Literature Review

Deploying SOGNO services to optimise the grid using Network Topology Processor and State Estimation

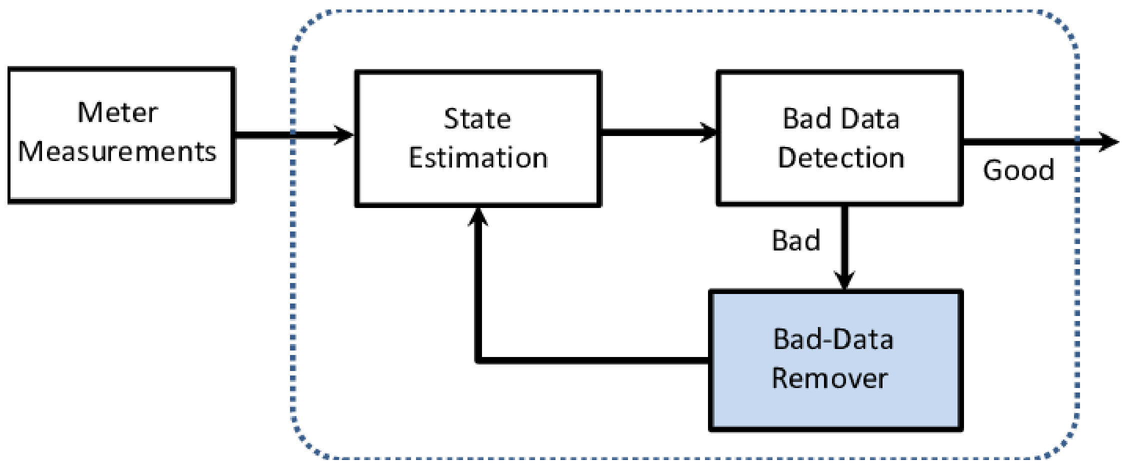
- Santhosh, Purvi

Contents

1. **Understanding EMS functionalities- NTP, SE**

- Network Topology Processor (NTP) is a system that is used analyse a network which typically constitute components that need to be monitored. And in the context of Power Systems, it is used to identify the topology (connectivity of the network) of the grid that is constituting the bus-sections and the switching devices (breakers and switches). This mainly focuses on logical data (the status of switching devices).

- While State Estimation (SE) aims to predict the current state of the network and the analog data (topological error- basic network issues like say while telemetry data transmission)



source: https://www.researchgate.net/figure/State-Estimator-for-a-power-system-21-22_fig2_281896001

- The conventional way is to approach the problem this way. Although newer algorithms have been found to optimise the topology processing that can support both conventional and PMU-only state estimators there are still some limitations.

  Refer: https://www.sciencedirect.com/science/article/abs/pii/S0378779613002241

2. Limitations to newer algorithms for Topology Processing

- **Complexity**: Some newer algorithms for topology processing may introduce increased computational complexity, especially when handling large-scale power systems with numerous nodes and branches. This complexity can lead to higher computational resource requirements and longer processing times.

- **Scalability**: Ensuring the scalability of topology processing algorithms to handle growing grid sizes and increasing data volumes from PMUs can be challenging. Scalability issues may arise when attempting to process and analyze large amounts of real-time data from distributed PMUs.

- **Data Quality**: Maintaining data quality is crucial for accurate topology processing. However, issues such as communication delays, synchronization errors, and data inaccuracies can still impact the effectiveness of newer algorithms.

- **Integration Challenges**: Integrating newer algorithms into existing EMS (Energy Management System) architectures and workflows may present challenges. Compatibility issues, data format conversions etc.

- **Efficiency**: Ensuring the efficiency of topology processing algorithms is essential for real-time grid monitoring and control. Algorithms must be resilient to data anomalies, network disturbances, and cyber threats to maintain grid stability and security.

- Based on my understanding from: https://sites.ecse.rpi.edu/~vanfrl/documents/publications/conference/2012/CP036_topologyprocessors_stateoftheart.pdf

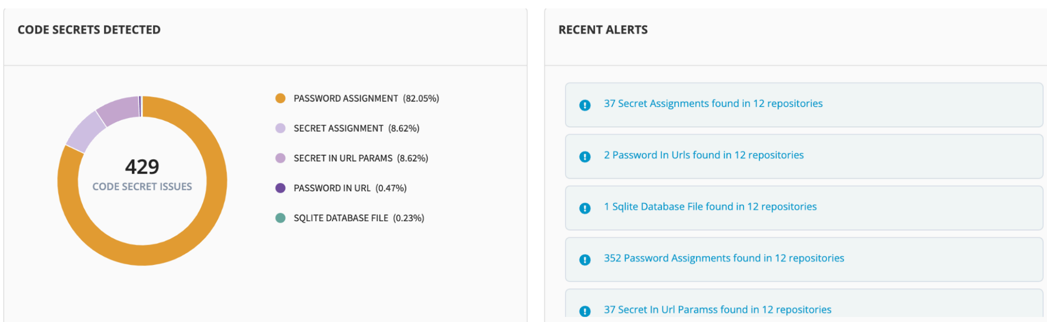
3. Advantages of Linux-based Energy Software Stacks for Grid Optimization

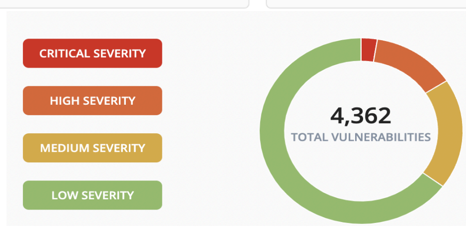
- **Flexibility**: allows you to customize the software to fit your specific needs, making it adaptable to various requirements.

- **Scalability and Performance**: can handle large amounts of data and processes efficiently, which is important for managing big and complex power grids.

- **Integration**: works well with different types of hardware and software, making it easier to connect and communicate with various parts of the energy system.

- **Security**: has strong security features to protect against cyber threats, helping to keep the power grid safe from attacks.





- **Community Support**: There is a large community of developers and users who contributed to SOGNO, offering support and sharing knowledge, which can help improve and innovate energy software solutions to various repositories that implement state estimations.

sources: https://security.lfx.linuxfoundation.org/#/a092M00001IkV6jQAF/overview , https://github.com/sogno-platform

4. Deploying SOGNO features to optimize the existing grid NTP methodology

STATE ESTIMATION USING PYVOLT:

(github url: https://github.com/sogno-platform/pyvolt.git) {nv_state_estimator.py}

<u>Based on my understanding of the code:</u>

```
def calculateJacobiMatrixSinj(measurements, nodes_num, Gmatrix, Bmatrix, inj_code, type):
    """
```

This function calculates the Jacobian matrix for power injection measurements, which is essential for linearizing the system equations around the operating point. By accurately computing the Jacobian matrix, the state estimation algorithm can iteratively converge to the most likely state of the system, improving accuracy and efficiency.

```
def calculateJacobiBranchPower(measurements, nodes_num, Gmatrix, Bmatrix, inj_code, type):
    """
```

This function computes the Jacobian matrix for branch power measurements, allowing for getting information about power flows through transmission lines. By including branch power measurements in the estimation process, the algorithm can better account for system constraints and topology, leading to more accurate results.

```
def calculateJacobiVoltagePmu(measurements, nodes_num, Gmatrix, Bmatrix, inj_code, type):
    """
```

These functions calculate the Jacobian matrices for voltage and current measurements obtained from Phasor Measurement Units (PMUs). PMU measurements provide high-resolution data in real-time, enabling the state estimator to quickly respond to changes in system conditions and disturbances. Inputing PMU measurements into the estimation process enhances the observability of the system and improves its ability to detect dynamic events.

```
def update_W_matrix(measurements, weights, W, type):
    """
```

This function updates the weight matrix used in the estimation process, incorporating information about measurement uncertainties. By adjusting the weights assigned to different measurements based on their reliability, the state estimator can give more weight to accurate measurements and downweight noisy or uncertain measurements, leading to more robust and reliable state estimates.

```
def update_h6_vector(measurements, V, iidx, nii, Yabs_matrix, Yphase_matrix, nodes_num, num_iter, inj_code, type):
    """
```

These functions update the measurement residual vectors based on the estimated state variables and measurement values. By iteratively refining the residual vectors, the state estimator can drive them closer to zero, indicating a better match between the observed and predicted measurements. This iterative refinement process improves the accuracy and convergence of the estimation algorithm.

- ⭐**It is evident that implementing the state estimation through the SOGNO packages is clearly efficient in getting better accuracy, reliable state estimates and ability to detect dynamic events than the conventional network topology processor, state estimation algorithms.**

5. References:

1. State-of-the-Art of Topology Processors for EMS and PMU Applications and Their Limitations
   Mostafa Farrokhabadi and Luigi Vanfretti
2. A. Monticelli
   State Esitmation in Electric Power Systems: A Generalized Approach