Interpolatory Approximations of PMU Data: Dimension Reduction and Pilot Selection

Sean Reiter, Mark Embree, Serkan Gugercin, Vassilis Kekatos, *Senior Member, IEEE*

***Abstract*—This work investigates the reduction of phasor measurement unit (PMU) data through low-rank matrix approximations. To reconstruct a PMU data matrix from fewer measurements, we propose the framework of interpolatory matrix decompositions (IDs). In contrast to methods relying on principal component analysis or singular value decomposition, IDs recover the complete data matrix using only a few of its rows (PMU datastreams) and/or a few of its columns (snapshots in time). This row-/column-based compression enables real-time monitoring of power transmission systems using measurements from a smaller subset of pilot datastreams, thereby minimizing communication bandwidth. The ID perspective gives a rigorous error bound on the quality of the data compression. We propose selecting the pilot measurements used in an ID via the discrete empirical interpolation method (DEIM), a greedy algorithm that aims to control the error bound. This bound yields a computable estimate of the reconstruction error during online operations. A violation of this estimate suggests a change in the system's operating conditions and thus serves as a tool for fault detection. Following a disturbance, DEIM can be used to localize the event source across all buses with high accuracy. Numerical tests on synthetic PMU data demonstrate DEIM's excellent performance in data compression and validate the proposed DEIM-based fault-detection and localization method.**

***Index Terms*—Low-rank, matrix decomposition, event monitoring, pilot bus, discrete empirical interpolation method (DEIM).**

I. Introduction

Data-driven methods for real-time power system monitoring have garnered significant attention due to the adoption of phasor measurement units (PMUs) in wide-area monitoring systems (WAMS). PMUs are *in situ* sensor devices that provide global positioning system (GPS)-synchronized phasor readings of nodal voltages, nodal currents, line currents, and their time derivatives, at a rate of 60–120 samples per second. These real-time streaming measurements can provide an accurate view of the system's operating condition, enabling operators to monitor network performance, detect disturbances (such as line trips or outages), and initiate corrective measures. However, data accumulation presents a significant roadblock to real-time operational benefits: e.g., a network of 100 PMUs, each with a sampling rate of 120 Hz, generates 200 gigabytes of data per day [1], [2]. Moreover, a large communication bandwidth is required to transmit PMU data to control centers [3]–[5]. Thus, reliably managing and reducing the scale of streaming PMU data becomes an essential research area.

PMU time series data can be organized in a matrix form: each row corresponds to a single PMU measurement stream, e.g., voltages at a particular bus, and each column contains a snapshot in time. It is well-documented in theory and industry practice that matrices of PMU data exhibit an approximate low-rank structure under both normal and abnormal conditions. This low-rank phenomenon has been attributed to the spatial-temporal correlations in PMU datastreams; see, e.g., [5]–[8]. Data-driven methods exploiting such dependencies in PMU data have been successfully applied to grid monitoring tasks, including detection and localization of disturbance events. Low-rank representations of PMU data are typically computed using methods that decompose the data into orthogonal components, e.g., the singular value decomposition (SVD) [9] or the closely related principal component analysis (PCA) [10]. Event detection algorithms based on PCA are proposed in [11]–[13]. Recognizing the underlying low dimension of PMU data, the authors in [5] propose to monitor a network using fewer pilot PMUs. In [8], low-dimensional subspaces derived from the SVD and subspace comparison metrics are used to identify, detect, and localize events in real time. The papers [14], [15] develop matrix completion-based methods for event detection that use the SVD [14] and arrange PMU data into low-rank Hankel matrices [15]. The paper [16] seeks to express a PMU matrix as a sum of a low-rank matrix, a noise matrix, and a row-sparse matrix that captures abnormal network behavior; this structure is leveraged to detect events. See [7] for a survey of low-rank methods in WAMS.

PCA/SVD provide *optimal* low-rank approximations to a complete PMU matrix by blending information from *all* rows and columns of the PMU matrix, and thus require large amounts of PMU data to be communicated across the network before compression can be applied at a central hub, such as a phasor data concentrator (PDC). Thus, PCA/SVD can be ill-suited for time-sensitive and bandwidth-limited tasks. Moreover, applications such as post-event analysis do not explicitly seek an optimal reconstruction of the PMU matrix, but rather aim to reveal a small subset of rows and columns that capture its low-dimensional structure and correspond to points of interest in the grid's operating history.

As an alternative, we explore *interpolatory matrix decompo-*

This work was supported by US National Science Foundation grants AMPS-1923221, AMPS-2318800, and EPCN-2500682.

Sean Reiter is with the Courant Institute of Mathematical Sciences, New York University, New York, NY USA 10012 (email: s.reiter@nyu.edu).

Mark Embree and Serkan Gugercin are with the Department of Mathematics, Virginia Tech, Blacksburg, VA USA 24061 (email: embree@vt.edu, gugercin@vt.edu).

Vassilis Kekatos is with the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN USA 47906 (email: kekatos@purdue.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier TPWRS-01877-2025.

sitions (IDs) [17]–[21] for the real-time dimension reduction (compression) of PMU data to enable fast and reliable methods for WAMS. In contrast to PCA/SVD, which use all available measurements, IDs aim to reconstruct the complete PMU data matrix using measurements collected from only a small number of PMU datastreams, which we refer to as *pilot streams* [5], and/or from downsampled time snapshots [3], [4], [22], which we call *pilot snapshots*. While suboptimal in approximation quality, an interpolatory decomposition is more suitable for *real-time* applications in WAMS. For instance, a reduced set of pilot streams can be monitored during real-time operations and used to approximately represent any non-pilot datastream via an ID, thereby significantly reducing the dimension of streaming PMU data.

Pilot selection is the task of selecting a reduced number of pilots offline to monitor during online operations for the purpose of minimizing communication bandwidth. The quality of the online dimension reduction achieved via an interpolatory approximation hinges on the choice of pilots, which are identified during an offline training phase. We propose using the *discrete empirical interpolation method* (DEIM) [18], [23]–[26] for adaptively performing this pilot selection. DEIM is a greedy algorithm that aims to minimize a computable upper bound on the interpolatory approximation error. This bound can certify whether a given collection of pilot streams or snapshots truly captures the low-rank character of the data, and can be leveraged for operational uses, such as online error estimation and disturbance event detection.

Combining the framework of IDs with DEIM, we propose an offline–online framework for real-time event monitoring using a reduced number of pilot datastreams. Offline, DEIM parses ambient data from all PMUs to select a minimal set of pilot streams until the interpolatory error bound meets a user-specified tolerance. Online, only data from these pilots are communicated to the control room. PMU data from the non-pilot buses are readily reconstructed from the pilot datastreams via an ID. Our work especially builds on the fundamental contribution of [5], whose authors pursue a similar goal of dimension reduction for real-time event detection. We observe that their method is actually a form of ID, and thus the error bound (14) applies to it. Our approach differs in its general framework based on IDs and in the way it selects pilots. By using DEIM, we seek to control the error bound, so much so that (14) serves as a viable error estimator during online operations, and provides a lightweight tripwire for detecting disturbances.

Contributions. The key contributions of this work are:

- Introducing IDs as a unified framework for compressing PMU data from fewer rows and/or columns, enabling the use of a rigorous error bound for certifying IDs for PMUs;
- Proposing the use of the discrete empirical interpolation method (DEIM) for selecting pilot streams and snapshots;
- Adapting the interpolatory error bound into an indicator for event detection during online monitoring;
- Applying DEIM to post-event data to locate faults.

Organization. We review the basics of IDs in the context of PMU data reduction in Section II, and describe how these low-rank methods can be used for the real-time reconstruction of streaming PMU data. Section III introduces DEIM for selecting the pilots that determine the ID. Building upon [5], in Section IV we describe an ID-DEIM framework for data-driven monitoring of power systems using a reduced set of pilot streams. Numerical tests using synthetically generated PMU data illustrate and validate the proposed methodology.

Notation. Bold lowercase and uppercase letters $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$ denote vectors and matrices. We use MATLAB notation to index: the (i,j)-th entry of $\mathbf{X}$ is denoted $\mathbf{X}(i,j) \in \mathbb{R}$; the i-th row of $\mathbf{X}$ is $\mathbf{X}(i,:) \in \mathbb{R}^{1\times n_2}$; the j-th column of $\mathbf{X}$ is $\mathbf{X}(:,j) \in \mathbb{R}^{n_1}$ and occasionally $\mathbf{x}_j \in \mathbb{R}^{n_1}$. Given a set of indices $\mathcal{K} = \{k_1, \ldots, k_m\}$, let $\mathbf{X}(:,\mathcal{K}) \in \mathbb{R}^{n_1 \times m}$ and $\mathbf{X}(\mathcal{K},:) \in \mathbb{R}^{m \times n_2}$ denote the columns and rows of $\mathbf{X}$ indexed by $\mathcal{K}$, and $m = |\mathcal{K}|$ denote the cardinality of $\mathcal{K}$. With $\mathbf{I}_n \in \mathbb{R}^{n\times n}$, $\mathbf{e}_i \in \mathbb{R}^n$, and $\mathbf{1}_n \in \mathbb{R}^n$ we denote the $n \times n$ identity, the i-th canonical unit vector ($\mathbf{e}_i$ is one in entry i and zero elsewhere), and the vector of all ones. We use $\|\cdot\|_2$ and $\|\cdot\|_{\mathrm{F}}$ to denote the spectral and Frobenius norms of a matrix, and $\cdot^{\mathsf{T}}$ to denote the vector/matrix transpose.

II. Low-Rank Approximation of PMU Matrices

Suppose a system operator collects data at T discrete time instances from N PMU datastreams. To simplify the exposition, assume each measured bus is instrumented with a single PMU. We assume that each row of a matrix $\mathbf{Y} \in \mathbb{R}^{N\times T}$ containing PMU time series data corresponds to a single measured quantity, e.g., each row contains voltage magnitudes from a particular bus. Multiple measured quantities are handled by organizing the data into distinct matrices and processing each matrix separately. We envision that $\mathbf{Y}$ is typically wide ($T > N$) due to the high sampling rate of PMUs, but this is not required; our discussion applies to $\mathbf{Y}$ of arbitrary dimension.

The underlying dimension of PMU data has been considered from a variety of perspectives; see, e.g. [3]–[8], [15]. Matrices of PMU data typically exhibit an underlying low-rank structure, regardless of whether the data are collected during ambient or irregular operating conditions. As a result, the underlying dimension (rank) of $\mathbf{Y}$ can be reduced by retaining only its dominant components computed via PCA [10] or the SVD; see Section 2.4 of [9]. The low-rank factors can be stored more efficiently, and subsequent computations and analyses on the reconstructed data can be expedited.

A. Approximations of PMU Data via the SVD/PCA

We briefly review how one can obtain a low-rank matrix approximating $\mathbf{Y}$ by truncating the trailing components of its SVD. Define $R = \operatorname{rank}(\mathbf{Y}) \le \min\{N, T\}$. The SVD of $\mathbf{Y}$ is

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{T}} = \sum_{k=1}^{R} \sigma_k \mathbf{u}_k \mathbf{v}_k^{\mathsf{T}} \tag{1}$$

where the diagonal matrix $\mathbf{\Sigma} \in \mathbb{R}^{R\times R}$ carries the singular values $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_R > 0$, and matrices $\mathbf{U} \in \mathbb{R}^{N\times R}$ and $\mathbf{V} \in \mathbb{R}^{T\times R}$ have orthonormal columns $\mathbf{u}_1, \ldots, \mathbf{u}_R$ and $\mathbf{v}_1, \ldots, \mathbf{v}_R$ called the left and right singular vectors.

Because real-world PMU data are corrupted by noise, $\mathbf{Y}$ typically has full rank $R = \min(N, T)$. Nonetheless, it can be

approximated well by a matrix of lower rank $K \ll R$ obtained from the leading terms in the SVD. By the Eckart–Young–Mirsky theorem (see, e.g., Theorem. 2.4.8 of [9]), an optimal rank $K < R$ approximation to $\mathbf{Y}$ in the spectral and Frobenius norms is

$$\mathbf{Y}_K := \mathbf{U}_K \mathbf{\Sigma}_K \mathbf{V}_K^\mathsf{T} = \sum_{k=1}^{K} \sigma_k \mathbf{u}_k \mathbf{v}_k^\mathsf{T} \tag{2}$$

obtained by summing the leading rank-one components $\sigma_k \mathbf{u}_k \mathbf{v}_k^\mathsf{T}$ corresponding to the largest K singular values. Here $\mathbf{U}_K \in \mathbb{R}^{N\times K}$, $\mathbf{\Sigma}_K \in \mathbb{R}^{K\times K}$, and $\mathbf{V}_K \in \mathbb{R}^{T\times K}$ are the submatrices of $\mathbf{U}, \mathbf{\Sigma}$, and $\mathbf{V}$ associated with those leading K singular values. The matrix $\mathbf{Y}_K$ solves

$$\mathbf{Y}_K = \underset{\mathbf{Z}\in\mathbb{R}^{N\times T}}{\arg\min} \quad \|\mathbf{Y} - \mathbf{Z}\| \quad \text{subj. to} \quad \operatorname{rank}(\mathbf{Z}) \le K$$

where $\|\cdot\|$ can be the spectral or the Frobenius matrix norm. This minimizer attains the approximation errors

$$\|\mathbf{Y} - \mathbf{Y}_K\|_2 = \sigma_{K+1} \quad \text{and} \quad \|\mathbf{Y} - \mathbf{Y}_K\|_\mathrm{F}^2 = \sum_{k=K+1}^{R} \sigma_k^2. \tag{3}$$

Evidently, $\mathbf{Y}_K$ approximates $\mathbf{Y}$ well if the $R-K$ trailing singular values are sufficiently small. In practice, the rank K is selected to deliver a relative approximation error below a certain threshold $0 < \alpha < 1$; for example, pick K so that

$$\frac{\|\mathbf{Y} - \mathbf{Y}_K\|_\mathrm{F}}{\|\mathbf{Y}\|_\mathrm{F}} = \frac{\left(\sum_{k=K+1}^{R} \sigma_k^2\right)^{1/2}}{\left(\sum_{k=1}^{R} \sigma_k^2\right)^{1/2}} \le \alpha. \tag{4}$$

This compression via the SVD is akin to keeping the K principal components of a matrix, as practiced in [5].[1]

As noted in the introduction, the SVD blends information from *all* PMU datastreams at *all* times. Because PMUs have limited capacity for handling data, measurements from every datastream must first be transmitted across dedicated synchrophasor communication links before the SVD can be applied to reduce the dimension of the data. Thus, the SVD is not typically feasible for time-sensitive applications involving large-scale systems and is better suited for offline tasks, such as post-event analysis.

B. Interpolatory Approximations of PMU Data

As an alternative to PCA/SVD for PMU data, we propose using *interpolatory matrix decompositions* (IDs) [17]–[21]. These low-rank factorizations, sketched in Figure 1, use only a few rows and/or columns of the matrix. For $K \le N$, an ID of $\mathbf{Y} \in \mathbb{R}^{N\times T}$ is a low-rank factorization of the form

$$\mathbf{Y} \approx \mathbf{Y}_\mathcal{S}^\mathcal{T} := \mathbf{C}^\mathcal{T} \mathbf{X}_\mathcal{S}^\mathcal{T} \mathbf{R}_\mathcal{S} \tag{5}$$

where $\mathbf{X}_\mathcal{S}^\mathcal{T} \in \mathbb{R}^{K\times K}$. The matrices $\mathbf{C}^\mathcal{T} := \mathbf{Y}(:, \mathcal{T}) \in \mathbb{R}^{N\times K}$ and $\mathbf{R}_\mathcal{S} := \mathbf{Y}(\mathcal{S}, :) \in \mathbb{R}^{K\times T}$ contain a subset of the columns and rows of $\mathbf{Y}$ indexed by $\mathcal{T} = \{t_1, t_2, \ldots, t_K\} \subset \{1, 2, \ldots, T\}$ and $\mathcal{S} = \{s_1, s_2, \ldots, s_K\} \subset \{1, 2, \ldots, N\}$. We refer to the PMU datastreams and time snapshots corresponding to the indices $\mathcal{S}$ and $\mathcal{T}$ as *pilot streams* and *pilot snapshots*, respectively, or simply *pilots* when referring to both. The ID in (5) aims to recover the complete matrix $\mathbf{Y}$ using only information contained in the selected columns $\mathbf{C}^\mathcal{T}$ and rows $\mathbf{R}_\mathcal{S}$ of $\mathbf{Y}$; the matrix $\mathbf{X}_\mathcal{S}^\mathcal{T}$ of smaller dimension $K \times K$ is chosen to make $\mathbf{Y}_\mathcal{S}^\mathcal{T}$ a good approximation of $\mathbf{Y}$.

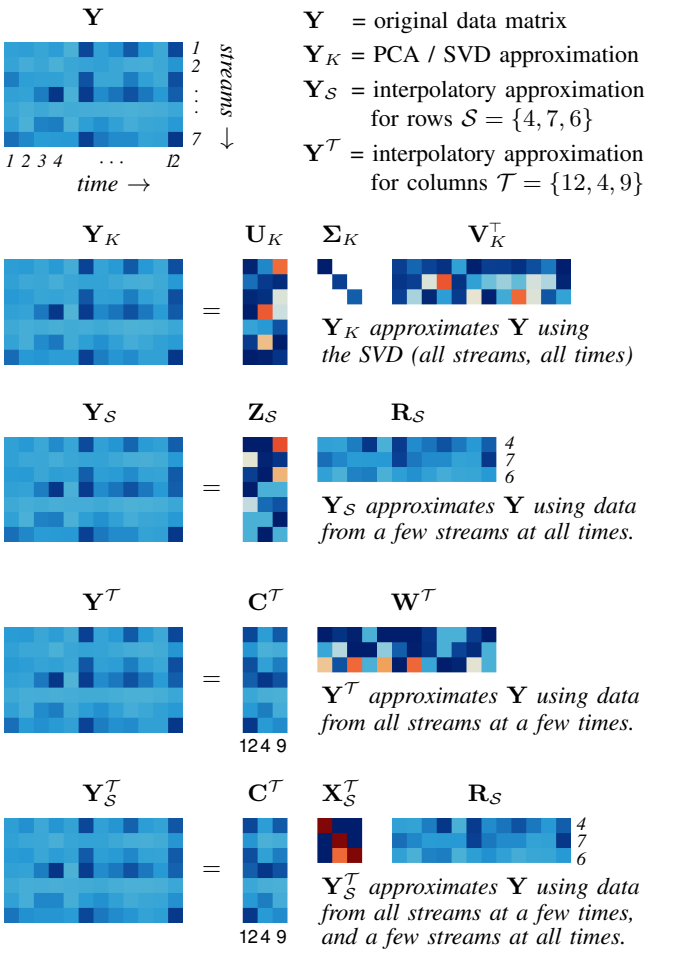


Figure 1: Sketch of four low-rank approximations ($\mathbf{Y}_K$, $\mathbf{Y}_\mathcal{S}$, $\mathbf{Y}^\mathcal{T}$, and $\mathbf{Y}_\mathcal{S}^\mathcal{T}$) to the PMU data matrix $\mathbf{Y}$. The bottom three are IDs. (See Figure 2 of [26] for a similar graphic illustrating the DEIM process.)

The approximation quality hinges on which rows and columns are selected for $\mathbf{R}_\mathcal{S}$ and $\mathbf{C}^\mathcal{T}$, and the choice for $\mathbf{X}_\mathcal{S}^\mathcal{T}$. In Section II-C, we describe a strategy for computing $\mathbf{X}_\mathcal{S}^\mathcal{T}$ via a least-squares fit of the data, once $\mathcal{S}$ and $\mathcal{T}$ are set. The numerical linear algebra literature has explored various strategies for selecting rows $\mathcal{S}$ and columns $\mathcal{T}$; see, e.g., [17]–[20]. In Section III, we propose a greedy strategy from [23], [24] to select rows and columns iteratively.

The form (5) is the most general; one may also consider IDs that only use the rows *or* columns of $\mathbf{Y}$. Utilizing only certain rows of $\mathbf{Y}$ amounts to using data collected from $K \le N$ pilot datastreams contained in $\mathbf{R}_\mathcal{S}$ to approximate the data from all datastreams, i.e., finding a matrix $\mathbf{Z}_\mathcal{S} \in \mathbb{R}^{N\times K}$ such that

$$\mathbf{Y} \approx \mathbf{Y}_\mathcal{S} := \mathbf{Z}_\mathcal{S} \mathbf{R}_\mathcal{S} \in \mathbb{R}^{N\times T}. \tag{6}$$

Note that in (6), columns of $\mathbf{Y}$ are not factored into the

[1] Strictly speaking, the data would first be prepared for PCA by subtracting the mean of each row from every entry in that row, replacing $\mathbf{Y}$ with $\mathbf{Y} - \boldsymbol{\mu}\mathbf{1}_T^\mathsf{T}$, where $\mu_j = (y_{j,1} + \cdots + y_{j,T})/T$. We do not apply any such preprocessing of $\mathbf{Y}$, and thus take PCA to be synonymous with the SVD.

approximation. The i-th row of $\mathbf{Z}_{\mathcal{S}}$ contains the weights that specify how the data collected from the K pilot streams in $\mathbf{R}_{\mathcal{S}}$ should be combined to approximate the data $\mathbf{Y}(i,:)$ from the i-th *non-pilot* datastream, $i \notin \mathcal{S}$:

$$\mathbf{Y}(i,:) \approx (\mathbf{Z}_{\mathcal{S}}\mathbf{R}_{\mathcal{S}})(i,:) = \sum_{k=1}^{K} \mathbf{Z}_{\mathcal{S}}(i,k)\mathbf{R}_{\mathcal{S}}(k,:). \quad (7)$$

Using only certain columns of $\mathbf{Y}$ amounts to using the $K \leq T$ pilot snapshots contained in $\mathbf{C}^{\mathcal{T}}$ to recover the full time series, i.e., finding a matrix $\mathbf{W}^{\mathcal{T}} \in \mathbb{R}^{K\times T}$ such that

$$\mathbf{Y} \approx \mathbf{Y}^{\mathcal{T}} \coloneqq \mathbf{C}^{\mathcal{T}}\mathbf{W}^{\mathcal{T}} \in \mathbb{R}^{N\times T}. \quad (8)$$

Now the j-th column of $\mathbf{W}^{\mathcal{T}}$ contains the weights that describe how the selected pilot snapshots in $\mathbf{C}^{\mathcal{T}}$ should be combined to produce an approximation of the data $\mathbf{Y}(:,j)$ at time t_j for $j \notin \mathcal{T}$, i.e.,

$$\mathbf{Y}(:,j) \approx \left(\mathbf{C}^{\mathcal{T}}\mathbf{W}^{\mathcal{T}}\right)(:,j) = \sum_{k=1}^{K} \mathbf{W}^{\mathcal{T}}(j,k)\mathbf{C}^{\mathcal{T}}(:,k). \quad (9)$$

Figure 1 compares different ID regimes to the SVD. The storage requirements for $\mathbf{C}^{\mathcal{T}}$, $\mathbf{X}_{\mathcal{S}}^{\mathcal{T}}$, and $\mathbf{R}_{\mathcal{S}}$ are similar to that of the SVD. Like the optimal approximation $\mathbf{Y}_K$ from the SVD, the interpolatory approximations $\mathbf{Y}_{\mathcal{S}}^{\mathcal{T}}$, $\mathbf{Y}_{\mathcal{S}}$, and $\mathbf{Y}^{\mathcal{T}}$ have rank K (or less). Unlike the orthogonal components $\mathbf{U}_K$ and $\mathbf{V}_K$ in the SVD, the low-rank factors $\mathbf{R}_{\mathcal{S}}$ and $\mathbf{C}^{\mathcal{T}}$ contain actual PMU data. These factors preserve qualitative features of the data; e.g., sparsity, or a particular voltage pattern if the data was collected following a disturbance [16].

Another virtue of the interpolatory approximations over $\mathbf{Y}_K$ is economy: $\mathbf{Y}_K$ is a blend of *all* $N \gg K$ rows and $T \gg K$ columns of $\mathbf{Y}$, so computing the SVD requires information from *all* PMU time series simultaneously. On the other hand, interpolatory approximations $\mathbf{Y}_{\mathcal{S}}$ and $\mathbf{Y}^{\mathcal{T}}$ only use $K = |\mathcal{S}| = |\mathcal{T}|$ rows and columns of $\mathbf{Y}$. Once the weights $\mathbf{Z}_{\mathcal{S}}$ and $\mathbf{W}^{\mathcal{T}}$ have been computed, a low-rank approximation to $\mathbf{Y}$ can be obtained while interacting *only* with $K \leq N$ pilot streams or $K \leq T$ down-sampled pilot snapshots. As observed in, e.g., [3]–[5], this reduction significantly lowers the bandwidth needs of synchrophasor communication networks. We point out that the sparse approximations of PMU data proposed in [3]–[5], [22] can be interpreted at the matrix level as one-sided interpolatory approximations (6) and (8).

C. Analyzing the Interpolatory Approximation Error

Because $\mathbf{Y}_K$ is the optimal rank-K approximation to $\mathbf{Y}$, the interpolatory approximation $\mathbf{Y}_{\mathcal{S}}^{\mathcal{T}}$ cannot be any better:

$$\sigma_{K+1} = \|\mathbf{Y} - \mathbf{Y}_K\|_2 \leq \|\mathbf{Y} - \mathbf{Y}_{\mathcal{S}}^{\mathcal{T}}\|_2. \quad (10)$$

The same holds for $\mathbf{Y}_{\mathcal{S}}$ and $\mathbf{Y}^{\mathcal{T}}$. We analyze here the quality of the approximation $\mathbf{Y}_{\mathcal{S}}^{\mathcal{T}}$ relative to the best approximation from PCA/SVD. For the moment, we assume the K row and column indices in $\mathcal{S}$ and $\mathcal{T}$ are given. The error $\|\mathbf{Y} - \mathbf{Y}_{\mathcal{S}}^{\mathcal{T}}\|_2$ depends on how we compute $\mathbf{X}_{\mathcal{S}}^{\mathcal{T}}$. One natural choice [17], [21] is

$$\mathbf{X}_{\mathcal{S}}^{\mathcal{T}} = \left(\mathbf{C}^{\mathcal{T}}\right)^{\dagger}\mathbf{Y}\left(\mathbf{R}_{\mathcal{S}}\right)^{\dagger}$$

where $\cdot^{\dagger}$ denotes the *Moore–Penrose pseudoinverse*; see, e.g., Section 5.5.2 of [9]. Assuming the rows of $\mathbf{R}_{\mathcal{S}}$ and the columns of $\mathbf{C}^{\mathcal{T}}$ are linearly independent, we have $(\mathbf{R}_{\mathcal{S}})^{\dagger} = \mathbf{R}_{\mathcal{S}}^{\mathsf{T}}\left(\mathbf{R}_{\mathcal{S}}\mathbf{R}_{\mathcal{S}}^{\mathsf{T}}\right)^{-1}$ and $\left(\mathbf{C}^{\mathcal{T}}\right)^{\dagger} = \left(\mathbf{C}^{\mathcal{T}^{\mathsf{T}}}\mathbf{C}^{\mathcal{T}}\right)^{-1}\mathbf{C}^{\mathcal{T}^{\mathsf{T}}}$. Thus, $\mathbf{Y}_{\mathcal{S}}^{\mathcal{T}}$ is given by

$$\mathbf{Y}_{\mathcal{S}}^{\mathcal{T}} = \mathbf{C}^{\mathcal{T}}\mathbf{X}_{\mathcal{S}}^{\mathcal{T}}\mathbf{R}_{\mathcal{S}} = \mathbf{C}^{\mathcal{T}}\left(\mathbf{C}^{\mathcal{T}}\right)^{\dagger}\mathbf{Y}\left(\mathbf{R}_{\mathcal{S}}\right)^{\dagger}\mathbf{R}_{\mathcal{S}}.$$

For the one-sided interpolatory approximations $\mathbf{Y}_{\mathcal{S}}$ and $\mathbf{Y}^{\mathcal{T}}$, the same idea can be applied to obtain $\mathbf{Z}_{\mathcal{S}}$ and $\mathbf{W}^{\mathcal{T}}$:

$$\mathbf{Z}_{\mathcal{S}} = \mathbf{Y}\left(\mathbf{R}_{\mathcal{S}}\right)^{\dagger} \quad \text{and} \quad \mathbf{W}^{\mathcal{T}} = \left(\mathbf{C}^{\mathcal{T}}\right)^{\dagger}\mathbf{Y}.$$

This construction highlights why IDs can effectively compress the entire PMU data set in $\mathbf{Y}$: the matrices $\mathbf{Z}_{\mathcal{S}}$ and $\mathbf{W}^{\mathcal{T}}$ are least squares fits of the data [21], e.g.,

$$\mathbf{Z}_{\mathcal{S}} = \operatorname*{arg\,min}_{\mathbf{Z}\in\mathbb{R}^{N\times K}} \|\mathbf{Y} - \mathbf{Z}\mathbf{R}_{\mathcal{S}}\|_{\mathrm{F}}. \quad (11)$$

Assuming the rows (columns) of $\mathbf{Y}$ are linearly independent, the solutions $\mathbf{Z}_{\mathcal{S}}$ ($\mathbf{W}^{\mathcal{T}}$) to (11) are unique. The pilot-based reconstruction from [5] in fact uses this choice of $\mathbf{Z}_{\mathcal{S}}$.

The quality of the IDs $\mathbf{Y}_{\mathcal{S}}^{\mathcal{T}}$, $\mathbf{Y}_{\mathcal{S}}$, and $\mathbf{Y}^{\mathcal{T}}$ can be assessed in a more quantitative way. Define $\mathbf{S} \coloneqq \mathbf{I}_N(:,\mathcal{S}) \in \mathbb{R}^{N\times K}$ and $\mathbf{T} \coloneqq \mathbf{I}_T(:,\mathcal{T}) \in \mathbb{R}^{T\times K}$ to be the matrices containing the K columns of the $N\times N$ and $T\times T$ identity matrices indexed by $\mathcal{S}$ and $\mathcal{T}$. From Theorem 4.1 of [18], we have that

$$\sigma_{K+1} \;\leq\; \|\mathbf{Y} - \mathbf{Y}_{\mathcal{S}}^{\mathcal{T}}\|_2 \;\leq\; (\eta_{\mathcal{S}} + \eta_{\mathcal{T}})\,\sigma_{K+1} \quad (12)$$

where error factors $\eta_{\mathcal{S}}, \eta_{\mathcal{T}} \geq 1$ are given by

$$\eta_{\mathcal{S}} \coloneqq \|\left(\mathbf{S}^{\mathsf{T}}\mathbf{U}_K\right)^{-1}\|_2 \quad \text{and} \quad \eta_{\mathcal{T}} \coloneqq \|\left(\mathbf{T}^{\mathsf{T}}\mathbf{V}_K\right)^{-1}\|_2. \quad (13)$$

Recall that $\mathbf{U}_K \in \mathbb{R}^{N\times K}$ and $\mathbf{V}_K \in \mathbb{R}^{T\times K}$ contain the leading K left and right singular vectors of $\mathbf{Y}$. The submatrices $\mathbf{S}^{\mathsf{T}}\mathbf{U}_K$ and $\mathbf{T}^{\mathsf{T}}\mathbf{V}_K$ are guaranteed to be nonsingular for certain row and column selection schemes, including the one we propose in Section III; see Lemma 3.2 of [18]. For the one-sided interpolatory approximations $\mathbf{Y}_{\mathcal{S}}$ and $\mathbf{Y}^{\mathcal{T}}$, we have the simplified bounds:

$$\begin{aligned} \sigma_{K+1} \;&\leq\; \|\mathbf{Y} - \mathbf{Y}_{\mathcal{S}}\|_2 \;\leq\; \eta_{\mathcal{S}}\,\sigma_{K+1} \\ \sigma_{K+1} \;&\leq\; \|\mathbf{Y} - \mathbf{Y}^{\mathcal{T}}\|_2 \;\leq\; \eta_{\mathcal{T}}\,\sigma_{K+1}. \end{aligned} \quad (14)$$

For details, see Lemma 4.2 of [18] and Theorem 1.5 of [27].

Let us unpack the error factors $\eta_{\mathcal{S}}$ and $\eta_{\mathcal{T}}$. The matrix $\mathbf{S}^{\mathsf{T}}\mathbf{U}_K$ is a $K\times K$ submatrix of $\mathbf{U}_K$. Note that $\mathbf{U}_k$ has orthonormal columns; $\eta_{\mathcal{S}}$ measures how far from orthonormal the *rows* of $\mathbf{U}_K$ corresponding to $\mathcal{S}$ are. Likewise, $\eta_{\mathcal{T}}$ measures how far from orthonormal the $\mathcal{T}$ rows of $\mathbf{V}_K$ are. The order of the indices in $\mathcal{S}$ and $\mathcal{T}$ does not affect $\eta_{\mathcal{S}}$ and $\eta_{\mathcal{T}}$.

The interpolatory error bounds in (12) and (14) hold for *any* collection of pilot indices $\mathcal{S}$ or $\mathcal{T}$. We can leverage these bounds to realize real-time computational benefits and performance guarantees. Although these ideas also apply to the two-sided formulation in (5), we present them for the one-sided approximations (6) and (8) for simplicity.

1. *Fast error monitoring*. Because $\mathbf{S}^{\mathsf{T}}\mathbf{U}_K$ and $\mathbf{T}^{\mathsf{T}}\mathbf{V}_K$ are small $K\times K$ matrices, the error indicators $\eta_{\mathcal{S}}$ and $\eta_{\mathcal{T}}$ will be much quicker to compute than the full approximation errors $\|\mathbf{Y} - \mathbf{Y}_{\mathcal{S}}\|_2$ or $\|\mathbf{Y} - \mathbf{Y}^{\mathcal{T}}\|_2$ for large N and T.

One does not even need to explicitly form the low-rank approximations $\mathbf{Y}_{\mathcal{S}}$ or $\mathbf{Y}^{\mathcal{T}}$ in (6) and (8), and hence $\mathbf{Z}_{\mathcal{S}}$ or $\mathbf{W}^{\mathcal{T}}$, to evaluate $\eta_{\mathcal{S}}$ and $\eta_{\mathcal{T}}$. This observation allows for fast *a priori* estimation of the interpolatory approximation error during online operations, or enables the error factors to be monitored as the pilots are selected.

2. *Pilot certification.* In an operational setting, one can use any desired strategy for picking pilots $\mathcal{S}$ and $\mathcal{T}$. The error factors $\eta_{\mathcal{S}}$ or $\eta_{\mathcal{T}}$ can then be (quickly) computed to certify if the chosen pilots capture the rank-K nature of the PMU data matrix. If the error bound is below a threshold, e.g., $\eta_{\mathcal{S}}\sigma_{K+1}, \eta_{\mathcal{T}}\sigma_{K+1} \leq \tau = 10^{-1}$, the selection $\mathcal{S}$ or $\mathcal{T}$ is accepted; otherwise, either replace some pilots, or increase K and add additional pilots.

We describe at length how each of the above ideas can be implemented in a practical operational scenario in Section IV.

One could consider selecting pilots $\mathcal{S}$ and $\mathcal{T}$ to explicitly minimize $\eta_{\mathcal{S}}$ and $\eta_{\mathcal{T}}$ over all possible configurations; however, such a minimization would involve combinatorial complexity. Instead, in Section III we advocate for a more efficient *greedy* algorithm that seeks to control the growth of the error factors $\eta_{\mathcal{S}}$ and $\eta_{\mathcal{T}}$ as new pilots are selected, one at a time.

III. Greedy Pilot Selection

We propose using the *discrete empirical interpolation method* (DEIM) index selection algorithm [18], [23]–[26] to select the pilot subsets $\mathcal{S}$ and $\mathcal{T}$. DEIM is a discrete variant of the empirical interpolation method [24]. Initially developed for resolving the "lifting bottleneck" in the model reduction of nonlinear dynamical systems [23] by constructing interpolatory approximations to vector-valued nonlinear functions, DEIM was applied to construct IDs in [18]. The DEIM procedure (independently) selects the row and column indices $\mathcal{S}$ and $\mathcal{T}$ by iteratively parsing the leading left and right singular vectors stored in $\mathbf{U}_K$ and $\mathbf{V}_K$ for a matrix $\mathbf{Y}$. At each iteration, DEIM attempts to adaptively minimize the growth of the error factors in (13) as each new index is added to $\mathcal{S}$ or $\mathcal{T}$, and, in practice, the DEIM indices typically yield small error factors. In the numerical tests of Section IV, DEIM selects pilot configurations that produce error factors $\eta_{\mathcal{S}}$ and $\eta_{\mathcal{T}}$ of size $\mathcal{O}(10^1)$ or less, whereas other, seemingly reliable, selection approaches produce factors of size $\mathcal{O}(10^4)$. Thus, in conjunction with the approximation error (12), we expect DEIM to provide an effective pilot selection strategy.

A. The Discrete Empirical Interpolation Method

We describe how DEIM operates on $\mathbf{U}_K$ to select $K \leq N$ pilot streams $\mathcal{S}$ from a data matrix $\mathbf{Y}$. The same process applied (independently) to $\mathbf{V}_K$ selects the pilot snapshots $\mathcal{T}$.

Our derivation uses special matrices called *interpolatory projectors*. Let $\mathbf{S}_k = \begin{bmatrix}\mathbf{e}_{s_1} & \cdots & \mathbf{e}_{s_k}\end{bmatrix} \in \mathbb{R}^{N\times k}$ denote the k columns of $\mathbf{I}_N$ specified by the distinct indices in $\mathcal{S}_k = \{s_1, \ldots, s_k\} \subset \{1, \ldots, N\}$, and let $\mathbf{U}_k = \mathbf{U}(:, 1:k) = \begin{bmatrix}\mathbf{u}_1 & \cdots & \mathbf{u}_k\end{bmatrix} \in \mathbb{R}^{N\times k}$ denote the leading k columns of the matrix $\mathbf{U} \in \mathbb{R}^{N\times R}$ of the left singular vectors of $\mathbf{Y}$. The *interpolatory projector* for $\mathcal{S}_k$ onto $\operatorname{span}(\mathbf{U}_k)$ is defined as

$$\mathbf{P}_k \coloneqq \mathbf{U}_k \left(\mathbf{S}_k^{\mathsf{T}}\mathbf{U}_k\right)^{-1} \mathbf{S}_k^{\mathsf{T}} \in \mathbb{R}^{N\times N}. \tag{15}$$

The matrix $\mathbf{S}_k^{\mathsf{T}}\mathbf{U}_k \in \mathbb{R}^{k\times k}$ is guaranteed to be invertible for indices $\mathcal{S}_k$ adaptively selected by DEIM (see Lemma 3.2 of [18]). One can readily verify that $\mathbf{P}_k$ satisfies the projector property: $\mathbf{P}_k^2 = \mathbf{P}_k$. More critically, $\mathbf{P}_k$ is an *interpolatory* projector in this sense: for any vector $\mathbf{x} \in \mathbb{R}^N$, the projected vector $\widehat{\mathbf{x}} \coloneqq \mathbf{P}_k\mathbf{x}$ exactly matches $\mathbf{x}$ in the pilot indices $\mathcal{S}_k$, i.e.,

$$\widehat{\mathbf{x}}(\mathcal{S}_k) = \mathbf{S}_k^{\mathsf{T}}\left(\mathbf{U}_k(\mathbf{S}_k^{\mathsf{T}}\mathbf{U}_k)^{-1}\mathbf{S}_k^{\mathsf{T}}\right)\mathbf{x} = \mathbf{x}(\mathcal{S}_k). \tag{16}$$

DEIM operates on the columns of $\mathbf{u}_k$ one at a time to select each new pilot index. Start with the selection of the first pilot s_1, corresponding to $k = 1$. In this simple case, the error factor $\eta_{\mathcal{S}_1}$ in (13) reduces to

$$\eta_{\mathcal{S}_1} = \left\|\left(\mathbf{S}_1^{\mathsf{T}}\mathbf{U}_1\right)^{-1}\right\|_2 = \frac{1}{|\mathbf{u}_1(s_1)|}$$

the reciprocal of the magnitude of the s_1 entry of the leading singular vector $\mathbf{u}_1$. Therefore, to minimize η_{s_1}, choose the datastream index $s_1 \in \{1, \ldots, N\}$ corresponding to the entry in $\mathbf{u}_1 \in \mathbb{R}^N$ having the largest magnitude.

- **Step 1.** Choose s_1 as the index corresponding to the entry of $\mathbf{u}_1$ with the largest magnitude:

$$s_1 = \underset{1\leq s\leq N}{\arg\max}\,|\mathbf{u}_1(s)|, \quad \mathbf{u}_1 = \begin{bmatrix}\times\\ \times\\ \times\\ \times\end{bmatrix} \leftarrow s_1 \quad .$$

  Construct $\mathbf{P}_1 \coloneqq \mathbf{u}_1\mathbf{e}_{s_1}^{\mathsf{T}}/\mathbf{u}_1(s_1)$, the interpolatory projector (15) for $\mathcal{S}_1 = \{s_1\}$ onto the span of $\mathbf{u}_1$.

The choice of the second index s_2 is more subtle. We should avoid choosing the same pilot ($s_2 = s_1$), which would result in an infinite error factor $\eta_{\mathcal{S}} = \|\left(\mathbf{S}_2^{\mathsf{T}}\mathbf{U}_2\right)^{-1}\|_2$. Using the intuition that $\eta_{\mathcal{S}} = \|\left(\mathbf{S}_k^{\mathsf{T}}\mathbf{U}_k\right)^{-1}\|$ is small when the rows of $\mathbf{U}_k$ selected by $\mathbf{S}_k$ are quite distinct, we choose s_2 so that the two rows $\mathbf{U}_2(\mathcal{S}_2, :)$ are as *independent as possible* for $\mathcal{S}_2 = \{s_1, s_2\}$. To guarantee that $s_2 \neq s_1$, i.e., that we select a distinct datastream, we remove a multiple of $\mathbf{u}_1$ from $\mathbf{u}_2$ to zero out the s_1 entry:

$$\mathbf{r}_2 \coloneqq \mathbf{u}_2 - \frac{\mathbf{u}_2(s_1)}{\mathbf{u}_1(s_1)}\mathbf{u}_1 = \mathbf{u}_2 - \mathbf{P}_1\mathbf{u}_2$$

giving $\mathbf{r}_2(s_1) = 0$ by the interpolatory property of $\mathbf{P}_1$ in index s_1. We can then select s_2 to be the index of the largest-magnitude entry of $\mathbf{r}_2$. Using this formulation, we summarize the next step of DEIM as follows.

- **Step 2.** Compute the residual of the interpolatory projection of $\mathbf{u}_2$ onto $\operatorname{span}(\mathbf{u}_1)$:

$$\mathbf{r}_2 = \mathbf{u}_2 - \mathbf{P}_1\mathbf{u}_2.$$

  Choose s_2 as the largest-magnitude entry of $\mathbf{r}_2$:

$$s_2 = \underset{1\leq n\leq N}{\arg\max}\,|\mathbf{r}_2(n)|, \quad \mathbf{r}_2 = \mathbf{u}_2 - \mathbf{P}_1\mathbf{u}_2 = \begin{bmatrix}\star\\ 0\\ \star\\ \star\end{bmatrix} \leftarrow s_2$$

  (The $\star$ indicates a modified entry from the $k = 1$ step.)

Subsequent steps, $k = 3, \ldots, K$, follow this same template.

Algorithm III.1: The discrete empirical interpolation method (DEIM) [23], [24].

Input: Matrix with orthonormal columns $\mathbf{U}_K = \begin{bmatrix}\mathbf{u}_1 & \cdots & \mathbf{u}_K\end{bmatrix} \in \mathbb{R}^{N\times K}$, $1 \le K < N$.
Output: Indices $\mathcal{S} = \{s_1, \ldots, s_K\} \subset \{1, \ldots, N\}$.
1 Choose the first index $s_1 = \arg\max_{1\le s\le N} |\mathbf{u}_1(s)|$.
2 Take $\mathcal{S}_1 = \{s_1\}$.
3 **for** $k = 2, \ldots, K$ **do**
4 Compute the residual by solving a k-dimensional linear system:

$$\mathbf{r}_k = \mathbf{u}_k - \mathbf{U}_{k-1}\left(\mathbf{S}_{k-1}^{\mathsf{T}}\mathbf{U}_{k-1}\right)^{-1}\mathbf{S}_{k-1}^{\mathsf{T}}\mathbf{u}_k.$$

5 Choose $s_k = \arg\max_{1\le s\le N} |\mathbf{r}_k(s)|$.
6 Take $\mathcal{S}_k = \mathcal{S}_{k-1} \cup \{s_k\}$.
7 **end**

- **Step k.** Construct the interpolatory projector $\mathbf{P}_{k-1}$ for datastreams $s_1, \ldots, s_{k-1}$ onto the span of $\mathbf{u}_1, \ldots, \mathbf{u}_{k-1}$ according to (15), and compute the residual

$$\mathbf{r}_k = \mathbf{u}_k - \mathbf{P}_{k-1}\mathbf{u}_k$$

such that $\mathbf{r}_k(s_1) = \cdots = \mathbf{r}_k(s_{k-1}) = 0$. Choose s_k to be the index of the largest-magnitude entry of $\mathbf{r}_k$:

$$s_k = \arg\max_{1\le s\le N} |\mathbf{r}_k(s)|.$$

We are assured that s_k is a new datastream that differs from $s_1, \ldots, s_{k-1}$, since $\mathbf{r}_k(s_1) = \cdots = \mathbf{r}_k(s_{k-1}) = 0$ but $\mathbf{r}_k \neq \mathbf{0}$ (otherwise, $\mathbf{u}_k \in \operatorname{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_{k-1}\}$, a contradiction).

Algorithm III.1 summarizes this procedure. (To select pilot snapshots $\mathcal{T}$, i.e., columns of $\mathbf{Y}$, simply apply this algorithm to the right singular vectors $\mathbf{V}_K$.) An efficient implementation avoids explicitly forming the interpolatory projectors $\mathbf{P}_k$, which are large, dense matrices. Rather, step 4 of Algorithm III.1 computes the *action* of $\mathbf{P}_k$ on the singular vector $\mathbf{u}_k$ without explicitly forming the inverse of $\mathbf{S}_{k-1}^{\mathsf{T}}\mathbf{U}_{k-1}$. Instead, a $(k-1)\times(k-1)$ linear system of equations is solved at every step. These systems can be solved efficiently by leveraging the nested structure of the coefficient matrix $\mathbf{S}_{k-1}^{\mathsf{T}}\mathbf{U}_{k-1}$ to compute its LU decomposition. In such an implementation, Algorithm III.1 can be carried out in $\mathcal{O}(NK^2) + \mathcal{O}(K^3)$ floating point operations (FLOPs). We refer to Section 2.1.2 of [25] for a more detailed complexity analysis. Typically, $K \ll N$ in practice. Given the requisite singular value and vector data (also required for PCA-based methods), the cost of DEIM is linear in N (or analogously T, when used to select column indices). Thus, DEIM can be realistically applied for pilot selection in large-scale settings.

At every iteration, the DEIM selection is designed to roughly minimize the incremental growth of the error factor $\eta_{\mathcal{S}}$ in the bound (14); see Lemma 3.2 in [23] for a proof. This explains why DEIM is an effective choice for computing the pilot sets $\mathcal{S}$ and $\mathcal{T}$, as illustrated in Section 6 of [18]. The DEIM algorithm is directly linked to the LU factorization with partial pivoting; see Section 3 of [18]. One alternative to the DEIM index selection algorithm is the QDEIM variant [25], which identifies the pilots $\mathcal{S}$ by applying a rank-revealing QR factorization to the rows of $\mathbf{U}_K$. The cost of the factorization is $2NK^2 - \frac{2}{3}K^3$ FLOPs (to leading order); see, e.g., Section 5.4.3 of [9]. Thus, the cost of QDEIM is similar to that of DEIM. The ultimate set of pilots $\mathcal{S}$ chosen by QDEIM is invariant under permutations of the columns of $\mathbf{U}_K$, although in practice, DEIM and QDEIM perform similarly. We emphasize that QDEIM is not iterative; the number of desired pilots must be specified in advance.

B. Numerical Tests

The code and data for reproducing the numerical tests presented in this manuscript are available at [28]. All numerical tests were performed on a MacBook Air with 8 gigabytes of RAM and an Apple M2 processor running macOS Sequoia version 15.2 with MATLAB 23.2.0.2515942 (R2023b) Update 7. We now demonstrate the ability of the row- and column-based IDs (6) and (8) to reduce the dimension of PMU data matrices using different strategies for selecting the pilots $\mathcal{S}$ and $\mathcal{T}$. We compare the following selection strategies.

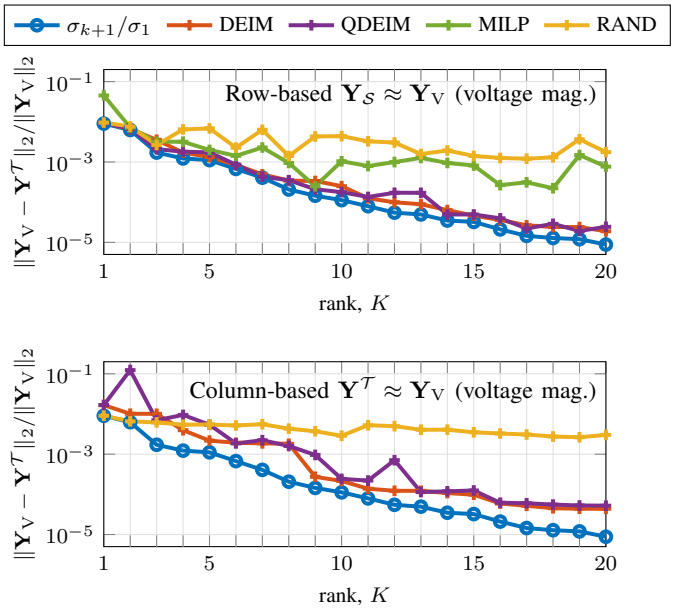
DEIM is Algorithm III.1 from [23], [24].
QDEIM is the QDEIM variant of DEIM from [25].
MILP selects the indices by solving a mixed-integer linear program (MILP) that minimizes the maximum absolute pairwise cosine similarity among the selected rows/columns, thereby favoring subsets whose vectors are nearly pairwise orthogonal. This approach is modeled after the pilot PMU selection strategy proposed in Section II of [5].[2] The program is solved using MATLAB's `intlinprog` command. The particular formulation of the MILP and the associated objective function are provided in the Appendix.
RAND is a random selection (MATLAB's `randi` command).

These strategies are applied to matrices $\mathbf{Y}$ of PMU data to identify the sets $\mathcal{S}$ and $\mathcal{T}$ with K indices. Then, matrices $\mathbf{Z}_{\mathcal{S}}$ and $\mathbf{W}^{\mathcal{T}}$ are computed from (11), and used to form rank-K IDs $\mathbf{Y}_{\mathcal{S}}$ and $\mathbf{Y}^{\mathcal{T}}$ of the matrix $\mathbf{Y}$ according to (6) and (8).

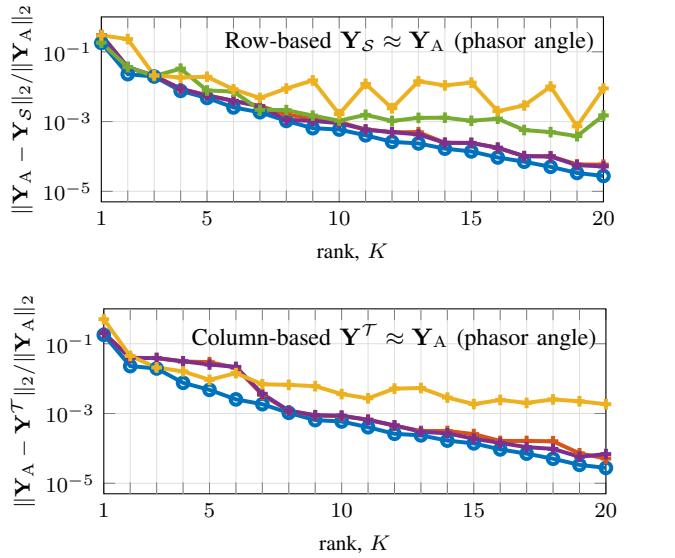
To assess the quality of the reduction, we compute the relative errors induced by $\mathbf{Y}_{\mathcal{S}}$ and $\mathbf{Y}^{\mathcal{T}}$ in the matrix 2-norm, and compare this against the relative best rank-k approximation error σ_{k+1}/σ_1 from the SVD. We also compute the associated error factors $\eta_{\mathcal{S}}$ and $\eta_{\mathcal{T}}$ for each selection strategy, although we do not report the values of these factors here; they are available in the accompanying code package [28].

We test the efficacy of our IDs on synthetic PMU data generated from transient simulations of the NETSNYPS 68-bus, 16-machine test system [29]. We perform the simulations with MATLAB's Power Systems Toolbox (PST) [30]. This setup is similar to that used in [5] and [8]. These data correspond to dynamic voltage waveforms and ignore the internal processing mechanism of a PMU, which is manufacturer-dependent. To mimic realistic operating conditions, following

[2]We note that [5] does not explicitly formulate the pilot selection strategy as an MILP; rather, the guiding principle is to choose pilot PMUs such that the cosine similarity among the selected datastreams is close to zero, i.e., the corresponding datastreams are as nearly orthogonal as possible. The specific implementation used to carry out this selection is not detailed therein.



(a) Relative errors for $K = 1, 2, \ldots, 20$ interpolatory matrix approximations $\mathbf{Y}_\mathcal{S}$ and $\mathbf{Y}^\mathcal{T}$ of the voltage magnitude data $\mathbf{Y}_\mathrm{V}$.



(b) Relative errors for $K = 1, 2, \ldots, 20$ interpolatory matrix approximations $\mathbf{Y}_\mathcal{S}$ and $\mathbf{Y}^\mathcal{T}$ of the phasor angle data $\mathbf{Y}_\mathrm{A}$.

Figure 2: Relative errors for rank $K = 1, 2 \ldots, 20$ interpolatory matrix approximations $\mathbf{Y}_\mathcal{S}$ and $\mathbf{Y}^\mathcal{T}$ of the matrices $\mathbf{Y}_\mathrm{V}, \mathbf{Y}_\mathrm{A} \in \mathbb{R}^{68\times 6000}$ containing 60 s worth of voltage magnitude and phasor angle data generated using the 68-bus, 16-machine NETSNYPS test system. The size of the column-based data prevents the use of MILP in the column-based approximation.

the setup of [5], zero-mean Gaussian noise is added to all of our synthetically generated PMU data, so that the signal-to-noise ratio (SNR) is 92 dB. This noise complies with the accuracy limit of less than 1% total vector error (TVE) specified by IEEE Standard C37.118.1 [31]. The components of a realistic PMU measurement chain can exhibit significantly different behavior during a fault, compared to steady-state operating conditions. These uncertainties are not reflected in our synthetic data, which are obtained from simulations. Real PMU data may also be corrupted by colored noise and suffer from outliers, whereas our methodology assumes white noise and outlier-free data. Validation of our proposed methodology on data obtained from a PMU emulator or real-world PMU data is a topic that we plan to consider in future work.

We present results on voltage magnitude and phasor angle data at every bus collected at a sampling rate of 100 Hz over a 60 s window. These data are organized into a pair of 68×6000 dimensional matrices $\mathbf{Y}_\mathrm{V}$ and $\mathbf{Y}_\mathrm{A}$ for voltage magnitudes and phasor angles. After 30 s of the simulation, a three-phase line fault is applied between buses 28 and 29 and cleared 0.2 s later. For the column-based IDs (8), we do not employ the MILP-based selection because the matrices required for solving the program do not fit in RAM.

Figure 2 shows the relative errors in the 2-norm. We compute rank $K = 1, 2, \ldots, 20$ row- and column-based IDs for the PMU data matrices $\mathbf{Y}_\mathrm{V}$ and $\mathbf{Y}_\mathrm{A}$ using the selection strategies outlined above and report the relative errors in Figures 2a and 2b. For both types of measurement data, the DEIM- and QDEIM-based IDs give results on par with those of the SVD for each rank K. For the voltage magnitude data, the MILP-based IDs perform poorly after an initial reduction for small K; the approximation errors oscillate as K increases. For $K \leq 10$, the MILP-based IDs produce approximations of the phasor angle data that are of similar quality to the DEIM- and QDEIM-based IDs. For $K > 10$, the row index selection by MILP did not converge as `intlinprog` terminated early after exploring 1 000 000 branch-and-bound nodes. Hence, the approximation error plateaus for these values of K. In all cases, the row-based IDs perform better than the column-based ones, as expected, since the column-based approximations have more indices to choose from (6000 vs. 68). These results demonstrate that, when combined with a reliable pilot selection strategy, IDs are an effective tool for reducing the dimension of various types of PMU data.

In the interest of space, we do not report the wall-clock times for the different pilot selection strategies and instead comment on these results for the row-based approximations. The specific timings for each value of K are available in the accompanying code package [28]. The DEIM- and QDEIM-based row indices are all computed in less than 1 second. For the voltage magnitude data, the MILP-based selections are computed in the range of 5–15 seconds for larger values of K. For the phasor angle data, the MILP-based selection requires significantly more time: for $K = 5$, the selection takes approximately 70 seconds; for $K \geq 8$, this selection takes more than 1 000 seconds.

IV. Data-driven monitoring with ID-DEIM

Several works have proposed using changes in the low-dimensional subspace spanned by streaming PMU data to detect and localize system events in real time; see, e.g., [5], [8], [11]–[13], [16], [32]. Here, we propose an offline-online data-driven framework for real-time monitoring based on the IDs presented in Section II and the DEIM index selection

algorithm described in Section III. The proposed framework builds upon the online monitoring algorithm presented in Section II of [5], insofar as only a reduced number of pilot PMU data streams are used to monitor the network. Any non-pilot datastreams in the network can be recovered from these pilots, effectively reducing the dimension of the streaming PMU data.

In contrast to the work of [5], our algorithm views this dimension reduction through the lens of IDs (6), enabling the use of the interpolatory error bound $\eta_{\mathcal{S}}\sigma_{K+1}$ in (14). This perspective yields a few key operational benefits:

- Offline, DEIM is applied to select pilot streams $\mathcal{S}$, increasing the number of pilots K until the bound $\eta_{\mathcal{S}}\sigma_{K+1}$ is less than a user-specified tolerance.
- Online, $\eta_{\mathcal{S}}\sigma_{K+1}$ serves as an *estimator* of the interpolatory reconstruction error; deterioration of this estimate suggests a change in the operating condition of the network, and can be used as a simple "tripwire" for detecting disturbances.
- Following such a detection, the DEIM algorithm is applied to the transient system response due to the disturbance to localize the source of the event purely from data, with high accuracy.

Figure 3 presents a flowchart depicting the two-stage online-offline workflow of the proposed method. Numerical tests are interspersed throughout this section to illustrate the proposed framework.

A. Adaptive DEIM-based Training of Pilots

At this point, we differentiate between *offline* training data $\mathbf{Y}_{\text{trn}} \in \mathbb{R}^{N\times T_{\text{trn}}}$ and *online* streaming data $\mathbf{Y}_{\text{obs}} \in \mathbb{R}^{N\times T_{\text{obs}}}$. For simplicity, we assume that $\mathbf{Y}_{\text{trn}}$ and $\mathbf{Y}_{\text{obs}}$ contain data corresponding to a single measured grid quantity. Different quantities are dealt with by placing them into different PMU data matrices and processing them separately. In the numerical results of this section, we consider voltage magnitude and phasor angle data. The positive integers T_{trn}, T_{obs} dictate the size of the training and online monitoring windows. The parameter T_{trn} is tunable; in our tests, we choose T_{trn} to correspond to $\sim 120\,\text{s}$ worth of data collected during normal (ambient) operating conditions. On the other hand, we take T_{obs} to be fixed, but not necessarily known *a priori*.

For the online monitoring portion of the algorithm, where possibly the full $\mathbf{Y}_{\text{obs}}$ is approximated from a few pilot datastreams via an ID $\mathbf{Y}_{\mathcal{S}} = \mathbf{Z}_{\mathcal{S}}\mathbf{R}_{\mathcal{S}}$, two things need to be computed offline from $\mathbf{Y}_{\text{trn}}$ to formulate $\mathbf{Y}_{\mathcal{S}}$:

1. The indices $\mathcal{S}$ corresponding to the pilot datastreams that will form the basis of the ID;
2. The matrix of weights $\mathbf{Z}_{\mathcal{S}} \in \mathbb{R}^{N\times K}$ that encodes how the pilot datastreams in $\mathcal{S}$ should be combined to reproduce measurements at any non-pilot datastreams.

To choose the pilots in $\mathcal{S}$, DEIM is applied to the leading left singular vectors of $\mathbf{Y}_{\text{trn}}$ until $\eta_{\mathcal{S}}\sigma_{K+1} \leq \tau$, where $\tau > 0$ is a user-specified error tolerance. In our tests, for modest values of τ, e.g., $\tau = 10^{-1}$, the error bound satisfies $\eta_{\mathcal{S}}\sigma_{K+1} \leq \tau$ for small values of K, e.g., $K \leq 5$. A reliable heuristic for tuning τ is provided by the singular values of

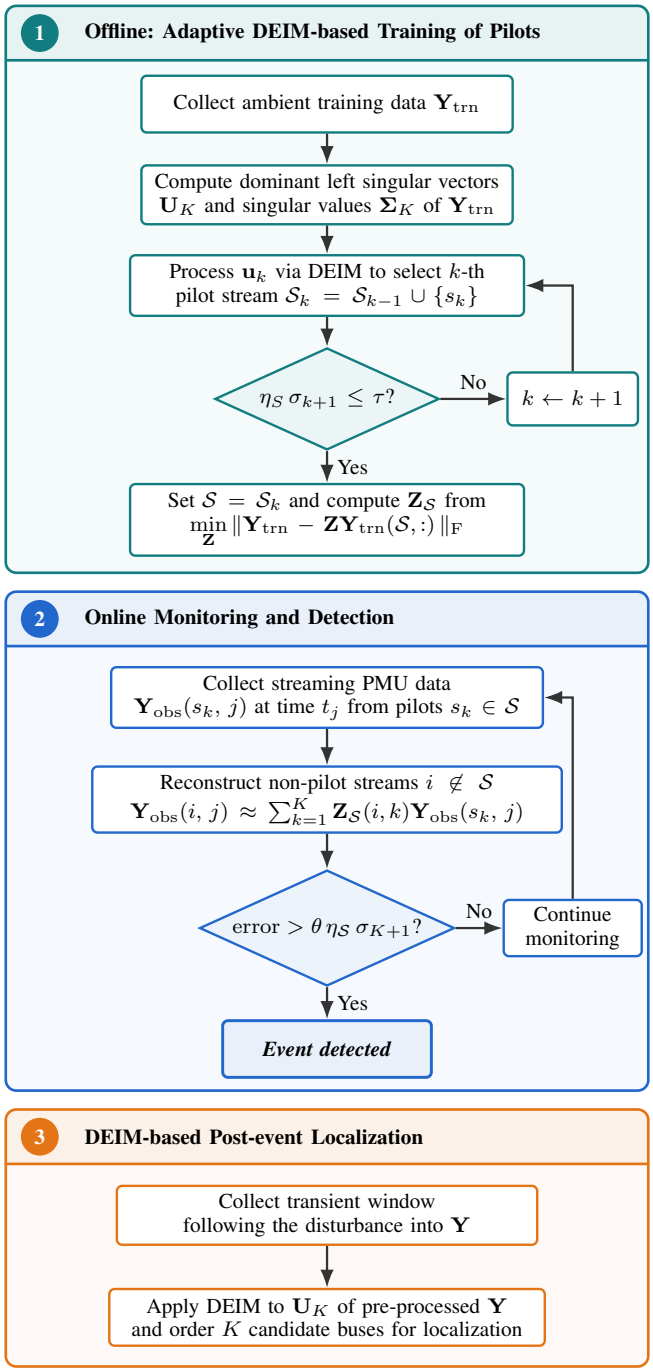
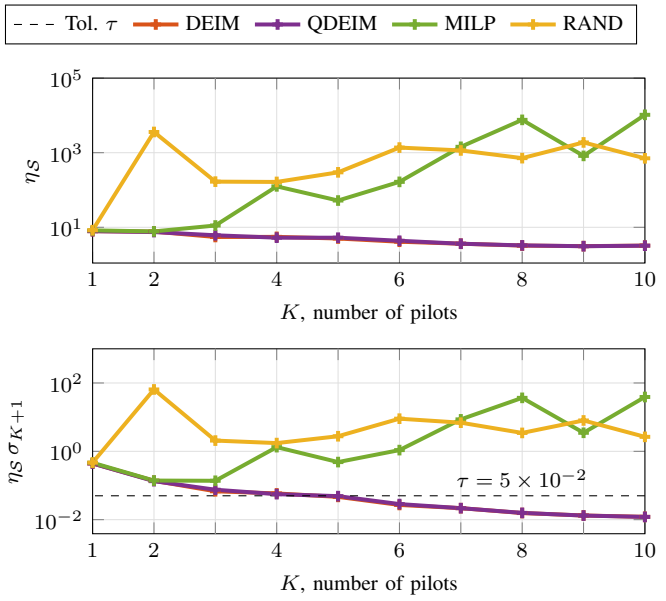


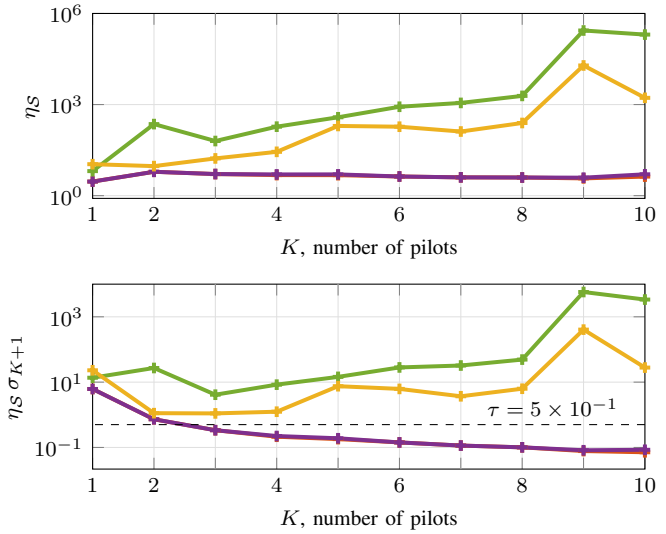
Figure 3: Flowchart of the online and offline workflows of the proposed ID-DEIM event monitoring framework for PMU-based event detection and post-event localization.

$\mathbf{Y}_{\text{trn}}$, since σ_{K+1} is the best rank-K approximation error. A smaller tolerance τ necessitates more pilots, giving a tradeoff between reconstruction accuracy and the bandwidth required to communicate across the K pilots. After the pilot streams $\mathcal{S}$ are identified, the matrix $\mathbf{Z}_{\mathcal{S}}$ is computed by solving the least-squares problem (11) with $\mathbf{Y} = \mathbf{Y}_{\text{trn}}$ and $\mathbf{R}_{\mathcal{S}} = \mathbf{Y}_{\text{trn}}(\mathcal{S},:)$.

This DEIM-based pilot selection method offers key advantages over other selection approaches. First, by design, the pilots chosen by DEIM yield small values of the error factor $\eta_{\mathcal{S}}$, and thus the bound in (14). Because of this, and the



(a) Evolution of the error factor $\eta_{\mathcal{S}}$ and the upper bound $\eta_{\mathcal{S}}\sigma_{K+1}$ throughout the adaptive training applied to the *voltage magnitude* data.



(b) Evolution of the error factor $\eta_{\mathcal{S}}$ and the upper bound $\eta_{\mathcal{S}}\sigma_{K+1}$ throughout the adaptive training applied the *phasor angle* data.

Figure 4: Evolution of the error factors $\eta_{\mathcal{S}}$ and the upper bounds $\eta_{\mathcal{S}}\sigma_{K+1}$ throughout the adaptive training as K pilots are chosen.

fact that we expect the data in $\mathbf{Y}_{\text{obs}}$ to exist near the same low-dimensional subspace spanned by $\mathbf{Y}_{\text{trn}}$, measurements from the non-pilot datastreams can be approximated from the pilots with high fidelity. Second, spurious retraining can be avoided by periodically recomputing $\eta_{\mathcal{S}}\sigma_{K+1}$ for the current set of pilots $\mathcal{S}$ but using a new batch of online data from all datastreams. So long as the heuristic "bound" $\eta_{\mathcal{S}}\sigma_{K+1}$ remains below an acceptable threshold, the current configuration of pilots is accepted. Otherwise, retraining is needed: DEIM is applied to $\mathbf{Y}_{\text{obs}}$ to select a new set of pilots.

Numerical Tests. We test the adaptive DEIM-based training of pilot-stream configurations on synthetic PMU data from the NETSNYPS 68-bus, 16-machine test system. Two sets of training data containing ambient voltage magnitudes and phasor angles are collected at a rate of 100 Hz over 120 s. To mimic ambient operating conditions, the mechanical power and exciter references of the generators are perturbed by zero-mean Gaussian white noise during the simulation.

The adaptive DEIM-based training algorithm is applied to the voltage magnitude and angle training data, separately, to compute pilot configurations for each measurement type. To observe the evolution of the error factor $\eta_{\mathcal{S}}$ and the bound $\eta_{\mathcal{S}}\sigma_{K+1}$ throughout the training procedure, we run DEIM for $K = 10$ iterations. We compare the DEIM-based training with pilot configurations computed from $\mathbf{Y}_{\text{trn}}$ using the QDEIM, MILP, and RAND selection schemes described in Section III-B. Because these other schemes are not inherently iterative like DEIM, we run them repeatedly for each fixed size $K = 1, 2, \ldots, 10$ of $\mathcal{S}$ (and thus, unlike DEIM, the resulting index sets need not be nested).

We report our results in Figure 4. The evolution of $\eta_{\mathcal{S}}$ and $\eta_{\mathcal{S}}\sigma_{K+1}$ as K, the number of pilots, grows for the voltage magnitude and phasor angle training data are shown in Figures 4a and 4b. We observe in both instances that these quantities steadily decrease with K as each new pilot is selected by DEIM and QDEIM. For the MILP- and RAND-based selections, both the error factor and thus the error bound tend to oscillate and generally *increase* as more pilots are added.

B. *Online Monitoring and Detection Using the Bound* (14)

After using $\mathbf{Y}_{\text{trn}}$ to determine the K pilot streams in $\mathcal{S}$ and the matrix $\mathbf{Z}_{\mathcal{S}}$ offline, dimension reduction of the online data $\mathbf{Y}_{\text{obs}}$ is achieved in real time via a rank-K ID $\mathbf{Y}_{\mathcal{S}}$. Suppose we want to recover data from one of the non-pilot streams $i \notin \mathcal{S}$ at the j-th time snapshot, i.e., $\mathbf{Y}_{\text{obs}}(i,\, j)$. To recover $\mathbf{Y}_{\text{obs}}(i,\, j)$, data from each of the K pilot streams $\mathbf{Y}_{\text{obs}}(s_k,\, j)$, $k = 1, \ldots, K$, are combined according to

$$\mathbf{Y}_{\text{obs}}(i,\, j) \approx \mathbf{Y}_{\mathcal{S}}(i,\, j) := \sum_{k=1}^{K} \mathbf{Z}_{\mathcal{S}}(i,k)\mathbf{Y}_{\text{obs}}(s_k,\, j). \quad (17)$$

We emphasize that the datastreams indicated by $\mathcal{S}$ are chosen based on the offline data $\mathbf{Y}_{\text{trn}}$, but the actual *online* datastreams in $\mathbf{Y}_{\text{obs}}$ are used to form the reconstruction (17). Because the training and online data are not the same, $\eta_{\mathcal{S}}\sigma_{K+1}$ (from $\mathbf{Y}_{\text{trn}}$) does not provide a theoretically rigorous upper bound for $|\mathbf{Y}_{\text{obs}}(i,j) - \mathbf{Y}_{\mathcal{S}}(i,j)|$, the online reconstruction error for the j-th sample of the non-pilot stream $i \notin \mathcal{S}$. However, the factor $\eta_{\mathcal{S}}\sigma_{K+1}$ can serve as an *estimator* of the online reconstruction error: because the training data $\mathbf{Y}_{\text{trn}}$ reflect normal operating conditions, we expect the low-dimensional subspaces spanned by $\mathbf{Y}_{\text{trn}}$ and $\mathbf{Y}_{\text{obs}}$ to be similar.

Suppose now that the actual reconstruction error of the j-th sample collected from a non-pilot stream $i \notin \mathcal{S}$ satisfies

$$\left|\mathbf{Y}_{\text{obs}}(i,j) - \mathbf{Y}_{\mathcal{S}}(i,j)\right| > \theta\, \eta_{\mathcal{S}}\, \sigma_{K+1} \quad (18)$$

for a calibration parameter $\theta > 0$. The condition (18) supposes that the *actual* reconstruction error exceeds the *estimated* error provided by the scaled error factor $\theta\, \eta_{\mathcal{S}}\, \sigma_{K+1}$. This violation suggests a flaw in the assumption that the low-dimensional

column space of $\mathbf{Y}_{\text{obs}}$ is close to that of $\mathbf{Y}_{\text{trn}}$, and thus signals that there has been a fundamental change in the network's operating condition, e.g., due to a disturbance.

We propose that the deterioration of the error estimate as in (18) be used as a simple mechanism for detecting changes to the network's operating conditions. The parameter $\theta > 0$ is a calibration multiplier that rescales the error estimator $\eta_{\mathcal{S}}\,\sigma_{K+1}$ and is used to balance the possibility of false alarms against missed disturbances. Taking $\theta > 1$ increases the chance of false negatives (FN: a change in the network occurs but goes unnoticed), whereas taking $\theta < 1$ will increase the chance of false positives (FP: a change in the network is detected where none has occurred). A practical way to choose θ is to compute the time-averaged value of the ratio $|\mathbf{Y}_{\text{obs}}(i,j) - \mathbf{Y}_{\mathcal{S}}(i,j)|/(\eta_{\mathcal{S}}\,\sigma_{K+1})$ for a non-pilot stream $i \notin \mathcal{S}$ during nominal operating conditions. Setting θ to this ratio would likely trigger an FP. Therefore, depending on the user's tolerance for FPs, θ can be chosen to be a value less than the computed ratio.

Necessarily, checking (18) requires continuously monitoring the interpolatory reconstruction error at some non-pilot datastreams $i \notin \mathcal{S}$. These non-pilot datastreams can be chosen based on the geography of the network, e.g., they may be collected from PMUs that are geographically distributed, to monitor for disturbances that are localized to a subnetwork. For our tests, we let the DEIM-based training procedure continue to run for M more iterations after $\eta_{\mathcal{S}}\sigma_{K+1} \leq \tau$ is satisfied, for some positive integer M. We use these M additional indices as the monitored non-pilot datastreams. Lastly, we mention that a similar detection mechanism based on the deterioration of the pilot-based reconstruction error is proposed in Section III.B of [5]. By comparison, our detector (18) is grounded in the theoretical upper bound on the interpolatory approximation error (14).

Numerical Tests. We investigate the proposed online monitoring algorithm with two numerical tests. First, we apply the algorithm to the voltage magnitude and phasor angle measurement data collected from the 60 s online simulation scenario of the NETSNYPS 68-bus, 16-machine test system from Section III-B during which a three-phase fault is applied to the line connecting buses 28 and 29 after 30 s. To select the pilot streams for each measurement type, we re-run the adaptive DEIM-based training algorithm on the 120 s of voltage magnitude and phasor angle training data from Section IV-A using $\tau = 5 \times 10^{-2}$ for the voltage data and $\tau = 5 \times 10^{-1}$ for the phasor angle data. (We use different tolerances for the two measurement types because the singular values of the voltage magnitude data decay more rapidly than those of the phasor angle data.)

The adaptive DEIM-based training algorithm selects $K = 5$ pilot streams for the voltage magnitude data and $K = 3$ pilot streams for the phasor angle data, both satisfying the prescribed error tolerances. For comparison, we run MILP to select $K = 5$ voltage magnitude pilot streams and $K = 3$ phasor angle pilot streams. Tables I and II report the pilot configurations for the voltage magnitudes and phasor angle data, along with the associated error bounds $\eta_{\mathcal{S}}\,\sigma_{K+1}$. For the same fixed K, the DEIM-based pilots produce an error bound

Table I: $K = 5$ voltage magnitude pilot datastreams as chosen by DEIM and MILP with a tolerance of $\tau = 5 \times 10^{-2}$, and values of the corresponding error estimator in (14).

	s_1	s_2	s_3	s_4	s_5	$\eta_{\mathcal{S}}\sigma_6$
DEIM	48	61	59	55	50	4.6488e−2
MILP	4	10	16	50	61	4.8467e−1

Table II: $K = 3$ phasor angle pilot datastreams as chosen by DEIM and MILP with a tolerance of $\tau = 5 \times 10^{-1}$, and values of the corresponding error estimator in (14).

	s_1	s_2	s_3	$\eta_{\mathcal{S}}\sigma_4$
DEIM	66	39	57	3.3689e−1
MILP	36	45	65	4.1213e0

that is roughly an order of magnitude less than the bound for the MILP-based pilots. For this experiment, we use $M = 2$ non-pilot streams to monitor for each measurement type: these are the next two pilots identified by DEIM after $K = 5$ and $K = 3$. The non-pilots correspond to the voltage magnitudes at buses 63 and 67, and the phasor angles at buses 61 and 68.

Following training, the reconstruction errors for the non-pilot datastreams are monitored and compared against the values of the error indicator $\theta\,\eta_{\mathcal{S}}\sigma_{K+1}$ (18). For this test, we set $\theta = 1$ as a baseline. The non-pilot voltage magnitude and phasor angle datastreams before and during the disturbance, along with their DEIM- and MILP-based reconstructions, are plotted in Figures 5 and 6. We also overlay the error plots with the values of the corresponding error estimator $\eta_{\mathcal{S}}\sigma_{K+1}$. Before the disturbance and during ambient operating conditions, both the DEIM- and MILP-based reconstructions of the non-pilot datastreams are accurate and well within their respective error estimates. Within one second of the line fault, the detection mechanism (18) is triggered by the DEIM-based reconstructions of the non-pilot voltage magnitude datastreams at both buses 63 and 67, and the non-pilot phasor angle datastream at bus 61. Therefore, the fault is correctly detected. For the non-pilot phasor angle datastream at bus 68, the estimator is very nearly violated; this motivates rescaling the estimator by values of $\theta < 1$. We emphasize however that the disturbance is still correctly detected because the condition (18) is triggered by at least one of the non-pilot streams. For the MILP-based reconstruction, the error remains within acceptable parameters according to the estimator (as $\eta_{\mathcal{S}}$ is excessively large), and thus no fault is detected.

For the second test, we verify the robustness of the approach in (18) in detecting disturbances on a collection of 359 event simulation scenarios of the 68-bus, 16-machine test system in MATLAB's PST. The dataset comprises three-phase line faults (111), line-to-ground faults (108), and loss-of-line events (tripping) without an electrical fault (140). The scenarios were generated by cycling through all transmission lines and applying the disturbance. The length of each simulation window is $T_{\text{obs}} = 100$ s. Each simulation is initialized from the same pre-disturbance operating point obtained by solving the network load-flow equations and initializing all dynamic

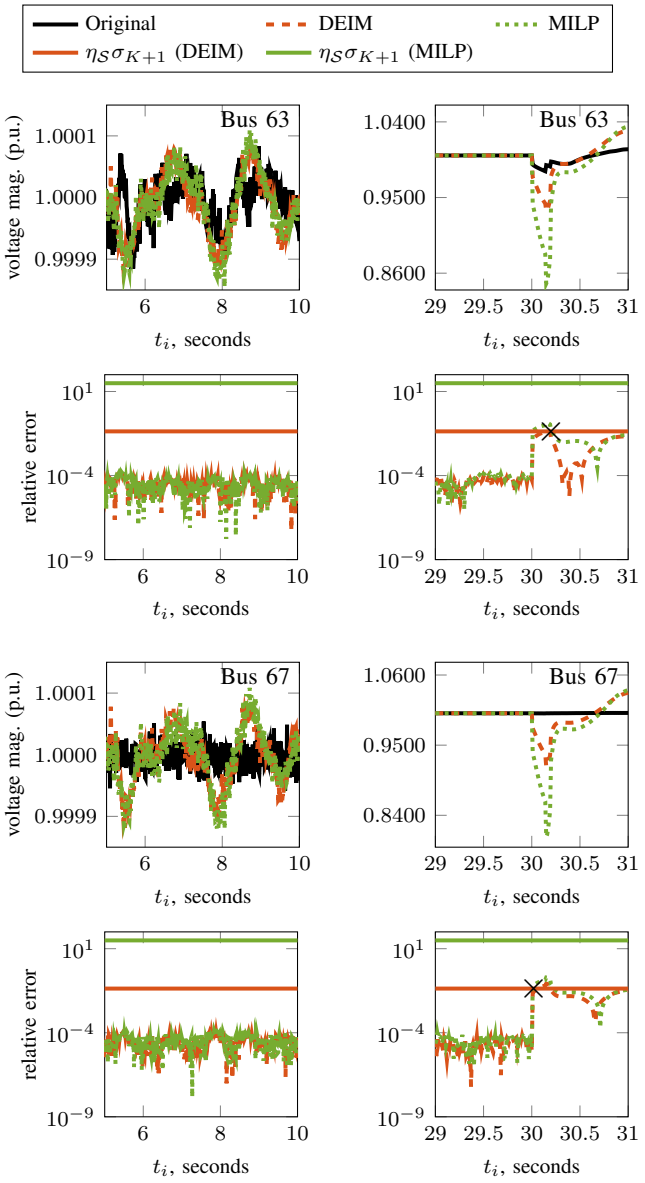


Figure 5: Interpolatory reconstruction and true data for the non-pilot voltage magnitude datastreams at bus 63 (top) and bus 67 (bottom) before and during a three-phase fault of the line between buses 28 and 29. The black marker $\times$ indicates the precise instance at which the error estimator is violated according to (18) with $\theta = 1$.

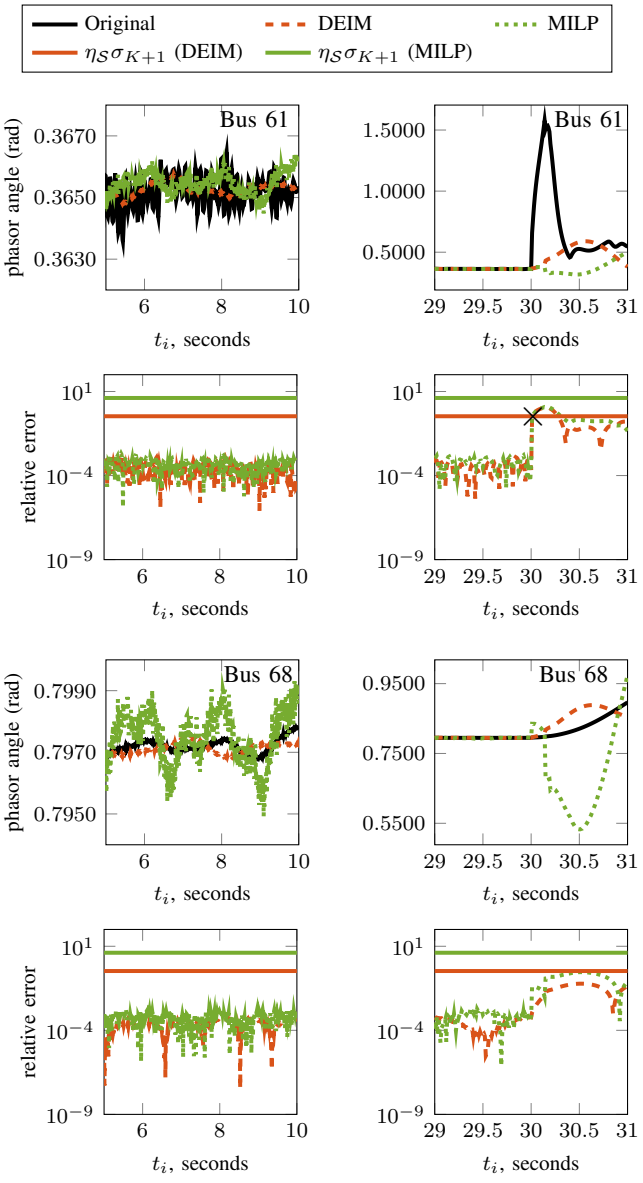


Figure 6: Interpolatory reconstruction and true data for the non-pilot voltage phasor angle datastreams at bus 61 (top) and bus 68 (bottom) before and during a three-phase fault of the line between buses 28 and 29. The black marker $\times$ indicates the precise instance at which the error estimator is violated according to (18) with $\theta = 1$.

states consistently with that operating point. For each disturbed line, we generate two distinct event scenarios: in the first, the disturbance occurs at a randomly selected time in the first half of the 100 s simulation window, and in the second, the disturbance occurs at a randomly selected time in the second half of the window. For a subset of these simulations, the inner Newton solve for the nonlinear-load bus voltages did not converge following the fault; these were not included in the final collection of 359 simulations. As before, the data are voltage magnitudes and phasor angles at each bus sampled at a rate of 100 Hz.

For monitoring the voltage magnitude and phasor angle datastreams, we use the same DEIM- and MILP-based pilot configurations reported in Tables I and II. We also monitor the same non-pilot voltage magnitude datastreams at buses $63, 67$ and non-pilot phasor angle datastreams at buses $61, 68$, as in the previous test. For each simulation scenario, we apply the detection mechanism (18) *separately* to the monitored voltage magnitude and phasor angle datastreams, and assess the resulting decisions independently. For each data type and simulation scenario, we classify the decision made by (18) as a TP (true positive: the disturbance is correctly detected at one of the non-pilots within 1 s of the event according to (18)), an FP (a fault is detected outside of this window at both or one of the non-pilots, and not detected correctly at the other), or an FN (no fault is detected at either of the non-pilots). We then evaluate the performance of the detection mechanism (18)

Table III: Precision, Recall, F_1 and F_2 scores for the event detection mechanism (18) using the DEIM- and MILP-based pilots for $\theta = 1$ and 10^{-2} applied to the 111 three-phase line fault simulations. The largest scores in each row are highlighted in **boldface**.

Voltage magnitudes	DEIM $\theta = 1$	DEIM $\theta = 10^{-2}$	MILP $\theta = 1$	MILP $\theta = 10^{-2}$
Precision	**0.9515**	0.9189	0.9063	0.9189
Recall	0.9245	**1.0000**	0.2685	**1.0000**
F_1	0.9378	**0.9577**	0.4143	**0.9577**
F_2	0.9298	**0.9827**	0.3125	**0.9827**
Phasor angles	**DEIM $\theta = 1$**	**DEIM $\theta = 10^{-2}$**	**MILP $\theta = 1$**	**MILP $\theta = 10^{-2}$**
Precision	0.3140	0.9279	0.0000	**0.9550**
Recall	0.5192	**1.0000**	0.0000	**1.0000**
F_1	0.3913	0.9626	0.0000	**0.9770**
F_2	0.4592	0.9847	0.0000	**0.9907**

Table IV: Precision, Recall, F_1 and F_2 scores for the event detection mechanism (18) using the DEIM- and MILP-based pilots for $\theta = 1$ and 10^{-2} applied to the 108 line-to-ground fault simulations. The largest scores in each row are highlighted in **boldface**.

Voltage magnitudes	DEIM $\theta = 1$	DEIM $\theta = 10^{-2}$	MILP $\theta = 1$	MILP $\theta = 10^{-2}$
Precision	0.9600	0.9537	**1.0000**	0.9537
Recall	0.9231	**1.0000**	0.2870	**1.0000**
F_1	0.9412	**0.9763**	0.4460	**0.9763**
F_2	0.9302	**0.9904**	0.3348	**0.9904**
Phasor angles	**DEIM $\theta = 1$**	**DEIM $\theta = 10^{-2}$**	**MILP $\theta = 1$**	**MILP $\theta = 10^{-2}$**
Precision	0.3059	0.9630	0.0000	**0.9907**
Recall	0.5306	**1.0000**	0.0000	**1.000**
F_1	0.3881	0.9811	0.0000	**0.9953**
F_2	0.4626	0.9923	0.0000	**0.9981**

using F_1 and F_2 scores [33], computed as

$$\mathrm{F}_1 = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}, \; \mathrm{F}_2 = \frac{5 \times \mathrm{Precision} \times \mathrm{Recall}}{4 \times \mathrm{Precision} + \mathrm{Recall}},$$

for $\mathrm{Precision} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$, $\mathrm{Recall} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$. Values close to 1 indicate a reliable detection mechanism. If $\mathrm{Precision}$ and Recall are both zero, we report the corresponding F_1 and F_2 scores as zero.

For the voltage magnitude and phasor angle data, we compute the F_1 and F_2 scores for four different detection mechanisms: (18) using the DEIM-based pilots with $\theta = 1$ and $\theta = 10^{-2}$, and (18) using the MILP-based pilots with the same two values of θ. We compute these scores separately for the three types of event simulations contained in the dataset: three-phase line faults, line-to-ground faults, and line tripping without a fault. These scores, along with the associated $\mathrm{Precision}$ and Recall, are recorded in Tables III, IV, and V. These experiments investigate the fidelity of the detection mechanism (18) as it applies to different types of disturbances.

We generally observe a substantial improvement in detection when θ is decreased from 1 to 10^{-2}. For the MILP-based pilot configuration, this improvement is extremely significant

Table V: Precision, Recall, F_1 and F_2 scores for the event detection mechanism (18) using the DEIM- and MILP-based pilots for $\theta = 1$ and 10^{-2} applied to the 140 loss-of-line (tripping) simulations. The largest scores in each row are highlighted in **boldface**.

Voltage magnitudes	DEIM $\theta = 1$	DEIM $\theta = 10^{-2}$	MILP $\theta = 1$	MILP $\theta = 10^{-2}$
Precision	0.2667	**0.9542**	0.0000	0.9189
Recall	0.0310	**0.9328**	0.0000	0.5075
F_1	0.0556	**0.9434**	0.0000	0.6538
F_2	0.0377	**0.9370**	0.0000	0.5574
Phasor angles	**DEIM $\theta = 1$**	**DEIM $\theta = 10^{-2}$**	**MILP $\theta = 1$**	**MILP $\theta = 10^{-2}$**
Precision	0.1667	**0.9516**	0.0000	0.8901
Recall	0.0667	**0.8806**	0.0000	0.6231
F_1	0.0952	**0.9147**	0.0000	0.7330
F_2	0.0758	**0.8939**	0.0000	0.6628

across the board, but most noticeably for the loss-of-line events. We attribute this behavior to the fact that the data deviate less dramatically from nominal operating conditions in response to such an event, and hence the unscaled estimator corresponding to $\theta = 1$ is less effective at capturing these small-scale deviations. For the voltage magnitude data, the DEIM-based pilots significantly outperform the MILP-based pilots for $\theta = 1$, and are on par with the MILP-based pilots for $\theta = 10^{-2}$. For the phasor angle data, the DEIM-based pilots perform better for $\theta = 1$, whereas the MILP-based pilots perform better for $\theta = 10^{-2}$.

C. Event Localization Using DEIM

After a disturbance occurs, it is imperative to find its source, e.g., the buses adjacent to a faulted line, quickly, so system operators can take corrective action to prevent cascading failures. Numerous works have explored the event location problem; see, e.g. [8], [11], [12]. As an alternative to these approaches, we propose using the DEIM algorithm to localize the source of disturbances. Once a disturbance has been detected using (18) (or any other detection mechanism), DEIM can be applied to a batch of data from *all* datastreams containing the transient system response following the disturbance. In our tests, as little as 1 s of data following the disturbance is needed to localize the event.

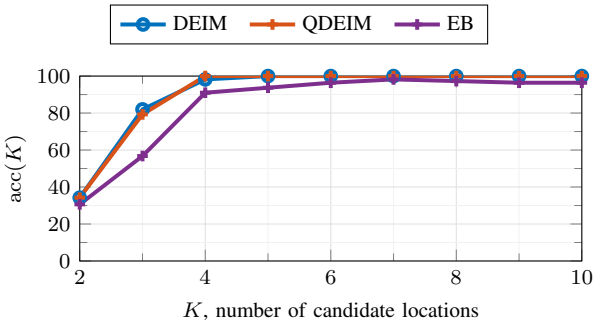
Numerical Tests. We demonstrate the ability of DEIM to localize disturbances using the same 359 simulation scenarios from Section IV-B. In the interest of space, we only report results for the voltage magnitude data. Results for the phasor angle data are available in the accompanying code package [28]. For comparison, we use the data-driven energy-based (EB) criterion for localizing affected buses from Section III.D of [8]. For each scenario, we assume that an event alert has been correctly issued. We then aggregate $0.5\,\mathrm{s}$ of data prior to the event and $1.0\,s$ of data directly after the event into the matrix $\mathbf{Y}$. Before attempting to localize the event's source, this matrix is preprocessed by removing the mean of the pre-event data, as in [8]. We apply DEIM, QDEIM, and EB to the leading K left singular vectors $\mathbf{U}_K$

of $\mathbf{Y}$ to select up to $K = 1, 2, \ldots, 10$ rows, each of which corresponds to a particular PMU in the network. Let $\mathcal{E}_K$ denote the top K buses identified by a given method, e.g., DEIM. Let $\mathcal{B}_i := \{s_{i_1}, s_{i_2}\} \subseteq \{1, \ldots, N\}$ contain the indices corresponding to the source of event i in the dataset; that is, $\mathcal{B}_i$ is a set containing a pair of bus indices connecting a faulted line for the line-based events. We measure the success of our method via the accuracy score

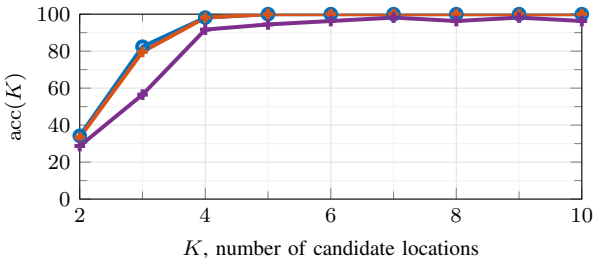
$$\mathrm{acc}(K) = \frac{1}{N_\mathrm{e}} \sum_{i=1}^{N_\mathrm{e}} 1(\mathcal{B}_i \subseteq \mathcal{E}_K) \times 100\,\%, \tag{19}$$

where $N_\mathrm{e} \geq 0$ is the number of events in the dataset, and $1(\mathcal{B}_i \subseteq \mathcal{E}_K)$ is an indicator function that equals 1 if $\mathcal{B}_i \subseteq \mathcal{E}_K$, i.e., the method correctly captures the source of the disturbance in its K candidate locations, and 0 otherwise. If *both* buses connected to the affected line are found within these k indices, we classify the method as having correctly localized the source of the event. This process is repeated for each of the 359 event simulation scenarios.

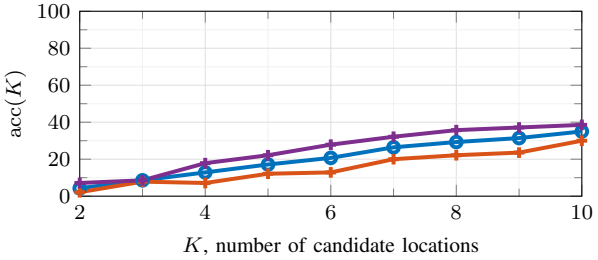
The accuracies (19) with which DEIM, QDEIM, or EB correctly identified the source of the event within $K = 2, \ldots, 10$ indices are plotted in Figure 7. For the three-phase and line-to-ground fault events and values of $K \geq 4$, all methods localize the source of the faulted line with greater than 90 percent accuracy. Significantly, DEIM and QDEIM are able to identify the source of the fault with 100% accuracy for $K \geq 5$. Thus, given the freedom to select enough indices, our DEIM- and QDEIM-based localization strategies correctly identify both affiliate buses connected to the faulted line in these fault-based scenarios, and perform marginally better than the reference approach in [8]. For the loss-of-line events without a fault, none of the considered methods achieve an accuracy score greater than 40 percent, and EB performs best for all values of $K = 2, \ldots, 10$. We observed similar behavior for the phasor angle data. In general, the source of the disturbance was harder to localize from these data, although EB was the most reliable on average.



(a) Localization accuracy of three-phase line fault event simulations (108 scenarios) for $K = 2, \ldots, 10$.



(b) Localization accuracy of line-to-ground line fault event simulations (108 scenarios) for $K = 2, \ldots, 10$.



(c) Localization accuracy of loss-of-line event simulations (108 scenarios) for $K = 2, \ldots, 10$.

Figure 7: Localization accuracy of all event simulations by DEIM, QDEIM, and EB for $K = 2, \ldots, 10$ using voltage magnitude data.

V. Conclusions

Interpolatory matrix decompositions (IDs) and the discrete empirical interpolation method (DEIM) provide effective tools for PMU data compression, network monitoring, and event detection. We have shown that IDs give effective low-rank approximations of PMU data, particularly for time-sensitive and bandwidth-limited applications such as wide-area monitoring. IDs can be maintained in real-time while interacting with $K \ll N$ pilot streams or $K \ll T$ pilot snapshots. Casting data compression in the mathematical framework of IDs provides a rigorous upper bound on the training error. To identify pilot buses for online monitoring, we employ DEIM, a greedy method that yields favorable error bounds. DEIM can be applied during offline training to adaptively select pilots until the error bound falls below a user's tolerance, yielding an estimate of the online reconstruction error. Any significant deterioration in this error estimate signals a notable change in the network's operating status relative to training conditions, providing a mechanism to detect disturbances. We have shown that DEIM and its QDEIM variant can robustly localize the source of disturbances.

Appendix

This appendix describes the Mixed-Integer Linear Programming (MILP) formulation discussed in Section III-B. Given a data matrix $\mathbf{Y} \in \mathbb{R}^{N \times T}$, the goal is to select a subset of K rows that are most orthogonal with each other. We next formally define the orthogonality metric. Following the procedure introduced in [5], the matrix $\mathbf{Y}$ is first projected onto its $m \geq K$ leading principal components, where m is chosen so that a certain amount of variance is preserved. Let $\widetilde{\mathbf{Y}} \in \mathbb{R}^{N \times T}$ denote the projected matrix, and $\widetilde{\mathbf{Y}}_{i,:}^\mathsf{T}$ denote its i-th row. If $\vartheta_{i,j}$ is the angle between the vectors $\widetilde{\mathbf{Y}}_{i,:}^\mathsf{T}$ and $\widetilde{\mathbf{Y}}_{j,:}^\mathsf{T} \in \mathbb{R}^{1 \times T}$, let us define the *cosine similarity*:

$$c_{i,j} := \cos(\vartheta_{i,j}) = \frac{\widetilde{\mathbf{Y}}_{i,:} \widetilde{\mathbf{Y}}_{j,:}^\mathsf{T}}{\|\widetilde{\mathbf{Y}}_{i,:}\|_2 \cdot \|\widetilde{\mathbf{Y}}_{j,:}\|_2}. \tag{20}$$

Note that $c_{i,j}$ is a scalar because $\widetilde{\mathbf{Y}}_{i,:}\widetilde{\mathbf{Y}}_{j,:}^{\mathsf{T}}$ is an inner product. Reference [5] selects a subset $\mathcal{S}$ of K rows so that the cosine similarity among the selected datastreams is as close to zero as possible (reflecting near-orthogonality of the vectors). We formulate this goal as the subset selection problem

$$\min_{\substack{\mathcal{S}\subseteq\{1,\dots,N\}\\ |\mathcal{S}|=K}} \max_{\substack{i<j\\ i,j\in\mathcal{S}}} |c_{i,j}|. \tag{21}$$

Solving (21) amounts to choosing K rows of $\widetilde{\mathbf{Y}}$ such that the largest pairwise (unsigned) cosine similarity among the selected rows is as small as possible, and hence the selected rows are as close to pairwise orthogonal as possible.

Problem (21) is not straightforward to solve, but can be posed as a MILP. For each row, introduce a binary decision variable taking the value $z_i = 1$ if the i-th row is selected, and $z_i = 0$ otherwise. Problem (21) can be reformulated as:

$$\min_{x,\{z_i\}_{i=1}^N} \quad x \tag{22a}$$

$$\text{subject to} \quad |c_{i,j}|(z_i + z_j - 1) \le x, \quad \forall i < j, \tag{22b}$$

$$\sum_{i=1}^{N} z_i = K, \tag{22c}$$

$$x \ge 0, \quad z_i \in \{0,1\} \quad \forall i. \tag{22d}$$

Because the epigraph variable x satisfies $x \ge 0$, the constraint (22b) is non-redundant only if $z_i = z_j = 1$. For those pairs, the constraint becomes $x \ge |c_{i,j}|$. Compiling all those pairs corresponding to non-redundant constraints in (22b), we get that $x \ge \max_{i,j\in\mathcal{S}} |c_{i,j}|$. Constraint (22c) enforces a budget of K rows. We solved this MILP in MATLAB using the `intlinprog` command with default settings. If instead of K rows, we want to select K columns of a matrix, the previous method would be applied to the transpose of $\mathbf{Y}$.

REFERENCES

[1] R. Klump, P. Agarwal, J. E. Tate, and H. Khurana, "Lossless compression of synchronized phasor measurements," in *IEEE Power and Energy Society General Meeting*. IEEE, 2010, pp. 1–7.

[2] P. H. Gadde, M. Biswal, S. Brahma, and H. Cao, "Efficient compression of PMU data in WAMS," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2406–2413, 2016.

[3] S. Das and T. S. Sidhu, "Application of compressive sampling in synchrophasor data communication in WAMS," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 450–460, 2013.

[4] S. Das, "Sub-Nyquist rate ADC sampling in digital relays and PMUs: Advantages and challenges," in *2016 IEEE 6th International Conference on Power Systems (ICPS)*. IEEE, 2016, pp. 1–6.

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

[6] N. Dahal, R. L. King, and V. Madani, "Online dimension reduction of synchrophasor data," in *PES T&D 2012*, 2012, pp. 1–7.

[7] M. Wang, J. H. Chow, D. Osipov, S. Konstantinopoulos, S. Zhang, E. Farantatos, and M. Patel, "Review of low-rank data-driven methods applied to synchrophasor measurement," *IEEE Open Access J. Power Energy*, vol. 8, pp. 532–542, 2021.

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

[9] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. Baltimore: Johns Hopkins University Press, 2012.

[10] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York: Springer, 2002.

[11] Z. Wang, Y. Zhang, and J. Zhang, "Principal components fault location based on WAMS/PMU measure system," in *2011 IEEE Power and Energy Society General Meeting*. IEEE, 2011, pp. 1–5.

[12] X. Liu, D. Laverty, R. Best, K. Li, D. Morrow, and S. McLoone, "Principal component analysis of wide-area phasor measurements for islanding detection—a geometric view," *IEEE Trans. Power Del.*, vol. 30, no. 2, pp. 976–985, 2015.

[13] M. Rafferty, X. Liu, D. M. Laverty, and S. McLoone, "Real-time multiple event detection and classification using moving window PCA," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2537–2548, 2016.

[14] 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, 2015.

[15] Y. Hao, M. Wang, J. H. Chow, E. Farantatos, and M. Patel, "Modelless data quality improvement of streaming synchrophasor measurements by exploiting the low-rank Hankel structure," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6966–6977, 2018.

[16] X. Kong, B. Foggo, K. Yamashita, and N. Yu, "Online voltage event detection using synchrophasor data with structured sparsity-inducing norms," *IEEE Trans. Power Syst.*, vol. 37, no. 5, pp. 3506–3515, 2021.

[17] M. W. Mahoney and P. Drineas, "CUR matrix decompositions for improved data analysis," *Proc. Nat. Acad. Sci.*, vol. 106, pp. 697–702, 2009.

[18] D. C. Sorensen and M. Embree, "A DEIM induced CUR factorization," *SIAM J. Sci. Comput.*, vol. 38, pp. A1454–A1482, 2016.

[19] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert, "Randomized algorithms for the low-rank approximation of matrices," *Proc. Nat. Acad. Sci.*, vol. 104, pp. 20 167–20 172, 2007.

[20] Y. Dong and P.-G. Martinsson, "Simpler is better: a comparative study of randomized pivoting algorithms for CUR and interpolative decompositions," *Adv. Comput. Math.*, vol. 49, 2023.

[21] G. W. Stewart, "Four algorithms for the efficient computation of truncated pivoted QR approximations to a sparse matrix," *Numer. Math.*, vol. 83, pp. 313–323, 1999.

[22] S. Liu, Y. Zhao, Z. Lin, Y. Liu, Y. Ding, L. Yang, and S. Yi, "Data-driven event detection of power systems based on unequal-interval reduction of PMU data and local outlier factor," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1630–1643, 2019.

[23] S. Chaturantabut and D. C. Sorensen, "Nonlinear model reduction via discrete empirical interpolation," *SIAM J. Sci. Comput.*, vol. 32, pp. 2737–2764, 2010.

[24] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, "An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations," *C. R. Math. Acad. Sci. Paris*, vol. 339, no. 9, pp. 667–672, 2004.

[25] Z. Drmac and S. Gugercin, "A new selection operator for the discrete empirical interpolation method——improved a priori error bound and extensions," *SIAM J. Sci. Comput.*, vol. 38, pp. A631–A648, 2016.

[26] E. P. Hendryx Lyons, "The discrete empirical interpolation method in class identification and data summarization," *WIREs Comput. Stat.*, vol. 16, no. 3, p. e1653, 2024.

[27] Y. P. Hong and C.-T. Pan, "Rank-revealing QR factorizations and the singular value decomposition," *Math. Comp.*, vol. 58, pp. 213–232, 1992.

[28] S. Reiter, "Code, data and results for numerical experiments in "Interpolatory Approximations of PMU Data: Dimension Reduction and Pilot Selection" (version 1.1)," Apr. 2026, https://doi.org/10.5281/zenodo.19772521.

[29] B. Pal and B. Chaudhuri, *Robust Control in Power Systems*. New York: Springer, 2005, ch. 4.

[30] 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, 1992.

[31] K. E. Martin, "Synchrophasor measurements under the IEEE standard C37. 118.1-2011 with amendment C37. 118.1 a," *"IEEE Trans. Power Del."*, vol. 30, no. 3, pp. 1514–1522, 2015.

[32] M. Wang, J. H. Chow, Y. Hao, S. Zhang, W. Li, R. Wang, P. Gao, C. Lackner, E. Farantatos, and M. Patel, "A low-rank framework of PMU data recovery and event identification," in *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, 2019, pp. 1–9.

[33] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," in *European Conference on Information Retrieval*. Springer, 2005, pp. 345–359.

Sean Reiter is a Courant Instructor with the Courant Institute of Mathematical Sciences at New York University. He obtained the B.S. degree in mathematics, the M.S. degree in mathematics, and the Ph.D. in mathematics in 2018, 2022, and 2025, respectively, from Virginia Tech, Blacksburg, Virginia, in the United States. His primary research interests are model-order reduction, data-driven modeling, rational approximation, numerical linear algebra, and data assimilation.

Mark Embree is a Professor in the Department of Mathematics at Virginia Tech. He obtained his D.Phil. from Oxford University in 2000. From 2002–2013 he was with the Department of Computational and Applied Mathematics at Rice University. From 2015–2025 he served as the faculty director of Virginia Tech's undergraduate major in Computational Modeling and Data Analytics. His current research interests include spectral theory, Schrödinger operators, and matrix computations.

Serkan Gugercin is a Professor of Mathematics at Virginia Tech, Blacksburg, Virginia, where he holds the Class of 1950 Professorship. He is also a core faculty member and a Deputy Director in the Division of Computational Modeling and Data Analytics. He received the B.S. degree in electrical and electronics engineering in 1992 from Middle East Technical University, Ankara, Turkey, and his M.S. and Ph.D. degrees in electrical engineering from Rice University, in 1999 and 2003, respectively. He received the National Science Foundation Early CAREER Award in Computational and Applied Mathematics in 2007, the Alexander von Humboldt Research Fellowship in 2016, and was named a SIAM Fellow in 2025. He is currently serving as an Associate Editor for Systems and Control Letters, and Computational Science and Engineering. His main research interests are model reduction, data-driven modeling, numerical linear algebra, approximation theory, and systems and control theory.

Vassilis Kekatos (SM'16) is an Associate Professor with the Schweitzer Power and Energy Systems group at the Elmore Family School of Electrical and Computer Engineering of Purdue University. He obtained his Ph.D. in Computer Science and Engineering from the Univ. of Patras, Greece, in 2007. He received a Marie Curie Fellowship from the European Commission during 2009-2012, and the US National Science Foundation CAREER Award in 2018. He was a postdoctoral research associate with the ECE Dept. at the Univ. of Minnesota. From 2015-2023, he was with the Bradley Department of ECE at Virginia Tech. From 2015 to 2022, he served as an Associate Editor for IEEE Trans. on Smart Grid, and now serves as an Associate Editor for IEEE Trans. on Power Systems. His current research focuses on optimization, machine learning, and quantum computing solutions for addressing power system computational tasks.