

BEAMFORMERS FOR SPARSE RECOVERY

Martin Sundin, Dennis Sundman, Magnus Jansson

ACCESS Linnaeus Center, Electrical Engineering
KTH Royal Institute of Technology, Stockholm, Sweden

masundi@kth.se, denniss@kth.se, magnus.jansson@ee.kth.se

ABSTRACT

In sparse recovery from measurement data a common approach is to use greedy pursuit reconstruction algorithms. Most of these algorithms have a correlation filter for detecting active components in the sparse data. In this paper, we show how modifications can be made for the greedy pursuit algorithms so that they use beamformers instead of the standard correlation filter. Using these beamformers, improved performance in the algorithms is obtained. In particular, we discuss beamformers for the average and worst case scenario and give methods for constructing them.

Index Terms— Compressed sensing, Greedy pursuit algorithms, Beamforming.

1. INTRODUCTION

An active field of research is the estimation of sparse vectors from linear measurements. A particularly interesting setup is compressive sensing, where it has been proven that far fewer random measurements are sufficient to fully reconstruct a sparse vector than a dense vector. This problem has received much attention [1, 2] and holds promises for several applications [3, 4, 5]. Several approaches are available for the reconstruction problem; combinatorial searches, convex relaxations [6] and greedy pursuit (GP) algorithms. Of these approaches, the GP algorithms provide promising performances at a low computational cost. Some common GP algorithms are Orthogonal matching pursuit (OMP) [7], Subspace pursuit (SP) [8] and CoSaMP [9], which all use a matched filter to find the support set.

The compressed sensing problem can be stated as

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \quad \text{s.t.} \quad \|\mathbf{x}\|_0 = K \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, with $m \ll n$, has column vectors of unit length, $\mathbf{x} \in \mathbb{R}^n$ and $\|\mathbf{x}\|_0 = |\text{supp}(\mathbf{x})|$ is the number of non-zero elements of \mathbf{x} . Many GP algorithms estimate the support set iteratively by selecting elements maximizing $|\mathbf{a}_i^\top \mathbf{y}|$ (i.e. a matched filter) and estimate non-zero vector components using least squares techniques.

In this paper, we investigate the possibility of improving the performance of GP algorithms by replacing the matched filter by a *beamformer*, i.e. we seek vectors \mathbf{b}_i such that the support set of a sparse vector can be estimated by selecting elements maximizing $|\mathbf{b}_i^\top \mathbf{y}|$. The goal of this paper is to find vectors \mathbf{b}_i that improves the performance of GP. The idea to use beamformers was first introduced by Schnass and Vandergheynst [10] which referred to the beamformer as a *sensing dictionary*. We extend the analysis of [10] by analyzing the average and worst-case scenarios for sparse recovery and deriving alternative beamformers. Finally we numerically

compare our results with the ones in [10]. The main contributions of this paper are:

- We analyze beamformers for the average and worst case scenario.
- We propose convex methods for calculating the beamformer.
- We give simple expressions for the average case beamformer.
- We give bounds for the probability of recovery in the noisy case.
- We numerically evaluate the beamformers and compare with the results in [10].

In particular, we exemplify the procedure by presenting necessary modifications to the OMP algorithm so that it can incorporate the beamformer. We call this algorithm OMPb and in the numerical evaluation we see how much performance gain is obtained for OMPb with some different beamformers. Although we focus on, and modify, the OMP algorithm, these results can be extended to most GP algorithms.

2. BEAMFORMER FOR NOISE-FREE RECOVERY

As mentioned in the introduction, we are in this paper interested in modifying the well known OMP algorithm so that it, instead of the matched filter, can correlate a residual with some other vector in order to find prominent support set indices. We present this modification in Algorithm 1.

Here $\mathbf{A}_{\hat{I}}$ denotes the submatrix formed by the columns in \hat{I} and $\hat{\mathbf{x}}_{\hat{I}}$ is the vector with elements from $\hat{\mathbf{x}}$ with indices in \hat{I} . If we set $\mathbf{B} = \mathbf{A}$, OMPb will reduce to regular OMP. Here, we are interested in finding $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n)$ so that OMPb performs better than OMP. Observe that regardless of beamformer \mathbf{B} provided to the algorithm, the complexity remains the same as that of OMP.

Algorithm 1 OMPb: OMP with beamformer

Input: $\mathbf{y}, \mathbf{A}, \mathbf{B}, K$

Initialization:

1: $\mathbf{r} = \mathbf{y}, \hat{I} = \emptyset, \hat{\mathbf{x}} = \mathbf{0}$.

Iteration:

1: **repeat**

2: $i = \arg \max_{j \notin \hat{I}} |\mathbf{b}_j^\top \mathbf{r}|$

3: $\hat{I} \leftarrow \hat{I} \cup \{i\}$

4: $\hat{\mathbf{x}}_{\hat{I}} = \mathbf{A}_{\hat{I}}^+ \mathbf{y}$

5: $\mathbf{r} = \mathbf{y} - \mathbf{A}\hat{\mathbf{x}}$

6: **until** $|\hat{I}| = K$ or $\|\mathbf{r}\|_2 \leq \epsilon$.

Output: $\hat{I}, \hat{\mathbf{x}}$

This work was partially supported by the Swedish Research Council under contract 621-2011-5847.

OMPb will in the first iteration choose an element in the support set if

$$\max_{i \in I} |\mathbf{b}_i^\top \mathbf{y}| > \max_{j \notin I} |\mathbf{b}_j^\top \mathbf{y}|. \quad (2)$$

If x_i is the element of \mathbf{x} of maximum modulus, (2) holds provided that

$$|\mathbf{b}_i^\top \mathbf{a}_i| > \sum_{j \in I \setminus \{i\}} |\mathbf{b}_i^\top \mathbf{a}_j| + \max_{k \notin I} \sum_{j \in I} |\mathbf{b}_k^\top \mathbf{a}_j|. \quad (3)$$

2.1. Worst Case Scenario

In order to maximize the worst case performance, we can minimize the right hand side of (3) while keeping the left hand side constant. However, in order to solve this problem, knowledge of the support set is required. We therefore instead choose to minimize the contribution of all possible support sets while keeping $\mathbf{b}_i^\top \mathbf{a}_i$ fixed, i.e.

$$\mathbf{b}_i = \arg \min_{\mathbf{b}} \left(\max_{|J| \leq K, i \notin J} \sum_{j \in J} |\mathbf{b}^\top \mathbf{a}_j| \right) \quad \text{s.t.} \quad \mathbf{b}^\top \mathbf{a}_i = 1. \quad (4)$$

The optimization problem (4) is convex and can be solved using methods from [11, 12] or programs like cvx [13]. The matrix \mathbf{B} minimizes the contribution from the *sidelobes* if we make the connection to array signal processing [14]. For this reason we refer to \mathbf{B} as a beamformer. When K is unknown, an alternative is to construct \mathbf{B} with $K = 1$, i.e. only minimize the largest sidelobe.

2.2. Average Case Scenario

When the support set is drawn at random and \mathbf{x}_I is a random variable, one approach to improving the average performance of GP algorithms to minimize the probability that the algorithm chooses an index not belonging to the support set. While the average case analysis with random coefficients and support set is difficult in general, the following lemma gives us some insight to the case where the non-zero elements are drawn from a Gaussian distribution.

Lemma 1. Let $\mathbf{c}, \mathbf{d} \in \mathbb{R}^n$ and let $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma_z^2 \mathbf{I})$, where \mathbf{I} is the $n \times n$ identity matrix, then

$$P(|\mathbf{c}^\top \mathbf{z}| > |\mathbf{d}^\top \mathbf{z}|) = \frac{1}{\pi} \arccos \left(\frac{||\mathbf{d}||_2^2 - ||\mathbf{c}||_2^2}{||\mathbf{c} - \mathbf{d}||_2 \cdot ||\mathbf{c} + \mathbf{d}||_2} \right)$$

Proof: By symmetry we have that

$$\begin{aligned} \bar{P} &= P(|\mathbf{c}^\top \mathbf{z}| > |\mathbf{d}^\top \mathbf{z}|) \\ &= 2P(\mathbf{c}^\top \mathbf{z} \geq 0, (\mathbf{c} - \mathbf{d})^\top \mathbf{z} \geq 0, (\mathbf{c} + \mathbf{d})^\top \mathbf{z} \geq 0) \end{aligned}$$

The last probability is given by the angle θ between the hyperplanes $(\mathbf{c} - \mathbf{d})^\top \mathbf{x} = 0$ and $(\mathbf{c} + \mathbf{d})^\top \mathbf{x} = 0$ divided by 2π . The angle between the hyperplanes is π minus the angle between the normal vectors. Using that $\pi - \arccos(t) = \arccos(-t)$ we find the probability

$$\begin{aligned} \bar{P} &= \frac{2}{2\pi} \arccos \left(-\frac{(\mathbf{c} - \mathbf{d})^\top (\mathbf{c} + \mathbf{d})}{||\mathbf{c} - \mathbf{d}||_2 \cdot ||\mathbf{c} + \mathbf{d}||_2} \right) \\ &= \frac{1}{\pi} \arccos \left(\frac{||\mathbf{d}||_2^2 - ||\mathbf{c}||_2^2}{||\mathbf{c} - \mathbf{d}||_2 \cdot ||\mathbf{c} + \mathbf{d}||_2} \right). \end{aligned}$$

This completes the proof. \square

When $\text{Cov}(\mathbf{z}) = \mathbf{C}$ we may pre-whiten \mathbf{z} by setting $\tilde{\mathbf{c}}_i = \mathbf{C}^{-1/2} \tilde{\mathbf{c}}_i$. The inner products of \mathbf{c}_i and \mathbf{c}_j are then replaced by inner products of $\tilde{\mathbf{c}}_i$ and $\tilde{\mathbf{c}}_j$ which can be computed using that $\tilde{\mathbf{c}}_i^\top \tilde{\mathbf{c}}_j = \mathbf{c}_i^\top \mathbf{C} \mathbf{c}_j$.

We can use Lemma 1 for the average case scenario by setting

$$\mathbf{c}_i = \mathbf{b}_i^\top \mathbf{A}_I, \mathbf{z} = \mathbf{x}_I$$

From Lemma 1 we get that the probability to choose $i \in I$ over $j \notin I$ is large when $||\mathbf{c}_i||_2$ is large and $||\mathbf{c}_j||_2$ is small. Since the support set is unknown, we minimize the maximum length of \mathbf{c}_i over all support sets while keeping $\mathbf{b}_i^\top \mathbf{a}_i$ fixed for all i , i.e. we choose the column vectors in \mathbf{B} as

$$\mathbf{b}_i = \arg \min_{\mathbf{b}} \left(\max_{|J| \leq K, i \notin J} \sum_{j \in J} (\mathbf{b}^\top \mathbf{a}_j)^2 \right) \quad \text{s.t.} \quad \mathbf{b}^\top \mathbf{a}_i = 1. \quad (5)$$

This is a convex optimization problem and can be solved using methods from e.g. [12]. Setting $K = n - 1$ gives the explicit solution

$$\mathbf{b}_i = \frac{(\mathbf{A} \mathbf{A}^\top)^{-1} \mathbf{a}_i}{\mathbf{a}_i^\top (\mathbf{A} \mathbf{A}^\top)^{-1} \mathbf{a}_i}, \quad (6)$$

that is $\mathbf{B} = (\mathbf{A}^+)^T \mathbf{D}$ where \mathbf{D} is a diagonal matrix with entries $D_{ii} = 1/(\mathbf{A}^+ \mathbf{A})_{ii}$ and \mathbf{A}^+ is the Moore-Penrose pseudoinverse of \mathbf{A} . Note that (6) can be interpreted as a Capon method for recovery of sparse random vectors [14].

Another motivation for using the pseudoinverse as a beamformer is to choose \mathbf{B} so that $\mathbf{B}^\top \mathbf{A} \mathbf{x}$ is as close as possible to \mathbf{x} in the mean square sense, i.e. we choose \mathbf{B} to minimize

$$\begin{aligned} \mathbb{E}[||\mathbf{x} - \mathbf{B}^\top \mathbf{A} \mathbf{x}||_2^2] &= \text{tr} \left((\mathbf{I} - \mathbf{B}^\top \mathbf{A}) \mathbb{E}[\mathbf{x} \mathbf{x}^\top] (\mathbf{I} - \mathbf{B}^\top \mathbf{A})^\top \right) \\ &= \frac{K \sigma_x^2}{n} ||\mathbf{I} - \mathbf{B}^\top \mathbf{A}||_F^2 \end{aligned}$$

where \mathbb{E} denotes the expectation value, we assumed that all support sets are chosen with equal probability and that the components of \mathbf{x}_I are random variables with $\mathbb{E}[x_i x_j | \{i, j\} \subset I] = \sigma_x^2 \delta_{ij}$. This gives us the minimizer

$$\mathbf{B} = (\mathbf{A}^+)^T$$

Note that we did not make any assumptions on the distribution of \mathbf{x}_I , so this argument holds also for non-Gaussian random signals, e.g. binary (± 1) signals. This beamformer is different from (6) since in general $\mathbf{b}_i^\top \mathbf{a}_i \neq 1$.

3. BEAMFORMER FOR NOISY RECOVERY

We here analyze beamformers for the noisy worst and average case scenario and propose new methods of calculating them.

3.1. Worst case scenario

When the measurements are corrupted with random noise

$$\mathbf{y} = \mathbf{A} \mathbf{x} + \mathbf{n}, \quad (7)$$

the worst case condition (3) for the beamformer to successfully recover the element $i \in I$ of maximum modulus in the first iteration becomes modified to [15]

$$|\mathbf{b}_i^\top \mathbf{a}_i| > \sum_{j \in I \setminus \{i\}} |\mathbf{b}_i^\top \mathbf{a}_j| + \max_{k \notin I} \sum_{j \in I} |\mathbf{b}_k^\top \mathbf{a}_j| + \frac{2}{|x_i|} |\mathbf{b}_i^\top \mathbf{n}|$$

When the noise is unbounded (e.g. Gaussian) we can never recover \mathbf{x} with probability one. We therefore instead try to maximize the probability of recovering $i \in I$.

Theorem 1. Assume that the additive noise is zero-mean Gaussian distributed, $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, $\mathbf{b}_i^\top \mathbf{a}_i = 1$ for all $i = 1, 2, \dots, n$ and

$$c = 1 - \mu_1(\mathbf{A}, \mathbf{B}, K) - \mu_1(\mathbf{A}, \mathbf{B}, K - 1) > 0$$

where $\mu_1(\mathbf{A}, \mathbf{B}, K)$ is the *cross cumulative coherence* [10]

$$\mu_1(\mathbf{A}, \mathbf{B}, K) = \max_{i, |J| \leq K, i \notin J} \sum_{j \in J} |\mathbf{b}_i^\top \mathbf{a}_j|$$

Then the probability \tilde{P} that OMPb recovers the component x_i of \mathbf{x} with maximum modulus in the first iteration obeys

$$\tilde{P} \geq 1 - 2Q\left(\frac{c|x_i|}{\sqrt{\mathbf{b}_i^\top \mathbf{C} \mathbf{b}_i}}\right)$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt$ is the tail probability of the normal distribution.

Proof: A sufficient condition for OMPb to recover $i \in I$ is

$$\frac{2}{|x_i|} |\mathbf{b}_i^\top \mathbf{n}| < 1 - \sum_{j \in I \setminus \{i\}} |\mathbf{b}_i^\top \mathbf{a}_j| - \max_{l \notin I} \sum_{j \in I} |\mathbf{b}_l^\top \mathbf{a}_j| \quad (8)$$

Since

$$\begin{aligned} \sum_{j \in I \setminus \{i\}} |\mathbf{b}_i^\top \mathbf{a}_j| &\leq \mu_1(\mathbf{A}, \mathbf{B}, K - 1) \\ \max_{l \notin I} \sum_{j \in I} |\mathbf{b}_l^\top \mathbf{a}_j| &\leq \mu_1(\mathbf{A}, \mathbf{B}, K) \end{aligned}$$

we find that (8) holds provided that $|\mathbf{b}_i^\top \mathbf{n}| < c|x_i|/2$. Using this we find that

$$\tilde{P} \geq P\left(|\mathbf{b}_i^\top \mathbf{n}| < \frac{c|x_i|}{2}\right) \quad (9)$$

When $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, then $z_i = \mathbf{b}_i^\top \mathbf{n}$ is Gaussian $\mathcal{N}(0, \sigma_i^2)$, where $\sigma_i^2 = \mathbf{b}_i^\top \mathbf{C} \mathbf{b}_i$. Using that $P(|z_i| < \epsilon) = 1 - 2Q(2\epsilon/\sigma_i)$ we arrive at the result. \square

Note that (9) also holds for non-gaussian noise distributions. Theorem 1 states that the probability of recovering the component of maximum modulus increases with increasing Signal-to-Noise Ratio (SNR), as can be expected. One way to maximize \tilde{P} is to maximize the argument of the Q -function. The argument is, however, a non-convex function of \mathbf{B} and is therefore difficult to maximize. A more accessible approach is set

$$\begin{aligned} \mathbf{b}_i &= \arg \min_{\mathbf{b}} \left(\max_{|J|=K, i \notin J} \sum_{j \in J} |\mathbf{b}^\top \mathbf{a}_j| + \lambda \mathbf{b}^\top \mathbf{C} \mathbf{b} \right) \\ \text{s.t. } \mathbf{b}^\top \mathbf{a}_i &= 1 \end{aligned} \quad (10)$$

where $\lambda \geq 0$ is a design parameter.

3.2. Average Case Scenario

For (7) with $\mathbf{x}_I \sim \mathcal{N}(\mathbf{0}, \sigma_x^2 \mathbf{I})$ and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ we can use a modification of Lemma 1 for the non-white case to find the probability. By rewriting

$$\mathbf{b}_i^\top (\mathbf{A} \mathbf{x} + \mathbf{n}) = (\mathbf{c}_i^\top, \mathbf{b}_i^\top) \begin{pmatrix} \mathbf{x}_I \\ \mathbf{n} \end{pmatrix},$$

we find the probability (11) at the top of the next page.

To minimize the influence of noise, we minimize the length of the vectors \mathbf{b}_j while maintaining the properties of the vectors \mathbf{c}_i . One approach is, as before, to penalize the length of \mathbf{b}_i by setting

$$\begin{aligned} \mathbf{b}_i &= \arg \min_{\mathbf{b}} \left(\max_{|J| \leq K, i \notin J} \sum_{j \in J} |\mathbf{b}^\top \mathbf{a}_j|^2 + \lambda \mathbf{b}^\top \mathbf{C} \mathbf{b} \right) \\ \text{s.t. } \mathbf{b}^\top \mathbf{a}_i &= 1, \end{aligned} \quad (12)$$

where λ is a design parameter. Again setting $K = n - 1$, we obtain the beamformer

$$\mathbf{b}_i = \frac{(\mathbf{A} \mathbf{A}^\top + \lambda \mathbf{C})^{-1} \mathbf{a}_i}{\mathbf{a}_i^\top (\mathbf{A} \mathbf{A}^\top + \lambda \mathbf{C})^{-1} \mathbf{a}_i}. \quad (13)$$

We see that both (10) and (13) converge to \mathbf{A} when $\lambda \rightarrow \infty$.

4. SIMULATIONS

For the simulational evaluation, we compare our proposed beamformers where $\mathbf{B}_{pinv} = (\mathbf{A}^+)^T$ and $\mathbf{B}_{spinv} = (\mathbf{A}^+)^T \mathbf{D}$, with regular OMP (i.e., $\mathbf{B} = \mathbf{A}$) and the Equiangular Tight Frame (ETF beamformer) (referred to as $\mathbf{B}_{ETF} = \text{ETF}(\mathbf{A})$) presented in [10]. The noisy setting is used to evaluate the robustness of the algorithms under noise. We compare these algorithms in noisy and noise-free environments for fixed m and n , where we vary K . We omit the worst case beamformers since they are time consuming to calculate and perform worse than \mathbf{B}_{pinv} and \mathbf{B}_{spinv} in the scenario considered. In all cases we chose the non-zero components in \mathbf{x} from a Gaussian independent identically distributed (i.i.d.) source. For the noisy signals, we characterize the level of measurement noise by the signal-to-measurement-noise (SMNR) ratio as

$$\text{SMNR} = \frac{\mathbb{E} \{ \|\mathbf{x}\|_2^2 \}}{\mathbb{E} \{ \|\mathbf{n}\|_2^2 \}}. \quad (14)$$

To compare the different algorithms, we define two different performance measures. The first one is the signal-to-reconstruction-error-ratio (SRER) defined as

$$\text{SRER} = \frac{\mathbb{E} \{ \|\mathbf{x}\|_2^2 \}}{\mathbb{E} \{ \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 \}}. \quad (15)$$

The SRER performance measure is natural for the noisy environment. For clean measurements, the SRER is not so convenient since for perfect recovery it reaches machine precision. Instead, we define another performance measure which provides a direct measure of estimating the underlying support set. This measure is a distortion defined by $d(I, \hat{I}) = 1 - (|I \cap \hat{I}|/|I|)$ [16] and we have recently used it in [17]. Considering a large number of realizations (signal vectors), we can compute the average of $d(I, \hat{I})$. We define the average support-set cardinality error (ASCE) as follows

$$\text{ASCE} = \mathbb{E} \{ d(I, \hat{I}) \} = 1 - \mathbb{E} \left\{ \frac{|I \cap \hat{I}|}{|I|} \right\}. \quad (16)$$

$$P(|\mathbf{b}_i^\top (\mathbf{A}\mathbf{x} + \mathbf{n})| > |\mathbf{b}_j^\top (\mathbf{A}\mathbf{x} + \mathbf{n})|) = \frac{1}{\pi} \arccos \left(\frac{\sigma_x^2 (\|\mathbf{c}_j\|_2^2 - \|\mathbf{c}_i\|_2^2) + (\mathbf{b}_j^\top \mathbf{C} \mathbf{b}_j - \mathbf{b}_i^\top \mathbf{C} \mathbf{b}_i)}{\sqrt{(\sigma_x^2 \|\mathbf{c}_j\|_2^2 + \sigma_x^2 \|\mathbf{c}_i\|_2^2 + \mathbf{b}_i^\top \mathbf{C} \mathbf{b}_i + \mathbf{b}_j^\top \mathbf{C} \mathbf{b}_j)^2 - 4(\sigma_x^2 \mathbf{c}_i^\top \mathbf{c}_j + \mathbf{b}_i^\top \mathbf{C} \mathbf{b}_j)^2}} \right) \quad (11)$$

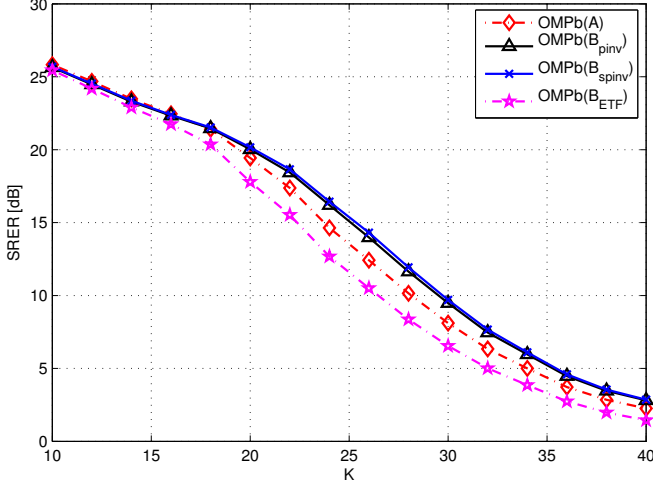


Fig. 1. Figure showing a noisy measurement case where SMNR = 20dB where the performance is measured in SRER.

Note that the ASCE has the range $[0, 1]$ and our objective is to achieve a lower ASCE. Along with SRER, the ASCE is used as the second performance evaluation measure because the principle objective of GP algorithms is to estimate the underlying support set.

Next we describe the simulation setup. Fixing the signal vector size n and the number of measurements m we do the following:

1. Choose a K .
2. Randomly generate an $m \times n$ sensing matrix \mathbf{A} where the components are drawn independently from an i.i.d. Gaussian source (i.e. $a_{i,j} \sim \mathcal{N}(0, 1)$) and then scale the columns of \mathbf{A} to unit-norm.
3. Generate our beamformers as $\mathbf{B}_{pinv} = (\mathbf{A}^+)^T$, $\mathbf{B}_{spinv} = (\mathbf{A}^+)^T \mathbf{D}$, where \mathbf{D} is a diagonal matrix with entries $D_{ii} = 1/(\mathbf{A}^+ \mathbf{A})_{ii}$ and $\mathbf{B}_{ETF} = \text{ETF}(\mathbf{A})$.
4. Generate a support-set I of cardinality K . The support-set is uniformly chosen from $\{1, 2, \dots, n\}$.
5. Randomly generate a sparse signal vector \mathbf{x} with non-zero components chosen independently from a Gaussian source.
6. Compute the measurement vector $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$, where \mathbf{n} is standard i.i.d. Gaussian noise.
7. Apply the OMPb algorithms on the data \mathbf{y} and the matrices \mathbf{A} , \mathbf{B}_{pinv} , \mathbf{B}_{spinv} and \mathbf{B}_{ETF} .

In the simulation procedure above, q number of different sensing matrices \mathbf{A} are created. For each sensing matrix, p data vectors are generated. In total, we will average over $q \cdot p$ data to evaluate the performance.

4.1. Parameters and Simulation Set-up

For the plots presented in this paper, we have chosen: $n = 500$, $m = 100$. We have chosen the number of matrices \mathbf{A} to $q = 400$ and the number of data-sets \mathbf{x} to 400 (i.e. $p = 400$), giving a total

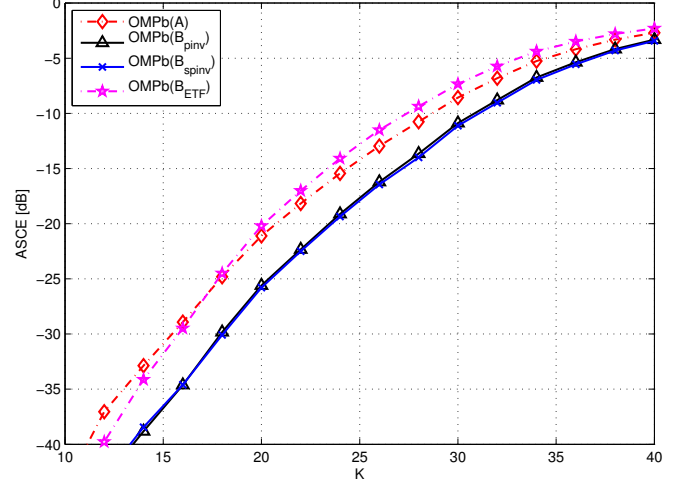


Fig. 2. Figure showing a clean measurement case where the performance measure is the ASCE.

number of $q \cdot p = 16 \cdot 10^4$ data in each data-point for statistics. We used 100 iterations in the computation of the ETF beamformer.

4.2. Analysis of the Simulation Results

Figure 1 shows that OMPb with \mathbf{B}_{pinv} and \mathbf{B}_{spinv} perform better than regular OMP for $K \geq 17$. We found that OMPb with \mathbf{B}_{ETF} performed worse than regular OMP, this seems to be because in this scenario $\frac{n}{m} = 5$ is large. When $\frac{n}{m}$ is smaller, the performance of ETF is greatly improved [10]. The simulation indicates that \mathbf{B}_{pinv} and \mathbf{B}_{spinv} seems well also in the noisy setting.

When varying K in the noise-free scenario we find that \mathbf{B}_{pinv} and \mathbf{B}_{spinv} give similar values of the ASCE for all values of K . For small K ($10 \leq K \leq 25$) the improvement is about 5 dB over regular OMP as can be seen in Figure 2. Again the performance of ETF seems to be due to the choice $\frac{n}{m}$.

5. CONCLUSION

In this paper we extend the work in [10] of using beamformers to improve the performance of greedy pursuit algorithms. By analyzing the worst and average case scenario we found beamformers that can be computed using convex optimization methods. For the average case we used Lemma 1, which gives the probability of choosing one index over another when \mathbf{x}_I is Gaussian, to motivate a beamformer which minimizes the sum of squared inner products. For the largest possible K we obtained the (scaled) pseudoinverse as a beamformer. Treating the noisy case resulted in similar beamformers. We treat the noisy case using similar methods and propose beamformers similar to the noise-free case. Simulations confirmed that OMPb with the beamformers \mathbf{B}_{pinv} and \mathbf{B}_{spinv} perform better than regular OMP.

6. REFERENCES

- [1] D.L. Donoho, “Compressed sensing,” *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, april 2006.
- [2] E.J. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, feb. 2006.
- [3] Michael Lustig, David Donoho, and John M. Pauly, “Sparse mri: The application of compressed sensing for rapid mr imaging,” *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.
- [4] Urs Gamper, Peter Boesiger, and Sebastian Kozerke, “Compressed sensing in dynamic mri,” *Magnetic Resonance in Medicine*, vol. 59, no. 2, pp. 365–373, 2008.
- [5] Matthew A. Herman and Thomas Strohmer, “High-resolution radar via compressed sensing,” *IEEE Trans. Signal Processing*.
- [6] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, Jan. 2001.
- [7] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Trans. Inf. Theor.*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.
- [8] Wei Dai and Olgica Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction,” *IEEE Trans. Inf. Theor.*, vol. 55, no. 5, pp. 2230–2249, May 2009.
- [9] Deanna Needell and Joel A. Tropp, “Cosamp: iterative signal recovery from incomplete and inaccurate samples,” *Commun. ACM*, vol. 53, no. 12, pp. 93–100, Dec. 2010.
- [10] K. Schnass and P. Vandergheynst, “Dictionary preconditioning for greedy algorithms,” *Trans. Sig. Proc.*, vol. 56, no. 5, pp. 1994–2002, May 2008.
- [11] Michael Grant, Stephen Boyd, and Yinyu Ye, “Disciplined convex programming,” in *Global Optimization: From Theory to Implementation, Nonconvex Optimization and Its Application Series*. 2006, pp. 155–210, Springer.
- [12] Włodzimierz Ogryczak and Arie Tamir, “Minimizing the sum of the k largest functions in linear time,” *Inf. Process. Lett.*, vol. 85, no. 3, pp. 117–122, Feb. 2003.
- [13] Inc. CVX Research, “CVX: Matlab software for disciplined convex programming, version 2.0 beta,” .
- [14] P. Stoica and R.L. Moses, *Spectral analysis of signals*, Pearson Prentice Hall, 2005.
- [15] An-Min Huang, Qun Wan, and Wan-Lin Yang, “Dictionary preconditioning for orthogonal matching pursuit in the presence of noise,” in *Communications, Circuits and Systems, 2009. ICCCAS 2009. International Conference on*, july 2009, pp. 419–422.
- [16] G. Reeves and M. Gastpar, “A note on optimal support recovery in compressed sensing,” *Proc. Asilomar Conf. Signals, Sys., and Comp.*, pp. 1576–1580, nov. 2009.
- [17] S. Chatterjee, D. Sundman, M. Vehkaperä, and M. Skoglund, “Projection-based and look ahead strategies for atom selection,” *IEEE Trans. Signal Processing*, vol. 60, no. 2, pp. 634–647, feb 2012.