Online Robust Subspace Clustering for Analyzing Incomplete Synchrophasor Measurements

Young-hwan Lee and Seung-Jun Kim

Dept. of Computer Science and Electrical Engineering
University of Maryland, Baltimore County, Baltimore, MD 21250
E-mail: {lee43,sjkim}@umbc.edu

***Abstract*—Phasor measurement units (PMUs) are instrumental for grid monitoring thanks to high sampling rates and precise synchronization. However, collecting and analyzing the PMU data are challenging due to the large volume, computational burden, and missing measurements. In this work, a robust subspace clustering model is adopted to perform reconstruction of missing measurements while simultaneously detecting outliers, which may be due to grid abnormalities or cyber-attacks. To mitigate the computational burden and processing delay of batch computation, an online algorithm is derived based on stochastic approximation. The storage cost of the signal templates (dictionary) used for representing the data is reduced via sparsification and pruning procedures. Numerical tests using simulated PMU data validate the performance of the proposed approach.**

***Keywords*—synchrophasor, subspace clustering, low rank representation, online algorithms**

I. Introduction

Accurate and timely monitoring of the power grid is critical for reliable and efficient grid operation. Anomalies in the grid states must be identified immediately in order to prevent equipment damages and catastrophic cascade failures. It is critical that cyber-attacks to the power system data are detected and their effect mitigated promptly. As volatile renewable energy resources are increasingly integrated, the importance of real-time monitoring is greater than ever.

Phasor measurement units (PMUs) are being increasingly deployed for power grid monitoring. A PMU is a device that measures voltage and current phasors at a high sampling rate. The PMU measurements can be finely synchronized across a wide area using the global positioning system (GPS). Thus, the synchrophasor measurements can reveal the dynamics of the grid, which is instrumental for grid protection and control.

There are various challenges associated with processing the PMU data. Portions of the large-volume measurements may be missing due to sensor malfunction or network congestion. The data may become corrupted from erroneous communication or cyber-attacks. Missing PMU measurements were reconstructed using low-rank matrix completion approaches in [1], [2], [3]. Measurements deviating from the postulated models were detected as outliers, capitalizing on the fact that such deviations are rare [3], [4]. However, these works were based on batch implementations, where the processing takes place offline after all necessary measurements have been collected, incurring high computational burden and memory requirements. As the batch algorithms yield significant processing delay, they may be unsuitable for real-time analysis and monitoring.

This work was supported in part by NSF grant 1547347.

In order to mitigate these drawbacks, online algorithms are desired, which process the incoming data stream sequentially and incrementally. An online robust principal component analysis (PCA) algorithm based on stochastic optimization was proposed in [5]. An online algorithm for the low-rank representation (LRR) model was developed in [6]. However, it was suggested in [6] 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 this paper, an online algorithm for the LRR-based robust subspace clustering model is developed for streaming data, potentially with missing and corrupt measurements. The dictionary is updated in an online fashion as well. The dictionary atoms are selected judiciously based on an appropriate sparsification criterion under a strict constraint on the dictionary size. Therefore, the proposed algorithm enjoys low computational complexity, processing delay, and storage overhead, as well as the ability to track slow variations in the underlying distribution of the data.

The rest of the paper is organized as follows. In Section II, a batch robust subspace clustering formulation with missing data is first presented. After that, the online algorithm is derived in Section III. Strategies to update the dictionary are discussed in Section IV. Results from numerical tests are presented in Section V. Conclusions are offered in Section VI.

II. Low Rank Representation for Incomplete Data

Let us denote the PMU measurements obtained from N buses in a power grid 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 LRR model is a union-of-subspaces model, where a (noise-free) measurement vector $\bar{\mathbf{z}}_t$ is assumed to lie 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 are well approximated by a linear subspace, and significant variations or changes in the operating point would lead to different subspaces [7], [2].

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 $\mathbf{D} \in \mathbb{R}^{N\times M}$. 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 union-of-subspaces structure. More specifically, consider the optimization problem

$$\min_{\mathbf{C}} \|\mathbf{C}\|_* \text{ subject to } \bar{\mathbf{Z}} = \mathbf{D}\mathbf{C} \tag{1}$$

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 $\mathbf{C}$ [8]. If $\mathbf{D} = \bar{\mathbf{Z}}$ and $\bar{\mathbf{Z}}$ has the skinny SVD given by $\bar{\mathbf{Z}} = \mathbf{U}_0\mathbf{\Sigma}_0\mathbf{V}_0^\top$, it can be shown that the solution of (1) is $\mathbf{C} = \mathbf{V}_0\mathbf{V}_0^\top$ [9]. Under the assumption that the data $\{\mathbf{z}_t\}$ are clean, that is, 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 $\mathbf{V}_0\mathbf{V}_0^\top$ can serve as the *affinity matrix*, whose (i, j)-th entries are nonzero only if $\bar{\mathbf{z}}_i$ and $\bar{\mathbf{z}}_j$ belong to the same subspace [10]. Various subspace clustering techniques rely on the affinity matrix [11]. In fact, if $\mathbf{D}$ contains atoms that span the individual subspaces $\{\mathcal{S}_k\}$, then the resulting $\mathbf{C}$ matrix can still be used as the affinity matrix [9].

As the PMU network collects data from a large-scale grid, it is often the case that some PMU measurements are not relayed to 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* entries, 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 PMU measurements may become corrupted by gross errors in the communication channel or even by cyber-attacks. 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). If the optimal $\mathbf{E}$ for (4) is denoted as $\overline{\mathbf{E}}$, then the optimal $\mathbf{E}$ for (3) will be given by $\mathbf{E} = \mathcal{P}_\Omega(\overline{\mathbf{E}})$.

III. Online Algorithm

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

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

$$\min_{\mathbf{C},\mathbf{E}} \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Z}) - \mathbf{D}\mathbf{C} - \mathbf{E}\|_F^2 + \lambda\|\mathbf{C}\|_* + \mu\|\mathcal{P}_\Omega(\mathbf{E})\|_1 \tag{5}$$

where $\lambda > 0$ is a parameter tuning the rank of $\mathbf{C}$. Inspired by the fact that the nuclear norm of a matrix $\mathbf{C}$, whose rank is no larger than R, can be expressed as [8]

$$\|\mathbf{C}\|_* = \min_{\substack{\mathbf{A}\in\mathbb{R}^{M\times R} \\ \mathbf{B}\in\mathbb{R}^{T\times R}}} \frac{1}{2}(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2) \text{ s.t. } \mathbf{C} = \mathbf{A}\mathbf{B}^\top \tag{6}$$

formulation (5) is modified to

$$\min_{\mathbf{A},\mathbf{B},\mathbf{E}} \left[\frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Z}) - \mathbf{D}\mathbf{A}\mathbf{B}^\top - \mathbf{E}\|_F^2 + \frac{\lambda}{2}(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2) + \mu\|\mathcal{P}_\Omega(\mathbf{E})\|_1\right]. \tag{7}$$

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, $\Omega_t := \{n : (n, t) \in \Omega\}$. Thanks to reformulation (7), the problem is now separable into different time slots and can be rewritten as

$$\min_{\mathbf{A},\{\mathbf{b}_t\},\{\mathbf{e}_t\}} \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{2}\|\mathcal{P}_{\Omega_t}(\mathbf{z}_t) - \mathbf{D}\mathbf{A}\mathbf{b}_t - \mathbf{e}_t\|_2^2 + \frac{\lambda}{2}\|\mathbf{b}_t\|_2^2 + \mu\|\mathcal{P}_{\Omega_t}(\mathbf{e}_t)\|_1\right) + \frac{\lambda}{2T}\|\mathbf{A}\|_F^2. \tag{8}$$

The main 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.

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

$$\{\mathbf{b}_t, \mathbf{e}_t\} = \arg\min_{\mathbf{b},\mathbf{e}} \left[\frac{1}{2}\|\mathcal{P}_{\Omega_t}(\mathbf{z}_t) - \mathbf{D}\mathbf{A}_{t-1}\mathbf{b} - \mathbf{e}\|_2^2 + \frac{\lambda}{2}\|\mathbf{b}\|_2^2 + \mu\|\mathcal{P}_{\Omega_t}(\mathbf{e})\|_1\right]. \tag{9}$$

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, with $S_\mu(x) := \text{sgn}(x)\max\{|x|-\mu, 0\}$, the coordinate descent proceeds for $l = 0, 1, 2, \cdots$ as

$$\mathbf{b}_t^{l+1} = (\mathbf{A}_{t-1}^\top \mathbf{D}^\top \mathbf{D}\mathbf{A}_{t-1} + \lambda \mathbf{I})^{-1} \mathbf{A}_{t-1}^\top \mathbf{D}^\top (\mathcal{P}_{\Omega_t}(\mathbf{z}_t) - \mathbf{e}_t^l) \tag{10}$$

$$\mathbf{e}_t^{l+1}|_{\Omega_t} = \mathcal{S}_\mu \left(\mathcal{P}_{\Omega_t}(\mathbf{z}_t) - \mathbf{D}\mathbf{A}_{t-1}\mathbf{b}_t^{l+1} \right) \big|_{\Omega_t} \tag{11}$$

$$\mathbf{e}_t^{l+1}|_{\Omega_t^c} = (-\mathbf{D}\mathbf{A}_{t-1}\mathbf{b}_t^{l+1})|_{\Omega_t^c} \tag{12}$$

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 (8) approaches the expected value, thanks to the law of large numbers. 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

$$\mathbf{A}_t = \mathbf{A}_{t-1} + \rho \left[\mathbf{D}^\top \left(\mathcal{P}_{\Omega_t}(\mathbf{z}_t) - \mathbf{D}\mathbf{A}_{t-1}\mathbf{b}_t - \mathbf{e}_t \right) \mathbf{b}_t^\top - \frac{\lambda}{T}\mathbf{A}_{t-1} \right] \tag{13}$$

where $\rho > 0$ is the step size.

IV. Dictionary Update

Clearly, in order for the union-of-subspaces structure to be captured, $\mathbf{D}$ must contain the vectors that can span the individual subspaces $\{\mathcal{S}_k\}$. If this is true, and if also the subspaces are independent and the data samples are strictly drawn from the union of subspaces, it can be shown that $\mathbf{C}$ from (1) would have its (m,t)-entry equal to zero only if the m-th atom in $\mathbf{D}$ and the t-th data sample $\mathbf{z}_t$ lie in different subspaces [9]. If the size of the batch of data samples $\mathbf{Z}$ is large enough, then a reasonable and trivial choice for $\mathbf{D}$ is to simply set $\mathbf{D} = \mathbf{Z}$. This is in fact the choice made in [6]. However, such a choice would not allow real-time implementation for streaming data. Thus, it is desirable to start with a small initial batch of dictionary, and update the dictionary continuously based on the incoming pool of data.

A critical issue in this context is to be selective in adding new atoms into the dictionary so that the size of the dictionary does not grow excessively, without sacrificing the diversity of the dictionary so that a good representation performance is achieved. For this, online sparsification and pruning procedures are employed.

A. Online Sparsification

Various online sparsification strategies have been developed, in particular, in the context of kernel-based adaptive filtering [12], [13], [14]. The main idea is to accept a new datum into the dictionary only when it is deemed to sufficiently contribute to the diversity based on some appropriate metric. Representative sparsification criteria include minimum pairwise distance, approximate linear dependence, coherence, and Babel measure, all of which can be shown to upper-bound the condition number of the kernel (Gram) matrix [14]. In this work, the minimum pairwise distance measure is adopted and tested due to its simplicity.

TABLE I
Online algorithm.

Input: Ω, $\mathcal{P}_\Omega(\mathbf{Z})$, $\lambda > 0$, $\mu > 0$, $\rho > 0$, M_0, M, δ^2
Output: $\mathbf{D}$, $\mathbf{A}$, $\{\mathbf{b}_t\}$, and $\{\mathbf{e}_t\}$

Initialize $\mathbf{D}_0 \in \mathbb{R}^{N \times M_0}$ and $\mathbf{A}_0$
For $t = 1, 2, \ldots, T$
  Set $l = 0$ and $\mathbf{e}_t^0 = \mathbf{0}$
  Repeat
    Update $\mathbf{b}_t^{l+1}$ by (10) with $\mathbf{D}$ replaced by $\mathbf{D}_{t-1}$
    Update $\mathbf{e}_t^{l+1}$ by (11)–(12) with $\mathbf{D}$ replaced by $\mathbf{D}_{t-1}$
    $l \leftarrow l + 1$
  Until convergence
  Set $\mathbf{b}_t = \mathbf{b}_t^l$ and $\mathbf{e}_t = \mathbf{e}_t^l$
  Update $\mathbf{A}_t$ by (13) with $\mathbf{D}$ replaced by $\mathbf{D}_{t-1}$
  Set $\mathbf{e}_t|_{\Omega_t^c} = \mathbf{0}$
  / Sparsification */*
  If $\Omega_t^c = \varnothing$, set $\hat{\mathbf{z}}_t = \mathbf{z}_t$
  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)
  If $\kappa_t \geq \delta^2$
    $\mathbf{D}_t = [\mathbf{D}_{t-1}, \hat{\mathbf{z}}_t]$
    $M_t = M_{t-1} + 1$
  Otherwise $\mathbf{D}_t = \mathbf{D}_{t-1}$ and $M_t = M_{t-1}$
  / Pruning */*
  If $M_t > M$
    Find $m^* := \arg\min_{m=1,2,\ldots,M} \|\mathbf{a}_t(m,:)\|_2^2$
    Remove the m^*-th column of $\mathbf{D}_t$ and m^*-th row of $\mathbf{A}_t$
    $M_t = M_t - 1$
  End if
Next t
Return $\mathbf{D} = \mathbf{D}_T$, $\mathbf{A} = \mathbf{A}_T$, $\{\mathbf{b}_t\}$, and $\{\mathbf{e}_t\}$

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 $\mathbf{D}_{t-1} = [\check{\mathbf{z}}_1, \check{\mathbf{z}}_2, \ldots, \check{\mathbf{z}}_{M_{t-1}}]$, where $\check{\mathbf{z}}_m \in \{\mathbf{z}_1, \ldots, \mathbf{z}_{t-1}\}$. For a datum $\mathbf{z}_t$ arriving at time t, the distance metric is computed as

$$\kappa_t := \min_{m=1,\ldots,M_{t-1}} \mathbf{z}_t^\top \mathbf{z}_t - \frac{(\mathbf{z}_t^\top \check{\mathbf{z}}_m)^2}{\check{\mathbf{z}}_m^\top \check{\mathbf{z}}_m}. \tag{14}$$

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 pragmatic strategy may be to use the reconstructed vector $\hat{\mathbf{z}}_t = \mathbf{D}_{t-1}\mathbf{A}_{t-1}\mathbf{b}_t$ in place of $\mathbf{z}_t$.

B. Pruning

When M_t exceeds a prespecified memory budget M, 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 [15]. Thus, upon denoting the m-th row of $\mathbf{A}_t$ as $\mathbf{a}_t(m,:)$,

$$m^* = \arg \min_{m=1,2,\ldots,M} \|\mathbf{a}_t(m,:)\|_2^2 \tag{15}$$

is found, and $\check{\mathbf{z}}_{m^*}$ is removed from $\mathbf{D}_t$. An alternative would be to look at $\{\mathbf{c}_t := \mathbf{A}_t\mathbf{b}_t\}$, which is not pursued here. The overall algorithm is listed in Table I.

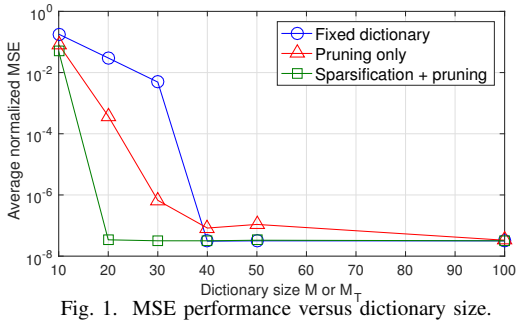

Fig. 1. MSE performance versus dictionary size.

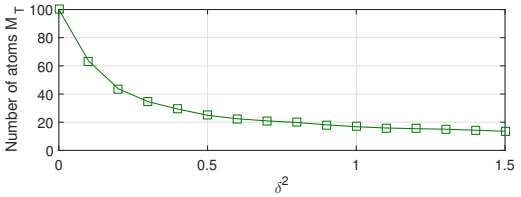

Fig. 2. Dictionary size versus sparsification threshold.

V. Numerical Tests

A. Tests Using Synthetic Data

Synthetic data vectors $\{\bar{\mathbf{z}}_t\}$ were generated by randomly sampling from 5 subspaces $\{\mathcal{S}_1, \ldots, \mathcal{S}_5\}$. The dimensions of the subspaces were 1, 1, 2, 3, and 3, and the ambient dimension N was equal to 20. The data vectors were normalized to have $\mathbb{E}\{\bar{\mathbf{z}}_t\} = \mathbf{0}$ and $\mathbb{E}\{\|\bar{\mathbf{z}}_t\|_2^2\} = 1$. A total of $T = 1000$ samples were generated to form data matrix $\bar{\mathbf{Z}} \in \mathbb{R}^{20\times 1000}$. It was assumed that a fraction 5% of the entries of $\bar{\mathbf{Z}}$ were missing. As a performance metric, the average normalized mean-square error (MSE), defined by $\|\bar{\mathbf{Z}} - \hat{\bar{\mathbf{Z}}}\|_F^2/\|\bar{\mathbf{Z}}\|_F^2$, was used. The averages were obtained from 20 independent trials.

Fig. 1 depicts the average normalized MSE of the proposed online algorithm. The curve with the circle markers represents the MSE performance as the dictionary size M is varied from 10 to 100. The dictionary is constructed by randomly sampling M columns from $\bar{\mathbf{Z}}$ and fixed throughout. It can be seen that the MSE performance is suboptimal when $M \leq 30$ since the randomly chosen columns in the dictionary might not contain all the samples necessary to span the 5 subspaces. The curve with the triangle markers corresponds to the case where the dictionary is initially constructed with only $M_0 = 5$ random data vectors, but subsequently updated through the pruning procedure with the maximum budget set to M. The value of δ^2 was set to zero, which means that no sparsification is performed; that is, all incoming data vectors were admitted to the dictionary. In this case, the MSE performance is better than the fixed dictionary case when $M \leq 30$ as the dictionary is continuously updated and thus the chance of the dictionary to span all 5 subspaces becomes significantly higher. However, as the size of the dictionary is increased, the performance advantage of pruning over a fixed dictionary is eventually lost due to the burden of continually updating matrix $\mathbf{A}_t$.

The merit of both extremes is obtained by performing sparsification with pruning, which is illustrated by the curve with square markers in Fig. 1. Note that the actual dictionary size M_T at time $t = T$ may be smaller than M, depending on δ^2, as shown in Fig. 2. Thus, the best trade-off between the MSE and effective dictionary size can be traced by adjusting δ^2.

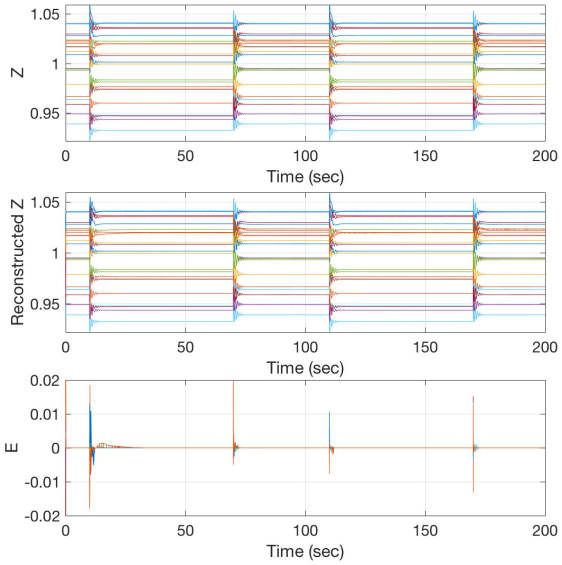

Fig. 3. Simulated PMU data.

B. Tests with Simulated PMU Data

The algorithm was also tested using simulated PMU data from the PSS/E simulator [16]. The simulated power system consists of 23 buses and 6 generators. (For detail, see [3].) Only the voltage magnitudes are used for simplicity to construct $\mathbf{Z}$. The top panel in Fig. 3 depicts the PMU measurements taken at a sampling rate of 40 samples/sec. A line trip was simulated at 10 sec, which caused oscillations across the system. The line was closed at 70 sec. The two events were replicated once more at 110 sec and 170 sec, respectively. As data preprocessing, the nominal per-unit quantity 1 was subtracted from each entry of $\mathbf{Z}$.

The middle panel in Fig. 3 shows the reconstruction from the data with 5% missing entries. Parameter λ was set to 10^{-3}, $\mu = 3 \times 10^{-4}$, and $\rho = 10^4$. Sparsification and pruning were done with $\delta^2 = 2\times 10^{-5}$ and $M = 200$, resulting in $M_T = 48$. It can be seen that the reconstruction is accurate. One can actually see some transient effects of the online algorithm in the beginning, which die out quite fast. The bottom panel of Fig. 3 depicts the outlier matrix $\mathbf{E}$. It is observed that the important events are detected as outliers.

VI. Conclusion

An online robust subspace clustering algorithm was derived based on the LRR model to reconstruct missing PMU measurements and to detect abnormalities in the grid. The online update of the signal dictionary was accomplished by sparsification and pruning procedures. It was observed that the best trade-off between performance and memory/computational burden can be traced by the proposed procedures. Tests with simulated PMU data verified that the algorithm can reconstruct in an online fashion the missing measurements, while detecting salient events as outliers.

REFERENCES

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

[2] 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. of the 2015 IEEE Global Conference on Signal and Information Processing*, Orlando, FL, Dec. 2015, pp. 785–789.

[3] S.-J. Kim, Y. Lee, and K. Y. Lee, "Robust subspace approaches for analyzing incomplete synchrophasor measurements," in *Proc. of the 9th IFAC Symposium on Control of Power and Energy Systems*, New Delhi, India, Dec. 2015, pp. 120–125.

[4] 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. of the 2015 48th Hawaii International Conference on System Sciences*, Kauai, HI, Jan. 2015, pp. 2637–2644.

[5] J. Feng, H. Xu, and S. Yan, "Online robust PCA via stochastic optimization," in *Proc. of the Advances in Neural Information Processing Systems*, vol. 26, Dec. 2013, pp. 404–412.

[6] L. Li, B. Zou, and X. Zhang, "Online learning for low-rank representation and its application in subspace clustering," *Journal of Computational Information Systems*, vol. 10, pp. 7125–7135, Aug. 2014.

[7] J. M. Lim and C. L. DeMarco, "Model-free voltage stability assessments via singular value analysis of PMU data," in *Proc. of the IREP Symposium-Bulk Power System Dynamics and Control IX (IREP)*, Rethymnon, Greece, Aug. 2013, pp. 1–10.

[8] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, pp. 471–501, Aug. 2010.

[9] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 171–184, Jan. 2013.

[10] J. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *Int'l. J. Computer Vision*, vol. 29, no. 3, pp. 159–179, 1998.

[11] R. Vidal, "Subspace clustering," *IEEE Sig. Proc. Mag.*, vol. 28, no. 2, pp. 52–68, 2011.

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

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

[14] P. Honeine, "Analyzing sparse dictionaries for online learning with kernels," *IEEE Transactions on Signal Proccessing*, vol. 63, pp. 6343–6353, Dec. 2015.

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

[16] Siemens, *PSS/E Documentation, Version 33.5*, 2013.