UNIVERSITY OF CALIFORNIA
RIVERSIDE

# Physics-Informed Machine Learning Models for Power Transmission Systems

A Thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science

in

Computer Science

by

Xianghao Kong

June 2022

Thesis Committee:

Nanpeng Yu, Chairperson
Stefano Lonardi
Ahmed Eldawy

The Thesis of Xianghao Kong is approved:

Committee Chairperson

University of California, Riverside

## Acknowledgments

Throughout the writing of this dissertation I have received a great deal of support and assistance.

I would first like to thank my advisor, Prof. Nanpeng Yu, whose expertise was invaluable in formulating the research questions and methodology. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

I would particularly like to acknowledge my teammates, Yuanbin Cheng, Jie Shi, Yuanqi Gao, Brandon Foggo and Koji Yamashita, for their wonderful collaboration and patient support.

I would also like to thank my family for their wise counsel and sympathetic ear. You are always there for me.

Finally, I could not have completed this dissertation without the support of my friends, Jinghan Yao and Lin Cong who provided stimulating discussions as well as happy distractions to rest my mind outside of my research.

The content of this thesis is a re-print of the materials that are appeared in the following publications:

- Xianghao Kong, Brandon Foggo, Koji Yamashita, and Nanpeng Yu, Online Voltage Event Detection Using Synchrophasor Data with Structured Sparsity-Inducing Norms, IEEE Transactions on Power Systems, 2021. (Chapter 2)

- Xianghao Kong, Koji Yamashita, Brandon Foggo, and Nanpeng Yu, Dynamic Parameter Estimation with Physics-based Neural Ordinary Differential Equations, IEEE Power and Energy Society General Meeting, 2022. (Chapter 3)

To my advisor and my family for all the support.

# ABSTRACT OF THE THESIS

Physics-Informed Machine Learning Models for Power Transmission Systems

by

Xianghao Kong

Master of Science, Graduate Program in Computer Science
University of California, Riverside, June 2022
Nanpeng Yu, Chairperson

In the past few decades, the rapid development of the United States power system has led to the continuous expansion of transmission networks and an increasing number of phasor measurement units (PMUs) have been deployed on the power system. Although voltage and current phasor data can be obtained in a real-time operation environment, it is still challenging to effectively utilize PMU data in a large distributed system. Simply using off-the-shelf machine learning algorithms to process PMU data does not yield models with sufficient performance in practice. In this thesis, the physical dynamics of the U.S.power system was synergistically combined with machine learning to monitor and model a power transmission system.

The first aspect was real-time data-driven power system monitoring. We developed an efficient data-driven framework to detect voltage events from PMU data streams. In particular, we developed an innovative Proximal Bilateral Random Projection (PBRP) algorithm to quickly decompose a PMU data matrix into a low-rank matrix, a row-sparse event-pattern matrix, and a noise matrix. The row-sparse pattern matrix significantly distinguishes events from normal behavior. These matrices were then fed into a clustering algorithm to separate voltage events from normal operating conditions. Large-scale numerical study results on real-world PMU data show that the proposed algorithm achieved higher F1 and F2 scores with 50% less computation time.

The second aspect was to model dynamic electric power generator parameters. Accurate estimation of dynamic parameters is crucial to building a reliable model for dynamical studies and reliable operation of the U.S. power system. A physics-based neural ordinary differential equations (ODE) approach was developed to learn the generator dynamic model parameters using PMU data. We designed a physics-based neural network to represent the swing equations of the power system dynamics. The parameters of the generator dynamic model were iteratively updated using the neural ODEs and the adjoint method. By exploiting the mini-batch scheme in neural ODE training, the parameter estimation performance was significantly improved with more than 50% computation speed up.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The advent of phasor measurement units (PMUs) provides system operators with time-synchronized voltage and current phasor measurements in real-time [7]. The widespread deployment of PMUs around the world enables the development of data-driven algorithms to estimate the dynamic parameters of power transmission systems [27, 49], detect power system events [6, 26], replace missing values [12], classify power system events [43, 44], and identify power system event signature [45]. Since the PMU data are generated from a physical network, the off-the-shelf machine learning algorithms often do not perform well on power system data. To improve the computation efficiency and accuracy of the machine learning algorithms, we developed physics-informed machine learning algorithms to monitor and model power transmission systems in real time.

In this chapter, we will introduce the background of big data in power transmission systems (Section 1.1); describethe technical challenges (Section 1.2); summarize the contributions of this thesis (Section 1.3); and finally, provide an organization of this thesis

(Section 1.4).

## 1.1 Background

Phasor Measurement Units (PMUs) are capable of recording both amplitudes and angles of voltage and current phasors. Time synchronization is provided by GPS. PMUs have been widely deployed in transmission networks worldwide to improve the situational awareness of power system operators [35]. The rapid adoption of PMUs in transmission grids has led to tremendous growth in the amount of synchrophasor data (Fig. 1.1). For example, with a 60 Hz sampling rate, each PMU can generate more than one gigabyte of data per day [46]. The wealth of PMU data enables a development of various data-driven applications to better monitor, model, protect, and control power systems.

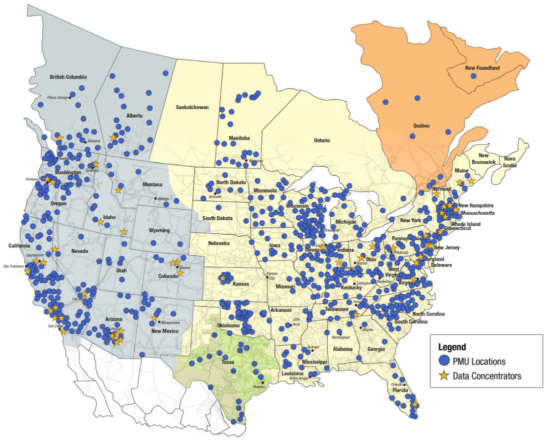


Figure 1.1: The deployment network of PMU in 2015 [1].

## 1.2 Technical Challenges

In this subsection, we present technical challenges associated with real-time power system event detection and dynamic parameter estimation.

### 1.2.1 Online Voltage Event Detection

The existing methods of using synchrophasor data for data-driven power system event detection can be roughly divided into three groups: traditional signal processing techniques [23, 24, 33, 42]or statistical analysis [4], deep learning techniques [43, 48, 55], and low-rank approximation methods [14,15,19,31,37]. However, these methods have their own pitfalls. Most signal processing techniques analyze PMU data streams collected from different locations separately and did not fully exploit underlying spatial-temporal correlations in the PMU dataset. Despite the high accuracy achieved by deep learning techniques, their success heavily depends on the availability of numerous high-quality event labels rarely available in practice. The PMU data decomposition methods by low-rank property are usually oversimplified and ignore the unique structural pattern of events.

### 1.2.2 Dynamic Parameter Estimation in Generators

Two representative works in this field formulate dynamic power system parameter estimation into nonlinear least-squares problem [20] and black-box neural network problem [38], respectively. The former may not be accurate enough for bulk power system disturbances, and the latter leads to low estimation accuracy due to the lack of power system domain knowledge in a neural network. There are also some works to solve this problem using variants of Kalman filter [11, 16] to estimate parameters, and statistical models like Bayesian approach [34] or Markov chain Monte Carlo method [51]. These methods either have low estimation accuracy or fall into a local optimum.

## 1.3 Contributions

### 1.3.1 Online Voltage Event Detection

This work investigated power system anomaly detection algorithms in streaming PMU data to issue an early alert to system operators. Mining the real-world data gathered by hundreds of PMUs, covering thousands of events across the U.S., we discovered that regional events mostly show a unique sparse structure in a noise matrix. Unique contributions of this work include:

- The proposed iterative matrix decomposition approach, PBRP, which greatly accelerates the solution of a general low-rank and sparse matrix decomposition problem where the residual matrix has a row-sparse structure.

- The data-driven event detection framework based on PBRP yields better theoretical and empirical computation efficiency than existing SVD-based subspace characterization approaches

- Unlike the model-based or supervised deep learning methods, our proposed voltage event detection algorithm does not rely on detailed physical model or a large amount of event labels for training.

- The numerical study on a large-scale real-world PMU dataset with hundreds of PMUs and voltage events shows that the PBRP-based event detection framework provides higher F1 and F2 scores than state-of-the-art algorithms. The proposed algorithm can also estimate event area/location along with PMUs that are sensitive to an event.

### 1.3.2 Dynamic Parameter Estimation in Generators

This work extended the prior research [34, 51] by converting a forward solver of the ODEs representing power system dynamics into physics-informed neural networks. The main contributions of this work are as follows:

- We adopt neural ODEs and the corresponding adjoint method to learn the parameters of dynamic generator models online, which provides accurate estimates for parameter gradients.

- By designing physics-based neural networks to represent the forward functions of ODEs, we are able to leverage the parallel computing capabilities of graphics processing units (GPUs) to accelerate the dynamic parameter learning. This advantage becomes more apparent as the grid size increases.

- By leveraging the mini-batch scheme in updating dynamic parameters, the estimation time can be shortened and the buildup of errors in the ODE solver can be reduced.

- Comprehensive numerical studies demonstrate that our proposed method can accurately estimate not only the inertia constant but also mechanical power inputs using PMU data during transmission line events.

## 1.4 Organization

The rest of the thesis is organized as follows. In Chapter 2, the online voltage event detection problem is solved based on a low-rank and sparse matrix decomposition method to extract effective features out of streaming data. In Chapter 3, critical parameters in power system dynamic models are estimated with the help of Neural ODE framework. The whole dissertation is concluded in the Chapter 4.

# Chapter 2

# Online Voltage Event Detection Using Synchrophasor Data with Structured Sparsity-Inducing Norms

Data-driven event detection algorithms are critical to making system operators aware of abnormal system conditions [23]. The early detection of power system events enables the operators to take corrective control actions in response to disturbance events. In this section, we investigate practical anomaly detection algorithms based on efficient low-rank and sparse matrix decomposition model in the streaming data systems to issue an early alert, which helps reduce maintenance costs.

## 2.1 Related Works

The existing literature on the data-driven power system event detection using synchrophasor data can be clustered into three groups. The first group of literature leverages signal processing techniques and statistical analysis to detect system events. Signal processing techniques such as discrete wavelet decomposition [23], dissipating energy flow [24], empirical mode decomposition [33], self-coherence spectrum [56], Teager-Kaiser energy operator [52], dynamic programming-based swinging door trending [8], graph signal processing [42] [10], and parallel detrended fluctuation analysis [22] are adopted to detect oscillation events [56], voltage events [23, 33, 42], frequency events [22, 23, 33, 42] and sequence of voltage and frequency events [52]. A multi-hypothesis statistical testing framework is developed in [4] to detect power line outages. This group of techniques has achieved great success at various types of power system events. However, most signal processing techniques analyze PMU data streams collected from different locations separately and did not fully exploit underlying spatial-temporal correlations in the PMU dataset.

The second group of papers adopts deep learning techniques to detect and classify abnormal events. Convolution neural networks (CNN)-based classifiers [48] are built using the rate of change of frequency and relative angle shift signals to detect and classify generator trip and load disconnection events. A novel information loading based regularization and a graph signal processing-based PMU sorting algorithm were developed to improve the parameter sharing scheme of the CNN framework [43]. An ensemble-based learning algorithm, combining multiple machine learning algorithms, is proposed in [55]. Despite the high accuracy achieved by deep learning techniques, their success heavily depends on the availability of numerous high-quality event labels rarely available in practice.

The third research group leverages the low-dimensionality and approximates low-rank properties of PMU data to detect power system events. Recognizing that high dimensional PMU data lie close to a low dimensional manifold, principle component analysis (PCA)-based event detection algorithms have been developed [15]. Two statistics derived from a moving window PCA on PMU data matrices are used to detect frequency and islanding events [37]. It has been shown that pilot PMUs identified by the PCA can be used to approximate non-pilot PMUs' data streams. An event alert will be issued when the normalized approximation error is larger than a pre-specified threshold [50]. In [31], a PCA-based method is developed to measure the similarity of operation states between a pair of buses, and the $k$-reachability is adopted to detect power system events. By exploiting the approximate low-rank property of PMU data matrices, subspace characterization [29] and matrix completion-based [14, 19] approaches have been proposed to detect power system events. The PCA and matrix completion-based event detection algorithms model PMU data as the sum of a low-rank matrix and a noise matrix. They overcome the shortcomings of algorithms in the first two groups and do not require a large number of event labels for training. However, the PMU data decomposition methods in the third group are oversimplified and ignore the unique structural pattern of events.

Meanwhile, sparsity-inducing norms have been widely adopted in other power grid fields. Examples include malicious cyber attack detection [13,18,30], imbalance identification [40] and line outage detection [58]. Routtenberg et al. [40] successfully leveraged the sparse structure in voltage measurements to localize imbalances in the power grid. Zhu et al. [58]

leverage sparse overcomplete representations with $l_1$ norms to identify sparse power line outages. Liu et al. [30] propose a novel matrix decompsition method based on $l_1$ norm to detect the malicious attacks in the power grid. Hao et al. [18] assume the measurement matrix can be recovered by a low-rank matrix and a sparse-attack matrix and apply $l_1$ norm to induce sparsity. The $l_1$ norm only constraints a matrix to be element-wise sparse, i.e., non-zero elements appear randomly in the matrix. This sparsity-inducing constraint is too weak to cater a structured sparsity, like row-sparse or column-sparse. To break this limitation, Gao et al. [13] proposed an attack identification algorithm with the $l_{21}$ norm, assuming that the measurement data matrix is a low-rank matrix plus a transformed column-sparse matrix. This time, the sparse matrix is not directly stripped from the original measurement matrix, but needs to be multiplied by a known transform matrix. To construct the transform matrix, we need to know the structure of the power grid in advance, as well as the impedance and admittance between different buses, which is hard to obtain in practice. Most of the algorithms in this group rely on singular value decomposition (SVD), whose computation time drastically increases with the number of PMUs and the analysis window length. This drawback dramatically limits the scalability of the event detection algorithm. Our research develops a novel method that decomposes the PMU measurement matrix into a low-rank matrix, a sparse matrix, and a noise matrix without a transform matrix in a computationally efficient manner.

## 2.2 Online Voltage Event Detection

Mining the real-world data gathered by hundreds of PMUs, covering thousands of events across the U.S., discovers that regional events mostly show a unique sparsity in the noise matrix (Fig. 2.2). As the row of the noise matrix corresponds to individual PMUs, the sparsity emerges in its row space, depending on how sensitive a PMU is to events. Among a wide variety of power system events, voltage-related events mainly triggered by system faults are recognized as a regional event. Therefore, the row-sparse property in the noise matrix can exert an effect, especially in an event with a significant voltage dip. This paper proposes further decomposing the noise matrix into a row-sparse event-pattern matrix and a pure noise matrix, in light of the above. This innovative low-rank and sparse matrix decomposition framework, extracting anomaly features from both the low-rank matrix and the row-sparse event-pattern matrix, enables event alerts. Finally, an unsupervised clustering technique is adopted to distinguish normal system operation data from that of the power system voltage events.

### 2.2.1 Overall Framework

The overall framework of the proposed online voltage event detection algorithm is summarized in Fig. 2.1. The proposed algorithm has three modules: a streaming matrix decomposition module, an anomaly feature extraction module, and an event detection module based on cluster analysis. The first modu le separately decomposes four types of streaming PMU data matrices in $w_1$-length windows. Each type of data matrix ($X$) would be decomposed into a low-rank matrix ($L$), a sparse event-pattern matrix ($S$), and a noise

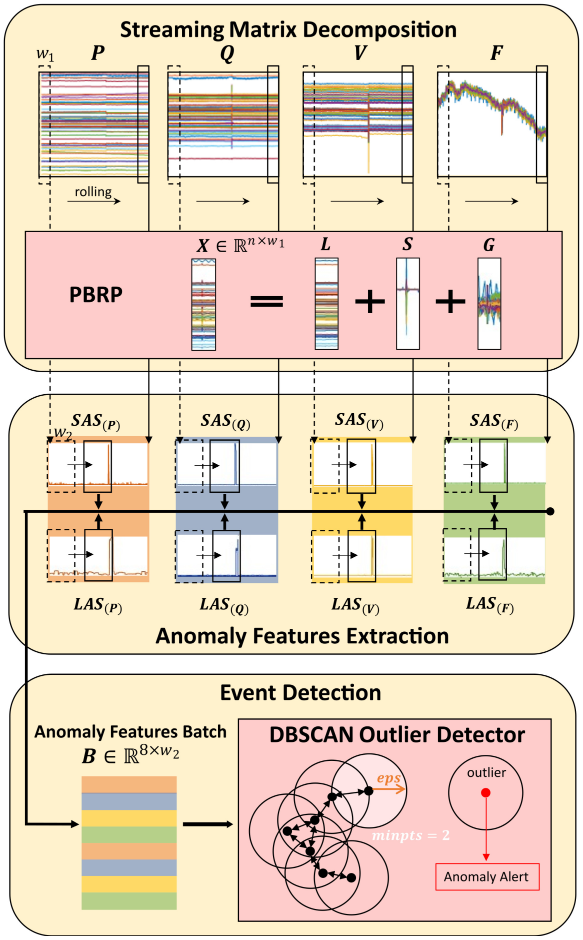


Figure 2.1: Overview of online event detection framework.

matrix ($G$) via PBRP algorithm. The second module extracts useful features, two anomaly scores, $LAS$ and $SAS$ from decomposed matrices, $L$ and $S$. The third module performs cluster analysis on extracted features within $w_2$-length windows to identify anomalies.

### 2.2.2 Low-Rank and Sparse Event-Pattern Matrix Decomposition

Let $n$ denote the number of PMUs under consideration. We collect the streaming PMU data into a matrix time series $X_t \in \mathbb{R}^{n \times w}$ by placing new instances of data in the rightmost column while removing the leftmost column. These matrices are decomposed in a specific way based on prior knowledge of their properties.

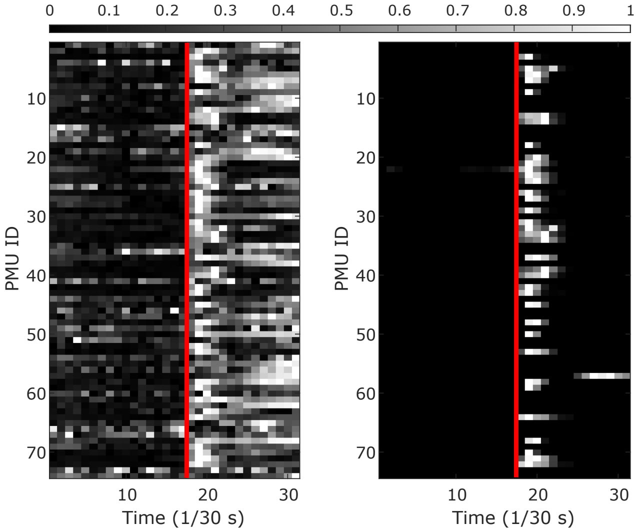


Figure 2.2: The heatmap of “$X - L$" (left) and “$X - L - G$” (right) for active power data.

The first property of note has to do with the rank of these matrices. As shown by small-scale empirical studies and theoretical derivations, voltage and current phasor data under normal conditions exhibits a low-rank structure [14]. Using large-scale PMU data from the Eastern Interconnection of the continental U.S. power transmission grid, the active power (P), reactive power (Q), voltage magnitude (V), and frequency (F) data does possess the low-rank property, which means the low-rank property holds up well during steady-state operation periods. In Table 2.1, the analysis on a representative event shows that the largest singular value of the reactive power data matrix accounts for 99.988% of the variance, while this percentage drops to 59.743% during event periods. Suppose that this normal behavior from the streaming PMU data matrices is decomposed as:

Table 2.1: Singular Value Decomposition of P, Q, V, and F data matrices over 1 second

| **Data Type** | **Electrical Quantity** | **Singular Value Percentage Variance** ($\frac{\sigma_i^2}{\sum_i \sigma_i^2}$) | | | **Singular Value Proportion** ($\frac{\sigma_i}{\sum_i \sigma_i}$) | | |
|---|---|---|---|---|---|---|---|
| | | **1st** | **2nd** | **3rd** | **1st** | **2nd** | **3rd** |
| Non-event Data | P (Active Power) | 99.999261% | 0.000536% | 0.000109% | 99.242040% | 0.229736% | 0.103609% |
| | Q (Reactive Power) | 99.988472% | 0.008789% | 0.001427% | 97.134683% | 0.910695% | 0.366895% |
| | V (Voltage Magnitude) | 99.999995% | 0.000005% | 0.000000% | 99.963393% | 0.022302% | 0.003274% |
| | F (Frequency) | 99.999999% | 0.000000% | 0.000000% | 99.996304% | 0.000824% | 0.000654% |
| Event Data | P (Active Power) | 95.003242% | 4.933310% | 0.045058% | 78.391273% | 17.863547% | 1.707207% |
| | Q (Reactive Power) | 59.743182% | 40.185730% | 0.068034% | 53.278058% | 43.695845% | 1.797904% |
| | V (Voltage Magnitude) | 99.545736% | 0.447828% | 0.006371% | 92.892350% | 6.230519% | 0.743139% |
| | F (Frequency) | 99.999994% | 0.000006% | 0.000000% | 99.971389% | 0.023498% | 0.001269% |

$$X = L + (X - L), \tag{2.1}$$

where $L$ is an approximation of $X$ with rank $r$. Then the matrix, $X-L$, contains information from the data that is residual from normal behavior, which is a promising first step towards event detection.

The second property of note has to do with the structure of $X - L$ during voltage event periods. It turns out that these matrices have specific patterns of sparsity that we can take advantage of. The main component of this structure comes directly from the fact that voltage events, when they occur, often significantly affect limited area/zones. As such, the number of PMUs interacting with a voltage event is prone to be limited (Fig. 2.2). The right subfigure shows the heatmap of an event pattern matrix representing one-second min-max normalized active power data. The event happens approximately at the red line.

Thus, we propose to decompose the PMU data matrix as follows:

$$X = L + S + G, \tag{2.2}$$

where $L \in \mathbb{R}^{n\times w}$ is a low-rank matrix, $S \in \mathbb{R}^{n\times w}$ is a row-sparse event-pattern matrix representing the impact of voltage events, and $G \in \mathbb{R}^{n\times w}$ denotes a noise matrix. Then, the problem can be formulated as:

$$\begin{aligned} \min_{L,S} \quad & \frac{1}{2}\|X - L - S\|_F^2 \\ s.t. \quad & \begin{cases} \mathit{rank(L)} = r, \\ \mathit{S\ is\ row\text{-}sparse.} \end{cases} \end{aligned} \tag{2.3}$$

It turns out that this “row-sparse" event pattern can be captured by using $l_{21}$ regularization, where the $l_{21}$ norm of $S$ is defined as $\|S\|_{21} = \sum_i \sqrt{\sum_j s_{ij}^2}$. In other words, by adding the $l_{21}$ norm on event-related matrix, $S$, to the objective function as a penalty term, the solution will yield the desired row-sparse structure. The problem can then be relaxed into a new one:

$$\begin{aligned} \min_{L,S} \quad & \frac{1}{2}\|X - L - S\|_F^2 + \lambda\|S\|_{21} \\ s.t. \quad & \mathit{rank(L)} = r, \end{aligned} \tag{2.4}$$

where the $\lambda$ is a penalty coefficient of the $l_{21}$ norm. This optimization problem (2.4) can be solved with Coordinate Descent. This means that we alternate between solving the following two sub-problems until $\frac{\|X-L^{(k)}-S^{(k)}\|_F^2}{\|X\|_F^2}$ converges:

$$\begin{cases} L^{(k)} = \text{argmin}_{rank(L)=r}\, \frac{1}{2}\|X - L - S^{(k-1)}\|_F^2 \\ S^{(k)} = \text{argmin}_S\, \frac{1}{2}\|X - L^{(k)} - S\|_F^2 + \lambda\|S\|_{21} \end{cases} \tag{2.5}$$

where $L^{(k)}$ and $S^{(k)}$ denote an estimate of $L$ and $S$, respectively, in the $k$-th iteration.

To solve the first sub-problem, we forgo time-consuming exact optimization and instead choose to approximate its solution via an enhanced version of Bilateral Random Projections (BRPs).

### 2.2.3 Bilateral Random Projections

Bilateral Random Projections (BRPs) are a fast and accurate method of low-rank matrix approximation. We showcase a new approximation approach here. First, consider the following column-row-echelon decomposition of a matrix $X \in \mathbb{R}^{n\times w}$ that has a rank $r$.

$$X = \underbrace{\begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_r \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} I_r \mid E_{w-r} \end{bmatrix}}_{E} P, \tag{2.6}$$

where $\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_r$ are any choice of $r$ linearly independent columns of $X$, $I_r \in \mathbb{R}^{r\times r}$, $E_{w-r} \in \mathbb{R}^{r\times(w-r)}$ and $P \in \mathbb{R}^{w\times w}$ acts on $E$ as a column permutation matrix ($EP$ is the reduced row-echelon form of $X$, and $P$ just moves the pivot columns of that echelon form back to their original positions). This decomposition expresses a matrix as a selection of $r$ linearly independent columns from $X$ and uses the matrix $E_{w-r}$ to generate the remaining columns from the selected ones. We can also express this same decomposition as $X = (XA_1)EP$ where $A_1$ acts on $X$ to select those $r$ independent columns. If we then use a matrix $A_2^T$ to select $r$ linearly independent *rows* of $X$ and collect them into a matrix $H^T$ (via left-application, i.e., $H^T = A_2^T X$), then we have that

$$A_2^T X = A_2^T CEP. \tag{2.7}$$

The left hand side of (2.7) has rank $r$, and $EP$ has rank $r$ as well due to its $I_r$ submatrix, so it must be the case that $A_2^T C$ has rank $r$ as well. But since $A_2^T C$ has dimension $r \times r$, this means that $A_2^T C$ is invertible. Thus $EP = (A_2^T C)^{-1} A_2^T X$. In other words:

$$X = CEP = C(A_2^T C)^{-1}(A_2^T X) = C(A_2^T C)^{-1} H^T. \tag{2.8}$$

To get from here to BRPs, all we need is relaxing the requirement that the matrices $A_1$ and $A_2$ choose $r$ linearly independent columns/rows of $X$, and instead let them randomly choose subspaces via random linear combinations of the columns/rows of $X$. This can be done by simply drawing the elements of $A_1$ and $A_2$ randomly, according to some distribution (in this case, the standard normal distribution). The Johnson–Lindenstrauss lemma [9] ensures that such random selection well approximates the process of selecting independent rows and columns of $X$. In accordance with [57], we enhance this approximation algorithm with a *power scheme* technique. The idea of this enhancement is simple — instead of approximating $X$ itself, $\tilde{X} = (XX^T)^q X$ for some integer $q \geq 1$ are approximated. This leads to a higher likelihood of randomly selecting the most important low-rank approximations because the matrix $\tilde{X}$, while having the same row and column spaces as $X$, has its singular values exponentiated to the power of $2q + 1$, which leads to a much higher discrepancy between the larger and smaller singular values. Once we have an approximation of $\tilde{X}$ given by $\tilde{X} \approx \tilde{L} = C(A_2^T C)^{-1} H^T$, we can use the same approximated subspaces of $\tilde{X}$ to get an approximation of $X$ via $X \approx L = \tilde{L}^{\frac{1}{2q+1}}$. This can be done efficiently by taking the QR decomposition of $C$ and $H$, denoted $Q_{col}R_{col}$ and $Q_{row}R_{row}$ respectively, and computing

$$L = \tilde{L}^{\frac{1}{2q+1}} = Q_{col} \left[ R_{col}(A_2^T C)^{-1} R_{row}^T \right]^{\frac{1}{2q+1}} Q_{row}^T \tag{2.9}$$

Finally, the aforementioned algorithm is iteratively performed using the previous iterations' $C$ matrix as the projection matrix for the next iterations' $H$ matrix and vice versa. We couple these iterations with an adaptive rank reduction scheme. This is done so that if a too high rank, $r$, as input is chosen, an even lower rank approximation can still be found if a good one exists. Specifically, at the end of each iteration, the enhanced algorithm checks if the rank of the combined row-space and column-space containing matrix, $A_2^T \tilde{X} A_1$, is below our rank parameter. If so, the algorithm reduces the rank parameter and continues the loop. However, if the rank of this matrix fail to reduce this iteration, the iteration is terminated. The entire algorithm can be found in Algorithm 1.

In Algorithm 1, it is worth mentioning that the power scheme is strong enough when $q \geq 3$ according to [57]. It costs $(2q+1)nwr$ floating-point operations (flops) to perform two projections and $r^2(n+w)$ flops to perform each QR decomposition. The matrix division in (2.9) requires an additional $nwr + 2wr^2 + 4r^3$ flops. In general, the rank $r$ is much smaller than $n$ or $w$. Thus, the computational complexity of the BRP-based decomposition method is $O(nwr)$. This is far faster than the traditional matrix decomposition based on the SVD, whose computational complexity is $O(\min(nw^2, n^2w))$ [25].

### 2.2.4 Proximal Methods

The second sub-problem in (2.5) has the form of

$$\min_{\boldsymbol{S} \in \mathbb{R}^{n \times w}} f(\boldsymbol{S}) + \lambda \Omega(\boldsymbol{S}), \tag{2.10}$$

where $f : \mathbb{R}^{n \times w} \to \mathbb{R}$ is a convex function, and $\Omega : \mathbb{R}^{n \times w} \to \mathbb{R}$ is a sparsity-inducing norm. In this problem, F is half the square of the Frobenius norm of the difference between $X - L$ and $S$, and $\Omega$ is the square of the $l_{21}$ norm. This is exactly the form of minimization that defines the *proximal operator* [2] given by:

$$Prox_{\lambda\Omega}(\boldsymbol{u}) = \underset{\boldsymbol{v}\in\mathbb{R}^w}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{v}\|_2^2 + \lambda\Omega(\boldsymbol{v}). \tag{2.11}$$

In this case, the $\boldsymbol{u}$ variable found in the generic proximal operator definition can be replaced with $X - L$, and $\boldsymbol{v}$ with $S$. In the specific case where $\Omega(\cdot)$ is the $l_{21}$ norm, (2.11) is called “group Lasso" [2], and the proximal operator for $l_{21}$ is

$$Prox_{\lambda\|\cdot\|_{21}}(X[i,:]) = (1-\lambda\|X[i,:]\|_2)_+ X[i,:], \tag{2.12}$$

where $X[i,:] \in \mathbb{R}^w$ is the $i$-th row of $X$, $\|X[i,:]\|_2 = \sqrt{\Sigma_j x_{ij}}$, and $(\cdot)_+ \triangleq max(\cdot, 0)$. From (2.12), we can see that the proximal operator maps a row of the target matrix at once.

### 2.2.5 Proximal BRP Algorithm

We call the proposed iterative approach to solve the matrix decomposition problem (2.5), *Proximal BRP* (*PBRP*). It is summarized in the Algorithm 2.

### 2.2.6 Feature Engineering and Anomaly Detection with DBSCAN

After individually obtaining the low-rank matrix $L$ and the sparse matrices associated with P, Q, V, and F, we take useful measurements from them - which we will call *anomaly scores*. Since the sparse matrix $S$ represents the event patterns, its $l_{21}$ norm is selected as the first anomaly score. We will call this score ‘$SAS$’ ($S$-Anomoly Score) for short. Furthermore, the sparsity of $S$ intuitively comes from the regional variance of events, and thus, $SAS$ summarizes the spatial features of the data. Drawing on the idea in [14], the maximum temporal difference of $L$ is treated as the second anomaly score:

$$LAS \triangleq \max_{i,j} \left( \left| \frac{L[i,j] - L[i,j-1]}{L[i,j-1]} \right| \right), \tag{2.13}$$

which summarizes the temporal features of the data.

After accumulating the two anomaly scores of P, Q, V, and F for $w_2$ time steps, an anomaly feature batch, $B \in \mathbb{R}^{8 \times w_2}$ is obtained. This feature batch can be regarded as a time series of length $w_2$ containing 8-dimensional feature vectors. Since we intend to build an unsupervised framework, we need an unsupervised algorithm to do classification on $B$. There are two widely used unsupervised clustering algorithms: k-means and DBSCAN. Compared with k-means, DBSCAN is more conducive to the detection of outliers [41]. DBSCAN is a clustering algorithm based on density, so it is more suitable for various complex-shaped datasets, while k-means is mainly intended for convex datasets with spherical distributions. In DBSCAN, there are three types of points: *core* points, *density-reachable* points, and *outliers*. They are defined as follows: If a point has at least '*minpts*' other points in its '*eps*' neighborhood, it is a *core* point. The points that are in the '*eps*' neighborhood of a *core point* are called a *density-reachable* point. Points that are neither core nor density reachable are identified as *outliers*.

In our proposed framework, we apply DBSCAN on $B$ every time it is updated. If the 8-dimensional feature vector corresponding to time $t$ is identified as the first outlier by the DBSCAN in the time series, the voltage event is deemed to occur at time $t$.

### 2.2.7 Summary of the Overall Event Detection Framework

The proposed voltage event detection framework is summarized in Algorithm 3, and illustrated in Fig. 2.1. For each time window $w_1$ and each data type $l$, the anomaly score ($AS$) that includes $SAS$ and $LAS$ is obtained. These anomaly scores are calculated based on $L$ and $S$ that are the outputs of the PBRP algorithm. Since P, Q, V, and F data streams have different scales, the corresponding penalty coefficients $\lambda_l$ are adjusted separately and adaptively in Lines 4-9. When the maximum relative standard deviation (RSD) $\frac{std_i}{\mu_i}$ for PMUs in $S$ exceeds a threshold $\theta$, $\lambda_l$ increases. The larger the RSD, the greater the degree of signal dispersion - thus requiring the weight of $S$ to be larger to alert an event. Correspondingly, $\lambda_l$ increases. However, if the $l_{21}$ norm of $S$ becomes zero, which means that no event pattern exists in $X$, $\lambda_l$ decreases. After all anomaly scores within a time window of $w_2$ are calculated, we construct the anomaly feature batch $B$ and apply the density-based cluster analysis on it to identify potential voltage events.

## 2.3 Numerical Study

The proposed event detection algorithm is validated using real-world PMU datasets that were recorded following voltage events. The online algorithm for PMU data processing and disturbance detection [14] and [19] are selected as benchmark algorithms.

### 2.3.1 Dataset Description

The PMU dataset is collected from the Eastern Interconnection of the U.S. power transmission grid. The dataset includes P, Q, V, and F readings from 187 PMUs with a

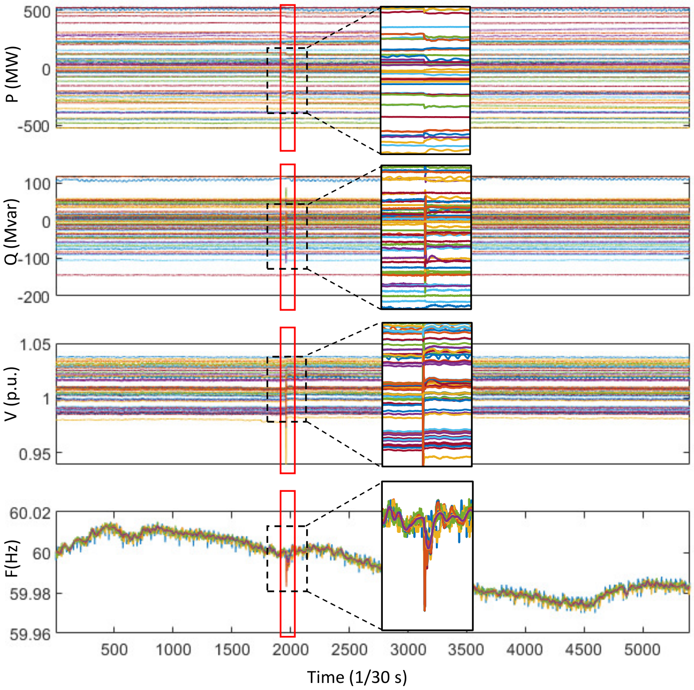


Figure 2.3: Voltage event example with four electric quantities.

sampling frequency of 30 Hz covering 668 labeled voltage events. Each event contains 3 minutes or 5400 samples of data (Fig. 2.3). Event datasets are divided into a validation dataset with 80% of event samples and a testing dataset with 20% of event samples. Our algorithm does not require training. The validation dataset is only used to determine the hyper-parameters of the entire framework, and the testing dataset helps evaluate the final performance.

### 2.3.2 Benchmark Method

The online algorithm for PMU data processing (OLAP) [14] and OLAP with a Hankel matrix (HOLAP) [19] are two state-of-the-art power system event detection methods. The Hankel matrix is more sensitive to the change in temporal correlation of the time series, but its size is larger. Both methods leverage the low-rank property of their target matrices (the original data matrix in OLAP and the data matrix constructed with a Hankel structure in HOLAP). OLAP computes the ratio of the first two singular values, denoted $\zeta$, in a short time window and detects system disturbances by using changes in this ratio. HOLAP computes the rank-1 approximation error of the original Hankel matrix as well as that of a permuted version of itself, and identifies an event by using the change in the ratio of these two errors. This later ratio is denoted as $\eta$. To make a fair comparison, OLAP and HOLAP are embedded into our proposed framework to replace PBRP by using $\zeta$ and $\eta$ as anomaly scores. In particular, Lines 3-9 in Algorithm 3 are replaced with the two comparison algorithms.

### 2.3.3 Hyper-parameter Settings

Important hyper-parameters of three algorithms are summarized in TABLE 2.2. The OLAP and HOLAP adopt the optimal parameters identified in [14] and [19] respectively. The DBSCAN module calculates Euclidean distance between individual samples. Only the *eps* is fine-tuned because all the algorithms are only sensitive to this hyper-parameter.

### 2.3.4 Numerical Results

We rely on the validation dataset to determine the optimal value of the hyper-parameter '*eps*' in DBSCAN. At the initial time $t$, the P, Q, V, F data in $w_1$-length window are respectively passed through the PBRP algorithm (Section 2.2.5) to obtain their corresponding $L$ and $S$. Then, we extract anomaly features $SAS$ and $LAS$ from the $L$ and $S$

Table 2.2: Hyper-Parameter Settings

| **Algorithm** | Window | PBRP | OLAP | HOLAP | DBSCAN |
|---|---|---|---|---|---|
| **Components** | Length | | | | |
| | $w_1 = 30$ | $\epsilon = 0.001$ | $\bar{w} = 5$ | $\kappa = 5$ | $minpts = 2$ |
| **Parameter** | $w_2 = 300$ | $\lambda = 10$ | | | $eps : 2 \sim 13$ |
| **Values** | | $r = 5$ | | | |
| | | $q = 5$ | | | |

matrices (Section 2.2.6), generating an 8-dimensional feature vector at time $t$. Packing it with $w_2 - 1$ feature vectors before time $t$ into a batch $B$, and send $B$ to DBSCAN for clustering analysis. If there is an outlier point, an alert will be issued. The result of the above process is shown in Fig. 2.4. After getting the best '*eps*', we use the best hyper-parameters to evaluate the final performance of our voltage event detection framework on the testing dataset.

### Performance of Voltage Event Detection Framework

Two commonly used evaluation metrics in classification problems, F1 and F2 scores, are used to evaluate the performance of our voltage event detection framework. The two F-scores are calculated based on *precision* and *recall* that are further derived based on True Positive (TP), False Positive (FP), and False Negative (FN). The TP means events are detected within 1 second of the labeled event time. The FP denotes the scenarios where the algorithm reports an event outside the above 1 second time window. The FN comprises the cases where no event is detected within the aforementioned 1 second time window.

*Recall* and *precision* are calculated as: $Recall = \frac{TP}{TP+FN}$, $Precision = \frac{TP}{TP+FP}$. F1 and F2 scores are derived based on *precision* and *recall* as follows: $F1 = \frac{2\times Precision\times Recall}{Precision+Recall}$, $F2 = \frac{5\times Precision\times Recall}{4\times Precision+Recall}$. The F1 score is the harmonic mean of the *precision* and *recall*, while the F2 score weighs *recall* higher than *precision*.

The F1 and F2 scores of the proposed PBRP and the two benchmark on the validation dataset are evaluated to select the appropriate hyper-parameter, *eps* (Fig. 2.4). Note that two *eps* are selected for each algorithm, one that optimizes F1 score and the other optimizes F2 score. After the hyper-parameters are selected, we apply the three algorithms to the testing dataset. The PBRP algorithm achieves significantly higher F1 and F2 scores than the benchmark algorithms on the testing dataset, mainly due to a substantial improvement in Recall (Table 2.3). Note that in Table 2.3 the top (bottom) three rows correspond to the hyperparameters optimized for F1 (F2) score.

The improvements over the benchmark can mostly be attributed to the ability to capture the spatial properties with the anomaly score $SAS$. Both OLAP and HOLAP are capable of capturing temporal anomalies. However, as seen in Fig. 2.5, these temporal anomalies, which are also captured by the $LAS$ indicator, does not become pronounced right away. This leads to delay in the detection of events and significantly decreases the recall. In contrast, the spatial anomaly indicators reach their peaks very quickly as soon as the event begins.

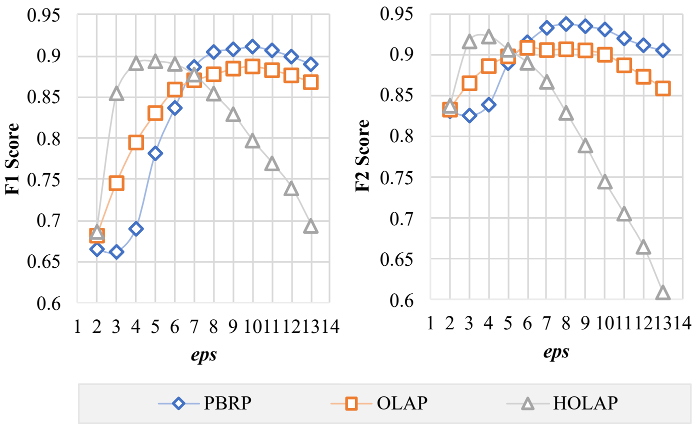


Figure 2.4: F scores of PBRP and benchmark algorithms on validation dataset.

Table 2.3: F scores of three algorithms on the testing dataset

| Statistics | OLAP | HOLAP | PBRP |
|---|---|---|---|
| **Precision** | **0.8889** | 0.8824 | 0.8881 |
| **Recall** | 0.8955 | 0.8955 | **0.9478** |
| **F1 Score** | 0.8922 | 0.8889 | **0.9170** |
| **Precision** | 0.8089 | **0.8571** | 0.8000 |
| **Recall** | 0.9478 | 0.9403 | **0.9851** |
| **F2 Score** | 0.9163 | 0.9224 | **0.9415** |

### Computational Efficiency

The computation complexity of BRP-based and SVD-based matrix decomposition approaches were discussed at the end of Section 2.2.3. This computational efficiency is crucial to apply the detection algorithms online. Faster event detection allows greater flexibility in the design of any submodules that follow.

To showcase that our method is faster than its competitors, we varied the number

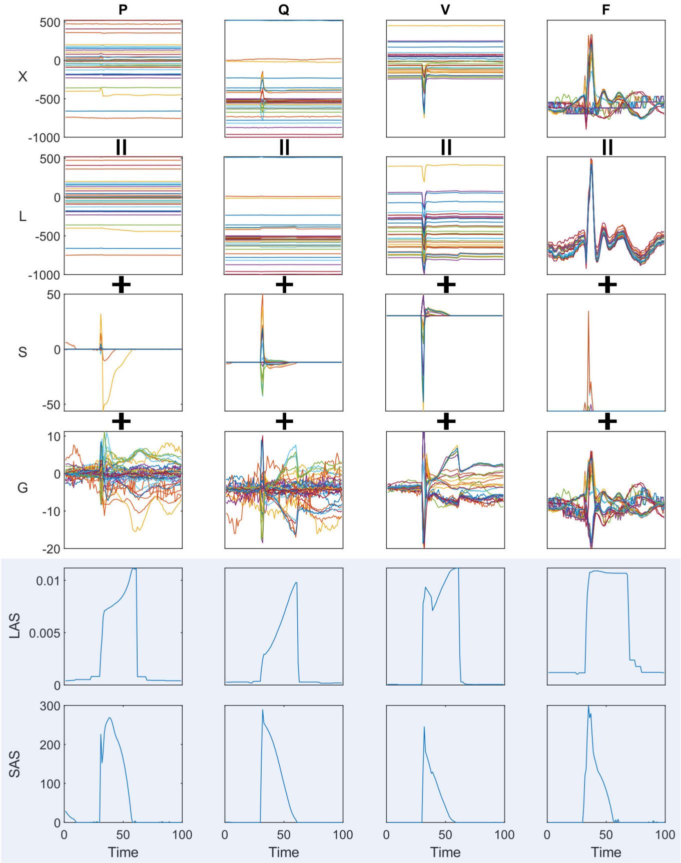


Figure 2.5: An example of decomposition of streaming PMU data matrix $X$.

of PMUs while fixing $w_1 = 30$ and $w_2 = 300$. Therefore, the size of the matrix to be decomposed is only proportional to the number of PMUs. All algorithm's computation times are reported in Table 3.4 after averaging over the detection of 100 randomly selected voltage events. The partial computation time excludes the time of the cluster analysis (Line 13 in Algorithm 3). The PBRP algorithm uses 50% less computation time compared to the benchmark algorithms. Furthermore, as the number of PMUs increases, the increase in the computation time of the PBRP algorithm is slower than that of the benchmark algorithms. Thus, the proposed algorithm shows better applicability to larger grids and larger numbers of sensors compared to benchmark algorithms.

Table 2.4: Average Computation Time of Algorithms Over Three-minute Time Period

| **Number of PMUs** | | 50 | 100 | 150 |
|---|---|---|---|---|
| **Computation** | HOLAP | 61.78/68.46 | 181.50/189.25 | 336.27/344.58 |
| **Time (s)** | OLAP | 7.53/15.01 | 9.58/17.33 | 16.99/24.79 |
| **(partial/total)** | PBRP | 2.18/8.46 | 3.13/9.40 | 4.29/10.53 |

### Identification of PMUs That Can Capture Voltage Events

As shown in Figs. 2.2 and 2.3, significant influence on voltage events is often observed in a limited number of PMUs. The sparsity structure in the $S$ matrix could help identify which PMUs are closely related to a particular voltage event. For example, we could first identify the non-zero elements in matrix $S$ at the start of the events. These row indices of these non-zero elements correspond to distinctive PMUs that firmly grasp an event. Fig. 2.6 illustrates the identification of the distinctive PMUs on the voltage event. The left column figures depict the original P, Q, V, and F data obtained from PMUs. The middle column is the heat map of sparse matrix $S$. We remove the PMUs in the original $X$ matrix corresponding to the rows with zero elements at timestamp 24 identified as the start of the event. The right column shows the filtered $X$, containing highly sensitive PMUs to the event. The algorithm effectively identifies all the PMUs with sizable dynamic behavior in P, Q, V, and F data streams.

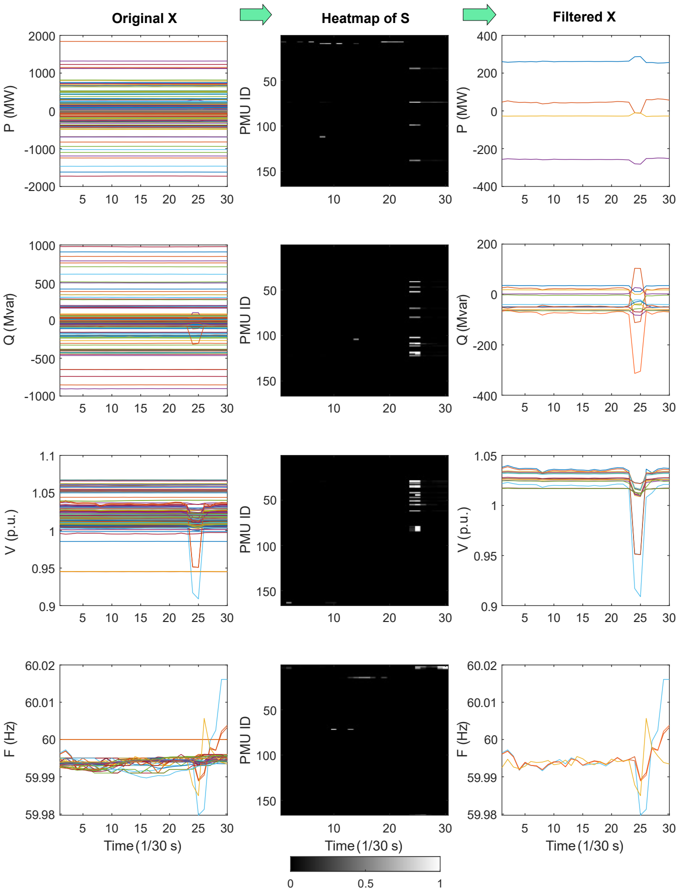


Figure 2.6: An example of identifying PMUs that are sensitive to voltage events.

## 2.4 Conlusion

This work reveals the distinctive sparsity structure of residual PMU data matrices during regional voltage events. This distinctive characteristic of voltage events motivates us to decompose the PMU data matrix into a low-rank matrix, a row-sparse event-pattern matrix, and a noise matrix. The key features extracted from the low-rank and row-sparse event-pattern matrices are leveraged in a clustering module to differentiate voltage events from normal operating conditions. A computationally efficient proximal bilateral random project-based algorithm, PBRP, is proposed to perform the matrix decomposition with structured sparsity-inducing norms. The feature extracted from the row-sparse event-pattern matrix significantly enhances the voltage event detection performance. A large-scale numerical study with real-world PMU data shows that our proposed *online* voltage event detection algorithm yields lower computation time, higher accuracy, and scalability than state-of-the-art benchmark. The proposed algorithm can also specify event area/zones by identifying the PMUs most sensitive to the detected event.

**Algorithm 1** Closed-Form BRPs with Power Scheme

**Input:** $X \in \mathbb{R}^{n \times w_1}$, rank $r$, power factor $q$

**Output:** $BRP(X) \triangleq L$

```
Initialize: $\tilde{X} = (XX^T)^q X$;
Initialize: $\forall i, j, A_{1,(i,j)} \sim \mathcal{N}(0,1), A_{2,(i,j)} \sim \mathcal{N}(0,1)$;
while true do
    $C = \tilde{X} A_1 = Q_{col} R_{col}$;
    $H = \tilde{X}^T A_2 = Q_{row} R_{row}$;
    if $rank(A_2^T C) < r$ then
        $r = rank(A_2^T C)$;
    else
        break;
    end if
    $A_1 = H$;
    $A_2 = C$;
end while
$L = Q_{col} [R_{col} (A_2^T C)^{-1} R_{row}^T]^{\frac{1}{2q+1}} Q_{row}^T$;
return $L$;
```

**Algorithm 2** Proximal BRP (PBRP)

```
Input: $X \in \mathbb{R}^{n \times w_1}$, rank $r$, power factor $q$, $\lambda$, $\epsilon$
Output: $L, S$
Initialization: $L = S = \mathbf{0}$
while $\frac{\|X-L-S\|_F^2}{\|X\|_F^2} \geq \epsilon$ do
    $L = BRP(X - S)$
    $S = Prox_{\lambda\|\cdot\|_{21}}(X - L)$;
end while
return $L, S$
```

**Algorithm 3** Voltage Event Detection Framework

**Input:** $X_{(l)} \in \mathbb{R}^{n \times T} (l \in \{P, Q, V, F\})$; Analysis window lengths: $w_1$, $w_2$; DBSCAN: $minpts, eps$; Adaptive Scheme: $\theta = 100, uprate = 1.1, downrate = 0.9$.

**Output:** Anomaly Alert

```
for $t = w_1 + 1 : T$ do
    for data type $l = \{P, Q, V, F\}$ do
        $AS_{(l)}[t] \xleftarrow{L,S} PBRP(X_{(l)}[:, t - w_1 : t], \lambda_{(l)})$;
        if $\max\limits_{i=1,\dots,n} |\frac{std_i}{\mu_i}| > \theta$ then
            $\lambda_{(l)} = \lambda_{(l)} \times uprate$;
        end if
        if $\|S\|_{21} == 0$ then
            $\lambda_{(l)} = \lambda_{(l)} \times downrate$;
        end if
    end for
    Construct the high-level feature batch:
        $B[t - w_2 + 1 : t] = \begin{pmatrix} AS_{(P)}[t - w_2 + 1 : t] \\ AS_{(Q)}[t - w_2 + 1 : t] \\ AS_{(V)}[t - w_2 + 1 : t] \\ AS_{(F)}[t - w_2 + 1 : t] \end{pmatrix}$;
    if $t \geq w_2$ then
        DBSCAN(normalize($B$), $eps$, $minpts$) $\rightarrow$ Alert;
    end if
end for
```

# Chapter 3

# Dynamic Parameter Estimation with Physics-based Neural Ordinary Differential Equations

High fidelity power system dynamic models are critical to both dynamic studies and reliable operation of the power system. Without accurate parameters, power engineers can not mimic historical disturbances and system events. But the nonlinearities and high dimensionality of the time-varying power system dynamic model make it challenging to estimate the parameters of generator dynamic models with high accuracy. This work proposes a physics-based neural ordinary differential equations (ODE) approach to estimate these parameters with PMU data.

## 3.1 Related Works

The topic of parameter estimation for power system and generator dynamic models has been studied extensively in the past [32, 53]. We briefly review a few representative research articles in this area. One of the first works formulates the dynamic parameter estimation problem as a nonlinear least squares problem using the sensitivities of the algebraic state of the system with respect to continuous dynamic state [20]. The parameters of the generator dynamic model are updated iteratively with a Gauss-Newton approach. The sensitivities are derived with respect to the initial operating condition, which may not be sufficiently accurate for bulk power system disturbances.

In another work [38], a black-box neural network is adopted with input neurons represented by transient stability indices and the output neurons represented by parameters of the generator dynamic model. The lack of power system domain knowledge in the black-box model led to low estimation accuracy and poor sample efficiency. Reference [11] used the Extended Kalman filtering (EKF) for this same task, but the linearization step resulted in similarly low parameter estimation accuracy. To address the shortcomings of the EKF approach, reference [16] applied the unscented Kalman filter (UKF). However, the accuracy of UKF significantly reduces if the signal-to-noise ratio is low. Reference [17] utilized a weighted least squares method by using sensitivities of measured modal frequencies and damping to the parameters. The drawback of this approach is that it relies on estimating dynamic modes of the power system, which may not be sufficiently accurate. A Bayesian approach is proposed in [34], which formulates the dynamic parameter estimation as a maximum a posteriori (MAP) problem. The discrete adjoint method is used to estimate the gradient of the loss function with respect to the dynamic parameters. A local optimization approach called the quasi-Newton method is applied to solve the MAP minimization problem. This approach may lead to local optima when initial dynamic parameters are drastically different from the ground truth, or the posterior distribution is non-Gaussian. To deal with non-Gaussian posterior distribution, a Markov chain Monte Carlo (MCMC) method aimed at finding the global optima for the MAP estimator is proposed [51].

## 3.2 Power System Dynamic Model

For ease of demonstration, a simplified dynamic model of a multi-machine interconnected power system is adopted. The proposed neural ODE-based parameter estimation technique can be applied to more complex dynamic models. The simplified model assumes that in the short observation period (a few seconds), the mechanical power input, $P_m$, is constant, and the classical model represents a generator with a constant voltage source behind a known transient reactance without damper winding. The terminal voltage $V_i$ and current phasors $I_i$ of all power plants are assumed to be measured by PMUs.

The differential equation of the classical generator model is represented by the swing equation shown in (3.1) [28].

$$\frac{M_0}{\omega_R}\ddot{\delta} = P_m - P_e, \tag{3.1}$$

where, $M_0$ is a inertia constant, $\mathrm{MW} \cdot \mathrm{s/MVA}$, $\omega_R$ denotes the rated rotor speed of a generator, $\delta$ is the angular position of a rotor relative to a synchronously rotating reference.

The algebraic equations coupling the classical generator model to the rest of the power system are represented by the following equations:

$$P_{e_i} = \Re\{E_i I_i^*\} = \Re\{Y_{\text{reduced}}^* E_i^2\}, \tag{3.2}$$

where $P_{e_i}$ denotes the active power output of the generator $i$, and $E_i$ is the generator $i$'s internal voltage phasor. $\Re\{\cdot\}$ extracts the real part of a complex number, and $*$ is the complex conjugate operation. $Y_{\text{reduced}} = Y_{gg} - Y_{gs} Y_{ss}^{-1} Y_{sg}$ is the reduced Y-bus matrix. $Y_{gg}$, $Y_{gs}$, $Y_{sg}$, and $Y_{ss}$ are sub-matrices of the admittance matrix of the entire system, where $g$ and $s$ correspond to the generator buses and other buses in the system. Constant impedance loads are assumed to be embedded into the Y-bus matrix.

The internal voltage of the classical generator dynamic model can be calculated with PMU measurements at the terminal as: $E_i = V_i + jx_{d_i}^{'} I_i$, where $x_{d_i}^{'}$ is the D-axis transient reactance of generator $i$. Thus, the active power output of unit $i$ can be calculated as:

$$P_{e_i} = \Re\{Y_{\text{reduced}}^* (V_i + jx_{d_i}^{'} I_i)^2\}. \tag{3.3}$$

## 3.3 Dynamic Parameter Estimation

The overall framework of the iterative neural ODE-based dynamic parameter estimation algorithm is shown in Fig. 3.1. We feed the initial states and start/end timestamps into a physics-informed neural network representing the ODEs to produce estimates of future states and PMU measurements. A loss function that quantifies the difference between the estimated and observed system states is summed over every time stamp, and its gradients (via adjoint) are back-propagated to dynamic parameters. Finally, the dynamic parameters

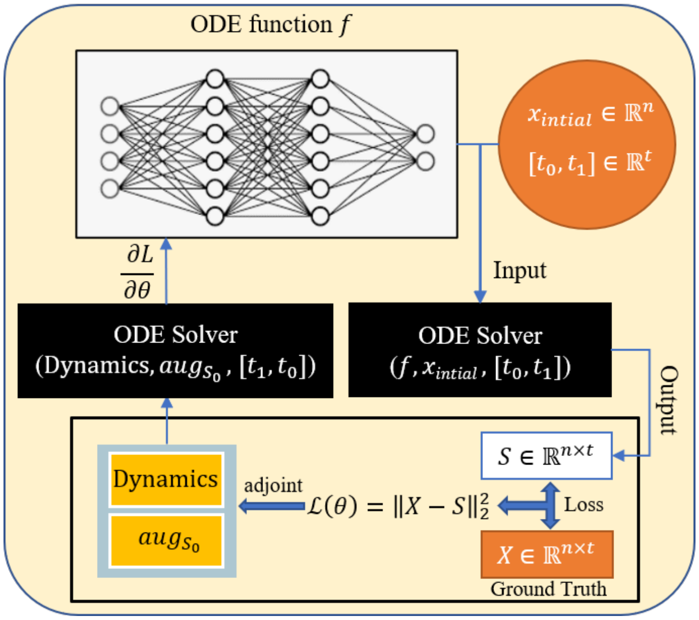

ODE function $f$
$x_{intial} \in \mathbb{R}^n$
$[t_0, t_1] \in \mathbb{R}^t$
$\frac{\partial L}{\partial \theta}$
Input
ODE Solver
(Dynamics, $aug_{S_0}$, $[t_1, t_0]$)
ODE Solver
($f$, $x_{intial}$, $[t_0, t_1]$)
Output
$S \in \mathbb{R}^{n \times t}$
Dynamics
$aug_{S_0}$
adjoint
$\mathcal{L}(\theta) = \|X - S\|_2^2$
Loss
$X \in \mathbb{R}^{n \times t}$
Ground Truth

Figure 3.1: Neural ODE-based dynamic parameter estimation framework.

are updated using gradient descent.

### 3.3.1 Overview of Neural ODEs

Neural ODEs are deep learning models which differ from standard machine learning methods in one key way - while standard machine learning methods map input variables to hidden variables for immediate use, Neural ODEs map input variables to the *derivative* of hidden variables which must first be integrated before their use [5]. Mathematically, We can write a standard neural network via the two equations $s = f_\alpha^{\text{in}}(x), y = f_\alpha^{\text{out}}(s)$. In contrast a neural ODE can be written as the two equations $\frac{\partial s}{\partial t} = f_\theta^{\text{in}}(x), y = f_\theta^{\text{out}}(s)$ where now the function $f_\theta^{in}$ returns time derivatives of the state variables instead of the state variables themselves. Here, $\alpha$ and $\theta$ represent the parameters of the two corresponding models.

The advantage of Neural ODEs over more typical models is that the hidden variable, $s$, is now actually a *smooth family* of hidden variables parameterized by a new variable, $t$. This variable is typically used to represent a continuous "depth" of the network, but can also represent a time variable when modeling dynamical systems, in which we have the single hidden variable per time instance. Our work will adopt the latter interpretation. Since our goal is to model an existing function of time, this hidden variable can be used directly as our output (i.e., $f_\theta^{\text{out}}$ is the identity function).

The disadvantage of using Neural ODEs is that hidden variables need to be integrated. In practice, this means they must be sent through an ODE solver. Furthermore, we need to take gradients of the solver regarding the parameters so that the ODE must also be back-propagated through.

The existing ODE solvers can be divided into two groups. One group consists of adaptive-step ODE solvers, such as the Dormand–Prince method [3]; another group consists of fixed-step ODE solvers, such as the Euler method [21] and the Runge-Kutta method [59]. The latter is faster and more widely used in the industry than the former. This paper adopts the explicit fourth-order Runge-Kutta method with the 3/8 rule.

### 3.3.2 Physics-Informed Neural Network Design

Neural ODEs typically use the expressivity of large neural networks to model the parameters of an ODE as a parametric black box. However, in our case, the ODE we are modeling possesses explicit mathematical expressions - namely, the swing equation. Thus, our neural network is explicitly modeled to the form of this known physical equation.

Let us assume that $n$ generators and that $|E_i|$ and $Y_{bus}$ are given. Denote the element in the $Y^*_{reduced}$ matrix located at the $i$-th row and $j$-th column as $|Y_{ij}|e^{\angle\phi_{ij}}$. Then, the dynamic equations of the $i$-th generator can be formulated as follows:

$$\begin{cases} \dot{\delta}_i = g_i(\omega_i[\mathrm{pu}]) \\ \dot{\omega}_i[\mathrm{pu}] = h_i(\delta_{i=1,\dots,n}) \\ g_i = \omega_R(\omega_i[\mathrm{pu}] - 1) \\ h_i = \frac{P_{m_i} - \sum_{j=1}^{n} |E_i||E_j||Y_{ij}| cos(\delta_i - \delta_j - \phi_{ij})}{M_{0_i}}, \end{cases} \tag{3.4}$$

where $\delta_i$ and $\omega_i$ are the rotor angle and the rotor angle speed of generator $i$ respectively. The unknown parameters are $P_{m_i}$ and $M_{0_i}$. A neural network structure is strictly derived following (3.4). An example of this design for two generators is visualized in Fig. 3.2. Only the parameters in yellow boxes will be updated, others are fixed. In the case of multiple generators, we only need to extend this figure horizontally. It is too difficult for Neural ODE to solve all the unknown noisy parameters in one layer. Therefore, we use two nonlinear layers $\exp(\cdot)$ and $\ln(\cdot)$ to separate them, improving the nonlinear fitting ability of neural network. Since the reciprocal of $M_{0_i}$ may be very small, it is amplified using a factor of $k$, and then an extra activation function is added before the output layer, which multiplies its input by a factor of $\frac{1}{k}$. The weights and bias are fixed in the first two layers. Thus, only the unknown parameters are updated. The neural network corresponding to $g_i$ is relatively simple since every weight and bias are fixed.

### 3.3.3 Loss Function and Gradient Descent

The following mean square error loss function is used to train the physics-informed neural network:

$$L(s) = \sum_{t=t_0}^{t_1} \|x(t) - s(t)\|_2^2, \tag{3.5}$$

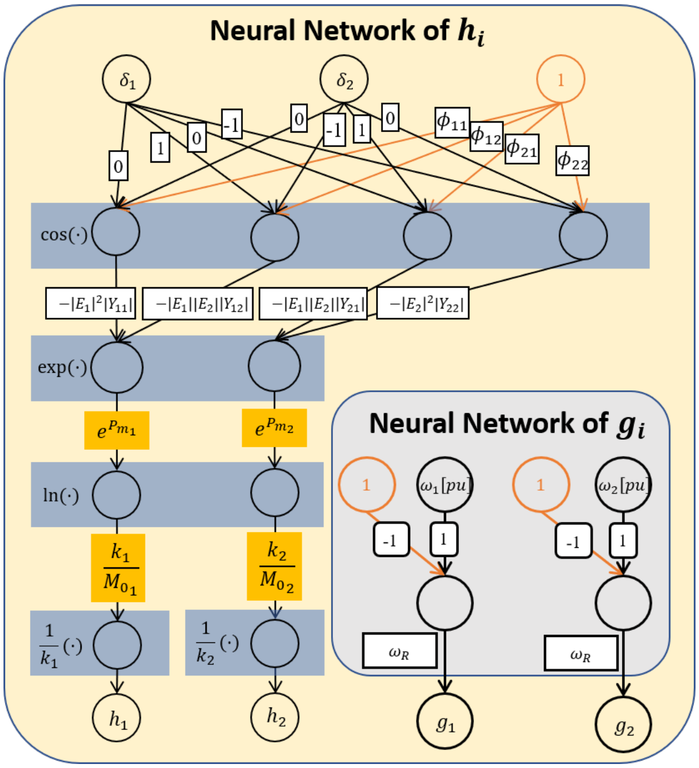


Figure 3.2: The diagram of the physics-informed neural network design.

where $x(t)$ represents the vector time-series for calculated state variables (i.e., $\delta_i$ and $\omega_i$) from the PMU data. $s(t)$ denotes the vector time-series of state variables estimated from the physics-based neural networks.

To calculate $\frac{\partial L}{\partial \theta}$, the gradients of $L$ with respect to $s(t)$ need to be computed first. The **adjoint method** is chosen to derive the gradient of $L$ with respect to the estimated state variables [36]. Specifically, a new time series, $a(t) = \frac{\partial L}{\partial s(t)}$, which we call the *adjoint* of $s(t)$, is created. It satisfies the following ODE [5]:

$$\frac{da(t)}{dt} = -a(t)^T \frac{\partial f(s(t), t, \theta)}{\partial s}, \tag{3.6}$$

where $f$ denotes two physics-based neural networks, $g$ and $h$, and $\theta$ denotes the parameters of the physics-based neural network $M_{0_i}$ and $P_{m_i}$.

The gradient of the loss function with respect to the neural network parameters, $\theta$, is a reverse integral over $[t_0, t_1]$ [5]:

$$\frac{dL}{d\theta} = -\int_{t_1}^{t_0} a(t)^T \frac{\partial f(s(t), t, \theta)}{\partial \theta} dt \tag{3.7}$$

**Algorithm 4** Calculate gradient w.r.t. $\theta$ by adjoint method.

```
Input: parameters in the neural network $\theta$, time span $[t_0, t_1]$, final state $s(t_1)$, adjoint $\frac{\partial L}{\partial s(t_1)}$
augS$_0 = [s(t_1), \frac{\partial L}{\partial s(t_1)}, 0]$;
Calculate dynamics of augmented state:
    Dynamics $= [f(s(t), t, \theta), -a(t)^T \frac{\partial f}{\partial s}, -a(t)^T \frac{\partial f}{\partial \theta}]$;
ODESolver(Dynamics, augS$_0, t_0, t_1, \theta) \Rightarrow$
    $[s(t_0), \frac{\partial L}{\partial s(t_0)}, \frac{\partial L}{\partial \theta}]$;
return $\frac{\partial L}{\partial \theta}$;
```

The gradient calculation steps are summarized in Algorithm 4. Finally, we update the dynamic parameters $\theta$ with the limited-memory Broyden-Fletcher-Goldfarb-Shanno method, which updates each weight with its own squared gradient.

## 3.4 Numerical Study

### 3.4.1 Simulation Set up

We generate pseudo PMU measurements from dynamic simulation data from a 3-machine 9-bus system (see Fig. 3.3). The power system parameters and initial operating

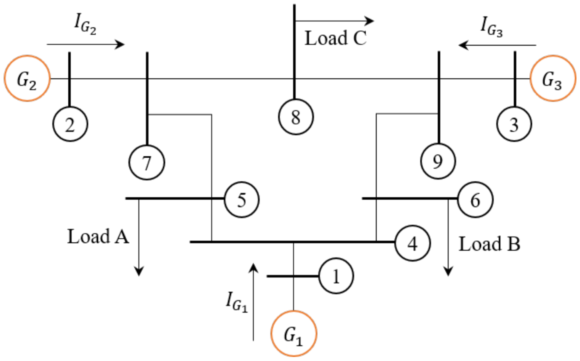


Figure 3.3: WECC 3-machine-9-bus system [47].

conditions are given in Tables 3.1 and 3.2. A single transmission line is disconnected at $5s$ as a disturbance, and the simulation time is set at $10s$. The initial values of six state variables are given in [47] as $\delta_{i=1,2,3} = [0.0396, 0.3447, 0.2304]$, $\omega_{i=1,2,3}[\text{pu}] = 1$. $V_i$ and $I_i$ are calculated from $\delta_i$ and $\omega_i$, and are considered to be our input data. Then, random Gaussian noise signal, $G \sim N(0, 0.001)$ is added to the magnitude and angle of $V_i$ and $I_i$, which is consistent with the estimated noise shown in a standard [39]. Finally, $\delta_i = \angle E_i$ can be indirectly derived from $E_i = V_i + jx'_{d_i} I_i$. Then, $\omega_i$ is calculated by $\omega_i[\text{pu}] = \frac{\delta_i[t+1]-\delta_i[t-1]}{4\pi\Delta T} + \omega_0$ [54] where $\omega_0 = 1$. Ground truth values of six parameters are: $P_{m_i} = \{0.7141, 1.6300, 0.8508\}$, $M_{0_i} = \{9.5515, 3.3333, 2.3516\}$.

Two disturbance scenarios are studied. The first one is a single line tripping between nodes 5 and 7. The dynamic response for this disturbance is the same as the one illustrated in [47]. The second one disconnects the line between nodes 8 and 9. Note that the rotor angles of the generators continuously grow following the line tripping events because damper components are not modeled. The dynamic simulation time step is set as 1/120 s

Table 3.1: Power System Parameters [47]

| Branch Impedance (pu) | | Capacitance (pu) | Load $P_L$ & $Q_L$(pu) |
|---|---|---|---|
| $z_{14} = 0.0576i$ | $z_{45} = 0.0100 + 0.0850i$ | $\frac{B_{38}}{2} = 0.1045i$ | $P_{L5} = 1.25$ |
| $x'_{d1} = 0.0608i$ | $z_{46} = 0.0170 + 0.0920i$ | $\frac{B_{45}}{2} = 0.0880i$ | $Q_{L5} = 0.50$ |
| $z_{27} = 0.0625i$ | $z_{57} = 0.0320 + 0.1610i$ | $\frac{B_{46}}{2} = 0.0790i$ | $P_{L6} = 0.90$ |
| $x'_{d2} = 0.1198i$ | $z_{69} = 0.0390 + 0.1700i$ | $\frac{B57}{2} = 0.1530i$ | $Q_{L6} = 0.30$ |
| $z_{39} = 0.0586i$ | $z_{78} = 0.0085 + 0.0720i$ | $\frac{B_{69}}{2} = 0.1790i$ | $P_{L8} = 1.00$ |
| $x'_{d3} = 0.1813i$ | $z_{89} = 0.0119 + 0.1008i$ | $\frac{B_{78}}{2} = 0.0745i$ | $Q_{L8} = 0.35$ |
| | | $\frac{B_{89}}{2} = 0.1045i$ | |

Note: Per unit values are calculated with 100 MVA base (and nominal voltage).

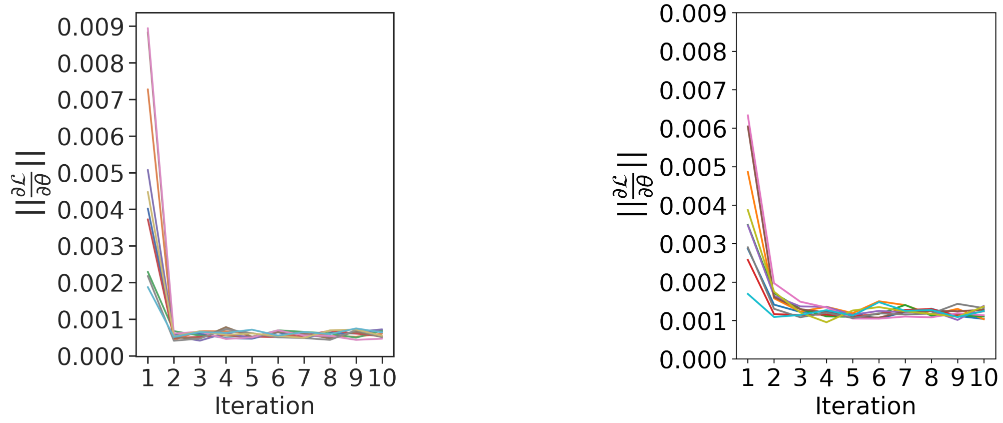
Table 3.2: Initial Condition of Three Generators [47]

| | $G_1$ | $G_2$ | $G_3$ |
|---|---|---|---|
| Active power (pu) | 0.7160 | 1.6300 | 0.8500 |
| Reactive power (pu) | 0.2700 | 0.0670 | -0.1090 |
| Terminal voltage (pu) | 1.0400 | 1.0250 | 1.0250 |

for both events. The PMU measurements of the generators are then downsampled to 30 Hz.

### 3.4.2 Dynamic Parameter Estimation Results

The neural ODE fitting framework is tested with random $[-10\%, 10\%]$ errors on the initial value of six parameters. The dataset for each disturbance contains 5 s of data, i.e., 150 data points. We set $k$ for the generators as $k_i = \{20, 7, 5\}$ to make sure the ratio $\frac{k_i}{M_{0_i}}$ is almost the same. In the training process, we set the batch size and batch time as 20 and 10 steps, respectively. This means that we randomly choose 20 data points as the initial state



(a) Disturbance 1, learning rate= 0.1

(b) Disturbance 2, learning rate= 0.9

Figure 3.4: The norm of six parameters' gradient of the neural ODE estimation framework.

in the range of $[t_0,\, t_1 - \frac{1}{3}]$. Then, starting from each initial state, we step forward-in-time through the ODE solver for 10 contiguous timestamps and back-propagate against each of these segments separately.

### Parameter Estimation for Two Disturbances

We display the norm of gradient $\|\frac{\partial \mathcal{L}}{\partial \theta}\|_2$ against the training iterations in Fig. 3.4. For each disturbance, we run the experiment 10 times with different random seed. Each colored line represents an experiment. As shown in the figure, our proposed algorithm converges within only two iterations for both disturbances.

We further quantify the accuracy and computation efficiency of our proposed neural ODE-based dynamic parameter estimation algorithm using the first disturbance. The estimation errors are calculated for the six unknown parameters with three different data lengths (1s, 3s, 5s). The relative estimation error (REE) $= \frac{|\hat{Para} - Para|}{Para} \times 100\%$ is used as the accuracy metric. We treat the algorithm in [34] as the baseline for comparison. The training stops when the change of $\|\frac{\partial \mathcal{L}}{\partial \theta}\|_2$ is less than a threshold value of 0.001. A large learning rate could lead to divergence, while a small one requires long computation time. For a fair comparison, the best learning rate is selected among 0.5, 0.05, and 0.005 to train our proposed and baseline model. The same random seeds are used in both methods.

### Parameter Estimation Accuracy

The REEs are calculated and shown in Table 3.3. The first and second number in each cell represent the REE of the baseline and the proposed algorithm respectively. When the PMU data length is 3s, our proposed algorithm achieves the lowest REE for the six unkonwn parameters. We can also observe that the estimation of mechanical power input is more accurate than that of generator inertia constant. Overall, the physics-based neural ODE algorithm outperforms the baseline algorithm in terms of estimation accuracy for most of the unknown parameters.

### Computation Time & Scalability

The computation time of the proposed and baseline algorithms for estimating dynamic parameters from different length of PMU data are reported in Table 3.4. Note that the learning rate of a scenario in an algorithm is selected such that divergence behavior is avoided. As shown in the Table 3.4, our proposed physics-based neural ODE algorithm has much shorter computation time than the baseline algorithm. When the data length is 3s, the running time of our model is just 4.82 seconds, which is nearly 8 times faster than the

Table 3.3: Relative Estimation Error (%) of Baseline and NeuralODE-based method

| | | Data Length | | | Initial |
|---|---|---|---|---|---|
| | | 1s | 3s | 5s | REE |
| Parameters | $P_{m_1}$ | 2.36 / **1.50** | 7.70 / **1.49** | 7.47 / **1.26** | 4.21 |
| | $P_{m_2}$ | 0.76 / **0.01** | 5.05 / **0.12** | 5.01 / **0.18** | 5.77 |
| | $P_{m_3}$ | 1.05 / **0.41** | 6.27 / **0.32** | 6.52 / **0.46** | 4.76 |
| | $M_{0_1}$ | 5.80 / **2.86** | 5.52 / **2.06** | 6.31 / **3.09** | 4.80 |
| | $M_{0_2}$ | 4.36 / **3.19** | 5.92 / **2.18** | 5.88 / **3.67** | 6.10 |
| | $M_{0_3}$ | **4.58** / 8.37 | 5.50 / **4.82** | **5.49** /10.51 | 5.33 |
| Average | | 3.15 / **2.72** | 5.99 / **1.83** | 6.11 / **3.20** | 5.16 |

Note: Baseline / Physics-based Neural ODE Algorithm.

Table 3.4: Running Time (s) of Baseline and Neural ODE-based method.

| Data Length | Running Time (second) | | Learning Rate |
|---|---|---|---|
| | Baseline | Neural ODE-based | |
| 1s | 8.38 | 3.78 | 0.5 / 0.5 |
| 3s | 38.55 | 4.82 | 0.05 / 0.5 |
| 5s | 100.25 | 4.67 | 0.05 / 0.5 |

Note: Baseline / Physics-based Neural ODE Algorithm.

baseline model. The mini-batch scheme of neural network training is leveraged in the proposed algorithm, which greatly shortens the model running time and makes the algorithm more scalable in handling longer training dataset.

## 3.5 Conclusion

This work develops an online physics-based neural ODE algorithm to estimate the parameters of the generator dynamic model. By synergistically combining the swing equation and neural ODE model, our proposed algorithm outperforms the state-of-the-art baseline algorithm in terms of estimation accuracy and computation time. Numerical studies on a 3-machine 9-bus power system show that our proposed model is capable of accurately estimating the dynamic parameters using just 3 seconds of noisy PMU data with 30 Hz sampling frequency. Furthermore, the entire dynamic parameter estimation procedure takes less than 5 seconds of computation time.

# Chapter 4

# Conclusions

In this thesis, we demonstrated the potential of physics-informed machine learning models in power transmission systems. In terms of data-driven monitoring, we designed an online anomaly detection algorithm based on a fast low-rank and structured sparse matrix decomposition approach, PBRP, to realize real-time monitoring of the streaming data systems. In terms of electric generator parameter estimation, we proposed a Neural ODE-based parameter estimation model. In general, the main work of this thesis is as follows:

We revealed the distinctive sparsity structure of residual PMU data matrices during regional voltage events. Crucial features extracted from low-rank and row-sparse event pattern matrices were utilized in the clustering module to distinguish voltage events from normal operating conditions. A computationally efficient near-end bilateral random item based algorithm PBRP was proposed to perform matrix factorization with structured sparsity-induced norm. The PBRP approach greatly accelerated the solution of a general low-rank and sparse matrix decomposition problem where the residual matrix has a row-sparse structure. Meanwhile, thanks to structural sparseness, PBRP also helped to pinpoint the location of events once they are detected (Chapter 2).

We extended the prior work by converting a forward solver of the ODEs representing power system dynamics into physics-informed neural networks. The whole idea was inspired by the Neural ODE framework that gets rid of the memory-consuming 'backpropagation' technique with the help of the ODE solver. With the help of mini-batch scheme, our proposed algorithm outperformed the state-of-the-art baseline algorithm in estimation accuracy and computation time. (Chapter 3).

# Bibliography

[1] Phil Ashton. *Exploiting Phasor Measurement Units for Enhanced Transmission Network Operation and Control*. PhD thesis, 05 2014.

[2] Francis Bach, Rodolph Jenatton, Julien Mairal, and Guillaume Obozinsk. *Optimization with sparsity-inducing penalties*. Now Foundations and Trends, first edition, 2012.

[3] M Calvo, JI Montijano, and L Randez. A fifth-order interpolant for the Dormand and Prince Runge-Kutta method. *J. Comput. Appl. Math.*, 29(1):91–100, 1990.

[4] Chen Chen, Jianhui Wang, and Hao Zhu. Effects of phasor measurement uncertainty on power line outage detection. 8(6):1127–1139, 2014.

[5] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *arXiv preprint arXiv:1806.07366*, 2018.

[6] Yuanbin Cheng, Nanpeng Yu, Brandon Foggo, and Koji Yamashita. Online power system event detection via bidirectional generative adversarial networks. *IEEE Transactions on Power Systems*, pages 1–1, 2022.

[7] CIGRE C4.34. Application of phasor measurement units for monitoring power system dynamic performance. Technical report, CIGRE, 2017.

[8] Mingjian Cui, Jianhui Wang, Jin Tan, Anthony R Florita, and Yingchen Zhang. A novel event detection method using PMU data with high precision. *IEEE Transactions on Power Systems*, 34(1):454–466, 2018.

[9] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of the johnson-lindenstrauss lemma. *Random Structure & Algorithm*, 22(1):60–65, Jan. 2003.

[10] Elisabeth Drayer and Tirza Routtenberg. Detection of false data injection attacks in smart grids based on graph signal processing. *IEEE Systems Journal*, 14(2):1886–1896, 2019.

[11] Lingling Fan and Yasser Wehbe. Extended Kalman filtering based real-time dynamic state and parameter estimation using PMU data. *Electr. Power Syst. Res.*, 103:168–177, 2013.

[12] Brandon Foggo and Nanpeng Yu. Online PMU missing value replacement via event-participation decomposition. *IEEE Transactions on Power Systems*, 37(1):488–496, 2022.

[13] Pengzhi Gao, Meng Wang, Joe H Chow, Scott G Ghiocel, Bruce Fardanesh, George Stefopoulos, and Michael P Razanousky. Identification of successive "unobservable" cyber data attacks in power systems through matrix decomposition. *IEEE Transactions on Signal Processing*, 64(21):5557–5570, 2016.

[14] Pengzhi Gao, Meng Wang, Scott G Ghiocel, Joe H Chow, Bruce Fardanesh, and George Stefopoulos. Missing data recovery by exploiting low-dimensionality in power system synchrophasor measurements. 31(2):1006–1013, 2015.

[15] Yinyin Ge, Alexander J Flueck, Dae-Kyeong Kim, Jong-Bo Ahn, Jae-Duck Lee, and Dae-Yun Kwon. Power system real-time event detection and associated data archival reduction based on synchrophasors. 6(4):2088–2097, 2015.

[16] MA González-Cagigal, JA Rosendo-Macías, and A Gómez-Expósito. Parameter estimation of fully regulated synchronous generators using unscented Kalman filters. *Electr. Power Syst. Res.*, 168:210–217, 2019.

[17] Song Guo, Sean Norris, and Janusz Bialek. Adaptive parameter estimation of power system dynamic model using modal information. 29(6):2854–2861, 2014.

[18] Jinping Hao, Robert J Piechocki, Dritan Kaleshi, Woon Hau Chin, and Zhong Fan. Sparse malicious false data injection attacks and defense mechanisms in smart grids. *IEEE Transactions on Industrial Informatics*, 11(5):1–12, 2015.

[19] Yingshuai Hao, Meng Wang, Joe H Chow, Evangelos Farantatos, and Mahendra Patel. Modelless data quality improvement of streaming synchrophasor measurements by exploiting the low-rank hankel structure. *IEEE Transactions on Power Systems*, 33(6):6966–6977, 2018.

[20] Ian A Hiskens and Alexander Koeman. Power system parameter estimation. *J. Electr. Electron. Eng. Australia*, 19:1–8, 1999.

[21] Jean Jacod and Philip Protter. Asymptotic error distributions for the Euler method for stochastic differential equations. *Ann. Probab.*, 26(1):267–307, 1998.

[22] Mukhtaj Khan, Phillip M Ashton, Maozhen Li, Gareth A Taylor, Ioana Pisica, and Junyong Liu. Parallel detrended fluctuation analysis for fast event detection on massive pmu data. 6(1):360–368, 2014.

[23] Do-In Kim, Tae Yoon Chun, Sung-Hwa Yoon, Gyul Lee, and Yong-June Shin. Wavelet-based event detection method using pmu data. 8(3):1154–1162, 2015.

[24] Kenta Kirihara, Jun Yamazaki, Panitarn Chongfuangprinya, Stavros Konstantinopoulos, Christoph Lackner, Joe H. Chow, Slava Maslennikov, and Yilu Liu. Speeding up the

dissipating energy flow based oscillation source detection. In *2019 International Conference on Smart Grid Synchronized Measurements and Analytics*, pages 1–8, College Station, TX, USA, May 2019.

[25] Luc Knockaert, Bernard De Backer, and Daniel De Zutter. Svd compression, unitary transforms, and computational complexity. 47(10):2724–2729, 1999.

[26] Xianghao Kong, Brandon Foggo, Koji Yamashita, and Nanpeng Yu. Online voltage event detection using synchrophasor data with structured sparsity-inducing norms. *IEEE Transactions on Power Systems*, pages 1–1, 2021.

[27] Xianghao Kong, Koji Yamashita, Brandon Foggo, and Nanpeng Yu. Dynamic parameter estimation with physics-based neural ordinary differential equations. In *2022 IEEE Power & Energy Society General Meeting (PESGM)*, pages 01–05. IEEE, 2022.

[28] Prabha Kundur. *Power Systems Stability and Control.* McGraw-Hill Education, 1st edition, Jan. 1994.

[29] Wenting Li, Meng Wang, and Joe H Chow. Fast event identification through subspace characterization of PMU data in power systems. In *2017 IEEE Power & Energy Society General Meeting*, pages 1–5, Chicago, IL, USA, Jul. 2017.

[30] Lanchao Liu, Mohammad Esmalifalak, Qifeng Ding, Valentine A Emesih, and Zhu Han. Detecting false data injection attacks on power grid by sparse optimization. *IEEE Transactions on Smart Grid*, 5(2):612–621, 2014.

[31] Shengyuan Liu, Yuxuan Zhao, Zhenzhi Lin, Yilu Liu, Yi Ding, Li Yang, and Shimin Yi. Data-driven event detection of power systems based on unequal-interval reduction of pmu data and local outlier factor. 11(2):1630–1643, 2019.

[32] A. Monticelli. Electric power system state estimation. 88(2):262–282, Feb. 2000.

[33] Sanjay Singh Negi, Nand Kishor, Kjetil Uhlen, and Richa Negi. Event detection and its signal characterization in pmu data stream. *IEEE Trans. Ind. Informat.*, 13(6):3108–3118, 2017.

[34] Noemi Petra, Cosmin G Petra, Zheng Zhang, Emil M Constantinescu, and Mihai Anitescu. A Bayesian approach for parameter estimation with uncertainty for dynamic power systems. 32(4):2735–2743, 2016.

[35] Arun G Phadke and Tianshu Bi. Phasor measurement units, WAMS, and their applications in protection and control of power systems. *Journal of Modern Power Systems and Clean Energy*, 6(4):619–629, 2018.

[36] Lev Semenovich Pontryagin. *Mathematical theory of optimal processes.* CRC press, 1987.

[37] Mark Rafferty, Xueqin Liu, David M Laverty, and Sean McLoone. Real-time multiple event detection and classification using moving window pca. 7(5):2537–2548, 2016.

[38] AHMA Rahim and AJ Al-Ramadhan. Dynamic equivalent of external power system and its parameter estimation through artificial neural networks. *Int. J. Electr. Power Energy Syst.*, 24(2):113–120, 2002.

[39] Ankur Singh Rana, Nisha Parveen, Shaziya Rasheed, and Mini S Thomas. Exploring IEEE standard for synchrophasor C37.118 with practical implementation. In *2015 Annual IEEE India Conf. (INDICON)*, pages 1–6. IEEE, 2015.

[40] Tirza Routtenberg and Yonina C Eldar. Centralized identification of imbalances in power networks with synchrophasor data. *IEEE Transactions on Power Systems*, 33(2):1981–1992, 2017.

[41] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

[42] Jie Shi, Brandon Foggo, Xianghao Kong, Yuanbin Cheng, Nanpeng Yu, and Koji Yamashita. Online event detection in synchrophasor data with graph signal processing. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–7, Tempe, AZ, USA, Nov. 2020.

[43] Jie Shi, Brandon Foggo, and Nanpeng Yu. Power system event identification based on deep neural network with information loading. *IEEE Transactions on Power Systems*, 36(6):5622–5632, 2021.

[44] Jie Shi, Koji Yamashita, and Nanpeng Yu. Power system event identification with transfer learning using large-scale real-world synchrophasor data in the united states. In *2022 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE, 2022.

[45] Jie Shi, Nanpeng Yu, Eamonn Keogh, Heng Kevin Chen, and Koji Yamashita. Discovering and labeling power system events in synchrophasor data with matrix profile. In *2019 IEEE Sustainable Power and Energy Conference (iSPEC)*, pages 1827–1832. IEEE, 2019.

[46] Chunming Tu, Xi He, Zhikang Shuai, and Fei Jiang. Big data issues in smart grid–a review. *Renewable and Sustainable Energy Reviews*, 79:1099–1107, 2017.

[47] Vijay Vittal, James D. McCalley, Paul M. Anderson, and A. A. Fouad. *Power System Control and Stability.* Wiley-IEEE Press, 3 edition, 2019.

[48] Weikang Wang, He Yin, Chang Chen, Abigail Till, Wenxuan Yao, Xianda Deng, and Yilu Liu. Frequency disturbance event detection based on synchrophasors and deep learning. 11(4):3593–3605, Jul. 2020.

[49] Zhengyu Wang, Lingling Fan, and Zhixin Miao. Data-driven dynamic model identification for synchronous generators. In *Proc. Annual North Am. Power Symp.*, Oct. 2019.

[50] Le Xie, Yang Chen, and PR Kumar. Dimensionality reduction of synchrophasor data for early event detection: Linearized analysis. 29(6):2784–2794, 2014.

[51] Yijun Xu, Can Huang, Xiao Chen, Lamine Mili, Charles H Tong, Mert Korkali, and Liang Min. Response-surface-based Bayesian inference for power system dynamic parameter estimation. 10(6):5899–5909, 2019.

[52] Ravi Yadav, Ashok Kumar Pradhan, and Innocent Kamwa. Real-time multiple event detection and classification in power system using signal energy transformations. *IEEE Trans. Ind. Informat.*, 15(3):1521–1531, 2018.

[53] P. Zarco and A.G. Exposito. Power system parameter estimation: a survey. 15(1):216–222, Feb. 2000.

[54] Jiecheng Zhao, Lingwei Zhan, He Yin, Fuhua Li, Wenxuan Yao, and Yilu Liu. Recent development of frequency estimation methods for future smart grid. *IEEE Open Access Journal of Power and Energy*, 7:354–365, 2020.

[55] Mengze Zhou, Yuhui Wang, Anurag K Srivastava, Yinghui Wu, and P Banerjee. Ensemble-based algorithm for synchrophasor data anomaly detection. 10(3):2979–2988, 2018.

[56] Ning Zhou and Jeff Dagle. Initial results in using a self-coherence method for detecting sustained oscillations. 30(1):522–530, 2014.

[57] Tianyi Zhou and Dacheng Tao. Bilateral random projections. In *2012 IEEE International Symposium on Information Theory Proceedings*, pages 1286–1290, Cambridge, MA, USA, Jul. 2012.

[58] Hao Zhu and Georgios B Giannakis. Sparse overcomplete representations for efficient identification of power line outages. *IEEE Transactions on Power Systems*, 27(4):2215–2224, 2012.

[59] David W Zingg and Todd T Chisholm. Runge–Kutta methods for linear ordinary differential equations. *Appl. Numer. Math.*, 31(2):227–238, 1999.