PETRELS: Parallel Subspace Estimation and Tracking by Recursive Least Squares From Partial Observations

Yuejie Chi, *Member, IEEE*, Yonina C. Eldar, *Fellow, IEEE*, and Robert Calderbank, *Fellow, IEEE*

***Abstract*—Many real world datasets exhibit an embedding of low-dimensional structure in a high-dimensional manifold. Examples include images, videos and internet traffic data. It is of great significance to estimate and track the low-dimensional structure with small storage requirements and computational complexity when the data dimension is high. Therefore we consider the problem of reconstructing a data stream from a small subset of its entries, where the data is assumed to lie in a low-dimensional linear subspace, possibly corrupted by noise. We further consider tracking the change of the underlying subspace, which can be applied to applications such as video denoising, network monitoring and anomaly detection. Our setting can be viewed as a sequential low-rank matrix completion problem in which the subspace is learned in an online fashion. The proposed algorithm, dubbed Parallel Estimation and Tracking by REcursive Least Squares (PETRELS), first identifies the underlying low-dimensional subspace, and then reconstructs the missing entries via least-squares estimation if required. Subspace identification is performed via a recursive procedure for each row of the subspace matrix in parallel with discounting for previous observations. Numerical examples are provided for direction-of-arrival estimation and matrix completion, comparing PETRELS with state of the art batch algorithms.**

***Index Terms*—Matrix completion, online algorithms, partial observations, recursive least squares, subspace identification and tracking.**

I. Introduction

When data is generated by a process that is governed by a small number of parameters, it can be represented as a low dimensional structure embedded in a much higher dimensional space. If the embedding is assumed linear, then the underlying low-dimensional structure becomes a linear subspace. Subspace Identification and Tracking (SIT) of a data stream plays an important role in various signal processing tasks such as online identification of network anomalies [2], moving target localization [3], beamforming [4], and video denoising [5].

A common way to determine low dimensional structure for static data is by using Principal Component Analysis (PCA) [6], which requires computing an eigendecomposition of an appropriate correlation matrix. In order to attempt to reduce the complexity in dynamic settings, typical SIT algorithms maintain an estimate of the underlying subspace at each time slot using all data collected at the current time and limited historical data about the subspace trajectory [7], [8].

When the data dimension is high, it may be impossible or prohibitively expensive to collect every data entry. In recommender systems it is unrealistic to expect every user to provide feedback on every product. In wireless sensor networks every measurement drains battery power and it is important to extend network lifetime by making fewer measurements. Hence there is growing interest in identifying and tracking a low-dimensional subspace from highly incomplete observations of the data stream.

Recent advances in Compressive Sensing (CS) [9]–[11] and Matrix Completion (MC) [12], [13] enable batch inference of data structure from observations that are highly incomplete with respect to the ambient dimension. CS enables reconstruction of a single vector from a few attributes by assuming it is sparse in a known basis or dictionary. MC reconstructs a matrix from a small subset of its entries assuming the matrix is low rank. It is equivalent to subspace identification from incomplete batch data since matrix reconstruction is straightforward once the row or column space is known.

MC does not require prior knowledge of rank and can be accomplished by minimizing the nuclear norm of the matrix [12], [13]. A common approach to render MC tractable is to pass to a convex relaxation of rank minimization just as sparse recovery can be made tractable by relying on ℓ_1-minimization [14]. Alternative approaches to MC include greedy algorithms such as OptSpace [15] and ADMiRA [16] which require an initial estimate of matrix rank.

The problem of testing whether a highly incomplete vector lies in a given subspace is also clearly related to MC. Here it is possible to show that hypothesis testing succeeds with high probability when the number of observed entries is slightly larger than the subspace rank [17]. We note that with high

Manuscript received July 12, 2012; revised January 11, 2013; accepted August 22, 2013. Date of publication September 20, 2013; date of current version November 01, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ta-Hsin Li. The work of Y. Chi and R. Calderbank was supported by ONR under Grant N00014-08-1-1110, by AFOSR under Grant FA 9550-09-1-0643, and by NSF under Grants NSF CCF-0915299 and NSF CCF-1017431. The work of Y. C. Eldar was supported in part by the Ollendorf Foundation, and by the Israel Science Foundation under Grant 170/10. This paper was presented in part at the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2012.

Y. Chi is with the Department of Electrical and Computer Engineering and the Department of Biomedical Informatics, The Ohio State University, Columbus, OH 43210 USA (e-mail: chi.97@osu.edu).

Y. C. Eldar is with the Department of Electrical Engineering, Technion, Israel Institute of Technology, Haifa 32000, Israel (e-mail: yonina@ee.technion.ac.il).

R. Calderbank is with the Department of Computer Science, Duke University, Durham, NC 27708 USA (e-mail: robert.calderbank@duke.edu).

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

Digital Object Identifier 10.1109/TSP.2013.2282910

probability it is also possible to estimate the covariance matrix of a dataset from incomplete batch data [18].

Given partial observations from a data stream, we introduce a new SIT algorithm, Parallel Estimation and Tracking by REcursive Least Squares, which we abbreviate as PETRELS. The underlying low-dimensional subspace is identified by minimizing a geometrically discounted sum of projection residuals on the observed entries at each time index. If missing entries are required then they can be reconstructed via least squares estimation. The discount factor maintains a balance between capturing long term behavior and responding to changes in that behavior. PETRELS represents the underlying subspace as the row space of a matrix and the discount factor is applied to each row of this matrix *in parallel*. The subspace is updated *recursively* so that it is not necessary to retain historical data indefinitely. Run time is on the order of $\mathcal{O}(r^3)$ per time index, where r is the subspace rank. Updating rows of the subspace matrix in parallel renders run time independent of the ambient dimension of the data stream.

If the underlying subspace is fixed and the data stream is fully observed, then we show that PETRELS converges to the true subspace by connecting to prior analysis of the Projection Approximation Subspace Tracking (PAST) algorithm [7]. For partially observed data, PETRELS is a second order stochastic gradient descent algorithm. We show that it always converges locally to a stationary point of the proposed objective function.

Section VI provides a numerical assessment of how well PETRELS is able to respond to changes in the underlying subspace. The context for the numerical examples is direction-of-arrival estimation and we measure the impact of the fraction of observed entries, the discount factor, and the subspace rank. We compare performance against the GROUSE algorithm [19] which uses rank-one updates to track the underlying subspace on the Grassmannian manifold. The performance of GROUSE is limited by the existence of "barriers" in the search path [20] which results in GROUSE being trapped at local minima. In contrast, updates in PETRELS are not restricted to the Grassmannian manifold. We show that PETRELS is better able to separate closely located modes and to respond quickly to changes in the underlying scene in the context of direction-of-arrival estimation. We also compare PETRELS with state of the art batch MC algorithms and show that it is competitive in terms of the tradeoff between run time and accuracy.

The rest of the paper is organized as follows. Section II introduces the problem of subspace tracking and describes prior work. Implementation of PETRELS is considered in Section III, while Section IV addresses convergence when the data stream is fully observed. Extensions to PETRELS that improve robustness, reduce complexity, and incorporate compressive measurements are presented in Section V. Numerical results are presented in Section VI and conclusions in Section VII.

II. Problem Statement and Related Work

A. Problem Statement

We consider the following problem. At each time n, a vector $\mathbf{x}_n \in \mathbb{R}^M$ is generated as:

$$\mathbf{x}_n = \mathbf{U}_n\mathbf{a}_n + \mathbf{n}_n \in \mathbb{R}^M, \tag{1}$$

where the columns of $\mathbf{U}_n \in \mathbb{R}^{M\times r_n}$ span a low-dimensional subspace, the vector $\mathbf{a}_n \in \mathbb{R}^{r_n}$ specifies the linear combination of columns, and $\mathbf{n}_n$ is additive white Gaussian noise distributed as $\mathbf{n}_n \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_M)$. When we analyze convergence in Section IV we will make the additional assumption that $\mathbf{a}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{r_n})$. The rank r_n of the underlying subspace at time n is allowed to change slowly over time. It is assumed to be bounded above by a constant r but it is not required to be known exactly at any specific time n. The entries in $\mathbf{x}_n$ might represent measurements in a sensor network, pixel values in a video frame, or individual movie ratings.

We collect only partial entries of the full vector $\mathbf{x}_n$ at time n. Partial observations correspond to point-wise multiplication of the vector $\mathbf{x}_n$ by a binary mask $\mathbf{P}_n = \mathrm{diag}\{\mathbf{p}_n\}$, where $\mathbf{p}_n = [p_{1n}, p_{2n}, \cdots, p_{Mn}]^T \in \{0,1\}^M$ with $p_{mn} = 1$ if the mth entry is observed at time n. The set of observed entries at time n is denoted by

$$\mathbf{y}_n = \mathbf{p}_n \odot \mathbf{x}_n = \mathbf{P}_n\mathbf{x}_n \in \mathbb{R}^M, \tag{2}$$

where $\odot$ stands for point-wise multiplication. We denote $\Omega_n = \{m : p_{mn} = 1\}$ as the set of observed entries at time n. In a random observation model every entry of the vector $\mathbf{x}_n$ is observed uniformly at random.

Given a sequence of incomplete observations $(\mathbf{y}_t, \mathbf{p}_t)_{t=1}^n$, we seek to identify and track changes in the underlying subspace. The output of our online algorithm at time n is an $M \times r$ matrix $\mathbf{D}_n$, where the rank of the estimated subspace $\mathbf{D}_n$ is assumed known and fixed throughout the algorithm as r. The target subspace is the column space of this matrix, and is ideally equivalent to the column space of $\mathbf{U}_n$. The following properties are desirable.

- *Small storage*: The storage required by the online algorithm should not grow with the volume of historical data.
- *Adaptivity*: The online algorithm should respond quickly to changes in the underlying subspace.
- *Convergence*: If the underlying subspace is constant then the subspace generated from the online algorithm should converge to the true subspace.

In Section III, we show that the algorithm proposed in this paper satisfies the first two desiderata. In Section IV, we prove that when the data stream is fully observed, our algorithm converges to the true subspace. If the data stream is partially observed then we are able to establish local convergence.

B. Conventional Subspace Identification and Tracking

The problem of subspace identification and tracking when the data $\mathbf{x}_n$ are fully observed has been widely studied in the signal processing literature (see [21] and the references therein). In this scenario, the Projection Approximation Subspace Tracking (PAST) algorithm [7] is most similar to our algorithm so we begin by describing PAST.

For simplicity assume $\mathbf{U}_n = \mathbf{U}$ is fixed over time, and consider a scalar function $J(\mathbf{W})$ with respect to a subspace $\mathbf{W} \in \mathbb{R}^{M\times r}$, given by

$$J(\mathbf{W}) = \mathbb{E}\|\mathbf{x}_n - \mathbf{W}\mathbf{W}^T\mathbf{x}_n\|_2^2, \tag{3}$$

where the expectation is taken with respect to $\mathbf{x}_n$. Let $\mathbf{C_x} = \mathbb{E}[\mathbf{x}_n\mathbf{x}_n^T] = \mathbf{U}\mathbf{U}^T + \sigma^2\mathbf{I}_M$ be the data covariance matrix assuming that $\mathbf{a}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_r)$. It is shown in [7] that the global minima of $J(\mathbf{W})$ is the only stable stationary point, and it is given by $\mathbf{W} = \mathbf{U}_r\mathbf{Q}$ with orthogonal columns, where $\mathbf{U}_r$ is composed of the r dominant eigenvectors of $\mathbf{C_x}$, and $\mathbf{Q} \in \mathbb{C}^{r\times r}$ is a unitary matrix. Without loss of generality, we can choose $\mathbf{W}$ to be composed of orthogonal columns which span the column space of $\mathbf{U}$. This motivates PAST to optimize the following function at time n without constraining $\mathbf{W}$ to have orthogonal columns:

$$\mathbf{W}_n = \underset{\mathbf{W}\in\mathbb{R}^{M\times r}}{\arg\min} \sum_{t=1}^{n} \alpha^{n-t}\|\mathbf{x}_t - \mathbf{W}\mathbf{W}^T\mathbf{x}_t\|_2^2 \quad (4)$$

$$\approx \underset{\mathbf{W}\in\mathbb{R}^{M\times r}}{\arg\min} \sum_{t=1}^{n} \alpha^{n-t}\left\|\mathbf{x}_t - \mathbf{W}\mathbf{W}_{t-1}^T\mathbf{x}_t\right\|_2^2. \quad (5)$$

The expectation in (3) is replaced in (4) by a sum in which prior observations are discounted by a geometric factor $0 \ll \alpha \leq 1$. This sum is further approximated in (5) where replacement of the second $\mathbf{W}$ by $\mathbf{W}_{t-1}$ leads to a recursion for $\mathbf{W}_n$. The matrix $\mathbf{W}_n$ is found by first estimating the coefficient vector $\hat{\mathbf{a}}_t$ as $\hat{\mathbf{a}}_t = \mathbf{W}_{t-1}^T\mathbf{x}_t$, then updating the matrix estimate as

$$\mathbf{W}_n = \underset{\mathbf{W}\in\mathbb{R}^{M\times r}}{\arg\min} \sum_{t=1}^{n} \alpha^{n-t}\|\mathbf{x}_t - \mathbf{W}\hat{\mathbf{a}}_t\|_2^2. \quad (6)$$

The PAST algorithm belongs to the class of power-based techniques, which include Oja's method [22], the Novel Information Criterion (NIC) method [23] and others. These algorithms have been analyzed in [24] within a uniform framework with slight variations for each approach. Specifically, the subspace estimate $\mathbf{W}_n \in \mathbb{R}^{M\times r}$ is updated at time n as

$$\mathbf{W}_n = \mathbf{C}_n\mathbf{W}_{n-1}\left(\mathbf{W}_{n-1}^T\mathbf{C}_n^2\mathbf{W}_{n-1}\right)^{-1/2}, \quad (7)$$

where $\mathbf{C}_n$ is the sample data covariance matrix given by

$$\mathbf{C}_n = \alpha_n\mathbf{C}_{n-1} + \mathbf{x}_n\mathbf{x}_n^T, \quad (8)$$

with $0 < \alpha_n \leq 1$. The normalization in (7) ensures that the updated subspace $\mathbf{W}_n$ is orthogonal. This normalization is not performed in all power-based algorithms.

If we were able to replace $\mathbf{C}_n$ in (7) by the ground truth $\mathbf{C_x}$, then it is shown in [24] that these power-based methods will converge to the principal subspace spanned by the most significant r eigenvectors of $\mathbf{C_x}$. When the entries of the data vector are fully observed, $\mathbf{C}_n$ converges rapidly to $\mathbf{C_x}$, explaining why power-based methods perform very well in practice. However, if the fraction of entries observed at time n is relatively small, then only a fraction of entries in $\mathbf{C}_{n-1}$ are updated at time n so that convergence is slow. Therefore, in the partially observed setting, it is ineffective to apply the above methods without substantial modification.

C. Matrix Completion

When $\mathbf{U}_n = \mathbf{U}$, our problem is closely related to the MC problem. Assume $\mathbf{X} \in \mathbb{R}^{M\times n}$ is a low-rank matrix, and $\mathbf{P} \in \{0,1\}^{M\times n}$ is a binary mask matrix with 0 at missing entries and 1 at observed entries. Let $\mathbf{Y} = \mathbf{P} \odot \mathbf{X} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$ be the observed partial matrix where the missing entries are filled in as zeros. MC aims to solve the following problem:

$$\min_{\mathbf{Z}} \ \mathrm{rank}(\mathbf{Z}) \quad \text{s.t. } \mathbf{Y} - \mathbf{P} \odot \mathbf{Z} = \mathbf{0}, \quad (9)$$

namely, to find a matrix with minimal rank such that the observed entries are satisfied. The rank constraint makes this optimization problem intractable. However, given weak conditions on $\mathbf{X}$ it can be shown that the solution coincides with that of the following nuclear norm minimization problem (see [12] for details):

$$\min_{\mathbf{Z}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{P} \odot \mathbf{Z}\|_F^2 + \mu\|\mathbf{Z}\|_*. \quad (10)$$

Here $\|\mathbf{Z}\|_*$ is the nuclear norm of $\mathbf{Z}$, i.e. the sum of singular values of $\mathbf{Z}$, and $\mu > 0$ is a regularization parameter. The nuclear norm of $\mathbf{Z}$ can be expressed as [25]

$$\|\mathbf{Z}\|_* = \min_{\mathbf{U},\mathbf{V}:\mathbf{Z}=\mathbf{U}\mathbf{V}^T} \frac{1}{2}\left(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2\right) \quad (11)$$

where $\mathbf{U} \in \mathbb{C}^{M\times r}$ and $\mathbf{V} \in \mathbb{C}^{n\times r}$. Substituting (11) into (10) we can rewrite the MC problem as

$$\min_{\mathbf{U},\mathbf{V}} \left\|\mathbf{P} \odot (\mathbf{X} - \mathbf{U}\mathbf{V}^T)\right\|_F^2 + \mu\left(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2\right). \quad (12)$$

Our problem formulation can be viewed as an online way of solving (12) which avoids large matrix multiplications. We simply define a random process that first selects columns of $\mathbf{X}$ uniformly and then selects a subset of entries uniformly from the given column. MC reduces to the problem of identifying the underlying column space since the matrix $\mathbf{X}$ can be recovered from this estimate by the method of least squares. The potential tradeoff between performance and complexity is explored in Section VI where PETRELS is compared with standard MC algorithms.

III. The PETRELS Algorithm

We now describe our proposed Parallel Estimation and Tracking by REcursive Least Squares (PETRELS) algorithm.

A. Objective Function

We assume that the rank r of the target subspace is known and that it remains fixed throughout. Note that the dimension of the true subspace may be smaller than r. Subspaces appear throughout as the column spaces of matrices and we shall describe our algorithm as a procedure for updating matrices. Given an $M \times r$ matrix $\mathbf{D}$, we define the total projection residual $f_n(\mathbf{D})$ on the observed entries at time n by

$$f_n(\mathbf{D}) = \min_{\mathbf{a}} \|\mathbf{P}_n(\mathbf{x}_n - \mathbf{D}\mathbf{a})\|_2^2. \quad (13)$$

At time n we select the r-dimensional subspace $\mathbf{D}_n$ that minimizes the loss function $F_n(\mathbf{D})$ given by

$$\mathbf{D}_n = \underset{\mathbf{D}\in\mathbb{R}^{M\times r}}{\arg\min} F_n(\mathbf{D}) = \underset{\mathbf{D}\in\mathbb{R}^{M\times r}}{\arg\min} \sum_{t=1}^{n} \lambda^{n-t} f_t(\mathbf{D}), \quad (14)$$

where the parameter $0 \ll \lambda \leq 1$ discounts past observations. To motivate the loss function in (14) we note that if $\mathbf{U}_n = \mathbf{U}$ is not changing over time, then the right hand side of (14) is minimized to zero when $\mathbf{D}_n$ spans the subspace defined by $\mathbf{U}$.

If $\mathbf{U}_n$ is slowly changing, then λ is used to control the memory and tracking ability at time n.

Before developing PETRELS, we note that if there are further constraints on the coefficients $\mathbf{a}$'s, a regularization term can be incorporated in $f_n(\mathbf{D})$ as:

$$f_n(\mathbf{D}) = \min_{\mathbf{a}\in\mathbb{R}^r} \|\mathbf{P}_n(\mathbf{x}_n - \mathbf{D}\mathbf{a})\|_2^2 + \beta\|\mathbf{a}\|_p, \tag{15}$$

where $p \geq 0$. For example, $p = 1$ enforces a sparsity constraint, and $p = 2$ enforces an energy constraint.

In (14) the discount factor λ is fixed, and the influence of past observations decreases geometrically; a more general online objective function can be given as

$$F_n(\mathbf{D}) = \lambda_n F_{n-1}(\mathbf{D}) + f_n(\mathbf{D}), \tag{16}$$

where the sequence $\{\lambda_n\}$ is used to control the memory and adaptivity of the algorithm in a more flexible way.

Fixing $\mathbf{D}$, $f_n(\mathbf{D})$ can be written as

$$f_n(\mathbf{D}) = \mathbf{x}_n^T\left(\mathbf{P}_n - \mathbf{P}_n\mathbf{D}(\mathbf{D}^T\mathbf{P}_n\mathbf{D})^\dagger\mathbf{D}^T\mathbf{P}_n\right)\mathbf{x}_n, \tag{17}$$

where † denotes the pseudo-inverse. Plugging this back into (14) the exact optimization problem becomes:

$$\mathbf{D}_n = \arg\min_{\mathbf{D}\in\mathbb{R}^{M\times r}} \sum_{t=1}^{n} \lambda^{n-t}\mathbf{x}_t^T\left[\mathbf{P}_t - \mathbf{P}_t\mathbf{D}(\mathbf{D}^T\mathbf{P}_t\mathbf{D})^\dagger\mathbf{D}^T\mathbf{P}_t\right]\mathbf{x}_t.$$

This problem requires storing all previous observations and is difficult to solve exactly. PETRELS provides an approximate solution.

B. PETRELS

The proposed PETRELS algorithm is summarized by Algorithm 1. At each time n, PETRELS alternates between estimating the coefficient vector $\mathbf{a}_n$ and updating the subspace $\mathbf{D}_n$. The estimate $\hat{\mathbf{a}}_n$ for the coefficient vector $\mathbf{a}_n$ is obtained by minimizing the projection residual on the subspace $\mathbf{D}_{n-1}$ derived at time $n-1$:

$$\begin{aligned}\hat{\mathbf{a}}_n &= \arg\min_{\mathbf{a}\in\mathbb{R}^r} \|\mathbf{P}_n(\mathbf{x}_n - \mathbf{D}_{n-1}\mathbf{a})\|_2^2 \\ &= \left(\mathbf{D}_{n-1}^T\mathbf{P}_n\mathbf{D}_{n-1}\right)^\dagger\mathbf{D}_{n-1}^T\mathbf{y}_n,\end{aligned} \tag{18}$$

where $\mathbf{D}_0 \in \mathbb{R}^{M\times r}$ is a random subspace initialization. The full vector $\mathbf{x}_n$ is then estimated as:

$$\hat{\mathbf{x}}_n = \mathbf{D}_{n-1}\hat{\mathbf{a}}_n. \tag{19}$$

The subspace $\mathbf{D}_n$ is then updated by minimizing

$$\mathbf{D}_n = \arg\min_{\mathbf{D}} \sum_{t=1}^{n} \lambda^{n-t}\|\mathbf{P}_t(\mathbf{x}_t - \mathbf{D}\hat{\mathbf{a}}_t)\|_2^2, \tag{20}$$

where $\hat{\mathbf{a}}_t, t = 1,\cdots,n$ are estimates from (18). We have simplified the problem of finding $\mathbf{D}_n$ by replacing the optimal coefficients appearing in (14) with previously estimated coefficients. The discount factor mitigates error propagation and enables the algorithm to recover from losses incurred by the use of these estimated coefficients.

Algorithm 1: PETRELS for SIT from Partial Observations

```
Input: a stream of vectors $\mathbf{y}_n$, observed patterns $\mathbf{P}_n$ and $\lambda$.
Initialization: an $M \times r$ random matrix $\mathbf{D}_0 = [\mathbf{d}_1^0, \mathbf{d}_2^0, \cdots, \mathbf{d}_M^0]^T$, and $(\mathbf{R}_m^0)^\dagger = \delta\mathbf{I}_r$, $\delta > 0$ for all $m = 1, \cdots, M$.
for $n = 1, 2, \cdots$ do
    $\hat{\mathbf{a}}_n = (\mathbf{D}_{n-1}^T\mathbf{P}_n\mathbf{D}_{n-1})^\dagger\mathbf{D}_{n-1}^T\mathbf{y}_n$.
    If stream reconstruction is required: $\hat{\mathbf{x}}_n = \mathbf{D}_{n-1}\hat{\mathbf{a}}_n$.
    for $m = 1, \cdots, M$ do
        $\beta_m^n = 1 + \lambda^{-1}\mathbf{a}_n^T(\mathbf{R}_m^{n-1})^\dagger\hat{\mathbf{a}}_n$,
        $\mathbf{v}_m^n = \lambda^{-1}(\mathbf{R}_m^{n-1})^\dagger\hat{\mathbf{a}}_n$,
        $(\mathbf{R}_m^n)^\dagger = \lambda^{-1}(\mathbf{R}_m^{n-1})^\dagger - p_{mn}(\beta_m^n)^{-1}\mathbf{v}_m^n(\mathbf{v}_m^n)^T$,
        $\mathbf{d}_m^n = \mathbf{d}_m^{n-1} + p_{mn}(x_{mn} - \hat{\mathbf{a}}_n^T\mathbf{d}_m^{n-1})(\mathbf{R}_m^n)^\dagger\hat{\mathbf{a}}_n$.
    end for
end for
```

The objective function in (20) decomposes into a parallel set of smaller problems, one for each row of $\mathbf{D}_n = [\mathbf{d}_1^n, \mathbf{d}_2^n, \cdots, \mathbf{d}_M^n]^T$. Thus

$$\mathbf{d}_m^n = \arg\min_{\mathbf{d}_m} \sum_{t=1}^{n} \lambda^{n-t}p_{mt}\left(x_{mt} - \hat{\mathbf{a}}_t^T\mathbf{d}_m\right)^2, \tag{21}$$

for $m = 1, \cdots, M$. To find the optimal $\mathbf{d}_m^n$, we set the derivative of (21) equal to zero, resulting in

$$\left(\sum_{t=1}^{n} \lambda^{n-t}p_{mt}\hat{\mathbf{a}}_t\hat{\mathbf{a}}_t^T\right)\mathbf{d}_m^n = \sum_{t=1}^{n} \lambda^{n-t}p_{mt}x_{mt}\hat{\mathbf{a}}_t.$$

This equation can be rewritten as

$$\mathbf{R}_m^n\mathbf{d}_m^n = \mathbf{s}_m^n, \tag{22}$$

where $\mathbf{R}_m^n = \sum_{t=1}^n \lambda^{n-t}p_{mt}\hat{\mathbf{a}}_t\hat{\mathbf{a}}_t^T$ and $\mathbf{s}_m^n = \sum_{t=1}^n \lambda^{n-t}p_{mt}x_{mt}\hat{\mathbf{a}}_t$. Therefore, $\mathbf{d}_m^n$ can be found as

$$\mathbf{d}_m^n = (\mathbf{R}_m^n)^\dagger\mathbf{s}_m^n. \tag{23}$$

When $\mathbf{R}_m^n$ is not invertible, we choose the least-norm solution.

We now show how (22) can be updated recursively. For $m = 1, \ldots, M$, we first rewrite

$$\mathbf{R}_m^n = \lambda\mathbf{R}_m^{n-1} + p_{mn}\hat{\mathbf{a}}_n\hat{\mathbf{a}}_n^T, \tag{24}$$

$$\mathbf{s}_m^n = \lambda\mathbf{s}_m^{n-1} + p_{mn}x_{mn}\hat{\mathbf{a}}_n. \tag{25}$$

Next, substitute (24) and (25) into (22) to obtain

$$\begin{aligned}\mathbf{R}_m^n\mathbf{d}_m^n &= \lambda\mathbf{s}_m^{n-1} + p_{mn}x_{mn}\hat{\mathbf{a}}_n \\ &= \lambda\mathbf{R}_m^{n-1}\mathbf{d}_m^{n-1} + p_{mn}x_{mn}\hat{\mathbf{a}}_n \\ &= \mathbf{R}_m^n\mathbf{d}_m^{n-1} - p_{mn}\hat{\mathbf{a}}_n\hat{\mathbf{a}}_n^T\mathbf{d}_m^{n-1} + p_{mn}x_{mn}\hat{\mathbf{a}}_n \\ &= \mathbf{R}_m^n\mathbf{d}_m^{n-1} + p_{mn}\left(x_{mn} - \hat{\mathbf{a}}_n^T\mathbf{d}_m^{n-1}\right)\hat{\mathbf{a}}_n,\end{aligned} \tag{26}$$

where $\mathbf{d}_m^{n-1}$ is the estimate for row m at time $n-1$. Hence

$$\mathbf{d}_m^n = \mathbf{d}_m^{n-1} + p_{mn}\left(x_{mn} - \hat{\mathbf{a}}_n^T \mathbf{d}_m^{n-1}\right)\left(\mathbf{R}_m^n\right)^{\dagger}\hat{\mathbf{a}}_n. \tag{27}$$

defines a recursive procedure for updating all rows of the matrix $\mathbf{D}_n$ in parallel.

Finally we note that the matrix $(\mathbf{R}_m^n)^{\dagger}$ can be updated without recourse to matrix inversion. We apply the Recursive Least Squares (RLS) updating formula for the general pseudo-inverse matrix [26], [27] to obtain

$$\begin{aligned}\left(\mathbf{R}_m^n\right)^{\dagger} &= \left(\lambda \mathbf{R}_m^{n-1} + p_{mn}\hat{\mathbf{a}}_n\hat{\mathbf{a}}_n^T\right)^{\dagger}\\ &= \lambda^{-1}\left(\mathbf{R}_m^{n-1}\right)^{\dagger} - p_{mn}\mathbf{G}_m^n.\end{aligned} \tag{28}$$

Here $\mathbf{G}_m^n = (\beta_m^n)^{-1}\mathbf{v}_m^n(\mathbf{v}_m^n)^T$, with β_m^n and $\mathbf{v}_m^n$ given as

$$\begin{aligned}\beta_m^n &= 1 + \lambda^{-1}\hat{\mathbf{a}}_n^T\left(\mathbf{R}_m^{n-1}\right)^{\dagger}\hat{\mathbf{a}}_n,\\ \mathbf{v}_m^n &= \lambda^{-1}\left(\mathbf{R}_m^{n-1}\right)^{\dagger}\hat{\mathbf{a}}_n.\end{aligned}$$

In RLS updating, the diagonal entries of the initial matrix $(\mathbf{R}_m^0)^{\dagger}$ are required to be large and for all $m = 1, \ldots, M$ we set $(\mathbf{R}_m^0)^{\dagger} = \delta\mathbf{I}_r$, $\delta > 0$. RLS updating is in general very efficient but care needs to be taken as finite precision operations suffer from numerical instability when running for a long time [28].

C. Second-Order Stochastic Gradient Descent

The PETRELS algorithm can be regarded as a second-order stochastic gradient descent method to solve (14) by using $\mathbf{d}_m^{n-1}$, $m = 1, \cdots, M$ as a warm start at time n. Specifically, we can write the gradient of $f_n(\mathbf{D})$ in (13) at $\mathbf{D}_{n-1}$ as

$$\left.\frac{\partial f_n(\mathbf{D})}{\partial \mathbf{D}}\right|_{\mathbf{D}=\mathbf{D}_{n-1}} = -2\mathbf{P}_n(\mathbf{x}_n - \mathbf{D}_{n-1}\hat{\mathbf{a}}_n)\hat{\mathbf{a}}_n^T, \tag{29}$$

where $\hat{\mathbf{a}}_n$ is given in (18). Then the gradient of $F_n(\mathbf{D})$ at $\mathbf{D}_{n-1}$ is given as

$$\left.\frac{\partial F_n(\mathbf{D})}{\partial \mathbf{D}}\right|_{\mathbf{D}=\mathbf{D}_{n-1}} = -2\sum_{t=1}^{n}\lambda^{n-t}\mathbf{P}_t(\mathbf{x}_t - \mathbf{D}_{n-1}\hat{\mathbf{a}}_t)\hat{\mathbf{a}}_t^T.$$

The Hessian for each row of $\mathbf{D}$ at $\mathbf{d}_m^{n-1}$ is therefore

$$\begin{aligned}\mathbf{H}_n\left(\mathbf{d}_m^{n-1}, \lambda\right) &= \left.\frac{\partial^2 F_n(\mathbf{D})}{\partial \mathbf{d}_m \partial \mathbf{d}_m^T}\right|_{\mathbf{d}_m = \mathbf{d}_m^{n-1}}\\ &= 2\sum_{t=1}^{n}\lambda^{n-t}p_{mt}\hat{\mathbf{a}}_t\hat{\mathbf{a}}_t^T.\end{aligned} \tag{30}$$

It follows that the update rule for each row $\mathbf{d}_m$ can be written as

$$\mathbf{d}_m^n = \mathbf{d}_m^{n-1} - \mathbf{H}_n^{-1}\left(\mathbf{d}_m^{n-1}, \lambda\right)\frac{\partial f_n(\mathbf{D})}{\partial \mathbf{d}_m^{n-1}}, \tag{31}$$

which is equivalent to second-order stochastic gradient descent. Therefore, PETRELS converges to a stationary point of $F_n(\mathbf{D})$ [29], [30]. Compared with first-order algorithms, PETRELS enjoys a faster convergence speed to the stationary point [29], [30].

D. Comparison With GROUSE

GROUSE is an algorithm proposed by Balzano *et al.* [19] for online identification of a low-rank subspace from highly incomplete observations. It does not discount prior observations and can be viewed as optimizing (14) for $\lambda = 1$. In fact GROUSE aims to solve the following optimization problem:

$$\mathbf{D}_n = \arg\min_{\mathbf{D}\in\mathcal{G}_r} G_n(\mathbf{D}) = \arg\min_{\mathbf{D}\in\mathcal{G}_r}\sum_{t=1}^{n} f_t(\mathbf{D}), \tag{32}$$

where $\mathcal{G}_r = \{\mathbf{D} \in \mathbb{R}^{M\times r} : \mathbf{D}^T\mathbf{D} = \mathbf{I}_r\}$ is the orthogonal Grassmannian rather than $\mathbb{R}^{M\times r}$.

The GROUSE algorithm performs *first-order* stochastic gradient descent on the orthogonal Grassmannian. It updates the subspace estimate along the direction of $\nabla f_n(\mathbf{D})|_{\mathbf{D}=\mathbf{D}_{n-1}}$ on $\mathcal{G}_r$, given by

$$\mathbf{D}_n = \mathbf{D}_{n-1} - \left[(\cos(\sigma\eta_n) - 1)\frac{\hat{\mathbf{x}}_n}{\|\hat{\mathbf{x}}_n\|_2} + \sin(\sigma\eta_n)\frac{\mathbf{r}_n}{\|\mathbf{r}_n\|_2}\right]\frac{\hat{\mathbf{a}}_n^T}{\|\hat{\mathbf{a}}_n\|_2}, \tag{33}$$

where $\sigma = \|\hat{\mathbf{x}}_n\|_2\|\mathbf{r}_n\|_2$ with $\hat{\mathbf{x}}_n$ given in (19), $\mathbf{r}_n = \mathbf{P}_n(\mathbf{x}_n - \hat{\mathbf{x}}_n)$, and η_n is the step-size at time n.

At time n GROUSE provides a fast rank one update of $\mathbf{D}_{n-1}$ by alternating between coefficient estimation (18) and subspace estimation (33). Since GROUSE is a first order gradient descent algorithm, given weak conditions on the step size, specifically

$$\lim_{t\to\infty}\eta_t = 0 \quad \text{and} \quad \sum_{t=1}^{\infty}\eta_t = \infty, \tag{34}$$

it is guaranteed to converge to a stationary point on $G_n(\mathbf{D})$. However this stationary point may not be a global optimum because of barriers in the search path on the Grassmannian [20]. Estimation of direction-of-arrival in Section VI provides an example where GROUSE is trapped at a local minima.

The performance of GROUSE depends strongly on proper tuning of the step size to satisfy (34). The performance of PETRELS depends on the discount factor λ, but without any tuning ($\lambda = 1$) it can still converge to the global optimum when the data is fully observed (see Section IV).

If we relax the objective function of GROUSE (32) to all rank-r subspaces $\mathbb{R}^{M\times r}$ by setting

$$\mathbf{D}_n = \arg\min_{\mathbf{D}\in\mathbb{R}^{M\times r}}\sum_{t=1}^{n} f_t(\mathbf{D}), \tag{35}$$

then the objective function becomes equivalent to PETRELS without discounting. It is then possible to solve (35) by second order stochastic gradient descent with an appropriate step size. The update rule for each row of $\mathbf{D}_n$ is then

$$\mathbf{d}_m^n = \mathbf{d}_m^{n-1} - \gamma_n\mathbf{H}_n^{-1}\left(\mathbf{d}_m^{n-1}, \lambda = 1\right)\frac{\partial f_n(\mathbf{D})}{\partial \mathbf{d}_m^{n-1}}, \tag{36}$$

where $\mathbf{H}_n(\mathbf{d}_m^{n-1}, \lambda = 1)$ is given in (30), and γ_n is the step-size at time n. In this paper we do not investigate the performance of PETRELS with this alternative update rule.

E. Complexity Issues

We now compare both storage complexity and computational complexity for PETRELS, GROUSE and the PAST algorithm. The storage complexity of PAST and GROUSE is $\mathcal{O}(Mr)$, which is the size of the low-rank subspace. On the other hand, PETRELS has a larger storage complexity of $\mathcal{O}(Mr^2)$, which is the total size of $\mathbf{R}_m^n$'s for each row. In terms of computational complexity, PAST has a complexity of $\mathcal{O}(Mr)$ per iteration, while PETRELS and GROUSE have a similar complexity on the order of $\mathcal{O}(|\Omega_n|r^3)$, where the main contribution to complexity comes from computation of the coefficient (18). Parallel implementation reduces the computational complexity of PETRELS to $\mathcal{O}(r^3)$. We note that partial observation can be used to reduce computational complexity when the ambient dimension is high.

IV. Global Convergence With Full Observations

PETRELS is a second order stochastic gradient descent, hence even when data is only partially observed it converges to a stationary point of $F_n(\mathbf{D})$. In general, convergence to a global optimum remains open. In this section we show convergence to a global optimum for the fully observed setting.

When data is fully observed and past observations are not discounted $(\lambda = 1)$ PETRELS is essentially equivalent to PAST [7] though the two algorithms differ in the method of estimating coefficients. With the notation of Section II-B, PAST forms the estimate $\hat{\mathbf{a}}_t = \mathbf{W}_{t-1}^T\mathbf{y}_t = \mathbf{W}_{t-1}^T\mathbf{x}_t$ whereas PETRELS forms the estimate $\hat{\mathbf{a}}_t = (\mathbf{D}_{t-1}^T\mathbf{D}_{t-1})^{-1}\mathbf{D}_{t-1}^T\mathbf{x}_t$.

Ljung [31] describes how Ordinary Differential Equations (ODEs) may be used to analyze stochastic recursive algorithms. This method is applied to PAST in [32] where asymptotic convergence in continuous time follows from the equilibrium behavior of the ODE:

$$\dot{\mathbf{R}} = \mathbb{E}\left[\tilde{\mathbf{a}}_n\tilde{\mathbf{a}}_n^T\right] - \mathbf{R} = \mathbf{W}^T\mathbf{C}_\mathbf{x}\mathbf{W} - \mathbf{R}, \tag{37}$$

$$\dot{\mathbf{W}} = \mathbb{E}\left[\mathbf{x}_n(\mathbf{x}_n - \mathbf{W}\tilde{\mathbf{a}}_n)^T\right]\mathbf{R}^\dagger = (\mathbf{I} - \mathbf{W}\mathbf{W}^T)\mathbf{C}_\mathbf{x}\mathbf{W}\mathbf{R}^\dagger, \tag{38}$$

where $\tilde{\mathbf{a}}_n = \mathbf{W}^T\mathbf{x}_n$, $\mathbf{R} = \mathbf{R}(t)$ and $\mathbf{W} = \mathbf{W}(t)$ are continuous time versions of $\mathbf{R}_n = \sum_{t=1}^n \hat{\mathbf{a}}_t\hat{\mathbf{a}}_t^T$ and $\mathbf{W}_n$, and the derivatives are taken with respect to t. It is proved in [32] that as t increases, $\mathbf{W}(t)$ converges to the global optima, i.e. to a matrix which spans the eigenvectors of $\mathbf{C}_\mathbf{x}$ corresponding to the r largest eigenvalues.

The asymptotic dynamics of the PETRELS algorithm are described by the following ODE:

$$\begin{aligned}\dot{\mathbf{R}} &= \mathbb{E}\left[\tilde{\mathbf{a}}_n\tilde{\mathbf{a}}_n^T\right] - \mathbf{R}\\ &= (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{C}_\mathbf{x}\mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1} - \mathbf{R},\end{aligned} \tag{39}$$

$$\begin{aligned}\dot{\mathbf{D}} &= \mathbb{E}\left[\mathbf{x}_n(\mathbf{x}_n - \mathbf{D}\tilde{\mathbf{a}}_n)^T\right]\mathbf{R}^\dagger\\ &= \left(\mathbf{I} - \mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\right)\mathbf{C}_\mathbf{x}\mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{R}^{-1}.\end{aligned} \tag{40}$$

Here $\tilde{\mathbf{a}}_n = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{x}_n$, $\mathbf{R} = \mathbf{R}(t)$ and $\mathbf{D} = \mathbf{D}(t)$ are continuous-time versions of $\mathbf{R}_n$ and $\mathbf{D}_n$. Now let $\widetilde{\mathbf{D}} = \mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1/2}$ and $\widetilde{\mathbf{R}} = (\mathbf{D}^T\mathbf{D})^{1/2}\mathbf{R}(\mathbf{D}^T\mathbf{D})^{1/2}$. From (40),

$$\begin{aligned}\mathbf{D}^T\dot{\mathbf{D}} &= \mathbf{D}^T\left(\mathbf{I} - \mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\right)\mathbf{C}_\mathbf{x}\mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{R}^{-1}\\ &= \mathbf{D}^T\mathbf{C}_\mathbf{x}\mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{R}^{-1} - \mathbf{D}^T\mathbf{C}_\mathbf{x}\mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{R}^{-1}\\ &= \mathbf{0},\end{aligned}$$

and

$$\frac{d}{dt}(\mathbf{D}^T\mathbf{D}) = \mathbf{D}^T\dot{\mathbf{D}} + \dot{\mathbf{D}}^T\mathbf{D} = \mathbf{0}.$$

Furthermore

$$\frac{d}{dt}f(\mathbf{D}^T\mathbf{D}) = \mathbf{0}$$

for any function of $\mathbf{D}^T\mathbf{D}$. Hence,

$$\dot{\widetilde{\mathbf{D}}} = \dot{\mathbf{D}}(\mathbf{D}^T\mathbf{D})^{-1/2} + \mathbf{D}\frac{d}{dt}(\mathbf{D}^T\mathbf{D})^{-1/2} = \dot{\mathbf{D}}(\mathbf{D}^T\mathbf{D})^{-1/2},$$

and

$$\begin{aligned}\dot{\widetilde{\mathbf{R}}} &= \frac{d}{dt}(\mathbf{D}^T\mathbf{D})^{-1/2}\mathbf{R}(\mathbf{D}^T\mathbf{D})^{\frac{1}{2}} + (\mathbf{D}^T\mathbf{D})^{\frac{1}{2}}\dot{\mathbf{R}}(\mathbf{D}^T\mathbf{D})^{\frac{1}{2}}\\ &\quad + (\mathbf{D}^T\mathbf{D})^{\frac{1}{2}}\mathbf{R}\frac{d}{dt}(\mathbf{D}^T\mathbf{D})^{\frac{1}{2}}\\ &= (\mathbf{D}^T\mathbf{D})^{\frac{1}{2}}\dot{\mathbf{R}}(\mathbf{D}^T\mathbf{D})^{\frac{1}{2}}.\end{aligned}$$

Therefore (39) and (40) can be rewritten as

$$\begin{aligned}\dot{\widetilde{\mathbf{R}}} &= \widetilde{\mathbf{D}}^T\mathbf{C}_\mathbf{x}\widetilde{\mathbf{D}} - \widetilde{\mathbf{R}},\\ \dot{\widetilde{\mathbf{D}}} &= (\mathbf{I} - \widetilde{\mathbf{D}}\widetilde{\mathbf{D}}^T)\mathbf{C}_\mathbf{x}\widetilde{\mathbf{D}}\widetilde{\mathbf{R}}^\dagger,\end{aligned}$$

which is equivalent to the ODE given in (37) and (38) that describes PAST. Hence the subspace estimate derived by PETRELS converges asymptotically to the same global optimum, that is to the rank-r principal subspace of $\mathbf{C}_\mathbf{x}$, with the same dynamics as the PAST algorithm.

V. Extensions to the PETRELS Algorithm

A. Simplified Update Rule

If we remove the partial observation operator from the objective function (20) then we obtain

$$\mathbf{D}_n = \arg\min_{\mathbf{D}} \hat{F}_n(\mathbf{D}) = \arg\min_{\mathbf{D}} \sum_{t=1}^n \lambda^{n-t}\|\hat{\mathbf{x}}_t - \mathbf{D}\hat{\mathbf{a}}_t\|_2^2, \tag{41}$$

where $\hat{\mathbf{a}}_t$ and $\hat{\mathbf{x}}_t$, $t = 1, \cdots, n$ are estimates from earlier steps calculated as in (18) and (19). It remains true that $\mathbf{d}_m^n = \arg\min_{\mathbf{d}_m} \hat{F}_n(\mathbf{d}_m) = \mathbf{d}_m^{n-1}$ if the corresponding mth entry of $\mathbf{x}_n$ is unobserved, i.e. $m \notin \Omega_n$. Indeed,

$$\begin{aligned}\hat{F}_n(\mathbf{d}_m) &= \sum_{t=1}^{n-1}\lambda^{n-t}\left\|\hat{x}_{mt} - \mathbf{d}_m^T\hat{\mathbf{a}}_t\right\|_2^2 + \left\|\left(\mathbf{d}_m^{n-1} - \mathbf{d}_m\right)^T\hat{\mathbf{a}}_t\right\|_2^2,\\ &= \lambda\hat{F}_{n-1}(\mathbf{d}_m) + \left\|\left(\mathbf{d}_m^{n-1} - \mathbf{d}_m\right)^T\hat{\mathbf{a}}_t\right\|_2^2\\ &\geq \lambda\hat{F}_{n-1}\left(\mathbf{d}_m^{n-1}\right) = \hat{F}_n\left(\mathbf{d}_m^{n-1}\right).\end{aligned}$$

The minimum is therefore obtained when $\mathbf{d}_m = \mathbf{d}_m^{n-1}$ for $m \notin \Omega_n$.

This modification leads to a simplified update rule for $\mathbf{R}_m^n$, since now the updating formula for all rows is identical as $\mathbf{R}_m^n = \mathbf{R}_n = \lambda\mathbf{R}_{n-1} + \hat{\mathbf{a}}_n\hat{\mathbf{a}}_n^T$ for all m. Hence the row update formula (27) is replaced by

$$\mathbf{D}_n = \mathbf{D}_{n-1} + \mathbf{P}_n(\mathbf{x}_n - \mathbf{D}_{n-1}\hat{\mathbf{a}}_n)\hat{\mathbf{a}}_n^T\mathbf{R}_n^\dagger, \tag{42}$$

which further reduces the storage required by PETRELS from $\mathcal{O}(Mr^2)$, to $\mathcal{O}(Mr)$, i.e. the size of the subspace. Numerical examples in Section VI suggest that this simplification leads to slower convergence, but that it may still have an advantage if the subspace rank is underestimated.

B. Incorporating Prior Information

It is possible to incorporate regularization terms into PETRELS to encode prior information about the data stream. In Section II-C, the data stream is partially observed columns drawn from a low rank matrix, and the low rank prior is encoded in (10) using the nuclear norm. In this Section we suppose that at time n the subspace $\mathbf{D}_n$ is updated according to

$$\mathbf{D}_n = \arg\min_{\mathbf{D}} \sum_{t=1}^{n} \lambda^{n-t} \left\|\mathbf{P}_t(\mathbf{x}_t - \mathbf{D}\hat{\mathbf{a}}_t)\right\|_2^2 + \mu_n \|\mathbf{D}\|_F^2, \quad (43)$$

where $\mu_n > 0$ is the regularization parameter. It follows from the analysis given in Section III-B that (43) decomposes into M parallel problems, one for each row of $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_M]^T$ as

$$\begin{aligned}\mathbf{d}_m^n &= \arg\min_{\mathbf{d}_m} \sum_{t=1}^{n} \lambda^{n-t} p_{mt} \left(x_{mt} - \hat{\mathbf{a}}_t^T \mathbf{d}_m\right)^2 + \mu_n \|\mathbf{d}_m\|_2^2 \\ &= \left(\sum_{t=1}^{n} \lambda^{n-t} p_{mt} \hat{\mathbf{a}}_t \hat{\mathbf{a}}_t^T + \mu_n \mathbf{I}\right)^{-1} \left(\sum_{t=1}^{n} \lambda^{n-t} p_{mt} x_{mt} \hat{\mathbf{a}}_t\right) \\ &= \left(\mathbf{T}_m^n\right)^{-1} \mathbf{s}_m^n.\end{aligned}$$

The matrix $\mathbf{T}_m^n$ can be updated as

$$\mathbf{T}_m^n = \lambda \mathbf{T}_m^{n-1} + p_{mn} \hat{\mathbf{a}}_t \hat{\mathbf{a}}_t^T + (\mu_n - \lambda \mu_{n-1}) \mathbf{I}_r,$$

and $\mathbf{s}_m^n$ can be updated as in (25).

C. Extension to Compressive Measurements

Until now, we have focused on direct observation of data. However, it is straightforward to modify PETRELS to handle compressive measurements in which the observation at time n is given by

$$\tilde{\mathbf{y}}_n = \mathbf{\Phi}_n \mathbf{x}_n, \quad (44)$$

where $\tilde{\mathbf{y}}_n \in \mathbb{R}^{|\Omega_n|}$, and $\mathbf{\Phi}_n \in \mathbb{R}^{|\Omega_n| \times M}$ is the measurement matrix. We estimate and track the underlying subspace from $\{\tilde{\mathbf{y}}_t, \mathbf{\Phi}_t\}_{t=1}^{\infty}$ by alternating between coefficient updates and subspace estimates. At time n, given the subspace $\mathbf{D}_{n-1}$ estimated at time $n-1$, we first estimate the coefficient vector $\hat{\mathbf{a}}_n$ as

$$\begin{aligned}\hat{\mathbf{a}}_n &= \min_{\mathbf{a}} \|\tilde{\mathbf{y}}_n - \mathbf{\Phi}_n \mathbf{D}_{n-1} \mathbf{a}\|_2^2 \\ &= \left(\mathbf{D}_{n-1}^T \mathbf{\Phi}_n^T \mathbf{\Phi}_n \mathbf{D}_{n-1}\right)^{\dagger} \mathbf{D}_{n-1}^T \mathbf{\Phi}_n^T \tilde{\mathbf{y}}_n, \quad (45)\end{aligned}$$

and then update the subspace by

$$\mathbf{D}_n = \min_{\mathbf{D}} \sum_{t=1}^{n} \lambda^{n-t} \|\tilde{\mathbf{y}}_t - \mathbf{\Phi}_t \mathbf{D} \hat{\mathbf{a}}_t\|_2^2. \quad (46)$$

Partial observation is a special case of compressive measurement where the matrices $\mathbf{\Phi}_n$ are partial identity matrices. It is not possible to parallelize updates for general measurement matrices, but it is still possible to update subspaces recursively. To see this, let $\mathbf{d} = \mathrm{vec}(\mathbf{D})$, and $\mathbf{d}^n = \mathrm{vec}(\mathbf{D}_n)$, where $\mathrm{vec}(\cdot)$ denotes column-wise vectorization. Note that

$$\mathbf{\Phi}_n \mathbf{D} \hat{\mathbf{a}}_n = \left(\hat{\mathbf{a}}_n^T \otimes \mathbf{\Phi}_n\right) \mathbf{d} \triangleq \mathbf{\Psi}_n \mathbf{d},$$

where $\otimes$ denotes the Kronecker product. We rewrite (46) as

$$\mathbf{d}^n = \min_{\mathbf{d}} \sum_{t=1}^{n} \lambda^{n-t} \|\tilde{\mathbf{y}}_t - \mathbf{\Psi}_t \mathbf{d}\|_2^2 = (\mathbf{R}^n)^{\dagger} \mathbf{s}^n, \quad (47)$$

where $\mathbf{R}^n = \sum_{t=1}^{n} \lambda^{n-t} \mathbf{\Psi}_t^T \mathbf{\Psi}_t$, and $\mathbf{s}^n = \sum_{t=1}^{n} \lambda^{n-t} \mathbf{\Psi}_t^T \tilde{\mathbf{y}}_t$. We now use the Woodbury matrix identity [26] to recursively update $(\mathbf{R}^n)^{\dagger}$ from $(\mathbf{R}^{n-1})^{\dagger}$ as earlier, so that (47) becomes

$$\mathbf{d}^n = \mathbf{d}^{n-1} + (\mathbf{R}^n)^{\dagger} \mathbf{\Psi}_n^T \mathbf{r}_n,$$

where $\mathbf{r}_n = \tilde{\mathbf{y}}_n - \mathbf{\Psi}_n \mathbf{d}^{n-1}$ is the projection residual at time n. Note that at time n, the new update rule involves inversion of a matrix of size $|\Omega_n| r$.

VI. Numerical Results

Our numerical results contain four parts. First we examine the influence of parameters specified in the PETRELS algorithm, such as discount factor, rank estimation, and its robustness to noise level. Next we look at the problem of direction-of-arrival estimation and show that PETRELS demonstrates performance superior to GROUSE by identifying and tracking all the targets almost perfectly even in low SNR. Thirdly, we compare our approach with MC, and show that PETRELS is at least competitive with state of the art batch algorithms. Finally, we provide numerical simulations for the extensions of the PETRELS algorithm.

A. Choice of Parameters

At each time n, a vector $\mathbf{x}_n$ is generated as

$$\mathbf{x}_n = \mathbf{D}_{true} \mathbf{a}_n + \mathbf{n}_n, \quad (48)$$

where $\mathbf{D}_{true}$ is an r-dimensional subspace generated with i.i.d. $\mathcal{N}(0,1)$ entries, $\mathbf{a}_n$ is an $r \times 1$ vector with i.i.d. $\mathcal{N}(0,1)$ entries, and $\mathbf{n}_n$ is an $M \times 1$ Gaussian noise vector with i.i.d. $\mathcal{N}(0, \epsilon^2)$ entries. We further fix the signal dimension $M = 500$ and the subspace rank $r_{true} = 10$. We assume that a fixed number of entries in $\mathbf{x}_n$, denoted by K, are revealed each time. This restriction is not necessary for PETRELS to work as is shown in the MC simulations, but we make it here in order to get a meaningful estimate of $\mathbf{a}_n$. Denoting the estimated subspace at time n by $\mathbf{D}_n$, we use the normalized subspace reconstruction error to examine the algorithm performance. This is calculated as $\|\mathcal{P}_{\mathbf{D}_n^{\perp}} \mathbf{D}_{true}\|_F^2 / \|\mathbf{D}_{true}\|_F^2$, where $\mathcal{P}_{\mathbf{D}_n^{\perp}}$ is the projection operator onto the orthogonal complement of $\mathbf{D}_n$.

The choice of discount factor λ plays an important role in how fast the algorithm converges. We assume $K = 50$, so that only 10% of the entries are observed, and the rank is given accurately as $r = 10$ in a noise-free setting where $\epsilon = 0$. We run the algorithm to time $n = 2000$, and find that the normalized subspace reconstruction error of the above data is minimized when λ is around 0.98 as shown in Fig. 1. Hence, we will keep $\lambda = 0.98$ hereafter.

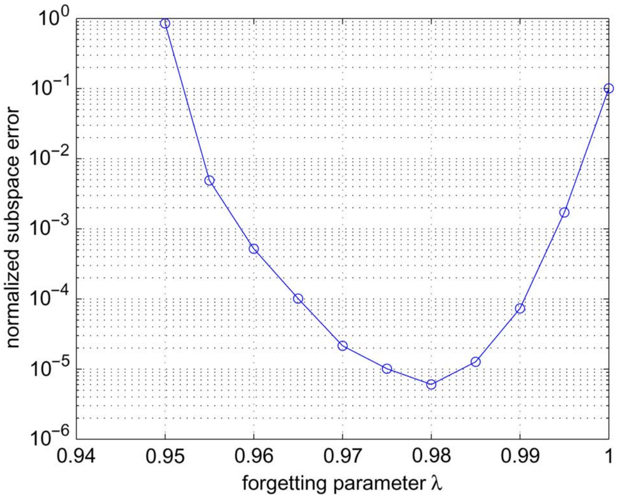


Fig. 1. The normalized subspace reconstruction error as a function of the discount factor λ after running the algorithm to time $n = 2000$ when 50 out of 500 entries of the signal are observed each time without noise.

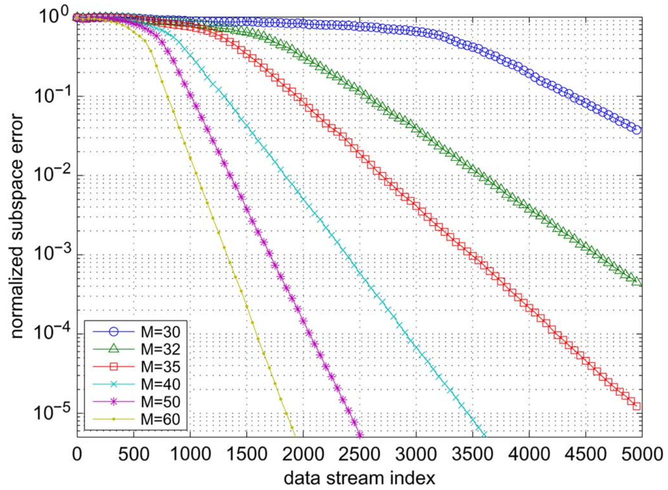


Fig. 3. Normalized subspace reconstruction error as a function of data stream index when the number of entries observed per time M out of 500 entries are varied with accurate rank estimation and no noise.

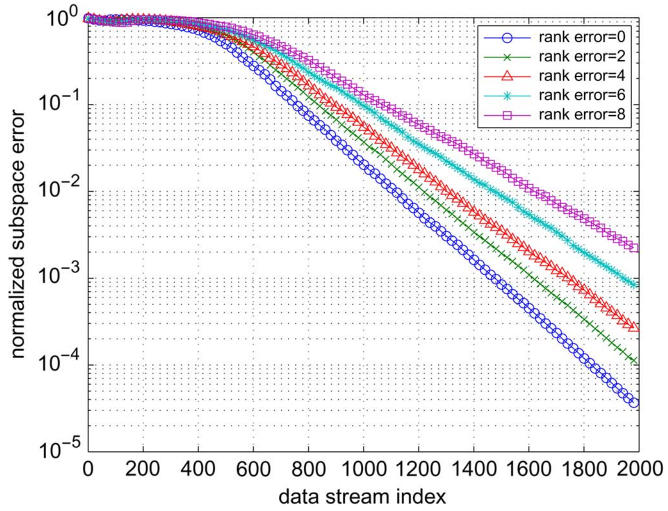


Fig. 2. Normalized subspace reconstruction error as a function of data stream index when the rank is over-estimated when 50 out of 500 entries of the signal are observed each time without noise.

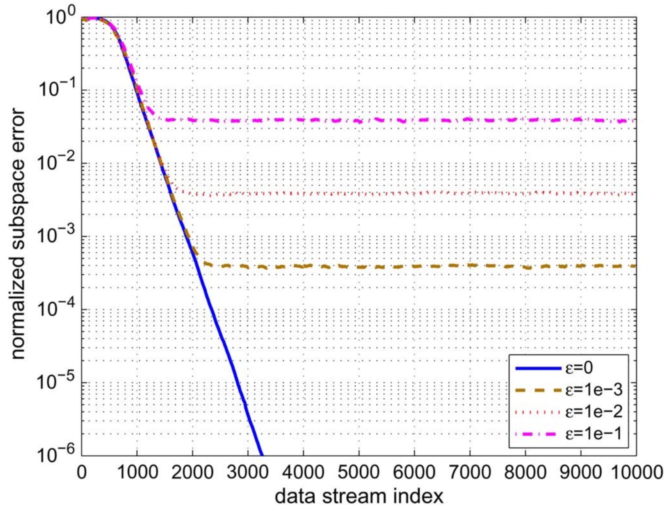


Fig. 4. Normalized subspace error versus data stream index with different noise level ϵ when 50 out of 500 entries of the signal are observed each time with accurate rank estimation.

In reality it is almost impossible to accurately estimate the intrinsic rank in advance. Fortunately the convergence rate of our algorithm degrades gracefully as the rank estimation error increases. In Fig. 2, the evolution of normalized subspace reconstruction error is plotted against data stream index, for rank estimation $r = 10$, 12, 14, 16, 18. We only examine over-estimation of the rank here since we can easily make it the case in applications. In the next section we show examples for the case of rank under-estimation.

Taking more measurements per time leads to faster convergence, as shown in Fig. 3. Theoretically it requires $M \sim \mathcal{O}(r \log r) \approx 23$ measurements to test if an incomplete vector is within a subspace of rank r [17]. The simulation shows our algorithm can work even when M is close to this lower bound.

Finally, the robustness of PETRELS is tested against the noise variance ϵ^2 in Fig. 4, where the normalized subspace reconstruction error is plotted as a function of the data stream index for different noise levels. The estimated subspace deviates from the ground truth as we increase the noise level, hence the normalized subspace error degrades gracefully and converges to an error floor determined by the noise variance.

We now consider a scenario where a subspace of rank $r = 10$ changes abruptly at time index $n = 3000$ and $n = 5000$, and examine the performance of GROUSE [19] and PETRELS in Fig. 5 when the rank is over-estimated by 4 and the noise level is $\epsilon = 10^{-3}$. The normalized residual error for the data stream, calculated as $\|\mathbf{P}_n(\mathbf{x}_n - \hat{\mathbf{x}}_n)\|_2 / \|\mathbf{P}_n \mathbf{x}_n\|_2$ at each time n, is shown in Fig. 5(a), and the normalized subspace error is shown in Fig. 5(b) respectively. Both PETRELS and GROUSE can successfully track the changed subspace, but PETRELS tracks the change faster.

B. Direction-of-Arrival Analysis

Given GROUSE [19] as a baseline, we evaluate the resilience of PETRELS to different data models and applications. We use

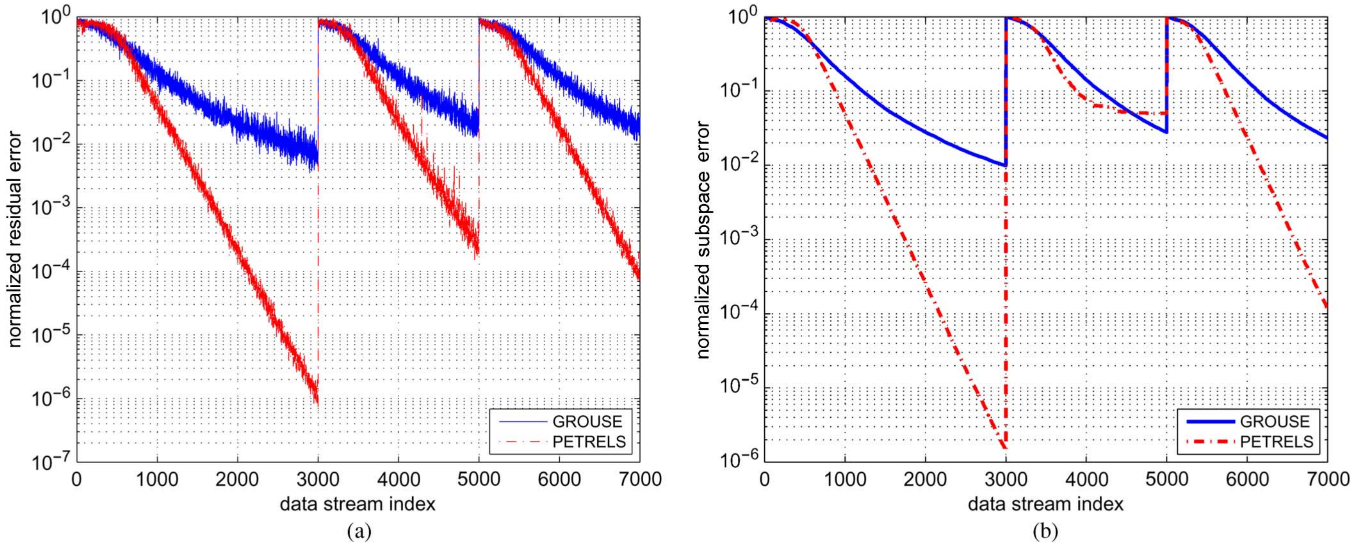


Fig. 5. Tracking a subspace with fixed rank $r = 10$. The rank is over-estimated by 4, the noise level is $\epsilon = 10^{-3}$, and 50 out of 500 entries of the signal are observed each time for both GROUSE and PETRELS. (a) Normalized residual error. (b) Normalized subspace error.

the following example of direction-of-arrival analysis in array processing to compare the performance of these two methods. Assume there are $M = 256$ sensors from a linear array, and the measurements from all sensors at time n are given as

$$\mathbf{x}_n = \mathbf{V}\boldsymbol{\Sigma}\mathbf{a}_n + \mathbf{n}_n. \tag{49}$$

Here $\mathbf{V} \in \mathbb{C}^{M \times p}$ is a Vandermonde matrix given by

$$\mathbf{V} = [\boldsymbol{\alpha}(\omega_1), \cdots, \boldsymbol{\alpha}(\omega_p)], \tag{50}$$

where $\boldsymbol{\alpha}(\omega_i) = [1, e^{j2\pi\omega_i}, \cdots, e^{j2\pi\omega_i(M-1)}]^T$, with $0 \le \omega_i < 1$, and $\boldsymbol{\Sigma} = \mathrm{diag}\{\mathbf{d}\} = \mathrm{diag}\{d_1, \cdots, d_p\}$ is a diagonal matrix which characterizes the amplitudes of each mode. The coefficients $\mathbf{a}_n$'s are generated with $\mathcal{N}(0,1)$ entries, and the noise is generated with $\mathcal{N}(0,\epsilon^2)$ entries, where $\epsilon = 0.1$.

At each time slot we collect measurements from $K = 30$ sensors uniformly at random. We are interested in identifying all $\{\omega_i\}_{i=1}^p$ and $\{d_i\}_{i=1}^p$. This can be done by applying the well-known ESPRIT algorithm [33] to the estimated subspace $\mathbf{D}_n$ of rank r at each time n, where r corresponds to the number of modes and can be estimated, for example via the Maximum Description Length (MDL) algorithm [34]. Specifically, let $\mathbf{D}_1$ and $\mathbf{D}_2$ be the submatrices of $\mathbf{D}_n$ with the first and the last $M-1$ rows of $\mathbf{D}_n$. The set of directions can be recovered from the eigenvalues of the matrix $\mathbf{T} = \mathbf{D}_1^\dagger \mathbf{D}_2$, denoted by λ_i, $i = 1, \cdots, r$, given as

$$\omega_i = \frac{1}{2\pi}\arg(\lambda_i), \quad i = 1, \cdots, r, \tag{51}$$

where $\arg(\lambda_i)$ is the phase of the complex number λ_i in $[0, 2\pi)$. The ESPRIT algorithm also plays a role in recovery of multi-path delays from low-rate samples of the channel output [35].

In a dynamic setting when the underlying subspace is varying, PETRELS is superior to GROUSE in terms of discarding out-of-date modes and picking up new modes. We divide the running time into 4 segments, with the frequencies and amplitudes in each segment specified as follows:

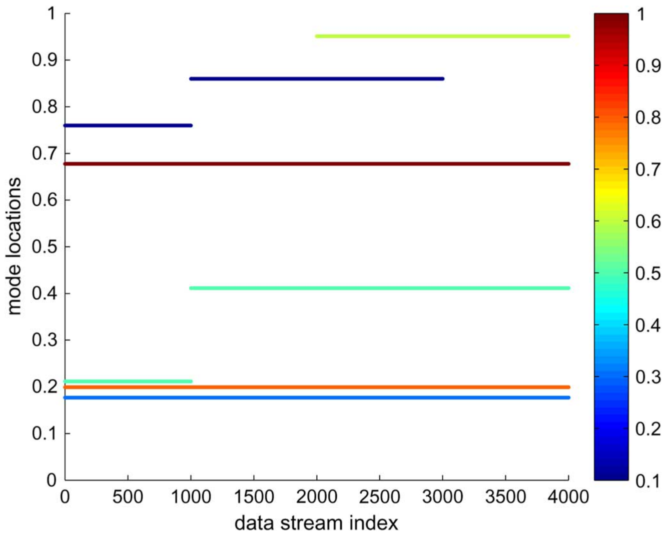


Fig. 6. Ground truth of the actual mode locations and amplitudes in a dynamic scenario.

1) Start with the same frequencies

$$\omega = [0.1769, 0.1992, 0.2116, 0.6776, 0.7599];$$

and amplitudes $d = [0.3, 0.8, 0.5, 1, 0.1]$.

2) Change two modes (only frequencies) at stream index 1000:

$$\omega = [0.1769, 0.1992, \mathbf{0.4116}, 0.6776, \mathbf{0.8599}];$$

and amplitudes $d = [0.3, 0.8, 0.5, 1, 0.1]$.

3) Add one new mode at stream index 2000:

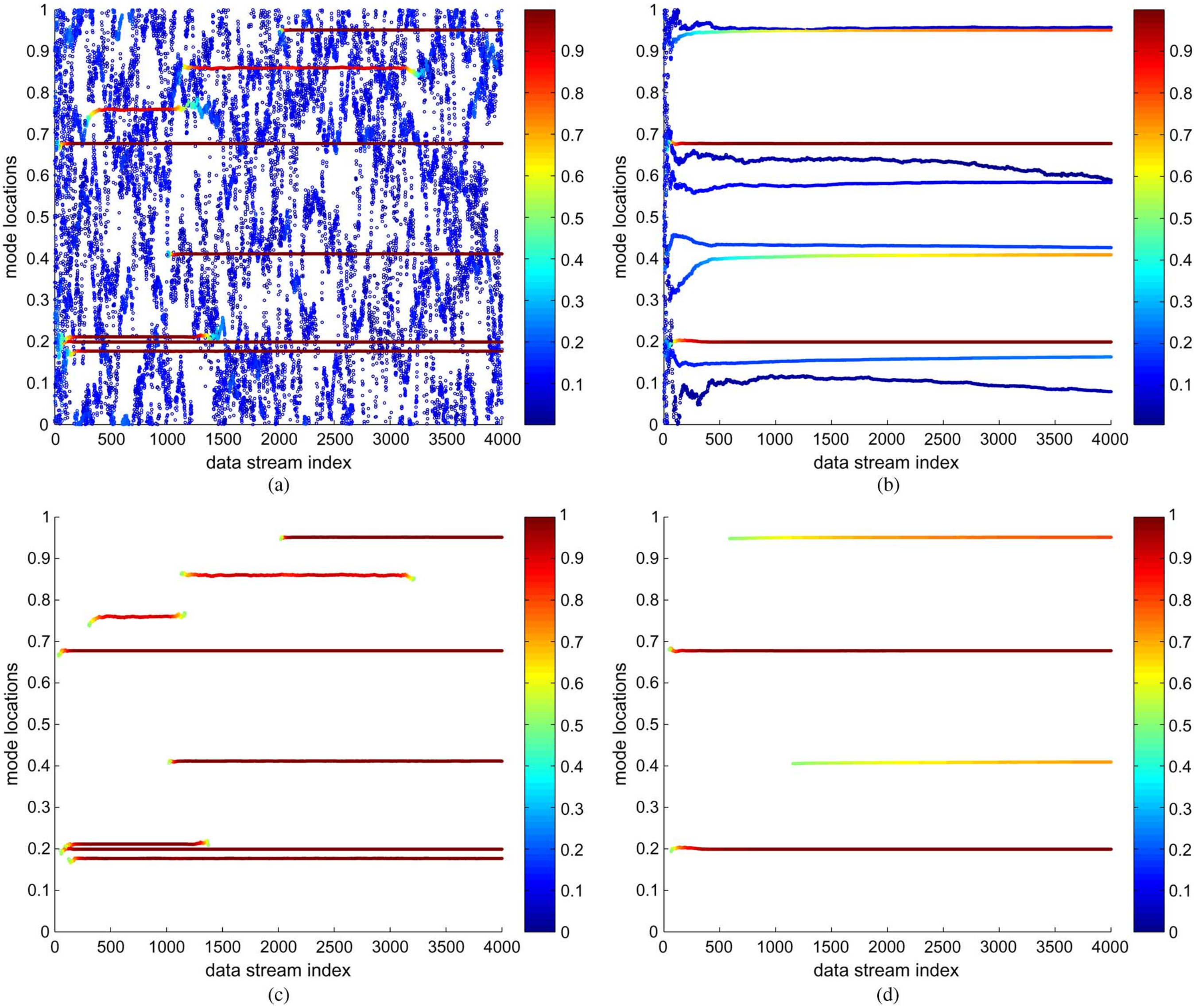
$$\omega = [0.1769, 0.1992, 0.4116, 0.6776, 0.8599, \mathbf{0.9513}];$$

and amplitudes $d = [0.3, 0.8, 0.5, 1, 0.1, \mathbf{0.6}]$.

4) Delete the weakest mode at stream index 3000:

$$\omega = [0.1769, 0.1992, 0.4116, 0.6776, 0.9513];$$

and amplitudes $d = [0.3, 0.8, 0.5, 1, 0.6]$.



(a)

(b)

(c)

(d)

Fig. 7. Tracking of mode changes in direction-of-arrival estimation using PETRELS and GROUSE algorithms: the estimated directions at each time for 10 modes are shown against the data stream in (a) and (b) for PETRELS and GROUSE respectively. The estimations in (a) and (b) are further thresholded with respect to level 0.5, and the thresholded results are shown in (c) and (d) respectively. All changes are identified and tracked successfully by PETRELS, but not by GROUSE. (a) Petrels; (b) Grouse; (c) Petrels (thresholded); (d) Grouse (thresholded).

Fig. 6 shows the ground truth of mode locations and amplitudes for the scenario above. Note that there are three closely located modes and one weak mode in the beginning, and various modes entering and exiting the scene, which makes the task challenging. We compare the performance of PETRELS and GROUSE in Fig. 7. The rank specified in both algorithms is $r = 10$, which is the number of estimated modes at each time index; in our case it is twice the number of the initial true modes.

The estimated mode locations and amplitudes of PETRELS and GROUSE are shown against the data stream index respectively in Fig. 7(a) and (b). The color shows the amplitude corresponding to the color bar. The direction-of-arrival estimations in Fig. 7(a) and (b) are further thresholded with respect to an amplitude level 0.5, and the thresholded results are shown in Fig. 7(c) and (d) for PETRELS and GROUSE respectively. PETRELS identifies all modes correctly. In particular, PETRELS distinguishes the three closely-spaced modes perfectly in the beginning, and identifies the weak modes that enter the scene at a later time. With GROUSE the closely spaced nodes are erroneously estimated as one mode, the weak mode is missing, and spurious modes have been introduced. PETRELS also fully tracked the later changes in accordance with the entrance and exit of each mode, while GROUSE is not able to react to changes in the data model.

Since the number of estimated modes at each time is greater than the number of true modes, the additional rank in the estimated subspace contributes “auxiliary modes” that do not belong to the data model. In PETRELS these modes become scatter points with small amplitudes as in Fig. 7(a), so they will not be identified as spurious targets in the scene. While in GROUSE, these auxiliary modes are tracked and appear as spurious modes as seen in Fig. 7(b).

C. Matrix Completion

We next compare performance of PETRELS for MC against batch algorithms including LMaFit [36], FPCA [37], Singular

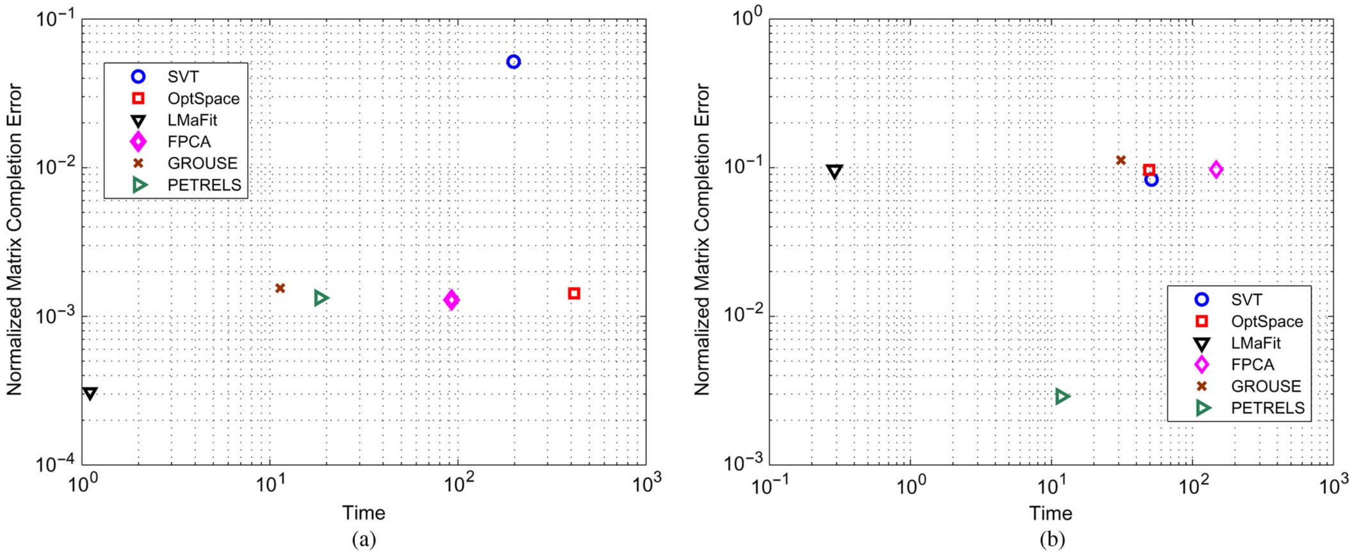


Fig. 8. Comparison of MC algorithms in terms of speed and accuracy: PETRELS is a competitive alternative for MC tasks when the low-rank matrix $\mathbf{X}$ is generated from a factorization model $\mathbf{X} = \mathbf{U}\mathbf{V}^T$ with the entries of $\mathbf{U} \in \mathbb{R}^{1000\times 10}$ and $\mathbf{V} \in \mathbb{R}^{2000\times 10}$ are from (a) $\mathcal{N}(0,1)$; and (b) $\mathcal{U}[0,1]$.

Value Thresholding (SVT) [38], OptSpace [15] and online GROUSE [19]. The low-rank matrix is generated from a matrix factorization model as $\mathbf{X} = \mathbf{U}\mathbf{V}^T \in \mathbb{R}^{1000\times 2000}$, where $\mathbf{U} \in \mathbb{R}^{1000\times 10}$ and $\mathbf{V} \in \mathbb{R}^{2000\times 10}$. The entries in $\mathbf{U}$ and $\mathbf{V}$ are generated from standard normal distribution $\mathcal{N}(0,1)$ (Gaussian data) or uniform distribution $\mathcal{U}[0,1]$ (uniform data). The sampling rate is taken to be 0.05, so only 5% of the entries of $\mathbf{X}$ are revealed.

The running time is plotted against the normalized matrix reconstruction error for Gaussian data and uniform data respectively in Fig. 8(a) and (b). The normalized matrix reconstruction error is calculated as $\|\hat{\mathbf{X}} - \mathbf{X}\|_F / \|\mathbf{X}\|_F$, where $\hat{\mathbf{X}}$ is the reconstructed low-rank matrix. PETRELS matches the performance of batch algorithms on Gaussian data and improves upon the accuracy of most algorithms on uniform data, where the Grassmaniann-based optimization approach may encounter "barriers" for its convergence. Note that different algorithms have different input parameter requirements. For example, OptSpace needs to specify the tolerance to terminate the iterations, which directly determines the trade-off between accuracy and running time; PETRELS and GROUSE require an initial estimate of the rank. Our simulation here only shows one particular realization and we simply conclude that PETRELS is competitive.

D. PETRELS Using Simplified Update Rule

We consider the same simulation setup as for Fig. 2, except that a subspace of rank 10 is generated by $\hat{\mathbf{D}}_{true} = \mathbf{D}_{true}\mathbf{\Sigma}$, where $\mathbf{\Sigma}$ is a diagonal matrix with 5 entries from $\mathcal{N}(0,1)$ and 5 entries from $0.01 \cdot \mathcal{N}(0,1)$. We examine the performance of the simplified PETRELS algorithm (with optimized $\lambda = 0.9$) proposed in Section V-A and the original PETRELS (with optimized $\lambda = 0.98$) algorithm. We consider both when the subspace rank is over-estimated as 12 and the rank is under-estimated as 8 where Fig. 9 shows the normalized subspace reconstruction error against data stream index. When the rank is over-estimated, the change in (9) will introduce more errors and converges slower compared with the original PETRELS algorithm; however, when the subspace rank is under-estimated, the simplified PETRELS performs better than PETRELS. This is an interesting feature of the proposed simplification, and quantitative justification of this phenomenon is beyond the scope of this paper. Intuitively, when the rank is under-estimated, the simplified PETRELS also uses the interpolated entries to update the subspace estimate, which seems to help the performance.

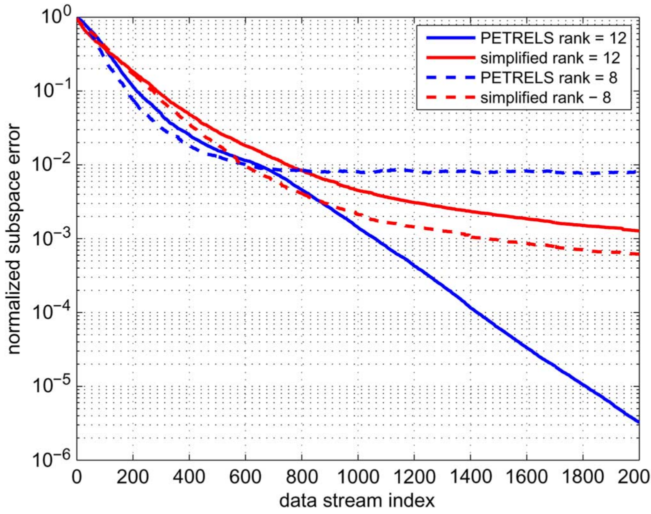


Fig. 9. Normalized subspace reconstruction error against data stream index when the rank is over-estimated as 12 or under-estimated as 8 for the original PETRELS and modified algorithm.

E. PETRELS With Compressive Measurements

We assume the data stream is generated using (48), where the subspace $\mathbf{D}_{true} \in \mathbb{R}^{100\times 10}$, and each time the data stream is measured using a matrix of size 20 × 100 with i.i.d. standard Gaussian entries. The underlying subspace is estimated via the modified PETRELS in Section V-C to handle compressive measurements. Fig. 10 shows the normalized subspace re-

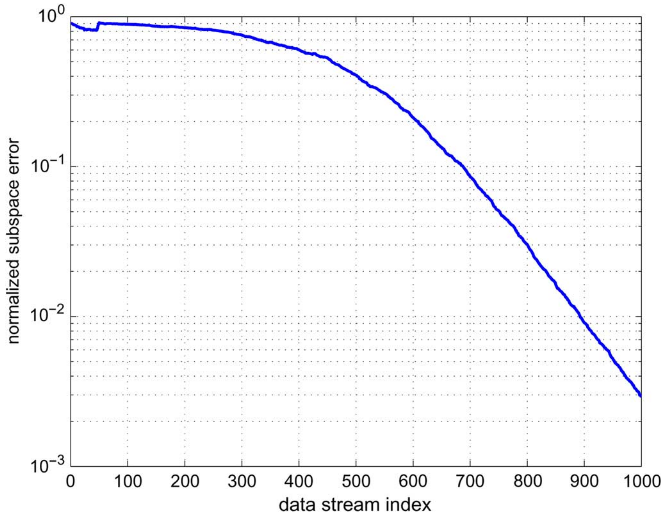


Fig. 10. Normalized subspace reconstruction error against data stream index when the size of the underlying subspace is 100×10, and 20 measurements are taken using a matrix of i.i.d. Gaussian entries at each time.

construction error against the data stream index with optimized $\lambda = 0.97$.

VII. Conclusions

We considered the problem of reconstructing a data stream from a small subset of its entries, where the data stream is assumed to lie in a low-dimensional linear subspace, possibly corrupted by noise. This has significant implications for lessening the storage burden and reducing complexity, as well as tracking the changes in the subspace for applications such as video denoising, network monitoring and anomaly detection when the problem size is large. The well-known low-rank MC problem can be viewed as a batch version of our problem. The PETRELS algorithm first identifies the underlying low-dimensional subspace via a discounted recursive procedure for each row of the subspace matrix in parallel, then reconstructs the missing entries via least-squares estimation if required. The discount factor allows the algorithm to capture long-term behavior as well as track the changes of the data stream. We have shown that PETRELS converges to a stationary point given it is a second-order stochastic gradient descent algorithm. When data is fully observed we further proved that PETRELS actually converges globally by making a connection to the PAST algorithm. We demonstrated demonstrate superior performance of PETRELS in direction-of-arrival estimation and showed that it is competitive with existing batch MC algorithms.

References

[1] Y. Chi, Y. C. Eldar, and R. Calderbank, "Petrels: Subspace estimation and tracking from partial observations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2012, pp. 3301–3304.

[2] T. Ahmed, M. Coates, and A. Lakhina, "Multivariate online anomaly detection using kernel recursive least squares," in *Proc. 26th IEEE Int. Conf. Comput. Commun.*, 2007, pp. 625–633.

[3] S. Shahbazpanahi, S. Valaee, and M. H. Bastani, "Distributed source localization using esprit algorithm," *IEEE Trans. Signal Process.*, vol. 49, no. 10, pp. 2169–2178, 2001.

[4] R. Kumaresan and D. Tufts, "Estimating the angles of arrival of multiple plane waves," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-19, no. 1, pp. 134–139, 1983.

[5] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.

[6] I. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Wiley Online Library, 2005.

[7] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 95–107, 1995.

[8] K. Crammer, "Online tracking of linear subspaces," in *Proc. COLT 2006*, 2006, vol. 4005, pp. 438–452.

[9] E. J. Candés and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, pp. 4203–4215, Dec. 2005.

[10] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[11] , Y. C. Eldar and G. Kutyniok, Eds., *Compressed Sensing: Theory and Applications*. New York, NY, USA: Cambridge Univ. Press, 2012.

[12] E. J. Candés and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2053–2080, 2009.

[13] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," *Found. Computat. Math.*, vol. 9, no. 6, pp. 717–772, 2008.

[14] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientif. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[15] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from noisy entries," *J. Mach. Learn. Res.*, pp. 2057–2078, 2010.

[16] K. Lee and Y. Bresler, "Admira: Atomic decomposition for minimum rank approximation," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4402–4416, 2010.

[17] L. Balzano, B. Recht, and R. Nowak, "High-dimensional matched subspace detection when data are missing," in *Proc. 2010 Int. Symp. ISIT*, Austin, TX, USA, Jun. 2010, pp. 1638–1642.

[18] K. Lounici, "High-dimensional covariance matrix estimation with missing observations," 2012 [Online]. Available: arXiv preprint arXiv:1201.2577

[19] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Proc. Allerton* , 2010.

[20] W. Dai, O. Milenkovic, and E. Kerman, "Subspace evolution and transfer (set) for low-rank matrix completion," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3120–3132, 2011.

[21] P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proc. IEEE*, vol. 78, no. 8, pp. 1327–1343, 1990.

[22] E. Oja, "A simplified neuron model as a principal component analyzer," *J. Math. Biol.*, vol. 15, no. 3, pp. 267–273, 1982.

[23] Y. Miao, "Fast subspace tracking and neural network learning by a novel information criterion," *IEEE Trans. Signal Process.*, vol. 46, no. 7, pp. 1967–1979, 1998.

[24] Y. Hua, Y. Xiang, T. Chen, K. Abed-Meraim, and Y. Miao, "A new look at the power method for fast subspace tracking," *Digit. Signal Process.*, vol. 9, no. 4, pp. 297–314, 1999.

[25] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *J. Mach. Learn. Res.*, vol. 11, pp. 1–26, 2009.

[26] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996, vol. 10, no. 8.

[27] C. D. Meyer, "Generalized inversion of modified matrices," *SIAM J. Appl. Math.*, pp. 315–323, 1973.

[28] J. Cioffi, "Limited-precision effects in adaptive filtering," *IEEE Trans. Circuits Syst.*, vol. 34, no. 7, pp. 821–833, 1987.

[29] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Proc. Adv. Neural Inf. Process. Syst.*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., 2008, vol. 20, pp. 161–168.

[30] L. Bottou, "Large-scale machine learning with stochastic gradient descent," *COMPSTAT2010 Book of Abstracts*, p. 270, 2008.

[31] L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Trans. Autom. Control*, vol. 22, no. 4, pp. 551–575, 1977.

[32] B. Yang, "Asymptotic convergence analysis of the projection approximation subspace tracking algorithm," *Signal Process.*, vol. 50, pp. 123–136, 1996.

[33] R. Roy and T. Kailath, "ESPRIT-Estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 7, pp. 984–995, Jul. 1989.

[34] J. Rissanen, “A universal prior for integers and estimation by minimum description length,” *Ann. Statist.*, vol. 11, no. 2, pp. 416–431, 1983.

[35] K. Gedalyahu and Y. C. Eldar, “Time-delay estimation from low-rate samples: A union of subspaces approach,” *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3017–3031, 2010.

[36] Z. Wen, W. Yin, and Y. Zhang, “Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm,” *Math. Program. Comput.*, vol. 4, no. 4, pp. 333–361, 2012.

[37] S. Ma, D. Goldfarb, and L. Chen, “Fixed point and Bregman iterative methods for matrix rank minimization,” *Math. Program.*, vol. 1, no. 1, pp. 1–27, 2009.

[38] J.-F. Cai, E. J. Candes, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM J. Optimiz.*, vol. 20, no. 4, pp. 1–28, 2008.

Yuejie Chi (S’09–M’12) received the Ph.D. degree in electrical engineering from Princeton University in 2012, and the B.E. (Hon.) degree in electrical engineering from Tsinghua University, Beijing, China, in 2007.

Since September 2012, she has been an assistant professor with the Department of Electrical and Computer Engineering and the Department of Biomedical Informatics, The Ohio State University. She has held visiting positions at Colorado State University, Stanford University, and Duke University, and interned at Qualcomm Inc. and Mitsubishi Electric Research Lab. Her research interests include high-dimensional data analysis, statistical signal processing, machine learning and their applications in communications, networks, imaging, and bioinformatics.

Dr. Chi received the Best Paper award from the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) in 2012. She received a Google Faculty Research Award in 2013, a Roberto Padovani scholarship from Qualcomm Inc. in 2010, and an Engineering Fellowship from Princeton University in 2007.

Yonina C. Eldar (S’98–M’02–SM’07–F’12) received the B.Sc. degree in physics and the B.Sc. degree in electrical engineering both from Tel-Aviv University (TAU), Tel-Aviv, Israel, in 1995 and 1996, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2002.

From January 2002 to July 2002, she was a Postdoctoral Fellow at the Digital Signal Processing Group at MIT. She is currently a Professor in the Department of Electrical Engineering, Technion-Israel Institute of Technology, Haifa, and holds The Edwards Chair in Engineering. She is also a Research Affiliate with the Research Laboratory of Electronics, MIT, and a Visiting Professor at Stanford University, Stanford, CA. Her research interests are in the broad areas of statistical signal processing, sampling theory and compressed sensing, optimization methods, and their applications to biology and optics.

Dr. Eldar was in the program for outstanding students at TAU from 1992 to 1996. In 1998, she held the Rosenblith Fellowship for study in electrical engineering at MIT, and in 2000, she held an IBM Research Fellowship. From 2002 to 2005, she was a Horev Fellow of the Leaders in Science and Technology program at the Technion and an Alon Fellow. In 2004, she was awarded the Wolf Foundation Krill Prize for Excellence in Scientific Research, in 2005 the Andre and Bella Meyer Lectureship, in 2007 the Henry Taub Prize for Excellence in Research, in 2008 the Hershel Rich Innovation Award, the Award for Women with Distinguished Contributions, the Muriel & David Jacknow Award for Excellence in Teaching, and the Technion Outstanding Lecture Award, in 2009 the Technion’s Award for Excellence in Teaching, in 2010 the Michael Bruno Memorial Award from the Rothschild Foundation, and in 2011 the Weizmann Prize for Exact Sciences. In 2012, she was elected to the Young Israel Academy of Science and to the Israel Committee for Higher Education, and elected an IEEE Fellow. In 2013 she received the Technion’s Award for Excellence in Teaching, and the Hershel Rich Innovation Award. She received several best paper awards together with her research students and colleagues. She is a Signal Processing Society Distinguished Lecturer, and Editor-in-Chief of *Foundations and Trends in Signal Processing*. In the past, she was a member of the IEEE Signal Processing Theory and Methods and Bio Imaging Signal Processing technical committees, and served as an associate editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the EURASIP *Journal of Signal Processing*, the SIAM *Journal on Matrix Analysis and Applications*, and the SIAM *Journal on Imaging Sciences*.

Robert Calderbank (M’89–SM’97–F’98) received the B.Sc. degree in 1975 from Warwick University, England, the M.Sc. degree in 1976 from Oxford University, England, and the Ph.D. degree in 1980 from the California Institute of Technology, all in mathematics.

He is a Professor of Electrical Engineering at Duke University, Durham, NC, where he now directs the Information Initiative at Duke (iiD) after serving as Dean of Natural Sciences (2010–2013). He was previously a Professor of Electrical Engineering and Mathematics at Princeton University where he directed the Program in Applied and Computational Mathematics. Prior to joining Princeton in 2004, he was Vice President for Research at AT&T, responsible for directing the first industrial research lab in the world where the primary focus is data at scale. At the start of his career at Bell Labs, his innovations were incorporated in a progression of voiceband modem standards that moved communications practice close to the Shannon limit. Together with P. Shor and colleagues at AT&T Labs, he showed that good quantum error correcting codes exist and developed the group theoretic framework for quantum error correction. He is a co-inventor of space-time codes for wireless communication, where correlation of signals across different transmit antennas is the key to reliable transmission.

Dr. Calderbank served as Editor-in-Chief of the IEEE TRANSACTIONS ON INFORMATION THEORY from 1995 to 1998, and as Associate Editor for Coding Techniques from 1986 to 1989. He was a member of the Board of Governors of the IEEE Information Theory Society from 1991 to 1996 and from 2006 to 2008. He was honored by the IEEE Information Theory Prize Paper Award in 1995 for his work on the Z4 linearity of Kerdock and Preparata Codes (joint with A. R. Hammons Jr., P. V. Kumar, N. J. A. Sloane, and P. Sole), and again in 1999 for the invention of space-time codes (joint with V. Tarokh and N. Seshadri). He has received the 2006 IEEE Donald G. Fink Prize Paper Award, the IEEE Millennium Medal, the 2013 IEEE Richard W. Hamming Medal, and he was elected to the US National Academy of Engineering in 2005.