Annihilating filter-based decoding in the compressed sensing framework

Ali Hormati and Martin Vetterli

School of Computer and Communication Science, Ecole Polytechnique Federale de Lausanne,
1015 Lausanne, Switzerland

ABSTRACT

Recent results in compressed sensing or compressive sampling suggest that a relatively small set of measurements taken as the inner product with universal random measurement vectors can well represent a source that is sparse in some fixed basis. By adapting a deterministic, non-universal and structured sensing device, this paper presents results on using the annihilating filter to decode the information taken in this new compressed sensing environment. The information is the minimum amount of nonadaptive knowledge that makes it possible to go back to the original object. We will show that for a k-sparse signal of dimension n, the proposed decoder needs $2k$ measurements and its complexity is of $O(k^2)$ whereas for the decoding based on the ℓ_1 minimization, the number of measurements needs to be of $O(k \log(n))$ and the complexity is of $O(n^3)$. In the case of noisy measurements, we first denoise the signal using an iterative algorithm that finds the closest rank k and Toeplitz matrix to the measurements matrix (in Frobenius norm) before applying the annihilating filter method. Furthermore, for a k-sparse vector with known equal coefficients, we propose an algebraic decoder which needs only k measurements for the signal reconstruction. Finally, we provide simulation results that demonstrate the performance of our algorithm.

Keywords: Compressed sensing, Sparse signals, Annihilating filter, Vandermonde matrix, Yule-Walker equation, Basis Pursuit, Orthogonal Matching Pursuit

1. INTRODUCTION

The tradition in the signal processing community has been to sample the data at high rates and try to compress the data acquired soon after it has been captured. This method is wasteful of sensing resources when the signal is compressible. It is interesting to merge the sampling and compression parts in one phase, which is the idea of Compressed Sampling or Sensing (CS). This means that one tries to measure the minimum amount of information necessary to reconstruct the object so further compression would prohibit perfect reconstruction. The question that naturally arises is what are the best measurements to take? Obviously, these measurements should be nonadaptive in the sense that they should not depend on the specific object they are dealing with (except that the object is compressible) and the measurements should convey enough information for a practical recovery mechanism of the original signal.

Let us set up the problem. Consider a discrete, finite dimensional data vector $x \in \mathbb{R}^n$ such that x can be well approximated by $k \ll n$ linear combination of vectors of a basis $\Psi = [\psi_1, \psi_2, \cdots, \psi_n]$. In this case, it is possible to write $x = \Psi\theta$ where θ is a column vector of length n with k dominant coefficients. The theory of compressed sensing as developed in Refs. 1 and 2 shows that x can be recovered from $m = O(k \log(n))$ projections given by $y = \Phi x$, where Φ is an $m \times n$ matrix incoherent with the columns of Ψ. Incoherency between Φ and Ψ means that the rows of Φ do not have a sparse representation in terms of the columns of Ψ, otherwise, the product $\Phi\Psi$ will be a sparse matrix and is not appropriate to measure another sparse vector θ.

Commonly used measurement matrices for CS are i.i.d Gaussian or Bernoulli (random ±1) random matrices which are incoherent with nearly all bases Ψ. Hence, when using a random matrix, CS is called universal in the sense that the encoder can measure almost any sparse signal with the same measurement matrix no matter in which basis the signal is sparse. However the knowledge of the random measurement matrix and the sparsity basis are necessary to recover the original signal x from the measurement vector y.

The recovery phase can be implemented as an optimization program that searches for the vector θ with the smallest ℓ_0 norm that conforms to the measurements in vector y. The ℓ_0 optimization problem succeeds with high probability with $m = k + 1$ measurements (Not all the signals can be recovered perfectly; those that can not be recovered have probability measure zero), see Ref. 3. However, the ℓ_0 optimization is NP-complete as described in Ref. 4. The signal recovery part of CS is achieved by solving an ℓ_1 norm minimization problem on the sparse vector θ: $min\|\theta\|_1$ subject to $\Phi\Psi\theta = y$. The practical aspect of CS is that by taking $m = O(k \log(n))$ measurements with an incoherent basis, with high probability, the ℓ_1 norm optimization will solve the NP-complete ℓ_0 equivalent. This optimization problem, also known as Basis Pursuit (BP) is significantly more approachable and can be solved with traditional linear programming techniques whose computational complexity is of $O(n^3)$, see Ref. 5. At the expense of slightly more measurements and less complexity, iterative greedy algorithms under the name of Matching Pursuit (MP) and Orthogonal Matching Pursuit (OMP) have also been developed for the recovery as described in Refs. 6 and 7. MP and OMP are greedy algorithms that iteratively reconstruct the original signal.

In this paper we propose a new sensing and reconstruction strategy in which by using a deterministic, non-universal and structured sensing device, it is possible to recover a signal which is k-sparse in an orthonormal or biorthogonal basis with only $2k$ measurements and $O(k^2)$ complexity. Considering the number of measurements and the complexity of BP, which are $O(k \log(n))$ and $O(n^3)$ respectively, our algorithm is thus interesting. In our new scheme, we exchange universality for fewer measurements and less complexity in the sense that the encoder should also be aware of the underlying sparsity basis. The measurement matrix Φ has a deterministic structure and is constructed by using an arbitrary Vandermonde matrix and the sparsity basis Ψ. On the decoder side, we propose the annihilating filter as an algebraic decoder to recover the original signal x. In the noisy case, we propose an iterative algorithm to denoise the signal before applying the annihilating filter. Moreover, we develop a new algebraic decoding algorithm that needs just k measurements in the special case when the sparsity vector θ has k equal coefficients.

This paper is organized as follows: in Section 2 we see the annihilating filter structure and its properties. In Section 3, we present our new encoding and decoding algorithms in both noiseless and noisy scenarios. In Section 4, the decoding structure for the sparse signals with equal coefficients is introduced. Finally, Section 5 reports the simulation results.

2. ANNIHILATING FILTERS

In this section we give an overview of the annihilating filter method as the tool that will be used in the rest of the paper to decode the CS information. This technique was mostly developed in the area of parametric spectral estimation where the problem is to estimate the parameters of a linear combination of complex exponentials from a set of measurements, see Ref. 8. This idea was also employed by Vetterli et al. to devise a sampling method for signals with finite rate of innovation, see Ref. 11. Consider the signal $S[n]$ which consists of the sum of k exponentials

$$S[n] = \sum_{i=0}^{k-1} c_i u_i^n, \tag{1}$$

in which $c_i \in \mathbb{R}$ and $u_i \in \mathbb{C}$. A filter $A[n]$ is called the annihilating filter of the signal $S[n]$ when $(A * S)[n] = 0 \quad \forall n \in \mathbb{N}$. It is easy to verify that the filter $H(z) = 1 - uz^{-1}$ annihilates the exponential signal $x[n] = u^n$ so we have the following proposition

Proposition 1. The signal $S[n]$ is annihilated by the filter

$$A(z) = \prod_{i=0}^{k-1}(1 - u_i z^{-1}) = \sum_{\ell=0}^{k} A[\ell] z^{-\ell}. \tag{2}$$

Proof. Note that

$$\begin{aligned}(A * S)[n] &= \sum_{\ell=0}^{k} A[\ell]S[n-\ell] && (3)\\ &= \sum_{\ell=0}^{k}\sum_{i=0}^{k-1} c_i A[\ell]u_i^{n-\ell} && (4)\\ &= \sum_{i=0}^{k-1} c_i \underbrace{\left(\sum_{\ell=0}^{k} A[\ell]u_i^{-\ell}\right)}_{A(u_i)} u_i^n = 0. && (5)\end{aligned}$$

The annihilating filter method consists of finding the values of c_i and u_i in $S[n]$ and is composed of three parts: First we need to find the annihilating filter that involves solving a linear system of equations; second, we need to find the roots of the z-transform of the filter which is a nonlinear operation; third, we must solve another linear system of equations to find the weights.

1. Finding the annihilating filter.
   The filter coefficients $A[\ell]$ should satisfy

$$\sum_{\ell=0}^{k} A[\ell]S[n-\ell] = 0 \ \forall n \in \mathbb{Z} \tag{6}$$

   In matrix form, the system in equation (6) is equivalent to

$$\begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ S[k] & S[k-1] & \cdots & S[0] \\ S[k+1] & S[k] & \cdots & S[1] \\ \vdots & \vdots & \ddots & \vdots \\ S[2k-1] & S[2k-2] & \cdots & S[k-1] \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \cdot \begin{pmatrix} A[0] \\ A[1] \\ \vdots \\ A[k] \end{pmatrix} = 0. \tag{7}$$

   If at least $2k$ values of $S[n]$ are available, the system of equations (7) will admit a solution when Rank(S) $= k$, where S is the matrix of the signal values $S[n]$ in equation (7).

2. Finding the u_i.
   Once the filter coefficients $A[n]$ are found, the values u_i are the roots of the annihilating filter $A(z)$.

3. Finding the c_i.
   To determine the weights c_i, it suffices to take k equations in (1) and solve the system for c_i. In matrix form, equations will form a Vandermonde system

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ u_0 & u_1 & \cdots & u_{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_0^{k-1} & u_1^{k-1} & \cdots & u_{k-1}^{k-1} \end{bmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{k-1} \end{pmatrix} = \begin{pmatrix} S[0] \\ S[1] \\ \vdots \\ S[k-1] \end{pmatrix}, \tag{8}$$

   and has a unique solution when $u_p \neq u_q, \forall p \neq q$.

3. THE DETERMINISTIC NON-UNIVERSAL CS FRAMEWORK

One of the properties introduced by employing random projections in the original CS structure is its universality. This allows the same encoding strategy irrespective of the sparsity basis Ψ in which the input signal is sparse (assuming of course that Ψ is known at the decoder). The price for this universality is that the recovery algorithms have high complexity with respect to the signal dimension and that the number of measurements should be larger than the degrees of freedom of the class of sparse signals for a successful reconstruction even in the noiseless case.

In the new CS framework developed in this paper, we use a tradeoff between universality on one side and number of measurements and complexity on the other side. The non-universality in the new structure comes from the fact that the encoder should have the knowledge of the sparsity basis Ψ to construct the measurement matrix Φ and take the measurements. We have the following proposition

Proposition 2. Let $x \in \mathbb{R}^n$ be a discrete time signal. Assume x is k-sparse in an orthonormal or biorthogonal basis $\Psi_{n\times n}$ and build the matrix $\tilde{\Psi}_{n\times n}$ such that $\tilde{\Psi} * \Psi = I_{n\times n}$. The rows of $\tilde{\Psi}$ are the dual basis vectors of the space which represents the signal. Take any Vandermonde matrix $\Gamma_{2k\times n}$ of the form

$$\Gamma_{2k\times n} = \begin{pmatrix} z_1^p & z_2^p & \cdots & z_n^p \\ z_1^{p+1} & z_2^{p+1} & \cdots & z_n^{p+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{p+2k-1} & z_2^{p+2k-1} & \cdots & z_n^{p+2k-1} \end{pmatrix}, \tag{9}$$

where z_i, $i = 1, \ldots, n$ are distinct, nonzero complex numbers and p is an arbitrary real number. Construct the measurement matrix $\Phi_{2k\times n} = \Gamma * \tilde{\Psi}$. Take the measurements $y = \Phi * x$. Then the measurement vector y is the unique representation of the data vector x and it is possible to perfectly reconstruct x with $O(k^2)$ operations. Moreover, the number of measurements can not be relaxed without sacrificing perfect reconstruction.

From Proposition 2, the following corollary is immediate

Corollary 1. Proposition 2 is applicable for the space of continuous-time signals x which are sparse in some biorthogonal or orthonormal continuous-time basis with a countable number of basis functions. In this case, the rows of the measurement matrix $\Phi_{2k\times .}$ will be continuous time signals.

In the two following subsections, we will see the encoding and decoding structure of our CS structure in detail.

3.1 Sensing Mechanism

On the sensing side of the system, the encoder chooses any Vandermonde matrix Γ of the form given in equation (9) with at least $2k$ rows. An appropriate choice for the elements z_i in equation (9) is the roots of unity $e^{j\frac{2\pi i}{n}}$. In this case, the rows of the Vandermonde matrix will be $2k$ consecutive rows of a full Fourier matrix of size $n \times n$. Since the signal x is k-sparse in the basis Ψ, it is possible to represent it as $x = \Psi\theta$ where θ is a vector of length n with only k nonzero elements. The encoder builds the fixed measurement matrix $\Phi = \Gamma * \tilde{\Psi}$. The vector of measurements, y, will be given by $y = \Phi * x = \Gamma * \theta$ so the Vandermonde matrix Γ is multiplied directly with the k-sparse vector θ.

If the number of measurements in the encoding process is greater or equal to $2k$, where k is the sparsity of the input signal, there is a one to one correspondence between the measurements and the k-sparse signals in the sense that two k-sparse signals x_1 and x_2 which give the same measurement vector y should be identical. This is the direct consequence of the fact that in a Vandermonde matrix of size $2k \times n$ as in equation (9) with distinct z_i, any $2k$ columns are linearly independent. Therefore, the two k-sparse data vectors x_1 and x_2 producing the same measurement vector y of size $2k$ must be identical; their difference $x = x_1 - x_2$ has at most $2k$ nonzero elements and will map to zero by $2k$ linearly independent columns of the Vandermonde matrix Γ so the difference vector x should be an all zero vector and $x_1 = x_2$. This argument points out that in the new CS mechanism, no information about the sparse signals is lost on the encoder side.

We remark that the original k-sparse signal x has $2k$ degrees of freedom that corresponds to the k positions and k weights of the non-zero elements of the k-sparse vector θ. Therefore, there is no CS algorithm of any kind that is able to find its way back for ***all*** k-sparse vectors x with less than $2k$ measurements; It is easy to show that for less than $2k$ measurements, there are two k-sparse vectors x_1 and x_2 that map to the same measurement vector y. This suggests that our CS structure is optimum in the number of measurements.

3.2 Recovery Mechanism

3.2.1 Noiseless recovery

The reconstruction algorithm uses standard linear algebra techniques based on the annihilating filter structure described in Section 2. Using the same model as in Proposition 2, the vector of measurements, $y = \Phi * x = \Gamma * \theta$, can be written as

$$y[n] = \sum_{i=1}^{k} c_i z_{n_i}^{p+n-1} \;\; n = 1\,, \ldots,\, 2k, \tag{10}$$

where c_i and n_i, $i = 1, \ldots, k$ represent the unknown nonzero coefficients and their positions in the vector θ. An annihilating filter of length $k+1$ is first identified as a polynomial that zeros out the measurements, $y*h = 0$. The zeros of the annihilator gives the values for the unknowns n_i which indicate the index of the basis vectors of Ψ that construct the sparse signal x. In this process, the annihilating filter equation is a Yule-Walker system which is solvable in $O(k^2)$ operations, see Ref. 9. To find the locations of the nonzero coefficients, the annihilating filter should be factorized. Recently it has been shown that factorizing a k degree polynomial can be done in $O(k^2)$ operations as described in Ref. 10. Finally the weights of the nonzero coefficients of the signal are obtained by solving a Vandermonde system in $O(k^2)$ operations, see Ref. 9, so the whole reconstruction complexity is $O(k^2)$. Compared to the Basis Pursuit which has complexity $O(n^3)$, our algorithm is computationally more efficient and requires less measurements in the general case when $k \ll n$.

3.2.2 Signal recovery with noisy measurements

Noise and model mismatch are two major factors that makes the solution presented in Section 3.2.1 only ideal. Quantization errors can also be an important source of noise to the measurements and model mismatch, when the input signal x is not exact sparse in the basis Ψ, is omnipresent in practice. To achieve robustness to these sources of non-idealities, there is no way but to increase the number of samples to more than $2k$.

Researchers in spectral analysis have tried to solve the related problem of finding sinusoids in noise for a number of years, see Ref. 8. One of the difficulties is that there is not yet an agreed optimal algorithm for this problem although there have been numerous evaluations of different methods as described in Ref. 12. Our choice for tackling the noise is based on what was presented in Ref. 13. The basic idea is first trying to denoise the measurements before applying the annihilating filter method to estimate the unknown parameters. Assume that the number of measurements is increased from the minimum $2k$ to M. From the noisy measurement vector y, we build the matrix A of size $(M - L) \times (L + 1)$ as

$$A_{(M-L)\times(L+1)} = \begin{pmatrix} y_{L+1} & y_L & \cdots & y_1 \\ \vdots & \vdots & \ddots & \vdots \\ y_M & y_{M-1} & \cdots & y_{M-L} \end{pmatrix}, \tag{11}$$

in which $L \geq k$ and $M - L \geq k$. In the noiseless scenario, the matrix A should be a Toeplitz matrix of rank k but due to noise in the measurements, neither of these properties are satisfied. The signal enhancement procedure is a convergent iterative algorithm that finds the closest Toeplitz matrix of rank k to matrix A where the distance between two matrices is measured by the Frobenius norm, see Ref. 14. For any given matrix A, the closest Toeplitz matrix to A is the one given by averaging the elements of matrix A along its diagonals. On the other hand, the closest rank k matrix to A is given by taking the SVD of A and keeping just its k dominant singular values. To find the closest Toeplitz matrix of rank k to matrix A, the signal enhancement algorithm iteratively maps the matrix A on the two spaces of (rank k) and (Toeplitz) matrices. Since the mappings are closed, the iterations converge to a matrix in the intersection of the two spaces which is closest to initial matrix A in the Frobenius norm, as shown in Ref. 14. The iterations stop when the ratio of the singular values $\frac{\sigma_{k+1}}{\sigma_k}$ is smaller than a pre-requisite factor. After this denoising step, we use the algorithm in Subsection 3.2.1 to get an estimate of the unknown parameters of the signal. Our observations show that as the matrix A is closer to a square matrix, i.e. $L \simeq [\frac{M}{2}]$, the enhanced matrix will be closer to the noiseless one.

Although the number of iterations necessary for convergent is typically small (less than 10), the computational cost of the algorithm is high compared to the noiseless case because in each iteration, it requires performing the SVD of a square matrix of size approximately $\frac{M}{2}$ which has complexity of $O(M^3)$, see Ref. 9. Since M is proportional to the sparsity level k with the oversampling factor c, $M = c \cdot k \ll n$, this complexity scales with k and not with the signal dimension n. Despite the computational load, the simulations done in Ref. 13 shows that the denoising algorithm exhibits a quasi-optimal behavior close to the Cramer-Rao bound down to noise levels of the order of 5 dB (depending on the number of measurements).

4. ALGEBRAIC RECOVERY OF SPARSE SIGNALS WITH EQUAL COEFFICIENTS

The class of sparse signals considered in the previous sections has $2k$ degrees of freedom which represents k basis vectors and their respective weights that constructs the input signal x. It is reasonable that to be able to reconstruct all the k-sparse signals perfectly, we need at least $2k$ measurements. With respect to this, it is natural to think that if there is some knowledge about the position or the values of the coefficients, the minimum number of measurements needed should be less than $2k$.

In this section we show the result of reducing the number of measurements from $2k$ to k whenever the weights of different basis vectors constructing the signal are equal and known a priori. The direct annihilating filter method discussed in Section 2 still needs $2k$ measurements and can not use the knowledge about the weights to reduce the number of measurements. The vector θ is now a set of equal weight discrete diracs with unknown positions as if the information is just on the position of the diracs and not on their weights.

Without loss of generality, assume that the sparsity inducing basis Ψ is equal to the identity matrix. We have the following proposition

Proposition 3. Let $x \in \mathbb{R}^n$ be a discrete time signal which consists of k discrete diracs with equal known weights (assume it to be equal to 1)

$$x[m] = \sum_{q=1}^{k} \delta\left[m - m_q\right] \quad m = 1, \cdots, n\,. \tag{12}$$

Let the measurement matrix be

$$\Phi_{k\times n} = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^k & x_2^k & \cdots & x_n^k \end{pmatrix}, \tag{13}$$

where x_i, $i = 1, \ldots, n$ are distinct nonzero complex numbers. Make the measurements as $y = \Phi * x$. Then the positions of the diracs are directly connected to the roots of the polynomial

$$H(x) = x^k - \alpha_1 x^{k-1} + \alpha_2 x^{k-2} - \cdots + (-1)^k \alpha_k, \tag{14}$$

in which $\alpha_0 = 1$ and α_i, $i = 1, 2, \ldots, k$ can be computed by the recursive formula

$$\alpha_{i+1} = \frac{\sum_{m=1}^{i+1} (-1)^{m+1} y[m] \alpha_{i-m+1}}{i+1}. \tag{15}$$

Proof. The measurements vector y can be written as

$$y[m] = \sum_{q=1}^{k} x_{m_q}^m, \; m = 1, \cdots, k \, . \tag{16}$$

Write $H(x)$ as

$$H(x) = \prod_{q=1}^{k} (x - x_{m_q}), \tag{17}$$

in which m_q, $q = 1, \ldots, k$ are the unknown positions of the nonzero elements of the input signal x. Define the set $S(j)$ as the set of all possible j tuples of $\{m_1, m_2, \ldots, m_k\}$. Expanding the product in equation (17), the coefficients α_j in (14) can be written as

$$\alpha_j = \sum_{S(j)} x_{n_1} x_{n_2} \cdots x_{n_j} \; , \tag{18}$$

in which $\{n_1, n_2, \ldots, n_j\}$ takes values from the elements of the set $S(j)$. Let $1 \leq i \leq k$ and compute the product $\alpha_i * y[1]$. There are two groups of terms that appears in the result; In the first group, all the elements in each term has power 1 and in the second group, one of the elements in the terms has power equal to 2. By rearranging the terms

$$\alpha_i * y[1] = (i+1) \overbrace{\sum_{S(i+1)} x_{n_1} x_{n_2} \cdots x_{n_{i+1}}}^{\alpha_{i+1}} + x_{m_1}^2 \sum_{S(i-1,m_1)} x_{n_1} \cdots x_{n_{i-1}} + \ldots + x_{m_k}^2 \sum_{S(i-1,m_k)} x_{n_1} \cdots x_{n_{i-1}}, \tag{19}$$

in which the notation $S(j, m_q)$ denotes all possible j tuples of $\{m_1, m_2, \ldots, m_k\}$ without m_q. To complete the proof, it is enough to show that

$$x_{m_1}^2 \sum_{S(i-1,m_1)} x_{n_1} \cdots x_{n_{i-1}} + \ldots + x_{m_k}^2 \sum_{S(i-1,m_k)} x_{n_1} \cdots x_{n_{i-1}} = \alpha_{i-1} y[2] - \alpha_{i-2} y[3] + \ldots + (-1)^{i+1} y[i+1]. \tag{20}$$

Consider the elements on the right hand side of equation (20) in which x_{m_1} has power 2. By factoring $x_{m_1}^2$, what remains is all the elements of α_{i-1} in which x_{m_1} is not present. In order to complete this set towards α_{i-1}, add and subtract the terms in α_{i-1} in which x_{m_1} is one of the elements. Now consider the elements on the right hand side of equation (20) in which x_{m_2} has power 2 and apply the same steps as it was done for x_{m_1}. Continuing this process up to the element x_{m_k}, it is possible to factorize the term $\alpha_{i-1} y[2]$ and the residual will be

$$x_{m_1}^3 \sum_{S(i-2,m_1)} x_{n_1} \cdots x_{n_{i-2}} + \ldots + x_{m_k}^3 \sum_{S(i-2,m_k)} x_{n_1} \cdots x_{n_{i-2}}. \tag{21}$$

By applying the same procedure for the expression in equation (21), one can extract the required terms needed on the left side of equation (20) and the last residual will be

$$(-1)^{i+1} \sum_{S(1)} x_{n_1}^{i+1} = (-1)^{i+1} y[i+1]. \tag{22}$$

Which completes the proof.

Consider the case in which the number of discrete diracs is known but the constant weight is unknown. Obviously, there is one more unknown in the system and the number of measurements should be increased. By adding the row of all one to the measurement matrix in equation (13), the unknown weight can be computed easily as the result of the division of the measurement corresponding to the new row by the number of diracs in the input. After dividing all the measurements by this weight, the algorithm can be run in its standard way.

On the other hand, assume that the weight is known but the number of diracs is unknown. To find the number of diracs in the input signal x, it is possible to apply the following algorithm: Take the mesurement, $y[i]$, and find the corresponding α_i by the recursive formula in equation (15), starting with $i = 1$ and increasing the value of i in each step by 1. Whenever α_{k+1} is equal to 0, the number k will be equal to the number of diracs in the input. This argument has the assumption that not any of α_j, $j = 0, \cdots, k$ is zero when there are k diracs in the input and this assumption can be made valid by choosing an appropriate measurement matrix in equation (13) (for example when all the elements of the first row of the measurement matrix are positive). It is interesting to note that again, one needs 1 more measurement in this case to compensate for his lack of knowledge about the number of diracs in the input.

5. SIMULATION RESULTS

In this section we provide simulation results showing the performance of our new sensing and recovery mechanism. We consider the input signal x of length $n = 64$ with $k = 4$ diracs in random positions. The amplitude of the diracs are drawn from a uniform distribution in $[-1, 1]$. The measurement matrix Φ is constructed based on $m = c \cdot k$ consecutive rows of the discrete Fourier matrix where c is the oversampling factor. Figure **??** shows the probability of finding the positions of the diracs in the input signal x as a function of the amount of noise added to the measurements for different values of the oversampling factor c, averaged over 1000 trials. Note that in the noiseless case when the oversampling factor is greater or equal to 2, the annihilating filter method can perfectly reconstruct the input signal. As shown in Ref. 3 for the noiseless case, the Basis Pursuit algorithm needs a high oversampling factor ($c = 6$ for this example) whenever the normalized sparsity defined as $s = \frac{k}{n}$ is small.

In the second scenario, we compare the annihilating filter and the Basis Pursuit recovery algoirthms in the noisy case. Consider the the same signal x of the first scenario and an oversampling factor $c = 6$. For the annihilating filter method, the measurement matrix is based on the discrete Fourier matrix and a random gaussian matrix is used to measure the input signal for the Basis Pursuit algorithm. We use the Basis Pursuit Denoising (BPDN) method to recover the input signal with its recommended penalizing parameter λ, see Ref. 5. In each trial, the mean square error between the input and the reconstructed signal is computed and is normalized by the input signal energy. Figure **??** shows the normalized mean square error (MSE) obtained for the two algorithms against the amount of noise added to the measurements, averaged over 500 trials. It is clear from figure **??** that the new sensing and recovery algorithms perform better than the BPDN algorithm in having less MSE in the reconstruction.

In table **??**, we compare the complexity of our proposed method with that of BPDN. For each value of the signal length n and sparsity level k, we run the two decoding algorithms and find the total time needed for the decoding processes. We put the speed factor as the ratio of the time needed for the BPDN to that of annihilating filter method in the table. We use CVX Matlab package to implement the BPDN algorithm. To have good time estimates, we average the time measurements over 100 trials. Our new method is much less complex than the BPDN, especially for large signals with small sparsity levels.

ACKNOWLEDGMENTS

This work was supported in part by the Swiss National Science Foundation under grant 200020-103729.

REFERENCES

1. D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory* **52**(4), pp. 1289 – 1306.
2. E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory* **52**(2), pp. 489 – 509.
3. M. Duarte, S. Sarvotham, D. Baron, M. Wakin, and R. Baraniuk, "Distributed compressed sensing," *Conference record of thirty-ninth asilomar conference on signals, systems and computers* **24**, pp. 1537 – 1541, October 2005.
4. B. K. Natarajan., "Sparse approximate solutions to linear systems," *SIAM Journal on Computing* **24**, pp. 227–234, April 1995.
5. S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing* **20**(1), pp. 33–61, 1998.
6. S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Transactions on Signal Processing* **41**(12), pp. 3397–3415, 1993.
7. J. Tropp and A. Gilbert, "Signal recovery from partial information via orthogonal matching pursuit," *preprint* , 2005.
8. P. Stoica and R. Moses, *Introduction to Spectral Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 2000.
9. G. Golub and C. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, London, 1996.
10. C. S. Burrus, J. W. Fox, G. A. Sitton, and S. Treitel, "The Lindsey-Fox algorithm for factoring polynomials," *preprint* , 2006.
11. M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Transactions on Signal Processing* **50**, pp. 1417–1428, June 2002.
12. H. Clergeot, S. Tressens, and A. Ouamri, "Performance of high resolution frequencies estimation methods compared to the Cramer-Rao bounds," *IEEE Transactions on Acoustics, Speech and Signal Processing* **37**, pp. 1703–1720, November 1989.
13. T. Blu, P. L. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot, "Sparse sampling of signal innovations: theory, algorithms and performance bounds," *Submitted to IEEE Signal Processing Magazine* , May 2007.
14. J. A. Cadzow, "Signal enhancement-a composite property mapping algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing* **36**, pp. 49–62, January 988.

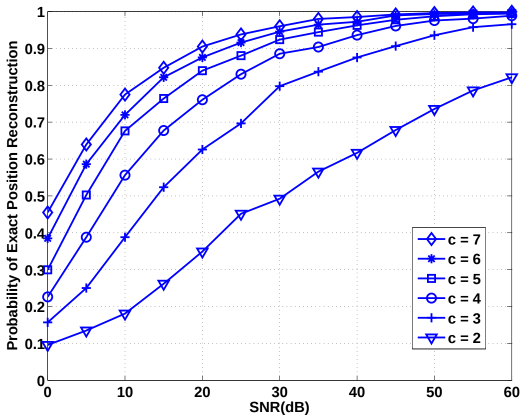

Figure 1. Probability of exact position reconstruction for the annihilating filter method versus SNR for different values of the oversampling factor c.

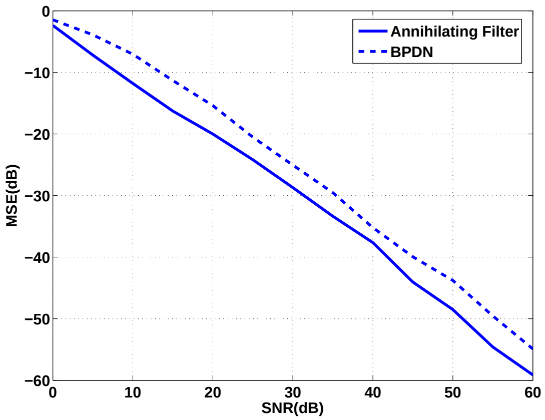

Figure 2. Mean square error of the BPDN and the annihilating filter methods versus SNR. The signal length is $n = 64$, the sparsity level is $k = 4$ and the oversampling factor is $c = 6$.

Table 1. The speed factor of the annihilating filter method versus BPDN for different values of sparsity level k and signal length n. The oversampling factor is set to $c = 6$ and SNR is 20 dB. The black boxes in the table are the cases where the number of measurements are more than the signal dimension n. The entries in the table show the factor by which our method is faster than BPDN.

	$k = 2$	$k = 4$	$k = 6$	$k = 8$	$k = 12$	$k = 24$
$n = 64$	77	41	31	24	■	■
$n = 128$	92	57	44	39	30	■
$n = 256$	124	90	72	68	53	35
$n = 512$	175	151	130	123	102	70
$n = 1024$	290	270	235	220	204	140