highly damped components likely seen in non-local events due to damping controllers



Level Based Clustering

- Steady State Frequency Deviation (ΔF_i) Clustering
- Information on steady state power loss or power surplus.



- *$\Delta F_i < 0$ => Overload (or generation loss)*
- *$\Delta F_i > 0$ => Surplus generation (or load loss)*

Validation on Test Events

Validation of classification algorithm on (known) recorded events:

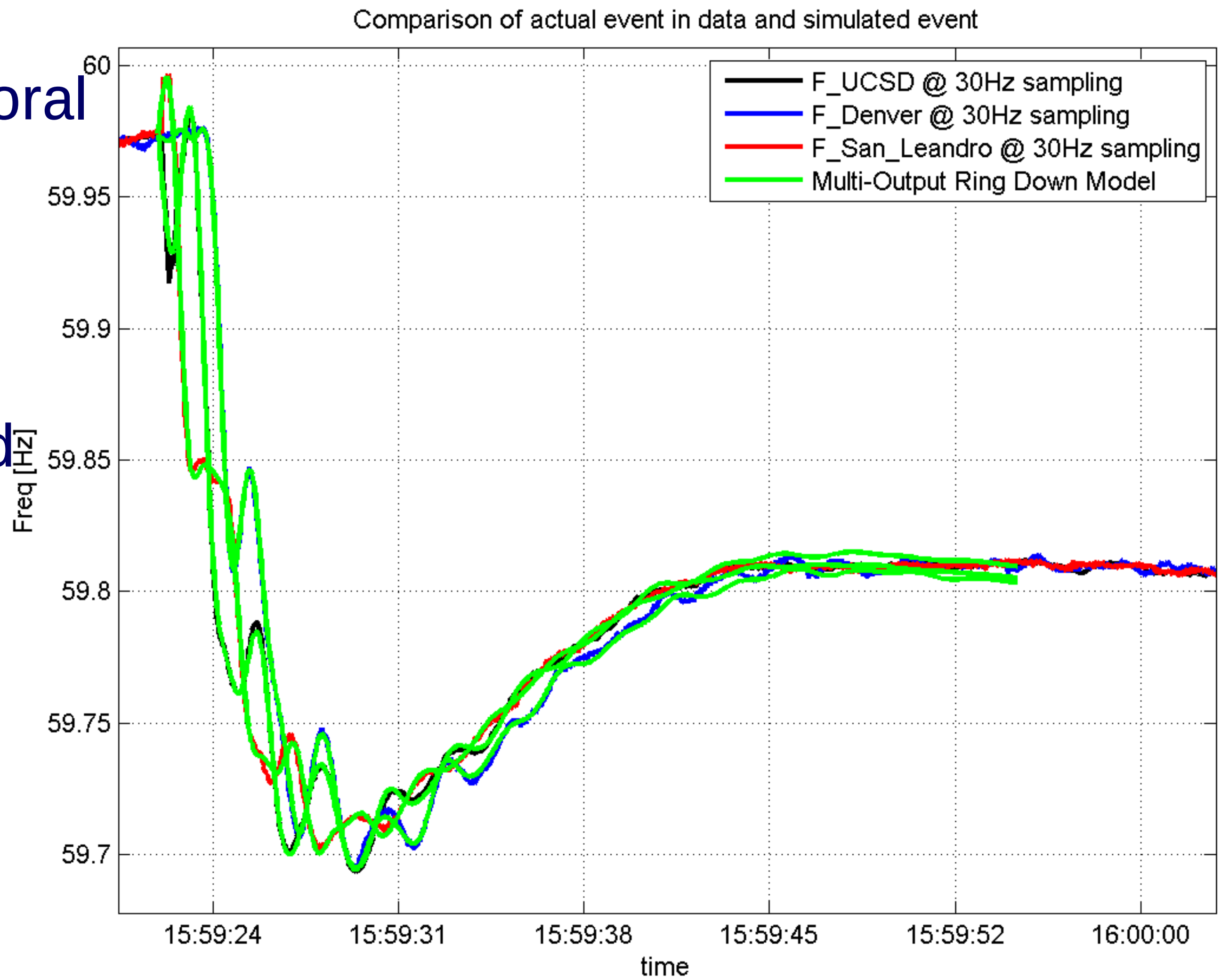
- Event 1- BPA Chief Joseph Brake test: HPFI, non-local event without any power loss.
- Event 2- San Diego Tornado warning: LPFI, local event without any power loss.
- Event 3- Montana generation loss: HPFI, non-local event with generation loss.

Test Events (Cluster)	Classifier 1				Classifier 2				Classifier 3	
	$C1$	$C2$	$C3$	$C4$	$C1$	$C2$	$C3$	$C4$	$C1$	$C2$
Event 1 (April 14^{th}, 2015)	X	-	X	-	X	-	X	-	X	-
Event 2 (January 6^{th}, 2016)	X	X	-	-	X	X	-	-	X	-
Event 3 (January 21^{st}, 2016)	X	-	X	-	X	-	X	-	-	X

SΣGMA Powered by OSIsoft SDSC SAN DIEGO SUPERCOMPUTER CENTER

Event Localization- Topology

- Use spatial and temporal knowledge of PMU measurements
- Use PMU network and topology
- Dynamic Model.
  (ring down analysis)



Localization- Grid Dynamics

- Use known (estimated) grid dynamics

- Inverse estimation problem:

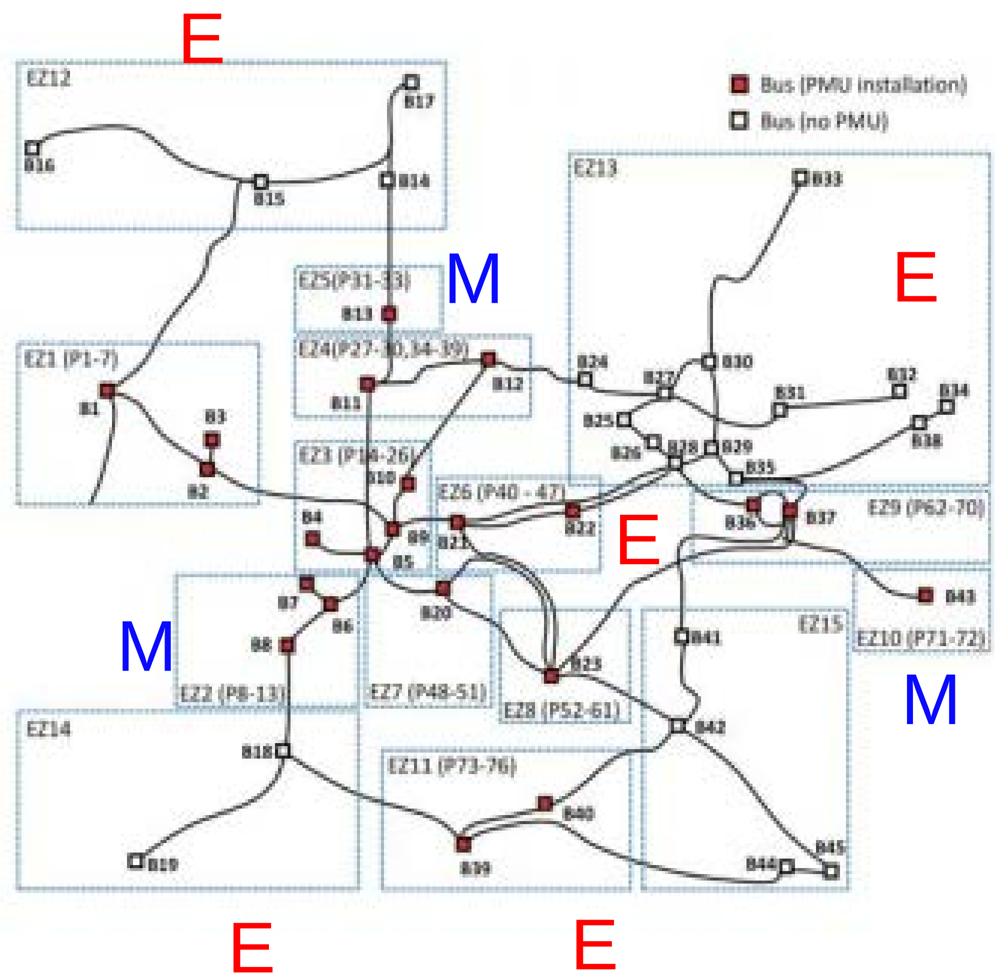
➢ Known model: $G_{ij}(\theta)$

between M-nodes and E-nodes;

➢ Extract nodal signatures;

➢ Run nodal model analysis at occurrence of each event;

➢ Estimated model: $\hat{G}_{ij}(\theta)$



➢ Look for signatures of event causing nodes

Thank you!