Sparse Detection with Integer Constraint Using Multipath Matching Pursuit

Byonghyo Shim, Suhyuk Kwon, Byungkwen Song

***Abstract*—In this paper, we consider a detection problem of the underdetermined system when the input vector is sparse and its elements are chosen from a set of finite alphabets. This scenario is popular and embraces many of current and future wireless communication systems. We show that a simple modification of multipath matching pursuit (MMP), recently proposed parallel greedy search algorithm, is effective in recovering the discrete and sparse input signals. We also show that the addition of cross validation (CV) to the MMP algorithm is effective in identifying the sparsity level of input vector.**

***Index Terms*—Compressed sensing, sparse signal recovery, greedy algorithms, multipath matching pursuit (MMP).**

I. Introduction

A. System Model

The relationship between a transmit signal $\mathbf{x}$ and a received signal vector $\mathbf{y}$ in many wireless communication systems can be expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} \tag{1}$$

where $\mathbf{H} \in \mathbb{C}^{m\times n}$ is the system (channel) matrix, $\mathbf{v} \sim \mathcal{N}(0, \sigma_v^2\mathbf{I})$ is the noise vector, and $\mathbf{x}$ is the input vector whose entries are chosen from a finite set of integer Ω. In this work, we are primarily concerned with the scenario where 1) the input signal $\mathbf{x}$ is sparse (i.e., number of nonzero elements in a signal vector is small) and 2) the dimension of observation vector $\mathbf{y}$ is smaller than that of the input vector ($m < n$). This system, which in essence is modeled as the underdetermined sparse system, is prevalent and embraces many of modern wireless communication systems such as wireless sensor network, source localization, multiuser detection, downlink in massive MIMO, relaying in ad hoc network, to name just a few.

B. Conventional Detection

The traditional way of detecting the input signals is to use the estimation technique such as least squares (LS) or linear minimum mean square error (LMMSE) estimation followed by the symbol slicing. Let $\tilde{\mathbf{x}}$ and $\hat{\mathbf{x}}$ be the output of LMMSE estimator and its sliced version, respectively, then

$$\tilde{\mathbf{x}} = (\mathbf{H}^T\mathbf{H} + \sigma_v^2\mathbf{I})^{-1}\mathbf{H}^T\mathbf{y} \tag{2}$$

$$\hat{\mathbf{x}} = Q_\Omega(\tilde{\mathbf{x}}) \tag{3}$$

B. Shim and S. Kwon are with Dept. of Electrical and Computer Engineering, Seoul National Univ., Seoul, Korea, 151-742 (email: bshim@snu.ac.kr), and B. Song is with Dept. of Electronics Engineering, Seokyeong Univ., Seoul, Korea, 136-701 (email: bksong@skuniv.ac.kr).

This research was supported by the research grant of Seokyeong University (2013) and National Research Foundation of Korea grant funded by Korea government (MEST) No. 2013056520.

where $Q_\Omega(\cdot)$ is the integer quantizer (a.k.a slicing function) which maps the input to the closest value in Ω (i.e., $Q_\Omega(z) = \arg\min_{\omega\in\Omega} \|z-\omega\|_2$). Since the system is underdetermined, these approaches, which attempt to find a solution by inverting the *entire* covariance matrix, do not perform well in general. Due to the fact that nonzero elements of the signal vector are chosen from the set of finite integers, one can alternatively consider the detection strategy (e.g., sphere decoding for maximum likelihood detection [1], [2]). Although it is desirable to exploit the discrete property of transmit signals, since the algorithm should be performed under the underdetermined system model, fundamental performance gap from the overdetermined system is unavoidable.

C. Detection via Sparse Signal Recovery Algorithm

A better treatment of the problem at hand is to use sparse recovery algorithm. In essence, the goal of the sparse recovery algorithm is to find the sparest set of atoms (column $\mathbf{h}_i$ of $\mathbf{H}$) that best represents the observation $\mathbf{y}$. In doing so, system model is converted from underdetermined to overdetermined and an accurate recovery of the original sparse signal is possible. While a dictionary (a collection of atoms) used in the signal/image processing (e.g., Fourier, Wavelet, Haar dictionaries) can be tailored to the design purpose, the dictionary in the communication systems is mostly in the form of channel matrix and needs to be estimated using the known signal so called the pilot signal.

Over the years many algorithms to find the sparsest representation of the observation vector from the (overcomplete) dictionary have been proposed. Two popular approaches are ℓ_1-minimization technique and greedy pursuit algorithm:

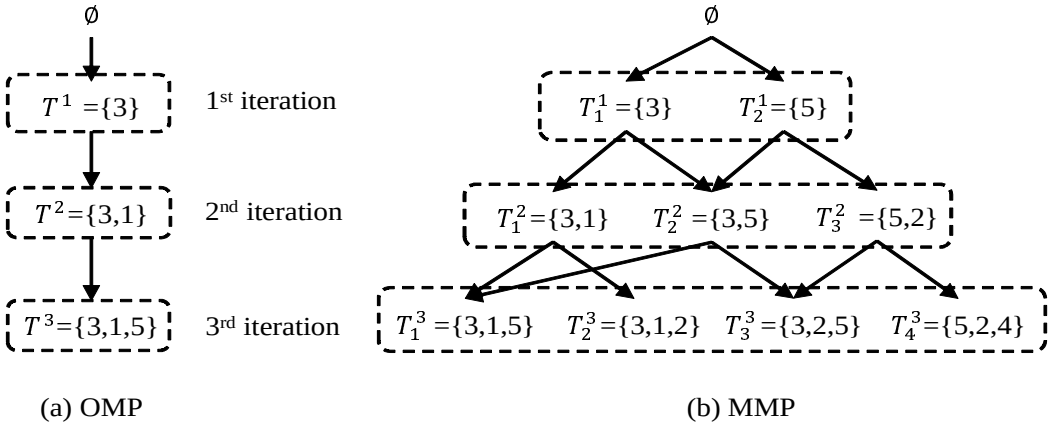
- **ℓ_1-minimization**: ℓ_1-minimization technique solves the problem

$$\min \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2 < \epsilon, \tag{4}$$

where $\epsilon > 0$ (in the noiseless setting, $\epsilon = 0$). In [3], it is shown that if the noise power is limited to ϵ and the number of observations is sufficiently large, ℓ_2-norm of the reconstruction error is within the constant multiple of ϵ (i.e., $\|\hat{\mathbf{x}}-\mathbf{x}\|_2 < c_0\epsilon$). Basis pursuit de-noising (BPDN) [4], also called Lasso [5], relaxes the hard constraint on the reconstruction error by introducing a soft weight λ as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2 + \lambda\|\mathbf{x}\|_1, \tag{5}$$

Since the ℓ_1-minimization problem is convex optimization problem, efficient solvers based on the linear programming (LP) exist (e.g., BP-interior [4]). From our



(a) OMP (b) MMP

Fig. 1. Comparison between the OMP and the MMP algorithm ($L = 2$ and $K = 3$). While OMP maintains a single candidate T^k in each iteration, MMP investigates multiple promising candidates T_j^k (subscript j counts the candidate in the i-th iteration).

perspective, unfortunately, it is not easy to incorporate the integer constraint into the LP based algorithm.

- **Greedy Pursuit**: Greedy pursuit attempts to find the support (index set of nonzero entries) of the input vector[1] in an iterative fashion, obtaining a sequence of possible estimates $(\hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_n)$. For example, orthogonal matching pursuit (OMP) picks a column of the channel matrix $\mathbf{H}$ one at a time using a greedy strategy [6]–[8]. In the k-th iteration, the estimate $\hat{\mathbf{x}}_k$ is generated by projecting the observation into the subspace spanned by the submatrix contructed from chosen columns.

While the use of sparse recovery algorithm is desirable in the sense that it exploits the sparsity of signal to be recovered, in itself it does not exploit the property that the nonzero element of $\mathbf{x}$ is from the set of finite alphabets. To exploit this information, one can use the sliced output $Q_\Omega(\hat{\mathbf{x}}_k)$ instead of $\hat{\mathbf{x}}_k$ as an estimate in each iteration. However, just using the slicer output $Q_\Omega(\hat{\mathbf{x}}_k)$ might not be effective, in particular for the sequential greedy algorithms like OMP, due to the error propagation. For example, if an incorrect index is chosen in an iteration of OMP, then the estimate would be incorrect. Furthermore, if this incorrect estimate is sliced, additional quantization error will be added on top of the estimation error, exacerbating the quality of the subsequent operation. For example, if $x_k = 0$ and $\hat{x}_k = 0.01$, then $Q_\Omega(\hat{x}_k) = 1$ for $\Omega = \{+1, -1\}$ (and hence $\|x_k - \hat{x}_k\| = 0.01$ and $\|x_k - Q_\Omega(\hat{x}_k)\| = 1$).

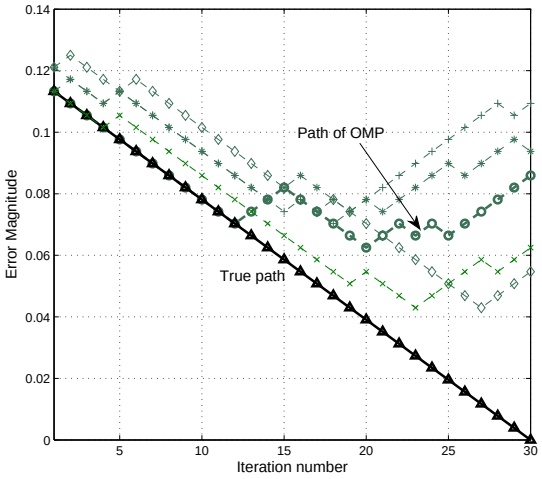


Fig. 2. Evolution of the error magnitude as a function of the iteration number of the proposed sMMP algorithm. For illustration purpose, we plot only a few paths. While the quantization error introduced by the integer slicer affects the incorrect paths (green colored paths in the figure), no such phenomenon happens to the true path. As a result, the performance difference between the true path and incorrect paths grows drastically as an iteration goes on.

II. Multipath Matching Pursuit (MMP) with Slicing

A. MMP Algorithm

While many of greedy recovery algorithms construct K-sparse estimate through the sequential process, recently proposed algorithm referred to as multipath matching pursuit (MMP) performs the parallel search to find multiple promising candidates and then chooses the best one minimizing the residual magnitude in the last minute [9]. As shown in Fig. 1, each candidate brings forth L child candidates in the MMP algorithm. In the k-th iteration, L indices $t_1, \cdots t_L$ whose columns are maximally correlated with the residual are chosen and each of these indices, in conjunction with previously selected indices, constructs a new candidate in the next iteration. Let $T_j^{k-1} = \{t_1, \cdots, t_{k-1}\}$ be the j-th candidate in the $(k-1)$-th iteration, then the set of L indices chosen from T_j^{k-1}, denoted as T^*, is expressed as

$$T^* = \arg\max_{|T|=L} \|(\mathbf{\Phi}'\mathbf{r}_j^{k-1})_T\|_2^2$$

[1]For example, if $\mathbf{x} = [1\ 0\ 0\ 0\ 2\ 0\ 4]^T$, then the support S is $S = \{1, 5, 7\}$.

TABLE I
THE SMMP ALGORITHM

Input: observation $\mathbf{y}$, channel matrix $\mathbf{H}$, modulation set Ω
sparsity K, number of path L
Output: estimated signal $\hat{\mathbf{x}}$
Initialization:
$k := 0$ (iteration index), $\mathbf{r}^0 := \mathbf{y}$ (initial residual), $T^0 := \{\emptyset\}$
while $k < K$ **do**
  $k := k + 1$, $u := 0$, $T^k := \emptyset$
  for $i = 1$ **to** $|T^{k-1}|$ **do**
    $T^* := \arg\max_{|T|=L} \|(\mathbf{H}'\mathbf{r}_i^{k-1})_T\|_2^2$ (*choose L best indices*)
    for $j = 1$ **to** L **do**
      $t_{tmp} := t_i^{k-1} \cup \{t_j\}$ (*construct a temporary path*)
      if $t_{tmp} \notin T^k$ **then** (*check if the path already exists*)
        $u := u + 1$ (*candidate index update*)
        $t_u^k := t_{tmp}$ (*path update*)
        $T^k := T^k \cup \{t_u^k\}$ (*update the set of path*)
        $\hat{\mathbf{x}}_u^k := (\mathbf{H}_{t_u^k}^T \mathbf{H}_{t_u^k})^{-1} \mathbf{H}_{t_u^k}^T \mathbf{y}$ (*perform estimation*)
        $\tilde{\mathbf{x}}_u^k := Q_\Omega(\hat{\mathbf{x}}_u^k)$ (**perform slicing**)
        $\mathbf{r}_u^k := \mathbf{y} - \mathbf{H}_{t_u^k} \hat{\mathbf{x}}_u^k$ (*residual update*)
      end if
    end for
  end for
end while
$u^* := \arg\min_u \|\mathbf{r}_u^K\|_2$
return $\hat{\mathbf{x}} = \hat{\mathbf{x}}_{u^*}^k$

where the residual is defined as $\mathbf{r}_j^{k-1} = \mathbf{y} - \mathbf{H}_{k-1}\hat{\mathbf{x}}_j^{k-1}$ and

$$\begin{aligned} \hat{\mathbf{x}}_j^{k-1} &= \mathbf{H}_{k-1}^{\dagger}\mathbf{y} = (\mathbf{H}_{k-1}^T\mathbf{H}_{k-1})^{-1}\mathbf{H}_{k-1}^T\mathbf{y}, \\ \mathbf{H}_{k-1} &= [\mathbf{h}_{t_1}\ \mathbf{h}_{t_2}\ \cdots\ \mathbf{h}_{t_{k-1}}]. \end{aligned} \tag{6}$$

The corresponding child candidates become $T^{k-1}\bigcup\{t_i\}$ for $i = 1, \cdots, L$. Although the number of candidates seems to increase by the factor of L in each iteration, due to the fact that many candidates are overlapping (see Fig. 1), actual number of child candidates is quite moderate. The main benefit of MMP, in the perspective of incorporating the integer slicer, is that it deteriorates the quality of incorrect candidate and at the same time improves the quality of correct one. This is because the quality of incorrect candidates gets worse due to the additional quantization noise caused by the slicing while no such phenomenon happens to the correct one. As a result, as shown in Fig. 2, the difference of estimation quality, measured in terms of the estimation error $\|\mathbf{x} - \hat{\mathbf{x}}\|_2$, between the final output (mostly it corresponds to the true path) and the rest (incorrect paths) increases as iteration goes on. The MMP algorithm with slicing, henceforth referred to as sMMP, is summarized in Table I.

B. Multipath Matching Pursuit with Cross Validation

Thus far, we assumed that the sparsity of the input vector ($K = \|\mathbf{x}\|_0 = \#\{x_i; x_i \neq 0\}$) is known in advance. Indeed, many greedy pursuit algorithms implicitly assume that the sparsity of the input vector is known in *a priori*. Since this assumption does not hold true in practice, absence of sparsity information can lead to either early or late termination of the recovery algorithm. In the former case, the transmit signal will not be fully recovered (*underfitting*), while in the latter case some of the noise vector is treated as the transmit signal (*overfitting*). In both scenarios, the reconstruction quality will be affected considerably (in general, more harmful to underfitting).

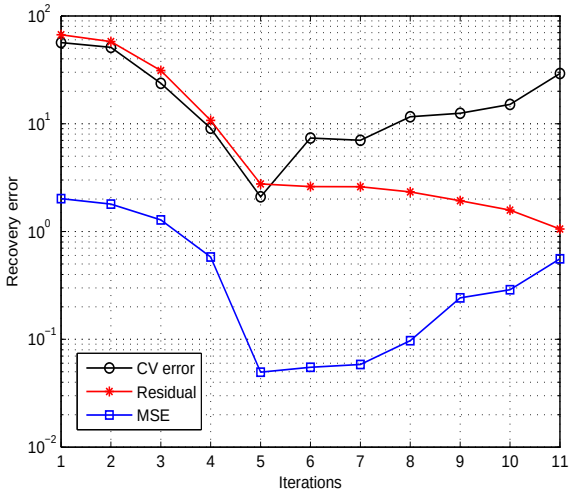


Fig. 3. Snapshot of the recovery error as a function of the iteration number ($m = 12$, $n = 24$, and $K = 5$).

Cross validation (CV) is a statistical technique to identify the model order (sparsity level K in this work) and thus avoid overfitting and underfitting of the parameter [10]. In CV, the received vector $\mathbf{y}$ is divided into a training vector $\mathbf{y}^{(t)}$ and a validation vector $\mathbf{y}^{(v)}$, which are given respectively by

$$\mathbf{y}^{(t)} = \mathbf{H}^{(t)}\mathbf{x} + \mathbf{v}^{(t)} \tag{7}$$

$$\mathbf{y}^{(v)} = \mathbf{H}^{(v)}\mathbf{x} + \mathbf{v}^{(v)}. \tag{8}$$

Using the training vector $\mathbf{y}^{(t)}$, a sequence of possible estimates $(\hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_n)$ is generated. For each estimate $\hat{\mathbf{x}}_i$, the validation vector $\mathbf{y}^{(v)}$ is used to compute the estimation error $\epsilon_k = \|\mathbf{y}^{(v)} - \mathbf{H}^{(v)}\hat{\mathbf{x}}_k\|_2$. When the algorithm is finished, an iteration count corresponding to the minimum validation error becomes the sparsity estimate ($\hat{K} = \arg\min_k \epsilon_k$) and $\hat{\mathbf{x}}_{\hat{K}}$ is chosen as the final output.

It is worth mentioning that in many applications, the residual based stopping criterion is widely used to identify the sparsity level (or iteration number) of the greedy algorithm. Basically, this scheme terminates the algorithm when the residual power is smaller than the pre-specified threshold ϵ (i.e., $\|\mathbf{r}^k\|_2 < \epsilon$). However, since the residual magnitude decreases monotonically and the rate of decay depends on the system parameters, it is not easy to identify the optimal point. Whereas, the ℓ_2-norm of the validation error ϵ_i usually has minimum value when an iteration number equals the sparsity level so that one can easily estimate the sparsity using CV (see Fig. 3). Finally, we note that since multiple candidates are investigated in sMMP, a candidate with the minimum validation error among all candidates is chosen as the final output.

III. SIMULATIONS AND DISCUSSIONS

This section describes the numerical experiments that illustrate effectiveness of the proposed approach. Other than the

proposed sMMP algorithm, we test the original MMP algorithm, OMP algorithm (with and without slicing[2]), CoSaMP algorithm [11], LMMSE estimation, and also Oracle LMMSE estimation. Note that Oracle estimator knows the support information in *a priori* and hence it solves the overdetermined system $\mathbf{y} = \mathbf{H}_T\mathbf{x}_T + \mathbf{v}$ where $\mathbf{H}_T$ is the submatrix of $\mathbf{H}$ containing columns indexed by T ($\mathbf{x}_T$ is defined in the same way). The performance of Oracle estimator is popularly used as a lower bound of the recovery algorithms.

A. Simulation Setup

The simulation setup is based on 12×24 channel matrix $\mathbf{H}$ whose entries drawn independently from complex Gaussian distribution $\mathcal{CN}(0, 1)$. Two sparsity levels ($K = 3$ and 5) are tested so that 12.5% and 20% of elements in $\mathbf{x}$ are nonzero. The positions of nonzero elements (i.e., symbol positions) are randomly selected and symbols of nonzero locations are chosen from 16-QAM modulation. We use the symbol error rate (SER) as a performance measure. For each point in the plot, we perform 24 trials for the CV operation and the average value of these trials is used as a sparsity estimate $\hat{K}$. Also, to measure the performance, we perform at least $5,000$ trials for each point of the tested algorithm.

B. Simulation Results

Fig. 4 shows the SER performance as a function of signal-to-noise-ratio (SNR). Since the system is underdetermined, LMMSE estimator exploiting whole channel matrix to estimate the sparse signal vector does not perform well. Whereas, performance of all sparse recovery algorithms we tested improves with the SNR. Due to the fact that multiple promising candidates are investigated, it is no wonder that the MMP algorithm exhibits the best performance among sparse recovery algorithms under test. As mentioned, since the slicing is effective in improving performance of MMP, the sMMP algorithm outperforms the original MMP by more than 1 dB gain. In Fig. 5, we plot the performance for $K = 5$ (i.e., 20% of signal vectors is active). While the performance of conventional sparse recovery algorithms is not so appealing for this less sparse scenario, the proposed sMMP algorithm is effective and performs close to the Oracle estimator (around 2 dB gap at 10^{-2} SER).

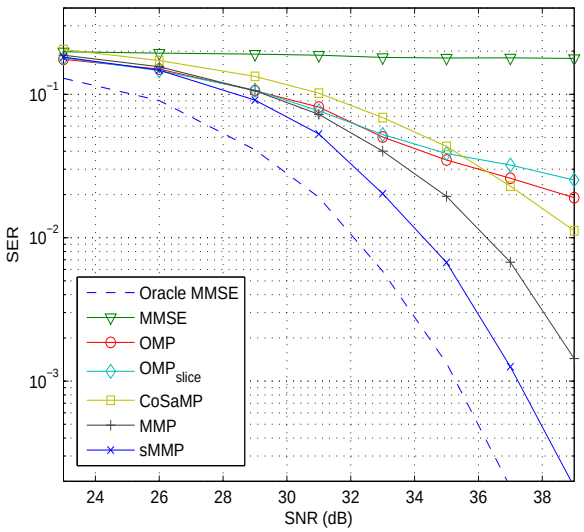


Fig. 4. Performance of the proposed method for $m = 12$, $n = 24$, and $K = 3$.

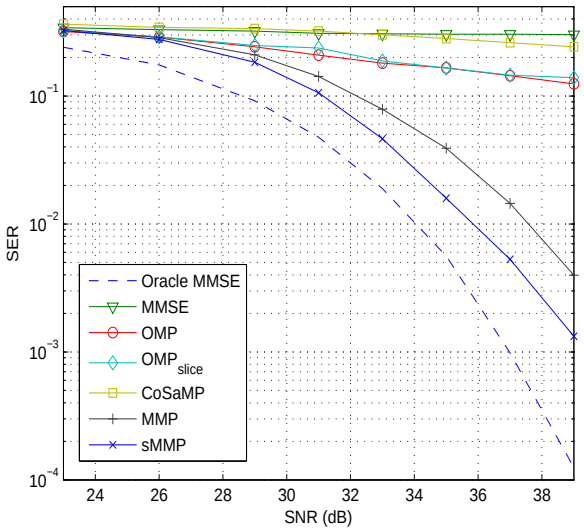


Fig. 5. Performance of the proposed method for $m = 12$, $n = 24$, and $K = 5$.

REFERENCES

[1] T. Cui and C. Tellambura, "An efficient generalized sphere decoder for rank-deficient MIMO systems," *IEEE Communications Letters*, vol. 9, no. 5, pp. 423-425, May 2005.

[2] B. Shim and I. Kang, "Sphere decoding with a prbabilistic tree pruning," *IEEE Trans. Signal Process.*, vol. 56, pp. 4867-4878, Oct. 2008.

[3] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207-1223, Aug. 2006.

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

[5] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, no. 1, pp. 267-288, 1996.

[2]In the OMP algorithm with slicing, slicing is performed for each iteration.

[6] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. of Asilomar Conf.*, pp. 40-44, Nov. 1993.

[7] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inform. Theory*, vol. 53, no. 12, pp. 4655-4666, Dec. 2007.

[8] J. Wang, S. Kwon, and B. Shim, "Generalized orthogonal matching pursuit," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6202-6216, Dec. 2012.

[9] S. Kwon, J. Wang, and B. Shim, "Multipath matching pursuit," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2986-3001, May 2014.

[10] R. Ward, "Compressed sensing with cross validation," *IEEE Trans. Inf. Theory*, vol. 55, no. 12, pp. 5773-5782, Dec. 2009.

[11] D. Needell and J. A. Tropp, "CoSaMP: iterative signal recovery from incomplete and inaccurate samples," *Commun. ACM*, vol. 53, no. 12, pp. 93-100, Dec. 2010.