

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Applications of the Orthogonal Matching Pursuit/ Nonlinear Least Squares Algorithm to Compressive Sensing Recovery

George C. Valley and T. Justin Shaw  
*The Aerospace Corporation*  
*United States*

## 1. Introduction

Compressive sensing (CS) has been widely investigated as a method to reduce the sampling rate needed to obtain accurate measurements of sparse signals (Donoho, 2006; Candes & Tao, 2006; Baraniuk, 2007; Candes & Wakin, 2008; Loris, 2008; Candes et al., 2011; Duarte & Baraniuk, 2011). CS depends on mixing a sparse input signal (or image) down in dimension, digitizing the reduced dimension signal, and recovering the input signal through optimization algorithms. Two classes of recovery algorithms have been extensively used. One class is based on finding the sparse target vector with the minimum  $\ell_1$  norm that satisfies the measurement constraint: that is, when the vector is transformed back to the input signal domain and multiplied by the mixing matrix, it satisfies the reduced dimension measurement. In the presence of noise, recovery proceeds by minimizing the  $\ell_1$  norm plus a term proportional to  $\ell_2$  norm of the measurement constraint (Candes and Wakin, 2008; Loris, 2008). The second class is based on „greedy“ algorithms such as orthogonal matching pursuit (Tropp and Gilbert, 2007) and iteratively, finds and removes elements of a discrete dictionary that are maximally correlated with the measurement.

There is, however, a difficulty in applying these algorithms to CS recovery for a signal that consists of a few sinusoids of arbitrary frequency (Duarte & Baraniuk, 2010). The standard discrete Fourier transform (DFT), which one expects to sparsify a time series for the input signal, yields a sparse result only if the duration of the time series is an integer number of periods of each of the sinusoids. If there are  $N$  time steps in the time window, there are just  $N$  frequencies that are sparse under the DFT; we will refer to these frequencies as being on the frequency grid for the DFT just as the time samples are on the time grid. To recover signals that consist of frequencies off the grid, there are several alternative approaches: 1) decreasing the grid spacing so that more signal frequencies are on the grid by using an overcomplete dictionary, 2) windowing or apodization to improve sparsity by reducing the size of the sidelobes in the DFT of a time series for a frequency off the grid, and 3) scanning the DFT off integer values to find the frequency (Shaw & Valley, 2010). However, none of these approaches is really practical for obtaining high precision values of the frequency and amplitude of arbitrary sinusoids. As shown below in Section 6, calculations with time windows of more than 10,000 time samples become prohibitively slow; windowing distorts the signal and in many cases, does not improve sparsity enough for CS recovery algorithms

to work; scanning the DFT off integer values requires performing the CS recovery algorithm over and over again with an unknown sparse transform and becomes prohibitively expensive when the number of sinusoids in the signal exceeds 1.

Here we present a new approach to recovering sparse signals to arbitrary accuracy when the parameters of the signal do not lie on a grid and the sparsifying transform is unknown. Our approach is based on orthogonal matching pursuit (OMP), which has been applied to recovering CS signals by many authors (Donoho et al., 2006; Tropp and Gilbert, 2007; Liu and Temlyakov, 2010; Huang and Zhu, 2011). The major difference between our work and previous work is that we add a nonlinear least squares (NLS) step to each OMP iteration. In the first iteration of conventional OMP applied to finding sinusoids, one finds the frequency that maximizes the correlation between the measurement matrix evaluated on an overcomplete dictionary and the CS measurement, solves a linear least squares problem to find the best estimate of the amplitude of the sinusoid at this frequency, and subtracts this sinusoid multiplied by the measurement matrix from the CS measurement. In the second iteration, one finds the frequency that maximizes the correlation between the measurement matrix and the residual measurement, solves a least squares problem for both frequencies to get new estimates of both amplitudes, and subtracts the sum of the two sinusoids multiplied by the measurement matrix from the previous residual. This process is described in detail in „Algorithm 3 (OMP for Signal Recovery)“ in the paper by Tropp and Gilbert (2007) and in our Table 1 in Section 3. Our approach proceeds in the same way as conventional OMP but we substitute a **Nonlinear Least Squares** step for the linear least squares step. In the NLS step, we use a minimizer to find better values for the frequencies without reference to a discrete grid. Because the amplitudes are extremely sensitive to the accuracy of the frequencies, this leads to a much better value for the amplitudes and thus to a much more accurate expansion of the input signal. Just as in conventional OMP, we continue our algorithm until a system level threshold in the residual is reached or until a known number of sinusoids is extracted. A related procedure for matching pursuit but not yet applied to compressive sensing or orthogonal matching pursuit is described by Jacques & De Vleeschouwer (2008). What we refer to as the NLS step appears in their Section V, eq. (P.2).

Our approach to CS recovery differs from most methods presented to date in that we assume our signal (or image) is sparse in some model as opposed to sparse under some transform. Of course, for every sparse model there is some sparsifying transform, but it may be easier in some problems to find the model as opposed to the transform. Models inevitably involve parameters, and in most cases of practical interest, these parameters do not lie on a discrete grid or lie on a grid that is too large for efficient discrete processing techniques (see the discussion in Section 1 of Jacques & De Vleeschouwer, 2008). For instance, to recover the frequency of a sinusoid between 0 and 1 to precision of  $10^{-16}$  would require  $10^{16}$  grid points. While we first developed our method to find the frequency and amplitude of sinusoids, like OMP it is readily adaptable to signals that are the superposition of a wide range of other models. In Section 2, we present background material on the OMP, NLS and CS methods on which our method is based. In Section 3, we develop the model-based OMP/NLS formulation. Sections 4 and 5 contains the application to signals that consist of a sum of a small number of sinusoids. Section 6 compares performance of our algorithm to conventional OMP using a linear least square step and to penalized ell-1 norm methods.

## 2. Background

Our method and results rely heavily on work in three well-known areas: orthogonal matching pursuit, nonlinear least squares and compressive sensing.

### 2.1 Compressive sensing

In compressive sensing (Donoho, 2006; Candes & Tao, 2006; Baraniuk, 2007), a sparse vector  $s$  of dimension  $N$  can be recovered from a measured vector  $y$  of dimension  $M$  ( $M \ll N$ ) after transformation by a sensing matrix  $\Theta$  as shown in eq. (1)

$$y = \Theta s + n \quad (1)$$

where  $n$  is a noise vector. Often,  $\Theta$  is factored into two matrices,  $\Theta = \Phi\Psi$  where  $\Phi$  is a „random“ mixing matrix and  $\Psi$  is a Hermitian matrix with columns that form a basis in which the input vector is sparse. A canonical example is the case in which the input is a time series with samples taken from a single sinusoid with an integer number of periods in the time window. These data are not sparse but are transformed into a sparse vector by the discrete Fourier transform (DFT). Note that although  $\Theta$  is not square and hence not invertible,  $\Psi$  is both square and invertible. Work in compressive sensing has shown (Donoho, 2006; Candes & Tao, 2006; Baraniuk, 2007) that under quite general conditions, all  $N$  components of  $s$  may be recovered from the much smaller number of measurements of  $y$ . With no noise ( $n = 0$ ) recovery proceeds by minimizing the ell-1 norm of a test vector  $s'$  (the ell-1 norm of  $s'$  is given by the sum of the absolute values of the elements of  $s'$ ) subject to the constraint  $y = \Theta s'$ . In the presence of noise, recovery proceeds by minimizing a linear combination of the ell-1 norm of the target vector and the ell-2 norm of the residual vector given by  $y - \Theta s$

$$s'(\lambda) = \operatorname{argmin}_s (\lambda \|s\|_1 + \|y - \Theta s\|_2^2) \quad (2)$$

where the parameter  $\lambda$  is chosen such that the signal is optimally recovered (Baraniuk, 2007; Loris, 2008).

### 2.2 Orthogonal Matching Pursuit method

Orthogonal matching pursuit (OMP) is an alternative method that can be used to find the target vector  $s$  from the measurement vector  $y$ . Matching pursuit has a rich history in signal processing long before CS and has appeared under many names (Mallat & Zhang, 1993; Pati et al., 1993; Davis et al., 1997). With the advent of CS, many variants of OMP have been applied to recovery including methods called MOMP, ROMP, CoSaMP, etc. (Needell and Tropp, 2008; Needell and Vershynin, 2009; Huang and Zhu, 2011) but with one exception (Jacques and De Vleeschouwer, 2008) discussed below, all of these methods recover frequencies (or other parameters) from discrete grids.

The basic idea of all matching pursuit algorithms is to minimize a cost function to obtain frequencies of sinusoids present in the signal. First, take the frequency corresponding to the smallest value of the cost function and calculate the linear least squares estimate for the complex amplitude at this frequency. Second, mix this sinusoid with the known CS mixing matrix  $\Phi$  and remove this mixed approximation to the first sinusoid from the CS measurement vector [ $y$  in eq. (1)]. This process is repeated until a known number of sinusoids is found or a system-defined threshold is reached. For frequencies not on the DFT

grid of the time series, OMP can be improved by evaluating the cost function on an overcomplete dictionary (Candes et al. 2011), but as in the  $\ell_1$  estimates discussed above, this step becomes computationally intractable long before machine precision can be obtained for arbitrary frequencies.

### 2.3 Nonlinear Least Squares method

Here we follow the development of nonlinear least squares (NLS) given by Stoica and Moses (1997). Their eq. (4.3.1) defines a cost function to be minimized as a function of the vectors  $\omega, \alpha, \phi$

$$f(\omega, \alpha, \phi) = \sum_t |y(t) - \sum_k \alpha_k \exp[i(\omega_k t + \phi_k)]|^2 \quad (3)$$

where the sums are over the number of sinusoids present in the signal,  $k = 1$  to  $K$  and the time points run from  $t = 0$  to  $N-1$ . Stoica and Moses also show (see their eqs. 4.3.2-4.3.8), that the frequency vector  $\omega$  is the critical unknown and the amplitude and phase (or complex amplitude) are simply „nuisance“ parameters that are obtained from  $\omega$ . While eq. (3) appears to require simultaneous solution for three real vectors, each of length  $K$ , Stoica and Moses (eqs. 4.3.2-4.3.8) show that the problem can be reduced to solving for just the frequency vector  $\omega$  and that the complex amplitude vector can be calculated directly from the frequency vector. We use a version of these equations below in eqs. (8) and (13).

In principle, solution of the CS analog of eq. (3) could be performed to directly obtain the parameters of a sparse signal, but in practice, direct solution of eq. (3) is not computationally practical (Stoica and Moses, 1997). The difficulty is that even for a small  $K$ , eq. (3) is highly multimodal (see for example, Fig. 1 in Li et al., 2000) and the solution requires extremely good first guesses for the vector  $\omega$ . Even with good initial values for  $\omega$ , performance guarantees are difficult to find and continue to be the subject of intense investigation (Salzo and Villa, 2011 and references therein).

Similar two-step model-based approaches to estimating the frequency and amplitude of real and complex sinusoids have been discussed previously in the literature (Stoica et al., 2000; Li et al., 2000; Chan and So, 2004; Christensen and Jensen, 2006). Stoica et al. discuss the use of NLS to obtain the amplitude for complex sinusoidal signals given the frequency; Li et al. and Chan and So discuss a combined matching pursuit NLS approach similar to ours for obtaining the frequencies of complex and real harmonic sinusoidal signals, respectively; and Christensen and Jensen use matching pursuit plus NLS to estimate frequencies in a signal that is the sum of arbitrary frequency real sinusoids. To the best of our knowledge, our paper is the first application of an OMP/NLS algorithm to estimate the frequency and amplitude from CS measurements.

## 3. Formulation of OMP with an NLS step for CS

### 3.1 Mathematical development

Consider a continuous time signal  $X(t)$  consisting of  $K$  complex sinusoids of the form

$$X(t) = \sum_{k=1}^K a_k \exp(2 \pi i f_k t) \quad (4)$$

where  $a_k$  is the complex valued amplitude and  $f_k$  is the real valued frequency of the  $k^{\text{th}}$  sinusoid. This signal model is broadly applicable [see Duarte and Baraniuk (2010) and



references therein]. We take  $f_{\min} = 0$  and  $f_{\max} = 1$  to set up our test problem; we sample  $X(t)$  at the Nyquist rate for complex signals,  $\Delta t = 1/f_{\max} = 1$ , to obtain the sampled time series  $X_S$  of length  $N$  from  $t = 0$  to  $t = N-1$  where  $N$  is the number of time samples. As in all applications of compressive sensing, we make a sparsity assumption,  $K \ll N$ , and mix the input signal vector  $X_S$  plus sampled noise  $n$  down in dimension to the measured vector  $y$  of dimension  $M$ :

$$y = \Phi(X_S + n), \quad (5)$$

where  $\Phi$  is an  $M \times N$  mixing matrix and  $K < M \ll N$ . Note that in eq. (5)  $n$  is added to  $X_S$  prior to mixing. In other models noise is added to the mixed version of  $X_S$ ,  $\Phi X_S$ , or even to  $\Phi$  itself. We generate the elements of  $\Phi$  using the pseudo-random number functions in our software (*Mathematica* and *Python*) such that they are taken uniformly from the set of nine complex numbers:  $\{-1 - i, -1, -1 + i, -i, 0, i, 1 - i, 1, 1 + i\}$  or equivalently, the elements are taken from the sum of random integers drawn from  $\{-1, 0, 1\}$  plus  $i$  times different random integers drawn from  $\{-1, 0, 1\}$ . We use a complex mixing matrix because our signal model is complex. The noise is assumed to be independent and identically distributed (i.i.d.) Gaussian noise with standard deviation  $\sigma/2^{1/2}$  and is added to the real and the imaginary part of each element of  $X_S$ , so that the covariance of  $n$  is  $\sigma^2 I$ , where  $I$  is the  $N \times N$  identity matrix. If the frequencies lie on the DFT frequency grid associated with the time series  $t = 0$  to  $t = N-1$ , eq. (5) can be solved for the frequencies by writing  $s = \text{DFT } X_S$ , substituting  $X_S = \text{IDFT } s$  (IDFT = Inverse DFT) in eq. (5), and solving  $y = \Phi(\text{IDFT } s + n)$  by minimizing the ell-1 norm of  $s$  subject to the measurement constraint eq. (5) if  $n = 0$  or by minimizing the ell-1 norm penalized by an arbitrary fraction of the constraint (LASSO) in the presence of noise (Candes & Wakin, 2008; Loris, 2008).

Although the noise is assumed to be i.i.d., the mixing matrix  $\Phi$  colors the noise in the observation vector  $y$ . The covariance of  $y$  is given by

$$\text{Cov}[y] = \sigma^2 \Phi \Phi^H, \quad (6)$$

and the standard maximum likelihood estimator requires definition of a weighting matrix  $W$ ,

$$W = (\Phi \Phi^H)^{-1}, \quad (7)$$

where the superscript  $H$  indicates the Hermitian conjugate (see Stoica et al., 2000 and Chan and So, 2004, for a discussion of weighted estimators in NLS). If the inverse in eq. (7) is ill-conditioned or does not exist, this indicates a poor choice of mixing matrix  $\Phi$  and another one should be chosen. The maximum likelihood estimator (MLE) for  $X_S$ ,  $C(Z)$  is solved by finding the vector  $Z$  that minimizes the weighted square of the residual given by

$$C(Z) = (\Phi Z - y)^H W (\Phi Z - y); \quad (8)$$

$Z$  is a vector taken from the linear subspace spanned by at most  $K$  complex sinusoids sampled over  $t = 0$  to  $N-1$  (see the corresponding equation for determining the amplitude and frequency of a sum of complex sinusoids in a system that does not have compressive sensing, Stoica and Moses, 1997, eq. 4.3.6). CS recovery is equivalent to determining the spectral support (that is, the  $K$  unknown frequencies) of the input signal  $X_S$ , or equivalently determining the vector  $Z$  that minimizes eq. (8) (Duarte & Baraniuk, 2010). In the absence of noise, weighting with  $W$  is unnecessary because the solution is exact and both the weighted

and un-weighted residuals are zero. Finding the  $K$  sinusoids that solve eq. (8) is the standard NLS problem and if this were computationally tractable, the problem would be solved. But as pointed out by Li et al. (2000) [see also the discussion in Stoica & Moses (1997)], “the NLS cost function in (3) is usually multimodal with many local minima,” and “the minimization of the NLS cost function requires the use of a very fine searching algorithm and may be computationally prohibitive.”

One way out of this dilemma is to use NLS in place of least squares within OMP. This has two advantages over using NLS by itself. First, the frequency band over which one has to search in NLS is reduced from the entire frequency band to the frequency grid spacing in the over-complete dictionary used in OMP. Second, the estimates of the frequencies at any given iteration are improved from the values on the grid by using NLS in the previous iteration (see the discussion of a similar continuous version of matching pursuit by Jacques & De Vleeschouwer, 2008).

The first step in our formulation is to define the vector function of frequency,  $x(f)$ , as the time series for a unity amplitude complex sinusoid at frequency  $f$  evaluated at integral sampling times  $t = 0$  to  $t = N - 1$ ,

$$x(f) = [1, e^{i2\pi f}, e^{i4\pi f}, \dots, e^{i2(N-1)\pi f}]. \quad (9)$$

Note that the solution for  $X_s$  in eq. (5) is a linear combination of  $K$  vectors  $x(f_i)$ , ( $i = 1, K$ ). To use OMP, we need an over-complete dictionary (Candes et al., 2010) which means that  $x(f)$  is evaluated on a fine grid oversampled by the factor  $N_f$  from the DFT grid. The second step is to define a function that can be evaluated on the fine grid to find a grid frequency close to one of the true frequencies in the input signal  $X_s$ . Here we use the function  $G(f, r)$  given by

$$G(f, r) = 1 / \left| r^* W r - \frac{[\Phi x(f)]^* W r}{\sqrt{[\Phi x(f)]^* W [\Phi x(f)]}} \right| \quad (10)$$

where initially  $r = y$  and subsequently,  $r$  equals the residual of  $y$  with 1 to  $K$  components removed as discussed below. We calculate the argmax of  $G(f, y)$  over  $f$  in the dictionary frequencies to make a first estimate of one of the frequencies present in the input signal  $X(t)$ . If there is no noise and if there is only one sinusoid, this procedure provides the dictionary vector whose frequency is nearest that of the input sinusoid. If multiple sinusoids are present, the maximum of  $G(f, y)$  occurs at one of the dictionary vectors whose frequency is near one of the input sinusoids provided that the dictionary is sufficiently over-complete and that  $\Phi$  possesses the restricted isometry property (Duarte and Baraniuk, 2010). Note that  $G(f, r)$  is the inverse square of the distance between  $r$  and the linear span of  $x(f)$  in the  $W$ -normed inner product space (defined by  $\langle a, b \rangle = a^H W b$ ). Thus finding the argmax of  $G(f, r)$  is equivalent to finding the argmax of the inner product of the residual with the product of  $\Phi$  times the dictionary vectors  $x(f_i)$  for all  $f_i$  on the over-complete frequency grid (see Tropp and Gilbert, 2007, Algorithm 3, Step 2).

Given estimates of the frequencies  $\{f_1, f_2, \dots, f_K\}$  present in the input signal, we can find estimates of the amplitudes of each sinusoid by using the least squares estimator  $A(U)$  for the amplitude vector  $\{a_1, a_2, \dots, a_K\}$  (see Stoica and Moses, 1997, eq. 4.3.8 and Stoica et al., 2000)

$$A(U) = (U^H W U)^{-1} U^H W y \quad (11)$$

where  $\mathbf{U}$  is the spectral support matrix given that depends on  $\{f_1, f_2, \dots, f_k\}$  through

$$\mathbf{U} = \Phi \{x(f_1), x(f_2), \dots, x(f_k)\} \quad (12)$$

Note that if there is no noise and if all frequencies are known exactly, eq. (11) can be verified by substituting  $\mathbf{y} = \mathbf{U} \mathbf{A}(\mathbf{U})$ , which is equivalent to eq. (5), on the right hand side of eq. (11).

Finally, starting from estimates of the frequencies and amplitudes from OMP as described above, apply weighted NLS to get better values. This is done by finding the frequency or set of frequencies  $\mathbf{f} = \{f_1, f_2, \dots, f_k\}$  that minimize the functional  $R(\mathbf{f})$  given by

$$R(\mathbf{f}) = \|\mathbf{A}[\mathbf{U}(\mathbf{f})]\mathbf{U}(\mathbf{f}) - \mathbf{y}\|^H \mathbf{W} \|\mathbf{A}[\mathbf{U}(\mathbf{f})]\mathbf{U}(\mathbf{f}) - \mathbf{y}\|, \quad (13)$$

which is the same as the weighted least squares estimator given by eq. (8) with the substitution of  $\mathbf{A}[\mathbf{U}(\mathbf{f})]$  defined by eq. (11) for the amplitude vector and  $\mathbf{U}(\mathbf{f})$  defined by eq. (12) for the mixed sinusoids (see the analogous equations in Stoica and Moses, 1997, eqs. 4.3.7 and 4.3.8). The product  $\mathbf{A}[\mathbf{U}(\mathbf{f})] \mathbf{U}(\mathbf{f})$  in eq. (13) is the same as  $\Phi \mathbf{Z}$  in eq. (8).

### 3.2 Algorithm description

As described in Table 1, the first step in the first iteration of the Do loop is estimation of the first frequency in the spectral support of the signal  $\mathbf{X}_s$ . This is given by the frequency of the sinusoid whose image after multiplication by  $\Phi$  has the maximum correlation with the observation vector  $\mathbf{y}$  (see, for example, Tropp and Gilbert, 2007 Algorithm 3, step 2). Here we use the equivalent form, the argmax of  $G(\mathbf{f}, \mathbf{y})$  with respect to  $\mathbf{f}$  to obtain the first estimate of the frequency of the first sinusoid  $f_1$  in eq. (4). At this point previous implementations of discrete OMP use the amplitude estimator eq. (11) to estimate the amplitude of the first sinusoid  $a_1 = \mathbf{A}[\Phi x(f_1)]$ , multiply this amplitude estimate times  $x(f_1)$ , given by eq. (9), and by the measurement or mixing matrix  $\Phi$  and subtract this vector from the measurement vector  $\mathbf{y}$  to form the first residual  $\mathbf{r}_1$ .

In our algorithm, we proceed differently by improving the precision of the frequency estimates using NLS before finding the amplitude estimate. We take the frequency  $f_1$  from the argmax of  $G(\mathbf{f}, \mathbf{y})$  evaluated on a discrete set of frequencies and use that as the starting value to solve the NLS problem given by eq. (13). We have used several methods and several different software packages to solve the NLS problem. A simple decimation routine [i.e., tabulating  $R(\mathbf{f})$  from  $f_1 - \Delta f$  to  $f_1 + \Delta f$  ( $\Delta f$  is the over-complete grid spacing) in 10 steps, finding the argmin, decreasing  $\Delta f$  by a factor of 10, tabulating and finding the argmin of  $R(\mathbf{f})$  again until the specified precision is reached] works well but is not very efficient. Powell's method in *Python* ("scipy.optimize.fmin\_powell") and one of the Newton methods, the PrincipalAxis method, and the Conjugate Gradient method in *Mathematica*'s minimizer "FindMinimum" all work and take less time than the decimation routine. A detailed investigation of minimizers for the NLS step in our version of OMP is beyond the scope of this chapter. The oversampling  $N_f$  required for our method and that required for conventional OMP are nearly identical as discussed below in Section 6.

Given the better value of  $f_1$ , we compute  $a_1$  from eq. (11) and a new value of the residual  $\mathbf{r}$  with the NLS estimate of the first signal removed from  $\mathbf{y}$  as in OMP

$$\mathbf{r}_1 = \mathbf{y} - \mathbf{A}(\mathbf{U}_1)\mathbf{U}_1 = \mathbf{y} - a_1 \Phi x(f_1). \quad (14)$$



where  $\mathbf{U}_1 = \Phi x(f_1)$ . The argmax of  $G(f, r_1)$  now yields a first estimate of the frequency of the second sinusoid,  $f_2$ . Next improve the estimates of both  $f_1$  and  $f_2$  by again solving the NLS problem by minimizing the functional  $R(f)$  over  $f = \{f_1, f_2\}$ . Note that this overwrites the previous estimate of the first frequency  $f_1$ . The amplitudes  $a_1$  and  $a_2$  are recalculated using (8) with  $\mathbf{U}_2$  given by

$$\mathbf{U}_2 = [ \Phi x(f_1), \Phi x(f_2) ]$$

(15)

for the latest values of  $f_1$  and  $f_2$ . Finally, in this iteration estimates of the first two sinusoids are removed from  $y$ :

$$r_2 = y - A(\mathbf{U}_2) \mathbf{U}_2.$$

(16)

If  $K$ , the total number of sinusoids present in the signal, is known, this process is repeated  $K$  times until  $f_K$  and  $a_K$  are obtained. In the absence of noise, the sum of these sinusoids solves (5) exactly and  $r_K = 0$ .

Inputs:
CS Mixing Matrix $\Phi$
Measured data $y$
Maximum number of sinusoids $K$ or threshold $T$
$f_{\min}, f_{\max}$
Oversampling ratio for dictionary $N_f$
Initialize
$\mathbf{U} = [ ]$
$r_0 = y$
$K_T = K$
$\mathbf{W} = (\Phi \Phi^H)^{-1}$
$\Delta f = (f_{\max} - f_{\min}) / (N N_f)$
Do $i = 1$ to $K$
$f_i = \text{Argmax } G(f, r_{i-1}) \text{ over } \{f_{\min}, f_{\min} + \Delta f, \dots, f_{\max} - \Delta f, f_{\max}\}$
$\{f_1, f_2, \dots, f_i\} = \text{Argmin}[R(f) \text{ with initial value } f = \{f_1, f_2, \dots, f_i\}]$
$\mathbf{U} = \{\Phi x(f_1), \Phi x(f_2) \dots \Phi x(f_i)\}$
$r_i = y - A(\mathbf{U}) \mathbf{U}$
If $r_i^H \mathbf{W} r_i < T$ :
$K_T = i$
Break
End If
End Do
Output of Do:
$K_T$
$\{f_1, f_2, \dots, f_{K_T}\}$
$\{a_1, a_2, \dots, a_{K_T}\} = A [\Phi x(f_1), \Phi x(f_2) \dots \Phi x(f_{K_T})]$

Table 1. OMP/NLS Algorithm.

There are two methods to handle the case where the actual number of sinusoids present is unknown, yet still smaller than  $K$ . The simpler method, applicable for high SNR ( $\sigma$  small compared to the smallest signal amplitude), is to perform  $K$  iterations of the OMP/NLS algorithm, which will incur an additional noise folding penalty, by projecting the additional noise dimensions onto the solution. The second method is to stop when the residual can be explained by noise alone through hypothesis testing. At the solution, the weighted squared residual  $r_k^H W r_k$  will display a  $\chi$ -squared statistic with  $2k$  degrees of freedom, where  $k$  is the actual number of sinusoids present in the signal. The hypothesis that the residual is caused by noise alone, is accepted when  $r_k^H W r_k < \sigma^2 T$  for some threshold  $T$  and rejected otherwise. The value for  $T$  is dependent on a user selectable significance level, the probability of incorrectly rejecting the given hypothesis. For a significance level of  $\alpha$ ,  $T = \text{CDF}^{-1}(1-\alpha)$ , where CDF is the cumulative distribution function of the chi-squared distribution with  $2k$  degrees of freedom. We used  $\alpha = 0.05$  in our simulations, but  $\alpha$  is an application-specific parameter.

## 4. Results for sparse sinusoids without noise

### 4.1 Signal composed of a single sinusoid

Consider first a signal of the form given by eq. (4) with  $K = 1$ ,  $f_1 = 0.44681715350529533$  and  $a_1 = 0.8018794857541801$ . Fig. 1 shows a plot of  $G(f, y)$  for  $N = 1024$ ,  $M = 4$ . Finding the argmax of  $G(f, y)$  evaluated for 32,768 frequencies between  $f_{\min}$  and  $f_{\max}$  ( $N_f = 32$ ) yields an initial value for the frequency and amplitude of  $f_1 = 0.446808$  and  $a_1 = 0.801070 + 0.026421i$ . Minimization of  $R[\{\Phi x(f_1)\}]$  starting at  $f_1 = 0.446808$  yields a final value  $f_1$  equal to the input frequency  $f_1$  with error less than  $1 \times 10^{-16}$  (machine precision) and the amplitude  $a_1$  through  $A[\{\Phi x(f_1)\}]$  equal to the input amplitude with error less than  $4 \times 10^{-16}$ .

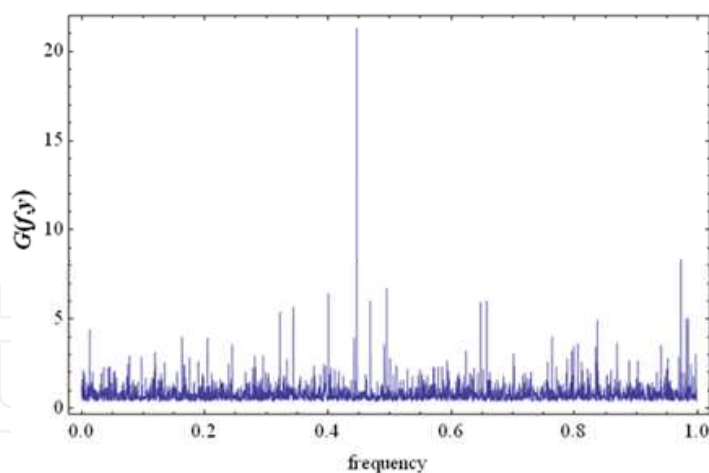


Fig. 1.  $G(f, y)$  as a function of frequency for a signal composed of a single sinusoid mixed with  $N = 1024 \times 4$ . Note the appearance of a single strong peak in the estimator that serves as an excellent starting value for minimizing the functional  $R(f)$  given in eq. (13).

### 4.2 Signal composed of a 20 sinusoids

The algorithm also works for multiple frequencies. More than 20 independent tests were performed for an input signal composed of 20 independent frequencies randomly chosen between 0 and 1; all frequency components have amplitude of 1. In all tests our algorithm recovered the 20 frequencies to machine precision with a  $128 \times 1024$  mixing matrix. For test 1,

shown in detail here, the closest frequency pairs in the signal are  $\{0.2663, 0.2689\}$  and  $\{0.7715, 0.7736\}$ , but while signals with nearly the same frequency are difficult cases, here the combined OMP/NLS recovers all the sinusoids to machine precision. Fig. 2 shows the initial calculation of  $G(f, y)$  for a  $128 \times 1024$  mixing matrix and 8192 frequency points ( $N_f = 8$ ). Note that most, but not all of the frequencies have peaks in the initial scan of  $G(f, y)$ . Fig. 3 shows  $G(f, r_{19})$  during the 20<sup>th</sup> iteration of the Do loop in the algorithm shown in Table 1. After refining the frequencies by finding the minimum of  $R(f)$  in (10), the frequency errors are reduced to less than  $10^{-16}$  and the amplitude errors are reduced to  $4 \times 10^{-14}$ . Our results compare favorably to those obtained using other recovery methods for a test problem with 20 arbitrary frequency complex sinusoids,  $N = 1024$ , and variable numbers of measurements  $M$  (Duarte and Baraniuk, 2010).

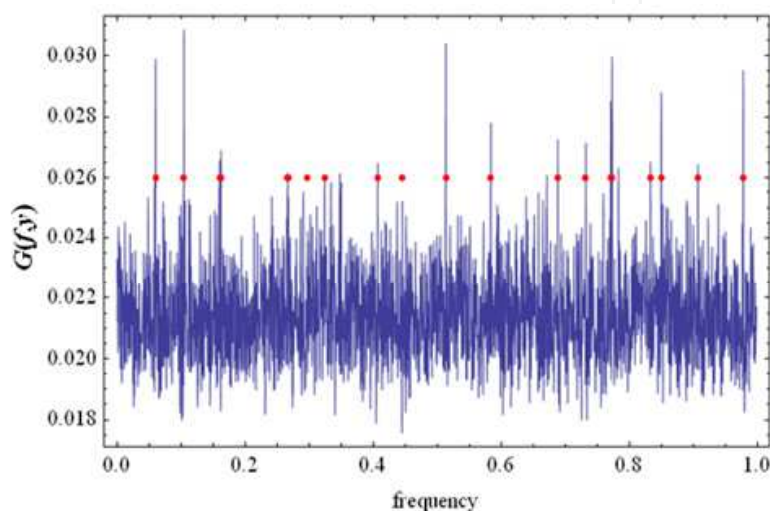


Fig. 2. The initial calculation of  $G(f, y)$  for a signal with 20 input frequencies mixed with a  $128 \times 1024$  matrix. The red dots indicate the input frequencies.

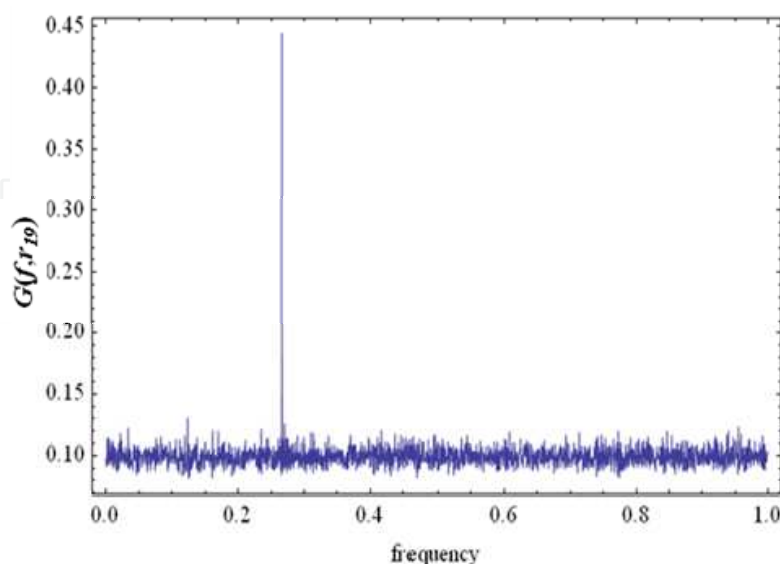


Fig. 3. The next to last calculation of  $G(f, r_{19})$  for a signal with 20 input frequencies mixed with a  $128 \times 1024$  matrix showing a large peak near the frequency of the only remaining sinusoid.

### 4.3 Signal composed of 2 sinusoids with large dynamic range

For signals composed of 2 well separated frequencies but widely different amplitudes in the absence of noise, we recover the amplitude and frequency of the 2 sinusoids when  $a_1 = 1$  and  $a_2$  is as small as  $10^{-14}$  with an  $8 \times 1024$  mixing matrix. For this case the amplitude and frequency of the large signal are recovered to machine precision while the frequency and amplitude error of the weak signal are  $3 \times 10^{-4}$  and 1%, respectively. Naturally, such performance is not found in the presence of noise as discussed below.

### 4.4 Signal composed of 2 sinusoids with closely spaced frequencies

We have also input a signal with 2 very closely spaced frequencies and unity amplitudes. For frequencies  $\{0.3389, 0.3390\}$  we recover the frequencies to machine precision with a  $16 \times 1024$  mixing matrix. Smaller values of  $M$  for the mixing matrix yield one root half way between the two frequencies. For frequencies  $\{0.33899, 0.33900\}$  mixed with  $16 \times 1024$  and  $32 \times 1024$  matrices the OMP part of our algorithm yields a signal with one frequency at 0.338995 and an amplitude of 1.9996. Attempts to find a second frequency yield a badly conditioned matrix for  $U^H W U$  and the inversion required to find the 2<sup>nd</sup> amplitude in eq. (11) fails. For a  $64 \times 1024$  mixing matrix OMP finds two separated estimates of the frequencies and this allows NLS determination of both frequencies to an accuracy of a few parts in  $10^5$ . These results are in contrast to those obtained using the “spectral compressive sensing” algorithms that use “a signal model that inhibits closely spaced sinusoids” (Duarte and Baraniuk, 2010).

### 4.5 Dependence on dimensions of the mixing matrix

We have investigated the requirements on  $M$ , the small dimension of the measurement matrix, to recover a signal composed of a small number of sinusoids using the OMP-NLS algorithm. Fig. 4 shows the fraction of failed recoveries as a function of  $M$  for a problem in which the signal is composed of 1, 3, 5, or 7 sinusoids and  $N = 128$ . For each value of  $K$  we performed 1000 trials so a failure fraction of 0.1 corresponds to 100 failures. The conventional relation between  $K$ ,  $M$ , and  $N$  for recovery is given by  $M = C K \log(N/K)$  (Baraniuk, 2007; Candes and Wakin, 2008). From Fig. 4 we see that the curves for  $K = 3, 5$  and 7 are equispaced and correspond to  $C \sim 1.5$ .

We have also investigated several different types of the measurement matrix as displayed in Fig. 5. The three curves correspond to three different measurement matrices. For the blue curve the mixing matrix is generated from the sum of random integers drawn from  $\{-1, 0, 1\}$  plus  $i$  times different random integers drawn from  $\{-1, 0, 1\}$ ; for the red curve, complex numbers with the real and imaginary parts given by reals uniformly distributed between -1 and 1 and  $i$  times uniformly distributed reals; for the magenta curve, the mixing matrix is generated from randomly chosen -1's and 1's. The magenta curve for a real mixing matrix made from 1's and -1's is inferior to the blue and red curves for the two complex mixing matrices. The differences between the red and blue curves in Fig. 5 appear to be random fluctuations and are in agreement with other CS results that Gaussian and Bernoulli measurement matrices perform equally well (Baraniuk, 2007; Candes and Wakin, 2008). Fig. 6 compares calculations with the weighting matrix given by eq. (7) to calculations with the weighting matrix set to the identity matrix. One can see that the green curve with the weighting matrix set to the identity matrix is significantly worse in the important region of less than 1% failure.

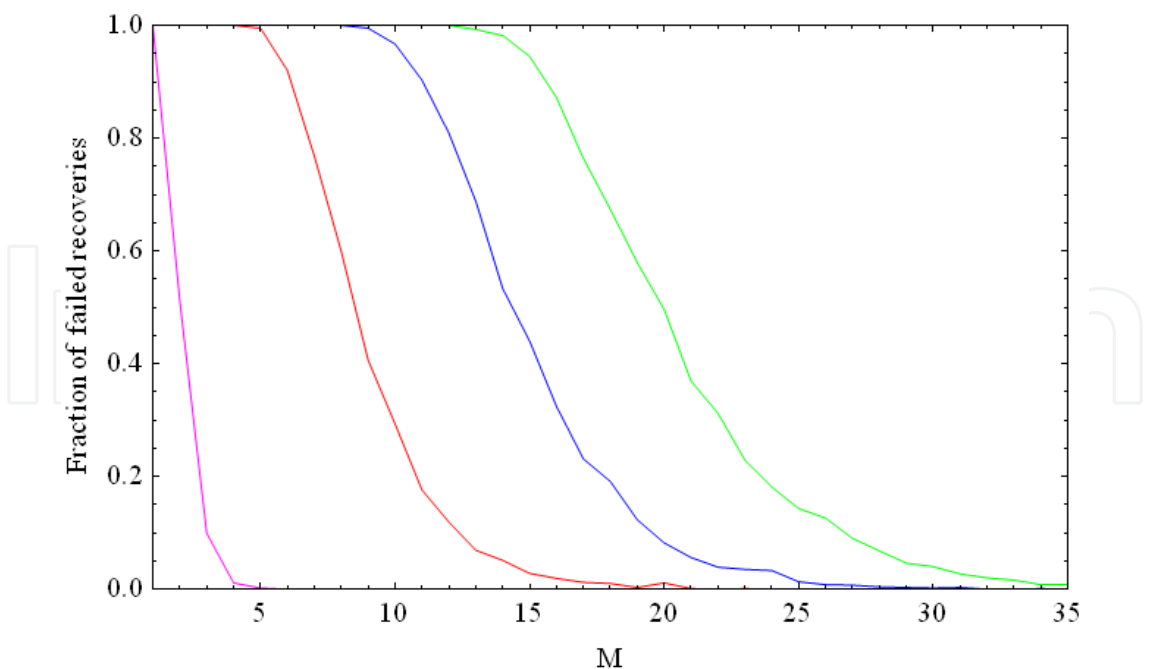


Fig. 4. Fraction of failed recoveries as a function of the small dimension of the mixing matrix  $M$  for signals consisting of 1 (magenta), 3 (red), 5 (blue) and 7 (green) sinusoids. The large dimension of the mixing matrix is  $N = 128$  and 1000 trials were performed for each value of  $M$ .

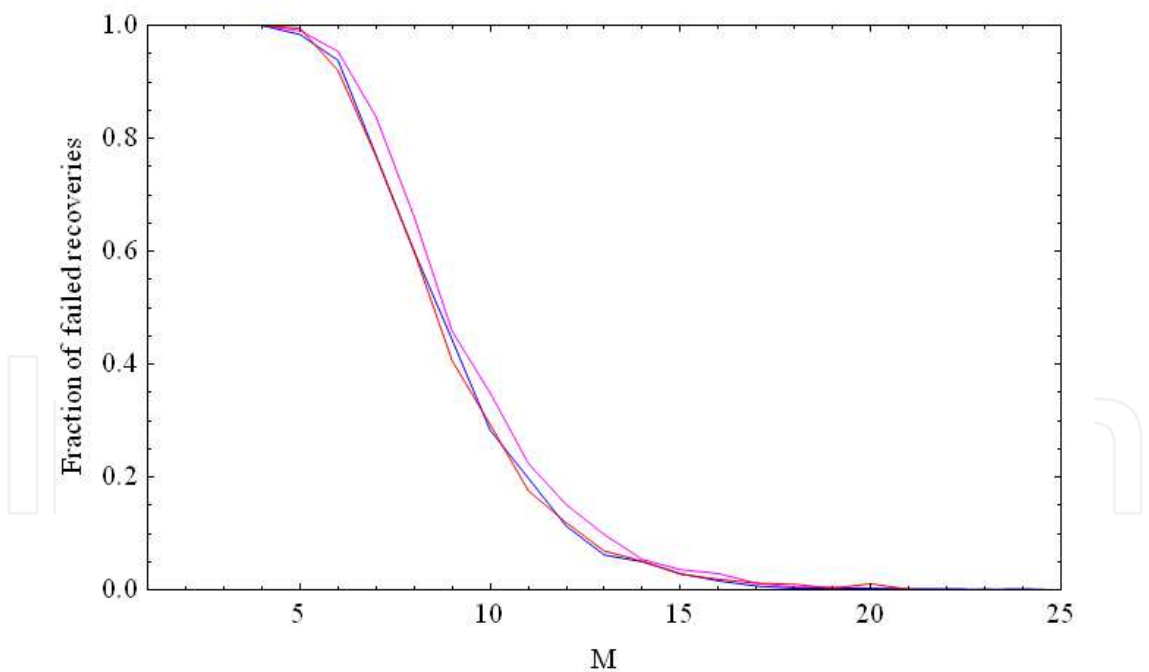


Fig. 5. Fraction of failed recoveries as a function of the small dimension of the mixing matrix  $M$ . For the blue curve the mixing matrix is generated from the sum of random integers drawn from  $\{-1,0,1\}$  plus  $i$  times different random integers drawn from  $\{-1,0,1\}$ ; for the red curve, complex numbers with the real and imaginary parts given by reals uniformly distributed between -1 and 1; for the magenta curve, the entries of the mixing matrix are randomly chosen from -1 and 1.



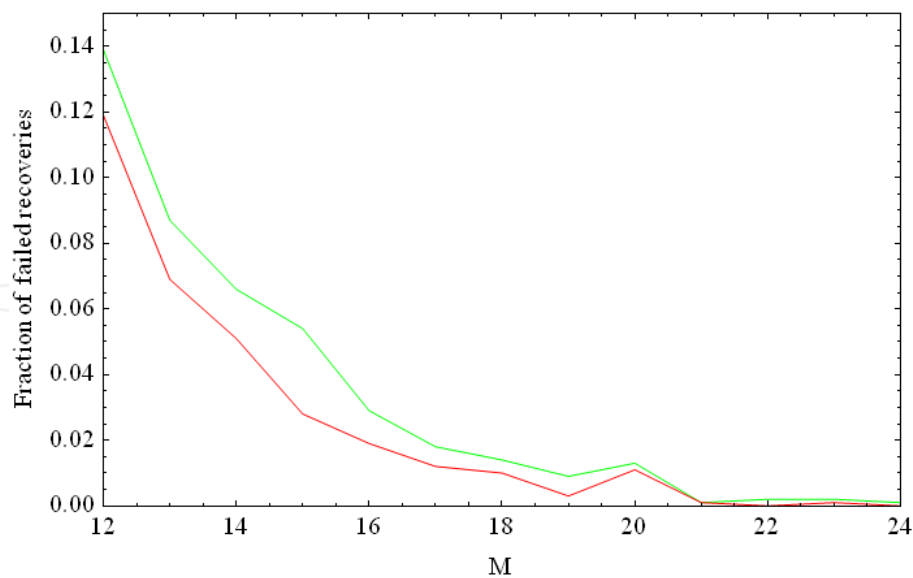
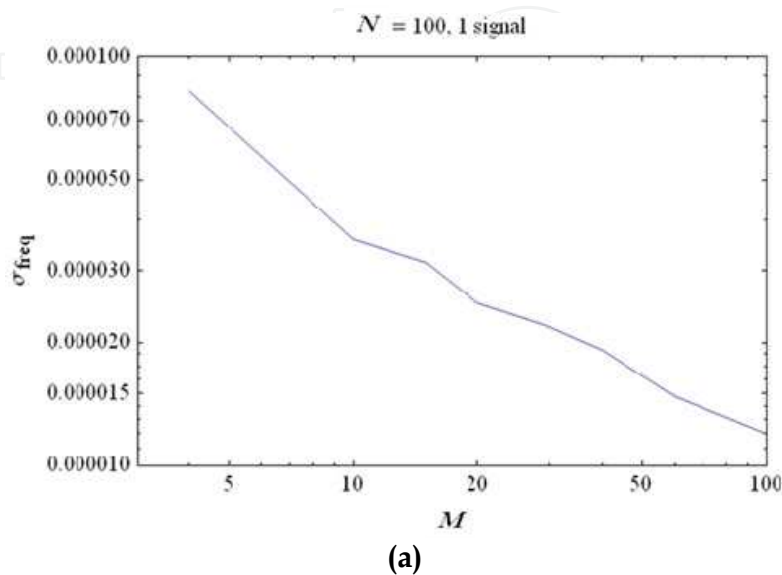


Fig. 6. Fraction of failed recoveries as a function of the small dimension of the mixing matrix  $M$ . The red curve is with the weighting matrix defined by eq. (7). The green curve has the weighting matrix set to the identity matrix.

## 5. Results for sparse sinusoids with noise

### 5.1 Signal composed of a single sinusoid with noise

Figs. 7 (a) and (b) show the error in the recovery of a single-frequency, unity amplitude signal as a function of the small dimension  $M$  of an  $M \times 1024$  mixing matrix  $\Phi$  with  $\sigma = 10^{-2}$  for 100 realizations of the noise. As  $M$  increases the standard deviations of the errors in both frequency and amplitude,  $\sigma_f$  and  $\sigma_a$ , decrease as expected since more measurements are made to average a given noise level. The decrease of about a factor of 3 in  $\sigma_f$  and  $\sigma_a$  for a factor of 10 increase in  $M$  is consistent with estimates based on SNR (Shaw and Valley, 2010; Davenport et al., 2006). Fig. 8 shows  $\sigma_f$  and  $\sigma_a$  as a function of  $s$  averaged over 20 different  $4 \times 1024$  mixing matrices. Both  $\sigma_f$  and  $\sigma_a$  are proportional to  $\sigma$  with  $\sigma_a$  about 2 to 3 orders of magnitude larger than  $\sigma_f$ .



(a)

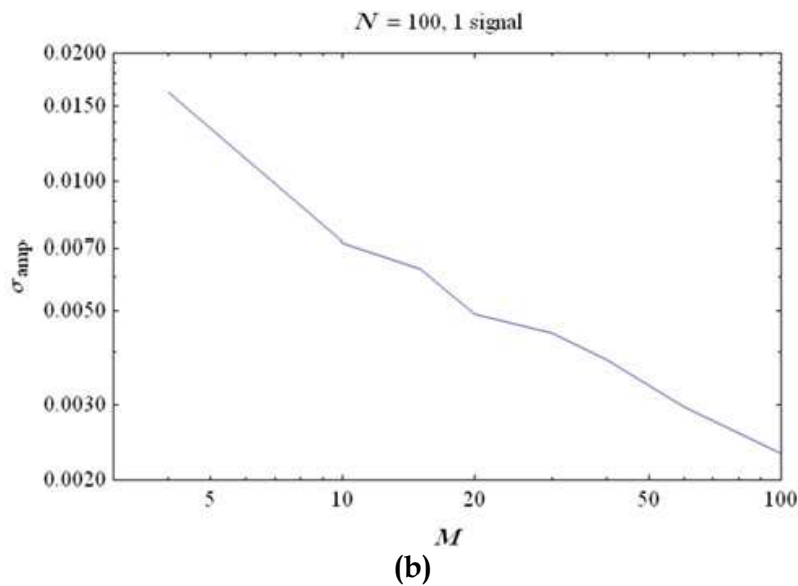


Fig. 7. Standard deviation of the errors in frequency and amplitude of sinusoids mixed by a mixing matrix  $\Phi$  with dimensions  $M \times 1024$  recovered using OMP/NLS as a function of the small dimension  $M$  of the mixing matrix  $\Phi$  for  $\sigma = 10^{-2}$ . The results are obtained from the average of 100 independent calculations. (a) Frequency, (b) amplitude error.

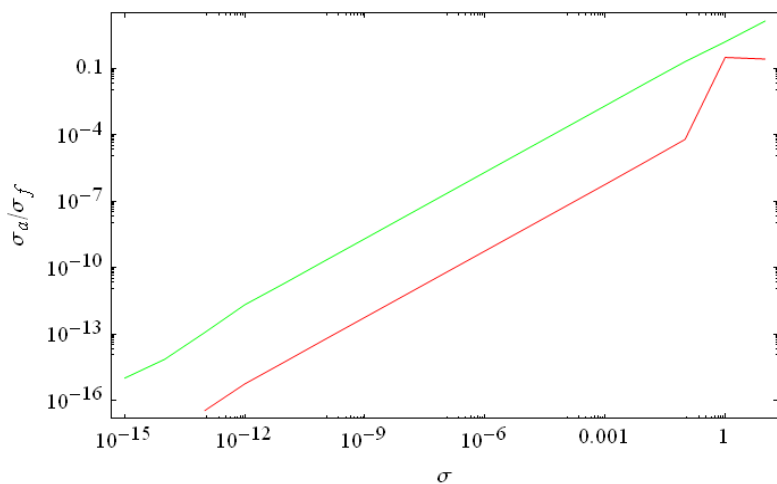


Fig. 8. Standard deviation of the frequency and amplitude errors,  $\sigma_f$  (lower red curve) and  $\sigma_a$  (upper green curve), as a function of  $\sigma$  averaged over 20 different  $4 \times 1024$  mixing matrices.

5.2 Signal composed of 2 sinusoids with 100:1 dynamic range

Noise also affects the ability of our algorithm to recover a small signal in the presence of a large signal. Figs. 9 and 10 show  $\sigma_f$  and  $\sigma_a$  for a test case in which the amplitudes are given by  $\{1.0, 0.01\}$ ,  $M = 10$ ,  $N = 1024$  and the frequencies are well separated. These results are for a single realization of the mixing matrix and averaged over 20 realizations of the noise. Note that as expected, the frequency and amplitude of the large-amplitude component are much better recovered than those of the small-amplitude component. Knowledge of the parameters of the small component essentially disappears for  $\sigma$  greater than about 0.005. Tests with the small amplitude equal to 0.001 and 0.0001 suggest that this threshold scales with the amplitude of the small signal.

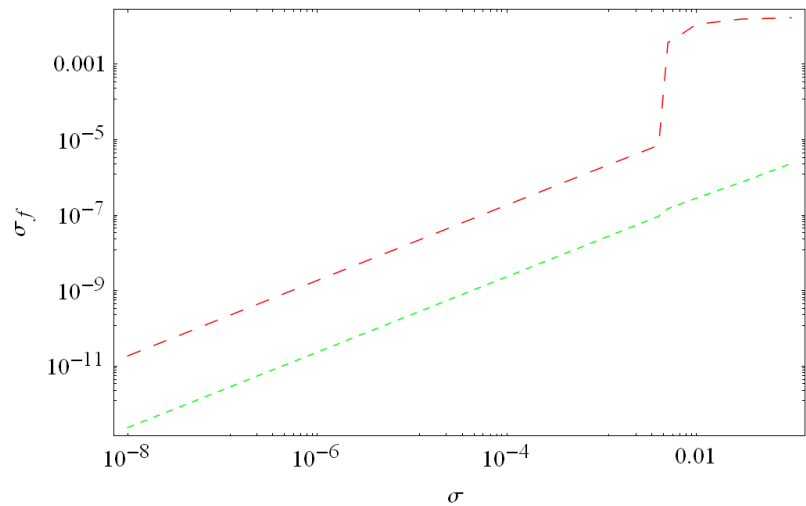


Fig. 9. Standard deviation of the error in the recovered frequency  $\sigma_f$  as a function of noise standard deviation  $\sigma$  for an input signal that consists of two complex sinusoids with amplitudes 1 and 0.01. The green, short dashed curve corresponds to the strong signal; the red, long dashed, to the weak signal. Each curve is averaged over 20 realizations of the noise.

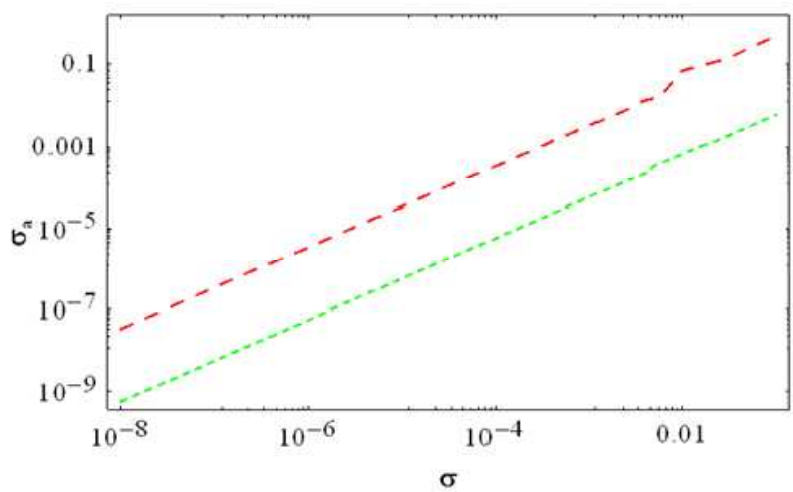


Fig. 10. Standard deviation of the amplitude error  $\sigma_a$  as a function of noise standard deviation  $\sigma$  for an input signal that consists of two complex sinusoids with amplitudes 1 and 0.01. The green, short dashed curve corresponds to the strong signal; red, long dashed, to the weak signal. Each curve is averaged over 20 realizations of the noise.

**5.3 Signal composed of 2 sinusoids with closely spaced frequencies in noise**

We have also investigated the ability of our algorithm to separate two closely spaced frequencies in the presence of noise. Fig. 11 shows  $\sigma_f$  and  $\sigma_a$  for the case with input frequencies  $\{0.3389, 0.3390\}$ , unity amplitude and a  $16 \times 1024$  mixing matrix. Note that significant amplitude error occurs at  $\sigma > 10^{-4}$  compared to the single frequency results. The frequencies are roughly correct but are not separated for  $\sigma > 10^{-2}$ .

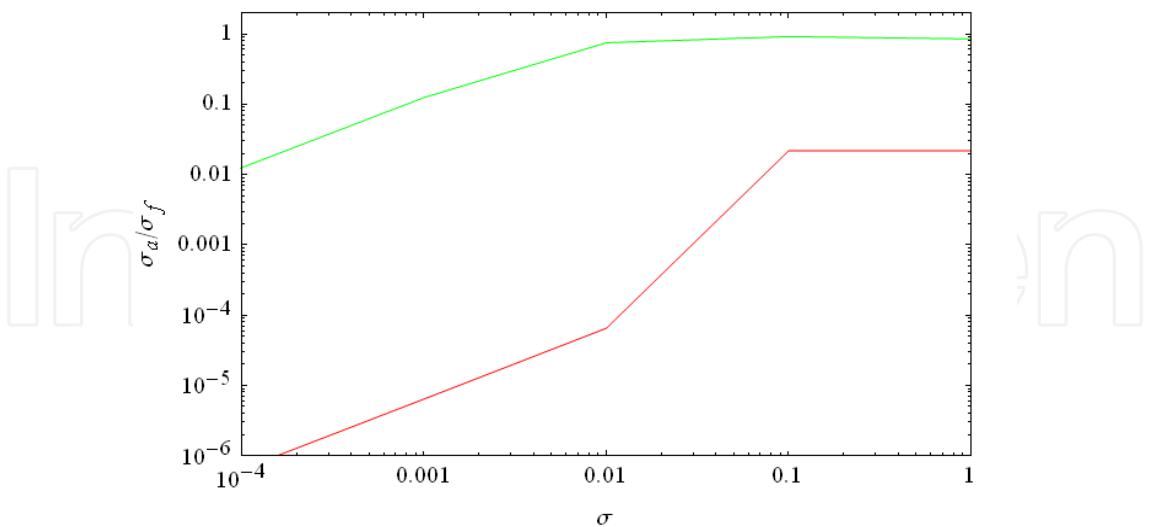


Fig. 11. Standard deviation in frequency  $\sigma_f$  (red-lower curve) and amplitude  $\sigma_a$  (green upper curve) for the case with input frequencies {0.3389, 0.3390}, unity amplitude and a 16x1024 mixing matrix.

6. Comparison with other recovery methods

In this section we compare our version of OMP with an NLS optimization step for the sinusoid frequency and amplitude at each iteration to two common methods for CS recovery: OMP with a linear least squares amplitude estimator at each iteration and convex optimization based on the ell-1 norm of the sparse target vector plus the ell-2 norm of the measurement constraint given by eq. (2). It should be noted that most of the cases presented in the previous sections cannot be solved with OMP/LS or penalized ell-1 norm methods so it is necessary to pick a special case to even perform the comparison. Consider a noise-free signal that consists of 5 unity amplitude sinusoids at 5 different frequencies. We assume  $N=1024$  time samples and an  $M=30 \times N=1024$  complex measurement matrix made up of the sum of random reals plus  $i$  times different random reals, both sets of reals uniformly distributed between -1 and 1.

6.1 Baseline case OMP-NLS

We performed 100 different calculations with the frequencies chosen by a pseudo-random number generator. In order to control the number of significant figures, we took the frequencies from rational numbers uniformly distributed between 0 and 1 in steps of  $10^{-6}$ . Table 2 shows the fraction of failed recoveries and the average standard deviation in the value of the recovered frequency as a function of the oversampling ratio.

6.2 OMP with Linear Least Squares

We performed the same 100 calculations using conventional OMP in which the NLS step is replaced by LS as in Tropp and Gilbert (2007). Note that the number of failed recoveries is

about the same as the baseline OMP-NLS but the frequency error is huge by comparison. This is the natural result of the frequency grid, which is the limit on the OMP resolution. Timing comparisons with our software show that OMP-NLS takes about 50% longer than conventional OMP. We have also windowed the OMP calculations in order to reduce „spectral leakage“ and hopefully achieve better performance. Aside from the lowered failure fraction for  $N_f = 2$ , windowing OMP appears to have no statistically significant effect.

Method\ $N_f$	1	2	4	8
OMP with NLS	95	41	11	6
OMP	96	35	11	6
OMP with window	93	19	9	10

(a)

Method\ $N_f$	1	2	4	8
OMP with NLS	$3.9 \cdot 10^{-15}$	$3.9 \cdot 10^{-15}$	$3.5 \cdot 10^{-15}$	$3.7 \cdot 10^{-15}$
OMP	0.000150	0.000136	0.000085	0.000060
OMP with window	0.000168	0.000141	0.000084	0.000059

(b)

Table 2. Comparing OMP with NLS to OMP and OMP with windowing for 4 values of the overcomplete dictionary  $N_f = 1,2,4,8$ . (a) failure fraction, %. (b) rms error in recovered frequencies.

We have also compared windowed OMP to OMP/NLS in the presence of noise. Fig. 12 shows the frequency and amplitude errors,  $\sigma_f$  and  $\sigma_a$ , as a function of the noise standard deviation  $\sigma$  for OMP (blue) and OMP-NLS (red) for a signal composed of two sinusoids with  $N = 128$ ,  $M = 20$  and  $N_f = 4$  averaged over 100 trials with randomly chosen input frequencies. Note that the OMP frequency error drops to an asymptote of about  $6 \times 10^{-4}$  and the OMP amplitude error to about 0.23 for  $\sigma < 0.1$  while the OMP-NLS errors continue to drop linearly proportional to  $\sigma$  for  $\sigma < 0.1$ .

6.3 Convex optimization

We have performed the same calculations with a penalized ell-1 norm code (Loris, 2008). None of these calculations returns reliable estimates of frequencies off the grid. Windowing helps recover frequencies slightly off the grid but not arbitrary frequencies. Subdividing the frequency grid allows finer resolution in the recovery but only up to the fine frequency grid.



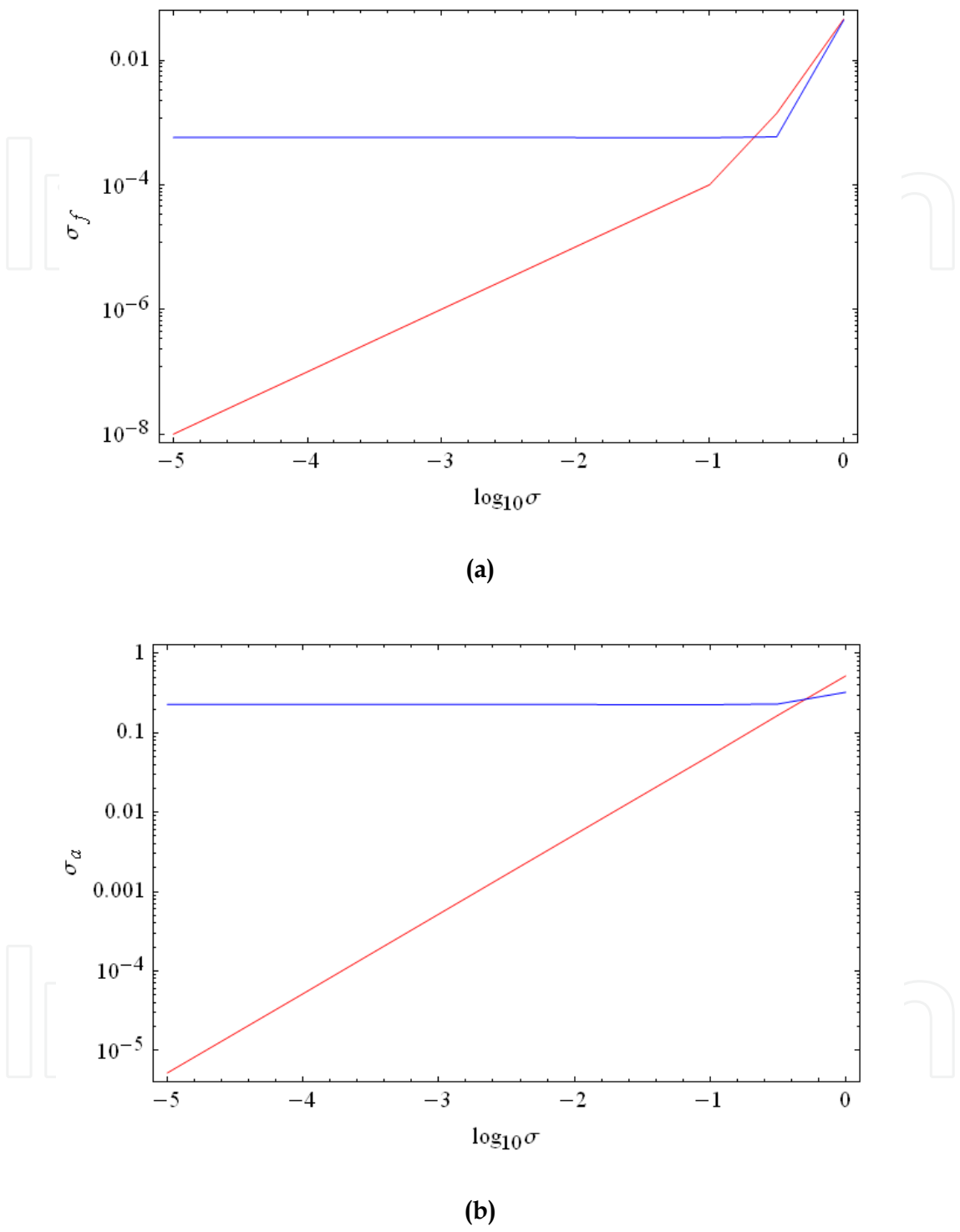


Fig. 12. Frequency and amplitude errors,  $\sigma_f$  and  $\sigma_a$ , as a function of the noise standard deviation  $\sigma$  for OMP (blue) and OMP-NLS (red) for a signal composed of a two sinusoids with  $N = 128$ ,  $M = 20$  and  $N_f = 4$  averaged over 100 trials with randomly chosen input frequencies. (a) Frequency error. (b) Amplitude error.

The  $\ell_1$  norm code used in our studies (Loris, 2008) can be used with the frequency grid subdivided by 8 or more, but the results are not sparse for the test case described above. More frequencies are returned than in the input signal. Good approximations (consistent with the OMP estimates) can be obtained by precisely thresholding the recovered vector  $s$  in eq. (2), but the threshold is dependent on the oversampling ratio and on the random seed used to generate the frequencies.

## 7. Performance estimates

As discussed above, this study is based on experimental or empirical evaluation (i.e. numerical simulations) of a proposed technique for recovering compressively sensed signals. The weakness of such a study is that calculations alone do not provide performance guarantees while the strength of such a study is that calculations can evaluate practical cases that would be encountered in real applications. Regardless, it is necessary to know when and how an algorithm fails for it to be of much use, and we can use prior work on performance guarantees for CS, OMP and NLS to help us.

Consider first the noise-free case in which the number of sinusoids is known. Here the difference between success and failure is computationally obvious. If the recovery is successful, the residual after extraction of the known number of sinusoids collapses to near the machine precision. If it fails, the residual remains at about the level of the initial measurement vector  $y$ . In the presence of noise the situation is similar except the collapse is to the system noise level. If the number of sinusoids is unknown, then recovery proceeds until the system noise level is reached, but statistical testing must be used to determine if the residual at this threshold is due to noise or incorrectly recovered signals.

Practical use of the OMP/NLS algorithm requires at a minimum empirical knowledge of where the algorithm fails and ultimately, performance guarantees and complexity estimates (operation counts). Since this algorithm couples two well known algorithms, in principle we can rely on previous work. The problem can be divided into 3 parts. First, one has to assess the compressive sensing part of the problem. Does the mixing matrix  $\Phi$  satisfy the appropriate conditions? Is the value of  $M$  large enough to recover the  $K$  unknowns? Are the measurements really sparse in the chosen model or even is the model applicable to the signal of interest? Our empirical observations suggest that it is difficult for a random number generator to pick a bad mixing matrix. Observations also suggest that the requirement on  $M$  for recovery is on the same order as that derived for grid-based CS,  $M \sim K \log(N/K)$ . Second, the sampling in the overcomplete dictionary must be fine enough that the first frequency found by the argmax of  $G(f, r)$  in (7) is near a true solution. If this is not the case due to insufficient granularity, multiple frequencies too close together, or high noise levels, the OMP cannot start. This issue is not restricted to our work but common to all matching pursuit algorithms. While we do not have performance guarantees here, we have noted empirically that lack of convergence is very easy to determine for a known number or sinusoids and known noise floor. Finally, the NLS must be able to converge. Here we can rely on the results given by (Stoica et al., 2000; Li et al., 2000; Chan and So, 2004; Christensen and Jensen, 2006) that the NLS achieves the Cramer Rao Bound. Empirically, we observe that the dictionary must be sufficiently overcomplete that the NLS is looking for a frequency solution in one local minimum.

## 8. Conclusion

The work reported in this chapter started with our work on compressive sensing for direction of arrival (DOA) detection with a phased array (Shaw and Valley, 2010). In that work, we realized that most work in compressive sensing concerned recovering signals on a sparse grid. In the DOA domain, that meant that targets had to be on a set of grid angles, which of course never happens in real problems. We found a recovery solution for a single target in that work by scanning the sparsifying DFT over an offset index that was a measure of the sine of the target angle but the solution was time consuming because the penalized  $\ell_1$  norm recovery algorithm had to be run multiple times until the best offset and best sparse solution was found and the procedure was not obviously extendable to multiple targets. This work led us to the concepts of orthogonal matching pursuit and removing one target (or sinusoid) at a time. But we still needed a reliable method to find arbitrary frequencies or angles not on a grid. The next realization was that nonlinear least squares could be substituted for the linear least squares used in most versions of OMP. This has proved to be an extremely reliable method and we have now performed 10's of thousands of calculations with this method. Since OMP is not restricted to finding sinusoids, it is natural to ask if OMP with NLS embedded in it works for other functions as well. We have not tried to prove this generally, but we have performed successful calculations using OMP-NLS with signals composed of multi-dimensional sinusoids such as would be obtained with 2D phased arrays (see also Li et al., 2001), signals composed of multiple sinusoids multiplied by chirps (i.e. sums of terms of the form  $a_k \exp(i\omega_k t + b_k t^2)$ ) and multiple Gaussian pulses within the same time window.

## 9. Acknowledgment

This work was supported under The Aerospace Corporation's Independent Research and Development Program.

## 10. References

- Baraniuk, R. G.; (2007). Compressive sensing, *IEEE Signal Processing Mag.*, Vol.24, No.4, pp.118-120, 124.
- Candes, E. J.; & Tao, T., (2006). Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inform. Theory*, Vol.52, pp. 5406-5425.
- Candes, E. J.; & Wakin, M. B., (2008). An introduction to compressive sampling, *IEEE Signal Processing Magazine*, Vol.21, pp. 21-30.
- Candes, E. J.; Eldar, Y. C., Needell, D., & Randall, P., (2011). Compressed sensing with coherent and redundant dictionaries, submitted to *Applied and Computational Harmonic Analysis*.
- Chan, K. W.; & So, H. C., (2004). Accurate frequency estimation for real harmonic sinusoids, *IEEE Signal Processing Lett.*, Vol.11, No.7, pp. 609-612.
- Christensen, M. G.; & Jensen, S. H., (2006). On perceptual distortion minimization and nonlinear least-squares frequency estimation, *IEEE Trans. Audio, Speech, Language Processing*, Vol.14, No.1, pp. 99-109.

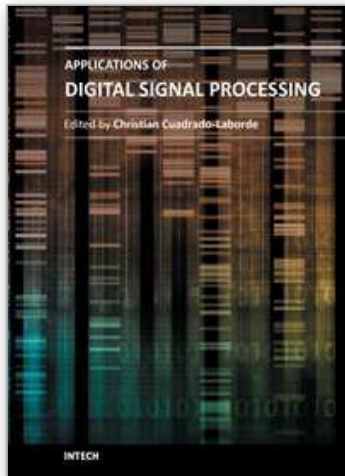
- Davenport, M.; Wakin, M., and Baraniuk, R., (2006). Detection and estimation with compressive measurements, Rice ECE Department Technical Report, TREE 0610.
- Davis, G.; Mallat, S., & Avellaneda, M., (1997). Greedy adaptive approximation, *Const. Approx.*, Vol.13, pp. 57-98.
- Donoho, D. L.; (2006). Compressed sensing, *IEEE Trans. Inform. Theory*, Vol.52, pp. 1289-1306, Sept. 2006.
- Duarte, M. F.; & Baraniuk, R. G., (2010). Spectral Compressive Sensing, submitted to *IEEE Trans. Signal Processing*.
- Hormati, A.; & M. Vetterli, (2007). Annihilating filter-based decoding in the compressed sensing framework, *Proc. SPIE*, Vol.6701, pp. 1-10.
- Huang, S.; & J. Zhu, (2011). Recovery of sparse signals using OMP and its variants: convergence analysis based on RIP, *Inverse Problems*, Vol.27, 035003 (14pp).
- Jacques, L.; & C. De Vleeschouwer, (2008). A Geometrical study of matching pursuit parameterization, *IEEE Trans. Signal Proc.*, Vol.56, pp. 2835-2848.
- Li, H.; Stoica, P., & Li, J., (2000). Computationally efficient parameter estimation for harmonic sinusoidal signals, *Signal Processing*, Vol.80, pp. 1937-1944.
- Li, H.; Sun, W., Stoica, P. & Li, J., (2001). 2-D sinusoidal amplitude estimation with application to 2-D system identification, *IEEE International Conf. On Acoustics, Speech, and Signal Proc.*, ICASSP, Vol.3, pp. 1921-1924.
- Loris, I.; (2008). L1Packv2: A *Mathematica* package for minimizing an ell-1-penalized functional, *Computer Phys. Comm.*, Vol.79, pp. 895-902.
- Mallat, S. G.; & Zhang, Z., (1993). Matching pursuits with time-frequency dictionaries, *IEEE Trans. Signal Processing*, Vol.41, No.12, pp. 3397-3415.
- Needell, D.; & Tropp, J. A., (2009). CoSaMP: Iterative signal recovery from incomplete and inaccurate samples, *Applied and Computational Harmonic Analysis*, Vol.26, No.3, pp. 301-321.
- Needell, D.; & Vershynin, R., (2009). Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit, *Found. Comput. Math.*, Vol.9, pp. 317-334.
- Pati, Y.C.; Rezaifar, R., & Krishnaprasad, P.S., (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition, *Proc. 27th Annu. Asilomar Conf. Signals. Systems, and Computers*, Pacific Grove, CA. Vol.1, pp. 40-44.
- Salzo, S.; & Villa, S., (2011). Convergence analysis of a proximal Gauss-Newton method, *arXiv:1103.0414v1*, pp. 1-29.
- Shaw, T. J.; & Valley, G. C., (2010). Angle of arrival detection in sparse systems using compressed sensing, *European Signal Processing Conference (EUSIPCO)*, Aalborg, Denmark, 23-27Aug. 2010, *EUSIPCO 2010 Digest*, pp. 1424-1428.
- Stoica, P.; & Moses, R. L., (1997). *Introduction to Spectral Analysis*, Upper Saddle River, NJ: Prentice Hall, pp. 146-151.
- Stoica, P.; Li, H., and Li, J. (2000). Amplitude estimation of sinusoidal signals: survey, new results, and an application, *IEEE Trans. Signal Processing*, Vol.48, No.2, pp. 338-352.

- Tropp, J. A.; & Gilbert, A. C., (2007). Signal recovery from random measurements via orthogonal matching pursuit, *IEEE Trans. Information Theory*, Vol.53, No.12, pp. 4655-4666.
- Vetterli, M.; Marziliano, P., & Blu, T., (2002). Sampling signals with finite rate of innovation, *IEEE Trans. Signal Processing*, Vol.50, No.6, pp. 1417-1428.

IntechOpen

IntechOpen





## **Applications of Digital Signal Processing**

Edited by Dr. Christian Cuadrado-Laborde

ISBN 978-953-307-406-1

Hard cover, 400 pages

**Publisher** InTech

**Published online** 23, November, 2011

**Published in print edition** November, 2011

In this book the reader will find a collection of chapters authored/co-authored by a large number of experts around the world, covering the broad field of digital signal processing. This book intends to provide highlights of the current research in the digital signal processing area, showing the recent advances in this field. This work is mainly destined to researchers in the digital signal processing and related areas but it is also accessible to anyone with a scientific background desiring to have an up-to-date overview of this domain. Each chapter is self-contained and can be read independently of the others. These nineteenth chapters present methodological advances and recent applications of digital signal processing in various domains as communications, filtering, medicine, astronomy, and image processing.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

George C. Valley and T. Justin Shaw (2011). Applications of the Orthogonal Matching Pursuit/ Nonlinear Least Squares Algorithm to Compressive Sensing Recovery, Applications of Digital Signal Processing, Dr. Christian Cuadrado-Laborde (Ed.), ISBN: 978-953-307-406-1, InTech, Available from:

<http://www.intechopen.com/books/applications-of-digital-signal-processing/applications-of-the-orthogonal-matching-pursuit-nonlinear-least-squares-algorithm-to-compressive-sen>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen