Statistical Estimation of Block-Sparse Time-Varying Signals from Multiple Measurement Vectors

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

Abstract

In this paper, we introduce a new sparse signal recovery algorithm, referred to as sparse Kalman tree search (sKTS), that provides a robust reconstruction of the sparse vector from the sequence of correlated observation vectors. The proposed sKTS algorithm builds on expectation-maximization algorithm and consists of two main operations: 1) Kalman smoothing for obtaining the \textit{a posteriori} statistics of the source signal vectors and 2) identification of the support of the signal vectors via greedy tree search algorithm. The performance of the sKTS algorithm is evaluated in the context of sparse channel estimation for wireless communications. Through computer simulations, we demonstrate that sKTS outperforms conventional sparse recovery algorithms and also performs close to the Oracle (genie-based) Kalman estimator.

Index Terms

Compressed sensing, block sparse signal, channel estimation, OFDM, 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.
1

Statistical Estimation of Block-Sparse Time-Varying Signals from Multiple Measurement Vectors

I. INTRODUCTION

Over the years, there have been increased demand of problem to recover 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.

So far, various signal recovery algorithms for CS have been proposed. Roughly speaking, these approaches are categorized into two classes; One 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], CoSaMP [3], and subspace pursuit [4]. The other approach is based on the probabilistic signal model, where the signal sparsity is described by the \textit{a priori} distribution of the signal and Bayesian framework is used to find the sparse solution [5], [6].

When multiple measurement vectors 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. This approach is often referred to as multiple measurement vectors (MMV) [7]–[12]. The MMV algorithms targeted for the deterministic signal recovery include the mixed-norm solution [7], [8] and convex relaxation [9] while the probabilistic MMP techniques include the M-SBL method [10], block-SBL [11], and auto-regressive SBL [12].

In this work, we are primarily concerned with the MMV problem when the observation vectors
are sequentially acquired, i.e.,

\[ y_n = B_n h_n + w_n, \]

where \( y_n \) and \( w_n \) are the \( N \times 1 \) observation and noise vector (\( \sim \mathcal{CN}(0, \sigma_n^2 I) \)), respectively, \( B_n \) is the \( N \times M \) system matrix, and \( h_n \) is the \( M \times 1 \) source signal vector. Our goal in this setup is to estimate the source signal \( h_n \) using the sequence of the observations \( \{y_n\} \) when 1) the source signal \( h_n \) is sparse (i.e., the number nonzero elements in \( h_n \) is small) and 2) the dimension of the observation vector \( y_n \) is smaller than that of the source vector \( (N \ll M) \). While many MMV techniques consider the scenario where the amplitude of \( h_n \) is changing independently and the support of \( h_n \) and the system matrix \( B_n \) are fixed over the period under consideration, we focus on the scenario where the amplitude of \( h_n \) is changing over time with certain temporal correlation structure. Additionally, our work addresses the scenario where \( B_n \) changes with time.

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 fast 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 block-sparse signal model since the support of the sparse signal is constant over the fixed interval under this assumption. Note that this model matches well with the characteristics of multi-path fading channels for wireless communications where the channel impulse response \( h_n \) should be estimated from the received signal \( y_n \). 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 [13]. We observe that when compared to the amplitude of the channel taps, the sparse 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 indicating the existence of the signal. In recovering the original signal vector \( h_n \) from the measurement vectors, we use the modified expectation-maximization (EM) algorithm [14]. The proposed scheme, dubbed as sparse-Kalman-Tree-Search (sKTS), consists of two main operations: 1) Kalman smoothing to gather the \textit{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 sparse 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 a tree.

We note that our work is distinct from recently proposed MMV approaches in two respects: First, in contrast to the previous statistical MMV approaches in [10], [11] using real-valued parameters to describe signal sparsity, the proposed method employs the deterministic discrete (binary) parameter vector indicating the existence of signal, which allows us to use the combinatorial optimization scheme to identify the support of the sparse signals. Moreover, while the approaches in [10], [11] are designed for the scenario where the system matrix $B_n$ is fixed over the multiple measurement vectors, the proposed method does not impose this constraint yet recovers the sparse vector accurately. Second, while the recent work in [15] 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 binary parameter vector using the modified EM algorithm.

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 $x$. $\| \cdot \|_p^2$ indicates an $\ell_p$-norm of a vector. Without a subscript $p$, it implicitly means the $\ell_2$-norm. $\text{diag}(\cdot)$ is a diagonal matrix that has elements on the main diagonal. $\text{Re}(x)$ and $\text{Im}(x)$ denote the real and imaginary parts of $x$, respectively. $\mathcal{CN}(m, \sigma^2)$ denotes the circular symmetric complex Gaussian density with mean $m$ and variance $\sigma^2$. $E[X]$ 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 $\text{Cov}(x, y) = E[xy^H] - E[x]E[y]^H$ and $\text{Cov}(x) = \text{Cov}(x, x)$. $Pr(A)$ 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 $B$. $e_i$ is the elementary vector whose $i$th
coordinate is one while the rest are all zero.

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 block-sparse signal model and then present the proposed sparse signal estimation scheme.

A. Block-Sparse Signal Model

We express a time-varying sparse signal $h_n$ as a product between a vector of random processes $s_n$ describing the amplitudes of nonzero entries in $h_n$ and the vector $\theta_i = [\theta_{i,0}, \cdots, \theta_{i,M-1}]^T$.
indicating the existence of signal. That is,

\[ h_n = \text{diag}(\theta_i)s_n, \]  

(2)

where \( i \) is the block index, the entry of \( \theta_i \) is either 0 or 1 depending on the existence of the signal

\[ \theta_{i,j} = \begin{cases} 1 & \text{if the } j\text{th entry of } h_n \text{ exists} \\ 0 & \text{otherwise}, \end{cases} \]  

(3)

and the time-varying amplitude \( s_n \in \mathbb{C}^M \) is modeled as Gauss-Markov random process

\[ s_{n+1} = A_i s_n + v_n, \]  

(4)

where \( v_n \in \mathbb{C}^M \) is the process noise vector (\( \sim \mathcal{CN}(0, V_i) \)) and \( A_i \in \mathbb{C}^{M \times N} \) is the state update matrix. Note that the block index \( i \) is associated with the interval of the length \( T, n \in [T i \leq n < T(i + 1)] \). As mentioned, we assume that the support of the underlying sparse signals is locally time-invariant so that \( \theta_i \) is constant in a block of consecutive measurement vectors. Using this together with the observation model in (1), we obtain the block-sparse signal model

\[ s_{n+1} = A_i s_n + v_n, \]

\[ h_n = \text{diag}(\theta_i)s_n \]

\[ y_n = B_n h_n + w_n. \]  

(5)

Since \( h_n \) follows Gaussian distribution when \( \theta_i \) is given, the \textit{a priori} distribution of the source signal \( h_n \) can be described by

\[ P_r(h_n) = \frac{1}{(2\pi)^m \det(\text{Cov}(h_n))} \exp \left( -(h_n - E[h_n])^H \text{Cov}(h_n)^{-1} (h_n - E[h_n]) \right), \]  

(6)

where

\[ E[h_n] = \text{diag}(\theta_i) \otimes E[s_n] \]

\[ \text{Cov}(h_n) = \text{diag}(\theta_i) \text{Cov}(s_n) \text{diag}(\theta_i). \]  

(7)
B. Derivation of Statistical Sparse Signal Estimation

When the multiple measurement vectors \( \{ y_{Ti}, \cdots, y_{T(i+1)-1} \} \) in the \( i \)th block are given, the maximum likelihood (ML) estimate of \( \theta_i \) is expressed as

\[
\theta_i^{\text{ML}} = \arg \max_{\theta_i \in \{0, 1\}^T} \sum_{j=0}^{M-1} \theta_{i,j} = K \ln \Pr(y_{1:T}; \theta_i), \tag{8}
\]

where \( y_{1:T} = [y_{Ti}, \cdots, y_{T(i+1)-1}]^T \) and \( K \) is the sparsity order (the number of nonzero entries) of \( h_n \). Note that the subscript 1 : \( T \) denotes the set of time indices for the \( i \)th block. Note also that the ML estimate \( \theta_i^{\text{ML}} \) is chosen among all candidates satisfying the sparsity constraint \( \sum_{j=0}^{M-1} \theta_{i,j} = K \). Since the ML problem in (8) involves the marginalization over all possible combinations of the latent variables \( s_{1:T} \), it is in general computationally unattractive to find out the solution using the direct approach. Better approach for this scenario would be 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 \( \hat{\theta}_i^{(l)} \), \( l = 0, 1, 2, \ldots \) by alternating two major steps (E-step and M-step), which are given, respectively

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

\[
Q \left( \theta_i; \hat{\theta}_i^{(l)} \right) = E \left[ \ln \Pr(y_{1:T}, s_{1:T}; \theta_i) \right| y_{1:T}; \hat{\theta}_i^{(l)}], \tag{9}
\]

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

\[
\hat{\theta}_i^{(l+1)} = \arg \max_{\theta_i \in \{0, 1\}^M} \max_{\sum_{j=0}^{M-1} \theta_{i,j} = K} Q \left( \theta_i; \hat{\theta}_i^{(l)} \right), \tag{10}
\]

where \( \hat{\theta}_i^{(l)} \) is the estimate of \( \theta_i \) 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 with a proper initialization of \( \hat{\theta}_i^{(0)} \), \( \theta_i \) can be accurately estimated with high probability (see Section IV).

1) **The E-step**: The goal of the E-step is to obtain a simple expression of the cost metric \( Q(\theta_i, \hat{\theta}_i^{(l)}) \) using the block-sparse signal model. First, \( \ln \Pr(y_{1:T}, s_{1:T}; \theta_i) \) is expressed as

\[
\ln \Pr(y_{1:T}, s_{1:T}; \theta_i) = \ln \Pr(y_{1:T}|s_{1:T}; \theta_i) + \ln \Pr(s_{1:T}; \theta_i) \tag{11}
\]

\[
= \sum_{n=T_i}^{T(i+1)-1} \ln \Pr(y_n|s_n; \theta_i) + \sum_{n=T_i}^{T(i+1)-1} \ln \Pr(s_n|s_{n-1}) \tag{12}
\]
Noting that $\Pr(y_n|s_n; \theta_i) \sim CN(B_n \text{diag}(\theta_i)s_n, \sigma^2_i I)$ and $\Pr(s_n|s_{n-1}) \sim CN(A_is_{n-1}, V_i)$, we have

$$\ln \Pr(y_{1:T}, s_{1:T}; \theta_i) = - \sum_{n=T_i}^{T(i+1)-1} \frac{1}{\sigma_w^2} ||y_n - B_n \text{diag}(\theta_i)s_n||^2$$

$$- \sum_{n=T_i}^{T(i+1)-1} (s_n - A_is_{n-1})^H V_i^{-1} (s_n - A_is_{n-1}) + C$$

$$= - \sum_{n=T_i}^{T(i+1)-1} \frac{1}{\sigma_w^2} ||y_n - B_n \text{diag}(\theta_i)s_n||^2 + C' \quad (13)$$

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

$$Q(\theta_i; \hat{\theta}^{(l)}_i) = C'' + \frac{1}{\sigma_w^2} \sum_{n=T_i}^{T(i+1)-1} \left\{ 2\text{Re} \left( y_n^H B_n \text{diag}(\theta_i)E \left[ s_n | y_{1:T}; \hat{\theta}^{(l)}_i \right] \right) \right\}$$

$$- \text{tr} \left[ B_n \text{diag}(\theta_i)E \left[ s_n s_n^H | y_{1:T}; \hat{\theta}^{(l)}_i \right] \text{diag}(\theta_i)B_n^H \right]. \quad (15)$$

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

$$\hat{s}_{n|1:T} = E \left[ s_n | y_{1:T}; \hat{\theta}^{(l)}_i \right]$$

$$\Sigma_{n|1:T} = \text{Cov} \left[ s_n s_n^H | y_{1:T}; \hat{\theta}^{(l)}_i \right].$$

Note that these \textit{a posteriori} statistics $\hat{s}_{n|1:T}$ and $\Sigma_{n|1:T}$ can be obtained by Kalman smoothing [16]. When $\hat{\theta}^{(l)}_i$ is given, from (5), the system equations for Kalman smoothing becomes

$$s_{n+1} = A_is_n + v_n$$

$$y_n = B_n \text{diag}(\theta_i)s_n + w_n. \quad (16)$$

We employ the fixed-interval Kalman smoothing algorithm performing sequential estimation of $\hat{s}_{n|1:T}$ and $\Sigma_{n|1:T}$ via forward and backward recursions in a block of observations $y_{1:T}$. Let $\hat{s}_{n|j}$ and $\Sigma_{n|j}$ be the conditional mean and covariance given the first $j$ observation vectors, i.e.,

$$\hat{s}_{n|j} = E \left[ s_n | y_{T_1}, \ldots, y_{T_{i+j}}; \hat{\theta}^{(l)}_i \right]$$

and $\Sigma_{n|j} = \text{Cov} \left[ s_n s_n^H | y_{B_1}, \ldots, y_{B_{i+j}}; \hat{\theta}^{(l)}_i \right]$, then the fixed-interval Kalman smoothing algorithm is summarized as
• Forward recursion rule:
  \[
  \hat{s}_{n|n-1} = A_i \hat{s}_{n-1|n-1}
  \]
  \[
  \Sigma_{n|n-1} = A_i \Sigma_{n-1|n-1} A_i^H + V
  \]
  \[
  K_n = \Sigma_{n|n-1} \text{diag}(\hat{\theta}_i^{(l)}) B_n^H \left( B_n \text{diag}(\hat{\theta}_i^{(l)}) \Sigma_{n|n-1} \text{diag}(\hat{\theta}_i^{(l)}) B_n^H + \sigma_w^2 I \right)^{-1}
  \]
  \[
  \hat{s}_{n|n} = \hat{s}_{n|n-1} + K_n (y_n - B_n \text{diag}(\hat{\theta}_i^{(l)}) \hat{s}_{n|n-1})
  \]
  \[
  \Sigma_{n|n} = \left( I - K_n B_n \text{diag}(\hat{\theta}_i^{(l)}) \right) \Sigma_{n|n-1},
  \]

• Backward recursion rule:
  \[
  S_n = \Sigma_{n|n} A_i \Sigma_{n+1|n}^{-1}
  \]
  \[
  \hat{s}_{n|1:T} = \hat{h}_{n|n} + S_n \left( \hat{h}_{n+1|1:T} - A_i \hat{h}_{n|n} \right)
  \]
  \[
  \Sigma_{n|1:T} = \Sigma_{n|n} + S_n \left( \Sigma_{n+1|1:T} - \Sigma_{n+1|n} \right)^H S_n^H.
  \]

Using \( \hat{s}_{n|1:T} \) and \( \Sigma_{n|1:T} \), \( Q \left( \theta_i; \hat{\theta}_i^{(l)} \right) \) can be rewritten as

\[
Q \left( \theta_i; \hat{\theta}_i^{(l)} \right) = C'' + \frac{1}{\sigma_w^2} \sum_{n=T_i}^{T(i+1)-1} \left\{ 2\text{Re} \left( y_n^H B_n \text{diag}(\theta_i) \hat{s}_{n|1:T} \right) - \text{tr} \left[ B_n \text{diag}(\theta_i) \left( \Sigma_{n|1:T} + \hat{s}_{n|1:T} \hat{s}_{n|1:T}^H \right) \text{diag}(\theta_i) B_n^H \right] \right\}.
\]

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

\[
\text{tr} \left[ B_n \text{diag}(\theta_i) \left( \Sigma_{n|1:T} + \hat{s}_{n|1:T} \hat{s}_{n|1:T}^H \right) \text{diag}(\theta_i) B_n^H \right] = \theta_i^T \left( \text{conj} \left( B_n^H B_n \right) \odot \left( \Sigma_{n|1:T} + \hat{s}_{n|1:T} \hat{s}_{n|1:T}^H \right) \right) \theta_i.
\]

From (25) and (26), we have

\[
Q \left( \theta_i; \hat{\theta}_i^{(l)} \right) = C'' + \frac{1}{\sigma_w^2} \sum_{n=T_i}^{T(i+1)-1} \left\{ 2\text{Re} \left( y_n^H B_n \text{diag}(\hat{s}_{n|1:T}) \right) \theta_i - \theta_i^T \left( \text{conj} \left( B_n^H B_n \right) \odot \left( \Sigma_{n|1:T} + \hat{s}_{n|1:T} \hat{s}_{n|1:T}^H \right) \right) \theta_i \right\}.
\]

Further, by denoting

\[
d_i^T = \sum_{n=T_i}^{T(i+1)-1} 2\text{Re} \left( y_n^H B_n \text{diag}(\hat{s}_{n|1:T}) \right)
\]

\[
\Phi_i = \sum_{n=T_i}^{T(i+1)-1} \left( \text{conj} \left( B_n^H B_n \right) \odot \left( \Sigma_{n|1:T} + \hat{s}_{n|1:T} \hat{s}_{n|1:T}^H \right) \right)
\]

November 20, 2014
we have

\[
Q \left( \theta_i; \hat{\theta}_{i}^{(l)} \right) = C'' + \frac{1}{\sigma_w^2} d_i^T \theta_i - \frac{1}{\sigma_w^2} \hat{\theta}_i^T \Phi \hat{\theta}_i. \tag{30}
\]

In summary, the E-step performs the Kalman smoothing operation in (17)-(24) to estimate \( \hat{s}_{n|1:T} \) and \( \sum_{n|1:T} \) and also operations in (28) and (29) to compute \( d_i^T \) and \( \Phi_i \) used in the computation of \( Q \left( \theta_i; \hat{\theta}_{i}^{(l)} \right) \).

2) **M-Step:** In the M-step, we find \( \theta_i \) maximizing \( Q \left( \theta_i; \hat{\theta}_{i}^{(l)} \right) \) in (30) as

\[
\hat{\theta}_{i}^{(l+1)} = \arg \max_{\theta_i \in \{0,1\}^M, \sum_{j=0}^{M-1} \theta_{i,j} = K} \hat{Q} \left( \theta_i; \hat{\theta}_{i}^{(l)} \right), \tag{31}
\]

where \( \hat{Q} \left( \theta_i; \hat{\theta}_{i}^{(l)} \right) = (d_i^T \theta_i - \theta_i^T \Phi_i \theta_i) \). In finding \( \hat{\theta}_{i}^{(l+1)} \), we need to check all possible combinations satisfying \( \sum_{j=0}^{M-1} \theta_{i,j} = K \). Since this brute force search is prohibitive for practical value of \( M \), we consider a computationally efficient tree search producing a sub-optimal solution to (31). The proposed approach, which in essence builds on the greedy search algorithm, examines candidate vectors to find out the most promising candidate of \( \theta_i \) 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 \( \theta_i = [0, \cdots, 0]^T \)), we construct the layer of the tree one at each iteration. In the first layer of the tree, only one entry of \( \theta_i \) is set to one. As
**TABLE I**
**SUMMARY OF THE GREEDY TREE SEARCH ALGORITHM**

| Input: $d_i$ and survival list $\Phi_i$ |
|-----------------------------------------|
| Initialization: Start with $\theta_i = [0, \cdots, 0]^T$ and $\Theta = \{[0, \cdots, 0]^T\}$. |
| for $k = 1 : K$ |
| for $j = 1 : R$ |
| for $m = 1 : M$ |
| Let $\theta$ 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 $\theta$ to one. |
| For $Q(x) \triangleq d_i^T x - x^T \Phi_i x$, evaluate $Q(\theta) - Q(\tilde{\theta})$ for all $\tilde{\theta} \in \Theta$. |
| If $Q(\theta) - Q(\tilde{\theta}) = 0$ for any $\tilde{\theta} \in \Theta$, then the candidate $\theta$ is duplicate node and hence we remove it. |
| If $Q(\theta) - \min_{\tilde{\theta} \in \Theta} Q(\tilde{\theta}) > 0$, add $\theta$ into $\Theta$. |
| end |
| end |
| end |
| Output: $\hat{\theta}^{(l+1)}_i = \arg \max_{\theta \in \Theta} Q(\theta)$. |

the layer increases, one additional entry is set to one and thus $K$ entries of $\theta_i$ are set to one in the $K$-th layer ($\|\theta_i\|_0 = K$) (see Fig. 2). At each layer of the tree, we evaluate the cost function $\hat{Q} \left( \theta_i; \hat{\theta}^{(l)}_i \right)$ 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 $\theta_i$ associated with the $R$ best nodes are called “survival list”. For example, the nodes in the first layer of the tree are expressed as $\theta^1_i = [1, 0, \cdots, 0]^T$, $\cdots$, $\theta^M_i = [0, \cdots, 0, 1]^T$. Among them, we choose the $R$ best nodes according to the path metric and then put those into the survival list. Next, for each node in the survival list, we construct the $M - 1$ child nodes in the second layer by setting one additional entry of $\theta_i$ to one. Note that since we do not distinguish the order of the bit assertion in $\theta_i$, two or more nodes might represent the same realization of $\theta_i$ during this process (see Fig. 2). When duplicate nodes are identified, we keep one and remove the rest from the tree. After removing all duplicate nodes, we choose the $R$ best nodes in the second layer of the tree and then move on to the next layer. This process is repeated until the tree reaches the bottom.

---

1For example, if $\theta_i = [1, 0, \cdots, 0]^T$ is in the survival list, then the child nodes of $\theta_i$ becomes $\theta_i = [1, 1, \cdots, 0]^T$, $\cdots$, $\theta_i = [1, 0, \cdots, 1]^T$. 

November 20, 2014 DRAFT
S
UMMARY OF THE PROPOSED ALGORITHM

Input: \( y_{1:T}, B_{1:T} \) for the \( i \)th block

STEP 1: Set \( l = 0 \). Start with \( \hat{\theta}^{(0)}_i = [1, \ldots, 1]^T \) and \( K^{(0)} \).

STEP 2: Run RTS Kalman smoother in (17) to (24).

STEP 3: Calculate \( d_i \) and \( \Phi_i \) in (28) and (29).

STEP 4: Obtain \( \hat{\theta}^{(l+1)}_i \) 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 \( \theta_i \) does not change (i.e., \( \hat{\theta}^{(l+1)}_i = \hat{\theta}^{(l)}_i \)) or the number of iterations reaches the limit, go to STEP 6.

STEP 6: Run Kalman smoother using \( \hat{\theta}^{(L)}_i \).

STEP 7: Obtain the final signal estimate from (32).

Output: \( \hat{h}_{1:T} \)

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

It is worth mentioning that one important issue to be discussed is how to estimate the sparsity parameter \( K \). One simple way to estimate the sparsity order \( K \) is to use the simple correlation method, where the observation vectors are correlated with the column vectors of \( B_n \) 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 would strongly depend on 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 \( Q \left( \theta_i, \hat{\theta}^{(l)}_i \right) \) is observed.

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

\[
\hat{h}_n = \text{diag}(\hat{\theta}^{(L)}_i)\hat{s}_{n|1:T}.
\]
C. 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 $\theta_i$ of the sparse signal vector under the sparsity constraint $\sum_{j=0}^{M-1} \theta_{i,j} = K$. Since the tree search to identify the support of $h_n$ 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 $\hat{Q}(\theta_i; \hat{\theta}_i^{(l)})$ is not so accurate. In order to reduce the chance of missing nonzero entries of sparse vector in early iterations, we search for the vector $\theta_i$ under relaxed sparsity constraint in the beginning and then gradually reduce the sparsity order $K$ as iterations go on. Let the sparsity order parameter used for the $l$th iteration be $K^{(l)}$. Then, we use sufficiently large value of $K^{(1)}$ initially\(^3\) and then gradually decreases $K^{(l)}$ through iterations, (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 $\theta_i$. The summary of the proposed algorithm is presented in Table II.

D. 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 of this modification is to produce the estimate of the source signal immediately whenever new measurement vector is available. In doing so, we can process the block seamlessly without waiting the reception of whole block of observations. First, instead of Kalman smoother, we employ the Kalman filter to conduct the operations of (17) to (21) in a forward direction. In order to ensure 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 consider two Kalman filters as an example\(^2\). In the first Kalman filter, we do not know the signal existence vector $\theta_i$ so that we set $\theta_i = [1, \cdots, 1]^T$ and run the

\(^{2}\)Note that we set $\theta_i^{(0)} = [1, \cdots, 1]$, i.e., $K^{(0)} = M$ since we have no knowledge on the sparse structure of the source signal vector in the beginning.

\(^{3}\)This setup corresponds to single EM iteration.
Kalman filter. Once $\hat{s}_{n|n}$ and $\Sigma_{n|n}$ are obtained, we compute $d_n^H$ and $\Phi_n$ in an auto-regressive update rule as

$$
\tilde{d}_n^H = (1 - \alpha)\tilde{d}_{n-1}^H + \alpha 2 \text{Re} \left( y_{n-1}^H B_{n-1} \text{diag}(\hat{s}_{n-1|n-1}) \right) 
$$

(33)

$$
\tilde{\Phi}_n = (1 - \alpha)\tilde{\Phi}_{n-1} + \alpha \left( \text{conj} \left( B_{n-1}^H B_{n-1} \right) \otimes \left( \Sigma_{n-1|n-1} + \hat{s}_{n-1|n-1} \hat{s}_{n-1|n-1}^H \right) \right) 
$$

(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 $d_n^H$ and $\Phi_n$ on the fly whenever the new measurement vector is available. Once $\tilde{d}_n^H$ and $\tilde{\Phi}_n$ are obtained, we next identify the signal existence indication vector $\theta_i$ using the greedy tree search. Using the newly obtained estimate $\hat{\theta}_i$, the second Kalman filter generates $s_{1:T}$. By multiplying this and $\hat{\theta}_i$, we get the final estimate of $h_{1:T}$. 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.

![Fig. 3. Real-time implementation of the sKTS algorithm.](image)

### III. Application to Channel Estimation

In this section, we study the application of the proposed scheme to the training-based channel estimation problems 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 wireless channels whose delay spread is larger than the number of significant paths are well modeled as a sparse signal vector in a discretized domain, the CS techniques have been used in the sparse channel estimation problem in [17]–[19]. While these 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 block 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 under-determined 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. Note that 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 sparse structure of the CIR vector, the sKTS algorithm overcomes the shortage of pilot signals. In the proposed scheme, we randomly allocate the pilot signals in time and frequency axis as shown in Fig. 4. Hence, while the support of \( h_n \) 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 \( p_n \in \mathbb{C}^N \)
and the observed signal vector $y_n \in \mathbb{C}^N$ of the OFDM system is expressed as

$$y_n = p_n \odot g_n + v_n$$

$$= \text{diag}(p_n)g_n + v_n$$

(35)

(36)

where $g_n$ is the vector representing frequency-domain channel response. Using the $P \times P$ DFT matrix $F_P$ whose $(i,j)$ entry is given by $e^{-j2\pi ij/P}$, the frequency-domain channel response is expressed in terms of the time-domain CIR as

$$g_n = \Pi_n F_P \Phi h_n,$$

(37)

where $h_n$ 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.,

$$\Pi_n = \begin{bmatrix} e_{c_n,1}^T \\ \vdots \\ e_{c_n,N}^T \end{bmatrix},$$

(38)

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

$$\Phi = [e_1 \cdots e_M].$$

(39)

Using (36) and (37), we have

$$y_n = B_n h_n + w_n,$$

(40)

where $B_n = \text{diag}(p_n)\Pi_n F_P \Phi$.

### 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 (see Fig. 5). Suppose that the length of the training symbols being transmitted is $N$, then the received signal at time $n$ is expressed as

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

(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
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)         | $K_n = \frac{\Sigma_{n-1} \text{diag}(\hat{\theta}_n)}{t_n^H \text{diag}(\hat{\theta}_n) \Sigma_{n-1} \text{diag}(\hat{\theta}_n) t_n + \sigma_n^2}$ |
| Eq. (28)         | $d_i^T \triangleq \sum_{n=T_i}^{T(i+1)-1} 2\text{Re}(\text{conj}(y_n) B_n \text{diag}(\hat{s}_n|1:T_i))$ |
| Eq. (29)         | $\Phi_i \triangleq \sum_{n=T_i}^{T(i+1)-1} \text{diag}(t_n^H (\Sigma_{n|1:T} + \hat{s}_n|1:T \hat{s}_n^H|1:T) \text{diag}(t_n)$ |

notation, we have

$$y_n = t_n^H h_n + w_n, \quad 0 \leq n \leq N,$$

(42)

where $h_n = [h_{n,0}, \ldots, h_{n,M-1}]^T$ and $t_n = [t_{n,0}^*, \ldots, t_{n-M-1}^*]^T$. Typically, recursive least square (RLS) and Kalman channel estimators have been widely used to estimate the channel vector $h_n$ [20], [21]. When the training period $N$ is small, the system becomes ill-posed and thus the estimation quality of these algorithms would be severely degraded. Whereas, by exploiting the block sparsity of $h_n$, the sKTS algorithm generates reliable estimate of $h_n$. Note that since the system matrix $B_n$ in the observation model (1) becomes a row vector $t_n^H$ in (42), each step of the algorithm can be simplified accordingly (see Table III).

IV. SIMULATIONS AND DISCUSSION

In this section, we study the performance of the proposed sKTS algorithm in the context of channel estimation scenarios we just described.

A. Simulation Setup

![Fig. 5. Transmission packet structure for SC systems.](#)
TABLE IV
PARAMETERS OF THE OFDM SYSTEMS

| Setup                                           | Specification |
|------------------------------------------------|---------------|
| Total number of subcarriers                     | 1024          |
| Bandwidth of each subcarrier                    | 15kHz         |
| Symbol duration                                 | 66.7μs        |
| CP length                                       | 16.7μs        |
| Interval between two consecutive pilot signals  | 0.25ms (three OFDM symbol) |
| Maximum delay spread of CIR                     | 13μs          |
| Sparsity order $K$                              | 8             |

1) **OFDM systems:** 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 every three OFDM symbols. For the OFDM symbol containing pilots, we allocate $N$ pilot symbols comprising the observation vector $y_n$. As described in the previous section, the location of pilot subcarriers is randomly chosen for each OFDM symbol. Considering the maximum channel delay spread specified in the Table IV, we set the dimension $M$ of $h_n$ to 200. The sparsity order $K$ is set to 8. In generating the complex Rayleigh fading frequency-selective channels, we use Jake’s model [22], where temporal correlation of the CIR taps for given Doppler spread $f_d$ (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 $J_0(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m!} \left( \frac{x}{2} \right)^{2m}$ 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 set the first (earliest) arrival path as a line of sight component and the rest ($K - 1$) paths based on Rayleigh fading model. Note that for every block of $T$ OFDM symbols, we change the delay structure of the CIR randomly.

2) **SC systems:** We consider the SC system whose transmission bandwidth is 8 MHz. Note that with the maximum delay spread of 13μs, the size of the CIR vector becomes $M = 104$ ($= 13 \cdot 10^{-6} \cdot 8 \cdot 10^6$). The rest of simulation parameters used in the SC system are similar to those in OFDM systems.

3) **Performance Test:** In our simulations, we evaluate the performance of the channel estimation 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 filter: standard Kalman filter [16] 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.
• Block OMP (B-OMP) ([17], [19]): The source signal is estimated from the $p$ measurement vectors. The $p$ measurement vectors are stacked to generate a new observation vector $\bar{y}$, which is expressed as

$$\bar{y} = \begin{bmatrix} y_{n-(p-1)/2} \\ \vdots \\ y_{n+(p-1)/2} \end{bmatrix} = \begin{bmatrix} B_{n-(p-1)/2} \\ \vdots \\ B_{n+(p-1)/2} \end{bmatrix} h_n + \begin{bmatrix} v_{n-(p-1)/2} \\ \vdots \\ v_{n+(p-1)/2} \end{bmatrix}. \hspace{1cm} (43)$$

Then, the OMP algorithm is applied to the new $Np \times 1$ measurement vector.

In order to determine the parameters of the Gauss-Markov process $A_i$ and $V_i$ for a given $f_d$, we minimize the approximation error between the Gauss-Markov process and the Jake’s model as suggested in [23]. As a metric to measure the estimation performance, we use the normalized mean square error (MSE) defined as $10 \log_{10} \frac{E[||h_n-h_n^*||^2]}{E[||h_n||^2]}$.

**B. Simulation Results**

We first investigate the sparse channel estimation performance for the OFDM system. In Fig. 6(a) and (b), we plot the MSE as a function of SNR for two distinct pilot densities ($N = 32$ and $N = 16$). For $N = 32$ and 16, the pilot signals occupy only 3.12 % and 1.06 % of resources, respectively. We set the Doppler rate to 0.05 and the number of the measurement vectors $T$ in a block to 20. We observe that when $N = 32$, the MSE performance of the sKTS algorithm is almost identical to that of the Oracle-based Kalman smoother over the whole SNR range under investigation. When $N = 16$, the performance of the sKTS is close to the Oracle bound in low and mid range SNR but there is slight performance gap in high SNR regime. Note that in both scenarios, the proposed scheme significantly outperforms the B-OMP and the conventional Kalman channel estimator. Since the B-OMP estimates the sparse channel under the assumption that the channel amplitude is invariant over the multiple observation vectors, it is no wonder that
the performance of this scheme does not improve with the number of measurement vectors $p$, in particular, when the channel amplitudes are changing fast. Whereas, the proposed sKTS scheme effectively tracks the variation of the sparse channel vector and identifies the support accurately so that estimation quality improves as the number of the observation vectors increases.

In order to observe that the gain in the channel vector estimation can be transferred to the symbol detection performance, we examine the symbol error rate (SER) performance when the estimated channel vector is used for data demodulation. In Fig. 7, we plot the SER as a function of SNR. In this simulations, we set $N = 32$ and use 16-QAM for data modulation. Similar to the MSE results we just observed, the proposed sKTS algorithm achieves dramatic SER performance gain over the other channel estimators under consideration. This shows that the proposed sKTS algorithm can achieve significant gain in demodulation performance, in particular when the pilot density is low.

We next investigate the performance of the proposed sKTS algorithm as a function of the number of pilots $N$ and the number of measurement vectors $T$. Fig. 8(a) shows the plot of the MSE performance of the sKTS algorithm for various sizes of measurement vector. It is shown that when the measurement size is large ($N \geq 32$), the sKTS algorithm achieves the Oracle bound over the whole SNR range under investigation. However, when the measurement size is small ($N < 32$), we can observe that the performance gap between two increases with the SNR. Fig. 8(b) shows the MSE performance for various measurement block sizes ($T = 8, 15, 20, \text{ and } 30$) when $N$ is set to 32. While the proposed sKTS algorithm performs close to the Oracle bound for a large block size ($T \geq 15$), a slight performance gap is observed with lower value of $T$.

Next, we compare the performance of the RT-sKTS described in Section II-D 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 of (33) and (33), we use the first 10 observation vectors for warming up purpose and then use the rest for measuring the MSE performance. Note that such warming up period is not typical since channel support would not be changed abruptly in many real applications. In Fig. 9, we see that the RT-sKTS algorithm achieves the performance 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 $\tilde{d}_n^H$ and $\tilde{\Phi}_n$. Nevertheless, as shown in Fig. 9 and Fig. 6 (a), the RT-sKTS algorithm maintains the performance gain over the conventional channel estimators.

Finally, we evaluate the performance of the sKTS algorithm for the SC systems. In Fig. 10, we plot the MSE performance of the channel estimators as a function of SNR. The length $N$ of the QPSK training symbols is set to 300 and Doppler rate is set to 0.05. In the B-OMP scheme, we set $p = 40$ (i.e., the number of observations used for channel estimation). We note that the B-OMP does not perform well even if $p$ is greater than 40 due to the variations of the channel gains. We see that the proposed sTKS method outperforms the conventional Kalman filter and the B-OMP scheme and also performs close to the Oracle-Kalman smoother.

V. Conclusions

In this paper, we studied the problem of estimating the time-varying sparse signals based on the sequence of the observation vectors. Exploiting the property that the support of sparse signals changes relatively slowly and thus can be well modeled as block sparse system, 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 for extracting the \textit{a posteriori} statistics of the source signals and the greedy tree search for identifying the support of the signal. From the case study of sparse channel estimation problem in orthogonal frequency division multiplexing (OFDM) and single carrier (SC) systems, we demonstrated that the proposed sKTS algorithm is effective in reconstructing sparse channel vectors.
APPENDIX A
DERIVATION OF (15)

From (9) and (14), we get

\[
Q \left( \theta; \hat{\theta}^{(l)} \right) = C' - \sum_{n=1}^{T(i+1)-1} E \left[ \frac{1}{\sigma_w^2} \| y_n - B_n \text{diag}(\theta_i) s_n \|^2 \right] \left( y_{1:T}; \hat{\theta}^{(l)}_i \right) 
\]

\[
= C'' + \frac{1}{\sigma_w^2} \sum_{n=1}^{T(i+1)-1} E \left[ \text{tr} \left( 2 \text{Re} \left( B_n \text{diag}(\theta_i) s_n y_n^H \right) \right) \right] \left( y_{1:T}; \hat{\theta}^{(l)}_i \right) 
\]

\[
- \text{tr} \left[ B_n \text{diag}(\theta_i) E \left[ s_n \ y_{1:T}; \hat{\theta}^{(l)}_i \right] \right] \text{diag}(B_n^H) 
\]

(44)

(45)

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

\[
Q \left( \theta; \hat{\theta}^{(l)} \right) = C'' + \frac{1}{\sigma_w^2} \sum_{n=1}^{T(i+1)-1} 2 \text{Re} \left( y_n^H B_n \text{diag}(\theta_i) E \left[ s_n \ y_{1:T}; \hat{\theta}^{(l)}_i \right] \right) 
\]

\[
- \text{tr} \left[ B_n \text{diag}(\theta_i) E \left[ s_n s_n^H \ y_{1:T}; \hat{\theta}^{(l)}_i \right] \right] \text{diag}(B_n^H) 
\]

(46)

(47)

APPENDIX B
DERIVATION OF (26)

Denoting \( b_{n,i} \) as the transpose of the ith row vector of \( B_n \), we can express the left term of (26) as

\[
\text{left term} = \sum_{j=0}^{M-1} b_{n,j}^T \text{diag}(\theta_i) \left( \Sigma_{n|1:T} + \hat{s}_{n|1:T} \hat{s}_{n|1:T}^H \right) \text{diag}(\theta_i) \text{conj}(b_{n,j}). 
\]

(49)

Since \( b_{n,j}^T \text{diag}(\theta_i) = \theta_i^T \text{diag}(b_{n,j}) \) and \( \text{diag}(\theta_i) \text{conj}(b_{n,j}) = \text{diag}(\text{conj}(b_{n,j})) \theta_i \), we further have

\[
\text{left term} = \sum_{j=0}^{M-1} \theta_i^T \text{diag}(b_{n,j}) \left( \Sigma_{n|1:T} + \hat{s}_{n|1:T} \hat{s}_{n|1:T}^H \right) \text{diag}(\text{conj}(b_{n,j})) \theta_i 
\]

(50)

\[
= \theta_i^T \sum_{j=0}^{M-1} \left[ \text{diag}(b_{n,j}^T) \left( \Sigma_{n|1:T} + \hat{s}_{n|1:T} \hat{s}_{n|1:T}^H \right) \text{diag}(\text{conj}(b_{n,j})) \right] \theta_i 
\]

(51)
and hence we finally have

$$\text{left term} = \theta_i^T \left( \text{conj} \left( B_n^H B_n \right) \odot \left( \Sigma_{n|1:T} + \hat{s}_{n|1:T} \hat{s}_{n|1:T}^H \right) \right) \theta_i. \quad (52)$$
Fig. 6. The plots of MSE as a function of SNR for (a) $N = 32$ and (b) $N = 16$. 
Fig. 7. The plots of SER as a function of SNR for $N = 16$. 
Fig. 8. The plots of MSE as a function of SNR for different values of (a) $N$ and (b) $T$.
Fig. 9. The performance of the RT-sKTS algorithm.
Fig. 10. The plot of MSE as a function of SNR in the SC systems.
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] 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.
[4] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction,” IEEE Trans. Information Theory, vol. 55, pp. 2230-2249, May 2009.
[5] D. P. Wipf and B. D. Rao, “Sparse Bayesian learning for basis selection,” IEEE Trans. Signal Processing, vol. 52, pp. 2153-2164, August 2004.
[6] S. Ji, Y. Xue, and L. Carin, “Bayesian compressed sensing,” IEEE Trans. Signal Processing, vol. 56, pp. 2346-2356, June 2008.
[7] 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.
[8] S. F. Cotter, B. D. Rao, K. Engan, K. Kreutz-Delgado, “Sparse solutions to linear inverse problems with multiple measurement vectors,” IEEE Trans. Signal Processing, vol. 53, pp. 2477-2488, July 2005.
[9] 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.
[10] D. P. Wipf and B. D. Rao, “An empirical Bayesian strategy for solving the simultaneous sparse approximation problem,” IEEE Trans. Signal Processing, vol. 55, pp. 3704-3716, July 2007.
[11] 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.
[12] 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.
[13] 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.
[14] 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.
[15] N. Vaswani, “Kalman filtered compressed sensing,” Proc. ICIP San Diego, CA USA, 2008, pp. 893-896.
[16] T. Kailath, A. H. Sayed, and B. Hassibi, Linear estimation, Prentice Hall, 2000.
[17] 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.
[18] 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 Processing, vol. 58, pp. 1708-1721, March 2010.
[19] S. F. Cotter and B. D. Rao, “Sparse channel estimation via matching pursuit with application to equalization,” IEEE Trans. Communications, vol. 50, pp. 374-377, March 2002.
[20] T. K. Akino, “Optimum-weighted RLS channel estimation for rapid fading MIMO channels,” IEEE Trans. Wireless Communications, vol. 7, pp.4248-4260, Nov. 2008.
[21] 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 Processing*, vol. 60, pp. 533-538, Jan. 2012.

[22] W. C. Jakes, “Microwave Mobile Communications,” New York, John Wiley & Sons Inc. 1975.

[23] K. E. Baddour and N. C. Beaulieu, “Autoregressive modeling for fading channel simulation,” *IEEE Trans. Wireless Commun.*, vol. 4, pp. 1650-1662, July 2005.