

# Statistical Recovery of Simultaneously Sparse Time-Varying Signals from Multiple Measurement Vectors

Jun Won Choi, *Member, IEEE*, and Byonghyo Shim, *Senior Member, IEEE*

## Abstract

In this paper, we propose a new sparse signal recovery algorithm, referred to as sparse Kalman tree search (sKTS), that provides a robust reconstruction of the sparse vector when the sequence of correlated observation vectors are available. The proposed sKTS algorithm builds on expectation-maximization (EM) algorithm and consists of two main operations: 1) Kalman smoothing to obtain the *a posteriori* statistics of the source signal vectors and 2) greedy tree search to estimate the support of the signal vectors. Through numerical experiments, we demonstrate that the proposed sKTS algorithm is effective in recovering the sparse signals and performs close to the Oracle (genie-based) Kalman estimator.

## Index Terms

Compressed sensing, simultaneously sparse signal, multiple measurement vector, expectation-maximization (EM) algorithm, maximum likelihood estimation

Jun Won Choi is with Dept. of Electrical Engineering, Hanyang University, Seoul Korea, e-mail: junwchoi@hanyang.ac.kr.  
Byonghyo Shim is with Dept. of Electrical Engineering, Seoul National University, Seoul Korea, e-mail: bshim@snu.ac.kr.

# Statistical Recovery of Simultaneously Sparse Time-Varying Signals from Multiple Measurement Vectors

## I. INTRODUCTION

Over the years, there has been growing interest in the recovery of high dimensional signals from a small number of measurements. This new paradigm, so called compressed sensing (CS), relies on the fact that many naturally acquired high dimensional signals inherently have low dimensional structure. In fact, since many real world signals can be well approximated as sparse signals (i.e., only a few entries of signal vector are nonzero), CS techniques have been applied to a variety of applications including data compression, source localization, wireless sensor network, medical imaging, data mining, to name just a few.

Over the years, various signal recovery algorithms for CS have been proposed. Roughly speaking, these approaches are categorized into two classes. The first approach is based on a deterministic signal model, where an underlying signal is seen as a deterministic vector and the sparsity promoting cost function (e.g.,  $\ell_1$ -norm) is employed to solve the problem. These approaches include the basis pursuit (BP) [1], orthogonal matching pursuit (OMP) [2], [3], CoSaMP [4], and subspace pursuit [5]. The second approach is based on the probabilistic signal model, where the signal sparsity is described by the *a priori* distribution of the signal and Bayesian framework is employed in finding the sparse solution [6], [7].

When the multiple measurement vectors (MMV) from different source signals with common support are available, accuracy of the sparse signal recovery can be improved dramatically by performing joint processing of these vectors [8]–[14]. Since the algorithms based on MMV usually performs better than those relying on single measurement vector, many efforts have been made in recent years to develop an efficient sparse recovery algorithm. The MMV-based recovery algorithms targeted for the deterministic signal recovery include the mixed-norm solution [8], [9] and convex relaxation [10] while the probabilistic approaches include the MMV sparse Bayesian

learning (SBL) method [11], block-SBL [12], auto-regressive SBL [13], and Kalman filtering-based SBL (KSBL) [14].

In this work, we are primarily concerned with the MMV-based signal recovery problem when the observation vectors are sequentially acquired. To be specific, we express the  $N \times 1$  observation vector  $\mathbf{y}_n$  acquired at time index  $n$  as

$$\mathbf{y}_n = \mathbf{B}_n \mathbf{h}_n + \mathbf{w}_n, \quad (1)$$

where  $\mathbf{B}_n$  is the  $N \times M$  system matrix,  $\mathbf{h}_n$  is the  $M \times 1$  source signal vector, and  $\mathbf{w}_n$  are the  $N \times 1$  noise vector. We assume that  $\mathbf{w}_n$  is modeled as a zero mean complex Gaussian random vector, i.e.,  $\mathcal{CN}(0, \sigma_w^2 \mathbf{I})$ . Our goal in this setup is to estimate the source signal  $\mathbf{h}_n$  using the sequence of the observations  $\{\mathbf{y}_n\}$  when 1) the source signal  $\mathbf{h}_n$  is sparse (i.e., the number nonzero elements in  $\mathbf{h}_n$  is small) and 2) the dimension of the observation vector  $\mathbf{y}_n$  is smaller than that of the source vector ( $N \ll M$ ). In particular, we focus on the scenario where the nonzero elements of  $\mathbf{h}_n$  change over time with certain temporal correlations. In this scenario, we assume that correlated time-varying signals are well modeled by Gauss-Markov process. Note that this model is useful in capturing local dynamics of signals in linear estimation theory [24].

The main purpose of this paper is to propose a new statistical sparse signal estimation algorithm for the sequential observation model we just described. The underlying assumption used in our model is that the nonzero amplitude of the sparse signals is changing in time, leading to different signal realizations for each measurement vector, yet the support of the signal amplitude is slowly varying so that the support remains unchanged over certain consecutive measurement vectors. We henceforth refer to this model as *simultaneously sparse signal with locally common support* since the support of the sparse signal remains constant over the fixed interval under this assumption. Many of signal processing and wireless communication systems are characterized by this model. For example, this model matches well with the characteristics of multi-path fading channels for wireless communications where the channel impulse response  $\mathbf{h}_n$  should be estimated from the received signal  $\mathbf{y}_n$ . Fig. 1 shows a record of the channel impulse responses (CIR) of underwater acoustic channels (represented over the propagation delay and time domain) measured from the experiments conducted in Atlantic ocean in USA [21]. We observe that when compared to the amplitude of the channel taps, the sparsity structure of the CIR is varying slowly. Thus, we can

readily characterize this time-varying sparse signal using the correlated random process along with a deterministic binary parameter representing the existence of the signal.

In recovering the original signal vector  $\mathbf{h}_n$  from the measurement vectors, we use the modified expectation-maximization (EM) algorithm [22]. The proposed scheme, dubbed as sparse-Kalman-Tree-Search (sKTS), consists of two main operations: 1) Kalman smoothing to gather the *a posteriori* statistics of the source signals from individual measurement vector within the block of interest and 2) identification of the support of the sparse signal vector using a greedy tree search algorithm. Treating the problem to identify the sparsity structure of the source signal as a combinatorial search, we propose a simple yet effective greedy tree search algorithm that examines the small number of promising candidates among all sparsity parameter vectors in the tree.

There exist several approaches to estimate the time-varying sparse signals under MMV model. In [15], [16], reweighted  $\ell_1$  optimization has been modified for the sequential dynamic filtering. In [14], modified SBL algorithm has been suggested to adopt autoregressive modeling. In [19], EM-based adaptive filtering scheme has been proposed in the context of sparse channel estimation. Other than these, notable approaches include turbo approximate message passing (AMP) [18], Lasso-Kalman [20], and Kalman filtered CS [17]. We note that our work is distinct from these approaches in the following two aspects. First, in contrast to the previous efforts using continuous (real-valued) parameters to describe signal sparsity in [14]–[16], the proposed method employs the deterministic discrete (binary) parameter vector that captures the on-off structure of signal sparsity. Due to the use of deterministic parameter vector, an effort to deal with the probabilistic model on signal sparsity is unnecessary. Also, since the search space is discretized, identification of parameter vector is done by the efficient search algorithm. Second, while the recent work in [17] estimates signal amplitude using Kalman smoother and then identifies the support of sparse signal by thresholding of the innovation error norm, our work pursues direct estimation of the binary parameter vector using the modified EM algorithm. We note that a part of this paper was presented in [23]. The distinctive contribution of the present work is that the algorithm is developed in a more generic system model and practical issues (e.g., parameter estimation and iteration control) and real-time implementation issues are elaborated. Further, extensive simulations for the practical applications are conducted to demonstrate the superiority of the proposed method.

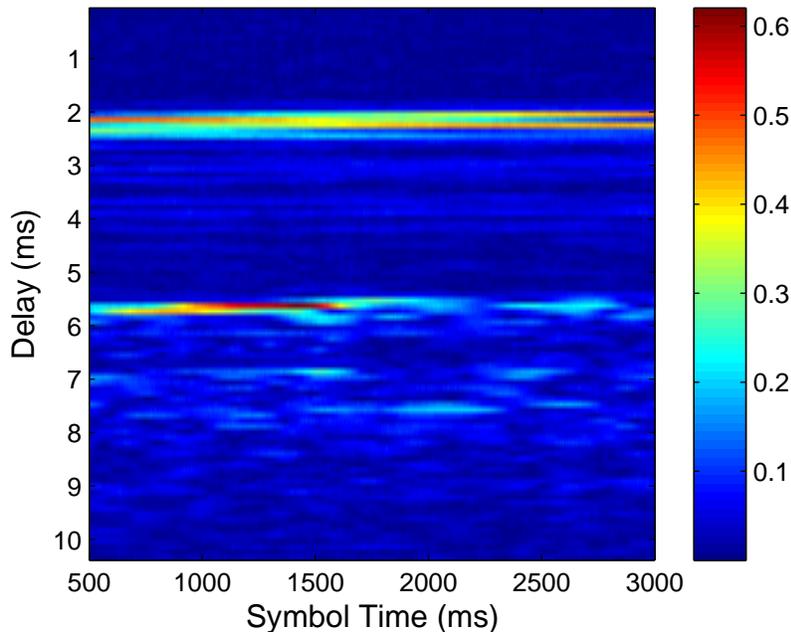


Fig. 1. A record of the channel impulse response of underwater acoustic channels measured off the coast of Martha's Vinyard, MA, USA.

The rest of this paper is organized as follows. In Section II, we briefly explain the sparse signal model and then present the proposed method. In Section III, we discuss the application of the proposed algorithm in the wireless channel estimation. In section IV, the simulation results are provided, and Section V concludes the paper.

Notation: Uppercase and lowercase letters written in boldface denote matrices and vectors, respectively. Superscripts  $(\cdot)^T$  and  $(\cdot)^H$  denote transpose and conjugate transpose (hermitian operator), respectively.  $\text{conj}(x)$  denotes the conjugation of the complex number  $x$ .  $\|\cdot\|_p$  indicates an  $\ell_p$ -norm of a vector. For the  $\ell_2$ -norm, we abbreviate a subscript  $p$  for simplicity.  $\text{diag}(\cdot)$  is a diagonal matrix having elements only on the main diagonal.  $\text{Re}(x)$  and  $\text{Im}(x)$  denote the real and imaginary parts of  $x$ , respectively.  $E[X]$  denotes the expectation of a random variable  $X$  and  $E[X|Y]$  denotes the conditional expectation of  $X$  given  $Y$ .  $E[X; \theta]$  means the expectation of  $X$  given the deterministic parameter  $\theta$ . The notations for covariance matrices are given by  $\text{Cov}(\mathbf{x}, \mathbf{y}) = E[\mathbf{x}\mathbf{y}^H] - E[\mathbf{x}]E[\mathbf{y}]^H$  and  $\text{Cov}(\mathbf{x}) = \text{Cov}(\mathbf{x}, \mathbf{x})$ .  $\text{Pr}(A)$  means the probability of the event  $A$ .  $\text{tr}(A)$  denotes a trace operation of the matrix  $A$ .  $A \odot B$  is the element-by-element

product (Hadamard product) of the matrices  $A$  and  $B$ .  $\mathbf{e}_i$  denotes the  $i$ th coordinate vector.

## II. PROPOSED SPARSE SIGNAL ESTIMATION TECHNIQUE

In this section, we consider the statistical estimation of the time-varying sparse signals from the sequentially collected observation vectors. As mentioned, our approach is based on the assumption that the support of the sparse signal varies slowly in time so that the multiple measurement vectors sharing common support can be used to improve the estimation quality of the sparse signals. In this section, we first describe the simultaneously sparse signal model and then present the proposed sparse signal estimation scheme.

### A. Simultaneously Sparse Signal Model

We express a time-varying sparse signal  $\mathbf{h}_n$  as a product of a vector of random processes  $\mathbf{s}_n$  describing the amplitudes of nonzero entries in  $\mathbf{h}_n$  and the vector  $\mathbf{c}_i = [c_{i,1}, \dots, c_{i,M}]^T$  indicating the existence of signal. That is,

$$\mathbf{h}_n = \text{diag}(\mathbf{c}_i)\mathbf{s}_n, \quad (2)$$

where  $i$  is the block index, the entry of  $\mathbf{c}_i$  is either 0 or 1 depending on the existence of the signal

$$c_{i,j} = \begin{cases} 1 & \text{if the } j\text{th entry of } \mathbf{h}_n \text{ exists} \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

and the time-varying amplitude  $\mathbf{s}_n \in \mathcal{C}^M$  is modeled as Gauss-Markov random process

$$\mathbf{s}_{n+1} = \mathbf{A}_n\mathbf{s}_n + \mathbf{v}_n, \quad (4)$$

where  $\mathbf{v}_n \in \mathcal{C}^M$  is the process noise vector ( $\sim \mathcal{CN}(0, \mathbf{V}_n)$ ) and  $\mathbf{A}_n \in \mathcal{C}^{M \times M}$  is the state update matrix. Note that the block index  $i$  is associated with the interval of the length  $T$ ,  $n \in [Ti + 1, T(i + 1)]$ . As mentioned, we assume that the support of the underlying sparse signals is locally time-invariant so that  $\mathbf{c}_i$  is constant in a block of consecutive measurement vectors. Using this together with the observation model in (1), we obtain the simultaneously

sparse signal model

$$\begin{aligned}\mathbf{s}_{n+1} &= \mathbf{A}_n \mathbf{s}_n + \mathbf{v}_n, \\ \mathbf{h}_n &= \text{diag}(\mathbf{c}_i) \mathbf{s}_n \\ \mathbf{y}_n &= \mathbf{B}_n \mathbf{h}_n + \mathbf{w}_n.\end{aligned}\tag{5}$$

Since  $\mathbf{h}_n$  follows Gaussian distribution for the given  $\mathbf{c}_i$ , the *a priori* distribution of the source signal  $\mathbf{h}_n$  can be described by

$$Pr(\mathbf{h}_n; \mathbf{c}_i) = \frac{1}{(2\pi)^M \det(\text{Cov}(\mathbf{h}_n))} \exp\left(-(\mathbf{h}_n - E[\mathbf{h}_n; \mathbf{c}_i])^H \text{Cov}(\mathbf{h}_n)^{-1} (\mathbf{h}_n - E[\mathbf{h}_n; \mathbf{c}_i])\right),\tag{6}$$

where

$$\begin{aligned}E[\mathbf{h}_n; \mathbf{c}_i] &= \text{diag}(\mathbf{c}_i) E[\mathbf{s}_n] \\ \text{Cov}(\mathbf{h}_n) &= \text{diag}(\mathbf{c}_i) \text{Cov}(\mathbf{s}_n) \text{diag}(\mathbf{c}_i).\end{aligned}\tag{7}$$

### B. Derivation of Statistical Sparse Signal Estimation

When the multiple measurement vectors  $\{\mathbf{y}_{T_{i+1}}, \dots, \mathbf{y}_{T(i+1)}\}$  in the  $i$ th block are available, the maximum likelihood (ML) estimate of  $\mathbf{c}_i$  is expressed as

$$\mathbf{c}_i^{\text{ML}} = \arg \max_{\mathbf{c}_i \in \{0,1\}^T, \sum_{j=1}^M c_{i,j} = K} \ln \Pr(\mathbf{y}_{1:T}; \mathbf{c}_i),\tag{8}$$

where  $\mathbf{y}_{1:T} = [\mathbf{y}_{T_{i+1}}, \dots, \mathbf{y}_{T(i+1)}]^T$  and  $K$  is the sparsity order (the number of nonzero entries) of  $\mathbf{h}_n$ . Note that the subscript  $1 : T$  denotes the set of time indices for the  $i$ th block. Note also that the ML estimate  $\mathbf{c}_i^{\text{ML}}$  is chosen among all candidates satisfying the sparsity constraint  $(\sum_{j=1}^M c_{i,j} = K)$ . Once  $\mathbf{c}_i^{\text{ML}}$  is obtained, we can estimate the amplitude vectors  $\mathbf{s}_n$  assuming that the signal support specified by  $\mathbf{c}_i^{\text{ML}}$  is true. Well known linear minimum mean square error (LMMSE) estimator (e.g. Kalman smoother) can be used to estimate  $\mathbf{s}_n$  and then  $\mathbf{c}_i^{\text{ML}}$  and the estimate of  $\mathbf{s}_n$  are combined to produce a final estimate of  $\mathbf{h}_n$ . Note that if the estimation of  $\mathbf{c}_i$  is correct, we can obtain the best achievable estimate of  $\mathbf{h}_n$ , which is equivalent to the solution attainable by so called ‘‘Oracle estimator’’. Since the ML problem in (8) involves the marginalization over all possible combinations of the latent variables  $\mathbf{s}_{1:T}$ , finding out the solution using the direct approach would be computationally unmanageable. Perhaps, a better way to deal

with the problem at hand is to use the EM algorithm. Recall that the EM algorithm is an efficient means to find out the ML estimate or maximum a posteriori (MAP) estimate of statistical signal model in the presence of unobserved latent variables. The EM algorithm generates a sequence of estimates  $\hat{\mathbf{c}}_i^{(l)}$ ,  $l = 0, 1, 2, \dots$  by alternating two major steps (E-step and M-step), which are given, respectively

- **Expectation step (E-step)**

$$Q(\mathbf{c}_i; \hat{\mathbf{c}}_i^{(l)}) = E \left[ \ln \Pr(\mathbf{y}_{1:T}, \mathbf{s}_{1:T}; \mathbf{c}_i) \middle| \mathbf{y}_{1:T}; \hat{\mathbf{c}}_i^{(l)} \right], \quad (9)$$

- **Maximization step (M-step)**

$$\hat{\mathbf{c}}_i^{(l+1)} = \arg \max_{\mathbf{c}_i \in \{0,1\}^M, \sum_{j=1}^M c_{i,j} = K} Q(\mathbf{c}_i; \hat{\mathbf{c}}_i^{(l)}), \quad (10)$$

where  $\hat{\mathbf{c}}_i^{(l)}$  is the estimate of  $\mathbf{c}_i$  at the  $l$ -th iteration. Although one cannot guarantee finding out the global optimal solution of (8) using the EM algorithm, we will empirically show that  $\mathbf{c}_i$  can be estimated accurately with a proper initialization of  $\mathbf{c}_i^{(0)}$  (see Section IV).

### C. The E-step

The goal of the E-step is to obtain a simple expression of the cost metric  $Q(\mathbf{c}_i, \hat{\mathbf{c}}_i^{(l)})$  using the simultaneously sparse signal model. First,  $\ln \Pr(\mathbf{y}_{1:T}, \mathbf{s}_{1:T}; \mathbf{c}_i)$  is expressed as

$$\ln \Pr(\mathbf{y}_{1:T}, \mathbf{s}_{1:T}; \mathbf{c}_i) = \ln \Pr(\mathbf{y}_{1:T} | \mathbf{s}_{1:T}; \mathbf{c}_i) + \ln \Pr(\mathbf{s}_{1:T}; \mathbf{c}_i) \quad (11)$$

$$= \sum_{n=T^{(i)}+1}^{T^{(i+1)}} \ln \Pr(\mathbf{y}_n | \mathbf{s}_n; \mathbf{c}_i) + \sum_{n=T^{(i)}+1}^{T^{(i+1)}} \ln \Pr(\mathbf{s}_n | \mathbf{s}_{n-1}) \quad (12)$$

Noting that  $\Pr(\mathbf{y}_n | \mathbf{s}_n; \mathbf{c}_i) \sim \mathcal{CN}(\mathbf{B}_n \text{diag}(\mathbf{c}_i) \mathbf{s}_n, \sigma_w^2 \mathbf{I})$  and  $\Pr(\mathbf{s}_n | \mathbf{s}_{n-1}) \sim \mathcal{CN}(\mathbf{A}_{n-1} \mathbf{s}_{n-1}, \mathbf{V}_{n-1})$ , we have

$$\begin{aligned} \ln \Pr(\mathbf{y}_{1:T}, \mathbf{s}_{1:T}; \mathbf{c}_i) &= - \sum_{n=T^{(i)}+1}^{T^{(i+1)}} \frac{1}{\sigma_w^2} \|\mathbf{y}_n - \mathbf{B}_n \text{diag}(\mathbf{c}_i) \mathbf{s}_n\|^2 \\ &\quad - \sum_{n=T^{(i)}+1}^{T^{(i+1)}} (\mathbf{s}_n - \mathbf{A}_{n-1} \mathbf{s}_{n-1})^H \mathbf{V}_{n-1}^{-1} (\mathbf{s}_n - \mathbf{A}_{n-1} \mathbf{s}_{n-1}) + C \end{aligned} \quad (13)$$

$$= - \sum_{n=T^{(i)}+1}^{T^{(i+1)}} \frac{1}{\sigma_w^2} \|\mathbf{y}_n - \mathbf{B}_n \text{diag}(\mathbf{c}_i) \mathbf{s}_n\|^2 + C', \quad (14)$$

where  $C$  and  $C'$  are the terms independent of  $\mathbf{c}_i$ . From (9) and (14), we further have (see Appendix A)

$$Q(\mathbf{c}_i; \hat{\mathbf{c}}_i^{(l)}) = C'' + \frac{1}{\sigma_w^2} \sum_{n=T_i+1}^{T(i+1)} \left\{ 2\text{Re} \left( \mathbf{y}_n^H \mathbf{B}_n \text{diag}(\mathbf{c}_i) E \left[ \mathbf{s}_n \middle| \mathbf{y}_{1:T}; \hat{\mathbf{c}}_i^{(l)} \right] \right) - \text{tr} \left[ \mathbf{B}_n \text{diag}(\mathbf{c}_i) E \left[ \mathbf{s}_n \mathbf{s}_n^H \middle| \mathbf{y}_{1:T}; \hat{\mathbf{c}}_i^{(l)} \right] \text{diag}(\mathbf{c}_i) \mathbf{B}_n^H \right] \right\}. \quad (15)$$

Let  $\hat{\mathbf{s}}_{n|1:T}$  and  $\Sigma_{n|1:T}$  be the conditional mean and covariance of  $\mathbf{s}_n$  when  $\mathbf{y}_{1:T}$  and  $\hat{\mathbf{c}}_i^{(l)}$  are given, i.e.,

$$\begin{aligned} \hat{\mathbf{s}}_{n|1:T} &= E \left[ \mathbf{s}_n \middle| \mathbf{y}_{1:T}; \hat{\mathbf{c}}_i^{(l)} \right] \\ \Sigma_{n|1:T} &= \text{Cov} \left[ \mathbf{s}_n \middle| \mathbf{y}_{1:T}; \hat{\mathbf{c}}_i^{(l)} \right]. \end{aligned}$$

Now We turn to the estimation of the *a posteriori* statistics  $\hat{\mathbf{s}}_{n|1:T}$  and  $\Sigma_{n|1:T}$ . In our work, we estimate  $\hat{\mathbf{s}}_{n|1:T}$  and  $\Sigma_{n|1:T}$  using Kalman smoothing [24]. When  $\hat{\mathbf{c}}_i^{(l)}$  is given, from (5), the system equations for Kalman smoothing becomes

$$\begin{aligned} \mathbf{s}_{n+1} &= \mathbf{A}_n \mathbf{s}_n + \mathbf{v}_n \\ \mathbf{y}_n &= \mathbf{B}_n \text{diag}(\hat{\mathbf{c}}_i^{(l)}) \mathbf{s}_n + \mathbf{w}_n. \end{aligned} \quad (16)$$

We employ the fixed-interval Kalman smoothing algorithm performing sequential estimation of  $\hat{\mathbf{s}}_{n|1:T}$  and  $\Sigma_{n|1:T}$  via forward and backward recursions in a block of observations  $\mathbf{y}_{1:T}$ . Let  $\hat{\mathbf{s}}_{n|j}$  and  $\Sigma_{n|j}$  be the conditional mean and covariance given the first  $j$  observation vectors, i.e.,  $\hat{\mathbf{s}}_{n|j} = E \left[ \mathbf{s}_n \middle| \mathbf{y}_{T_i+1}, \dots, \mathbf{y}_{T_i+j}; \hat{\mathbf{c}}_i^{(l)} \right]$  and  $\Sigma_{n|j} = \text{Cov} \left[ \mathbf{s}_n \middle| \mathbf{y}_{T_i+1}, \dots, \mathbf{y}_{T_i+j}; \hat{\mathbf{c}}_i^{(l)} \right]$ , then the fixed-interval Kalman smoothing algorithm is summarized as

- **Forward recursion rule:**

$$\hat{\mathbf{s}}_{n|n-1} = \mathbf{A}_{n-1} \hat{\mathbf{s}}_{n-1|n-1} \quad (17)$$

$$\Sigma_{n|n-1} = \mathbf{A}_{n-1} \Sigma_{n-1|n-1} \mathbf{A}_{n-1}^H + \mathbf{V}_{n-1} \quad (18)$$

$$K_n = \Sigma_{n|n-1} \text{diag}(\hat{\mathbf{c}}_i^{(l)}) \mathbf{B}_n^H \left( \mathbf{B}_n \text{diag}(\hat{\mathbf{c}}_i^{(l)}) \Sigma_{n|n-1} \text{diag}(\hat{\mathbf{c}}_i^{(l)}) \mathbf{B}_n^H + \sigma_w^2 \mathbf{I} \right)^{-1} \quad (19)$$

$$\hat{\mathbf{s}}_{n|n} = \hat{\mathbf{s}}_{n|n-1} + K_n (\mathbf{y}_n - \mathbf{B}_n \text{diag}(\hat{\mathbf{c}}_i^{(l)}) \hat{\mathbf{s}}_{n|n-1}) \quad (20)$$

$$\Sigma_{n|n} = \left( \mathbf{I} - K_n \mathbf{B}_n \text{diag}(\hat{\mathbf{c}}_i^{(l)}) \right) \Sigma_{n|n-1}, \quad (21)$$

• **Backward recursion rule:**

$$\mathbf{S}_n = \Sigma_{n|n} \mathbf{A}_n \Sigma_{n+1|n}^{-1} \quad (22)$$

$$\widehat{\mathbf{s}}_{n|1:T} = \widehat{\mathbf{s}}_{n|n} + \mathbf{S}_n (\widehat{\mathbf{s}}_{n+1|1:T} - \mathbf{A}_n \widehat{\mathbf{s}}_{n|n}) \quad (23)$$

$$\Sigma_{n|1:T} = \Sigma_{n|n} + \mathbf{S}_n (\Sigma_{n+1|1:T} - \Sigma_{n+1|n})^H \mathbf{S}_n^H. \quad (24)$$

Using  $\widehat{\mathbf{s}}_{n|1:T}$  and  $\Sigma_{n|1:T}$ ,  $Q(\mathbf{c}_i; \hat{\mathbf{c}}_i^{(l)})$  can be rewritten as

$$Q(\mathbf{c}_i; \hat{\mathbf{c}}_i^{(l)}) = C''' + \frac{1}{\sigma_w^2} \sum_{n=Ti+1}^{T(i+1)} \left\{ 2\text{Re}(\mathbf{y}_n^H \mathbf{B}_n \text{diag}(\mathbf{c}_i) \widehat{\mathbf{s}}_{n|1:T}) \right. \\ \left. - \text{tr}[\mathbf{B}_n \text{diag}(\mathbf{c}_i) (\Sigma_{n|1:T} + \widehat{\mathbf{s}}_{n|1:T} \widehat{\mathbf{s}}_{n|1:T}^H) \text{diag}(\mathbf{c}_i) \mathbf{B}_n^H] \right\}. \quad (25)$$

Note that the second term in the right-hand side of (25) is expressed as (see Appendix B)

$$\text{tr}[\mathbf{B}_n \text{diag}(\mathbf{c}_i) (\Sigma_{n|1:T} + \widehat{\mathbf{s}}_{n|1:T} \widehat{\mathbf{s}}_{n|1:T}^H) \text{diag}(\mathbf{c}_i) \mathbf{B}_n^H] \\ = \mathbf{c}_i^T (\text{conj}(\mathbf{B}_n^H \mathbf{B}_n) \odot (\Sigma_{n|1:T} + \widehat{\mathbf{s}}_{n|1:T} \widehat{\mathbf{s}}_{n|1:T}^H)) \mathbf{c}_i. \quad (26)$$

From (25) and (26), we have

$$Q(\mathbf{c}_i; \hat{\mathbf{c}}_i^{(l)}) = C''' + \frac{1}{\sigma_w^2} \sum_{n=Ti+1}^{T(i+1)} \left\{ 2\text{Re}(\mathbf{y}_n^H \mathbf{B}_n \text{diag}(\widehat{\mathbf{s}}_{n|1:T})) \mathbf{c}_i \right. \\ \left. - \mathbf{c}_i^T (\text{conj}(\mathbf{B}_n^H \mathbf{B}_n) \odot (\Sigma_{n|1:T} + \widehat{\mathbf{s}}_{n|1:T} \widehat{\mathbf{s}}_{n|1:T}^H)) \mathbf{c}_i \right\}. \quad (27)$$

Further, by denoting

$$\mathbf{d}_i^T = \sum_{n=Ti+1}^{T(i+1)} 2\text{Re}(\mathbf{y}_n^H \mathbf{B}_n \text{diag}(\widehat{\mathbf{s}}_{n|1:T})) \quad (28)$$

$$\Phi_i = \sum_{n=Ti+1}^{T(i+1)} (\text{conj}(\mathbf{B}_n^H \mathbf{B}_n) \odot (\Sigma_{n|1:T} + \widehat{\mathbf{s}}_{n|1:T} \widehat{\mathbf{s}}_{n|1:T}^H)), \quad (29)$$

we have

$$Q(\mathbf{c}_i; \hat{\mathbf{c}}_i^{(l)}) = C''' + \frac{1}{\sigma_w^2} \mathbf{d}_i^T \mathbf{c}_i - \frac{1}{\sigma_w^2} \mathbf{c}_i^T \Phi_i \mathbf{c}_i. \quad (30)$$

In summary, the E-step performs the Kalman smoothing operation in (17)-(24) to estimate  $\widehat{\mathbf{s}}_{n|1:T}$  and  $\Sigma_{n|1:T}$  and also operations in (28) and (29) to compute  $\mathbf{d}_i^T$  and  $\Phi_i$  used in the computation of  $Q(\mathbf{c}_i; \hat{\mathbf{c}}_i^{(l)})$ .

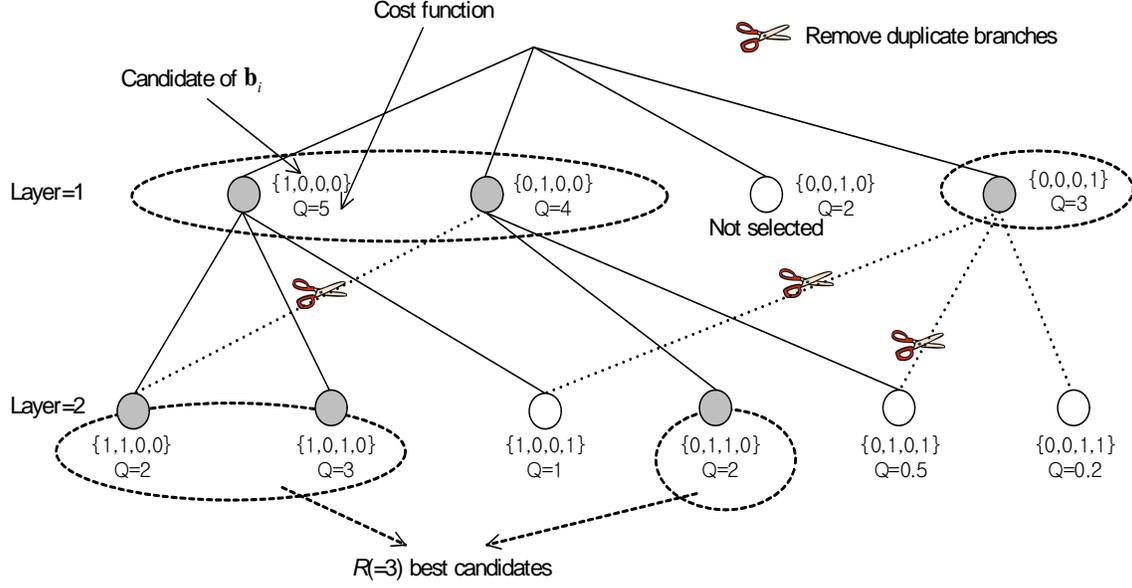


Fig. 2. Illustration of the proposed greedy tree search algorithm for the size of the source vector  $M = 4$  and the sparsity order  $K = 2$ . Since the number of candidates chosen in each layer is set to  $R = 3$ ,  $R(= 3)$  nodes with the largest  $Q$  values are chosen in each layer.

#### D. M-Step

In the M-step, we find  $\mathbf{c}_i$  maximizing  $Q(\mathbf{c}_i; \hat{\mathbf{c}}_i^{(l)})$  in (30) as

$$\hat{\mathbf{c}}_i^{(l+1)} = \arg \max_{\mathbf{c}_i \in \{0,1\}^M, \sum_{j=1}^M c_{i,j} = K} \hat{Q}(\mathbf{c}_i; \hat{\mathbf{c}}_i^{(l)}), \quad (31)$$

where  $\hat{Q}(\mathbf{c}_i; \hat{\mathbf{c}}_i^{(l)}) = (\mathbf{d}_i^T \mathbf{c}_i - \mathbf{c}_i^T \Phi_i \mathbf{c}_i)$ . In finding  $\hat{\mathbf{c}}_i^{(l+1)}$ , we need to check all possible combinations satisfying the sparsity constraint  $\sum_{j=1}^M c_{i,j} = K$ . Since this brute force search is prohibitive for practical values of  $M$ , we consider a computationally efficient search algorithm returning a sub-optimal solution to the problem in (31). The proposed approach, which in essence builds on the greedy tree search algorithm, examines candidate vectors to find out the most promising candidate of  $\mathbf{c}_i$  in a cost effective manner. The tree structure used for the proposed greedy search algorithm is illustrated in Fig. 2. Starting from a root node of the tree (associated with  $\mathbf{c}_i = [0, \dots, 0]^T$ ), we construct the layer of the tree one at each iteration. In the first layer of the tree, only one entry of  $\mathbf{c}_i$  is set to one. For example, the nodes in the first layer of the tree are expressed as  $\mathbf{c}_i^1 = [1, 0, \dots, 0]^T, \dots, \mathbf{c}_i^M = [0, \dots, 0, 1]^T$ . As the layer increases, one additional entry is set to one and thus  $K$  entries of  $\mathbf{c}_i$  are set to one in the  $K$ -th layer ( $\|\mathbf{c}_i\|_0 = K$ ) (see

TABLE I  
SUMMARY OF THE GREEDY TREE SEARCH ALGORITHM

Input: $\mathbf{d}_i$ and survival list $\Phi_i$
Initialization: Start with $\Theta = \{[0, \dots, 0]^T\}$ .
for $k = 1 : K$
for $j = 1 : R$
for $m = 1 : M$
Let $\theta_j$ be the $j$ th element of $\Theta$ .
If the $m$ -th entry is already one, skip the loop. Otherwise, set the $m$ -th entry of $\theta_j$ to one.
For $Q(\mathbf{x}) \triangleq \mathbf{d}_i^T \mathbf{x} - \mathbf{x}^T \Phi_i \mathbf{x}$ , evaluate $Q(\theta_j) - Q(\theta)$ for all $\theta \in \Theta$ .
If $Q(\theta_j) - Q(\theta) = 0$ for any $\theta \in \Theta$ , then the candidate $\theta_j$ is duplicate node and hence we remove it.
If $Q(\theta_j) > \min_{\theta \in \Theta} Q(\theta)$ , add $\theta_j$ into $\Theta$ .
end
end
end
Output: $\hat{\mathbf{c}}_i^{(l+1)} = \arg \max_{\theta \in \Theta} Q(\theta)$ .

Fig. 2). At each layer of the tree, we evaluate the cost function  $\hat{Q}(\mathbf{c}_i; \hat{\mathbf{c}}_i^{(l)})$  for each node and then choose the  $R$  best nodes whose cost function is maximal. The rest of nodes are discarded from the tree. The candidates of  $\mathbf{c}_i$  associated with the  $R$  best nodes at each layer are called “survival list”. For each node in the survival list, we construct the  $M - 1$  child nodes in the second layer by setting one additional entry of  $\mathbf{c}_i$  to one<sup>1</sup>. Note that since we do not distinguish the order of the bit assertion in  $\mathbf{c}_i$ , two or more nodes might represent the same realization of  $\mathbf{c}_i$  during this process (see Fig. 2). When duplicate nodes are identified, we keep only one and discard the rest from the tree. After removing all duplicate nodes, we choose the  $R$  best nodes and then move on to the next layer. This process is repeated until the tree reaches the bottom layer of the tree. We note that since the tree search complexity is proportional to the depth of the tree ( $K$ ), the dimension of source vector ( $M$ ), and the number of nodes being selected ( $R$ ), one can easily show that the complexity of the proposed tree search is  $O(MRK)$ . Hence, with small values of  $R$  and  $K$ , the computational complexity is reasonably small and proportional to

<sup>1</sup>For example, if  $\mathbf{c}_i = [1, 0, \dots, 0]^T$  is in the survival list, then the child nodes of  $\mathbf{c}_i$  becomes  $\mathbf{c}_i = [1, 1, \dots, 0]^T, \dots, \mathbf{c}_i = [1, 0, \dots, 1]^T$ .

TABLE II  
SUMMARY OF THE PROPOSED ALGORITHM

Input: $\mathbf{y}_{1:T}$ , $\mathbf{B}_{1:T}$ for the $i$ th block
STEP 1: Set $l = 0$ . Start with $\hat{\mathbf{c}}_i^{(0)} = [1, \dots, 1]^T$ and $K^{(0)}$ .
STEP 2: Run Kalman smoother in (17) to (24).
STEP 3: Calculate $\mathbf{d}_i$ and $\Phi_i$ in (28) and (29).
STEP 4: Obtain $\hat{\mathbf{c}}_i^{(l+1)}$ by running the greedy tree search.
STEP 5: Set $K \leftarrow K^{(l+1)}$ , $l \leftarrow l + 1$ and then go back to STEP 2.
If the estimate of $\mathbf{c}_i$ does not change (i.e., $\hat{\mathbf{c}}_i^{(l+1)} = \hat{\mathbf{c}}_i^{(l)}$ ) or the number of iterations reaches the limit, go to STEP 6.
STEP 6: Run Kalman smoother using $\hat{\mathbf{c}}_i^{(L)}$ .
STEP 7: Obtain the final signal estimate from (32).
Output: $\hat{\mathbf{h}}_{1:T}$

the dimension  $M$  of the source signal vector. The proposed tree search algorithm is summarized in Table I.

It is worth mentioning that one important issue to be considered is how to estimate the sparsity order  $K$ . One simple way is to use the simple correlation method, where the observation vectors are correlated with the column vectors of  $\mathbf{B}_n$  and  $K$  is chosen as the number of the column vectors whose absolute correlation exceeds the predefined threshold. While this approach is simple to implement, the performance might be affected by the estimation quality of  $K$ . One can alternatively consider a simple heuristic that terminates the tree search when a big drop in the cost metric  $Q(\mathbf{c}_i, \hat{\mathbf{c}}_i^{(l)})$  is observed.

After all iterations are finished (i.e.  $l = L$ ) and  $\hat{\mathbf{c}}_i^{(L)}$  is obtained, we use the Kalman smoother once again to compute  $\hat{\mathbf{s}}_{n|1:T}$  using the newly updated  $\hat{\mathbf{c}}_i^{(L)}$ . The final estimate of  $\mathbf{h}_n$  is expressed as

$$\hat{\mathbf{h}}_n = \text{diag}(\hat{\mathbf{c}}_i^{(L)})\hat{\mathbf{s}}_{n|1:T}. \quad (32)$$

### E. Iteration Control

In this subsection, we discuss how to configure the control parameters in performing the iterations of the EM algorithm. In each iteration, the proposed scheme estimates the support  $\mathbf{c}_i$

of the sparse signal vector under the sparsity constraint  $\sum_{j=1}^M c_{i,j} = K$ . Since the tree search to identify the support of  $\mathbf{h}_n$  is based on the greedy principle, it is possible that the support elements might not be accurately identified, especially for the initial iterations where the cost metric  $\widehat{Q}(\mathbf{c}_i; \widehat{\mathbf{c}}_i^{(l)})$  is not so accurate. In order to reduce the chance of missing nonzero entries of sparse vector in early iterations, we search for the sparse signal vector  $\mathbf{c}_i$  under relaxed sparsity constraint in the beginning and then gradually reduce the sparsity order  $K$  as the number of iterations increases. Let the sparsity order parameter used for the  $l$ th iteration be  $K^{(l)}$ . Then, we use sufficiently large value of  $K^{(1)}$  initially<sup>2</sup> and then decreases  $K^{(l)}$  monotonically (i.e.,  $K^{(l)} \geq K^{(l+1)}$ ) until  $K^{(l)}$  equals the target sparsity order  $K$ . In doing so, we can substantially reduce the chance of missing support elements and at the same time gradually improve the estimation quality of  $\mathbf{c}_i$ . The summary of the proposed algorithm is presented in Table II.

#### F. Estimation of $\mathbf{V}_n$

Although the proposed algorithm is designed based on the dynamic sparse model in (5),  $\mathbf{A}_n$  and  $\mathbf{V}_n$  are generally unknown in practice. We are often interested in the scenario where the elements of the signal vector  $\mathbf{s}_n$  are uncorrelated with each other and  $\mathbf{A}_n$  and  $\mathbf{V}_n$  are fixed over the block of interval  $T$ . Then,  $\mathbf{A}_n$  and  $\mathbf{V}_n$  have a diagonal form, i.e.,  $\mathbf{A}_n = \text{diag}(\alpha_1, \dots, \alpha_M)$  and  $\mathbf{V}_n = \text{diag}((1 - \alpha_1^2)\sigma_{s,1}^2, \dots, (1 - \alpha_M^2)\sigma_{s,M}^2)$ . In general,  $\alpha_j$  can be determined using the temporal correlations of the  $j$ th element of  $\mathbf{s}_n$ , which can be known a priori or estimated from the data separately. While the estimation of  $\alpha_i$  is relatively easy, such is not the case for  $\sigma_{s,j}^2$ . One accurate but computationally expensive approach is to estimate these variances using EM formulation. In this work, we do not pursue this approach due to complexity concern and compute a rough estimate of  $\sigma_{s,j}^2$  from  $\frac{1}{T} \sum_{n=Ti+1}^{T(i+1)} \mathbf{b}_{n,j}^H \mathbf{y}_n / \|\mathbf{b}_{n,j}\|_2^2$ , where  $\mathbf{b}_{n,j}$  is the  $j$ th column of the matrix  $\mathbf{B}_n$ . After running all iterations, the refined estimate of  $\mathbf{V}_n$  can be obtained by taking sample covariance matrix of the estimated signal vectors in the processing block and used for the final Kalman smoothing step described in Section II-C.

<sup>2</sup>Initially, we set  $\mathbf{c}_i^{(0)} = [1, \dots, 1]$  (i.e.,  $K^{(0)} = M$ ) since we have no knowledge on the sparsity structure of the source signal vector in the beginning.

### G. Real-time Implementation

Since the proposed algorithm we described in the previous subsections performs batch processing by running several iterations of E-step and M-step in a block, it might not be suitable for real-time applications. By slightly modifying the algorithm, we can reduce latency and also speed up the operations substantially. The main idea behind this modification is to return the estimate of the source signal immediately after the new measurement vector is provided. In doing so, we can process the block seamlessly without waiting for the reception of whole block of observations. First, instead of Kalman smoother, we employ the Kalman filter to conduct the operations from (17) to (21) in a forward direction. In order to ensure the real-time processing, we need to use multiple Kalman filters, where each Kalman filter corresponds to single iteration of the EM algorithm. For the sake of simplicity, we here consider two Kalman filters as an example<sup>3</sup>. In the first Kalman filter, we do not know the signal existence vector  $\mathbf{c}_i$  so that we set  $\mathbf{c}_i = [1, \dots, 1]^T$  and run the Kalman filter. Once  $\hat{\mathbf{s}}_{n|n}$  and  $\Sigma_{n|n}$  are obtained, we replace the computation of  $\mathbf{d}_n^H$  and  $\Phi_n$  in (28) and (29) by an auto-regressive update rule

$$\tilde{\mathbf{d}}_n^H = (1 - \alpha)\tilde{\mathbf{d}}_{n-1}^H + \alpha 2\text{Re}(\mathbf{y}_{n-1}^H \mathbf{B}_{n-1} \text{diag}(\hat{\mathbf{s}}_{n-1|n-1})) \quad (33)$$

$$\tilde{\Phi}_n = (1 - \alpha)\tilde{\Phi}_{n-1} + \alpha (\text{conj}(\mathbf{B}_{n-1}^H \mathbf{B}_{n-1}) \odot (\Sigma_{n-1|n-1} + \hat{\mathbf{s}}_{n-1|n-1} \hat{\mathbf{s}}_{n-1|n-1}^H)), \quad (34)$$

where  $\alpha$  is a forgetting factor controlling the speed of update. Note that by using (33) and (34) instead of (28) and (29), we can compute approximation of  $\mathbf{d}_n^H$  and  $\Phi_n$  on the fly whenever the new measurement vector is available. Once  $\tilde{\mathbf{d}}_n^H$  and  $\tilde{\Phi}_n$  are obtained, we next identify the signal existence indication vector  $\mathbf{c}_i$  using the greedy tree search described in Section II-D. Using the newly obtained estimate  $\hat{\mathbf{c}}_i$ , the second Kalman filter generates  $\hat{\mathbf{s}}_{n|n}$ . By multiplying this and  $\hat{\mathbf{c}}_i$ , we get the final estimate of  $\mathbf{h}_n$ . Note that it is straightforward to employ more than two Kalman filters to produce better estimate of  $\mathbf{c}_i$ . To distinguish this from the original sKTS algorithm, in the sequel, we refer it to as real-time sKTS (RT-sKTS) algorithm. The block diagram of the RT-sKTS algorithm is depicted in Fig. 3.

### H. Convergence Behavior

Since the proposed sKTS algorithm is derived based on EM algorithm, we can study the convergence of the EM algorithm to understand the behavior of the sKTS algorithm. In general,

<sup>3</sup>This setup corresponds to single EM iteration.

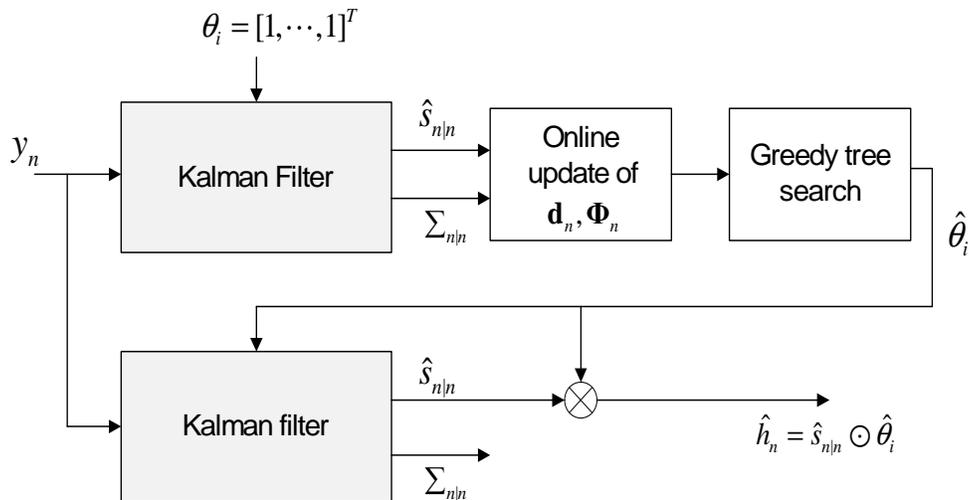


Fig. 3. Real-time implementation of the sKTS algorithm.

the convergence of the EM algorithm can be shown in two steps. First step is to show that the likelihood function  $\Pr(\mathbf{y}_{1:T}; \mathbf{c}_i^{(l)})$  is non-decreasing function of the iteration. Using the standard analysis for EM algorithm, one can show that the sKTS algorithm satisfies this property. The second step is to show that  $\mathbf{c}_i^{(l)}$  converges to a stationary point of the likelihood function  $\Pr(\mathbf{y}_{1:T}; \mathbf{c}_i)$ . Since we do not use the continuous model for  $\mathbf{c}_i^{(l)}$  (the parameter vector  $\mathbf{c}_i^{(l)}$  is binary), unfortunately, it is not easy to prove this convergence property. Nevertheless, we show from numerical experiments that the sKTS algorithm finds an accurate estimate of  $\mathbf{c}_i$  in a small number of iterations.

### III. APPLICATION TO WIRELESS CHANNEL ESTIMATION

In this section, we study the application of the proposed scheme to the training-based channel estimation problem in wireless communication systems. In many communication systems, estimation of channels is done before the symbol detection since the channel estimate is required for the detection of the transmitted symbols. Also, to perform the precoding and user scheduling in the transmitter, accurate estimate of the channel vector should be fed back from the receiver to the transmitter. Since the wireless channels whose delay spread is larger than the number of significant paths are well modeled as a sparse signal vector in a discretized delay domain, the CS techniques have been used in the sparse channel estimation problem [25]–[27]. While

existing approaches perform the sparse channel estimation using only a single observation vector or multiple observation vectors under the assumption that the CIR vector is invariant in the block, the proposed sKTS algorithm exploits the simultaneously sparse structure of time-domain CIR, which matches well with physical characteristics of multi-path fading channels. In this section, we describe the application of the proposed method to the channel estimation problem in the orthogonal frequency division multiplexing (OFDM) and the single carrier (SC) systems.

#### A. OFDM systems

We first consider the channel estimation problem of the OFDM systems. In our simulations, we focus on the scenario where the number of the pilot symbols transmitted per OFDM symbol is much smaller than the length of the CIR, thereby forming underdetermined systems in the estimation of the CIR. Note that this scenario will be prevalent when a large number of transmit antennas are deployed (e.g., in large-scale multi-input multi-output systems) since the required number of the pilot signals is proportional to the number of the transmit antennas. Since too much pilot overhead will eat out the resources and eventually limit the throughput of the systems, it is desirable to estimate the channel with small number of resources. However, when the number of the pilot signals is small, conventional channel estimators do not perform well due to the lack of observations. Whereas, by exploiting the sparsity structure of the CIR vector, the sKTS algorithm overcomes the shortage of pilot signals. In the proposed scheme, as shown in Fig. 4, we randomly allocate the pilot signals in time and frequency axis to make the better conditioned system matrix  $\mathbf{B}_n$ . As a result, while the support of  $\mathbf{h}_n$  is invariant for several OFDM symbols, the composite system matrix is varying per symbol.

Let  $P$ ,  $N$ , and  $M$  be the total number of the subcarriers, the number of the pilot subcarriers, and the length of the time-domain CIR, respectively. Further, let  $N$  be the number of pilot subcarriers per OFDM symbol. Then the relationship between the pilot signal vector  $\mathbf{p}_n \in \mathcal{C}^N$  and the observed signal vector  $\mathbf{y}_n \in \mathcal{C}^N$  of the OFDM system is expressed as

$$\mathbf{y}_n = \mathbf{p}_n \odot \mathbf{g}_n + \mathbf{v}_n \quad (35)$$

$$= \text{diag}(\mathbf{p}_n)\mathbf{g}_n + \mathbf{v}_n \quad (36)$$

where  $\mathbf{g}_n$  is the vector representing frequency-domain channel response. Using the  $P \times P$  DFT matrix  $F_P$  whose  $(i, j)$  entry is given by  $e^{-\frac{j2\pi}{P}ij}$ , the frequency-domain channel response is

expressed in terms of the time-domain CIR as

$$\mathbf{g}_n = \Pi_n F_P \Phi \mathbf{h}_n, \quad (37)$$

where  $\mathbf{h}_n$  is the  $M \times 1$  vector representing the time-domain CIR and  $\Pi_n$  is the  $N \times P$  matrix that selects the  $N$  rows of  $F_P$  depending on the location of pilot subcarriers, i.e.,

$$\Pi_n = \begin{bmatrix} \mathbf{e}_{c_{n,1}}^T \\ \vdots \\ \mathbf{e}_{c_{n,N}}^T \end{bmatrix}. \quad (38)$$

Recall that  $\{c_{n,1}, c_{n,2}, \dots, c_{n,N}\}$  are the pilot subcarrier indices at the  $n$ th OFDM symbol and  $\Phi$  is the  $P \times M$  matrix choosing the first  $M$  columns of  $F_P$ , i.e.,

$$\Phi = \begin{bmatrix} \mathbf{e}_1 & \dots & \mathbf{e}_M \end{bmatrix}. \quad (39)$$

Using (36) and (37), we obtain the observation model for which the proposed sKTS scheme can be readily applied

$$\mathbf{y}_n = \mathbf{B}_n \mathbf{h}_n + \mathbf{w}_n, \quad (40)$$

where  $\mathbf{B}_n = \text{diag}(\mathbf{p}_n) \Pi_n F_P \Phi$ .

### B. Single Carrier Systems

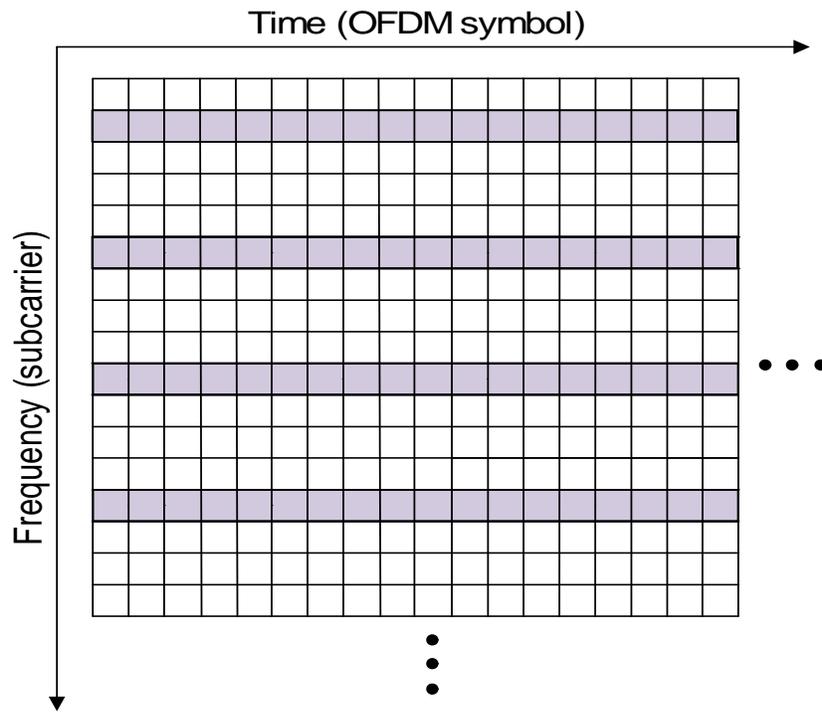
In the single carrier (SC) transmission system, the known training symbols are sent from the transmitter to the receiver before the transmission of the data symbols. Suppose that the length of the training symbols being transmitted is  $N$ , then the received signal at time  $n$  is expressed as

$$y_n = \sum_{l=1}^M h_{n,l} t_{n-l+1} + w_n, \quad 0 \leq n \leq N, \quad (41)$$

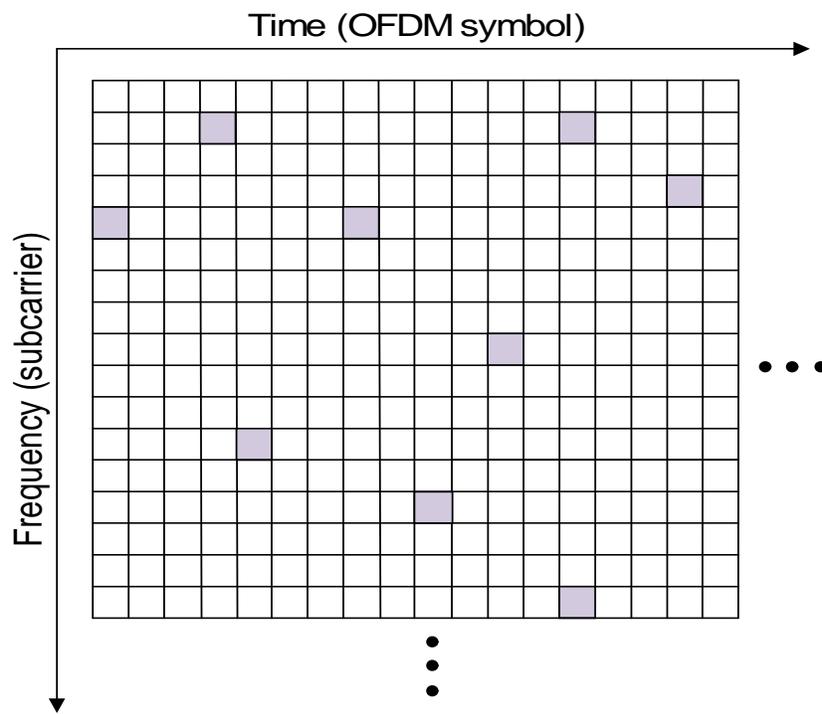
where  $h_{n,l}$  is the  $l$ th tap of the CIR at time  $n$  and  $t_i$  is the  $i$ th training symbol. Using a vector notation, we have

$$y_n = \mathbf{t}_n^H \mathbf{h}_n + w_n, \quad 0 \leq n \leq N, \quad (42)$$

where  $\mathbf{h}_n = [h_{n,1}, \dots, h_{n,M}]^T$  and  $\mathbf{t}_n = [t_n^*, \dots, t_{n-M+1}^*]^T$ . To estimate the channel vector  $\mathbf{h}_n$ , recursive least square (RLS) or Kalman channel estimators have been popularly used [28], [29].



(a)



(b)

Fig. 4. Pilot allocation in (a) conventional systems (comb-type assignment) vs. (b) proposed sKTS algorithm.

TABLE III  
SIMPLIFIED ALGORITHM FOR CHANNEL ESTIMATION IN SC SYSTEMS

Operational step	Algorithm change
Eq. (19)	$K_n = \frac{\Sigma_{n n-1} \text{diag}(\hat{\mathbf{e}}_i^{(l)}) \mathbf{t}_n}{\mathbf{t}_n^H \text{diag}(\hat{\mathbf{e}}_i^{(l)}) \Sigma_{n n-1} \text{diag}(\hat{\mathbf{e}}_i^{(l)}) \mathbf{t}_n + \sigma_w^2}$
Eq. (28)	$\mathbf{d}_i^T \triangleq \sum_{n=T_{i+1}}^{T^{(i+1)}} 2\text{Re}(\text{conj}(y_n) \mathbf{B}_n \text{diag}(\hat{\mathbf{s}}_{n 1:T}))$
Eq. (29)	$\Phi_i \triangleq \sum_{n=T_{i+1}}^{T^{(i+1)}} \text{diag}(\mathbf{t}_n^H) (\Sigma_{n 1:T} + \hat{\mathbf{s}}_{n 1:T} \hat{\mathbf{s}}_{n 1:T}^H) \text{diag}(\mathbf{t}_n)$

When the training period  $N$  is small, the system becomes ill-posed and thus the estimation quality of these channel estimation algorithms would be severely degraded. Whereas, by exploiting the simultaneously sparse structure of  $\mathbf{h}_n$  and treating  $\mathbf{t}_n$  as a system matrix, the sKTS algorithm generates reliable estimate of  $\mathbf{h}_n$ . Note that since the system matrix is a row vector  $\mathbf{t}_n^H$  in (42), each step of the algorithm can be simplified (see Table III).

#### IV. SIMULATIONS AND DISCUSSION

In this section, we study the performance of the proposed sKTS algorithm. We first conduct simulations with the synthetic data, and then we test the performance of sKTS algorithm in wireless channel estimation and MRI image reconstruction problem. In our simulations, we compare the performance of the following algorithms:

- Proposed sKTS algorithm: we set the tree search parameter  $R$  to 5. Only two iterations with  $K^{(0)} = 2K$  and  $K^{(1)} = K$  are performed.
- Conventional Kalman smoother: standard Kalman smoother [24] is used.
- Oracle-based Kalman smoother: Kalman smoothing is performed under the perfect knowledge on the support of the CIR. This algorithm provides the best achievable performance bound of the proposed sKTS algorithm.
- OMP algorithm [27]: greedy projection is used to estimate the signal support.
- KSBL algorithm [14]: the SBL algorithm [12] is extended to sequential signal estimation based on the autoregressive model.
- RW1L-DF algorithm [16]: the reweighted  $\ell_1$  optimization is modified to perform sequential dynamic filtering.

As a metric to measure the estimation performance, we use the normalized mean square error (MSE) defined as

$$\text{MSE} = 10 \log_{10} \frac{E[\|\mathbf{h}_n - \hat{\mathbf{h}}_n\|^2]}{E[\|\mathbf{h}_n\|^2]}. \quad (43)$$

#### A. Experiments with synthetic data

1) *Simulation Setup:* We first evaluate the performance of the sKTS algorithm using the synthetic data generated from the signal model in (16). Here we assume that  $\mathbf{A}_n$  and  $\mathbf{V}_n$  are given by  $\alpha \mathbf{I}_{M \times M}$  and  $\sigma_s^2(1 - \alpha^2) \mathbf{I}_{M \times M}$ . The location of nonzero elements of  $\mathbf{h}_n$  is fixed over the interval of  $T = 30$  and changes randomly for different intervals. Note that slow variation of support is not considered in the simulations. The signal dimension  $M$  is set to 200 and the sparsity order  $K$  is set to 15. Entries of the matrix  $\mathbf{B}_n$  are generated from i.i.d. Gaussian distribution  $\mathcal{N}(0, 1/M)$ .

2) *Simulation Results:* Fig. 5 presents the MSE performance of signal recovery algorithms as a function of the SNR. In Fig. 5 (a) and (b), the auto-regressive parameter  $\alpha$  is set to 0.8 and 0, respectively. Note that when  $\alpha = 0$ , each element of  $\mathbf{s}_n$  is temporally uncorrelated. For both scenarios, the dimension  $N$  of the measurement vector is set to 40. As shown in the figure, the proposed sKTS algorithm outperforms competing recovery algorithms and also performs close to the Oracle-based Kalman smoother for the whole SNR regime. Since the dimension  $N$  of the measurement vector is much smaller than the dimension  $M$  of the source signal ( $N \ll M$ ), it is no wonder that the conventional Kalman smoother does not perform well. Since the OMP algorithm uses each measurement vector independently, its performance is also not appealing.

Next, we investigate the MSE performance when the system matrix  $\mathbf{B}_n$  does not change over the processing block. Since the system matrix is fixed, if  $\mathbf{B}_n$  is ill-conditioned, recovery algorithm suffer from severe performance loss. In order to alleviate this phenomenon, we increase the dimension  $N$  of the measurement vector to 60 and also use the iteration control by setting  $K^{(0)} = 4K$ ,  $K^{(1)} = 2K$ , and  $K^{(2)} = K$  (that is, we perform three iterations with different sparsity parameters). Fig. 6 shows the MSE performance as a function of SNR. While the sKTS algorithm suffers from considerable performance loss when  $N$  is set to 45, it performs well when  $N$  becomes 65.

We next take look at the convergence behavior of the sKTS algorithm and the KSBL algorithm. In this test, we set  $N$  and  $\alpha$  to 45 and 0.8, respectively. As shown in Fig. 7, the sKTS algorithm

TABLE IV  
PARAMETERS OF THE OFDM SYSTEMS

Setup	Specification
Total number of subcarriers	1024
Bandwidth of each subcarrier	15 kHz
Symbol duration	66.7 $\mu$ s
CP length	16.7 $\mu$ s
Interval between two consecutive pilot signals	0.25 ms (three OFDM symbol)
Maximum delay spread of CIR	13 $\mu$ s
Modulation order	QPSK
Code rate	1/2

performs close to the Oracle-based Kalman smoother after two iterations, while the KSBL algorithm requires five iterations until the performance converges. Due to this reason, even though the computational complexity of the greedy tree search is a bit higher than the complexity of M-step in the EM algorithm, overall complexity of the approaches are more or less similar. In fact, our numerical experiments demonstrate that it takes 15.69 seconds for the sKTS algorithm to finish recovery process while the time required for the KSBL is 14.97 seconds.

### B. Experiments in application to channel estimation

In this subsection, we study the performance of the proposed scheme in application to channel estimation in OFDM systems.

1) *Simulation Setup*: The specific parameters for the OFDM system are summarized in the Table IV. In generating pilot symbols, we use the quadrature phase shift keying (QPSK) pseudo-random sequence. The pilot signals are transmitted for every three OFDM symbols. For the OFDM symbol containing pilots, we assign  $N$  pilot symbols. As described in the previous section, the location of pilot subcarriers is randomly chosen for each OFDM symbol. The remaining OFDM resources are filled with data symbols. The binary information bits are encoded using half rate convolutional code with generation polynomials (171, 133) and the coded bits are modulated to QPSK symbols. Each code block contains 23,360 coded bits. Considering the maximum channel delay spread specified in the Table IV, we set the dimension  $M$  of  $\mathbf{h}_n$  to 200.

TABLE V  
3GPP CHANNEL MODELS

Extended Pedestrian A model (EPA)	Delay = [0, 30, 70, 90, 110, 190, 410] ns
	Power = [0, -1, -2, -3, -8, -17.2, -20.8] dB
Extended Vehicular A model (EVA)	Delay = [0, 30, 150, 310, 370, 710, 1090, 1730, 2510] ns
	Power = [0, -1.5, -1.4, -3.6, -0.6, -9.1, -7.0, -12.0, -16.9] dB
Extended Typical Urban model (ETU)	Delay = [0, 50, 120, 200, 230, 500, 1600, 2300, 5000] ns
	Power = [-1, -1, -1, 0, 0, 0, -3, -5, -7] dB

In generating the complex Rayleigh fading frequency-selective channels, we use Jake's model [30], where temporal correlation of the CIR taps for given Doppler frequency  $f_d$  (Hz) is expressed as  $J_0(2\pi f_d T_s)$ , where  $T_s$  is the interval between consecutive pilot symbols in time ( $T_s = 0.25ms$ ) and  $J_0(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m! \Gamma(m+1)} \left(\frac{x}{2}\right)^{2m}$  is the zero-th order Bessel function. For convenience, we use *Doppler rate* defined as Doppler frequency normalized by pilot transmission rate, i.e.,  $f_d T_s$ . We use two types of channel models: 1) the exact  $K$ -sparse channel model where the location of the  $K$  nonzero taps is randomly chosen for every block of  $T$  OFDM symbols and 2) the practical channel models specified by 3GPP Long Term Evolution (LTE) standard [32] (see Table V). Note that the channel taps in the standard LTE channel model are only approximately sparse. In order to determine the parameters of the Gauss-Markov process  $\mathbf{A}_n$  and  $\mathbf{V}_n$  for a given  $f_d$ , we minimize the approximation error between the Gauss-Markov process and the Jake's model as suggested in [31]. Using the CIR estimates obtained by the sparse signal recovery algorithms, the transmitted symbols are detected by the MMSE equalizer in frequency domain. Then, the channel decoder is followed to detect the information bits. To evaluate the performance of the recovery algorithms, we measure bit error rate (BER) at the output of the channel decoder.

2) *Simulation Results*: We test the performance of the channel estimators when the exact  $K$ -sparse channels are used. The sparsity order  $K$  for these channels is set to 8 and the dimension  $N$  of the measurement vector is set to 32. Note that when  $N = 32$ , the pilot resources occupy 3.12% of the overall OFDM resources. We assume that the sparsity structure remains unchanged over the block of  $T = 30$  pilot containing OFDM symbols. We set the Doppler rate  $D_r$  to 0.05. In Fig. 8 (a) and (b), we plot the MSE and BER performance of the recovery algorithms as a function of SNR. From the figure, we clearly observe that the sKTS algorithm performs best

among all algorithms under test and also performs close to that of the Oracle-based Kalman smoother.

We next investigate the performance of the proposed sKTS algorithm when the practical LTE channel models are used. In this test, we observe the behavior of the algorithms for four distinctive scenarios: a) EVA channel with  $D_r = 0.1$  and  $N = 60$ , b) EVA channel with  $D_r = 0.05$  and  $N = 60$ , c) EPA channel with  $D_r = 0.02$  and  $N = 48$ , and d) EPA channel with  $D_r = 0.005$ ,  $N = 48$ . We set  $K = 60$  and  $K = 24$  for EVA and EPA channel models since the EVA channel exhibits longer delay spread. In Fig. 9, we observe that the sKTS algorithm maintains the performance gain over the competing algorithms for wide range of Doppler rates. Note that when compared to the results of the exact  $K$ -sparse channel model, we see that the performance gap between the sKTS and KSBL is a bit reduced.

Next, we compare the performance of the RT-sKTS described in Section II-G with the original sKTS algorithm. In this simulations, we set  $N = 32$  and  $T = 30$ . For the RT-sKTS algorithm, we set  $\alpha = 0.4$ . In order to test the performance in a harsh condition, we arbitrarily change the delay structure of the CIR for every 30 observation vectors. To ensure the convergence of the online update strategy in (33) and (34), we use the first 10 observation vectors for warming up purpose and then use the rest for measuring the MSE performance. Note that in practice, such warming up period would not be necessary since the support of channel vector would not be changed abruptly in many real applications. In Fig. 10, we see that the RT-sKTS algorithm performs close to the original sKTS algorithm in low and mid range SNR regime. In the high SNR regime, however, the RT-sKTS algorithm suffers slight performance loss due to the approximation step of  $\tilde{\mathbf{d}}_n^H$  and  $\tilde{\Phi}_n$ . Nevertheless, as shown in Fig. 10 and Fig. 8 (a), the RT-sKTS algorithm maintains the performance gain over the conventional channel estimators.

### C. Experiments in Dynamic MRI Application

In this subsection, we investigate the performance of the sKTS algorithms in the reconstruction of the dynamic MRI images. In our test, we use a sequence of  $32 \times 32$  dimensional cardiac images shown in Fig. 12<sup>4</sup>. We generate the measurements by performing two dimensional discrete wavelet transform (DWT) with a 2-level Daubechies-4 wavelet, applying two dimensional DFT

<sup>4</sup>These images are decimated from the original  $128 \times 128$  images [34]. The raw image data is available online [33].

matrix and taking the  $N$  randomly chosen frequency-domain image samples. After adding the Gaussian noise to the image, we recover the original image using the recovery algorithms. We set  $N = 358$ , which corresponds to about 35% of the image size (i.e.,  $M = 1024$ ). We could empirically observe that the location of nonzero coefficients in wavelet image is slowly changing (i.e., support change occurs for only a few places), which matches well with our simultaneous sparse signal model. In order to capture the most of signal energy, we set  $K = 152$  for all images<sup>5</sup>. In Fig. 12, we plot the MSE of the several image recovery schemes obtained for each image. The sKTS algorithm outperforms the basis pursuit denoising (BPDN) [1] and RW1L-DF [16] and also performs close to the Oracle-based Kalman smoother. Note that we could not include modified CS scheme in [34] in our numerical experiments since large number of measurement samples is required for the first image.

## V. CONCLUSIONS

In this paper, we studied the problem to estimate the time-varying sparse signals when the sequence of the correlated observation vectors are available. In many signal processing and wireless communication applications, the support of sparse signals changes slowly in time and thus can be well modeled as simultaneously sparse signal, we proposed a new sparse signal recovery algorithm, referred to as sparse Kalman tree search (sKTS), that identifies the support of the sparse signal using multiple measurement vectors. The proposed sKTS scheme performs the Kalman smoothing to extract the *a posteriori* statistics of the source signals and the greedy tree search to identify the support of the signal. From the case study of sparse channel estimation problem in orthogonal frequency division multiplexing (OFDM) and image reconstruction in dynamic MRI, we demonstrated that the proposed sKTS algorithm is effective in recovering the dynamic sparse signal vectors.

<sup>5</sup>Actually, we set  $K$  to the the number of coefficient containing 99.9% of the signal energy.

APPENDIX A  
DERIVATION OF (15)

From (9) and (14), we get

$$Q(\mathbf{c}_i; \hat{\mathbf{c}}_i^{(l)}) = C' - \sum_{n=T_i+1}^{T(i+1)} E \left[ \frac{1}{\sigma_w^2} \|\mathbf{y}_n - \mathbf{B}_n \text{diag}(\mathbf{c}_i) \mathbf{s}_n\|^2 \middle| \mathbf{y}_{1:T}; \hat{\mathbf{c}}_i^{(l)} \right] \quad (44)$$

$$\begin{aligned} &= C'' + \frac{1}{\sigma_w^2} \sum_{n=T_i+1}^{T(i+1)} \left\{ E \left[ \text{tr} [2\text{Re}(\mathbf{B}_n \text{diag}(\mathbf{c}_i) \mathbf{s}_n \mathbf{y}_n^H)] \middle| \mathbf{y}_{1:T}; \hat{\mathbf{c}}_i^{(l)} \right] \right. \\ &\quad \left. - E \left[ \text{tr} [\mathbf{B}_n \text{diag}(\mathbf{c}_i) \mathbf{s}_n \mathbf{s}_n^H \text{diag}(\mathbf{c}_i) \mathbf{B}_n^H] \middle| \mathbf{y}_{1:T}; \hat{\mathbf{c}}_i^{(l)} \right] \right\} \quad (45) \end{aligned}$$

$$\begin{aligned} &= C'' + \frac{1}{\sigma_w^2} \sum_{n=T_i+1}^{T(i+1)} \left\{ \text{tr} \left[ 2\text{Re} \left( \mathbf{B}_n \text{diag}(\mathbf{c}_i) E \left[ \mathbf{s}_n \middle| \mathbf{y}_{1:T}; \hat{\mathbf{c}}_i^{(l)} \right] \mathbf{y}_n^H \right) \right] \right. \\ &\quad \left. - \text{tr} \left[ \mathbf{B}_n \text{diag}(\mathbf{c}_i) E \left[ \mathbf{s}_n \mathbf{s}_n^H \middle| \mathbf{y}_{1:T}; \hat{\mathbf{c}}_i^{(l)} \right] \text{diag}(\mathbf{c}_i) \mathbf{B}_n^H \right] \right\}, \quad (46) \end{aligned}$$

$$(47)$$

where  $C'$  and  $C''$  are the terms independent of  $\mathbf{c}_i$ . Using the property of the trace, i.e.,  $\text{tr}(ABC) = \text{tr}(BCA) = \text{tr}(CAB)$ , we have

$$\begin{aligned} Q(\mathbf{c}_i; \hat{\mathbf{c}}_i^{(l)}) &= C'' + \frac{1}{\sigma_w^2} \sum_{n=T_i+1}^{T(i+1)} \left\{ 2\text{Re} \left( \mathbf{y}_n^H \mathbf{B}_n \text{diag}(\mathbf{c}_i) E \left[ \mathbf{s}_n \middle| \mathbf{y}_{1:T}; \hat{\mathbf{c}}_i^{(l)} \right] \right) \right. \\ &\quad \left. - \text{tr} \left[ \mathbf{B}_n \text{diag}(\mathbf{c}_i) E \left[ \mathbf{s}_n \mathbf{s}_n^H \middle| \mathbf{y}_{1:T}; \hat{\mathbf{c}}_i^{(l)} \right] \text{diag}(\mathbf{c}_i) \mathbf{B}_n^H \right] \right\} \quad (48) \end{aligned}$$

APPENDIX B  
DERIVATION OF (26)

Denoting  $\mathbf{b}_{n,i}$  as the transpose of the  $i$ th row vector of  $\mathbf{B}_n$ , we can express the lefthand term of (26) as

$$\text{left term} = \sum_{j=1}^M \mathbf{b}_{n,j}^T \text{diag}(\mathbf{c}_i) (\Sigma_{n|1:T} + \hat{\mathbf{s}}_{n|1:T} \hat{\mathbf{s}}_{n|1:T}^H) \text{diag}(\mathbf{c}_i) \text{conj}(\mathbf{b}_{n,j}). \quad (49)$$

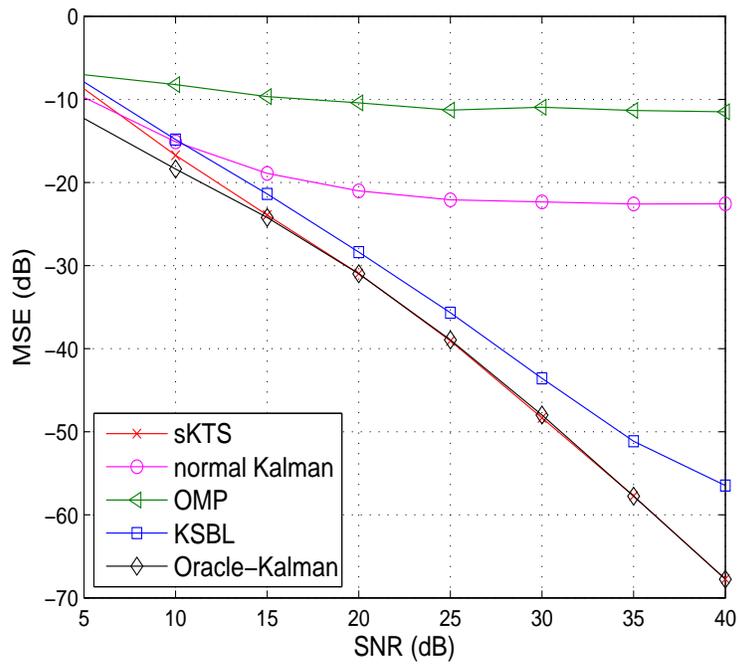
Since  $\mathbf{b}_{n,j}^T \text{diag}(\mathbf{c}_i) = \mathbf{c}_i^T \text{diag}(\mathbf{b}_{n,j})$  and  $\text{diag}(\mathbf{c}_i) \text{conj}(\mathbf{b}_{n,j}) = \text{diag}(\text{conj}(\mathbf{b}_{n,j})) \mathbf{c}_i$ , we further have

$$\text{left term} = \sum_{j=1}^M \mathbf{c}_i^T \text{diag}(\mathbf{b}_{n,j}^T) (\Sigma_{n|1:T} + \hat{\mathbf{s}}_{n|1:T} \hat{\mathbf{s}}_{n|1:T}^H) \text{diag}(\text{conj}(\mathbf{b}_{n,j})) \mathbf{c}_i \quad (50)$$

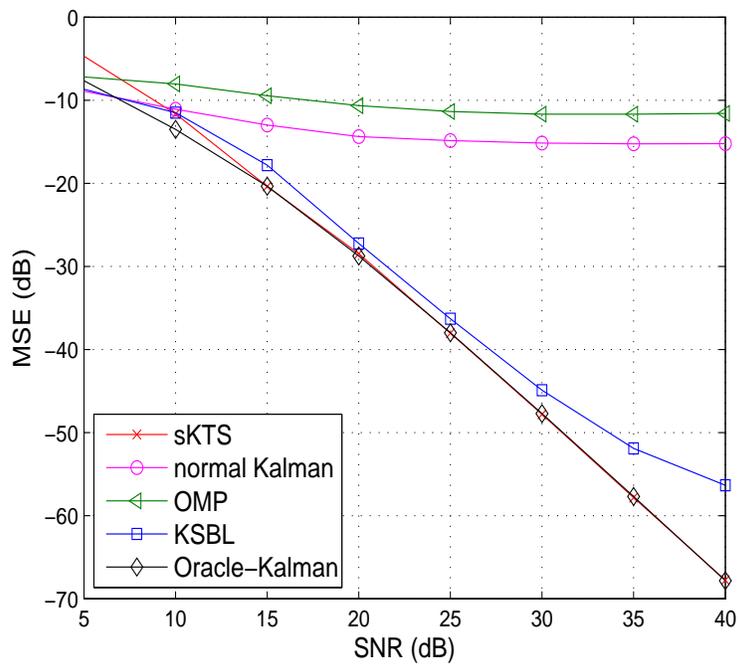
$$= \mathbf{c}_i^T \sum_{j=1}^M [\text{diag}(\mathbf{b}_{n,j}^T) (\Sigma_{n|1:T} + \hat{\mathbf{s}}_{n|1:T} \hat{\mathbf{s}}_{n|1:T}^H) \text{diag}(\text{conj}(\mathbf{b}_{n,j}))] \mathbf{c}_i, \quad (51)$$

and hence we finally have

$$\text{left term} = \mathbf{c}_i^T (\text{conj}(\mathbf{B}_n^H \mathbf{B}_n) \odot (\Sigma_{n|1:T} + \widehat{\mathbf{s}}_{n|1:T} \widehat{\mathbf{s}}_{n|1:T}^H)) \mathbf{c}_i. \quad (52)$$



(a)



(b)

Fig. 5. The MSE performance as a function of SNR with (a)  $\alpha = 0.8$  and (b)  $\alpha = 0$ .

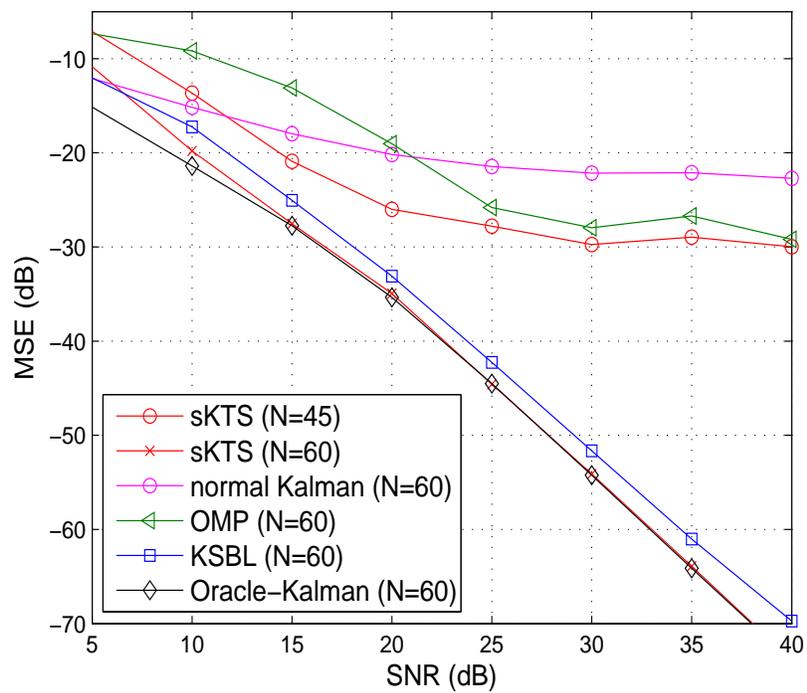


Fig. 6. The MSE performance when  $\mathbf{B}_n$  is fixed.

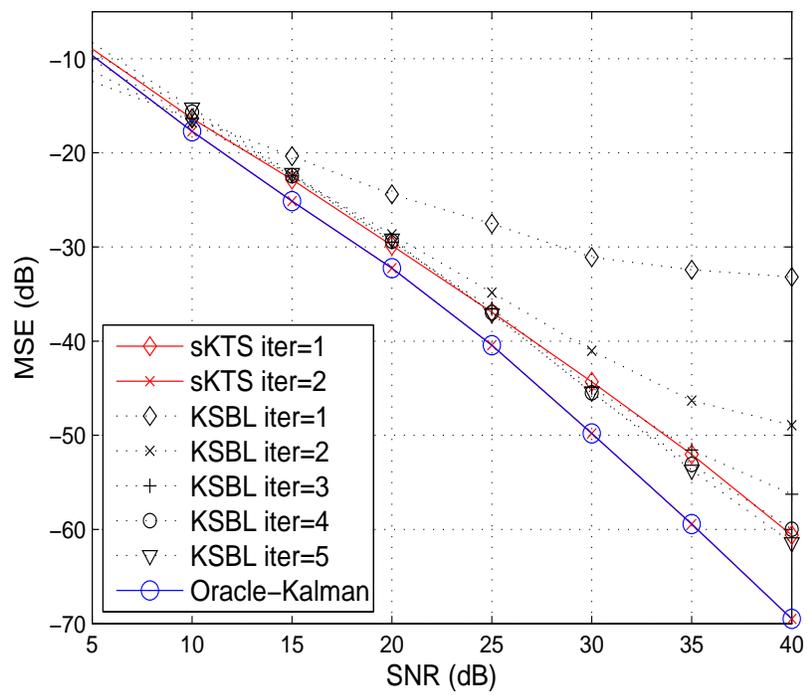
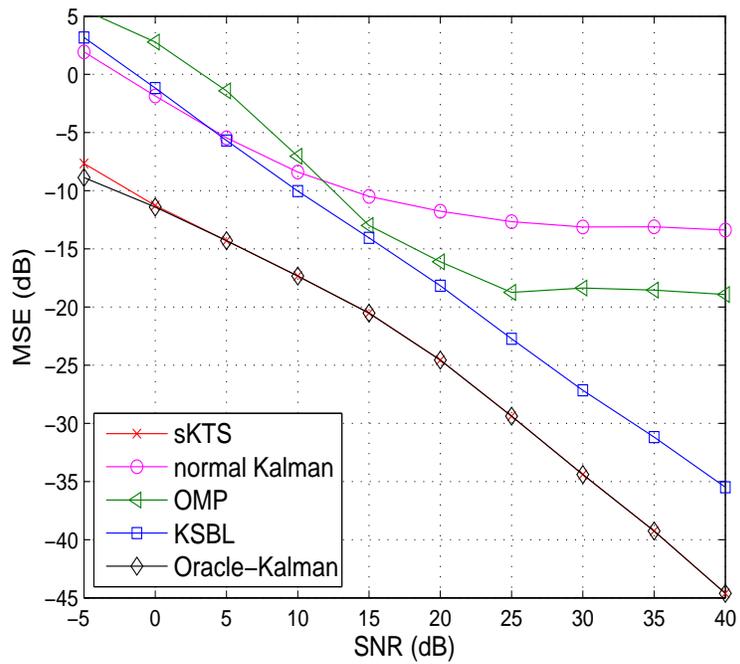
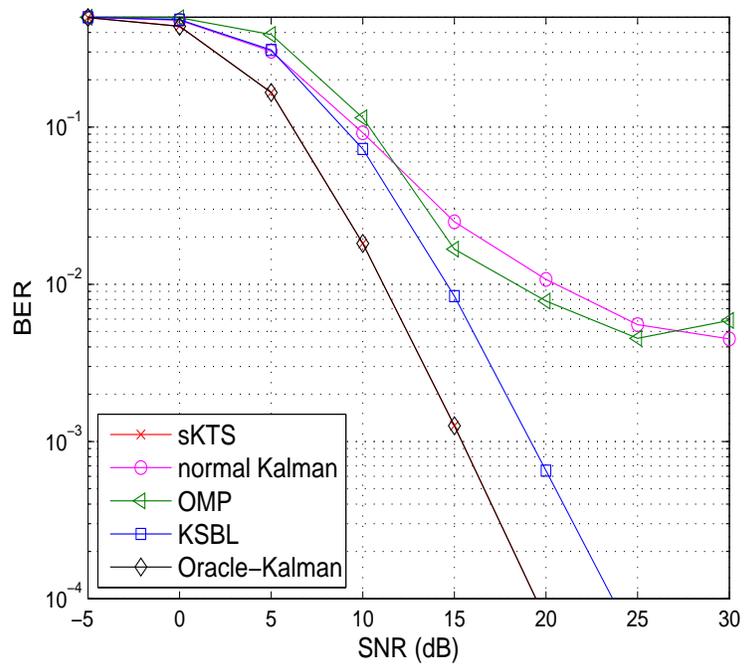


Fig. 7. The MSE performance of the sKTS and KSBL as a function of the number of iterations.

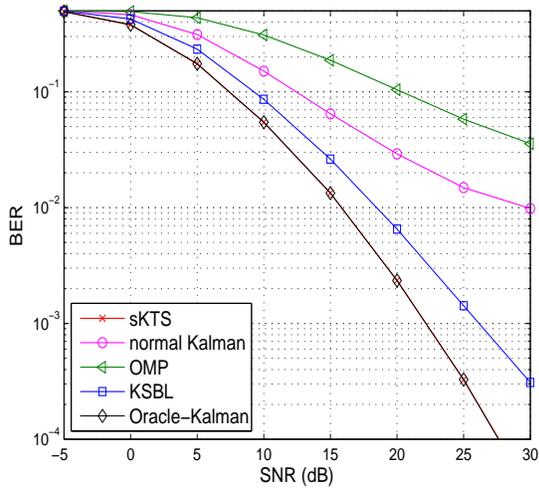


(a)

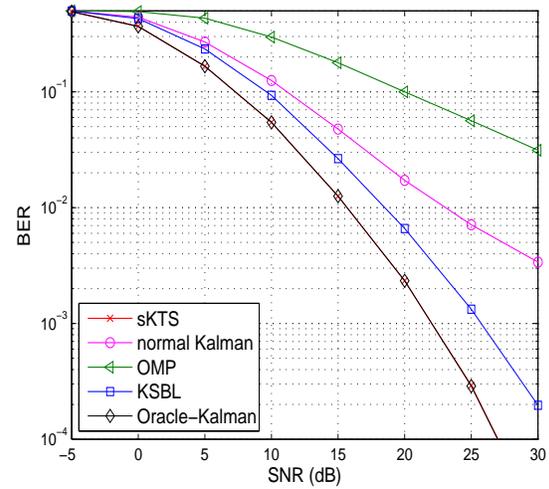


(b)

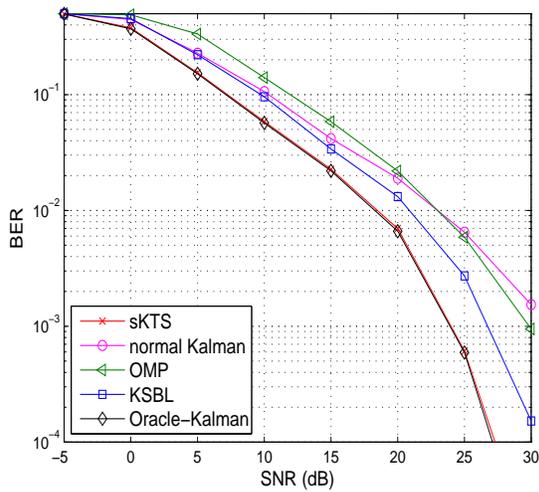
Fig. 8. The performance of the recovery algorithms for the  $K$ -sparse channel model: a) MSE and b) BER performance.



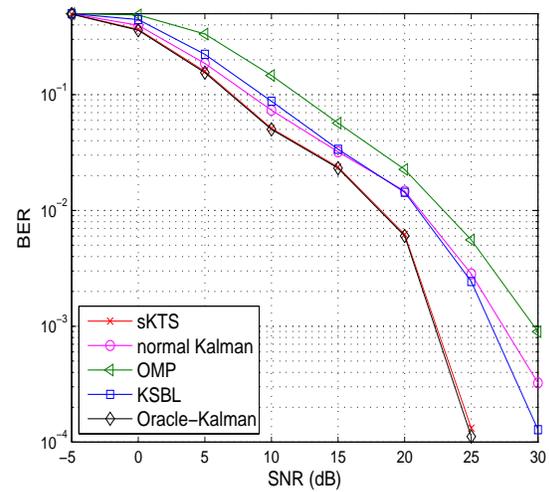
(a)



(b)



(c)



(d)

Fig. 9. The BER performance as a function of SNR for the four distinctive scenarios of a) EVA channel with  $D_r = 0.1$  and  $N = 60$ , b) EVA channel with  $D_r = 0.05$  and  $N = 60$ , c) EPA channel with  $D_r = 0.02$  and  $N = 48$ , and d) EPA channel with  $D_r = 0.005$  and  $N = 48$ .

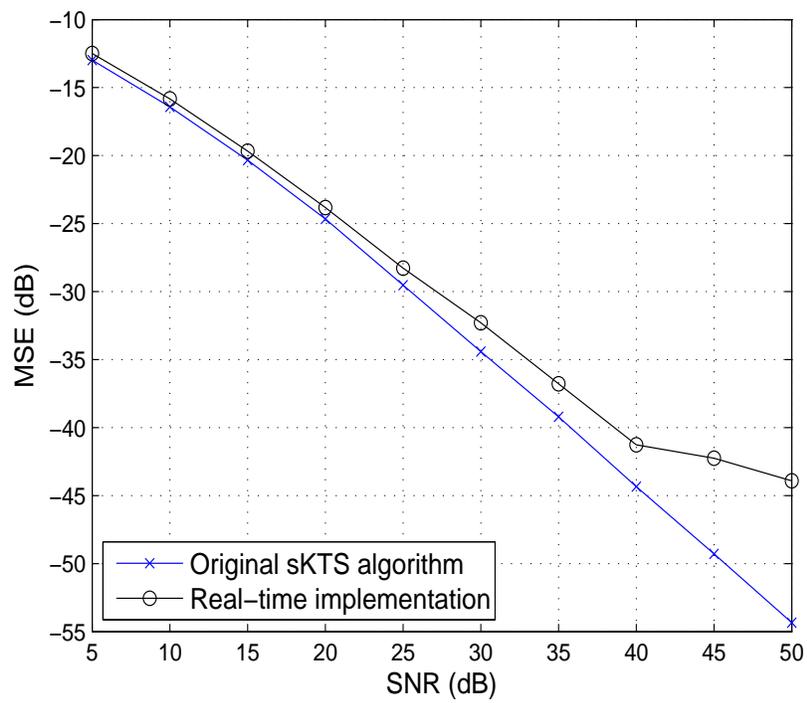


Fig. 10. The MSE performance of the RT-sKTS algorithm.



Fig. 11. A sequence of cardiac MRI images.

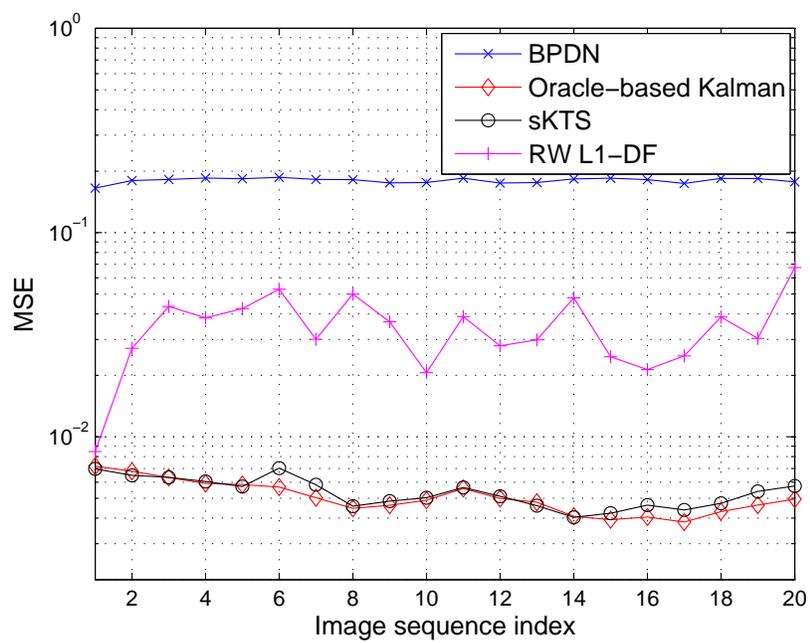


Fig. 12. The MSE performance of the dynamic MRI image as a function of image sequence index.

## REFERENCES

- [1] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Information Theory*, vol. 52, pp. 489-509, Feb. 2006.
- [2] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Information Theory*, vol. 53, pp. 4655-4666, Dec. 2007.
- [3] J. Wang, S. Kwon, and B. Shim, "Generalized orthogonal matching pursuit," *IEEE Trans. Signal Process.*, vol. 60, pp. 6202-6216, Dec. 2012.
- [4] D. Needell and J. A. Tropp, "CoSaMP: iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, vol. 26, pp. 301-321, 2009.
- [5] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Information Theory*, vol. 55, pp. 2230-2249, May 2009.
- [6] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, pp. 2153-2164, August 2004.
- [7] S. Ji, Y. Xue, and L. Carin, "Bayesian compressed sensing," *IEEE Trans. Signal Process.*, vol. 56, pp. 2346-2356, June 2008.
- [8] J. A. Tropp, A. C. Gilbert, M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: greedy pursuit," *Signal Processing*, vol. 86, pp. 572-588, 2006.
- [9] S. F. Cotter, B. D. Rao, K. Engan, K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 53, pp. 2477-2488, July 2005.
- [10] J. A. Tropp, A. C. Gilbert, M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part II: convex relaxation," *Signal Processing*, vol. 86, pp. 589-602, 2006.
- [11] D. P. Wipf and B. D. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Trans. Signal Process.*, vol. 55, pp. 3704-3716, July 2007.
- [12] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, pp. 912-926, Sept. 2011.
- [13] Z. Zhang and B. D. Rao, "Sparse signal recovery in the presence of correlated multiple measurement vectors," *Proc. ICASSP 2010*, Dallas, TX USA, 2010, pp. 3986-3989.
- [14] R. Prasad, C. R. Murphy and B. D. 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, July 2014.
- [15] A. S. Charles, M. Asif, J. Romberg, and C. Rozwell, "Sparsity penalties in dynamical system estimation," *Proc. Conf. on Information Sciences (CISS)*, Mar. 2011, pp. 1-6.
- [16] A. S. Charles and C. J. Rozell, "Re-weighted  $\ell_1$  dynamic filtering for time-varying sparse signal estimation," <http://arxiv.org/abs/1208.0325>.
- [17] N. Vaswani, "Kalman filtered compressed sensing," *Proc. ICIP* San Diego, CA USA, 2008, pp. 893-896.
- [18] J. Ziniel and P. Schniter, "Dynamic compressive sensing of time-varying signals via approximate message passing," *IEEE Trans. Signal Processing*, vol 61, pp. 5270-5284, Nov. 2013.
- [19] N. Kalouptsidis, G. Mileounis, B. Babadi, and V. Tarokh, "Adaptive algorithms for sparse system identification," *Signal Processing*, vol. 91, pp. 1910-1919, Aug. 2011.
- [20] D. Angelosante, S. Roumeliotis, and G. Giannakis, "Lasso-Kalman smoother for tracking sparse signals," *Asilomar conference*, Pacific Grove, CA USA, Nov. 2009, pp. 181-185.

- [21] J. W. Choi, T. J. Riedl, K. Kim, A. C. Singer, and J. C. Preisig, "Adaptive linear turbo equalization over doubly selective channels," *IEEE Journal of Oceanic Engineering*, vol. 36, pp. 473-489, Oct. 2011.
- [22] A. P. Demster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc. B*, vol. 39, pp. 1-38, 1977.
- [23] J. W. Choi, K. Kim, T. J. Riedl, and A. C. Singer, "Iterative estimation of sparse and doubly-selective multi-input multi-output (MIMO) channel," *Proc. Signals, Systems and Computers Asilomar Conference*, Nov. 2009, pp. 620-624.
- [24] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation*, Prentice Hall, 2000.
- [25] W. U. Bajwa, J. Haupt, A. M. Sayeed and R. Nowak, "Compressed channel sensing: a new approach to estimating sparse multipath channels," *Proceedings of the IEEE*, vol. 98, pp. 1058-1076, June 2010.
- [26] C. R. Berger, S. Zhou, J. C. Preisig and P. Willett, "Sparse channel estimation for multicarrier underwater acoustic communication: from subspace methods to compressed sensing," *IEEE Trans. Signal Process.*, vol. 58, pp. 1708-1721, March 2010.
- [27] S. F. Cotter and B. D. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Trans. Commun.*, vol. 50, pp. 374-377, March 2002.
- [28] T. K. Akino, "Optimum-weighted RLS channel estimation for rapid fading MIMO channels," *IEEE Trans. Wireless Comm.*, vol. 7, pp.4248-4260, Nov. 2008.
- [29] M. B. Loiola, R. R. Lopes, and J. M. T. Romano, "Modified Kalman filters for channel estimation in orthogonal space-time coded systems," *IEEE Trans. Signal Process.*, vol. 60, pp. 533-538, Jan. 2012
- [30] W. C. Jakes, "Microwave Mobile Communications," New York, John Wiley & Sons Inc, 1975.
- [31] K. E. Baddour and N. C. Beaulieu, "Autoregressive modeling for fading channel simulation," *IEEE Trans. Wireless Commun.*, vol. 4, pp. 1650-1662, July 2005.
- [32] 3GPP TS 36.101. "User Equipment (UE) Radio Transmission and Reception." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network (E-UTRA)*. URL: <http://www.3gpp.org>.
- [33] <http://www.ece.iastate.edu/~namrata/software.html>.
- [34] N. Vaswani and W. Lu, "Modified-CS: modifying compressive sensing for problems with partially known support," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4595-4607, Sept. 2010.