Online Robust Subspace Clustering with Application to Power Grid Monitoring

Young-hwan Lee[1,3] Seung-Jun Kim[1] Kwang Y. Lee[2] Taesik Nam[3]

E-mails: {lee43,sjkim}@umbc.edu, Kwang_Y_Lee@baylor.edu, tsnam@kitech.re.kr

[1]Dept. of Computer Science and Electrical Engineering
University of Maryland, Baltimore County, Baltimore, MD 21250, USA

[2]Dept. of Electrical and Computer Engineering
Baylor University, Waco, TX 76798, USA

[3]Green Energy & Nano Technology R&D Group
Korea Institute of Industrial Technology, Gwangju 61012, South Korea

***Abstract*—In this work, a robust subspace clustering algorithm is developed to exploit the inherent union-of-subspaces structure in the data for reconstructing missing measurements and detecting anomalies. Our focus is on processing an incessant stream of large-scale data such as synchronized phasor measurements in the power grid, which is challenging due to computational complexity, memory requirement, and missing and corrupt observations. In order to mitigate these issues, a low-rank representation (LRR) model-based subspace clustering problem is formulated that can handle missing measurements and sparse outliers in the data. Then, an efficient online algorithm is derived based on stochastic approximation. The convergence property of the algorithm is established. Strategies to maintain a representative yet compact dictionary for capturing the subspace structure are also proposed. The developed method is tested on both simulated and real phasor measurement unit (PMU) data to verify the effectiveness and is shown to significantly outperform existing algorithms based on simple low-rank structure of data.**

Keywords— Anomaly detection, incomplete measurement, low-rank representation, online algorithm, subspace clustering, phasor measurement unit, power system monitoring, synchrophasor.

I. Introduction

Grid status monitoring is an important prerequisite for reliable and efficient operation of the power system. Failure to identify anomalies in the grid states in an accurate and timely manner may result in inefficient use of resources, equipment damages, system instabilities, and even cascading blackouts. It is also critical that cyber-attacks to the power system and its data be detected, and their effect mitigated promptly.

The benefit of employing synchrophasor measurements for power system monitoring has been widely recognized in recent years [1], [2]. The phasor measurement unit (PMU) is a device that can measure voltage phasors at buses, typically at a rate of 30 samples per second or higher. Based on the global positioning system (GPS), PMU measurements can achieve precise synchronization across a wide region. Compared to the conventional supervisory control and data acquisition (SCADA) protocol, which provides measurements at a frequency of a few samples per minute, PMU data can reveal dynamic changes in the grid states, significantly improving the monitoring capability.

As the high-speed large-scale spatio-temporal measurement of the grid is enabled, data-driven approaches for power system monitoring became important. Departing from the more conventional model-driven paradigm, which requires detailed and accurate system models and parameters, the data-driven methods utilize the ample measurements to learn the salient structures in the data and perform inferences. Voltage stability was predicted using singular value decomposition (SVD), spline extrapolation, and neural networks, trained on measured and simulated data [3]–[6]. Line outages were identified using machine learning techniques [7], [8]. A real-time event identification method was developed based on unique subspace signatures present in the dynamics of PMU data [9]. Co-occurring multiple events were recognized from frequency measurements through sparse coding techniques [10]. The onset of events was detected based on compressed PMU data using dynamic programming optimization [11]. Periodic forced oscillations caused by rogue inputs were detected from PMU measurements in [12].

There are some challenges associated with analyzing the large-scale data generated by PMUs. The processing of up to tera-bits of daily measurements for large-scale systems require significant dimensionality reduction that can still preserve informative features in the data [13], [14]. Furthermore, such data are often generated in a streaming fashion and need to be processed in real time. Thus, processing algorithms are constrained to be of low computational complexity. Based on the data, fast and accurate inference must be performed to detect anomalies and disturbances occurring in the systems. As the data volume grows large, it is natural to have missing and corrupt entries in the data due to various reasons,

This work was supported in part by US National Science Foundation (NSF) grants 1547347 and 1631838, the National IT Industry Promotion Agency (NIPA) grant funded by the Korean government (MSIT: Ministry of Science and ICT) (No. 1711171061), and the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry and Energy (MOTIE) of the Republic of Korea (No. 20222A10100030).

such as sensor failure and communication errors. In critical infrastructures, the measurements may also be tampered by cyber-attacks, which must be detected and isolated.

Exploiting inherent structures in the data shows promises in mitigating these challenges. Missing measurements in power grid data were reconstructed using low-rank matrix completion approaches in [15]. Subspace clustering approaches were devised and tested for synchrophasor data [16], [17]. Measurements deviating from the postulated models were detected as indicating disturbance events or cyber-attacks [17]–[19]. The classification problem to detect the attacks in the smart grid was tackled using supervised and semi-supervised methods [20], [21]. However, these works were based on batch implementations, where the processing takes place offline after measurements have been collected, or the model is re-trained frequently. Thus, the methods may incur large computational and memory burden and may not be suitable for real-time monitoring applications. Deep learning models, such as the convolutional neural network (CNN), deep autoencoder (DAE), generative adversarial network (GAN), and long short-term memory (LSTM) network, were employed to detect anomalies in power system data, but they typically demand large datasets and long training time [22]–[27].

To mitigate these issues, online algorithms are desired, which can process the data stream sequentially and incrementally. Online identification of low-dimensional subspaces from data with missing entries was tackled in [28]. Online algorithms for robust principal component analysis (PCA) were derived for processing outlier-corrupted data with low-dimensional structures in [29], [30]. Online learning of sparse coding dictionary models was proposed in [31]. An online nonnegative matrix factorization algorithm was derived in [32]. An online algorithm for missing PMU data estimation was developed based on low rank matrix completion in [33].

In this paper, the subspace clustering model is adopted, in which data points are assumed to lie in a union of subspaces [34]. Such a model is readily justified since subspaces can capture different states of the dynamic system operations [3], [18]. While the subspace clustering model can subsume low-rank subspace models, it can capture the richer union-of-subspaces structure, and thus can be used in high-rank situations as well—i.e., when the sum of the dimensionalities of individual subspaces is higher than the ambient measurement dimension [16], [35].

A host of approaches have been developed for learning subspace clustering structures, including the k-plane clustering [36], generalized PCA [37], sparse subspace clustering [38], and low-rank representation (LRR) [39]. Here the LRR framework is employed for its good performance and extendability to online implementation. It is noted that an online algorithm for the LRR model was developed in [40]. However, in addition to neglecting the missing data issue and employing a costly second-order update rule, it was suggested in [40] to use the entire data matrix itself as the representative dictionary. Therefore, the algorithm can be started only after the data collection is done. If the initial portion of the streaming data is used as the dictionary, the dictionary may lose the representative power if the data distribution changes over time.

In our work, the dictionary for subspace representation is updated in an online fashion as well. More specifically, the dictionary atoms are selected incrementally yet judiciously based on an appropriate sparsification criterion. Furthermore, to maintain the dictionary size under a given memory budget, the atoms are adaptively discarded, which is called a pruning procedure.

Some preliminary results were presented in conferences [17], [41]. Compared to these precursors, the current journal version contains slightly different formulations for the consistency of batch and online problems. Furthermore, rigorous proofs are provided for the convergence of the derived batch and online algorithms. Extended numerical tests are performed with synthetic data using grids of different sizes, and with real PMU data. Various aspects of the performance of the proposed algorithms are also compared with existing alternatives that adopt low-rank models of data.

Our contributions can be summarized as follows.

1) We formulate a novel LRR-based robust subspace clustering problem accommodating both incomplete and corrupt measurements.
2) Both batch and online algorithms are derived with provable convergence guarantees. The online algorithm enjoys low computational complexity, processing delay, and storage overhead. In addition, it can track slow variations in the underlying dynamics. Online sparsification and pruning procedures are also proposed to maintain a compact dictionary to represent the subspaces.
3) The algorithms are applied to the synchrophasor data analysis for power grid monitoring. Numerical tests are performed to validate the performance of reconstructing missing measurements and detecting anomalous events using both simulated and real PMU data. The algorithms are compared with existing alternatives and shown to be significantly superior.

The rest of the paper is organized as follows. In Section II, the subspace clustering formulation with missing and corrupt data is presented. The batch and online algorithms are derived, and then strategies to update the dictionaries on the fly are discussed in Section III. Results from the numerical tests are presented in Section IV. The conclusion is provided in Section V.

II. Low Rank Representation For Incomplete Data

Let us denote the N-channel measurements obtained from various sensors at time t as $\mathbf{z}_t \in \mathbb{R}^N$. Collecting the measurements over T time intervals, matrix $\mathbf{Z}$ is formed as $\mathbf{Z} := [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T] \in \mathbb{R}^{N \times T}$. The subspace clustering model postulates that a (noise-free) measurement vector $\bar{\mathbf{z}}_t$ lies in one of K subspaces $\{\mathcal{S}_k\}_{k=1}^K$; that is, $\bar{\mathbf{z}}_t \in \cup_{k=1}^K \mathcal{S}_k$. Such a model is useful for PMU data since small variations in the node voltages lead to the measurements that are well approximated by a linear subspace, and significant variations in the operating point or disturbances in the system would result in measurements that belong to different subspaces [3], [16]. Thus, $\bar{\mathbf{z}}_t$ can be represented by a linear combination of a set of template vectors, called *atoms*, which are collected as the columns in a dictionary matrix $\mathbf{D} \in \mathbb{R}^{N \times M}$, where M is the dictionary size or the number of atoms. Then, with $\mathbf{c}_t \in \mathbb{R}^M$ representing the vector of coefficients, one has $\bar{\mathbf{z}}_t \approx \mathbf{D}\mathbf{c}_t$.

Interestingly, upon defining $\mathbf{C} := [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_T] \in \mathbb{R}^{M \times T}$, the LRR model insists that $\mathbf{C}$ has a rank that is much smaller than $\min\{M, T\}$. It turns out that this constraint can reveal the subspace clustering structure. More specifically, consider the optimization problem with $\bar{\mathbf{Z}} := [\bar{\mathbf{z}}_1, \ldots, \bar{\mathbf{z}}_T]$

$$\min_{\mathbf{C}} \; \|\mathbf{C}\|_* \tag{1a}$$

$$\text{subject to } \bar{\mathbf{Z}} = \mathbf{D}\mathbf{C} \tag{1b}$$

where $\|\mathbf{C}\|_*$ denotes the nuclear norm of $\mathbf{C}$, or the sum of its singular values. Minimizing $\|\mathbf{C}\|_*$ is tantamount to promoting a low rank in $\mathbf{C}$ [42]. If $\mathbf{D} = \bar{\mathbf{Z}}$ and $\bar{\mathbf{Z}}$ has the skinny SVD given by $\bar{\mathbf{Z}} = \mathbf{U}_0 \mathbf{\Sigma}_0 \mathbf{V}_0^\top$ ($^\top$ denotes transposition), it can be shown that the solution of (1) is $\mathbf{C} = \mathbf{V}_0 \mathbf{V}_0^\top$ [39]. Under the assumption that the data $\{\bar{\mathbf{z}}_t\}$ are clean, that is, they lie exactly in the union of subspaces, and the set of subspaces $\{\mathcal{S}_k\}$ are independent, which means that the dimension of the union (the rank of $\bar{\mathbf{Z}}$) is equal to the sum of the individual dimensions of the subspaces, it can be shown that the (i, j)-th entry of $\mathbf{V}_0 \mathbf{V}_0^\top$ is nonzero only if $\bar{\mathbf{z}}_i$ and $\bar{\mathbf{z}}_j$ belong to the same subspace [43]. Thus, $\mathbf{V}_0 \mathbf{V}_0^\top$ can serve as the *affinity matrix*, which indicates whether a given pair of data points lie in the same subspace. In fact, various subspace clustering techniques first compute the affinity matrix, and then extract the clusters [34]. Note also that if $\mathbf{D}$ contains the atoms that span the individual subspaces $\{\mathcal{S}_k\}$, then the resulting $\mathbf{C}$ can still be used as the affinity matrix [39].

As the data may be collected at a high rate and transported over some communication infrastructure, it may happen that some measurements do not arrive at the processing unit on time, resulting in *incomplete* measurements. Let Ω denote the set of indices for the entries of $\mathbf{Z}$ corresponding to the observed measurements, and Ω^c the missing ones. Also, define operator $\mathcal{P}_\Omega(\mathbf{Z})$, which keeps the observed entries unchanged, while setting the missing ones to zero. In other words, with the (i, j)-entry of $\mathbf{Z}$ denoted as Z_{ij}, the (i, j)-entry of $\mathcal{P}_\Omega(\mathbf{Z})$ is defined as

$$[\mathcal{P}_\Omega(\mathbf{Z})]_{ij} := \begin{cases} Z_{ij}, & \text{if } (i,j) \in \Omega \\ 0, & \text{if } (i,j) \in \Omega^c. \end{cases} \tag{2}$$

In addition, the measurements may become corrupted by gross errors in the communication channel, or even by cyber-attacks. Such measurements need to be detected and isolated. Anomalous measurements containing significant deviations from nominal operational states also need to be identified. Assuming that only a small portion of the data matrix $\mathbf{Z}$ is corrupted, one can adopt a robust LRR model, which postulates that $\mathbf{Z} = \mathbf{D}\mathbf{C} + \mathbf{E}$, where $\mathbf{E} \in \mathbb{R}^{N \times T}$ is a sparse matrix. Overall, an optimization problem for robust LRR with incomplete measurements can be posed as

$$\min_{\mathbf{C},\mathbf{E}} \; \|\mathbf{C}\|_* + \mu \|\mathbf{E}\|_1 \tag{3a}$$

$$\text{subject to } \mathcal{P}_\Omega(\mathbf{Z}) = \mathcal{P}_\Omega(\mathbf{D}\mathbf{C} + \mathbf{E}) \tag{3b}$$

where $\|\mathbf{E}\|_1$ is the ℓ_1-norm of $\mathbf{E}$ defined as the sum of absolute values of all entries in $\mathbf{E}$, and $\mu > 0$ is a parameter that balances the low rank of $\mathbf{C}$ and the sparseness of $\mathbf{E}$. Group or other structured sparsity can be easily incorporated to capture the correlations in the corruption patterns.

It can be easily verified that at the optimum, $\mathbf{E}$ will have the entries in Ω^c equal to 0, or $\mathbf{E}|_{\Omega^c} = \mathbf{0}$. Therefore, the optimal objective of (3) is equal to that of

$$\min_{\mathbf{C},\mathbf{E}} \; \|\mathbf{C}\|_* + \mu \|\mathcal{P}_\Omega(\mathbf{E})\|_1 \tag{4a}$$

$$\text{subject to } \mathcal{P}_\Omega(\mathbf{Z}) = \mathbf{D}\mathbf{C} + \mathbf{E}. \tag{4b}$$

The optimal $\mathbf{C}$ for (3) will be identical to the optimal $\mathbf{C}$ for (4). Once the optimal $\mathbf{E}$ for (4) is denoted as $\overline{\mathbf{E}}$, the optimal $\mathbf{E}$ for (3) is simply given by $\mathbf{E} = \mathcal{P}_\Omega(\overline{\mathbf{E}})$.

III. Algorithm Derivation

A. Batch Algorithm

An iterative algorithm to solve (4) can be derived, which processes the entire data set $\mathbf{Z}$ in a batch fashion. The batch solution is useful for offline analysis of historical data and serves as a performance benchmark for the online algorithm that will be developed in Section III-B.

To solve (4) efficiently, the alternating direction method of multipliers (ADMM) is employed [44]. First, a copy of variable $\mathbf{C}$ is introduced as $\overline{\mathbf{C}}$ to obtain a formulation equivalent to (4) as

$$\min_{\mathbf{C},\mathbf{E},\overline{\mathbf{C}}} \; \|\overline{\mathbf{C}}\|_* + \mu \|\mathcal{P}_\Omega(\mathbf{E})\|_1 \tag{5a}$$

$$\text{subject to } \mathcal{P}_\Omega(\mathbf{Z}) = \mathbf{D}\mathbf{C} + \mathbf{E} \tag{5b}$$

$$\mathbf{C} = \overline{\mathbf{C}}. \tag{5c}$$

Upon introducing Lagrange multiplier matrices $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ associated with constraints (5b) and (5c), the augmented Lagrangian can be written as

$$\begin{aligned} &L_\rho(\mathbf{C}, \mathbf{E}, \overline{\mathbf{C}}; \mathbf{\Lambda}_1, \mathbf{\Lambda}_2) \\ &:= \|\overline{\mathbf{C}}\|_* + \mu \|\mathcal{P}_\Omega(\mathbf{E})\|_1 + \langle \mathbf{\Lambda}_1, \mathcal{P}_\Omega(\mathbf{Z}) - \mathbf{D}\mathbf{C} - \mathbf{E} \rangle \\ &+ \langle \mathbf{\Lambda}_2, \mathbf{C} - \overline{\mathbf{C}} \rangle + \frac{\rho}{2} \|\mathcal{P}_\Omega(\mathbf{Z}) - \mathbf{D}\mathbf{C} - \mathbf{E}\|_F^2 + \frac{\rho}{2} \|\mathbf{C} - \overline{\mathbf{C}}\|_F^2 \end{aligned} \tag{6}$$

where $\rho > 0$ is a positive constant, and $\|\cdot\|_F$ denotes the Frobenius norm. The ADMM is an iterative algorithm that minimizes L_ρ alternatingly with respect to two blocks of variables. Here, $\mathbf{C}$ is taken as one block of variables, and $(\overline{\mathbf{C}}, \mathbf{E})$ as the other block. Then, at iteration k, based on the k-th (current) iterates $\mathbf{C}_k, \overline{\mathbf{C}}_k, \mathbf{E}_k, \mathbf{\Lambda}_1^k, \mathbf{\Lambda}_2^k$, the ADMM generates the next iterates as

$$\overline{\mathbf{C}}_{k+1}, \mathbf{E}_{k+1} = \arg\min_{\overline{\mathbf{C}},\mathbf{E}} L_\rho(\mathbf{C}_k, \mathbf{E}, \overline{\mathbf{C}}; \mathbf{\Lambda}_1^k, \mathbf{\Lambda}_2^k) \tag{7}$$

$$\mathbf{C}_{k+1} = \arg\min_{\mathbf{C}} L_\rho(\mathbf{C}, \mathbf{E}_{k+1}, \overline{\mathbf{C}}_{k+1}; \mathbf{\Lambda}_1^k, \mathbf{\Lambda}_2^k) \tag{8}$$

$$\begin{bmatrix} \mathbf{\Lambda}_1^{k+1} \\ \mathbf{\Lambda}_2^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{\Lambda}_1^{k} \\ \mathbf{\Lambda}_2^{k} \end{bmatrix} + \rho \begin{bmatrix} \mathcal{P}_\Omega(\mathbf{Z}) - \mathbf{D}\mathbf{C}_{k+1} - \mathbf{E}_{k+1} \\ \mathbf{C}_{k+1} - \overline{\mathbf{C}}_{k+1} \end{bmatrix}. \tag{9}$$

TABLE I
BATCH ROBUST SUBSPACE CLUSTERING ALGORITHM.

```
Input: $\Omega$, $\mathcal{P}_\Omega(\mathbf{Z})$, $\mathbf{D}$, $\mu > 0$, $\rho > 0$, $tol > 0$
Output: $\mathbf{C}$ and $\mathbf{E}$
Initialize: $\mathbf{C}_0$, $\mathbf{E}_0$, $\mathbf{\Lambda}_1^0$, $\mathbf{\Lambda}_2^0$, and $k = 0$
While not converged
    Perform SVD: $\mathbf{C}_k + \frac{1}{\rho}\mathbf{\Lambda}_2^k = \mathbf{U\Sigma V}^\top$
    $\overline{\mathbf{C}}_{k+1} = \mathbf{U}\mathcal{S}_{1/\rho}(\mathbf{\Sigma})\mathbf{V}^\top$
    $\mathbf{E}_{k+1}|_\Omega = \mathcal{S}_{\mu/\rho}((\mathbf{Z} - \mathbf{DC}_k + \frac{1}{\rho}\mathbf{\Lambda}_1^k)|_\Omega)$
    $\mathbf{E}_{k+1}|_{\Omega^c} = (-\mathbf{DC}_k + \frac{1}{\rho}\mathbf{\Lambda}_1^k)|_{\Omega^c}$
    $\mathbf{C}_{k+1} = (\mathbf{D}^\top\mathbf{D} + \mathbf{I})^{-1}\Big(\overline{\mathbf{C}}_{k+1} + \mathbf{D}^\top(\mathcal{P}_\Omega(\mathbf{Z}) - \mathbf{E}_{k+1})$
              $+ \frac{1}{\rho}(\mathbf{D}^\top\mathbf{\Lambda}_1^k - \mathbf{\Lambda}_2^k)\Big)$
    $\mathbf{\Lambda}_1^{k+1} = \mathbf{\Lambda}_1^k + \rho(\mathcal{P}_\Omega(\mathbf{Z}) - \mathbf{DC}_{k+1} - \mathbf{E}_{k+1})$
    $\mathbf{\Lambda}_2^{k+1} = \mathbf{\Lambda}_2^k + \rho(\mathbf{C}_{k+1} - \overline{\mathbf{C}}_{k+1})$
    Check $\|\mathcal{P}_\Omega(\mathbf{Z}) - \mathbf{DC}_{k+1} - \mathbf{E}_{k+1}\|_F^2 < tol$
    and $\|\mathbf{C}_{k+1} - \bar{\mathbf{C}}_{k+1}\|_F^2 < tol$
    $k \leftarrow k + 1$
End while
Set $\mathbf{E}_k|_{\Omega^c} = \mathbf{0}$.
Return $\mathbf{C} = \mathbf{C}_k$ and $\mathbf{E} = \mathbf{E}_k$.
```

It is noted that (7) can be split into the optimizations with respect to $\overline{\mathbf{C}}$ and $\mathbf{E}$ separately. The optimization for $\overline{\mathbf{C}}$ can be written as

$$\overline{\mathbf{C}}_{k+1} = \arg\min_{\overline{\mathbf{C}}} \|\overline{\mathbf{C}}\|_* - \langle \mathbf{\Lambda}_2^k, \overline{\mathbf{C}} \rangle + \frac{\rho}{2}\|\mathbf{C}_k - \overline{\mathbf{C}}\|_F^2 \quad (10)$$

$$= \arg\min_{\overline{\mathbf{C}}} \|\overline{\mathbf{C}}\|_* + \frac{\rho}{2}\|\overline{\mathbf{C}} - \mathbf{C}_k - \frac{1}{\rho}\mathbf{\Lambda}_2^k\|_F^2. \quad (11)$$

Thus, with the SVD of $\mathbf{C}_k + \rho^{-1}\mathbf{\Lambda}_2^k = \mathbf{U\Sigma V}^\top$, and upon defining a soft-thresholding operator $\mathcal{S}_\mu(\cdot)$ as

$$S_\mu(x) := \begin{cases} x - \mu, & \text{if } x > \mu \\ x + \mu, & \text{if } x < -\mu \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

$\overline{\mathbf{C}}_{k+1}$ is given by [45]

$$\overline{\mathbf{C}}_{k+1} = \mathbf{U}\mathcal{S}_{1/\rho}(\mathbf{\Sigma})\mathbf{V}^\top \quad (13)$$

where $\mathcal{S}_{1/\rho}(\mathbf{\Sigma})$ applies the soft-thresholding operation entry-wise.

The optimization problem for $\mathbf{E}_{k+1}$ is given by

$$\mathbf{E}_{k+1} = \arg\min_{\mathbf{E}} \ \mu\|\mathcal{P}_\Omega(\mathbf{E})\|_1 - \langle \mathbf{\Lambda}_1^k, \mathbf{E} \rangle + \frac{\rho}{2}\|\mathcal{P}_\Omega(\mathbf{Z}) - \mathbf{DC}_k - \mathbf{E}\|_F^2 \quad (14)$$

whose closed-form solution is given by

$$\mathbf{E}_{k+1}|_\Omega = \mathcal{S}_{\mu/\rho}\left(\left(\mathbf{Z} - \mathbf{DC}_k + \frac{1}{\rho}\mathbf{\Lambda}_1^k\right)\Big|_\Omega\right) \quad (15)$$

$$\mathbf{E}_{k+1}|_{\Omega^c} = \left(-\mathbf{DC}_k + \frac{1}{\rho}\mathbf{\Lambda}_1^k\right)\Big|_{\Omega^c}. \quad (16)$$

Finally, the update equation for $\mathbf{C}_{k+1}$ can be derived from (8) as

$$\mathbf{C}_{k+1} = (\mathbf{D}^\top\mathbf{D} + \mathbf{I})^{-1}\Big(\overline{\mathbf{C}}_{k+1} + \mathbf{D}^\top(\mathcal{P}_\Omega(\mathbf{Z}) - \mathbf{E}_{k+1}) + \frac{1}{\rho}(\mathbf{D}^\top\mathbf{\Lambda}_1^k - \mathbf{\Lambda}_2^k)\Big) \quad (17)$$

where $\mathbf{I}$ is the $M \times M$ identity matrix.

The resulting batch robust subspace clustering (RSC) algorithm is described in Table I. Compared to a similar algorithm derived in [39], the algorithm in Table I can handle missing entries and provides a provable convergence guarantee. The convergence follows from the standard ADMM literature. On the contrary, the algorithm in [39] alternates among three blocks of variables, for which convergence has not been established in general [46].

Proposition 1: *(Convergence of the algorithm in Table I) The iterates* $\{\mathbf{C}_k, \mathbf{E}_k, \mathbf{\Lambda}_1^k, \mathbf{\Lambda}_2^k\}$ *generated from the batch algorithm are bounded, and every limit point of* $\{\mathbf{C}_k, \mathbf{E}_k\}$ *is an optimal solution to* (3).

Proof: See Appendix A.

B. Online Algorithm

Contrary to the batch analysis method that processes a bulk of measurements together, the online algorithm updates the estimates of $\mathbf{C}$ and $\mathbf{E}$ each time a new datum arrives. This allows low-delay real-time analysis. Since only the latest data sample is involved, the online update rules are typically of low complexity and require a small memory footprint as well.

To facilitate the derivation of an online algorithm, (4) is first re-written as an unconstrained problem as

$$\min_{\mathbf{C},\mathbf{E}} \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Z}) - \mathbf{DC} - \mathbf{E}\|_F^2 + \lambda\|\mathbf{C}\|_* + \mu\|\mathcal{P}_\Omega(\mathbf{E})\|_1 \quad (18)$$

where $\lambda > 0$ is a parameter tuning the rank of $\mathbf{C}$. Note that the nuclear norm of a matrix $\mathbf{C}$, whose rank is no larger than R, can be expressed as [42]

$$\|\mathbf{C}\|_* = \min_{\mathbf{A}\in\mathbb{R}^{M\times R}, \mathbf{B}\in\mathbb{R}^{T\times R}} \frac{1}{2}(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2) \quad \text{subject to } \mathbf{C} = \mathbf{AB}^\top. \quad (19)$$

Based on this, (18) is modified to [see also [47]]

$$\min_{\mathbf{A},\mathbf{B},\mathbf{E}} \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Z}) - \mathbf{DAB}^\top - \mathbf{E}\|_F^2 + \frac{\lambda}{2}(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2) + \mu\|\mathcal{P}_\Omega(\mathbf{E})\|_1. \quad (20)$$

Proposition 2: *If the optimal* $\mathbf{C}$ *for* (18) *has a rank no larger than* R*, then* (20) *achieves the same optimal objective as* (18).

Proof: Let $\mathbf{A}^*$ and $\mathbf{B}^*$ be the optimal solution to (20), and set $\mathbf{C}^* = \mathbf{A}^*\mathbf{B}^{*\top}$. If one minimizes $\frac{1}{2}(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2)$ subject to $\mathbf{C}^* = \mathbf{AB}^\top$ (call the corresponding optimal solution $\check{\mathbf{A}}$ and $\check{\mathbf{B}}$), one should be able to further reduce the objective of (20) by using $\check{\mathbf{A}}$ and $\check{\mathbf{B}}$. However, since $\mathbf{A}^*$ and $\mathbf{B}^*$ are already optimal for (20), one must have $\frac{1}{2}(\|\check{\mathbf{A}}\|_F^2 + \|\check{\mathbf{B}}\|_F^2) = \frac{1}{2}(\|\mathbf{A}^*\|_F^2 + \|\mathbf{B}^*\|_F^2) = \|\mathbf{C}^*\|_*$. ■

Let $\mathbf{b}_t$ represent the t-th column of $\mathbf{B}^\top$, i.e., $\mathbf{B}^\top = [\mathbf{b}_1, \cdots, \mathbf{b}_T]$, and $\mathbf{e}_t$ the t-th column of $\mathbf{E}$, i.e., $\mathbf{E} = [\mathbf{e}_1, \cdots, \mathbf{e}_T]$. The set Ω_t denotes the set of indices of the observed entries at time t, that is, $\Omega_t := \{n : (n, t) \in \Omega\}$.

TABLE II
ONLINE RSC ALGORITHM.

```
Input: $\Omega$, $\mathcal{P}_\Omega(\mathbf{Z})$, $\mathbf{D}$, $\lambda > 0$, $\mu > 0$, $\rho_t > 0$
Output: $\mathbf{A}$, $\{\mathbf{b}_t\}$, and $\{\mathbf{e}_t\}$
For $t = 1, 2, \ldots, T$
    Set $l = 0$ and $\mathbf{e}_t^0 = \mathbf{0}$
    Repeat
        Update $\mathbf{b}_t^{l+1}$ by (23)
        Update $\mathbf{e}_t^{l+1}$ by (24)–(25)
        $l \leftarrow l + 1$
    Until convergence
    Set $\mathbf{b}_t = \mathbf{b}_t^l$ and $\mathbf{e}_t = \mathbf{e}_t^l$
    Update $\mathbf{A}_t$ by (29)
    Set $\mathbf{e}_t|_{\Omega_t^c} = \mathbf{0}$
Next $t$
Return $\mathbf{A} = \mathbf{A}_T$, $\{\mathbf{b}_t\}$, and $\{\mathbf{e}_t\}$
```

Thanks to reformulation (20), the problem is now separable into different time slots and can be rewritten as

$$\min_{\mathbf{A},\{\mathbf{b}_t\},\{\mathbf{e}_t\}} \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{2}\|\mathcal{P}_{\Omega_t}(\mathbf{z}_t)-\mathbf{D}\mathbf{A}\mathbf{b}_t-\mathbf{e}_t\|_2^2 + \frac{\lambda}{2}\|\mathbf{b}_t\|_2^2+\mu\|\mathcal{P}_{\Omega_t}(\mathbf{e}_t)\|_1\right)+\frac{\lambda}{2T}\|\mathbf{A}\|_F^2. \tag{21}$$

The basic idea for deriving the online algorithm is to update $\mathbf{b}_t$ and $\mathbf{e}_t$ based on $\mathcal{P}_{\Omega_t}(\mathbf{z}_t)$ at each time t, without revisiting the past entries $\{\mathbf{b}_\tau\}$ and $\{\mathbf{e}_\tau\}$ for $\tau = 1, 2, \cdots, t-1$. Furthermore, $\mathbf{A}$ is updated based on the stochastic gradient descent (SGD) method to reduce computational complexity [48].

First, the update for $\mathbf{b}_t$ and $\mathbf{e}_t$ at time t is based on the previous iterate of $\mathbf{A}_{t-1}$ via solving

$$\{\mathbf{b}_t, \mathbf{e}_t\} = \arg\min_{\mathbf{b},\mathbf{e}} \frac{1}{2}\|\mathcal{P}_{\Omega_t}(\mathbf{z}_t)-\mathbf{D}\mathbf{A}_{t-1}\mathbf{b}-\mathbf{e}\|_2^2 + \frac{\lambda}{2}\|\mathbf{b}\|_2^2+\mu\|\mathcal{P}_{\Omega_t}(\mathbf{e})\|_1. \tag{22}$$

To solve this problem, the coordinate descent method is adopted, where $\mathbf{b}_t$ and $\mathbf{e}_t$ are obtained by fixing one and solving for the other alternately until they both converge. Let $\mathbf{e}_t|_{\Omega_t}$ and $\mathbf{e}_t|_{\Omega_t^c}$ denote the entries of $\mathbf{e}_t$ whose indices are in Ω_t and Ω_t^c, respectively. Then, the coordinate descent proceeds for $l = 0, 1, 2, \cdots$ as

$$\mathbf{b}_t^{l+1} = (\mathbf{A}_{t-1}^\top\mathbf{D}^\top\mathbf{D}\mathbf{A}_{t-1}+\lambda\mathbf{I})^{-1}\mathbf{A}_{t-1}^\top\mathbf{D}^\top(\mathcal{P}_{\Omega_t}(\mathbf{z}_t)-\mathbf{e}_t^l) \tag{23}$$

$$\mathbf{e}_t^{l+1}|_{\Omega_t} = \mathcal{S}_\mu\left(\mathcal{P}_{\Omega_t}(\mathbf{z}_t)-\mathbf{D}\mathbf{A}_{t-1}\mathbf{b}_t^{l+1}\right)|_{\Omega_t} \tag{24}$$

$$\mathbf{e}_t^{l+1}|_{\Omega_t^c} = (-\mathbf{D}\mathbf{A}_{t-1}\mathbf{b}_t^{l+1})|_{\Omega_t^c} \tag{25}$$

where $S_\mu(\mathbf{x})$ for vector $\mathbf{x}$ applies element-wise.

The update for $\mathbf{A}$ is based on the SGD method. The key observation is that as T increases, the objective of (21) approaches the expected value, thanks to the law of large numbers. That is, upon defining

$$h(\mathbf{b},\mathbf{e},\mathbf{A},\mathbf{z}_t,\Omega_t;\mathbf{D}) := \frac{1}{2}\|\mathcal{P}_{\Omega_t}(\mathbf{z}_t)-\mathbf{D}\mathbf{A}\mathbf{b}-\mathbf{e}\|_F^2 + \frac{\lambda}{2}\|\mathbf{b}\|_2^2+\mu\|\mathcal{P}_{\Omega_t}(\mathbf{e})\|_1+\frac{\lambda}{2T}\|\mathbf{A}\|_F^2 \tag{26}$$

and

$$g(\mathbf{A},\mathbf{z}_t,\Omega_t;\mathbf{D}) := \min_{\mathbf{b},\mathbf{e}} h(\mathbf{b},\mathbf{e},\mathbf{A},\mathbf{z}_t,\Omega_t;\mathbf{D}) \tag{27}$$

problem (21) tends to

$$\min_{\mathbf{A}} \mathbb{E}\left[g(\mathbf{A},\mathbf{z}_t,\Omega_t;\mathbf{D})\right] \tag{28}$$

where the expectation $\mathbb{E}[\cdot]$ is with respect to $\mathbf{z}_t$ and Ω_t. Instead of computing the gradient of the entire cost function, the SGD takes the *instantaneous* derivative using only the current data sample. Thus, the update rule for $\mathbf{A}$ becomes

$$\mathbf{A}_t = \mathbf{A}_{t-1}+\rho_t\Big[\mathbf{D}^\top\big(\mathcal{P}_{\Omega_t}(\mathbf{z}_t) - \mathbf{D}\mathbf{A}_{t-1}\mathbf{b}_t-\mathbf{e}_t\big)\mathbf{b}_t^\top-\frac{\lambda}{T}\mathbf{A}_{t-1}\Big] \tag{29}$$

where $\rho_t > 0$ is the step size. The derived online RSC algorithm is listed in Table II. The convergence of the algorithm is established in the following proposition.

Proposition 3: *(Convergence of the algorithm in Table II) Suppose that $\{\mathbf{z}_t\}$ are bounded and the iterates $\{\mathbf{A}_t\}$ generated by the algorithm in Table II are bounded as well. Then, for $\rho_t = 1/Lt$ with a large enough $L > 0$, $\{\mathbf{A}_t\}$ converge to the stationary point of* (28) *almost surely.*

Proof: See Appendix B.

C. Dictionary Update

Clearly, for the union-of-subspaces structure to be captured, $\mathbf{D}$ must contain the vectors that span the individual subspaces $\{\mathcal{S}_k\}$. If this is true, provided that the subspaces are independent, and the data samples are strictly drawn from $\cup_{k=1}^{K}\mathcal{S}_k$, it can be shown that $\mathbf{C}$ from (1) will have its (m,t)-entry equal to zero if the m-th atom in $\mathbf{D}$ and the t-th data sample $\mathbf{z}_t$ lie in different subspaces [39].

A choice for the dictionary often made in the literature is to use the batch dataset $\mathbf{Z}$ itself by setting $\mathbf{D} = \mathbf{Z}$ [40]. Obviously, such a choice does not result in truly online processing since the algorithm would then require the entire data set to be available first. One could instead make use of the historical data. However, this might incur sizable memory requirements. Moreover, the dictionary should still be updated from time to time to keep up with any slow drift in the data distribution. A desirable option would be to start with a small initial dictionary, which is then updated continuously based on the incoming data stream.

In this work, instead of learning the dictionary from the data as in the dictionary learning frameworks [31], it is constructed directly from the online measurement vectors. A critical issue in this context is to be selective in adding a new measurement into the dictionary so that the size of the dictionary does not grow excessively, but still the diversity of the atoms is maintained so that a good representation capability is achieved. For this, online sparsification and pruning procedures are proposed next.

1) Online Sparsification: Various online sparsification strategies have been developed, in particular, in the context of kernel-based adaptive filtering literature [49]–[51]. The main idea is to accept a new datum into the dictionary only when it is deemed to sufficiently contribute to the diversity of the dictionary based on an appropriate metric. Representative

TABLE III
ONLINE RSC ALGORITHM WITH DICTIONARY UPDATE.

```
Input: $\Omega$, $\mathcal{P}_\Omega(\mathbf{Z})$, $\lambda > 0$, $\mu > 0$, $\rho_t > 0$, $M_0$, $M$, $\delta^2$
Output: $\mathbf{D}$, $\mathbf{A}$, $\{\mathbf{b}_t\}$, and $\{\mathbf{e}_t\}$
 1: Initialize $\mathbf{D}_0 \in \mathbb{R}^{N \times M_0}$ and $\mathbf{A}_0$
 2: For $t = 1, 2, \ldots, T$
 3:     Set $l = 0$ and $\mathbf{e}_t^0 = \mathbf{0}$
 4:     Repeat
 5:         Update $\mathbf{b}_t^{l+1}$ by (23) with $\mathbf{D}$ replaced by $\mathbf{D}_{t-1}$
 6:         Update $\mathbf{e}_t^{l+1}$ by (24)–(25) with $\mathbf{D}$ replaced by $\mathbf{D}_{t-1}$
 7:         $l \leftarrow l + 1$
 8:     Until convergence
 9:     Set $\mathbf{b}_t = \mathbf{b}_t^l$ and $\mathbf{e}_t = \mathbf{e}_t^l$
10:     Update $\mathbf{A}_t$ by (29) with $\mathbf{D}$ replaced by $\mathbf{D}_{t-1}$
11:     Set $\mathbf{e}_t|_{\Omega_t^c} = \mathbf{0}$
    /* sparsification */
12:     If $\Omega_t^c = \varnothing$, set $\hat{\mathbf{z}}_t = \mathbf{z}_t$
13:     Otherwise go to line 23
        (Optionally, let $\hat{\mathbf{z}}_t = \mathbf{D}_{t-1}\mathbf{A}_{t-1}\mathbf{b}_t$ and proceed to the next line)
14:     If $\min_{m \in \{1,\ldots,M_{t-1}\}} [\|\hat{\mathbf{z}}_t\|_2^2 - (\hat{\mathbf{z}}_t^\top \check{\mathbf{z}}_m)^2 / \|\check{\mathbf{z}}_m\|_2^2] \geq \delta^2$
15:         $\mathbf{D}_t = [\mathbf{D}_{t-1}, \hat{\mathbf{z}}_t]$
16:         $M_t = M_{t-1} + 1$
17:     Otherwise $\mathbf{D}_t = \mathbf{D}_{t-1}$ and $M_t = M_{t-1}$
    /* pruning */
18:     If $M_t > M$
19:         Find $m^* := \arg\min_{m \in \{1,\ldots,M\}} \|\mathbf{a}_t(m,:)\|_2^2$
20:         Remove the $m^*$-th column of $\mathbf{D}_t$ and $m^*$-th row of $\mathbf{A}_t$
21:         $M_t = M_t - 1$
22:     End if
23: Next $t$
24: Return $\mathbf{D} = \mathbf{D}_T$, $\mathbf{A} = \mathbf{A}_T$, $\{\mathbf{b}_t\}$, and $\{\mathbf{e}_t\}$
```

sparsification criteria include minimum pairwise distance, approximate linear dependence, coherence, and Babel measure, all of which can be shown to eventually upper-bound the condition number of the kernel (Gram) matrix [51]. In this work, the minimum pairwise distance measure is adopted for its simplicity.

Let us first focus on the case where there are no missing entries in the data. Suppose that at time $t-1$, M_{t-1} data vectors $\check{\mathbf{z}}_1, \check{\mathbf{z}}_2, \ldots, \check{\mathbf{z}}_{M_{t-1}}$ are in the dictionary as $\mathbf{D}_{t-1} = [\check{\mathbf{z}}_1, \check{\mathbf{z}}_2, \ldots, \check{\mathbf{z}}_{M_{t-1}}]$, where $\check{\mathbf{z}}_m \in \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{t-1}\}$. Then, at time t, a new datum $\mathbf{z}_t$ arrives. The distance metric computed for $\mathbf{z}_t$ is given by

$$\kappa_t := \min_{m \in \{1,\ldots,M_{t-1}\}} \min_{\zeta} \|\mathbf{z}_t - \zeta \check{\mathbf{z}}_m\|_2^2 \tag{30}$$

$$= \min_{m \in \{1,\ldots,M_{t-1}\}} \|\mathbf{z}_t\|_2^2 - \frac{(\mathbf{z}_t^\top \check{\mathbf{z}}_m)^2}{\|\check{\mathbf{z}}_m\|_2^2}. \tag{31}$$

If κ_t is greater than some threshold δ^2, then $\mathbf{z}_t$ is admitted to the dictionary. That is, one sets $M_t = M_{t-1} + 1$ and $\check{\mathbf{z}}_{M_t} = \mathbf{z}_t$. Otherwise, $\mathbf{z}_t$ is discarded. When there are missing entries, that is, $\Omega_t^c \neq \varnothing$, a simple pragmatic strategy is to discard this measurement, which is used in the numerical tests. An alternative would be to use the reconstructed vector $\hat{\mathbf{z}}_t = \mathbf{D}_{t-1}\mathbf{A}_{t-1}\mathbf{b}_t$ in place of $\mathbf{z}_t$.

Through sparsification, the size of the dictionary M_t tends only to increase as time goes on. One can show that as long as the data vectors $\mathbf{z}_t$ belong to a compact set, the size of the dictionary does not increase without bound [49]. However, the resulting size of the dictionary may still be too large in practice. To contain the size of the dictionary within a prescribed memory budget, a pruning procedure may be employed as explained next.

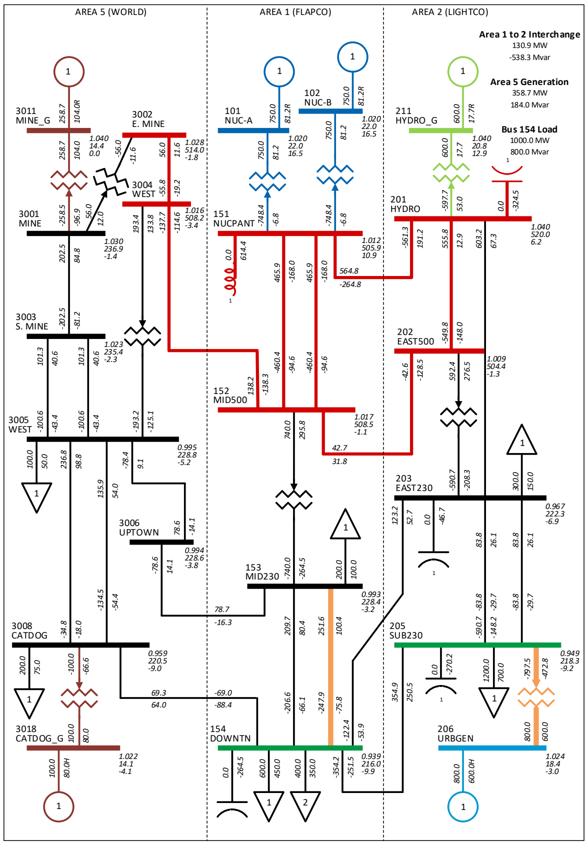

Fig. 1. A 23-bus power system.

2) Pruning: When M_t exceeds a memory budget M for the dictionary, that is, from the last sparsification step, $M_t = M + 1$ results, pruning removes an atom from the current dictionary $\mathbf{D}_t$. A reasonable strategy is to discard the atom that has the least contribution for representing the data seen so far [52]. Thus, upon denoting the m-th row of $\mathbf{A}_t$ as $\mathbf{a}_t(m,:)$, one finds

$$m^* = \arg \min_{m \in \{1,2,\ldots,M\}} \|\mathbf{a}_t(m,:)\|_2^2 \tag{32}$$

and $\check{\mathbf{z}}_{m^*}$ is removed from $\mathbf{D}_t$. The corresponding row in $\mathbf{A}_t$ should also be removed before the next iteration. Note that the atom that was just added to the dictionary is not pruned immediately. An alternative strategy would be to look at $\{\mathbf{c}_t := \mathbf{A}_{t-1}\mathbf{b}_t\}$, but this is not pursued here since it requires a larger memory overhead. The overall online RSC algorithm including the dictionary update is described in detail in Table III.

IV. Numerical Tests

The performance of the proposed algorithms was verified by numerical tests. Average performance values were obtained based on 20 runs. First, the results using the simulated PMU data are reported, followed by the real PMU data results.

A. Tests with Simulated PMU Data

1) A 23-bus System: The algorithms were first tested on simulated PMU data generated using the PSS/E simulator [53].

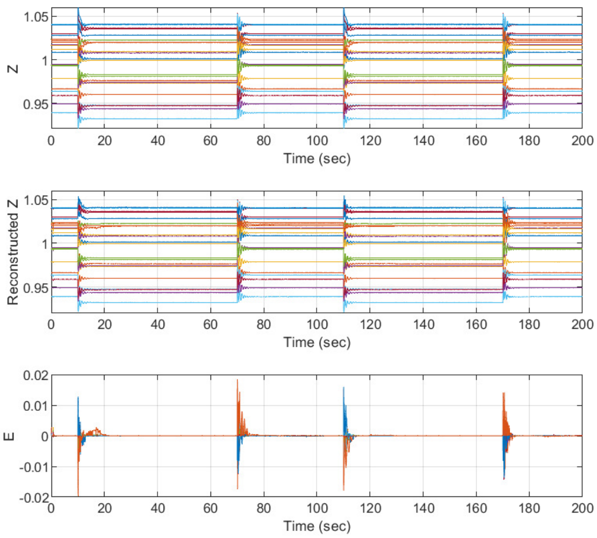

Fig. 2. Online algorithm results for simulated PMU data. (Top) Simulated measurements. (Middle) Reconstruction from data with 5% of the entries missing. (Bottom) Estimated sparse component matrix.

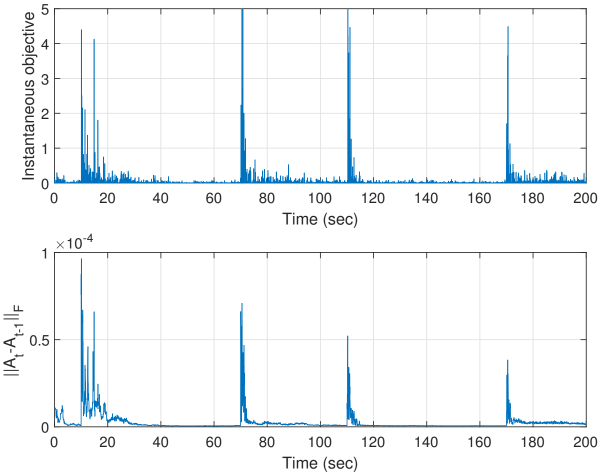

Fig. 3. Convergence of the online algorithm.

The simulated power system consists of 23 buses and 6 generators, as shown in Fig. 1. The PMU at each bus acquires measurements at a sampling rate of 40 samples per second and a signal-to-noise power ratio (SNR) of 92 dB. More detailed grid parameters are provided in [17]. To simulate events, the transmission line connecting buses 3001 and 3003 was tripped at $t = 10$ seconds and closed at $t = 70$ seconds. Then, the same line was tripped again at $t = 110$ seconds and closed back at $t = 170$ seconds. This is to ensure the convergence of online algorithms in the first half of the data so that the algorithm performance can be assessed accurately in the steady state during the second half. For simplicity, only the voltage magnitudes were used in the experiment. Employing multiple modalities, such as using the magnitudes and the angles together, resulted in very similar results. The voltage magnitudes are shown in the top panel of Fig. 2. For preprocessing, the nominal per-unit quantity 1 was subtracted to construct the data matrix $\mathbf{Z}$. To verify that the proposed algorithms can accurately recover missing measurements, 5% of the entries in $\mathbf{Z}$ were randomly removed. We also manually identified the outliers in $\mathbf{Z}$ as the part from the beginning of an event to when the system returns to its steady state. Let O denote the set of the time indices of the events (outliers). Then, $\mathbf{Z}|_O$ and $\mathbf{Z}|_{O^c}$ represent the outlier and the inlier columns of $\mathbf{Z}$, respectively.

The middle panel of Fig. 2 shows the voltage magnitudes reconstructed using the algorithm in Table III. The memory budget M was set to 100, and $\lambda = 10^{-2}$, $\mu = 5 \times 10^{-3}$, $\rho_t = 10^{-2}$, and $\delta^2 = 10^{-6}$ were used. The initial dictionary $\mathbf{D}_0$ was constructed by randomly choosing $M_0 = 5$ columns from $\mathbf{Z}|_{O^c}$. The maximum rank R of the low-rank matrix $\mathbf{C}$ was set to 10 based on the singular value distribution of $\mathbf{Z}$. It can be seen that the reconstructed voltage magnitudes match well with the true values. The bottom panel of Fig. 2 shows the sparse error component $\mathbf{E}$ estimated from the algorithm. It is seen that the estimated sparse errors indicate the disturbance events accurately.

To check the convergence of the algorithm, the instantaneous objective $h(\mathbf{b}_t, \mathbf{e}_t, \mathbf{A}_t, \mathbf{z}_t, \Omega_t; \mathbf{D}_t)$ in (26) is plotted in the top panel of Fig. 3. The objective rapidly converges both initially and after each disturbance event. In the bottom panel of Fig. 3, the Frobenius norm of the difference of the consecutive iterates of $\{\mathbf{A}_t\}$, namely $\|\mathbf{A}_t - \mathbf{A}_{t-1}\|_F$, is plotted. It can be seen that whenever there is a disturbance in $\mathbf{z}_t$, $\mathbf{A}_t$ tries to track it. However, as the effects of the disturbances vanish quickly as seen in Fig. 2, $\mathbf{A}_t$ captures only the nominal structure, and the disturbances are detected by the sparse error component.

Next, the performance of the proposed algorithms is compared with that of benchmark algorithms. Specifically, the online robust principal component analysis (ORPCA) algorithm [29] and the improved version [54] of the robust online subspace estimation and tracking algorithm (ROSETA) [55] are considered. The ORPCA objective in [29, Eq. (6)] is very similar to the objective function of (21), but it does not account for missing observations. Thus, the ORPCA formulation is modified here, which turns out to be the same as (21) with $\mathbf{D} = \mathbf{I}$. As for the improved ROSETA (iROSETA), it can be verified that the objective function can be expressed as

$$\frac{1}{T}\sum_{t=1}^{T}\left(\frac{\gamma}{2}\|\mathcal{P}_{\Omega_t}(\mathbf{z}_t) - \mathbf{A}\mathbf{b}_t - \mathbf{e}_t\|_2^2 + \mu\|\mathcal{P}_{\Omega_t}(\mathbf{e}_t)\|_1\right) \quad (33)$$

where $\mathbf{A}$ is again $M \times R$. Note also that μ is a parameter to be specified, whereas γ is adapted automatically by the algorithm. It can be seen that (33) lacks the regularizers on $\mathbf{A}$ and $\mathbf{b}_t$, which means that the rank of the subspace is fixed to R specified, and R no longer plays the role of the *maximum* rank. In the experiment, we set the values of R for the RSC and ORPCA algorithms equal to 10, but for iROSETA, both $R = 5$ and $R = 10$ were tested.

Fig. 4 depicts the receiver operating characteristic (ROC) of detecting the disturbance events. Each curve shows the tradeoff between the true positive rate and the false positive rate.

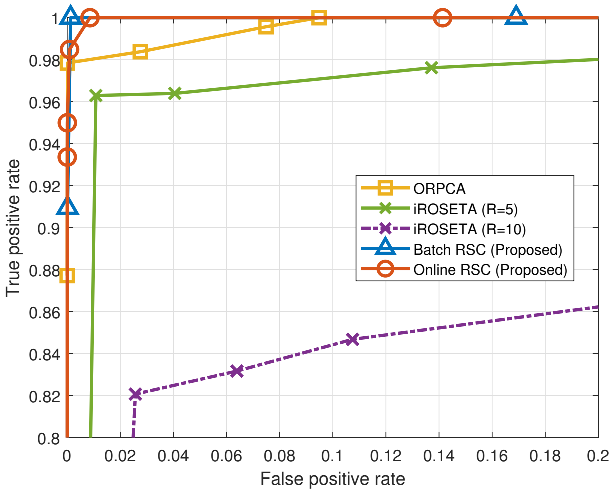

Fig. 4. Comparison of detection performances of disturbance events.

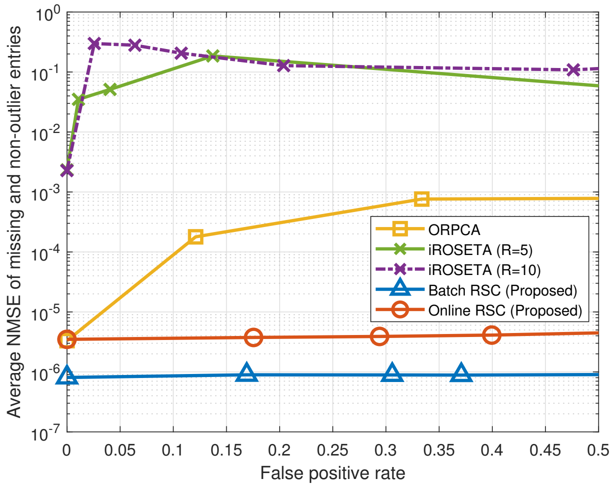

Fig. 5. NMSE performance for reconstructing missing observations.

The false positive rate is the probability that the algorithm erroneously detects an anomaly when the grid is in the normal condition. The true positive rate is the probability that the algorithm correctly detects a disturbance event during such an event. When a disturbance event happens, the sparse error vector $\mathbf{e}_t$ will have non-zero entries. To compute the true positive rate, we assume that the event is detected if any of the sparse error vectors during the event duration is non-zero. The curves with the triangle, circle, and square markers depict the results obtained by the batch RSC, online RSC, and ORPCA algorithms, respectively. The curves with the cross markers correspond to the performance of iROSETA, where the solid curve is for $R = 5$ and the dash-dot for $R = 10$. Each curve was obtained by varying μ (as well as λ for ORPCA and online RSC) in the corresponding algorithm. As expected, the batch RSC algorithm performs the best. Among the online algorithms, the online RSC algorithm is seen to be superior to other algorithms. In the case of iROSETA, the one with $R = 5$ yields better performance than $R = 10$, as the true rank of $\mathbf{Z}$ is in fact closer to 5.

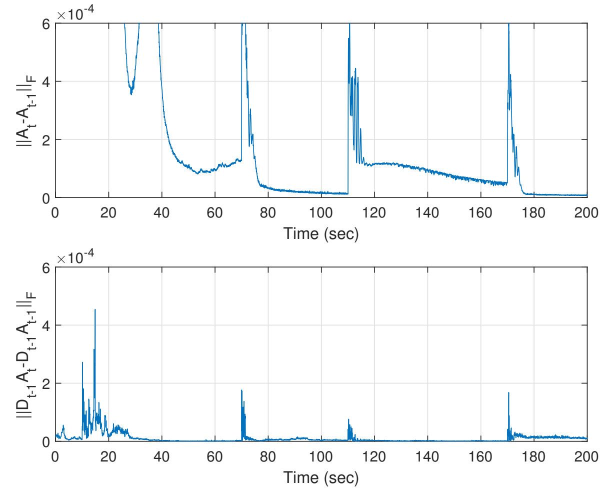

Fig. 6. Evolution of estimated low-rank subspaces. (Top) ORPCA. (Bottom) Online RSC.

We also compared the reconstruction performance of the missing observations. To do this, the average normalized mean-square error (NMSE) between the true and the reconstructed measurements is computed for the missing observations outside the disturbance events. (Recall that the missing observations cannot be reconstructed for outliers.) Specifically, the average NMSE is defined as

$$\frac{\|(\mathbf{Z} - \hat{\mathbf{Z}})|_{\Omega^c \cap \mathcal{O}^c}\|_F^2}{\|\mathbf{Z}|_{\Omega^c \cap \mathcal{O}^c}\|_F^2} \tag{34}$$

where $\hat{\mathbf{Z}} = \mathbf{DC}$. Since λ and μ not only affect the NMSE but also the event detection performance, one must hold the event detection performance of the algorithms at the same level for fair comparison. In Fig. 5, the average NMSEs are compared at different false positive rates. It can be seen again that among the online algorithms, online RSC achieves the smallest NMSEs across all false positive rates less than 0.5.

To see why online RSC performs better than ORPCA in terms of both event detection and missing entry estimation even though the formulations of the two algorithms are very similar, the evolution of the estimated low-rank subspaces is examined. In Fig. 6, the top panel shows the differences $\|\mathbf{A}_t - \mathbf{A}_{t-1}\|_F$ of successive iterates of $\mathbf{A}_t$ in ORPCA, where $\mathbf{A}_t$ spans the low-rank subspace at each time t. Likewise, the variation of the subspace in online RSC can be captured by $\|\mathbf{D}_{t-1}\mathbf{A}_t - \mathbf{D}_{t-1}\mathbf{A}_{t-1}\|_F$, which is depicted in the bottom panel of Fig. 6. Recall that the difference of the two algorithms lies in that in online RSC, dictionary $\mathbf{D}_t$ is continually estimated, while in ORPCA, essentially an identity matrix is used as the dictionary. It can be observed from the figure that ORPCA takes much more time to stabilize than online RSC, both initially as well as after disturbance events, even though ORPCA employs computationally costly block coordinate descent (BCD) while online RSC uses SGD. It can be deduced that the dynamic dictionary of online RSC with efficient sparsification and pruning strategies significantly improves the stability of the algorithm in the presence of outliers, which contributes to the performance.

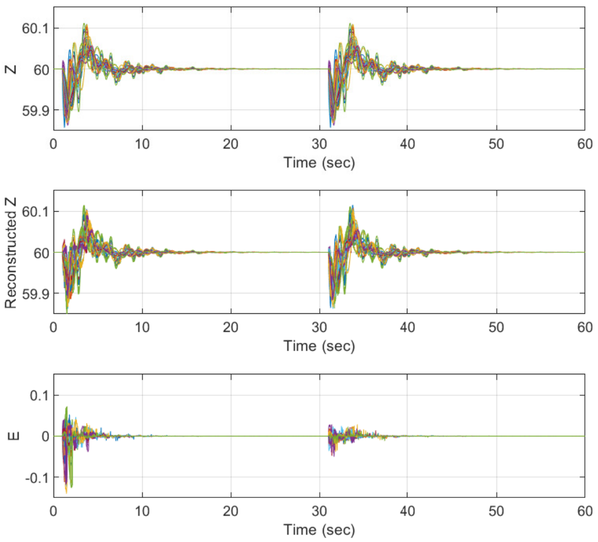

Fig. 7. (Top) Simulated frequency measurements for an IEEE 68-bus system. (Middle) Reconstruction with 5% missing entries. (Bottom) Estimated sparse error component.

2) An IEEE 68*-bus System:* The proposed algorithm was tested with a data set for a larger power system. The data set was generated for an IEEE 68-bus system using the GridSTAGE (Grid Spatio-Temporal Adversarial scenario GEneration) simulator, a multivariate spatio-temporal data generation tool for simulating adversarial scenarios [56], [57]. A sampling rate of 50 samples per second and an SNR of 92 dB were used. A load change event started at $t = 1$ second and ended at $t = 1.25$, which is repeated once more from $t = 31$ to $t = 31.25$. Note that even though an event occurs during 0.25 seconds, it takes about 10 seconds for the grid to return to its steady state. Thus, we view the entire period from the start of an event to the return to the steady state as the disturbance duration. Also, while the data set contains voltage magnitudes, voltage phase angles, and frequencies, only the frequencies (shown in the top panel of Fig. 7) were utilized in our experiment to show the various availability of the algorithm. The nominal frequency 60 Hz was subtracted from each entry when constructing the data matrix $\mathbf{Z}$ for preprocessing. To test the performance of estimating missing observations, 5% of the entries in $\mathbf{Z}$ were randomly obliterated.

The middle panel of Fig. 7 depicts the reconstructed one from the incomplete frequency measurement. Parameters were set as $\lambda = 10^{-4}$, $\mu = 10^{-6}$, and $\rho_t = 10^{-2}$. The sparsification and pruning procedures were utilized with $\delta^2 = 10^{-6}$. It can be seen that the reconstructed frequencies match the true ones faithfully. The bottom panel of Fig. 7 depicts the sparse error matrix $\mathbf{E}$, where the disturbance events are well captured.

The average NMSE performance of the online RSC, ORPCA, and iROSETA algorithms is shown in Fig. 8. The rank parameter R was set to 30 for online RSC, 50 for ORPCA, and 20 for iROSETA, which correspond to the best NMSE values for the individual algorithms. It is observed that online RSC achieves the best performance when the false positive rate is smaller than 0.2. It should be noted that operating with the false positive rate higher than 0.2 is hardly useful as the true positive rate of all three algorithms already reach 100% when the false positive rate is as small as 1%. Compared to the NMSE performance for the 23-bus system shown in Fig. 5, it is also observed that the NMSE performances for different algorithms are similar here. We suspect that this is due to the higher redundancy in the data resulting from a larger number of channels.

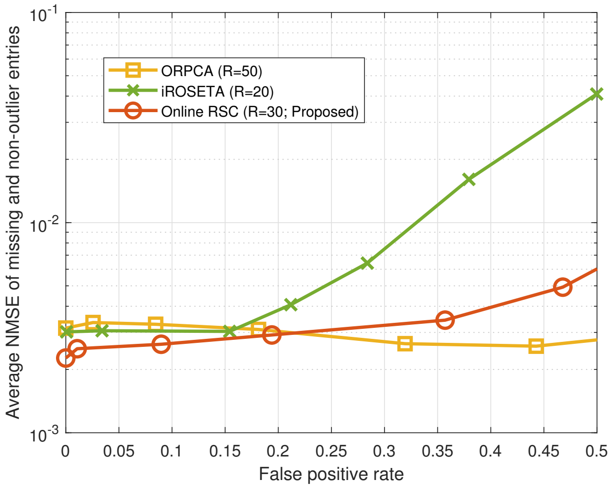

Fig. 8. Average NMSE for missing entry estimation for IEEE 68-bus system.

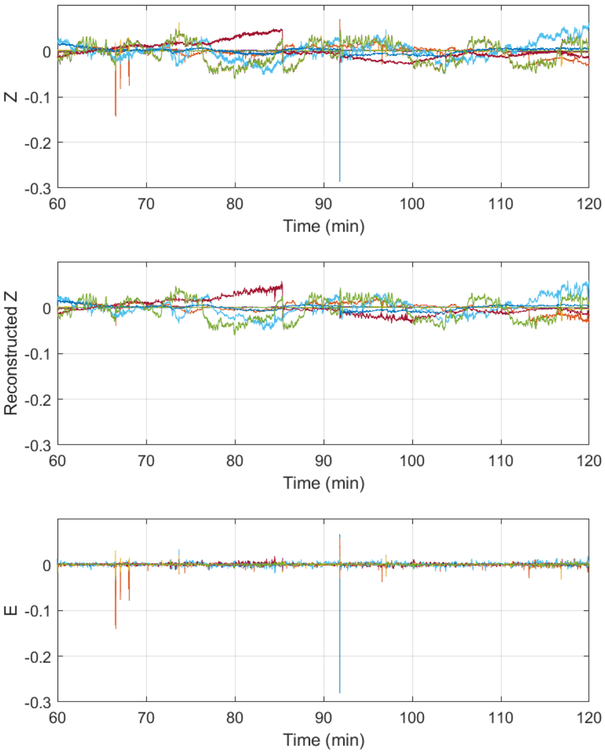

Fig. 9. Real PMU data experiment.

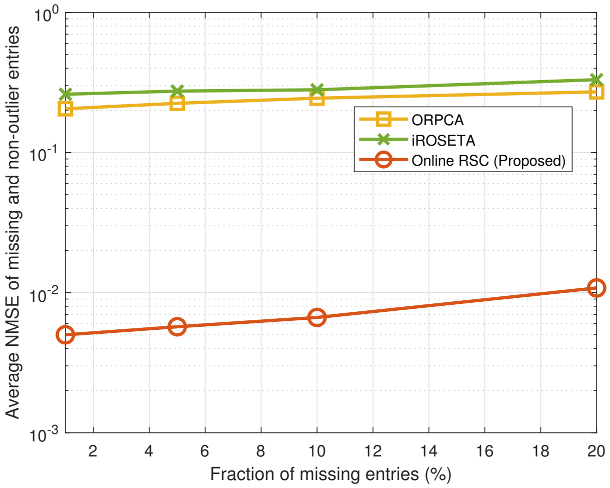

Fig. 10. Reconstruction performance comparison of missing entries for real PMU data.

B. Tests with Real PMU Data

The algorithm was tested with real PMU data acquired from the Texas ERCOT grid [58]. The data set is comprised of the measurements from four PMU stations, located at the Harris 69 kV Substation, McDonald Observatory, the University of Texas-Pan American, and Brazos Electric in Waco, taken for one hour at the rate of 30 samples per second. The data set contains the voltage magnitudes, phase angles, and frequencies. The voltage magnitudes were preprocessed by subtracting the mean and also dividing by the mean value. For the phase angles, the first bus was chosen as a reference, and the angle differences between other buses and the reference bus were calculated. Then, the mean of the differences were subtracted. For the frequencies, the nominal value 60 Hz was subtracted from the frequency measurements. The resulting data matrix is repeated once more to construct $\mathbf{Z}$ corresponding to a 2-hour duration.

The second half of $\mathbf{Z}$, containing the voltage magnitudes, phasor angles, and frequencies, is plotted in the top panel of Fig. 9. It is seen that the real data contain a few outliers, which may be due to actual events or corrupt measurements. The middle panel corresponds to the reconstruction from the online RSC algorithm using the data with 5% of the entries randomly missing. The bottom panel shows the sparse errors. We used $\lambda = 1$, $\mu = 5 \times 10^{-4}$, $\delta^2 = 10^{-6}$, and $\rho_t = 10^4$. The memory budget was set as $M_0 = 5$ and $M = 100$. It is seen that the sparse error captures all the prominent outliers. It also produces small nonzero values throughout, which may indicate events worth looking or could simply be ignored through appropriate thresholding, depending on the detailed requirements of grid monitoring.

Fig. 10 depicts the performance of recovering incomplete measurements as the missing percentage varies from 1% to 20%. The three curves correspond to the NMSE performances of online RSC, ORPCA, and iROSETA, which were obtained by fixing the false positive rates around 0.1, and varying μ and λ. The values of R were set to 10 for the online RSC and ORPCA, and to 5 for iROSETA, respectively, based on the singular value distribution of $\mathbf{Z}$. As can be seen, the online RSC algorithm performs much better than others. It can be noted that the recovered missing entries are quite accurate, even though the number of PMUs is small and realistic noise is included in the data.

V. Conclusion

A RSC formulation has been proposed that can reconstruct missing measurements and detect corrupt entries based on a union-of-subspaces structure present in the data. Both batch and online algorithms have been derived with convergence guarantees. The online RSC algorithm enjoys low computational complexity and small memory footprint—suitable for real-time processing of large-scale streaming data. To construct a representative yet compact dictionary for capturing the subspaces, online sparsification and pruning methods were also proposed. The algorithms were applied to synchrophasor measurement data for power grid monitoring. The numerical tests performed on simulated and real PMU data validated the effectiveness of the proposed algorithms. In particular, the online RSC algorithm with sparsification and pruning strategies was shown to achieve a performance on par with the batch counterpart using low-complexity updates, faithfully reconstructing missing entries and accurately capturing disturbance events. The performance of the proposed algorithm was also shown to outperform existing alternatives. Future research directions include incorporating nonconvex spectral regularizers for better performance [47], building event classifiers based on the reduced-dimensionality features, and distributed implementation for scalability.

Appendix A
Proof of Proposition 1

The claim can be proved by the straightforward application of Proposition 4.2 in [59, Sec. 3.4]. Specifically, it is clear that (5) can be written in the form of

$$\min_{x,z} G_1(x) + G_2(z)$$
$$\text{subject to } x \in C_1,\ z \in C_2,\ Ax = z \tag{35}$$

by defining

$$A := \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{D} & \mathbf{I} \end{bmatrix},\ x := \begin{bmatrix} \mathbf{C} \\ \mathbf{\Theta} \end{bmatrix},\ z := \begin{bmatrix} \bar{\mathbf{C}} \\ \mathbf{E} \end{bmatrix}, \tag{36}$$

$C_1 = \mathbb{R}^{M \times T} \times \{\mathcal{P}_\Omega(\mathbf{Z})\}$, $C_2 = \mathbb{R}^{(M+N)\times T}$, $G_1(x) = 0$, and $G_2(z) = \|\bar{\mathbf{C}}\|_* + \mu\|\mathcal{P}_\Omega(\mathbf{E})\|_1$. It is noted that $\mathbf{\Theta}$ is a variable constrained to be equal to constant $\mathcal{P}_\Omega(\mathbf{Z})$. Clearly G_1 and G_2 are closed and convex, and the optimal solution set of (5) can be assumed nonempty. ■

Appendix B
Proof of Proposition 3

The convergence can be established by viewing the SGD update as an instance of the stochastic successive upper-bound minimization (SSUM) algorithm for solving (28) [60]. To use the SSUM algorithm, one needs to construct a tight upper-bound $\hat{g}(\mathbf{A}, \bar{\mathbf{A}}, \mathbf{z}_t, \Omega_t)$ of $g(\mathbf{A}, \mathbf{z}_t, \Omega_t)$ at $\mathbf{A} = \bar{\mathbf{A}}$. To do this, first note that the solution $(\mathbf{b}_t, \mathbf{e}_t)$ to (27) is unique and thus Danskin's theorem can be invoked to assert that the gradient of g with respect to $\mathbf{A}$ is given by

$$\nabla_{\mathbf{A}} g(\mathbf{A}, \mathbf{z}_t, \Omega_t) = -\mathbf{D}^\top(\mathcal{P}_{\Omega_t}(\mathbf{z}_t) - \mathbf{D}\mathbf{A}\mathbf{b}_t - \mathbf{e}_t)\mathbf{b}_t^\top + \frac{\lambda}{T}\mathbf{A}. \tag{37}$$

Since $\mathbf{z}_t$ is bounded, $\nabla_{\mathbf{A}} g$ is Lipschitz continuous. Let L be the Lipschitz constant. Then, it can be shown that

$$\begin{aligned} \hat{g}(\mathbf{A}, \bar{\mathbf{A}}, \mathbf{z}_t, \Omega_t) &:= g(\bar{\mathbf{A}}, \mathbf{z}_t, \Omega_t) \\ &+ \langle \nabla_{\mathbf{A}} g(\bar{\mathbf{A}}, \mathbf{z}_t, \Omega_t), \mathbf{A} - \bar{\mathbf{A}} \rangle + \frac{L}{2}\|\mathbf{A} - \bar{\mathbf{A}}\|_F^2 \end{aligned} \tag{38}$$

is a tight upper-bound of $g(\mathbf{A}, \mathbf{z}_t, \Omega_t)$. That is,

$$\hat{g}(\bar{\mathbf{A}}, \bar{\mathbf{A}}, \mathbf{z}_t, \Omega_t) = g(\bar{\mathbf{A}}, \mathbf{z}_t, \Omega_t) \tag{39}$$

$$\hat{g}(\mathbf{A}, \bar{\mathbf{A}}, \mathbf{z}_t, \Omega_t) \geq g(\mathbf{A}, \mathbf{z}_t, \Omega_t) \text{ for all } \mathbf{A}. \tag{40}$$

Furthermore, it is verified that $\hat{g}$ is strongly convex in $\mathbf{A}$, g and $\hat{g}$ are continuous in $\mathbf{A}$, and g, $\hat{g}$, and their derivatives are bounded. Then, the SSUM iterates are given by

$$\mathbf{A}_t = \mathbf{A}_{t-1} - \rho_t \nabla_{\mathbf{A}} g(\mathbf{A}_{t-1}, \mathbf{z}_t, \Omega_t) \tag{41}$$

where $\rho_t = 1/Lt$, which is exactly the update in (29). Thus, $\mathbf{A}_t$ is guaranteed to converge to the stationary point of (28) almost surely [60, Theorem 1]. ■

References

[1] A. G. Phadke and J. S. Thorp, *Synchronized Phasor Measurements and Their Applications*. New York, NY: Springer, 2008.

[2] J. D. L. Ree, V. Centeno, J. S. Thorp, and A. G. Phadke, "Synchronized phasor measurement applications in power systems," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 20–27, Jun. 2010.

[3] J. M. Lim and C. L. DeMarco, "Model-free voltage stability assessments via singular value analysis of PMU data," in *Proc. IREP Symp. - Bulk Power Syst. Dyn. Contr. - IX Optim. Security and Contr. Emerg. Power Grid*, Rethymnon, Greece, Aug. 2013, pp. 1–10.

[4] H.-Y. Su and C.-W. Liu, "Estimating the voltage stability margin using PMU measurements," *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 3221–3229, Jul. 2016.

[5] Y. Lee, Y. Zhao, S.-J. Kim, and J. Li, "Predicting voltage stability margin via learning stability region boundary," in *Proc. IEEE Int. Wksp. Comput. Adv. Multi-Sensor Adapt. Process.*, Curacao, Dutch Antilles, Dec. 2017, pp. 1–5.

[6] J. Li, Y. Zhao, Y. Lee, and S.-J. Kim, "Learning to infer voltage stability margin using transfer learning," in *Proc. IEEE Data Sci. Wksp.*, Minneapolis, MN, Jun. 2019.

[7] A. Y. Abdelaziz, S. F. Mekhamer, M. Ezzat, and E. F. El-Saadany, "Line outage detection using support vector machine (SVM) based on the phasor measurement units (PMUs) technology," in *Proc. IEEE Power & Energy Soc. General Meeting*, San Diego, CA, Jul. 2012, pp. 1–8.

[8] Y. Zhao, J. Chen, and H. V. Poor, "Efficient neural network architecture for topology identification in smart grid," in *Proc. IEEE Global Conf. Signal Info. Process.*, Washington, DC, Dec. 2016, pp. 811–815.

[9] W. Li, M. Wang, and J. H. Chow, "Real-time event identification through low-dimensionality subspace characterization of high-dimensional synchrophasor data," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 4937–4947, Sep. 2018.

[10] W. Wang, L. He, P. Markham, H. Qi, Y. Liu, Q. C. Cao, and L. M. Tolbert, "Multiple event detection and recognition through sparse unmixing for high-resolution situational awareness in power grid," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1654–1664, Jul. 2014.

[11] M. Cui, J. Wang, J. Tan, A. R. Florita, and Y. Zhang, "A novel event detection method using PMU data with high precision," *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 454–466, Jan. 2019.

[12] J. Follum and J. W. Pierre, "Detection of periodic forced oscillations in power systems," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2423–2433, May 2016.

[13] M. Patel, S. Aivaliotis, E. Ellen *et al.*, "Real-time application of synchrophasors for improving reliability," North American Electric Reliability Corporation (NERC), Tech. Rep., Oct. 2010.

[14] N. Dahal, R. L. King, and V. Madani, "Online dimension reduction of synchrophasor data," in *Proc. IEEE PES Transm. Distrib. Conf. Expo.*, Orlando, FL, May 2012, pp. 1–7.

[15] P. Gao, M. Wang, S. G. Ghiocel, and J. H. Chow, "Modeless reconstruction of missing synchrophasor measurements," in *Proc. IEEE Power & Energy Soc. General Meeting*, National Harbor, MD, Jul. 2014, pp. 1–5.

[16] P. Gao, M. Wang, J. H. Chow, M. Berger, and L. M. Seversky, "Matrix completion with columns in union and sums of subspaces," in *Proc. IEEE Global Conf. Signal Info. Process.*, Orlando, FL, Dec. 2015, pp. 785–789.

[17] S.-J. Kim, Y. Lee, and K. Y. Lee, "Robust subspace approaches for analyzing incomplete synchrophasor measurements," in *Proc. 9th IFAC Symp. Contr. Power Energy Syst.*, vol. 48, no. 30. New Delhi, India: Elsevier, Dec. 2015, pp. 120–125.

[18] L. Xie, Y. Chen, and P. R. Kumar, "Dimensionality reduction of synchrophasor data for early event detection: Linearized analysis," *IEEE Trans. Power Syst.*, vol. 29, no. 6, pp. 2784–2794, Nov. 2014.

[19] M. Wang, J. H. Chow, P. Gao, and X. T. Jiang, "A low-rank matrix approach for the analysis of large amounts of power system synchrophasor data," in *Proc. 48th Hawaii Int'l. Conf. Syst. Sci.*, Kauai, HI, Jan. 2015, pp. 2637–2644.

[20] Y. Chen, L. Xie, and P. R. Kumar, "Power system event classification via dimensionality reduction of synchrophasor data," in *Proc. the 8th IEEE Sensor Array Multichannel Sig. Process. Wksp. (SAM)*, A Coruña, Spain, Jun. 2014, pp. 57–60.

[21] M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1773–1786, Mar. 2015.

[22] S. Basumallik, R. Ma, and S. Eftekharnejad, "Packet-data anomaly detection in PMU-based state estimator using convolutional neural network," *Int'l. J. Electr. Power Energy Syst.*, vol. 107, pp. 690–702, May 2019.

[23] J. Wang, D. Shi, Y. Li, J. Chen, H. Ding, and X. Duan, "Distributed framework for detecting PMU data manipulation attacks with deep autoencoders," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 4401–4410, Jul. 2019.

[24] X. Deng, D. Bian, W. Wang, Z. Jiang, W. Yao, W. Qiu, N. Tong, D. Shi, and Y. Liu, "Deep learning model to detect various synchrophasor data anomalies," *IET Gener. Transm. Distrib.*, vol. 14, no. 24, pp. 5739–5745, Dec. 2020.

[25] Y. Zhang, J. Wang, and B. Chen, "Detecting false data injection attacks in smart grids: A semi-supervised deep learning approach," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 623–634, Jan. 2021.

[26] I. Siniosoglou, P. Radoglou-Grammatikis, G. Efstathopoulos, P. Fouliras, and P. Sarigiannidis, "A unified deep learning anomaly detection and classification approach for smart grid environments," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 1137–1151, Jun. 2021.

[27] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Deep autoencoder-based anomaly detection of electricity theft cyberattacks in smart grids," *IEEE Syst. J.*, vol. 16, no. 3, pp. 4106–4117, Sep. 2022.

[28] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Proc. 48th Annu. Allerton Conf. Commun. Control Comput.*, Monticello, IL, Sep. 2010, pp. 704–711.

[29] J. Feng, H. Xu, and S. Yan, "Online robust PCA via stochastic optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, Lake Tahoe, NV, Dec. 2013, pp. 404–412.

[30] M. Mardani, G. Mateos, and G. B. Giannakis, "Dynamic anomalography: Tracking network anomalies via sparsity and low rank," *IEEE J. Sel. Top. Signal Process.*, vol. 7, no. 1, pp. 50–66, Feb. 2013.

[31] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, no. 2, pp. 19–60, Oct. 2010.

[32] R. Zhao and V. Y. F. Tan, "Online nonnegative matrix factorization with outliers," *IEEE Trans. Sig. Process.*, vol. 65, no. 3, pp. 555–570, Feb. 2017.

[33] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh, and G. Stefopoulos, "Missing data recovery by exploiting low-dimensionality in power system synchrophasor measurements," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1006–1013, Mar. 2016.

[34] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[35] B. Eriksson, L. Balzano, and R. Nowak, "High-rank matrix completion," in *Proc. Int. Conf. Artif. Intell. Stat.*, vol. 22, La Palma, Spain, Apr. 2012, pp. 373–381.

[36] P. S. Bradley and O. L. Mangasarian, "K-plane clustering," *J. Glob. Optim.*, vol. 16, no. 1, pp. 23–32, Jan. 2000.

[37] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945–1959, Dec. 2005.

[38] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[39] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[40] L. Li, B. Zou, and X. Zhang, "Online learning for low-rank representation and its application in subspace clustering," *J. Comput. Inf. Syst.*, vol. 10, no. 16, pp. 7125–7135, Aug. 2014.

[41] Y. Lee and S.-J. Kim, "Online robust subspace clustering for analyzing incomplete synchrophasor measurements," in *Proc. IEEE Global Conf. Signal Info. Process.*, Washington, DC, Dec. 2016, pp. 816–820.

[42] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, Aug. 2010.

[43] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *Int. J. Comput. Vis.*, vol. 29, no. 3, pp. 159–179, Sep. 1998.

[44] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jul. 2011.

[45] J.-F. Cai, E. J. Candés, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010.

[46] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Math. Program. Ser. A*, vol. 155, no. 1, pp. 57–79, Jan. 2016.

[47] X. Jia, X. Feng, W. Wang, and L. Zhang, "Generalized unitarily invariant gauge regularization for fast low-rank matrix recovery approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 1627–1641, Apr. 2021.

[48] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. New York, NY: Springer-Verlag, 2003.

[49] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least squares algorithm," *IEEE Trans. Sig. Process.*, vol. 52, no. 8, pp. 2275–2285, Jul. 2004.

[50] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Trans. Sig. Process.*, vol. 57, no. 3, pp. 1058–1067, Mar. 2009.

[51] P. Honeine, "Analyzing sparse dictionaries for online learning with kernels," *IEEE Trans. Sig. Process.*, vol. 63, no. 23, pp. 6343–6353, Dec. 2015.

[52] F. Sheikholeslami, D. Berberidis, and G. B. Giannakis, "Kernel-based low-rank feature extraction on a budget for big data streams," in *Proc. IEEE Global Conf. Signal Info. Process.*, Orlando, FL, Dec. 2015, pp. 928–932.

[53] Siemens, *PSS/E 33.5 Program Operation Manual*, 2013.

[54] H. Mansour, "A short note on improved ROSETA," *arXiv preprint arXiv:1710.05961*, Oct. 2017.

[55] H. Mansour and X. Jiang, "A robust online subspace estimation and tracking algorithm," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Brisbane, Australia, Apr. 2015, pp. 4065–4069.

[56] J. H. Chow and K. W. Cheung, "A toolbox for power system dynamics and control engineering education and research," *IEEE Trans. Power Syst.*, vol. 7, no. 4, pp. 1559–1564, Nov. 1992.

[57] J. Zhang and A. D. Domínguez-García, "Augmenting the power system toolbox: Enabling automatic generation control and providing a platform for cyber security analysis," in *Proc. of North Amer. Power Symp. (NAPS)*, Denver, CO, Sep. 2016, pp. 1–5.

[58] M. Grady, "EE394J-2, Power System Engineering II, Spring 2012," 2012, (Accessed on Oct. 12, 2017). [Online]. Available: http://users.ece.utexas.edu/~grady/EE394J_2_Spring12.html

[59] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[60] M. Razaviyayn, M. Sanjabi, and Z.-Q. Luo, "A stochastic successive minimization method for nonsmooth nonconvex optimization with applications to transceiver design in wireless communication networks," *Math. Program., Ser. B*, vol. 157, no. 2, pp. 515–545, May 2016.