A Non-iterative Online Bayesian Algorithm for the Recovery of Temporally Correlated Sparse Vectors

Geethu Joseph and Chandra R. Murthy *Senior Member, IEEE*

***Abstract*—In this paper, we address the problem of online (sequential) recovery of temporally correlated sparse vectors sharing a common support, from noisy underdetermined linear measurements. The temporal correlation of the sparse vectors is modeled using a first-order autoregressive process. The online algorithm is formulated using the sparse Bayesian learning framework and is implemented using a sequential expectation-maximization procedure. Our algorithm is non-iterative in nature, and requires less computational and memory resources compared to offline processing. We analyze the convergence of the algorithm in the case when the sparse vectors are uncorrelated, using tools from stochastic approximation theory. We show that the sequence of the covariance estimates converge either to the global minimum of the offline equivalent cost function or to the all zero vector, regardless of the sparsity level of the signal. Through numerical results, we demonstrate the efficacy of the proposed online algorithm and compare it with its offline counterpart as well as with existing online sparse vector recovery algorithms.**

***Index Terms*—Sparse signal recovery, Kalman filter, multiple measurement vectors.**

I. Introduction

In many applications, such as wireless channel tracking [1], radar signal processing [2], [3], and biomedical imaging [4]–[7], the goal is to recover a sequence of sparse vectors from their noisy underdetermined linear measurements. Furthermore, the sparse signals exhibit additional structure, such as a common support and temporal correlation. For example, successive instantiations of a time-varying wireless channel have the same power delay profile, and the nonzero coefficients of these instantiations are temporally correlated, and can be modeled using a first-order auto-regressive (AR) process. Hence, our goal in this paper is to develop algorithms that exploit the structure in the signal to reconstruct a sequence of sparse vectors using multiple measurement vectors (MMV). However, exploiting the additional structure can lead to higher latency, memory, and computational complexity. Therefore, we are particularly interested in developing non-iterative algorithms with low complexity and bounded latency.

The extensions of popular sparse signal recovery algorithms like the focal underdetermined system solver (FOCUSS) [8], iterative hard thresholding, orthogonal matching pursuit (OMP) [9], compressive sampling matching pursuit (CoSaMP) [10], approximate message passing (AMP) [11], and sparse Bayesian learning (SBL) [12] to handle the MMV case have been shown to perform better than their single measurement vector counterparts. The recovery performance can be further enhanced if the algorithm exploits the temporal correlation across the sparse vectors [1], [13], [14]. The aforementioned algorithms are offline in nature, i.e., they process the entire set of measurement vectors in a single batch. Hence, when the data set is large, these algorithms suffer from poor efficiency and scalability. On the other hand, *online* algorithms process small batches of the measurement vectors at a time and recover the sparse vectors sequentially, resulting in low-complexity implementations. Online algorithms offer the additional benefit of low latency between the measurement and estimation, which may be necessary in certain applications. For example, in a real-time broadband communication system with high data rate and high mobility, offline estimation of the wireless channel is infeasible.

Several sequential algorithms for sparse signal recovery have been proposed in the literature [15]–[21]. An online algorithm for recovery for sparse signal with comon support is proposed in [15]. However, the algorithm does not account for the temporal correlation in the signal. A non-iterative modified OMP algorithm for sequential recovery of sparse signals is proposed in [16] for the case when the coefficient in the autoregression is unity. A combination of Kalman filtering and dynamic programming is proposed in [17]. This algorithm is slow because it runs l_1 optimization multiple times for every measurement vector. Another iterative sequential algorithm that decouples the support recovery step from the Kalman filtering-based amplitude estimation step is presented in [18]. However, the algorithm requires one to tune a number of parameters beforehand. An alternate iterative online algorithm that jointly estimates the amplitude and support is hierarchical Bayesian Kalman filtering [19]. This algorithm does not require one to tune many parameters, but suffers from high complexity. Another algorithm for the sequential recovery of sparse signals is dynamic sparse coding [20]. The algorithm executes an optimization procedure based on gradient descent, and is also iterative in nature.

The above discussed algorithms do not allow one to improve the current estimate using a small set of future measurements. For scenarios that often arise in communication related applications (e.g., wireless channel estimation), a small delay is allowed if the estimation performance can be improved. An online algorithm that allows a bounded delay between the measurement and estimation by combining the Kalman smoothing and the SBL framework is proposed in [21]. However, the algorithm runs multiple rounds of the expectation-maximization (EM) procedure for every measurement vector,

The authors are with the Dept. of ECE at IISc, Bangalore, India, Emails:{geethu, cmurthy}@ece.iisc.ernet.in

This work was financially supported in part by the Ministry of Electronics and Information Technology (MEITY), Govt. of India. The work of G. Joseph was financially supported in part by the Intel India PhD fellowship.

which defeats the purpose of online computations where the main aim is a simple implementation with minimal resource requirements. This motivates us to develop a *non-iterative online* algorithm which does not require parameter tuning and allows a small delay between the measurement and estimation, for the reconstruction of temporally correlated sparse vectors with common support. To be specific, by the term *non-iterative* we mean that, as every measurement vector arrives, we do not run an iterative procedure until convergence of some metric. Our goal is to design an algorithm which does one round of update using the measurement vector, and wait for the next measurement vector. Also, the term *online* refers to the processing of the measurement vectors in a serial fashion in the order they arrive, without waiting for the entire input available before the start of processing.

Our proposed online algorithm is based on the SBL framework [22], [12]. The SBL approach offers superior performance compared to other algorithms like l_1 minimization and OMP, and does not require one to tune the algorithm parameters. Moreover, it naturally extends to incorporate the temporal correlation structure in the signal model. However, its complexity and memory requirements increase with the number of measurements to be processed, which limits its practical application. Our algorithm overcomes this drawback, and is computationally efficient, while retaining the good performance of SBL. Our main contributions are as follows:

- *Algorithm Development:* We present a non-iterative online algorithm for recovering temporally correlated sparse vectors, in Section III. We propose two schemes for implementation: a fixed lag scheme and a sawtooth lag scheme. We also discuss an efficient method to initialize the algorithm.
- *Complexity Analysis:* We compare the proposed schemes with their offline counterparts from [1], [12] in terms of computational complexity and memory requirements, in Section III-D.
- *Convergence Guarantees:* Using tools from stochastic approximation theory, we prove the convergence of the proposed algorithm for the case when the sparse vectors are uncorrelated, in Section IV. This result holds irrespective of the sparsity level of the signal and the initialization of the algorithm, both under noisy and noiseless cases.
- *Empirical Validation:* In Section V-A, we empirically show that the convergence of the error in the signal covariance falls as a negative power of the number of measurement vectors. Further, we illustrate the performance of the algorithms through Monte Carlo simulations, in terms of the MSE, support recovery rate and run time, and compare them with the offline algorithms proposed in [1], [12], in Section V-B. We also compare the proposed scheme with the other online algorithms in the literature.

Overall, the algorithm proposed in this paper is useful when the underdetermined linear measurements are to be processed in real time, and when there is temporal correlation in the signal of interest in addition to simultaneous sparsity.

Notation: In the sequel, boldface small letters denote vectors and boldface capital letters denote matrices. The i^{th} entry of a vector $\boldsymbol{a}$ is represented as $\boldsymbol{a}[i]$, and the symbols $\|\cdot\|$ and $\|\cdot\|_\infty$ denote the l_2 norm and the l_∞ norm of a vector, respectively. The symbols $(\cdot)^{\mathsf{T}}$, $|\cdot|$, $(\cdot)^\dagger$ and $\text{Tr}\{\cdot\}$ denote the transpose, the determinant, the pseudo inverse, and the trace of a matrix, respectively. Also, $\text{Diag}\{\cdot\}$ represents a vector of diagonal entries of a square matrix or a diagonal matrix with entries of the argument vector on the diagonal, depending on the context. Also, $\odot$ represents the Khatri-Rao product of matrices. The notation $\boldsymbol{I}$, $\boldsymbol{0}$, $\boldsymbol{1}$, and $\mathbb{R}_+$ represent the identity matrix, the all zero matrix (or vector), the all ones vector, and the set of all nonnegative real numbers, respectively. Throughout the paper, $\boldsymbol{\Gamma} = \text{Diag}\{\boldsymbol{\gamma}\}$, $\boldsymbol{\Gamma}_k = \text{Diag}\{\boldsymbol{\gamma}_k\}$ and $\boldsymbol{\Gamma}_{\text{opt}} = \text{Diag}\{\boldsymbol{\gamma}_{\text{opt}}\}$, where $\boldsymbol{\gamma}$, $\boldsymbol{\gamma}_k$ and $\boldsymbol{\gamma}_{\text{opt}}$ are vectors, and we use the notations $\boldsymbol{\Gamma}$ and $\boldsymbol{\gamma}$ interchangeably.

II. Problem Setup and Background

Consider the MMV model given by

$$\boldsymbol{y}_k = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{w}_k, k = 1, 2, \ldots \tag{1}$$

where $\boldsymbol{A}_k \in \mathbb{R}^{m\times N}$ is the known measurement matrix at the k^{th} time instant and $\boldsymbol{y}_k \in \mathbb{R}^m$ is the corresponding noisy measurement. Here, $\boldsymbol{w}_k$ is a zero mean Gaussian distributed noise with a full rank covariance matrix $\boldsymbol{R}_k$. The number of measurements m is assumed to be smaller than the number of unknowns N which makes the system underdetermined. The unknown sequence of vectors $\{\boldsymbol{x}_k, k = 1, 2, \ldots\}$ are sparse, i.e., the number of nonzero entries, S, is small compared to the size of the vector, N. The $\boldsymbol{x}_k$ are simultaneously sparse, that is, they share a common support. This implies that the indices of the nonzero entries of all the sparse vectors coincide. Also, the nonzero entries of $\{\boldsymbol{x}_k, k = 1, 2, \ldots\}$ are temporally correlated. The temporal correlation of the sparse vectors is modeled using a first order AR process, and is given by

$$\boldsymbol{x}_k = \boldsymbol{D}\boldsymbol{x}_{k-1} + \boldsymbol{z}_k. \tag{2}$$

We define $\boldsymbol{x}_0 \triangleq \boldsymbol{0}$ and $\boldsymbol{D} \in [0, 1)^{N\times N}$ is the known diagonal correlation matrix. Note that, in our model, the sparse vectors are temporally correlated, but because $\boldsymbol{D}$ and the covariance of $\boldsymbol{z}_k$ are both assumed to be diagonal, there is no intra-vector correlation. Also, the support of $\boldsymbol{z}_k$ coincides with that of $\{\boldsymbol{x}_k, k = 1, 2, \ldots\}$.

A. Estimation Objectives

The objective of this work is to estimate the sparse vectors on-the-fly, without storing all the measurement data and the corresponding measurement matrices. The maximum delay allowed between the measurement and estimation is $\Delta < \infty$, and therefore our goal is to recursively estimate $\boldsymbol{x}_k$ using the measurements up to time $k + \Delta$, denoted by $\boldsymbol{y}^{k+\Delta}$. Throughout the paper, we use subscripts to denote the value of a variable at a particular time instant (e.g., $\boldsymbol{y}_k$ denotes the observation at time k), and superscripts to denote the sequence of observations up to a particular time instant $\big($e.g., $\boldsymbol{y}^\ell$ denotes the sequence of observations $\{\boldsymbol{y}_k, k = 1, 2, \ldots, \ell\}\big)$.

We design an online scheme inspired by the SBL algorithm [22], [12]. The extension of SBL for the recovery of simultaneous sparse vectors imposes a common prior on the unknown vectors, namely, $\boldsymbol{x}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$ [12]. The covariance matrix $\boldsymbol{\Gamma} \in \mathbb{R}_+^{N \times N}$ is a diagonal matrix with N hyperparameters $\boldsymbol{\gamma} \in \mathbb{R}_+^N$ along the diagonal. In SBL, we compute the ML estimate $\boldsymbol{\gamma}_{\text{ML}}$ of $\boldsymbol{\gamma}$, which in turn gives the MAP estimate of the sparse vectors.

In the following subsections, we contrast the offline and online approaches to estimating the hyperparameters and sparse vectors, which serves to bring out the primary estimation objectives of this work. We start with the online case.

1) Online: Let $\boldsymbol{\gamma}^{k-1}$ denote the sequence of estimates of the hyperparameters $\boldsymbol{\gamma}$ till time $k-1$. At time k, we want to compute the estimate of the hyperparameter vector $\boldsymbol{\gamma}_k$, using $\boldsymbol{y}^{k+\Delta}$ and $\boldsymbol{\gamma}^{k-1}$. Since we do not want to store the complete set of past measurements, we recursively update $\boldsymbol{\gamma}_k$ using a small set of measurements $\{\boldsymbol{y}_t, t = k, k+1, \ldots, k+\Delta\}$ and $\boldsymbol{\gamma}_{k-1}$. The update rule for $\boldsymbol{\gamma}_k$ is discussed in Section III.

Using $\boldsymbol{\gamma}_k$, the online estimate of $\boldsymbol{x}_k$ is computed as its conditional mean given $\boldsymbol{y}^{k+\Delta}$, with $\boldsymbol{\Gamma}_t$ as the covariance of $\boldsymbol{x}_t$ for $t = 1, 2, \ldots, k-1$, and $\boldsymbol{\Gamma}_k$ as the covariance of $\boldsymbol{x}_t$ for $t = k, k+1, \ldots, k+\Delta$. Mathematically,

$$\hat{\boldsymbol{x}}_k = \mathbb{E}\left\{\boldsymbol{x}_k | \boldsymbol{y}^{k+\Delta}; \boldsymbol{\gamma}^{k-1}, \boldsymbol{\gamma}_k\right\}. \tag{3}$$

The estimate $\hat{\boldsymbol{x}}_k$ is obtained using fixed interval Kalman smoothing on a data block of size $\Delta + 1$ [23]. That is, $\boldsymbol{x}_k$ is recursively updated using the set of measurement vectors $\{\boldsymbol{y}_t, t = k, k+1, \ldots, k+\Delta\}$ and $\boldsymbol{\gamma}_k$. Note that $\boldsymbol{\gamma}^{k-1}$ is not used in the estimation of $\boldsymbol{x}_k$.

We emphasize that, with the estimate of $\boldsymbol{\gamma}_k$ in hand, the estimation of $\boldsymbol{x}_k$ is a straightforward application of the Kalman filtering principle. The key contribution of this paper is the development of a recursive, online technique for estimating $\boldsymbol{\gamma}_k$ and its convergence analysis. We next discuss the offline case.

2) Offline: In the offline setting, we find the ML estimate $\boldsymbol{\gamma}^{\text{OFF}}$ of $\boldsymbol{\gamma}$ given the entire sequence $\boldsymbol{y}^K$, where K denotes the total number of measurements [1], [12]. The estimation procedure is detailed in Section II-B. The estimate of $\boldsymbol{x}_k$ is computed as its conditional mean given $\boldsymbol{y}^K$, using $\text{Diag}\left\{\boldsymbol{\gamma}^{\text{OFF}}\right\}$ as the signal covariance matrix. Mathematically,

$$\hat{\boldsymbol{x}}_k^{\text{OFF}} = \mathbb{E}\left\{\boldsymbol{x}_k | \boldsymbol{y}^K; \boldsymbol{\gamma}^{\text{OFF}}\right\}, \tag{4}$$

for $k = 1, 2, \ldots, K$. These estimates are computed efficiently using fixed interval Kalman smoothing on the data block $\boldsymbol{y}^K$.

Thus, the primary goal in both the offline and online algorithms is the estimation of $\boldsymbol{\gamma}$. In the offline case, a single estimate of $\boldsymbol{\gamma}$ is computed using the entire set of observations. In the online version, a sequence of estimates are computed using small batches of observations, and in a recursive manner.

In the next subsection, we first describe the offline SBL algorithm for the correlated MMV problem, which we refer to as the *offline Kalman MMV SBL (KM-SBL)* algorithm [1].

B. Offline KM-SBL Algorithm

The offline algorithm uses the expectation-maximization (EM) procedure, which treats the unknowns $\boldsymbol{x}^K$ as the hidden data and the observations $\boldsymbol{y}^K$ as the known data. The EM procedure iterates between two steps: an expectation step (E-step) and a maximization step (M-step). Let $\boldsymbol{\gamma}^{(r-1)}$ be the estimate of $\boldsymbol{\gamma}$ at the r^{th} iteration.[1] The E-step computes $Q\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(r-1)}\right)$, which is the marginal log-likelihood of the observed data. The M-step computes the hyperparameters that maximize $Q\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(r-1)}\right)$.

$$\textbf{E-step: } Q\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(r-1)}\right) = \mathbb{E}_{\boldsymbol{x}^K | \boldsymbol{y}^K; \boldsymbol{\gamma}^{(r-1)}}\left\{\log p\left(\boldsymbol{y}^K, \boldsymbol{x}^K; \boldsymbol{\gamma}\right)\right\}$$
$$\textbf{M-step: } \boldsymbol{\gamma}^{(r)} = \underset{\boldsymbol{\gamma} \in \mathbb{R}_+^{N \times 1}}{\arg\max}\, Q\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(r-1)}\right). \tag{5}$$

Simplifying $Q\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(r-1)}\right)$ we get,

$$Q\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(r-1)}\right) = c_K - \frac{K}{2}\log|\boldsymbol{\Gamma}| - \frac{1}{2}\text{Tr}\left\{\boldsymbol{\Gamma}^{-1}\boldsymbol{C}_{1|K,\boldsymbol{\gamma}^{(r-1)}}\right\} - \frac{1}{2}\sum_{t=2}^{K}\text{Tr}\left\{\boldsymbol{\Gamma}^{-1}\left(\boldsymbol{I} - \boldsymbol{D}^2\right)^{-1}\boldsymbol{T}_{t|K,\boldsymbol{\gamma}^{(r-1)}}\right\}. \tag{6}$$

where the constant c_K is independent of $\boldsymbol{\gamma}$, and the $N \times N$ matrices are defined as follows:

$$\begin{aligned}
\boldsymbol{T}_{t|K,\boldsymbol{\gamma}^{(r-1)}} &\triangleq \boldsymbol{C}_{t|K,\boldsymbol{\gamma}^{(r-1)}} + \boldsymbol{D}\boldsymbol{C}_{t-1|K,\boldsymbol{\gamma}^{(r-1)}}\boldsymbol{D} \\
&\quad - 2\boldsymbol{D}\boldsymbol{C}_{t,t-1|K,\boldsymbol{\gamma}^{(r-1)}} \\
\boldsymbol{C}_{t|K,\boldsymbol{\gamma}^{(r-1)}} &\triangleq \boldsymbol{P}_{t|K,\boldsymbol{\gamma}^{(r-1)}} + \widehat{\boldsymbol{x}}_{t|K,\boldsymbol{\gamma}^{(r-1)}}\widehat{\boldsymbol{x}}_{t|K,\boldsymbol{\gamma}^{(r-1)}}^{\text{T}} \\
\boldsymbol{C}_{t,t-1|K,\boldsymbol{\gamma}^{(r-1)}} &\triangleq \boldsymbol{P}_{t,t-1|K,\boldsymbol{\gamma}^{(r-1)}} + \widehat{\boldsymbol{x}}_{t|K,\boldsymbol{\gamma}^{(r-1)}}\widehat{\boldsymbol{x}}_{t-1|K,\boldsymbol{\gamma}^{(r-1)}}^{\text{T}},
\end{aligned} \tag{7}$$

for $t \leq K$. Here, the mean $\widehat{\boldsymbol{x}}_{t|K,\boldsymbol{\gamma}^{(r-1)}} \triangleq \mathbb{E}\left\{\boldsymbol{x}_t | \boldsymbol{y}^K; \boldsymbol{\gamma}^{(r-1)}\right\}$; and the covariance $\boldsymbol{P}_{t|K,\boldsymbol{\gamma}^{(r-1)}}$ and the cross-covariance $\boldsymbol{P}_{t,t-1|K,\boldsymbol{\gamma}^{(r-1)}}$ are defined as

$$\boldsymbol{P}_{t|K,\boldsymbol{\gamma}^{(r-1)}} \triangleq \mathbb{E}\left\{\tilde{\boldsymbol{x}}_t\tilde{\boldsymbol{x}}_t^{\text{T}} \,\middle|\, \boldsymbol{y}^K; \boldsymbol{\gamma}^{(r-1)}\right\} \tag{8}$$

$$\boldsymbol{P}_{t,t-1|K,\boldsymbol{\gamma}^{(r-1)}} \triangleq \mathbb{E}\left\{\tilde{\boldsymbol{x}}_t\tilde{\boldsymbol{x}}_{t-1}^{\text{T}} \,\middle|\, \boldsymbol{y}^K; \boldsymbol{\gamma}^{(r-1)}\right\}, \tag{9}$$

where $\tilde{\boldsymbol{x}}_t = \boldsymbol{x}_t - \widehat{\boldsymbol{x}}_{t|K,\boldsymbol{\gamma}^{(r-1)}}$. The calculation of the variables $\widehat{\boldsymbol{x}}_{t|K,\boldsymbol{\gamma}^{(r-1)}}$, $\boldsymbol{P}_{t|K,\boldsymbol{\gamma}^{(r-1)}}$, and $\boldsymbol{P}_{t,t-1|K,\boldsymbol{\gamma}^{(r-1)}}$ is implemented using fixed interval Kalman smoothing [23]. Maximizing $Q\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(r-1)}\right)$ with respect to $\boldsymbol{\gamma}$, we get the following M-step:

$$\boldsymbol{\gamma}^{(r)} = \frac{1}{K}\text{Diag}\left\{\left(\boldsymbol{I} - \boldsymbol{D}^2\right)^{-1}\sum_{t=2}^{K}\boldsymbol{T}_{t|K,\boldsymbol{\gamma}^{(r-1)}} + \boldsymbol{C}_{1|K,\boldsymbol{\gamma}^{(r-1)}}\right\}. \tag{10}$$

We note that the latency in estimating $\boldsymbol{x}_K$ is 0, that of $\boldsymbol{x}_{K-1}$ is 1, and so on. Hence, the average latency of the offline KM-SBL algorithm is $\frac{1}{K}\sum_{t=1}^{K}(K-t) = (K-1)/2$. We now present our proposed online algorithm.

III. Online Algorithm Development

In the online version of KM-SBL, we process the data sequentially, without waiting for the complete input to arrive or storing all the data that has already arrived. Since we do not store data, it is not feasible to compute the mean $\widehat{\boldsymbol{x}}_{t|K}$,[2] the covariance $\boldsymbol{P}_{t|K}$, and the cross-covariance $\boldsymbol{P}_{t,t-1|K}$. Instead, we approximate them with $\widehat{\boldsymbol{x}}_{t|t+\Delta}$, $\boldsymbol{P}_{t|t+\Delta}$, and $\boldsymbol{P}_{t,t-1|t+\Delta}$, respectively. Then,

$$Q_k\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{k-1}\right) \approx a_k - \frac{k}{2}\log|\boldsymbol{\Gamma}| - \frac{1}{2}\text{Tr}\left\{\boldsymbol{\Gamma}^{-1}\boldsymbol{C}_{1|\Delta}\right\} - \frac{1}{2}\text{Tr}\left\{\boldsymbol{\Gamma}^{-1}\left(\boldsymbol{I} - \boldsymbol{D}^2\right)^{-1}\sum_{t=2}^{k}\boldsymbol{T}_{t|t+\Delta}\right\} \tag{11}$$

[1] For ease of notation, we omit the superscript OFF here.

[2] For brevity, we drop $\boldsymbol{\gamma}$ from the subscript.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSP.2017.2725220, IEEE Transactions on Signal Processing

where the constant a_k is independent of $\boldsymbol{\gamma}$.

Maximizing $Q_k\left(\boldsymbol{\gamma},\boldsymbol{\gamma}^{k-1}\right)$ with respect to $\boldsymbol{\gamma}$, we have the following recursion

$$\boldsymbol{\gamma}_k = \frac{1}{k}\text{Diag}\left\{\left(\boldsymbol{I}-\boldsymbol{D}^2\right)^{-1}\sum_{t=2}^{k}\boldsymbol{T}_{t|t+\Delta}+\boldsymbol{C}_{1|\Delta}\right\} \tag{12}$$

$$= \boldsymbol{\gamma}_{k-1} + \frac{1}{k}\text{Diag}\left\{\left(\boldsymbol{I}-\boldsymbol{D}^2\right)^{-1}\boldsymbol{T}_{k|k+\Delta}-\boldsymbol{\Gamma}_{k-1}\right\}. \tag{13}$$

Thus, $\boldsymbol{\gamma}_k$ can be estimated using $\boldsymbol{\gamma}_{k-1}$ and $\boldsymbol{T}_{k|k+\Delta}$. We next present a procedure to recursively estimate $\boldsymbol{T}_{k|k+\Delta}$.

A. Implementation of the Algorithm

In order to compute $\boldsymbol{T}_{k|k+\Delta}$, we need to recursively update the mean $\widehat{\boldsymbol{x}}_{k|k+\Delta}$, the auto-covariance $\boldsymbol{P}_{k|k+\Delta}$, and the cross-covariance $\boldsymbol{P}_{k,k-1|k+\Delta}$. We describe two implementations: a fixed lag scheme and a sawtooth lag scheme.

1) Fixed Lag Scheme: We consider a Kalman filter designed for the following state space model with state variables as $\boldsymbol{x}_k$ and measurement variable as $\widetilde{\boldsymbol{y}}_k \triangleq \boldsymbol{y}_{k+\Delta}$. From (2),

$$\widetilde{\boldsymbol{y}}_k = \boldsymbol{A}_{k+\Delta}\boldsymbol{D}^{\Delta}\boldsymbol{x}_k + \boldsymbol{A}_{k+\Delta}\sum_{i=0}^{\Delta-1}\boldsymbol{D}^i\boldsymbol{z}_{k+\Delta-i}+\boldsymbol{w}_{k+\Delta}$$

$$= \widetilde{\boldsymbol{A}}_k\boldsymbol{x}_k+\widetilde{\boldsymbol{w}}_k \tag{14}$$

where $\widetilde{\boldsymbol{A}}_k \triangleq \boldsymbol{A}_{k+\Delta}\boldsymbol{D}^{\Delta}$ and $\widetilde{\boldsymbol{w}}_k \sim \mathcal{N}\left(0,\widetilde{\boldsymbol{R}}_k\right)$. Since the covariance of $\boldsymbol{z}_{k+\Delta-i}$ is $(\boldsymbol{I}-\boldsymbol{D}^2)\boldsymbol{\Gamma}$, it is easy to show that

$$\widetilde{\boldsymbol{R}}_k = \boldsymbol{A}_{k+\Delta}\left(\boldsymbol{I}-\boldsymbol{D}^{2\Delta}\right)\boldsymbol{\Gamma}\boldsymbol{A}_{k+\Delta}^{\mathrm{T}}+\boldsymbol{R}_{k+\Delta}. \tag{15}$$

The new state space model is given by (2) and (14). The Kalman filter equations for the new system are given below:

$$\widehat{\boldsymbol{x}}_{k|k+\Delta-1} = \boldsymbol{D}\widehat{\boldsymbol{x}}_{k-1|k+\Delta-1} \tag{16}$$

$$\boldsymbol{P}_{k|k+\Delta-1} = \boldsymbol{D}\boldsymbol{P}_{k-1|k+\Delta-1}\boldsymbol{D}+\left(\boldsymbol{I}-\boldsymbol{D}^2\right)\boldsymbol{\Gamma} \tag{17}$$

$$\boldsymbol{J}_k = \boldsymbol{P}_{k|k+\Delta-1}\widetilde{\boldsymbol{A}}_k^{\mathrm{T}}\left(\widetilde{\boldsymbol{A}}_k\boldsymbol{P}_{k|k+\Delta-1}\widetilde{\boldsymbol{A}}_k^{\mathrm{T}}+\widetilde{\boldsymbol{R}}_k\right)^{-1} \tag{18}$$

$$\widehat{\boldsymbol{x}}_{k|k+\Delta} = (\boldsymbol{I}-\boldsymbol{J}_k\widetilde{\boldsymbol{A}}_k)\widehat{\boldsymbol{x}}_{k|k+\Delta-1}+\boldsymbol{J}_k\boldsymbol{y}_{k+\Delta} \tag{19}$$

$$\boldsymbol{P}_{k|k+\Delta} = (\boldsymbol{I}-\boldsymbol{J}_k\widetilde{\boldsymbol{A}}_k)\boldsymbol{P}_{k|k+\Delta-1} \tag{20}$$

$$\boldsymbol{P}_{k,k-1|k+\Delta} = (\boldsymbol{I}-\boldsymbol{J}_k\widetilde{\boldsymbol{A}}_k)\boldsymbol{D}\boldsymbol{P}_{k-1|k+\Delta-1}. \tag{21}$$

As every measurement vector $\boldsymbol{y}_{k+\Delta}$ arrives, the algorithm updates $\boldsymbol{\gamma}$ using (13). Then, the online estimate of $\boldsymbol{x}_k$ can be computed using forward and backward recursions of a fixed interval Kalman smoother on the block of data of size $\Delta+1$, at times $t=k,k+1,\ldots,k+\Delta$, as described in Section II-A1.

Remark: The above scheme is not applicable when $\boldsymbol{D}=\boldsymbol{0}$ and $\Delta>0$, because $\boldsymbol{y}_{k+\Delta}$ is independent of $\boldsymbol{x}_k$ in this case. Also, the fixed lag scheme only uses the latest measurement vector to update $\boldsymbol{\gamma}$, while one can achieve better performance by using all the available measurements in a window around the time instant of interest.

In the following subsection, we propose a sawtooth lag scheme that addresses the above issues.

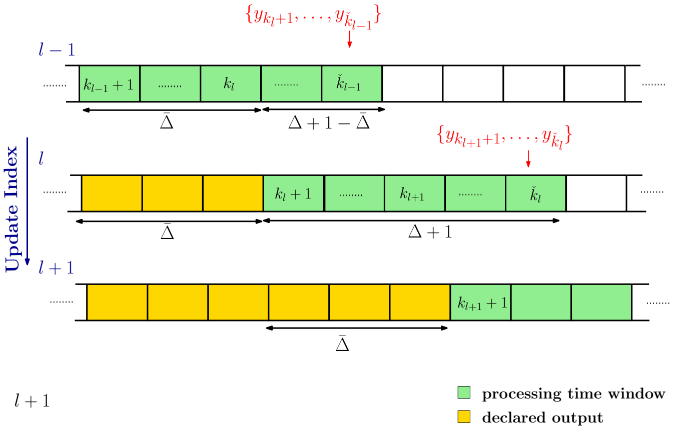


Figure 1. The sawtooth lag processing scheme: Each box represents a time (sampling) instant with which it is indexed, and each row corresponds to an update index, with the index indicated in blue. The set of $\boldsymbol{y}$ in red represents the new measurement set processed in each update. A green box (with indices $k_l+1=(l-1)\bar{\Delta}+1$ to $\check{k}_l=(l-1)\bar{\Delta}+\Delta+1$) indicates that the state statistics corresponding to the index on box are updated, a yellow box (with indices $k\le k_l=(l-1)\bar{\Delta}$) indicates that the state statistics are not updated, and a white box (with indices $k\ge\check{k}_l=(l-1)\bar{\Delta}+\Delta+1$) indicates that the state statistics have not been computed yet. The processing window indicated by green is shifted by $\bar{\Delta}$ after every update.

2) Sawtooth Lag Scheme: In this scheme, we update $\boldsymbol{\gamma}$ as every data block of size $\bar{\Delta}\le\Delta+1$ arrives; see Figure 1. Consider $k\in[k_l+1,k_l+\bar{\Delta}]$ where $k_l\triangleq(l-1)\bar{\Delta}$ for the update index $l=1,2,\ldots$. We replace the fixed lag variables $\widehat{\boldsymbol{x}}_{k|k+\Delta}$, $\boldsymbol{P}_{k|k+\Delta}$, and $\boldsymbol{P}_{k,k-1|k+\Delta}$ with variables $\widehat{\boldsymbol{x}}_{k|\check{k}_l}$, $\boldsymbol{P}_{k|\check{k}_l}$, and $\boldsymbol{P}_{k,k-1|\check{k}_l}$, respectively, where $\check{k}_l\triangleq k_l+\Delta+1$. We compute these variables using the estimate of $\boldsymbol{\gamma}$ obtained in the previous update, $\boldsymbol{\gamma}_{l-1}$. For the l^{th} update, (12) modifies to

$$\boldsymbol{\gamma}_l = \frac{1}{k_{l+1}}\text{Diag}\left\{\left(\boldsymbol{I}-\boldsymbol{D}^2\right)^{-1}\sum_{i=1}^{l}\sum_{\substack{t=k_i+1,\\ t\neq1}}^{k_{i+1}}\boldsymbol{T}_{t|\check{k}_i}+\boldsymbol{C}_{1|\Delta}\right\}$$

$$= \boldsymbol{\gamma}_{l-1}+\frac{1}{k_{l+1}}\sum_{t=k_l+1}^{k_{l+1}}\text{Diag}\left\{\left(\boldsymbol{I}-\boldsymbol{D}^2\right)^{-1}\boldsymbol{T}_{t|\check{k}_l}-\boldsymbol{\Gamma}_{l-1}\right\}. \tag{22}$$

To compute $\boldsymbol{T}_{t|\check{k}_l}$, we run the fixed interval Kalman smoothing algorithm on overlapping blocks of data of size $\Delta+1$, and discard the last $\Delta+1-\bar{\Delta}$ values of every block (this is referred to as sawtooth lag smoothing [24]). The processing window is shifted by $\bar{\Delta}$ after every update. The update equations are comprised of forward recursions and backward recursions. In the forward recursions, we estimate $\widehat{\boldsymbol{x}}_{t|t}$ and $\boldsymbol{P}_{t|t}$ for $t=k_l+1,k_l+2,\ldots,\check{k}_l$ using a Kalman filter as given below:

$$\hat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{D}\hat{\boldsymbol{x}}_{t-1|t-1} \tag{23}$$

$$\boldsymbol{P}_{t|t-1} = \boldsymbol{D}\boldsymbol{P}_{t-1|t-1}\boldsymbol{D}+(\boldsymbol{I}-\boldsymbol{D}^2)\boldsymbol{\Gamma} \tag{24}$$

$$\boldsymbol{J}_t = \boldsymbol{P}_{t|t-1}\boldsymbol{A}_t^{\mathrm{T}}\left(\boldsymbol{A}_t\boldsymbol{P}_{t|t-1}\boldsymbol{A}_t^{\mathrm{T}}+\boldsymbol{R}_t\right)^{-1} \tag{25}$$

$$\hat{\boldsymbol{x}}_{t|t} = (\boldsymbol{I}-\boldsymbol{J}_t\boldsymbol{A}_t)\hat{\boldsymbol{x}}_{t|t-1}+\boldsymbol{J}_t\boldsymbol{y}_t \tag{26}$$

$$\boldsymbol{P}_{t|t} = (\boldsymbol{I}-\boldsymbol{J}_t\boldsymbol{A}_t)\boldsymbol{P}_{t|t-1} \tag{27}$$

$$\boldsymbol{P}_{\check{k}_l,\check{k}_l-1|\check{k}_l} = \left(\boldsymbol{I}-\boldsymbol{J}_{\check{k}_l}\boldsymbol{A}_{\check{k}_l}\right)\boldsymbol{D}\boldsymbol{P}_{\check{k}_l-1|\check{k}_l-1}. \tag{28}$$

In the backward recursions, we estimate $\widehat{\boldsymbol{x}}_{t|\check{k}_l}$, $\boldsymbol{P}_{t|\check{k}_l}$ and $\boldsymbol{P}_{t,t-1|\check{k}_l}$ in the reverse order. For $t = \check{k}_l, \check{k}_l - 1, \ldots, k_l + 2$ we get the following smoothing equations:

$$\boldsymbol{G}_{t-1} = \boldsymbol{P}_{t-1|t-1}\boldsymbol{D}\boldsymbol{P}_{t|t-1}^{-1} \tag{29}$$

$$\hat{\boldsymbol{x}}_{t-1|\check{k}_l} = \hat{\boldsymbol{x}}_{t-1|t-1} + \boldsymbol{G}_{t-1}(\hat{\boldsymbol{x}}_{t|\check{k}_l} - \hat{\boldsymbol{x}}_{t|t-1}) \tag{30}$$

$$\boldsymbol{P}_{t-1|\check{k}_l} = \boldsymbol{P}_{t-1|t-1} + \boldsymbol{G}_{t-1}(\boldsymbol{P}_{t|\check{k}_l} - \boldsymbol{P}_{t|t-1})\boldsymbol{G}_{t-1}^{\mathrm{T}} \tag{31}$$

For $t \neq \check{k}_l$

$$\boldsymbol{P}_{t,t-1|\check{k}_l} = \boldsymbol{P}_{t|t}\boldsymbol{G}_{t-1}^{\mathrm{T}} + \boldsymbol{G}_t\Big(\boldsymbol{P}_{t+1,t|\check{k}_l} - \boldsymbol{D}\boldsymbol{P}_{t|t}\Big)\boldsymbol{G}_{t-1}^{\mathrm{T}}. \tag{32}$$

The average latency of the fixed lag scheme is Δ, whereas that of the sawtooth lag scheme is $\Delta - \left(\bar{\Delta} - 1\right)/2$. In the sawtooth lag scheme, $\bar{\Delta}$ also controls the frequency of update of $\boldsymbol{\gamma}$. If $\bar{\Delta}$ is large, the average latency decreases, but the $\boldsymbol{\gamma}$ gets updated more slowly. So, there is a tradeoff between the accuracy and the latency in selecting $\bar{\Delta}$.

Next, we discuss the special case of $\boldsymbol{D} = \boldsymbol{0}$. We refer to this algorithm as the *online M-SBL algorithm*, as there is no role for Kalman filtering when $\boldsymbol{D} = \boldsymbol{0}$.

Online M-SBL: When the sparse vectors are uncorrelated, i.e., $\boldsymbol{D} = \boldsymbol{0}$, (22) simplifies to the following recursion:

$$\boldsymbol{\gamma}_l = \boldsymbol{\gamma}_{l-1} + \frac{1}{k_{l+1}} \sum_{t=k_l+1}^{k_{l+1}} \mathrm{Diag}\left\{\boldsymbol{P}_t(\boldsymbol{\gamma}_{l-1}) + \widehat{\boldsymbol{x}}_t(\boldsymbol{y}_t, \boldsymbol{\gamma}_{l-1})\widehat{\boldsymbol{x}}_t(\boldsymbol{y}_t, \boldsymbol{\gamma}_{l-1})^{\mathrm{T}} - \boldsymbol{\Gamma}_{l-1}\right\}. \tag{33}$$

where

$$\boldsymbol{P}_t(\boldsymbol{\gamma}) \triangleq \boldsymbol{\Gamma} - \boldsymbol{\Gamma}\boldsymbol{A}_t^{\mathrm{T}}\left(\boldsymbol{A}_t\boldsymbol{\Gamma}\boldsymbol{A}_t^{\mathrm{T}} + \boldsymbol{R}_t\right)^{-1}\boldsymbol{A}_t\boldsymbol{\Gamma} \tag{34}$$

$$\widehat{\boldsymbol{x}}_t(\boldsymbol{y}, \boldsymbol{\gamma}) \triangleq \boldsymbol{P}_t(\boldsymbol{\gamma})\boldsymbol{A}_t^{\mathrm{T}}\boldsymbol{R}_t^{-1}\boldsymbol{y}. \tag{35}$$

We note that this implementation depends only on $\bar{\Delta}$, and not on Δ, because $\left\{\boldsymbol{y}_t, t = k_{l+1}+1, k_{l+1}+2, \ldots, \check{k}_l\right\}$ and $\{\boldsymbol{x}_t, t = k_l + 1, k_l + 2, \ldots, k_{l+1}\}$ are independent.

To summarize, we have presented a fixed lag scheme and a sawtooth lag scheme, for computing $\boldsymbol{T}_{k|k+\Delta}$ recursively using the data in batches. We next discuss the initialization of the algorithm and several interesting special cases.

B. Discussion

1) Initialization: The initial estimate of $\boldsymbol{\gamma}$ can be obtained from the first $\Delta + 1$ input measurements vectors using the offline KM-SBL algorithm. The one round of the offline KM-SBL algorithm can be interpreted as an estimation step, and the recursive update of $\boldsymbol{\gamma}$ using (13) can be interpreted as a tracking process. In fact, if $\boldsymbol{\gamma}$ is slowly varying over time, the recursive update step (13) can track its temporal variations.

2) Special Cases: We make a few interesting observations about the algorithm in the following special cases:

(a) When $\boldsymbol{D} = \boldsymbol{0}$, the sparse vectors are uncorrelated and thus $\widehat{\boldsymbol{x}}_{t|K} = \widehat{\boldsymbol{x}}_{t|t+\Delta}$, $\boldsymbol{P}_{t|K} = \boldsymbol{P}_{t|t+\Delta}$, and $\boldsymbol{P}_{t,t-1|K} = \boldsymbol{P}_{t,t-1|t+\Delta}$. Hence, there is no approximation in (11). On the other hand, as the correlation coefficient increases, the approximation in (11) becomes loose.

(b) When $\boldsymbol{D} = \boldsymbol{0}$ and $\Delta = 0$, the fixed lag and the sawtooth lag schemes become identical.

(c) When $\Delta = 0$, the filter for the modified state space reduces to the original Kalman filter equations [23].

Scheme		Computational cost	Memory demand
KM-SBL ($\boldsymbol{D} \neq \boldsymbol{0}$)	Offline	$\mathcal{O}(KN^3)$	$\mathcal{O}\left(KN^2\right)$
	Fixed lag	$\mathcal{O}\left(KN^2m\right)$	$\mathcal{O}\left(\Delta N^2\right)$
	Sawtooth lag	$\mathcal{O}(KN^3)$	$\mathcal{O}\left(\Delta N^2\right)$
M-SBL ($\boldsymbol{D} = \boldsymbol{0}$)	Offline	$\mathcal{O}(KN^2m)$	$\mathcal{O}\left(Km + N^2\right)$
	Online	$\mathcal{O}(KN^2m)$	$\mathcal{O}\left(\Delta m + N^2\right)$

Table I
COMPARISON OF THE ONLINE SCHEMES WITH THE OFFLINE SCHEME WHEN K OBSERVATIONS ARE AVAILABLE.

(d) When $\bar{\Delta} = 1$, the latency of the sawtooth lag scheme equals Δ for all sparse vectors, similar to the fixed lag scheme. Nonetheless, the two schemes are different, because of the forward and backward recursions in the sawtooth lag scheme.

C. Refinements

1) Different Learning Rates: Instead of $1/k$ in (13), any sequence of positive numbers b_k can be used in the recursive algorithm as long as the following conditions are satisfied:

$$0 \leq b_k \leq 1 \qquad \sum_{k=1}^{\infty} b_k = \infty \qquad \sum_{k=1}^{\infty} b_k^2 < \infty. \tag{36}$$

The modified algorithm is given by

$$\boldsymbol{\gamma}_k = \boldsymbol{\gamma}_{k-1} + b_k \mathrm{Diag}\left\{\left(\boldsymbol{I} - \boldsymbol{D}^2\right)^{-1}\boldsymbol{T}_{k|k+\Delta} - \boldsymbol{\Gamma}_{k-1}\right\}. \tag{37}$$

A good choice for the sequence is $b_k = 1/k^{\alpha}$, $1/2 < \alpha \leq 1$, since $\sum_{k=1}^{\infty} 1/k^{\alpha}$ converges if $\alpha > 1$ and diverges otherwise. In Section V, we empirically show that the modified algorithm converges faster than the original version (see Figure 2).

2) Improved Online M-SBL: Notice that the online M-SBL algorithm in (33) does not use the observations $\boldsymbol{y}_t$, $t = k_{l+1} + 1, k_{l+1}+2, \ldots, \check{k}_l$, even though they are available at time k_{l+1}. Hence, we modify the update step in (33) to update $\boldsymbol{\gamma}$ using all the available measurement vectors $\boldsymbol{y}^{\check{k}_l}$, and then estimate the sparse vectors $\hat{\boldsymbol{x}}_{k_l+1}$ to $\hat{\boldsymbol{x}}_{k_{(l+1)}}$, as follows:

$$\boldsymbol{\gamma}_l = \boldsymbol{\gamma}_{l-1} + \frac{1}{\check{k}_l} \sum_{t=\check{k}_l-\bar{\Delta}+1}^{\check{k}_l} \mathrm{Diag}\bigg\{\boldsymbol{P}_t(\boldsymbol{\gamma}_{l-1}) + \widehat{\boldsymbol{x}}_t(\boldsymbol{y}_t, \boldsymbol{\gamma}_{l-1})\widehat{\boldsymbol{x}}_t(\boldsymbol{y}_t, \boldsymbol{\gamma}_{l-1})^{\mathrm{T}} - \boldsymbol{\Gamma}_{l-1}\bigg\}. \tag{38}$$

Thus, for each update, we use only the latest available block of size $\bar{\Delta}$, and not the past values which have already been used. Hence, in this case, we need not store any of the past measurements or the sparse vector estimates.

D. Complexity Analysis

We now briefly discuss the computational complexity and memory requirements of the proposed algorithms.

1) Computational Cost: We assume that the multiplication of a $p \times q$ matrix with a $q \times r$ matrix requires $\mathcal{O}(pqr)$ floating-point operations (flops), and the inversion of a $p \times p$ positive definite matrix requires $\mathcal{O}(p^3)$ flops [25].

We note that the computational cost per update of $\boldsymbol{\gamma}$ in the online scheme depends only on Δ (which is $\ll K$), although the overall computational complexity does depend on the number of sparse vectors K. However, simulation results show that the overall run time of our online algorithms grow slowly with K when compared to their offline counterparts (see Figure 3a). The order-wise complexity of the online M-SBL algorithm (33) is similar to the online KM-SBL fixed-lag scheme, but its run time is much smaller than KM-SBL since it does not involve Kalman filtering or smoothing. Note that, the computational cost of the offline algorithms correspond to the complexity of a single iteration, while that of the online algorithms correspond to the overall complexity, as they are non-iterative in nature.

2) Memory Requirement: In the offline KM-SBL algorithm, we need to save all measurement vectors, because of which, the memory requirement grows linearly with K. For the online KM-SBL schemes, we need to save data only over a small processing time window of size Δ. Thus, the memory requirement for the online schemes scales with Δ. The variables that need to be stored are the statistics (mean and covariance) of the sparse vectors which is of the order N^2 values.

When the sparse vectors are uncorrelated (online M-SBL algorithm), we need to store only the measurement vectors of order $m \ll N^2$ values, and not the statistics of the past sparse vectors. Also, for the update of the hyperparameter $\boldsymbol{\gamma}$, we need an extra working memory of the order N^2 to compute the covariance matrices $\boldsymbol{P}_k$. Thus, the overall memory demand for the offline M-SBL is of the order $Km + N^2$, while that for the online algorithm is of the order $\Delta m + N^2$.

We compare the computational demands and the memory requirements of the three schemes in Table I.

IV. Convergence Analysis

In the section, we study the convergence properties of the proposed online algorithm under the following assumptions:

(A1) The measurement matrices are identical, i.e., $\boldsymbol{A}_k = \boldsymbol{A}$, $\forall k$, and without loss of generality, Rank $\{\boldsymbol{A}\} = m$.

(A2) The noise covariance matrix is the same for all measurements, i.e., $\boldsymbol{R}_k = \boldsymbol{R}$, $\forall k$.

(A3) The sparse vectors are uncorrelated, i.e., $\boldsymbol{D} = \boldsymbol{0}$.

The above assumptions are standard in the MMV literature, and are referred to as the joint sparsity model-2 (JSM-2) [8], [10]–[12]. The assumptions simplify the recursive algorithm, and make the analysis tractable. Since $\boldsymbol{D} = \boldsymbol{0}$, the fixed lag scheme discussed in Section III-A1 is not applicable, and we focus our analysis on the sawtooth lag implementation. We start with the case when $\bar{\Delta} = 1$. A similar analysis follows for $\bar{\Delta} > 1$, and we discuss this case later in the sequel.

When $\boldsymbol{A}_k = \boldsymbol{A}$ and $\boldsymbol{R}_k = \boldsymbol{R}$, (33)-(35) simplify to

$$\boldsymbol{\gamma}_k = \boldsymbol{\gamma}_{k-1} + \frac{1}{k}\text{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma}_{k-1})\right\} + \frac{1}{k}\text{Diag}\left\{\widehat{\boldsymbol{x}}(\boldsymbol{y}_k, \boldsymbol{\gamma}_{k-1})\widehat{\boldsymbol{x}}(\boldsymbol{y}_k, \boldsymbol{\gamma}_{k-1})^{\mathrm{T}} - \boldsymbol{\Gamma}_{k-1}\right\} \quad (39)$$

where $\boldsymbol{P}(\boldsymbol{\gamma})$ and $\widehat{\boldsymbol{x}}(\boldsymbol{y}, \boldsymbol{\gamma})$ are as defined in (34) and (35), with $\boldsymbol{A}_t$ and $\boldsymbol{R}_t$ replaced by $\boldsymbol{A}$ and $\boldsymbol{R}$, respectively. We can rewrite (39) as a stochastic approximation recursion as follows:

$$\boldsymbol{\gamma}_k = \boldsymbol{\gamma}_{k-1} + \frac{1}{k}\boldsymbol{f}(\boldsymbol{\gamma}_{k-1}) + \frac{1}{k}\boldsymbol{e}_k. \quad (40)$$

Here, $\boldsymbol{f}(\boldsymbol{\gamma})$ is the mean field function, given by

$$\boldsymbol{f}(\boldsymbol{\gamma}) \triangleq \text{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma}) + \boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}^{-1}\mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\right\}\boldsymbol{R}^{-1}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})\right\} - \boldsymbol{\gamma}, \quad (41)$$

where the expectation is over the distribution of $\boldsymbol{y}$, and $\boldsymbol{e}_k$ is given by

$$\boldsymbol{e}_k \triangleq \text{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma}_{k-1}) + \widehat{\boldsymbol{x}}(\boldsymbol{y}_k, \boldsymbol{\gamma}_{k-1})\widehat{\boldsymbol{x}}(\boldsymbol{y}_k, \boldsymbol{\gamma}_{k-1})^{\mathrm{T}}\right\} - \boldsymbol{\gamma}_{k-1} - \boldsymbol{f}(\boldsymbol{\gamma}_{k-1}).$$

Further, using $\boldsymbol{P}(\boldsymbol{\gamma})$ from (34),

$$\boldsymbol{P}(\boldsymbol{\gamma}) - \boldsymbol{\Gamma} = -\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma} \quad (42)$$

$$\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}^{-1} = \boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{R}\right)^{-1}. \quad (43)$$

Thus, we get (44) and (45) at the top of the next page.

We next present the convergence results of the algorithm. We begin with a proposition which shows that the sequence of $\boldsymbol{\gamma}_k$ generated by the proposed algorithm is bounded.

Proposition 1. *If $\boldsymbol{\gamma}_0$ is a nonnegative vector, the sequence $\boldsymbol{\gamma}_k$ generated by* (39) *remains in a compact subset of $\mathbb{R}_+^N$ almost surely (*a.s.*).*

Proof: See Appendix A. ■

The next question to be answered is about the values to which the sequence $\boldsymbol{\gamma}_k$ could converge. The following theorem characterizes the asymptotic behavior of the algorithm.

Theorem 1. *Assume that the nonzero entries of $\boldsymbol{x}$ are orthogonal, and the diagonal matrix $\boldsymbol{\Gamma}_{opt} \triangleq \mathbb{E}\left\{\boldsymbol{x}\boldsymbol{x}^T\right\}$. If $\boldsymbol{\gamma}_0$ is a nonnegative vector, then the sequence $\boldsymbol{\gamma}_k$ of the proposed online M-SBL algorithm given by* (39) *converges to the set $\{\boldsymbol{0}\} \cup \{\boldsymbol{\gamma} \in \mathbb{R}_+^N : \boldsymbol{A}(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{opt})\boldsymbol{A}^T = \boldsymbol{0}\}$* a.s. *Further, if Rank $\{\boldsymbol{A} \odot \boldsymbol{A}\} = N$, the sequence $\boldsymbol{\gamma}_k$ converges to a point in the two-element set $\{\boldsymbol{0}, \boldsymbol{\gamma}_{opt}\}$* a.s.

Proof: See Appendix B. ■

We make the following observations from Theorem 1.

- The results are independent of the following parameters:
  (a) sparsity level of the unknown vectors
  (b) initialization of the algorithm (however, $\boldsymbol{\gamma}_0 \in \mathbb{R}_+^N$)
  (c) distribution of the sparse vectors (even though the algorithm is designed assuming a Gaussian distribution), as long as the entries are orthogonal
  (d) properties of $\boldsymbol{A}$, such as its restricted isometry constant or mutual coherence
  (e) construction of $\boldsymbol{A}$, i.e., it can be deterministic or random, with normalized or unnormalized columns.
- The convergence guarantee of the original M-SBL algorithm in [12] holds only in the noiseless case. However, our generalized result applies whether noise is present or not. Hence, the result is practically more useful.
- The condition that the nonzero entries of $\boldsymbol{x}$ should be orthogonal is similar to the orthogonality condition

$$\boldsymbol{f}(\boldsymbol{\gamma}) = \text{Diag}\left\{\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\left(\mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\right\}-\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}-\boldsymbol{R}\right)\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}\right\} \tag{44}$$

$$\boldsymbol{e}_k = \text{Diag}\left\{\boldsymbol{\Gamma}_{k-1}\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}_{k-1}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\left(\boldsymbol{y}_k\boldsymbol{y}_k^{\mathrm{T}}-\mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\right\}\right)\left(\boldsymbol{A}\boldsymbol{\Gamma}_{k-1}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}_{k-1}\right\}. \tag{45}$$

required for the convergence guarantee of the original M-SBL algorithm in the noiseless case [12]. In fact, the orthogonality condition in [12] is hard to achieve since the number of sparse vectors to be estimated is finite. In that sense, ours is a more reasonable assumption.

- The M-SBL cost function [12] is defined as

$$\begin{aligned} V_{\text{M-SBL}}(\boldsymbol{\gamma}) &= \lim_{k\to\infty}\left[\frac{1}{k}\sum_{t=1}^{k}\boldsymbol{y}_t^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{y}_t \right. \\ &\qquad \left. + \log\left|\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right|\right] \\ &= \text{Tr}\left\{\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\left(\boldsymbol{A}\boldsymbol{\Gamma}_{\text{opt}}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)\right\} \\ &\qquad - \log\left|\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\right|. \end{aligned} \tag{46}$$

  We note that $V_{\text{M-SBL}}(\boldsymbol{\gamma}) - \log|\boldsymbol{A}\boldsymbol{\Gamma}_{\text{opt}}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}| - m$ is the Kullback-Leibler (KL) divergence between two Gaussian distributions: $\mathcal{N}(\boldsymbol{0}, \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R})$ and $\mathcal{N}(\boldsymbol{0}, \boldsymbol{A}\boldsymbol{\Gamma}_{\text{opt}}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R})$. The global minimum of $V_{\text{M-SBL}}(\boldsymbol{\gamma})$ is therefore achieved at $\{\boldsymbol{\gamma}\in\mathbb{R}_+^N : \boldsymbol{A}(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}_{\text{opt}})\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{0}\}$. Hence, the set to which our algorithm converges contains all the points achieving the global minimum of $V_{\text{M-SBL}}(\boldsymbol{\gamma})$.
- Since $V_{\text{M-SBL}}(\boldsymbol{\gamma})$ is a function of $\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}$, the smallest set to which M-SBL can converge is $\{\boldsymbol{\gamma}\in\mathbb{R}_+^N : \boldsymbol{A}(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}_{\text{opt}})\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{0}\}$. The $\boldsymbol{\gamma}_k$ output by the proposed algorithm converges to the union of this set with $\boldsymbol{0}$.
- It can be shown that the algorithm is guaranteed to converge to a sparse solution, where, by sparse solution, we mean one with no more than m nonzero entries. Given any s-sparse vector $\boldsymbol{\gamma}_{\text{opt}}$ and sensing matrix $\boldsymbol{A}$, we can always construct a pair $(\boldsymbol{x}_c, \boldsymbol{y}_c)$ such that $\boldsymbol{y}_c = \boldsymbol{A}\boldsymbol{x}_c$ and $\boldsymbol{x}_c = \boldsymbol{\Gamma}_{\text{opt}}^{1/2}(\boldsymbol{A}\boldsymbol{\Gamma}_{\text{opt}}^{1/2})^{\dagger}\boldsymbol{y}_c$. By [26, Theorem 1], $\boldsymbol{\gamma}_{\text{opt}}$ is the global minimizer of the SBL cost function constructed under a noiseless measurement model using $\boldsymbol{y}_c$ and $\boldsymbol{A}$. Further, from [26, Theorem 2], it is known that every local minimum of the SBL cost function is achieved at a sparse solution (even in the presence of noise). Now, the SBL cost is a function of $\boldsymbol{\Gamma}$ only through $\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}$. Hence, the set $\{\boldsymbol{\gamma}\in\mathbb{R}_+^N : \boldsymbol{A}(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}_{\text{opt}})\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{0}\}$ consists of local minima of this SBL cost function, which implies that the elements of the set are all sparse. Therefore, the algorithm is guaranteed to converge to a sparse solution.

We can extend the above convergence results to the refined algorithm given by (37) using the following corollary.

Corollary 1. *Consider the modified online M-SBL algorithm given by* (37) *and having learning rates satisfying* (36). *Under the assumptions of Theorem 1, the sequence* $\boldsymbol{\gamma}_k$ *converges to a point in the set* $\{\boldsymbol{0}\}\cup\{\boldsymbol{\gamma}\in\mathbb{R}_+^N : \boldsymbol{A}(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}_{opt})\boldsymbol{A}^{T}=\boldsymbol{0}\}$ a.s. *Further, if Rank* $\{\boldsymbol{A}\odot\boldsymbol{A}\} = N$, *the sequence* $\boldsymbol{\gamma}_k$ *converges to a point in the set* $\{\boldsymbol{0}, \boldsymbol{\gamma}_{opt}\}$ a.s.

The proof of the above is similar to that of Theorem 1 because the only properties of the sequence $1/k$ (in (13)) that are used in Theorem 1 are the ones listed in (36).

We now consider to the more general case where $\bar{\Delta} \geq 1$. As in the previous case, the algorithm can be rewritten as a stochastic approximation recursion as follows:

$$\boldsymbol{\gamma}_l = \boldsymbol{\gamma}_{l-1} + \frac{1}{l}\boldsymbol{f}(\boldsymbol{\gamma}_{l-1}) + \frac{1}{l}\tilde{\boldsymbol{e}}_l, \tag{47}$$

where $\boldsymbol{f}(\boldsymbol{\gamma})$ is as defined in (41), and

$$\begin{aligned} \tilde{\boldsymbol{e}}_l \triangleq -\boldsymbol{f}(\boldsymbol{\gamma}_{l-1}) + \frac{1}{\bar{\Delta}}\sum_{t=k_l+1}^{k_l+\bar{\Delta}} \text{Diag}\Big\{&\boldsymbol{P}(\boldsymbol{\gamma}_{l-1}) \\ &+ \widehat{\boldsymbol{x}}(\boldsymbol{y}_t, \boldsymbol{\gamma}_{l-1})\widehat{\boldsymbol{x}}(\boldsymbol{y}_t, \boldsymbol{\gamma}_{l-1})^{\mathrm{T}}\Big\}. \end{aligned} \tag{48}$$

The following theorem characterizes the asymptotic behavior of the above algorithm. Using the theorem, we can also derive a corollary similar to Corollary 1. However, we omit the statement to avoid repetition.

Theorem 2. *Under the assumptions of Theorem 1, the sequence* $\boldsymbol{\gamma}_l$ *output by the online M-SBL algorithm given by* (47) *converges to the set* $\{\boldsymbol{0}\}\cup\{\boldsymbol{\gamma}\in\mathbb{R}_+^N : \boldsymbol{A}(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}_{opt})\boldsymbol{A}^{T}=\boldsymbol{0}\}$ a.s. *Further, if Rank* $\{\boldsymbol{A}\odot\boldsymbol{A}\} = N$, *the sequence* $\boldsymbol{\gamma}_l$ *converges to a point in the set* $\{\boldsymbol{0}, \boldsymbol{\gamma}_{opt}\}$ a.s.

Proof: The algorithm given by (47) differs from the algorithm given by (40) only in the last term. The only place where this term plays a role in the proof in Appendix B is via Lemma 1. Hence, it suffices to show that $\lim_{l\to\infty}\sum_{i=1}^{l}\frac{1}{i}\tilde{\boldsymbol{e}}_i$ exists and is finite. From (48), we get

$$\begin{aligned} \tilde{\boldsymbol{e}}_l = \text{Diag}\Big\{&\boldsymbol{\Gamma}_{l-1}\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}_{l-1}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1} \\ &\left(\mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\right\} - \frac{1}{\bar{\Delta}}\sum_{t=k_i+1}^{k_{i+1}}\boldsymbol{y}_t\boldsymbol{y}_t^{\mathrm{T}}\right)\left(\boldsymbol{A}\boldsymbol{\Gamma}_{l-1}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}_{l-1}\Big\}. \end{aligned}$$

Now the result follows by replacing $\boldsymbol{e}_k$ in the proof of Lemma 1 with $\tilde{\boldsymbol{e}}_l$. ■

We can also get similar convergence results for the improved M-SBL algorithm given by (38), as follows.

Corollary 2. *Under the assumptions of Theorem 1, the sequence* $\boldsymbol{\gamma}_l$ *output by the improved online M-SBL algorithm given by* (38) *converges to* $\{\boldsymbol{0}\}\cup\{\boldsymbol{\gamma}\in\mathbb{R}_+^N : \boldsymbol{A}(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}_{opt})\boldsymbol{A}^{T}=\boldsymbol{0}\}$ a.s. *Further, if Rank* $\{\boldsymbol{A}\odot\boldsymbol{A}\} = N$, *the sequence* $\boldsymbol{\gamma}_l$ *converges to a point in the set* $\{\boldsymbol{0}, \boldsymbol{\gamma}_{opt}\}$ a.s.

Proof: Under the assumptions of Theorem 1, the improved online algorithm given by (38) is equivalent to the original algorithm given by (33) except that it uses $\bar{\Delta}$ measurement vectors $\{\boldsymbol{y}_t, t = \check{k}_l - \bar{\Delta} + 1, \check{k}_l - \bar{\Delta} + 2, \ldots, \check{k}_l\}$ instead of $\bar{\Delta}$ measurements $\{\boldsymbol{y}_t, t = k_l + 1, k_l + 2, \ldots, k_{l+1}\}$ used by the original version. Since the measurement vectors are independent and identically distributed, the rest of the proof is the same as that of Theorem 1. ■

Algo.	**Rademacher Dist.**			**Gaussian Dist.**		
	$\bar{\Delta}=1$	$\bar{\Delta}=3$	$\bar{\Delta}=5$	$\bar{\Delta}=1$	$\bar{\Delta}=3$	$\bar{\Delta}=5$
$\alpha=0.6$	1.69	1.30	1.17	1.18	1.09	0.96
$\alpha=0.8$	0.87	0.79	0.72	0.86	0.78	0.71
$\alpha=1.0$	0.49	0.47	0.43	0.49	0.47	0.43

Table II
VALUE OF ERROR-FIT POWER FUNCTION PARAMETER p WHEN $\boldsymbol{D}=\boldsymbol{0}$.

V. SIMULATION RESULTS

We use the following setup to evaluate the performance of the algorithm and corroborate the theoretical results. We generate sparse signals of length $N=60$, each with $s=6$ nonzero entries. The locations of nonzero coefficients are chosen uniformly at random, and the nonzero entries are independent and identically distributed with zero mean and unit variance. The length of measurement vector is chosen as $m=20$. The measurement matrices $\boldsymbol{A}_k$ are generated with independent and Gaussian distributed entries with zero mean, and the columns are normalized to have unit Euclidean norm.

We study the properties of the algorithm for both uncorrelated and highly correlated cases in the following subsections. For the uncorrelated case, we consider the improved online algorithm given by (38).

A. Convergence

To study the convergence of the algorithm, we consider three different learning rates $b_k=1/k^{\alpha}$: $\alpha=0.6, 0.8$ and 1. The maximum delay between the measurement and estimation is taken as $\Delta=5$. To highlight the convergence behavior, we initialize the hyperparameters with a fixed value $4\cdot\boldsymbol{1}$, irrespective of the measurements. The SNR is chosen as 20 dB for all the results in this subsection.

1) Uncorrelated Case: We generate the sparse vectors from two distributions: Gaussian and Rademacher distribution. The mean squared error (MSE) in the estimated hyperparameters when $\bar{\Delta}=3$ are plotted in Figure 2a. The curves labeled `Fit` are the fitted curves on the error using the function: $f(x)=ax^{-p}$ where a and p are parameters. The result for other values of $\bar{\Delta}$ is similar, and we summarize the values of p in Table II. Our observations from the results are as follows:

Convergence: The algorithm converges to the true $\boldsymbol{\gamma}$, and not to the other equilibrium point, $\boldsymbol{\gamma}=\boldsymbol{0}$, in all cases. This happens even if we initialize the algorithm with very small values such as $10^{-2}\cdot\boldsymbol{1}$.

Sparse vector distribution: The algorithm works equally well for both Gaussian (which is continuous) and Rademacher distribution (which is discrete), as guaranteed by Theorem 2.

Learning rate: The smaller the α, the larger the learning rate b_k, and hence the larger the weightage given to the update term $\text{Diag}\left\{\left(\boldsymbol{I}-\boldsymbol{D}^2\right)\boldsymbol{T}_{k|k+\Delta}-\boldsymbol{\Gamma}_{k-1}\right\}$ in (37), leading to faster convergence. Since $1/2<\alpha\leq 1$ is required for theoretical convergence guarantee, a value of α close to $1/2$ ensures the fastest convergence. However, we have also observed from our experiments that $\alpha\leq 1/2$ leads to even faster convergence. Hence, in practice, one could try using $\alpha\leq 1/2$, but the convergence would not be guaranteed by our analysis.

Value of $\bar{\Delta}$: As $\bar{\Delta}$ increases, the exponent p slightly decreases. This is because when $\bar{\Delta}$ increases, the hyperparameter $\boldsymbol{\gamma}$ gets updated less frequently. Hence, a lower $\bar{\Delta}$ improves the convergence rate and estimation accuracy, but at the cost of higher average latency and computational complexity. This is further illustrated in the following subsections.

2) Highly Correlated Case: Figures 2b and 2c show the MSE in the hyperparameter estimates when $\bar{\Delta}=3$, for the fixed lag and sawtooth lag schemes, respectively. A few interesting observations from the figures are as follows:

Correlation coefficient: As the correlation coefficient increases, the convergence becomes slower. This is because the approximation in (11) becomes loose as the correlation increases, as discussed in Section III-B.

Implementation scheme: We see that the convergence behavior of the fixed lag and sawtooth lag schemes are similar. However, the gap between the curves when the correlation coefficient is 0.9 and 0.95 is smaller for the fixed lag scheme compared to the sawtooth lag scheme. Further discussion about this is provided in Section V-B2.

Learning rates: As observed in the uncorrelated case, the convergence is faster for small values of α. However, the gap between the curves for the two correlation coefficients is wider for smaller values of α. This is because as α decreases, the weightage given to the update term in (37) increases, and thus, it becomes more sensitive to the approximation in (11).

B. Algorithm Performance: Varying Paramters

We evaluate the performance of the proposed algorithm using the three metrics defined below. We let $\hat{\boldsymbol{x}}_k$ and $\boldsymbol{x}_k$ denote the estimate and true value of the sparse vector, respectively.

(i) Relative mean square error (RMSE)

$$\text{RMSE}\triangleq\frac{\sum_{k=1}^{K}\|\hat{\boldsymbol{x}}_k-\boldsymbol{x}_k\|^2}{\sum_{k=1}^{K}\|\boldsymbol{x}_k\|^2}. \tag{49}$$

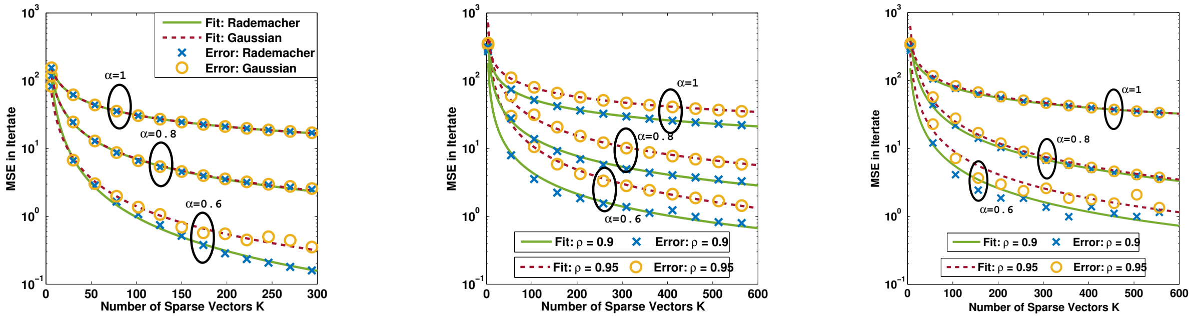
(ii) Support recovery rate (SRR)

$$\text{SRR}\triangleq 1-\frac{1}{K}\sum_{k=1}^{K}\frac{|\text{Supp}\{\hat{\boldsymbol{x}}_k-\boldsymbol{x}_k\}|}{N}. \tag{50}$$

(iii) Run time, which is the time required to complete the computations. It measures the computational complexity.

We refer to the RMSE and SRR metrics jointly as the recovery performance of the algorithm. We consider two methods to initialize the hyperparameter vector $\boldsymbol{\gamma}$ for the online schemes, which we term *proper* initialization and *fixed* initialization. Proper initialization refers to initializing $\boldsymbol{\gamma}$ with its estimate obtained from the first $\Delta+1$ measurements using the offline KM-SBL algorithm. Fixed initialization refers to initializing $\boldsymbol{\gamma}$ with a fixed vector (which we take as $4\cdot\boldsymbol{1}$).

1) Uncorrelated Case: Figures 3a-3f show the performance of the different schemes when $\boldsymbol{D}=\boldsymbol{0}$. The curves labeled `Offline` correspond to the performance of the offline M-SBL algorithm, which is our benchmark, and all other curves correspond to the improved online sawtooth lag scheme discussed in Section III-C2. The curves labeled `Init` $\bar{\Delta}=1$, `Init` $\bar{\Delta}=3$ and `Init` $\bar{\Delta}=5$ correspond to the online algorithm with proper initialization, while the curves labeled



(a) (b) (c)

Figure 2. Convergence of the hyperparameters to the true value for different learning rates $b_k = 1/k^{\alpha}$, $\alpha = 0.6, 0.8$ and 1 when $\Delta = 5$. For (2a) we choose $\boldsymbol{D} = \mathbf{0}$, and for (2b) and (2c) $\boldsymbol{D} = \rho \boldsymbol{I}$, and the value of $\bar{\Delta} = 3$ is chosen as for (2a) and (2c). Further, (2b) corresponds to the fixed lag scheme and (2c) corresponds to the sawtooth lag scheme. The markers show the error value corresponding to the two distributions and the dotted line shows the curve fitted on the error using a power function. We infer that the procedure converges to the true value with the MSE in estimation being a power function of K. The rate of convergence improves as the value of α is decreased.

No Init $\bar{\Delta} = 1$, No Init $\bar{\Delta} = 3$ and No Init $\bar{\Delta} = 5$ correspond to the online algorithm with fixed initialization. Our observations from the results are as follows:

Initialization: The online scheme with proper initialization closely matches with the offline scheme in terms of the recovery performance. On the other hand, the online scheme with fixed initialization requires significantly smaller time for execution, but the convergence is slower.

Number of sparse vectors K: As K increases, the quality of the covariance estimate improves (as seen in Section V-A), and this, in turn, leads to better recovery performance; see Figures 3a and 3b. From Figure 3c, we see that the run time increases almost linearly with K for the offline scheme and the online scheme with fixed initialization. With proper initialization, the run time is roughly constant with K, as most of execution time is spent in computing the initialization of $\boldsymbol{\gamma}$.

SNR: The recovery performance of all algorithms improve with increase in SNR, see Figures 3d and 3e. Also, the gap between the online scheme with proper initialization and the offline scheme virtually closes beyond an SNR of 10 dB. From Figure 3f, the run time remains almost constant with SNR, even though the offline scheme and the online scheme with proper initialization use an iterative step to estimate $\boldsymbol{\gamma}$.

Sparsity level: The recovery performance of all algorithms degrade with increase in sparsity level (number of non-zero entries), see Figures 3g and 3h. However, the SRR performance of the algorithm with fixed initialization degrades significantly with the increase in the sparsity level. From Figure 3i, the run time remains almost constant with sparsity level, since the complexity does not depend on the sparsity level.

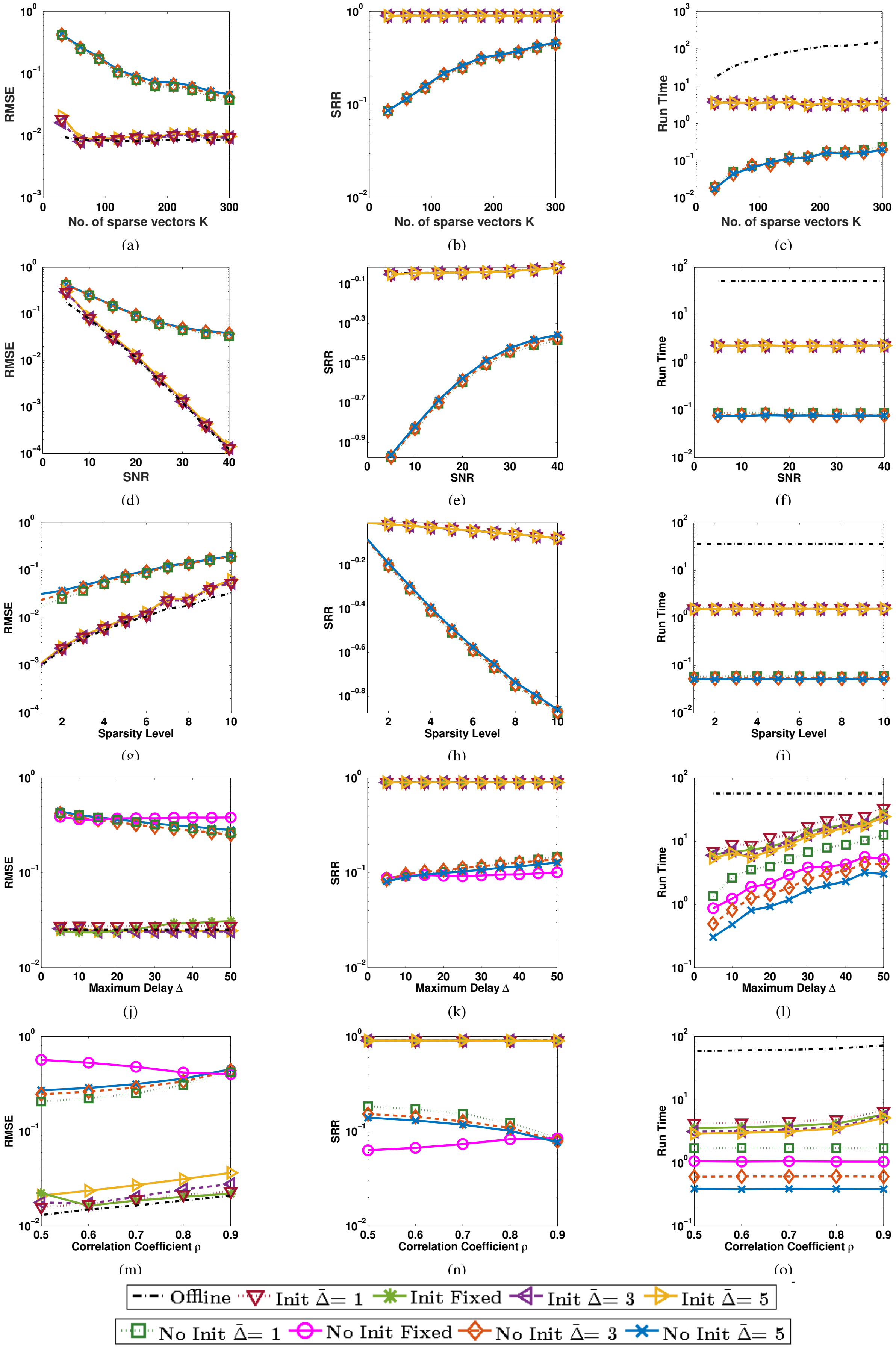
Output batch-size $\bar{\Delta}$: The performance of online schemes do not vary much with $\bar{\Delta}$, as can be seen from Figures 3a-3f. However, the recovery performance is slightly better and the run time is slightly worse for smaller values of $\bar{\Delta}$, as $\boldsymbol{\gamma}$ is updated more frequently.

Maximum delay Δ: The performance of the algorithm with varying maximum delay Δ is similar to that of the highly correlated case as shown in Figure 3j-Figure 3l. We omit the plot due to lack of space. The performance of the online schemes improve as Δ increases, and the proper initialization can greatly improve the recovery performance compared to fixed initialization. The run time of the online scheme with proper initialization increases with Δ, because the number of measurement vectors used to initialize $\boldsymbol{\gamma}$ increases. However, the behavior the run time of the online schemes for the uncorrelated case is different from that of the highly correlated case, as discussed in Section III-D. This is because the online algorithms use Kalman smoothing in the correlated case, and the complexity of Kalman smoothing increases with Δ. In the uncorrelated case, the complexity is independent of Δ, thus the run time remains constant for all values of Δ.

2) Highly Correlated Case: Figures 3j-3m show the performance of the different algorithms when the sparse vectors are highly correlated ($\boldsymbol{D} \neq \mathbf{0}$). The curves labeled Init Fixed and No Init Fixed correspond to the fixed lag scheme with proper and fixed initialization, respectively, while the other labels are as in the previous plots. Our observations from the results are as follows:

Implementation schemes: As discussed in Section III-A, for the same output batch-size of $\bar{\Delta} = 1$, the sawtooth lag scheme outperforms the fixed lag scheme, at the cost of a higher run time. This is because the sawtooth lag scheme uses all the available measurements for updating the hyperparameters, while the fixed lag scheme uses only the latest available measurement. Comparing the fixed lag scheme with the sawtooth lag scheme with higher output batch-sizes ($\bar{\Delta} = 3$ and 5), the fixed lag scheme is slower but more accurate, as it updates the hyperparameters more frequently.

Correlation coefficient ρ: The performance of the algorithms with varying correlation coefficient ρ (recall $\boldsymbol{D} = \rho \boldsymbol{I}$) is shown in Figures 3m-3o. As ρ increases, the recovery performance of the sawtooth lag scheme decreases, while that of the fixed lag scheme improves. This seemingly counterintuitive behavior can be explained as follows. In the offline case, an increase in ρ can worsen the support recovery of the sparse vectors, but helps the estimation of the amplitude of the nonzero entries. A combination of these effects determine the overall performance of the algorithm, and we see that the recovery performance slightly degrades as the ρ increases. A similar trend was observed in the SRR for the temporal MMV-



(a) (b) (c)

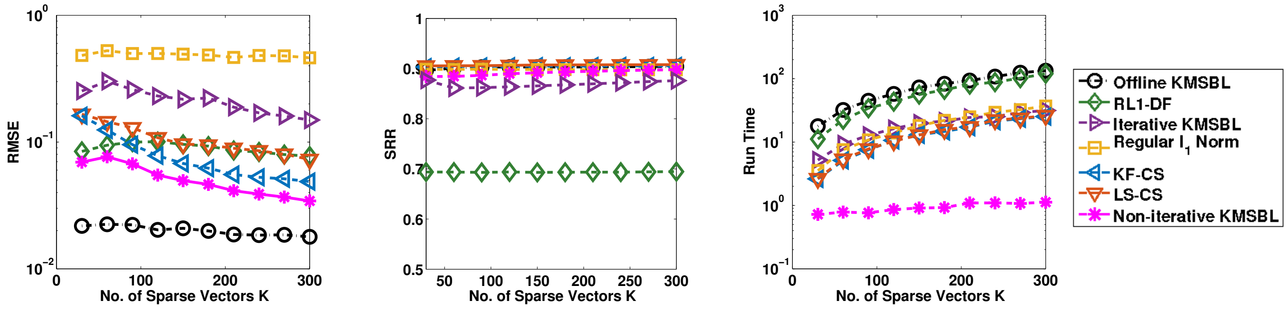
(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 3. Performance of the proposed algorithms relative to the offline algorithm. Unless otherwise mentioned in the plot $K = 150$, $\Delta = 5$, $\rho = 0.9$ and SNR = 20 dB. For (3a)-(3f) $\boldsymbol{D} = \mathbf{0}$ (uncorrelated case, where we use the M-SBL based algorithm), and for (3j)-(3o) $\boldsymbol{D} = \rho\boldsymbol{I}$ (correlated case, where we use the KM-SBL algorithm). The performance of the proposed algorithm is comparable to the offline algorithm, but it requires significantly lower run time for all the settings shown here. Initializing the algorithms from the first $\Delta + 1$ measurements using the offline KM-SBL algorithm offers better RMSE but at the cost of increased run time.



(a) (b) (c)

Figure 4. Comparison of RMSE, SRR and run time of the proposed algorithm with the existing online schemes, when $\boldsymbol{D} = 0.9\boldsymbol{I}$, $\Delta = 0$ and SNR = 20 dB. The proposed algorithm requires one order of magnitude lower run time than all the other schemes to which it is compared, for achieving a similar performance, while achieving a similar SRR and slightly superior RMSE.

SBL (TM-SBL) algorithm for recovering correlated sparse vectors [27, Figure 2]. In case of the sawtooth lag scheme, in addition to the above, an increase in ρ also makes the approximation in (11) loose. Due to this, the degradation in the recovery performance of the sawtooth lag scheme is large compared to the offline algorithm. In case of the fixed lag scheme, apart from the effects discussed above, an increase in ρ also improves ρ^{Δ}, the correlation between the state and the observation in the new state space model (described by (2) and (14)). This improves the quality of the estimate output by the Kalman filter, and in turn helps the recovery. The overall effect of these is an improvement in the recovery performance of the fixed lag scheme. A more rigorous study of the effect of ρ an interesting topic for future work.

The run time of the algorithm remains the same for all values of ρ for the fixed initialization case, as its complexity is independent of ρ. However, the run time of the online schemes with proper initialization is higher in the highly correlated case. This is because, when data is highly correlated, the initialization phase using the offline scheme takes more iterations to converge. We can see a similar slight increase in the run time of the offline scheme in the highly correlated case.

Maximum delay Δ*:* As the delay increases, the recovery performance of the online schemes increases for both methods of initialization. The change is more evident for the fixed initialization case, as the recovery performance of with proper initialization is very close to that of the offline scheme. We also observe that the improvement in recovery performance is small for the fixed lag scheme compared to the sawtooth lag scheme. This is because of the reduced correlation $\left(\boldsymbol{D}^{\Delta}\right)$ between the state and the observation of the new state space model given by (2) and (14). Also as pointed out earlier, the run time of the online schemes increases with Δ.

Output batch-size $\bar{\Delta}$*:* The performance of the online algorithms remains constant with $\bar{\Delta}$ for both the correlated and uncorrelated case. However, the gap between the run time curves is wider for the correlated case. This is because each update of γ is computationally more expensive due to the Kalman smoothing in the correlated case.

The performance of the online algorithms with K and SNR in the highly correlated case is similar to that observed in the uncorrelated case. We omit these plots due to lack of space.

C. Comparison with Existing Algorithms

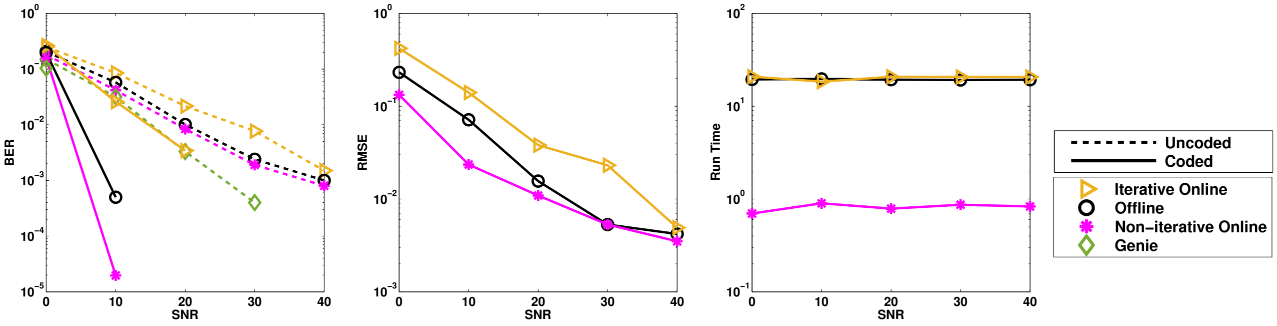
In Figure 4a-Figure 4c, we compare the proposed algorithm, labeled `Non-iterative KMSBL`, with the following algorithms (labels in brackets):

(i) Offline KM-SBL [1] (`Offline KMSBL`)
(ii) Reweighted l_1 dynamic filtering [17] (`RL1-DF`)
(iii) Iterative online KM-SBL [21] (`Iterative KMSBL`)
(iv) Standard l_1 minimization based algorithm on each measurement vector [28] (`Regular` l_1 `Norm`)
(v) Kalman compressed sensing [18] (`KF-CS`)
(vi) Least squares compressed sensing [15] (`LS-CS`)

Here, we choose $\Delta = 0$, as the other online schemes except the iterative online KM-SBL algorithm are not designed for $\Delta > 0$. We also note that we extended the Kalman compressed sensing algorithm in [18] to handle a first-order AR process with correlation matrix $\boldsymbol{D} \in [0, 1]^N$, while the original algorithm only considers $\boldsymbol{D} = \boldsymbol{I}$. The recovery performance of the proposed scheme is comparable with the other online schemes algorithms, and approaches the offline performance as K increases. However, the run time of the proposed scheme is significantly lower than all the other schemes. Moreover, the rate of increase of the run time of the proposed scheme with K is much smaller than the other schemes. The significant reduction in the run time is primarily due to the non-iterative nature of the proposed scheme. Since all other algorithms are iterative in nature, their complexity and hence run time depends linearly on the number of iterations which, in turn, depends on N, m, K, the threshold used for stopping the iterations, etc. This brings out the major difference between the other algorithms and the proposed online non-iterative schemes. Thus, our scheme is both fast and accurate, as promised in Section I.

D. Sparse Orthogonal Frequency Division Multiplexing Channel Estimation

In this subsection, we consider the sparse orthogonal frequency division multiplexing (OFDM) channel estimation problem as an application of our proposed algorithm [1]. We list the simulation parameters in Table III. The sparse channel is of length $N = 59$, which taken as the length of the cyclic prefixing (CP), with $s = 6$ nonzero entries for each channel instantiation (PedB channel model [29]). In each OFDM symbol, $m = 20$ pilot symbols are placed uniformly, and the number of OFDM symbols K is taken as 150. We assume that



(a) (b) (c)

Figure 5. Comparison of the BER, RMSE and run time of the proposed algorithm with existing schemes, namely, the offline KM-SBL [1] (`Offline`), iterative online KM-SBL [21] (`Online Iterative`), a receiver with perfect knowledge of channel (`Genie`), for sparse OFDM channel estimation. The proposed algorithm requires over one order of magnitude lower run time than the existing schemes, and achieves similar or better BER and RMSE.

the algorithms estimate the channel once in every OFDM slot, which gives $\Delta = 6$. We consider both coded[3] and uncoded scenarios and three metrics for the performance comparison: BER, MSE in channel estimation, and run time per channel vector estimation. We estimate the channel using the pilot symbols, and decode the data using the channel estimate (for details, refer to [1]). In Figure 5a-Figure 5c, we compare the performance of the proposed algorithm, labeled `Online Non-iterative`, with the following three schemes (labels in brackets):

(i) Offline KM-SBL [1] (`Offline`)
(ii) Iterative online KM-SBL [21] (`Online Iterative`)
(iii) Receiver with perfect knowledge of channel (`Genie`)

As mentioned earlier, the other online schemes are not applicable here, as we take $\Delta > 0$. From the figure, we infer that the BER and the MSE performance of the proposed algorithm is better than the offline algorithm which was originally proposed for the channel estimation problem [1]. This is because the offline algorithm processes the data in blocks of size 6, and does not reuse the past measurements blocks, whereas our algorithm uses information from all past measurement blocks to estimate the channel vectors for the current block. Moreover, our algorithm has an added advantage of significantly reduced run time.

VI. Conclusions

In this work, we introduced an online algorithm for recovering a sequence of temporally correlated joint sparse vectors from noisy linear underdetermined measurements. The temporal correlation is modeled using a first-order AR process. We developed the algorithm by combining the sequential EM procedure and the SBL framework, and proposed two schemes for implementation: the fixed lag and sawtooth lag schemes. Our algorithm is non-iterative in nature, and does not require any parameter tuning. We also provided a rigorous convergence analysis of the proposed algorithm. Simulations showed that the performance of the proposed algorithm is close to that of the offline algorithm, but it demands less memory and computational resources, both when the sparse vectors are uncorrelated and highly correlated. However, the unit correlation coefficient is a case where the proposed approach fails to effectively recover the single sparse vector; devising online algorithms for this scenario is an interesting line for future work. It would also be interesting to extend the convergence analysis to the correlated sparse vector case.

Parameter		Value
OFDM (3GPP/LTE broadband standard [31])	Transmission bandwidth	2.5 MHz
	Sub-frame duration	0.5 ms
	Subcarrier spacing	15 kHz
	Sampling frequency	3.84 MHz
	FFT size	256
	No. of data subcarriers	200
	OFDM symbol/slot	6
	CP length	16.67 μs
Channel	Environment	Pedestrian B [29]
	Model	Jakes model [32]
	Norm. Doppler freq.	10^{-3}
Coding and modulation		rate 1/2 Turbo code and QPSK
Pulse shaping		Raised cosine with rolloff factor= 0.5 [33]

Table III
Simulation parameters for OFDM channel estimation

Appendix A
Proof of Proposition 1

We first prove a lemma to show that the noise term $\boldsymbol{e}_k$ is bounded, which then enables us to establish the required result.

Lemma 1. *In the proposed online algorithm given by* (40), $\lim_{k\to\infty}\sum_{t=1}^{k}\frac{1}{t}\boldsymbol{e}_t$ *exists and is finite.*

Proof: We define $\boldsymbol{l}_k = \sum_{t=1}^{k}\frac{1}{t}\boldsymbol{e}_t$, and $\mathcal{F}_k$ as the $\sigma-$algebra generated by $\boldsymbol{y}^k$. Then, $\mathbb{E}\left\{\boldsymbol{l}_k|\mathcal{F}_{k-1}\right\} = \mathbb{E}\left\{\boldsymbol{l}_{k-1}|\mathcal{F}_{k-1}\right\} + \frac{1}{k}\mathbb{E}\left\{\boldsymbol{e}_k|\mathcal{F}_{k-1}\right\} = \boldsymbol{l}_{k-1}$. Thus, $\boldsymbol{l}_{k-1}$ is a martingale. Further, using the orthogonality property of martingales [34],

$$\mathbb{E}\left\{\|\boldsymbol{l}_k\|^2\right\} = \sum_{t=1}^{k}\mathbb{E}\left\{\|\boldsymbol{l}_t - \boldsymbol{l}_{t-1}\|^2\right\} = \sum_{t=1}^{k}\frac{1}{t^2}\mathbb{E}\left\{\|\boldsymbol{e}_t\|^2\right\}. \quad (51)$$

We note that $\|\boldsymbol{y}\|_\infty < \infty$ a.s., thus (45) shows that $\|\boldsymbol{e}_t\| < \infty$ a.s., if $\|\boldsymbol{\gamma}_{k-1}\|_\infty < \infty$. When $\|\boldsymbol{\gamma}_{k-1}\|_\infty \to \infty$, from (45), it

[3] For the Turbo code generation, we use the publicly available software [30].

can be shown that

$$\begin{aligned}&\lim_{\|\boldsymbol{\gamma}\|_\infty\to\infty} \boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\\ &=\lim_{\|\boldsymbol{\gamma}\|_\infty\to\infty}\|\boldsymbol{\gamma}\|_\infty^{-\frac{1}{2}}\boldsymbol{\Gamma}^{\frac{1}{2}}\left[\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\big(\|\boldsymbol{\gamma}\|_\infty^{-1}\boldsymbol{\Gamma}\big)\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}^{-\frac{1}{2}}\right]^{\dagger}\boldsymbol{R}^{-\frac{1}{2}}. \quad (52)\end{aligned}$$

Hence, all entries of $\lim_{\boldsymbol{\gamma}\to\infty}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}$ are finite, and $\|\boldsymbol{e}_t\| < \infty$ with probability one. Thus, $\mathbb{E}\left\{\|\boldsymbol{e}_t\|^2\right\}$ is bounded, and hence by Jensen's inequality and (51), the martingale is bounded in $\mathcal{L}^1$. Applying Doob's forward convergence theorem [34] to each coordinate of the martingale $\boldsymbol{l}_k[i], i=1,2,\ldots,N$, the limit $\lim_{k\to\infty}\boldsymbol{l}_k=\lim_{k\to\infty}\sum_{t=1}^{k}\frac{1}{t}\boldsymbol{e}_t$ exists, and is finite. ■

We now formally prove Proposition 1.

Proof: Using (40), we have,

$$\begin{aligned}\boldsymbol{\gamma}_k=\frac{k-1}{k}\boldsymbol{\gamma}_{k-1}+\frac{1}{k}\mathrm{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma}_{k-1})\right.\\ \left.+\widehat{\boldsymbol{x}}(\boldsymbol{y}_k,\boldsymbol{\gamma}_{k-1})\widehat{\boldsymbol{x}}(\boldsymbol{y}_k,\boldsymbol{\gamma}_{k-1})^{\mathrm{T}}\right\}. \quad (53)\end{aligned}$$

All entries of $\mathrm{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma}_{k-1})+\widehat{\boldsymbol{x}}(\boldsymbol{y}_k,\boldsymbol{\gamma}_{k-1})\widehat{\boldsymbol{x}}(\boldsymbol{y}_k,\boldsymbol{\gamma}_{k-1})^{\mathrm{T}}\right\}$ are nonnegative. This ensures that $\boldsymbol{\gamma}_k[i]\geq 0$ for $i=1,2,\ldots,N$ and $\forall k$, if $\boldsymbol{\gamma}_0$ is a nonnegative vector. Thus, the sequence $\boldsymbol{\gamma}_k$ is bounded from below.

Next, we use [35, Theorem 7] to show that the sequence is bounded from above, and hence it remains in a compact set. For that, we check if the conditions below hold in our case:

(i) The function $\boldsymbol{f}$ is Lipschitz
(ii) $\lim_{k\to\infty}\sum_{t=1}^{k}\frac{1}{t}\boldsymbol{e}_t$ exists
(iii) The function $\boldsymbol{f}_\infty(\boldsymbol{\gamma})=\lim_{c\to\infty}\boldsymbol{f}(c\boldsymbol{\gamma})/c$ is continuous, and the ordinary differential equation (ODE)

$$\frac{d}{dt}\boldsymbol{\gamma}(t)=\boldsymbol{f}_\infty(\boldsymbol{\gamma}(t)) \quad (54)$$

has the origin as its unique globally asymptotic stable equilibrium.

Since $\boldsymbol{P}(\boldsymbol{\gamma})$ and $\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}$ are positive semidefinite, all of their diagonal entries are nonnegative. Hence, using (41),

$$\begin{aligned}\boldsymbol{f}(\boldsymbol{\gamma})&\geq-\boldsymbol{\gamma}+\mathrm{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}^{-1}\mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\right\}\boldsymbol{R}^{-1}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})\right\}\\ &\geq-\boldsymbol{\gamma}, \quad (55)\end{aligned}$$

where $\boldsymbol{a}\geq\boldsymbol{b}$ denotes that every entry of $\boldsymbol{a}$ is greater than or equal to the corresponding entry of $\boldsymbol{b}$. Further, since the matrix $\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}$ is positive semidefinite, every diagonal entry of $\boldsymbol{P}(\boldsymbol{\gamma})=\boldsymbol{\Gamma}-\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}$ is less than the corresponding diagonal entry of $\boldsymbol{\Gamma}$. Thus, we get

$$\begin{aligned}\boldsymbol{f}(\boldsymbol{\gamma})&\leq\mathrm{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}^{-1}\mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\right\}\boldsymbol{R}^{-1}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})\right\}\\ &\leq\lambda\mathrm{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}^{-2}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})\right\} \quad (56)\end{aligned}$$

where λ is the largest eigenvalue of the positive semidefinite matrix $\mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\right\}$, and $\boldsymbol{a}\leq\boldsymbol{b}$ denotes an entry-wise inequality. Thus,

$$-\boldsymbol{\gamma}[i]\leq\boldsymbol{f}(\boldsymbol{\gamma})[i]\leq\lambda\mathrm{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}^{-2}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})\right\}[i], \quad (57)$$

for $i=1,2,\ldots,N$. To further bound the last term of the inequality, we use (43) to get

$$\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}^{-2}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})=\boldsymbol{\Gamma}^{\frac{1}{2}}\boldsymbol{B}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{\Gamma}^{\frac{1}{2}}. \quad (58)$$

where $\boldsymbol{B}\triangleq\boldsymbol{\Gamma}^{\frac{1}{2}}\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-\frac{1}{2}}$. This implies

$$\begin{aligned}&\mathrm{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}^{-2}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})\right\}[i]\\ &\quad=\boldsymbol{\gamma}[i]\boldsymbol{B}[i]^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{B}[i] \quad (59)\\ &\quad\leq\boldsymbol{\gamma}[i]\boldsymbol{B}[i]^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{B}[i], \quad (60)\end{aligned}$$

where $\boldsymbol{B}[i]\in\mathbb{R}^N$ is the i^{th} column of $\boldsymbol{B}^{\mathrm{T}}$. Then, we have

$$\begin{aligned}\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}&=\boldsymbol{\Gamma}^{\frac{1}{2}}\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\\ &=\boldsymbol{I}-\left(\boldsymbol{I}+\boldsymbol{\Gamma}^{\frac{1}{2}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\right)^{-1}. \quad (61)\end{aligned}$$

This shows that $\boldsymbol{I}-\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}$ is a positive semidefinite matrix, and its diagonal entries are nonnegative. Thus, $\boldsymbol{B}[i]^{\mathrm{T}}\boldsymbol{B}[i]\leq 1$, for $i=1,2,\ldots,N$. Hence, we get

$$\mathrm{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}^{-2}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})\right\}[i]\leq\bar{\lambda}\boldsymbol{\gamma}[i], \quad (62)$$

where $\bar{\lambda}$ is the largest eigenvalue of $\boldsymbol{R}^{-1}$. Substituting this relation in (57), we get

$$-\boldsymbol{\gamma}[i]\leq\boldsymbol{f}(\boldsymbol{\gamma})[i]\leq\bar{\lambda}\lambda\boldsymbol{\gamma}[i]. \quad (63)$$

Thus, (i) is satisfied. The assumption (ii) is true by Lemma 1. To check (iii), we start with (44) to get

$$\begin{aligned}\boldsymbol{f}_\infty(\gamma)&=\lim_{c\to\infty}\frac{1}{c}\mathrm{Diag}\Big\{c^2\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}\big(c\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\big)^{-1}\big(\mathbb{E}\left\{\boldsymbol{y}_k\boldsymbol{y}_k^{\mathrm{T}}\right\}\\ &\qquad-c\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}-\boldsymbol{R}\big)\left(c\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}\Big\} \quad (64)\\ &=-\lim_{c\to\infty}\mathrm{Diag}\Big\{\boldsymbol{\Gamma}\left(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\right)^{\mathrm{T}}\Big[\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\\ &\qquad\left(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\right)^{\mathrm{T}}+\boldsymbol{I}/c\Big]^{-1}\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\Big\} \quad (65)\\ &=-\mathrm{Diag}\left\{\boldsymbol{\Gamma}\left(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\right)^{\dagger}\left(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\right)\right\}. \quad (66)\end{aligned}$$

Note that $\mathrm{Rank}\{(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}})\}=\min\{\mathrm{Rank}\left\{\boldsymbol{\Gamma}\right\},m\}$. When $\mathrm{Rank}\{(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}})\}=\mathrm{Rank}\left\{\boldsymbol{\Gamma}\right\}$, $\boldsymbol{f}_\infty(\gamma)=-\boldsymbol{\gamma}$. Since $\mathbf{0}$ is the only globally asymptotically stable equilibrium of the ODE $\frac{d}{dt}\gamma(t)=-\boldsymbol{\gamma}(t)$, (iii) holds. When $\mathrm{Rank}\{\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\}=m$,

$$\left(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\right)^{\dagger}=\boldsymbol{\Gamma}^{\frac{1}{2}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}^{-\frac{1}{2}}\left(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}^{-\frac{1}{2}}\right)^{-1}, \quad (67)$$

which implies the following:

$$(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}})^{\dagger}(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}) \;=\; \boldsymbol{\Gamma}^{\frac{1}{2}}\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}.$$

Since the diagonal entries of $\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathrm{T}}\right)^{-1}\boldsymbol{A}$ are positive, the only possible equilibrium for the ODE is $\mathbf{0}$. However, when $\boldsymbol{\gamma}=\mathbf{0}$, $\mathrm{Rank}\{\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\}\neq m$ which is a contradiction. Hence, there is no equilibrium point with $\mathrm{Rank}\{\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\}=m$. Thus, (iii) holds, and the proof is complete. ■

Appendix B
Proof of Theorem 1

Before we prove the main theorem, we need two lemmas.

Lemma 2. *The solution set of $\boldsymbol{f}(\boldsymbol{\gamma}) = \mathbf{0}$ is $\{\mathbf{0}\} \cup \{\boldsymbol{\gamma} \in \mathbb{R}^N : \boldsymbol{A\Gamma A}^T = \boldsymbol{A\Gamma}_{opt}\boldsymbol{A}^T\}$, when $\mathbb{E}\left\{\boldsymbol{yy}^T\right\} = \boldsymbol{A\Gamma}_{opt}\boldsymbol{A}^T + \boldsymbol{R}$.*

Proof: From (44), we get

$$\boldsymbol{f}(\boldsymbol{\gamma}) = \mathrm{Diag}\big\{\boldsymbol{\Gamma A}^{\mathrm{T}}\left(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}\right)^{-1}\boldsymbol{A}\left(\boldsymbol{\Gamma}_{\mathrm{opt}} - \boldsymbol{\Gamma}\right) \boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}\right)^{-1}\boldsymbol{A\Gamma}\big\}. \quad (68)$$

Clearly, $\boldsymbol{\gamma} = \mathbf{0}$ is a zero of $\boldsymbol{f}(\boldsymbol{\gamma})$. Let us consider the solutions whose support is the vector $\boldsymbol{s} \in \{0,1\}^N$ and $\boldsymbol{s} \neq \mathbf{0}$, and let the number of nonzero entries in $\boldsymbol{s}$ be denoted by s. The union of the solutions over all possible supports gives the solution set. Let $\boldsymbol{\gamma_s} \in \mathbb{R}^{s\times 1}$ be the vector of nonzero entries of $\boldsymbol{\gamma}$ and $\boldsymbol{A_s} \in \mathbb{R}^{m\times s}$ be the matrix formed by restricting $\boldsymbol{A}$ to the s columns corresponding to the support $\boldsymbol{s}$. Let $\boldsymbol{B_s} = \left(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}\right)^{-\frac{1}{2}}\boldsymbol{A_s} \in \mathbb{R}^{m\times s}$, and $\boldsymbol{B} = \left(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}\right)^{-\frac{1}{2}}\boldsymbol{A} \in \mathbb{R}^{m\times N}$. Then, the reduced set of equations corresponding to $\boldsymbol{f}(\boldsymbol{\gamma}) = \mathbf{0}$ is given by

$$\mathrm{Diag}\left\{\boldsymbol{B}_{\boldsymbol{s}}^{\mathrm{T}}\boldsymbol{B_s\Gamma_s B}_{\boldsymbol{s}}^{\mathrm{T}}\boldsymbol{B_s}\right\} = \mathrm{Diag}\left\{\boldsymbol{B}_{\boldsymbol{s}}^{\mathrm{T}}\boldsymbol{B\Gamma}_{\mathrm{opt}}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B_s}\right\}, \quad (69)$$

where $\boldsymbol{\Gamma_s} = \mathrm{Diag}\left\{\boldsymbol{\gamma_s}\right\}$ is an invertible matrix. We note that the above system of equations is linear in the vector $\boldsymbol{\gamma_s}$, for any given fixed matrices $\boldsymbol{B_s}$ and $\boldsymbol{B}$. However, $\mathrm{Diag}\left\{\boldsymbol{B}_{\boldsymbol{s}}^{\mathrm{T}}\boldsymbol{B_s\Gamma_s B}_{\boldsymbol{s}}^{\mathrm{T}}\boldsymbol{B_s}\right\} = \left(\boldsymbol{B}_{\boldsymbol{s}}^{\mathrm{T}}\boldsymbol{B_s}\right) \circ \left(\boldsymbol{B}_{\boldsymbol{s}}^{\mathrm{T}}\boldsymbol{B_s}\right)\boldsymbol{\gamma_s}$, where $\circ$ represents the Hadamard product of matrices. Thus, the solution set of the system of equations is an affine space $\mathcal{U}_{\boldsymbol{s}}$ of dimension given by

$$\begin{aligned}\dim(\mathcal{U}_{\boldsymbol{s}}) &= s - \mathrm{Rank}\left\{\left(\boldsymbol{B}_{\boldsymbol{s}}^{\mathrm{T}}\boldsymbol{B_s}\right) \circ \left(\boldsymbol{B}_{\boldsymbol{s}}^{\mathrm{T}}\boldsymbol{B_s}\right)\right\} && (70)\\ &= s - \mathrm{Rank}\left\{\left(\boldsymbol{B_s} \odot \boldsymbol{B_s}\right)^{\mathrm{T}}\left(\boldsymbol{B_s} \odot \boldsymbol{B_s}\right)\right\} && (71)\\ &= s - \mathrm{Rank}\left\{\boldsymbol{B_s} \odot \boldsymbol{B_s}\right\}. && (72)\end{aligned}$$

We now consider another affine space $\mathcal{W}_{\boldsymbol{s}}$ of dimension $s - \mathrm{Rank}\left\{\boldsymbol{B_s} \odot \boldsymbol{B_s}\right\}$ given by the set of $\boldsymbol{\gamma_s}$ satisfying

$$\mathrm{vec}\left\{\boldsymbol{B_s\Gamma_s B}_{\boldsymbol{s}}^{\mathrm{T}}\right\} = \left(\boldsymbol{B_s} \odot \boldsymbol{B_s}\right)\boldsymbol{\gamma_s} = \mathrm{vec}\left\{\boldsymbol{B\Gamma}_{\mathrm{opt}}\boldsymbol{B}^{\mathrm{T}}\right\}. \quad (73)$$

It is easy to see that $\mathcal{W}_{\boldsymbol{s}} \subseteq \mathcal{U}_{\boldsymbol{s}}$ and $\dim(\mathcal{U}_{\boldsymbol{s}}) = \dim(\mathcal{W}_{\boldsymbol{s}})$, which implies $\mathcal{W}_{\boldsymbol{s}} = \mathcal{U}_{\boldsymbol{s}}$. Rearranging, we get, for $\boldsymbol{\gamma_s} \in \mathcal{U}_{\boldsymbol{s}}$,

$$\begin{aligned}&\left(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}\right)^{-\frac{1}{2}}\boldsymbol{A_s\Gamma_s A}_{\boldsymbol{s}}^{\mathrm{T}}\left(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}\right)^{-\frac{1}{2}}\\ &\quad = \left(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}\right)^{-\frac{1}{2}}\boldsymbol{A\Gamma}_{\mathrm{opt}}\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}\right)^{-\frac{1}{2}}. \quad (74)\end{aligned}$$

Thus, $\boldsymbol{A\Gamma A}^{\mathrm{T}} = \boldsymbol{A_s\Gamma_s A}_{\boldsymbol{s}}^{\mathrm{T}} = \boldsymbol{A\Gamma}_{\mathrm{opt}}\boldsymbol{A}^{\mathrm{T}}$, and $\mathcal{U}_{\boldsymbol{s}} \subseteq \{\boldsymbol{\gamma} : \boldsymbol{A\Gamma A}^{\mathrm{T}} = \boldsymbol{A\Gamma}_{\mathrm{opt}}\boldsymbol{A}^{\mathrm{T}}\}$, for all support sets $\boldsymbol{s} \neq \mathbf{0}$. From (68), it is easy to see that $\{\boldsymbol{\gamma} \in \mathbb{R}^N : \boldsymbol{A}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathrm{opt}}\right)\boldsymbol{A}^{\mathrm{T}} = \mathbf{0}\}$ satisfies $\boldsymbol{f}(\boldsymbol{\gamma}) = \mathbf{0}$. Therefore, $\underset{\boldsymbol{s}\in\{0,1\}^N\setminus\mathbf{0}}{\cup}\mathcal{U}_{\boldsymbol{s}} = \{\boldsymbol{\gamma} : \boldsymbol{A\Gamma A}^{\mathrm{T}} = \boldsymbol{A\Gamma}_{\mathrm{opt}}\boldsymbol{A}^{\mathrm{T}}\}$. Thus, we get that the solution set of $\boldsymbol{f}(\boldsymbol{\gamma}) = \mathbf{0}$ is $\{\mathbf{0}\} \cup \{\boldsymbol{\gamma} \in \mathbb{R}^N : \boldsymbol{A}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathrm{opt}}\right)\boldsymbol{A}^{\mathrm{T}} = \mathbf{0}\}$. ■

We define some notation to state the next lemma. The notation $\boldsymbol{X} \succ \mathbf{0}$ denotes that $\boldsymbol{X}$ is a positive definite matrix and $\boldsymbol{X} \succcurlyeq \mathbf{0}$ denotes that $\boldsymbol{X}$ is a positive semidefinite matrix.

Lemma 3. *The set $\mathbb{O} = \{\boldsymbol{\gamma} \in \mathbb{R}^N : \boldsymbol{A\Gamma A}^T + \boldsymbol{R} \succ \mathbf{0}\}$ is an open set and its closure is $\{\boldsymbol{\gamma} \in \mathbb{R}^N : \boldsymbol{A\Gamma A}^T + \boldsymbol{R} \succcurlyeq \mathbf{0}\}$.*

Proof: Let $\boldsymbol{\gamma} \in \mathbb{O}$. Then, $\boldsymbol{u}^{\mathrm{T}}(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R})\boldsymbol{u} > 0\ \forall \boldsymbol{u} \in \mathbb{R}^m \setminus \{\mathbf{0}\}$, and the minimum eigenvalue of $\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}$ is strictly greater than some $\beta > 0$. We need to show that there exists an $\epsilon > 0$ such that $\boldsymbol{A\widetilde{\Gamma}A}^{\mathrm{T}} + \boldsymbol{R}$ is positive definite for all $\widetilde{\boldsymbol{\gamma}}$ in the ϵ-neighborhood of $\boldsymbol{\gamma}$, i.e., $\|\boldsymbol{\gamma} - \widetilde{\boldsymbol{\gamma}}\| < \epsilon$.

For a given $\boldsymbol{u} \in \mathbb{R}^m \setminus \{\mathbf{0}\}$, if $\boldsymbol{u}^{\mathrm{T}}(\boldsymbol{A\widetilde{\Gamma}A}^{\mathrm{T}} + \boldsymbol{R})\boldsymbol{u} \geq \boldsymbol{u}^{\mathrm{T}}(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R})\boldsymbol{u}$, then $\boldsymbol{u}^{\mathrm{T}}(\boldsymbol{A\widetilde{\Gamma}A}^{\mathrm{T}} + \boldsymbol{R})\boldsymbol{u} > 0$. Otherwise,

$$\begin{aligned}\boldsymbol{u}^{\mathrm{T}}\left(\boldsymbol{A\widetilde{\Gamma}A}^{\mathrm{T}} + \boldsymbol{R}\right)\boldsymbol{u} &= \boldsymbol{u}^{\mathrm{T}}\left(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}\right)\boldsymbol{u}\\ &\quad - \left|\boldsymbol{u}^{\mathrm{T}}\boldsymbol{A}\left(\boldsymbol{\Gamma} - \widetilde{\boldsymbol{\Gamma}}\right)\boldsymbol{A}^{\mathrm{T}}\boldsymbol{u}\right| && (75)\\ &\geq \left(\beta - \|\boldsymbol{\Gamma} - \widetilde{\boldsymbol{\Gamma}}\|_2\|\boldsymbol{A}\|_2^2\right)\|\boldsymbol{u}\|^2 && (76)\\ &\geq \left(\beta - \epsilon\|\boldsymbol{A}\|_2^2\right)\|\boldsymbol{u}\|^2, && (77)\end{aligned}$$

where $\|\cdot\|_2$ denotes the induced l_2 norm. We can always find an $\epsilon > 0$ such that $\left(\beta - \epsilon\|\boldsymbol{A}\|_2^2\right) > 0$. Therefore, $\boldsymbol{u}^{\mathrm{T}}(\boldsymbol{A\widetilde{\Gamma}A}^{\mathrm{T}} + \boldsymbol{R})\boldsymbol{u} > 0\,\forall\,\boldsymbol{u} \in \mathbb{R}^m \setminus \{\mathbf{0}\}$, and thus $\mathbb{O}$ is an open set.

To prove the second part of the lemma, suppose the sequence $\boldsymbol{\gamma}_k \in \mathbb{O}$ converges to $\boldsymbol{\gamma}$. Then, for any vector $\boldsymbol{u} \in \mathbb{R}^m \setminus \{\mathbf{0}\}$, $\boldsymbol{u}^{\mathrm{T}}\left(\boldsymbol{A\Gamma}_k\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{R}\right)\boldsymbol{u}$ converges to $\boldsymbol{u}^{\mathrm{T}}\left(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}\right)\boldsymbol{u}$ by the continuity of the function. Therefore, $\boldsymbol{u}^{\mathrm{T}}\left(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}\right)\boldsymbol{u} \geq 0$ since $\boldsymbol{u}^{\mathrm{T}}\left(\boldsymbol{A\Gamma}_k\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{R}\right)\boldsymbol{u} > 0$, and thus $\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R} \succcurlyeq \mathbf{0}$. Conversely, if there is exists a $\boldsymbol{\gamma} \in \mathbb{R}^m$ such that $\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R} \succcurlyeq \mathbf{0}$, the sequence $\boldsymbol{\gamma}_k = \boldsymbol{\gamma} + (1/k)\mathbf{1}$ converges to $\boldsymbol{\gamma}$. We also note that $\boldsymbol{A\Gamma}_k\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{R} = \boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R} + (1/k)\boldsymbol{AA}^{\mathrm{T}} \succ \mathbf{0}$ since $\boldsymbol{A}$ has full row rank. Thus, there exists a sequence $\{\boldsymbol{\gamma}_k\} \in \mathbb{O}$ that converges to $\boldsymbol{\gamma}$. Hence, the proof is complete. ■

Proof of Theorem 1

We prove the convergence using [36, Theorem 2] which states that: Suppose $\boldsymbol{f}(\cdot)$ is a continuous vector field defined on an open set $\mathbb{O} \subset \mathbb{R}^N$ such that $\mathbb{G} = \{\boldsymbol{\gamma} \in \mathbb{O} : \boldsymbol{f}(\boldsymbol{\gamma}) = \mathbf{0}\}$ is a compact subset of $\mathbb{O}$. Then the distance of the sequence $\boldsymbol{\gamma}_k$ given by (40) to the set $\mathbb{G}$ converges to 0 *a.s.* provided:

(i) There exists a $\mathcal{C}^1$ function $V : \mathbb{O} \to \mathbb{R}_+$ such that
  a) $V(\boldsymbol{\gamma}) \to \infty$ if $\boldsymbol{\gamma} \to$ the boundary of $\mathbb{O}$ or $\|\boldsymbol{\gamma}\| \to \infty$
  b) $\langle\nabla_{\boldsymbol{\gamma}}V(\boldsymbol{\gamma}), \boldsymbol{f}(\boldsymbol{\gamma})\rangle < 0$, $\forall\boldsymbol{\gamma} \notin \mathbb{G}$.

(ii) $\boldsymbol{\gamma}_k$ belongs to a compact set of $\mathbb{O}$.

(iii) $\lim_{k\to\infty}\sum_{t=1}^{k}\frac{1}{t}\boldsymbol{e}_t$ exists and is finite.

To check whether assumptions (i)-(iii) hold in our case, we define the set $\mathbb{O} = \{\boldsymbol{\gamma} : \mathrm{Rank}\left\{\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}\right\} = m\}$ which is an open set by Lemma 3. Note that $\boldsymbol{f}$ is a continuous function of $\boldsymbol{\gamma}$. Also, the inverse image of the compact set $\{\mathbf{0}\}$ by $\boldsymbol{f}(\boldsymbol{\gamma})$ is compact, and hence, $\mathbb{G}$ is a compact subset of $\mathbb{O}$.

We define the $\mathcal{C}^1$ function in (i) as follows:

$$\begin{aligned}V(\boldsymbol{\gamma}) &= \mathrm{Tr}\left\{\left(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}\right)^{-1}\left(\boldsymbol{A\Gamma}_{\mathrm{opt}}\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{R}\right)\right\}\\ &\quad - \log\left|\left(\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}\right)^{-1}\left(\boldsymbol{A\Gamma}_{\mathrm{opt}}\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{R}\right)\right|. \quad (78)\end{aligned}$$

Note that $V(\boldsymbol{\gamma}) - m$ gives the KL divergence between $\mathcal{N}(\mathbf{0}, \boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R})$ and $\mathcal{N}(\mathbf{0}, \boldsymbol{A\Gamma}_{\mathrm{opt}}\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{R})$. Therefore, $V(\boldsymbol{\gamma}) \geq m > 0$. By Lemma 3, if $\boldsymbol{\gamma}$ is on the boundary of $\mathbb{O}$, at least one eigenvalue of $\boldsymbol{A\Gamma A}^{\mathrm{T}} + \boldsymbol{R}$ is zero. Hence, (ia) is satisfied. The gradient of $V(\boldsymbol{\gamma})$ is given by

$$\begin{aligned}\nabla_{\boldsymbol{\gamma}} V(\boldsymbol{\gamma}) &= \text{Diag}\left\{\boldsymbol{A}^{\mathrm{T}}\nabla_{\{\boldsymbol{A\Gamma A}^{\mathrm{T}}+\boldsymbol{R}\}}V\left(\boldsymbol{A\Gamma A}^{\mathrm{T}}+\boldsymbol{R}\right)\boldsymbol{A}\right\}\\ &= \text{Diag}\left\{\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A\Gamma A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}_{\text{opt}}\right)\right.\\ &\qquad\left.\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A\Gamma A}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\right\}. \quad (79)\end{aligned}$$

Substituting this relation in (44) gives $\boldsymbol{f}(\boldsymbol{\gamma}) = -\boldsymbol{\Gamma}^2\nabla_{\boldsymbol{\gamma}}V(\boldsymbol{\gamma})$. Therefore, for $\boldsymbol{\gamma} \in \mathbb{O}\setminus\mathbb{G}$, we have $\langle\nabla_{\boldsymbol{\gamma}}V(\boldsymbol{\gamma}), \boldsymbol{f}(\boldsymbol{\gamma})\rangle < 0$. Thus, (ib) is satisfied.

Assumptions (ii) and (iii) holds because of Proposition 1 and Lemma 1, respectively. Hence, $\boldsymbol{\gamma}_k$ converges to the set $\mathbb{G}$. Further, Proposition 1 shows that $\boldsymbol{\gamma}_k \geq 0$, and hence, we get that $\boldsymbol{\gamma}_k$ converges to the set $\{\boldsymbol{0}\}\cup\{\boldsymbol{\gamma}\in\mathbb{R}_+^N : \boldsymbol{A}(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}_{\text{opt}})\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{0}\}$. Finally, if Rank $\{\boldsymbol{A}\odot\boldsymbol{A}\} = N$, then $\left\{\boldsymbol{\gamma}\in\mathbb{R}_+^N : \boldsymbol{A}(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}_{\text{opt}})\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{0}\right\} = \left\{\boldsymbol{\gamma}_{\text{opt}}\right\}$. Thus, the proof is complete. ■

References

[1] R. Prasad, C. Murthy, and B. Rao, "Joint approximately sparse channel estimation and data detection in OFDM systems using sparse Bayesian learning," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3591–3603, Jul. 2014.

[2] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3010–3022, Aug. 2005.

[3] J. H. G. Ender, "On compressive sensing applied to radar," *Signal Processing*, vol. 90, no. 5, pp. 1402–1414, May 2010.

[4] I. F. Gorodnitsky, J. S. George, and B. D. Rao, "Neuromagnetic source imaging with FOCUSS: a recursive weighted minimum norm algorithm," *Electroencephalogr. Clin. Neurophysiol.*, vol. 95, no. 4, pp. 231–251, Oct. 1995.

[5] D. Wipf, J. Owen, H. Attias, K. Sekihara, and S. Nagarajan, "Robust Bayesian estimation of the location, orientation, and time course of multiple correlated neural sources using MEG," *NeuroImage*, vol. 49, no. 1, pp. 641–655, Jan. 2010.

[6] U. Gamper, P. Boesiger, and S. Kozerke, "Compressed sensing in dynamic MRI," *Magn. Reson. Med.*, vol. 59, no. 2, pp. 365–373, Feb. 2008.

[7] Z. Zhang, T.-P. Jung, S. Makeig, Z. Pi, and B. D. Rao, "Spatiotemporal sparse Bayesian learning with applications to compressed sensing of multichannel physiological signals," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 6, pp. 1186–1197, Nov. 2014.

[8] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2477–2488, Jul. 2005.

[9] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. part I: Greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, Mar. 2006.

[10] J. D. Blanchard, M. Cermak, D. Hanle, and Y. Jing, "Greedy algorithms for joint sparse recovery," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1694–1704, Apr. 2014.

[11] J. Ziniel and P. Schniter, "Efficient high-dimensional inference in the multiple measurement vector problem," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 340–354, Jan. 2013.

[12] D. Wipf and B. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3704–3716, Jul. 2007.

[13] R. Zdunek and A. Cichocki, "Improved M-FOCUSS algorithm with overlapping blocks for locally smooth sparse signals," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4752–4761, Oct. 2008.

[14] Z. Zhang and B. D. Rao, "Sparse signal recovery in the presence of correlated multiple measurement vectors," in *Proc. ICASSP*, Mar. 2010.

[15] N. Vaswani, "LS-CS-residual (LS-CS): Compressive sensing on least squares residual," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4108–4120, Aug 2010.

[16] X. Zhu, L. Dai, W. Dai, Z. Wang, and M. Moonen, "Tracking a dynamic sparse channel via differential orthogonal matching pursuit," in *Proc. MILCOM*, Oct. 2015.

[17] A. S. Charles, A. Balavoine, and C. J. Rozell, "Dynamic filtering of time-varying sparse signals via l_1 minimization," *IEEE Trans. Signal Process.*, vol. 64, no. 21, pp. 5644–5656, Nov 2016.

[18] N. Vaswani, "Kalman filtered compressed sensing," in *ICIP*, Oct. 2008.

[19] E. Karseras, K. K. Leung, and W. Dai, "Tracking dynamic sparse signals using hierarchical Bayesian Kalman filters," in *Proc. ICASSP*, May 2013.

[20] R. Chalasani and J. C. Principe, "Dynamic sparse coding with smoothing proximal gradient method," in *Proc. ICASSP*, May 2014.

[21] G. Joseph, C. R. Murthy, R. Prasad, and B. D. Rao, "Online recovery of temporally correlated sparse signals using multiple measurement vectors," in *Proc. Globecom*, Dec. 2015.

[22] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–214, Sep. 2001.

[23] B. Anderson and J. Moore, *Optimal filtering*. Courier Dover, 2005.

[24] V. Krishnamurthy and J. B. Moore, "On-line estimation of hidden markov model parameters based on the Kullback-Leibler information measure," *IEEE Trans. Signal Process.*, vol. 41, no. 8, pp. 2557–2573, Aug. 1993.

[25] R. Hunger, "Floating point operations in matrix-vector calculus," Munich University of Technology, TUM-LNS-TR-05-05, Tech. Rep. TUM-LNS-TR-05-05, Sep. 2007.

[26] D. Wipf and B. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[27] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," *IEEE Trans. Signal Process.*, vol. 5, no. 5, pp. 912–926, Sep. 2011.

[28] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, Jan. 2001.

[29] "Guidelines for evaluation of radio transmission technologies (RTTs) for IMT-2000," ITU, Tech. Rep. M.1225, Feb. 1997.

[30] C. Studer, C. Benkeser, S. Belfanti, and Q. Huang, "Design and implementation of a parallel turbo-decoder ASIC for 3GPP-LTE," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 8–17, Jan. 2011.

[31] J. Zyren and W. McCoy, "Overview of the 3GPP long term evolution physical layer," Freescale Semiconductor, Inc., Austin, TX, USA, Tech. Rep. 3GPPEVOLUTIONWP, Jul. 2007.

[32] Y. Zheng and C. Xiao, "Simulation models with correct statistical properties for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 6, no. 51, pp. 920–928, Jun. 2003.

[33] "Universal mobile telecommunications system (UMTS), selection procedures for the choice of radio transmission technologies of the UMTS," ETSI, Sophia-Antipolis, France, Tech. Rep. UMTS 21.01 version 3.0.1, Nov. 1997.

[34] D. Williams, *Probability with martingales*. Cambridge University Press, 1991.

[35] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

[36] B. Delyon, "General results on the convergence of stochastic algorithms," *IEEE Trans. Autom. Control*, vol. 41, no. 9, pp. 1245–1255, Sep. 1996.

Geethu Joseph received the B. Tech. degree in Electronics and Communication Engineering from National Institute of Technology, Calicut, India, in 2011, and the M.E. degree in Signal Processing from Indian Institute of Science, Bangalore, India, in 2014. She was awarded the Prof. I. S. N. Murthy medal for the year 2012-2014 for being the best M.E. student (signal processing) in the Department of Electrical Communication Engineering, IISc. She is currently working towards the Ph.D degree at the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India. Her research interests include statistical signal processing, adaptive filter theory, sparse Bayesian learning and compressive Sensing.

Chandra R. Murthy received the B.Tech. degree in Electrical Engineering from the Indian Institute of Technology Madras, Chennai, India, in 1998, the M.S. and Ph.D. degrees in Electrical and Computer Engineering from Purdue University, West Lafayette, IN and the University of California, San Diego, CA, in 2000 and 2006, respectively.

From 2000 to 2002, he worked as an engineer for Qualcomm Inc., San Jose, USA, where he worked on WCDMA baseband transceiver design and 802.11b baseband receivers. From 2006 to 2007, he worked as a staff engineer at Beceem Communications Inc., Bangalore, India on advanced receiver architectures for the 802.16e Mobile WiMAX standard. Currently, he is working as an Associate Professor in the department of Electrical Communication Engineering at the Indian Institute of Science, Bangalore, India. His research interests are in the areas of Cognitive Radio, Energy Harvesting Wireless Sensors and MIMO systems with channel-state feedback. He is an associate editor for the IEEE Transactions on Signal Processing, an editor for the IEEE Transactions on communications, an elected member of the IEEE SPCOM Technical Committee for the years 2014-16, and has been re-elected for the years 2016-19. He is currently the Chapter Chair of the IEEE Signal Processing Society, Bangalore Chapter. He served as an associate editor for the IEEE Signal Processing Letters during the years 2012-16.