Compressive Parameter Estimation for Sparse Translation-Invariant Signals Using Polar Interpolation

Fyhn, Karsten; Duarte, Marco F. ; Jensen, Søren Holdt

Published in:
I E E E Transactions on Signal Processing

DOI (link to publication from Publisher):
10.1109/TSP.2014.2385035

Publication date:
2015

Document Version
Early version, also known as pre-print

Compressive Parameter Estimation for Sparse Translation-Invariant Signals Using Polar Interpolation

Karsten Fyhn, *Student Member, IEEE,* Marco F. Duarte, *Member, IEEE,*

and Søren Holdt Jensen, *Senior Member, IEEE*

Abstract

We propose new compressive parameter estimation algorithms that make use of polar interpolation to improve the estimator precision. Moreover, we evaluate six algorithms for estimation of parameters in sparse translation-invariant signals, exemplified with the time delay estimation problem. The evaluation is based on three performance metrics: estimator precision, sampling rate and computational complexity. We use compressive sensing with all the algorithms to lower the necessary sampling rate and show that it is still possible to attain good estimation precision and keep the computational complexity low. The proposed algorithms are based on polar interpolation and our numerical experiments show that they outperform existing approaches that either leverage polynomial interpolation or are based on a conversion to an frequency-estimation problem followed by a super-resolution algorithm. The algorithms studied here provide various tradeoffs between computational complexity, estimation precision and necessary sampling rate. The work shows that compressive sensing for the class of sparse translation-invariant signals allows for a lower sampling rate and that the use of polar interpolation increases the estimation precision.

Index Terms

Compressive sensing, translation-invariant signals, interpolation, time delay estimation.

I. Introduction

Compressive sensing (CS) is a technique to simultaneously acquire and reduce the dimensionality of sparse signals in a randomized fashion. More precisely, in the CS framework, a signal $\mathbf{f} \in \mathbb{C}^N$ is sampled by M linear measurements of the form $\mathbf{y} = \mathbf{A}\mathbf{f}$, where $\mathbf{A} \in \mathbb{C}^{M \times N}$ is a sensing matrix and $M < N$. In practice, the measurements are acquired in the presence of additive signal and measurement noise $\mathbf{n}$ and $\mathbf{w}$, respectively, in which case we have $\mathbf{y} = \mathbf{A}(\mathbf{f} + \mathbf{n}) + \mathbf{w}$.

In many applications, the signal $\mathbf{f}$ is not sparse but has a sparse representation in some dictionary $\mathbf{D} \in \mathbb{C}$. In other words, we have $\mathbf{f} = \mathbf{D}\mathbf{x}$, where $\mathbf{x} \in \mathbb{C}$ is K-sparse (i.e. $\|\mathbf{x}\|_0 \leq K$). Under certain conditions on the matrix $\mathbf{A}$ [1], [2], we can

The authors would like to thank Dr. Tobias Lindstrøm Jensen of Aalborg University for many helpful discussions about the convex optimization problem formulation. This work is supported by The Danish Council for Strategic Research under grant number 09-067056, Danish Center for Scientific Computing and by a EliteForsk travel scholarship under grant number 11-116371. K. Fyhn and S. H. Jensen are with Dept. of Electronic Systems, Aalborg University, Denmark. E-mail: {kfn,shj}@es.aau.dk. M. F. Duarte is with Dept. of Electrical and Computer Engineering, University of Massachusetts Amherst, USA. E-mail: mduarte@ecs.umass.edu.

recover $\mathbf{x}$ from the measurements $\mathbf{y}$ through the following ℓ_1-minimization problem (which we refer to as ℓ_1-*synthesis*):

$$\hat{\mathbf{x}} = \min_{\tilde{\mathbf{x}} \in \mathbb{C}^N} \|\tilde{\mathbf{x}}\|_1 \quad \text{s.t.} \quad \|\mathbf{A}\mathbf{D}\tilde{\mathbf{x}} - \mathbf{y}\|_2 \leq \epsilon, \tag{1}$$

where ϵ is an upper bound on the noise level $\|\mathbf{A}\mathbf{n} + \mathbf{w}\|_2$. Note that optimal recovery of $\mathbf{x}$ from the optimization in Eqn. (1) is guaranteed only when the elements of the dictionary $\mathbf{D}$ form an orthonormal basis, and thus are incoherent [3], [4]. However, in many applications, the signal of interest is sparse in an overcomplete dictionary or a frame, rather than in a basis.

Classic CS requires sparsity in some matrix dictionary to work, but in many cases a signal may be sparse with respect to some parametric model instead. Previous work has shown that CS may experience problems in such cases, when using the traditional dictionary-based approach [5]. One such class of signals is sparse translation-invariant signals. Here, translation invariance or translation symmetry refers to the Euclidian norm of the signal, which must remain the same after translation. Let $\mathbf{g}(b_i) = \mathcal{M}_{\mathbf{g}}(b_i),\ \mathbf{g}(b_i) \in \mathbb{C}^N$ denote a point in the signal manifold $\mathcal{M}_{\mathbf{g}}(\cdot)$ parameterized by a translation parameter b_i. A function $\mathbf{g}(b_i)$ is translation-invariant if it fulfills two requirements: 1) preservation of the ℓ_2-norm under translation, $\|\mathbf{g}(b_1)\|_2 = \|\mathbf{g}(b_2)\|_2, \forall b_1, b_2$, and 2) locally constant (symmetrical) curvature of the signal manifold, $\|\mathbf{g}(b - \mathbf{g}(b - \Delta))\|_2 = \|\mathbf{g}(b - \mathbf{g}(b + \Delta))\|_2, \forall b$, where Δ is some sufficiently small change in the parameter.

Two examples of estimation problems with translation-invariant signals are Time Delay Estimation (TDE) and Frequency Estimation (FE). TDE of one or more known signal waveforms from sampled data is of interest in several fields such as radar, sonar, wireless communications, audio, speech and medical signal processing. The TDE problem is often defined as receiving a known signal with an unknown delay and amplitude coefficient that must be estimated. Similarly, FE concerns the estimation of the frequency components of a received sum of exponentials, which is of interest in seismology, audio, speech and music processing, radar and sonar.

For this type of estimation problems there can be different parameters and performance metrics. In this work, we focus on three important performance metrics: estimation precision, computational complexity, and necessary sampling rate to acquire the analog signal. We use CS to lower the necessary sampling rate while still providing good estimation precision. The algorithms we evaluate vary in computational complexity and, not surprisingly, the most computationally heavy algorithms perform the best. In some cases the difference between the best and worst algorithms' estimation precision performance is four orders of magnitude, while the computational complexity is two orders of magnitude larger. It follows that this becomes a design trade-off for individual problems.

We propose two algorithms that leverage *polar interpolation* to improve the estimation precision. Interpolation is necessary because of the required discrete dictionary in CS systems. With a dictionary matrix, we assume the delay or frequency parameter takes values from a finite set only:

$$\mathbf{D} = \begin{bmatrix} \mathbf{g}(b_1) & \mathbf{g}(b_2) & \cdots & \mathbf{g}(b_N) \end{bmatrix}. \tag{2}$$

In reality, the parameter is drawn from a continuous interval. One way to overcome this is to increase the number of atoms in the dictionary; however, this increases the coherence. Instead, the proposed algorithms feature a dictionary that can sparsely represent any sparse translation-invariant signal with a parameter drawn from a continuous interval.

In a recent paper, it was shown how polar interpolation may be utilized for FE in the case where the amplitude coefficients are real and non-negative [6]. In this paper, we will show that if both positive and negative complex amplitude coefficients are allowed, the coherence introduced by the FE dictionary to enable interpolation does not allow for a unique, sparse solution. We postulate that this broader problem may be solved with stronger constraints or a different convex optimization formulation, but we do not focus on this extension. Instead we have our main focus on the use of interpolation and CS to solve the TDE problem where such coherence is not an issue.

The contribution of this paper consists of mainly two points: 1) Two proposed algorithms that outperform other TDE algorithms, but may also be used for other sparse translation-invariant signals and 2) An evaluation of different TDE algorithms' performance when coupled with CS.

In the next section, we present previous work in the area of interpolation between dictionary elements and TDE estimation on a continuous parameter space. In Section III we review the polar interpolation technique and introduce an advanced convex optimization formulation to handle coefficient vectors that are not real and non-negative. This is followed by Section IV in which we introduce an iterative, greedy algorithm based on interpolation. In Section V we evaluate the proposed algorithms and compare them to other state-of-the-art TDE algorithms. We investigate their performance for well-spaced pulses and for overlapping pulses, and we evaluate the estimators' performance under varying levels of measurement noise and signal noise. Finally, Section VI concludes the paper.

II. Previous Work

Prior work on the problem of sparsity in parametric dictionaries includes [7], [8], which use a gradient descent approach to approximate solutions off the grid for a generic greedy algorithm. Another common method is to use parabolic or polynomial interpolation on a sampled autocorrelation function to increase the precision for sampled data [9]–[11]. The simplest and most often used polynomial interpolation is fitting a parabola around the correlation peak. In [10] it is proposed to use a Direct Correlator function for parabolic interpolation:

$$b_i = -\frac{\Delta}{2}\frac{\hat{R}_{\mathbf{f}}[n+1] - \hat{R}_{\mathbf{f}}[n-1]}{\hat{R}_{\mathbf{f}}[n+1] - 2\hat{R}_{\mathbf{f}}[n] + \hat{R}_{\mathbf{f}}[n-1]} + n\Delta,$$

where b_i is the translation parameter to estimate, Δ is the spacing in time between samples of the discrete autocorrelation function:

$$\hat{R}_{\mathbf{f}}[m] = \sum_{l=1}^{N} \mathbf{f}[l] \cdot \mathbf{f}[l-m], \tag{3}$$

and n is the index of the largest absolute entry in $\hat{R}_{\mathbf{f}}$. This estimator is easily implemented in a greedy algorithm, where an estimate of the discrete autocorrelation is readily available as the signal proxy. In some cases, it is possible to improve the estimation using different polynomial interpolation techniques for different problems, see, e.g., the references in [12]. Interpolation-based algorithms improve the estimation precision but suffers from interference problems if the signal components are not orthogonal to each other. The polynomial interpolation approach is similar to one of the two algorithms proposed in [13], one using a first-order Taylor expansion, the other a form of polar interpolation. The authors show that polar interpolation

outperforms Taylor expansion. In our work we extend upon the polar interpolation approach. In [14]–[16], the authors use coherence rejection to better estimate a solution. Additionally, [15] uses polynomial interpolation. In [16] the coherence rejection is implemented as functions that inhibit coherent atoms in the recovery algorithms. This function is used in greedy algorithms to trim the proxy before selecting the strongest correlating atom. Based on the coherence between a subset S of atoms from the dictionary and its complement, we can define the η-coherence band of the index set S as

$$B_\eta(S) = \bigcup_{k \in S} \{i \mid \mu(i,k) > \eta\},\ i \in \{1, 2, \ldots, P\}, \tag{4}$$

where $\mu(i,k) = |\langle \mathbf{g}(b_i), \mathbf{g}(b_k) \rangle|$ is the coherence between two atoms, $\mathbf{g}(b_i)$ and $\mathbf{g}(b_k)$ in the dictionary $\mathbf{D}$. The authors use the band exclusion function to avoid selecting coherent dictionary elements in various greedy algorithms. When applied to the Orthogonal Matching Pursuit (OMP) algorithm, the resulting enhanced algorithm is called Band-excluded Orthogonal Matching Pursuit (BOMP) [16].

Another approach to time delay estimation is to use FFT-based methods, where the problem is converted to a frequency estimation problem and solved using line spectral estimation approaches such as the Multiple Signal Classification (MUSIC) algorithm [17] or the Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) algorithm [18]. This approach exploits the fact that the dictionary matrix is cyclic. In [19], the TDE problem is converted to an FE problem and solve it by means of the ESPRIT algorithm. This is done by pre-multiplying the matrix product $\mathbf{G}^{-1}\mathbf{F}$ on the received signal vector, $\mathbf{f} = \mathbf{D}\mathbf{x}$:

$$\mathbf{y} = \mathbf{G}^{-1}\mathbf{F}\mathbf{f} = \mathbf{G}^{-1}\mathbf{F}\mathbf{D}\mathbf{x} = \mathbf{G}^{-1}\mathbf{G}\mathbf{F}\mathbf{x} = \mathbf{F}\mathbf{x}, \tag{5}$$

where $\mathbf{G}$ is a diagonal matrix with the Fourier transform of the first column of $\mathbf{D}$ on the diagonal and zero elsewhere and $\mathbf{F}$ is the DFT matrix. Because $\mathbf{D}$ is a cyclic matrix, it is diagonalized by the DFT matrix, i.e. $\mathbf{F}\mathbf{D} = \mathbf{G}\mathbf{F}$. Then $\mathbf{y}$ contains a sum of exponentials and we may then use a super-resolution algorithm to estimate the frequencies, which can be directly mapped to delays. However, this method has certain pitfalls. As mentioned in [20], [21] the spectrum of the pulse in $\mathbf{G}$ must be nonzero everywhere and the noise can no longer be assumed white, due to the multiplication with the inverse of the known spectrum. The signal used in this work spans the entire spectrum in which we sample and therefore does not suffer from the first problem. The noise will be colored, but in our numerical experiments this does not seem to decrease the performance much.

A similar method has also been implemented using analog filters and using CS in [22]; however, it has limitations similar to those of the approach in Eqn. (5). The method relies on filters that are tailored to the Fourier transform of the signal, similarly to the $\mathbf{G}$ matrix used in the above. These filters must be stably invertible, which becomes a problem if the spectrum is zero or close to zero at some frequencies. Furthermore, these filters must also result in a coloring of the noise. The method in [19] may also be used with CS by first reconstructing the signal using e.g. ℓ_1 synthesis as in Eqn. (1) followed by estimation.

III. Polar Interpolation

One way to remedy the discretization of the parameter space implicit in CS is to use interpolation. In [13], a *polar interpolation* approach for translation-invariant signals has been derived. Such signals can be written as a linear combination of shifted versions of a waveform. In a nutshell, the interpolation procedure exploits the fact that translated versions of a waveform form a manifold which lies on the surface of a hypersphere. Thus, any sufficiently small segment of the manifold can be well-approximated by an arc of a circle, and an arbitrarily-shifted waveform can be accurately approximated by a point in one such arc connecting dictionary elements.

Define the signals of interest as:

$$\mathbf{f}(\mathbf{a},\mathbf{b}) = \sum_{k=1}^{K} a_k \mathbf{g}(b_k), \tag{6}$$

where K is the number of signal components, $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & \dots & a_K \end{bmatrix} \in \mathbb{C}^{1\times K}$ is a vector of complex amplitude coefficients, $\mathbf{g}(b)$ is a translation-invariant parametric signal, parameterized by a translation parameter from $\mathbf{b} = \begin{bmatrix} b_1 & b_2 & \dots & b_K \end{bmatrix} \in \mathbb{R}^{1\times K}$.

In this case, the dictionary $\mathbf{D}$ from Eqn. (2) samples the translation parameter space with step size Δ, and we approximate each segment of the manifold $\{\mathcal{M}_{\mathbf{g}}(b_n),\ b_n \in [b_p - \frac{\Delta}{2}, b_p + \frac{\Delta}{2}]\}$ by a circular arc containing the three exponentials $\{\mathbf{g}(b_p - \frac{\Delta}{2}), \mathbf{g}(b_p), \mathbf{g}(b_p + \frac{\Delta}{2})\}$. Making use of trigonometric identities, the polar interpolation approximates the waveform $\mathbf{g}(b_n)$ using the arc containing b_p, where $b_p = [\![b_n]\!] = \text{round}\left(\frac{b_n}{\Delta}\right)\Delta$, so that $b_n = b_p + \Delta_n,\ \Delta_n \in (-\frac{\Delta}{2}, \frac{\Delta}{2})$. Here, $b_p = [\![b_n]\!]$ signifies the selection of the closest atom in the dictionary b_p to the input parameter b_n. This arc is parametrized as follows [13]:

$$\begin{aligned}
\tilde{\mathbf{g}}(b_n) &= \mathbf{c}(b_p) + r\cos\left(\frac{2\Delta_n}{\Delta}\theta\right)\mathbf{u}(b_p) + r\sin\left(\frac{2\Delta_n}{\Delta}\theta\right)\mathbf{v}(b_p), \\
\begin{bmatrix} \mathbf{c}(b_p)^T \\ \mathbf{u}(b_p)^T \\ \mathbf{v}(b_p)^T \end{bmatrix} &= \begin{bmatrix} 1 & r\cos(\theta) & -r\sin(\theta) \\ 1 & r & 0 \\ 1 & r\cos(\theta) & r\sin(\theta) \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{g}(b_p - \frac{\Delta}{2})^T \\ \mathbf{g}(b_p)^T \\ \mathbf{g}(b_p + \frac{\Delta}{2})^T \end{bmatrix},
\end{aligned} \tag{7}$$

where r is the ℓ_2 norm of each element of the dictionary and θ is the angle between $\mathbf{g}(b_p)$ and $\mathbf{g}(b_p - \frac{\Delta}{2})$:

$$\begin{aligned}
r &= \|\mathbf{g}(b_p)\|_2, \\
\theta &= \frac{\text{Re}\{\langle \mathbf{g}(b_p), \mathbf{g}(b_p - \frac{\Delta}{2})\rangle\}}{\|\mathbf{g}(b_p)\|_2 \cdot \|\mathbf{g}(b_p - \frac{\Delta}{2})\|_2}
\end{aligned}$$

for all $p \in \{1, 2, \dots, P\}$. In order to extend the above approximation to include multiple waveforms, we introduce three

dictionaries that sample the parameter space $[\![\Omega_J]\!] = \{[\![b_1]\!], [\![b_2]\!], \ldots, [\![b_J]\!]\}$:

$$\begin{aligned}
\tilde{\mathbf{f}}(\Omega_J) &= \mathbf{C}([\![\Omega_J]\!])\boldsymbol{\alpha} + \mathbf{U}([\![\Omega_J]\!])\boldsymbol{\beta} + \mathbf{V}([\![\Omega_J]\!])\boldsymbol{\gamma}, \\
\mathbf{C}(\Omega_J) &= \begin{bmatrix} \mathbf{c}([\![b_1]\!]) & \mathbf{c}([\![b_2]\!]) & \cdots & \mathbf{c}([\![b_J]\!]) \end{bmatrix} \in \mathbb{C}^{N\times J}, \\
\mathbf{U}(\Omega_J) &= \begin{bmatrix} \mathbf{u}([\![b_1]\!]) & \mathbf{u}([\![b_2]\!]) & \cdots & \mathbf{u}([\![b_J]\!]) \end{bmatrix} \in \mathbb{C}^{N\times J}, \\
\mathbf{V}(\Omega_J) &= \begin{bmatrix} \mathbf{v}([\![b_1]\!]) & \mathbf{v}([\![b_2]\!]) & \cdots & \mathbf{v}([\![b_J]\!]) \end{bmatrix} \in \mathbb{C}^{N\times J},
\end{aligned} \tag{8}$$

where $\boldsymbol{\alpha}$ represents the amplitude of the signal and $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ controls the parameter translations.

A. Simple convex optimization problem

The three coefficient vectors, $\boldsymbol{\alpha}, \boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, can be estimated using the following constrained convex optimization problem from [13], which is a variant of the classical Basis Pursuit Denoising algorithm [23]:

$$\begin{aligned}
(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) &= \mathrm{T}(\mathbf{y}, \Omega_J) \\
&= \underset{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}}{\arg\min} \frac{1}{2\sigma^2}\|\mathbf{y} - \tilde{\mathbf{f}}(\Omega_J)\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \\
&\text{s.t.} \left\{ \begin{array}{c} \alpha_j \geq 0, \\ \sqrt{\beta_j^2 + \gamma_j^2} \leq \alpha_j r, \\ \alpha_j r\cos(\theta) \leq \beta_j \leq \alpha_j r, \end{array} \right\} \text{ for } j = 1, \ldots, J,
\end{aligned} \tag{9}$$

where $\mathbf{y}$ is the received compressed signal and σ^2 is the squared norm of the measurement and signal noise. Here, λ is used as a weighting factor between sparsity and fidelity. The constraints for the optimization problem ensure that the solution consists of points on the arcs used for approximation. The first constraint ensures we have only nonnegative signal amplitudes. The second enforces the trigonometric relationship among each triplet α_j, β_j, and γ_j. The last constraint ensures that the angle between the solution and $\mathbf{g}(b_p)$ is restricted to the interval $[-\theta, \theta]$. It is necessary to scale $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ after the optimization problem [13]:

$$(\beta_j, \gamma_j) \leftarrow \left(\frac{\beta_j \alpha_j r}{\sqrt{\beta_j^2 + \gamma_j^2}}, \frac{\gamma_j \alpha_j r}{\sqrt{\beta_j^2 + \gamma_j^2}} \right), \ \forall j. \tag{10}$$

This is because the inequality of the second constraint should in fact be an equality. However, the equality would violate the convexity assumption of the optimization. After this normalization, we obtain the signal estimate from Eqn. (8) and the frequency estimates using the one-to-one relation:

$$\alpha_n \mathbf{c}(b_p) + \beta_n \mathbf{u}(b_p) + \gamma_n \mathbf{v}(b_p) \approx a_n \mathbf{g}\left(b_p + \tfrac{\Delta}{2\theta} \tan^{-1}(\tfrac{\gamma_n}{\beta_n}) \right), \tag{11}$$

where the argument of $\mathbf{g}(\cdot)$ is the estimate of b_n. The change in index from j to n is because only the K absolute largest entries in $\boldsymbol{\alpha}$ and the corresponding entries in $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are used for estimation, as they represent the active atoms. The authors in [13] have named this algorithm Continuous Basis Pursuit (CBP). However, their formulation assumes non-negative real values for

the amplitude coefficients $\mathbf{a}$, which precludes many real-world settings. Additionally, their choice of fidelity/sparsity trade-off in the convex optimization formulation does not distinguish between noise and approximation error in the polar interpolation. To address these issues, we propose an improved convex optimization formulation in this section.

B. Advanced convex optimization problem

One of the contributions of this paper is an improved convex optimization formulation of Eqn. (9). To achieve this we first introduce a metric for the approximation noise, which is used together with the signal and measurement noise σ^2 as a measure of uncertainty in the fidelity of the solution in the optimization problem. The reason for these approximation errors is that the fitting of a circle to the manifold is rarely perfect. This approximation error δ is a function of the choice of waveform $\mathbf{g}(\cdot)$, spacing Δ, and the translation parameter b_n. Let $b_n = b_p + \Delta_n$ be an arbitrary parameter value, defined using an atom in the dictionary b_p and the translation variable $\Delta_n \in (-\frac{\Delta}{2}, \frac{\Delta}{2})$. The interpolation is based on the assumption that the ratio between $\Delta/2$ and the arbitrary translation variable Δ_n is equal to the ratio between θ and the angle θ_n between $\mathbf{g}(b_p)$ and $\mathbf{g}(b_n)$. Define the ratio of angles as:

$$\frac{\theta_n}{\theta} = \frac{\mathrm{Re}\{\langle \mathbf{g}(b_p), \mathbf{g}(b_n)\rangle\}\|\mathbf{g}(b_p)\|_2 \cdot \|\mathbf{g}(b_p + \frac{\Delta}{2})\|_2}{\mathrm{Re}\{\langle \mathbf{g}(b_p), \mathbf{g}(b_p + \frac{\Delta}{2})\rangle\}\|\mathbf{g}(b_p)\|_2 \cdot \|\mathbf{g}(b_n)\|_2} \tag{12}$$

$$= \frac{\mathrm{Re}\{\langle \mathbf{g}(b_p), \mathbf{g}(b_n)\rangle\}}{\mathrm{Re}\{\langle \mathbf{g}(b_p), \mathbf{g}(b_p + \frac{\Delta}{2})\rangle\}}, \tag{13}$$

as the Euclidian norm of any of the vectors on the manifold is equal to r. Therefore, define the following bound:

$$\left|\frac{\mathrm{Re}\{\langle \mathbf{g}(b_p), \mathbf{g}(b_n)\rangle\}}{\mathrm{Re}\{\langle \mathbf{g}(b_p), \mathbf{g}(b_p + \frac{\Delta}{2})\rangle\}}\right| \leq \left|\frac{\Delta_n}{\Delta/2}\right| + \delta \tag{14}$$

This bound cannot be calculated in closed form for all classes of waveforms $\mathbf{g}(\cdot)$, but it may be numerically simulated for choices of Δ and b_n. Assuming that the manifold is smooth, it is possible to find the approximation error δ as a function of Δ and Δ_n for all possible choices of b_p. Then, by finding the maximum value of that function, we obtain the worst-case bound on the interpolation error. To compute this bound ζ, we find the distance between the actual vector on the manifold and the approximated vector, based on the value of b that gives the maximum error:

$$\begin{aligned} \zeta &= \|\mathbf{g}(\hat{b}) - \tilde{\mathbf{g}}(\hat{b})\|_2, \\ \hat{b} &= \arg\max_{\hat{b}} \left|\frac{\mathrm{Re}\{\langle \mathbf{g}(b_p), \mathbf{g}(\hat{b})\rangle\}}{\mathrm{Re}\{\langle \mathbf{g}(b_p), \mathbf{g}(b_p + \frac{\Delta}{2})\rangle\}} - \frac{\Delta_n}{\Delta/2}\right| \end{aligned} \tag{15}$$

This value may then be input into the convex optimization solver. The reason why the error is found on the signal $\tilde{\mathbf{g}}(\hat{b})$, rather than on the parameter estimate $\hat{b}$, is because the fidelity constraint in the convex optimization formulation is based on the function reconstruction error.

To include the approximation error and extend the optimization problem to also allow for arbitrary complex amplitude

coefficients, we reformulate the problem formulation from Eqn. (9) using variable substitution:

$$
\begin{aligned}
\boldsymbol{\alpha} &= \boldsymbol{\alpha}^{r,p} - \boldsymbol{\alpha}^{r,n} + j(\boldsymbol{\alpha}^{i,p} - \boldsymbol{\alpha}^{i,n}),\ \boldsymbol{\alpha} \in \mathbb{C}^{1\times J} \\
\boldsymbol{\beta} &= \boldsymbol{\beta}^{r,p} - \boldsymbol{\beta}^{r,n} + j(\boldsymbol{\beta}^{i,p} - \boldsymbol{\beta}^{i,n}),\ \boldsymbol{\beta} \in \mathbb{C}^{1\times J} \\
\boldsymbol{\gamma} &= \boldsymbol{\gamma}^{r,p} - \boldsymbol{\gamma}^{r,n} + j(\boldsymbol{\gamma}^{i,p} - \boldsymbol{\gamma}^{i,n}),\ \boldsymbol{\gamma} \in \mathbb{C}^{1\times J} \\
\mathbf{x}_{\boldsymbol{\alpha}} &= \begin{bmatrix} \boldsymbol{\alpha}^{r,p} & \boldsymbol{\alpha}^{r,n} & \boldsymbol{\alpha}^{i,p} & \boldsymbol{\alpha}^{i,n} \end{bmatrix},\ \mathbf{x}_{\boldsymbol{\alpha}} \in \mathbb{C}^{1\times 4J} \\
\mathbf{x}_{\boldsymbol{\beta}} &= \begin{bmatrix} \boldsymbol{\beta}^{r,p} & \boldsymbol{\beta}^{r,n} & \boldsymbol{\beta}^{i,p} & \boldsymbol{\beta}^{i,n} \end{bmatrix},\ \mathbf{x}_{\boldsymbol{\beta}} \in \mathbb{C}^{1\times 4J} \\
\mathbf{x}_{\boldsymbol{\gamma}} &= \begin{bmatrix} \boldsymbol{\gamma}^{r,p} & \boldsymbol{\gamma}^{r,n} & \boldsymbol{\gamma}^{i,p} & \boldsymbol{\gamma}^{i,n} \end{bmatrix},\ \mathbf{x}_{\boldsymbol{\gamma}} \in \mathbb{C}^{1\times 4J} \\
\mathbf{x} &= \begin{bmatrix} \mathbf{x}_{\boldsymbol{\alpha}} & \mathbf{x}_{\boldsymbol{\beta}} & \mathbf{x}_{\boldsymbol{\gamma}} \end{bmatrix}^T, \\
\mathbf{E}(\Omega_J) &= \begin{bmatrix} \mathbf{C}(\Omega_J) & -\mathbf{C}(\Omega_J) & j\mathbf{C}(\Omega_J) & -j\mathbf{C}(\Omega_J) & \mathbf{U}(\Omega_J) & -\mathbf{U}(\Omega_J) & \cdots & -j\mathbf{V}(\Omega_J) \end{bmatrix}.
\end{aligned} \tag{16}
$$

We then use $\mathbf{x}$ as the optimization variable, together with another variable, $\mathbf{t}$, which is used in a mixed $\ell_1 - \ell_2$ norm to control the sparsity. Then, the convex optimization problem becomes:

$$
\begin{aligned}
\mathbf{x} = \mathrm{T}(\mathbf{y}, \mathbf{A}, \Omega_J) = \min_{\mathbf{x},\mathbf{t}} \|\mathbf{y} - \mathbf{A}\mathbf{E}(\Omega_J)\mathbf{x}\|_2^2 + \frac{\lambda}{2(\sigma^2+\zeta)}\|\mathbf{t}\|_1 \\
\text{s.t.} \left\{ \begin{array}{c} \sqrt{\mathbf{x}_{\boldsymbol{\beta}}(j)^2 + \mathbf{x}_{\boldsymbol{\gamma}}(j)^2} \le \mathbf{x}_{\boldsymbol{\alpha}}(j) r, \\ \mathbf{x}_{\boldsymbol{\alpha}}(j) r \cos(\theta) \le \mathbf{x}_{\boldsymbol{\beta}}(j) \le \mathbf{x}_{\boldsymbol{\alpha}}(j) r, \end{array} \right\} \text{ for } j = 1, \ldots, 4J, \\
\left\{ \mathbf{t}(j) \ge \sqrt{\boldsymbol{\alpha}^{r,p}(j)^2 + \boldsymbol{\alpha}^{r,n}(j)^2 + \boldsymbol{\alpha}^{i,p}(j)^2 + \boldsymbol{\alpha}^{i,n}(j)^2} \right\} \text{ for } j = 1, \ldots, J,
\end{aligned} \tag{17}
$$

Here we have included the CS measurement matrix $\mathbf{A}$, which was not part of the work in [13]. This formulation allows for both complex and negative amplitudes. This optimization formulation, when applied with all parameter values used in the dictionary $\mathbf{D}$, we name Complex Continuous Basis Pursuit (CCBP):

$$
(\mathbf{x}) = \mathrm{T}_{\mathrm{CCBP}}(\mathbf{y}, \mathbf{A}, \Omega_{CCBP}), \tag{18}
$$

where $\Omega_{CCBP} = \{b_1, b_2, \ldots, b_P\}$ is the set of all translation parameters that appear in the dictionary for the application of interest. Parameter estimates are then obtained using Eqns. (10-11). CCBP has a high computational complexity; it operates on matrices of size $12N$, whereas other CS algorithms operate on matrices of size N. However, its interpolation step has one important advantage: translation-invariance and interpolation enables CCBP to reconstruct arbitrary translation invariant sparse signals while requiring only a small subset of the N parameters to be contained in the corresponding dictionary. This makes it possible to incorporate the convex optimization solver into a greedy algorithm that quickly finds a rough estimate, which is then improved upon by a convex optimization solver.

IV. Interpolating Band-excluded Orthogonal Matching Pursuit

To be able to leverage both the accuracy of the convex optimization solvers and the speed of a greedy algorithm, we propose a greedy algorithm, which may improve upon its estimate using the convex optimization in (17). In [6] it is shown how the Subspace Pursuit algorithm [24] may be utilized for this purpose. However, in that work the frequencies to estimate are well separated, whereas in this work, we also evaluate the algorithms for overlapping pulses with the band exclusion function disabled. In that case the Subspace Pursuit algorithm may pick an incorrect dictionary element that is coherent with a strong signal component rather than the correct dictionary element for a weak signal component. This happens because the Subspace Pursuit algorithm attempts to find all the pulses in the signal in one iteration. Instead, we utilize the BOMP algorithm with interpolation, termed Interpolating Band-excluded Orthogonal Matching Pursuit (IBOMP). This is a greedy algorithm with an optional convex optimization problem. The algorithm improves upon the BOMP algorithm by using interpolation in each iteration to enhance the estimate of the translation parameter. The IBOMP algorithm is shown in Algorithm 1. First, the best correlating atom index i_n is found by generating a proxy for the sparse signal. This proxy is trimmed based on the band exclusion function $B_\eta(S)$, as defined in Eqn. (4). The selected atom, i_n, is then input to an interpolation function, $\mathrm{T}(\cdot)$. This function outputs an estimated translation parameter, which is used to create a new atom for a signal dictionary, $\mathbf{B}$, by using the original parametric signal model. This new signal dictionary is used to find the basis coefficients $\mathbf{a}$ using least squares. Then, a new residual is calculated and n and S are updated. This loop runs K times, i.e. once for each pulse in the signal. After the greedy algorithm is done the estimates may be improved upon by running the CCBP algorithm on a limited parameter set based on the current parameter estimates. When exiting the loop the estimates found by the greedy algorithm are put into a new set, Ω, together with ξ adjacent indices. This is necessary because the parameter values generating $\mathbf{y}$ may not be sufficiently incoherent and may therefore skew the peaks of the proxy estimate. Therefore, as a precaution, we include the closest neighbors on each side. The set Ω is input to the convex optimization in (17) along with the measurement matrix and the received signal. The output from the CCBP algorithm is used to generate new estimates of the reconstructed signal $\tilde{\mathbf{f}}$ and the parameter vector $\tilde{\mathbf{b}}$.

Algorithm 1 Interpolating Band-excluded Orthogonal Matching Pursuit (IBOMP)

INPUTS: Compressed signal $\mathbf{y}$, interpolation function $\mathrm{T}(\cdot)$, dictionary $\mathbf{D}$, measurement matrix $\mathbf{A}$ and number of adjacent indicies to include in the CCBP algorithm ξ.
OUTPUTS: Reconstructed signal $\tilde{\mathbf{f}}$ and parameter estimates $\tilde{\mathbf{b}}$.
Initialize: $\mathbf{y}_{res} = \mathbf{y}$, $\mathbf{B} = \emptyset$, $n = 1$ and $S^n = \emptyset$.
while $n \leq K$ **do**
  $i_n = \arg\max_i |\langle \mathbf{y}_{res}, \mathbf{A}\mathbf{D}_i \rangle|$, $i \notin B_0(S^{n-1})$
  $\hat{b}_n = \mathrm{T}(\mathbf{y}_{res}, \mathbf{A}, i_n)$
  Include sampled version of $f(t - \hat{b}_n)$ as new atom in $\mathbf{B}$
  $\mathbf{a} = (\mathbf{A}\mathbf{B})^\dagger \mathbf{y}$
  $\mathbf{y}_{res} = \mathbf{y} - \mathbf{A}\mathbf{B}\mathbf{a}$
  $n = n + 1$
  $S^n = S^{n-1} \cup \{i_n\}$
end while
$\Omega = \cup\{\Delta(s - \xi), \Delta s, \Delta(s + \xi) | s \in S^n\}$
Use $\mathrm{T}(\mathbf{y}, \mathbf{A}, \Omega)$ from Eqn. (17) to obtain $\mathbf{x}$
Obtain $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{b}}$ using (11) and (8)

In this work we use two interpolation functions: parabolic interpolation and polar interpolation.

Parabolic Interpolation Function

We define the parabolic interpolation function based on Eqn. (3) as follows:

$$\mathrm{T_{Pa}}(\mathbf{y}_{res}, \mathbf{A}, i_n) = -\frac{\Delta}{2}\frac{\hat{R}[i_n+1]-\hat{R}[i_n-1]}{\hat{R}[i_n+1]-2\hat{R}[i_n]+\hat{R}[i_n-1]}+i_n\Delta, \tag{19}$$

where $\hat{R}[m]$ is defined as:

$$\hat{R}[m] = \sum_{l=1}^{N} \mathbf{y}_{res}[l]\cdot\mathbf{Ag}[l-m], \tag{20}$$

In the IBOMP algorithm there is no reason to calculate the $\hat{R}[m]$ function as it is identical to the proxy in the greedy algorithm.

Polar Interpolation Function

The polar interpolation function is based on Eqn. (7). We reformulate those equations to a linear least squares problem:

$$\mathbf{y}_{res,n} \approx \mathbf{A}\begin{bmatrix}\mathbf{g}(b_p-\frac{\Delta}{2}) & \mathbf{g}(b_p) & \mathbf{g}(b_p+\frac{\Delta}{2})\end{bmatrix}\left(\begin{bmatrix}1 & r\cos(\theta) & -r\sin(\theta)\\ 1 & r & 0\\ 1 & r\cos(\theta) & r\sin(\theta)\end{bmatrix}^{-1}\right)^{T}\mathbf{x},\ \mathbf{x}=\begin{bmatrix}a_i\\ a_i r\cos\left(\frac{2\Delta_n\theta}{\Delta}\right)\\ a_i r\sin\left(\frac{2\Delta_n\theta}{\Delta}\right)\end{bmatrix}. \tag{21}$$

In this formula, a rotation matrix rotates the three $\mathbf{g}$ vectors to form a new, general basis for the circle arc and $\mathbf{x}$ scales the vectors in that basis to estimate the received signal. Given a signal or residual $\mathbf{y}_{res,n}$ and the atom $\mathbf{g}(b_p)$ in the dictionary that correlates the strongest with the residual, we may solve Eqn. (21) as a linear least squares problem with $\mathbf{x}$ as the unknown. From the estimate $\hat{\mathbf{x}} = \{\hat{x}_1, \hat{x}_2, \hat{x}_3\}$, we may obtain an estimate of $b_n = b_p + \Delta_n$ as:

$$b_n = b_p + \arctan\left(\frac{\hat{x}_3}{\hat{x}_2}\right)\frac{\Delta}{2\theta}. \tag{22}$$

We term the interpolation function in Eqn. (21) $\mathrm{T_{Po}}(\mathbf{y}_{res}, \mathbf{A}, i_n)$, where i_n is the index in the dictionary for $\mathbf{g}(b_p)$.

The IBOMP algorithm finds one estimate of a pulse using either parabolic interpolation as in Eqn. (19) or polar interpolation as in Eqn. (21) and then removes that estimated waveform from the residual $\mathbf{y}_{res,n}$, after which it continues to work on the residual. After the greedy algorithm has found a number of promising estimates, we may improve upon these with the CCBP algorithm. Another solution would be to use CCBP in each iteration of the BOMP algorithm. However, this increases the computational complexity and in our experiments we have not found that this improves performance.

For the band exclusion function, we set $\eta = 0$ if we know that the pulses are well spaced (i.e. orthogonal). In that case, the band exclusion does not inhibit two pulses from interfering, but inhibits the algorithm from finding the same pulse again due to a large remaining residual. Otherwise, if we are only interested in identifying pulses with a given spacing, we may adjust η to reflect this. If we cannot make any assumption to the spacing, we set $\eta = 1$.

In the numerical experiments, we investigate the effect of the optional CCBP algorithm after the greedy algorithm. To distinguish between the IBOMP algorithm with and without this optional step, we write IBOMP+CCBP if the algorithm uses the CCBP algorithm and IBOMP if it does not. Furthermore, to distinguish whether parabolic interpolation or polar interpolation

is used, we use PaIBOMP and PoIBOMP instead of IBOMP.

V. Numerical Experiments

To evaluate the proposed algorithms, we first must find good parameter values for the convex optimization problem. The two parameters ζ and λ signify approximation error and sparsity trade-off, respectively. This analysis shows why the FE problem is more complex than the TDE problem when assuming both positive and negative complex amplitude coefficients. The analysis is followed by experiments for the TDE problem that evaluate the proposed algorithms in different scenarios. We investigate their performance for well-spaced pulses and for overlapping pulses and we investigate the performance when the signal experiences signal noise instead of measurement noise. All the code along with the results and figures in this paper is available at www.sparsesampling.com/cpe following the principle of Reproducible Research [25].

Before explaining the experiments further, we define the two types of signals and the dictionaries that are used in all the following experiments. For both types, the general signal model is as defined in Eqn. (6).

For the TDE numerical experiments, we let the pulse model $g(t)$ be a chirp signal defined as

$$g(t,b_n) = \frac{1}{\sqrt{\mathcal{E}_g}} \cdot e^{j2\pi(f_0+\frac{\Delta f}{2T}(t-b_n))(t-b_n)} \cdot p(t-b_n), \quad p(t) = \begin{cases} \frac{T}{2}(1+\cos(2\pi t/T)), & t \in (0,T) \\ 0, & \text{otherwise} \end{cases}, \tag{23}$$

where $f_0 = 1$MHz is the center frequency, $\Delta f = 40$MHz is the sweeped frequency , and $T = 1\mu$s is the duration of the chirp in time. The chirp is limited in time by a raised cosine pulse and normalized to unit energy. We generate a sampled time signal, $\mathbf{g}(b_n)$ by sampling the pulse function:

$$\mathbf{g}(b_n) = \begin{bmatrix} g_1(b_n) & g_2(b_n) & \cdots & g_N(b_n) \end{bmatrix}, \quad g_i(b_n) = g(t-(i-1)T_s, b_n) \tag{24}$$

Here, T_s is the sampling period. We sample the signal at 50MHz, since the corresponding bandwidth of the signal contains more than 99% of its energy. For each signal we take $N = 500$ samples. The dictionary $\mathbf{D}$ for the TDE problem is a circulant matrix with shifted versions of $\mathbf{g}(b_n)$:

$$\begin{aligned} \mathbf{D}_{TDE} &= \begin{bmatrix} \mathbf{g}(b_1) & \mathbf{g}(b_2) & \cdots & \mathbf{g}(b_N) \end{bmatrix} \\ &= \begin{bmatrix} g[0] & g[N-1] & \cdots & g[1] \\ g[1] & g[0] & \ddots & g[2] \\ \vdots & \vdots & \ddots & \vdots \\ g[N-1] & g[N-2] & \cdots & g[0] \end{bmatrix}, \end{aligned} \tag{25}$$

where $\mathbf{g}(0) = \begin{bmatrix} g[0] & g[1] & \cdots & g[N] \end{bmatrix}^T$. This means that the spacing between atoms in this dictionary is equal to the sampling rate, T_s.

For the FE numerical experiments we generate frequency-sparse signals of length $N = 100$ containing K complex sinusoids

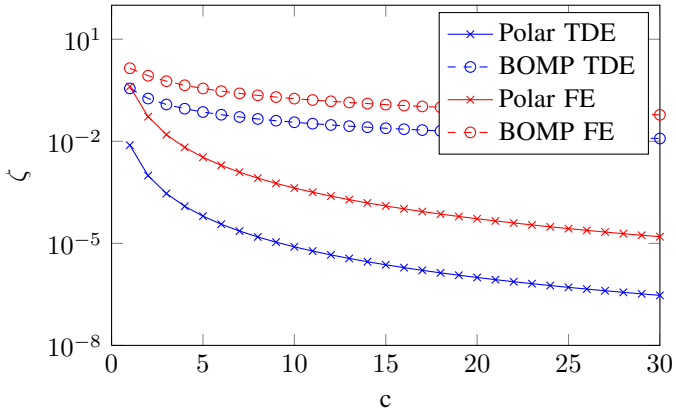

Fig. 1: Polar estimation approximation error analysis.

with frequencies selected uniformly at random. The continuous signal function then becomes:

$$g(t, b_n) = \frac{1}{\sqrt{N}} \exp^{j2\pi b_n t/N}. \tag{26}$$

The basic dictionary for this signal is a DFT matrix with spacing 1Hz between atoms.

A. ζ and λ analysis

We first investigate the approximation error parameter, ζ. We have conducted numerical experiments on the bound in Eqn. (15). These experiments are conducted for both the TDE and FE problem, to show that the approximation error is problem-specific.

The approximation error from Eqn. (15) depends on the specific signal model and the dictionary spacing Δ. For each of the two signal models, we have performed numerical experiments for a range of spacings. For the TDE problem, the spacing is defined as $\Delta = \frac{T_s}{c}$, where c is called the redundancy factor and is used as the experiment variable. For the FE problem, the spacing is defined as $\Delta = \frac{1}{c}$. In each experiment, we pick a center atom in the dictionary and uniformly sample the parameter space around that atom using 100 samples. Each sample constitutes a parameter value b to input into the equations in Eqn. (15). The result of the experiment is shown in Fig. 1. In the figure, the approximation error is compared to the maximum approximation noise from BOMP, i.e. the approximation error when a parameter lies exactly in between two atoms in the dictionary. As can be seen, the FE signals suffer from a higher approximation noise than the TDE case. For the TDE problem, good performance is achievable without any redundancy in the dictionary, i.e. for $c = 1$, whereas for FE a higher redundancy factor is needed. This, however, increases computation time significantly and introduces coherence.

When the coherence of the dictionary increases, it becomes less likely to find a unique, sparse solution to the problem. This is because of the coherence of the redundant dictionary and the looseness of the fidelity constraint in Eqn. (17), due to the approximation error. This is best illustrated using the Spark of the dictionary. Given a matrix $\mathbf{D}$ we define $\sigma = \text{Spark}(\mathbf{D})$ as the smallest possible number such that there exists a subgroup of σ columns from $\mathbf{D}$ that are linearly dependent [26]. The Spark is computationally heavy to compute, but an upper bound can be found [26]. Define a sequence of optimization

problems, $i = 1, \ldots, N$:

$$\tilde{\mathbf{x}}_i^0 = \min_{\mathbf{x} \in \mathbb{C}^N} \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \mathbf{D}\mathbf{x} = \bar{0},\ \mathbf{x}_i = 1, \tag{27}$$

$$\text{Spark}(\mathbf{D}) = \min_{1 \leq i \leq N} \|\tilde{\mathbf{x}}_i^0\|_0 \tag{28}$$

The optimization problem is however not computationally feasible due to the ℓ_0 term. Instead, we use a ℓ_1 norm, which is solvable in polynomial time using standard solvers. Because $\|\tilde{\mathbf{x}}_i^0\|_0 \leq \|\tilde{\mathbf{x}}_i^1\|_0$, we obtain the upper bound on the Spark:

$$\tilde{\mathbf{x}}_i^1 = \min_{\mathbf{x} \in \mathbb{C}^N} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{D}\mathbf{x} = \bar{0},\ \mathbf{x}_i = 1, \tag{29}$$

$$\text{Spark}(\mathbf{D}) \leq \min_{1 \leq i \leq N} \|\tilde{\mathbf{x}}_i^1\|_0 \tag{30}$$

Using the two dictionaries defined for the TDE and FE problems, we have found this upper bound on the Spark. For the TDE problem $\text{Spark}(\mathbf{D}_{TDE}) \leq N$, because all the columns are linearly independent. There is no redundancy and the matrix has full rank. Hence, coherence is not a problem in the TDE case. For the FE problem with $c = 5$ we have $\text{Spark}(\mathbf{D}_{\text{FE}}) \leq 101$. This problem seems to contradict the results from [6], where polar interpolation works well for the FE problem. However, in that work the amplitude coefficients are real and non-negative. If we find the spark with those assumptions i.e., solve the following optimization problem:

$$\tilde{\mathbf{x}}_i^1 = \min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{D}\mathbf{x} = \bar{0},\ \mathbf{x}_i = 1,\ \mathbf{x} \geq \bar{0}, \tag{31}$$

$$\text{Spark}_+(\mathbf{D}) \leq \min_{1 \leq i \leq N} \|\tilde{\mathbf{x}}_i^1\|_0, \tag{32}$$

the upper bound becomes $\text{Spark}_+(\mathbf{D}_{\text{FE}}) \leq N$ even though the matrix $\mathbf{D}$ does not have full rank.

The first result shows that polar interpolation is easier to apply to the TDE problem, at least with the signal model chosen for this work, than to the FE problem. It does not mean that polar interpolation cannot be applied to the FE problem, but it will require a different convex optimization formulation in which the constraints are tightened further, to shrink the solution set. This may be possible in the case where the problem allows for some specific assumptions, e.g. some symmetry in the spectrum which can be formulated as constraints in the optimization problem.

In the following we limit our focus to the TDE problem and show the estimator performance in different scenarios. First, however, we must investigate what the other optimization variable, λ, shall be. This is no trivial task as its optimal value changes depending on the function $g(t)$, the subsampling ratio κ, the SNR, etc. To visualize this and to find a good candidate for λ for later experiments, we have evaluated different choices of λ for the TDE problem, while also varying κ and the SNR. The estimator performance is evaluated in terms of the mean squared error (MSE) on the $\mathbf{b}$ parameter, termed the $\mathbf{b}$-MSE. This corresponds to the sample variance of the estimators and is a measure of estimator precision. We perform Monte Carlo experiments to get an average result on the error. In each experiment, we generate a time signal with one pulse ($K = 1$) by sampling the signal function in Eqn. (6). The real and imaginary part of the amplitude coefficient a are drawn from a uniform distribution between 1 and 10. As shown in Fig. 1 there is no need for a redundant dictionary matrix for the TDE problem, so we use $c = 1$, i.e. the dictionary has size 500×500. The results are shown in Fig. 2. The colorbar signifies the $\mathbf{b}$-MSE

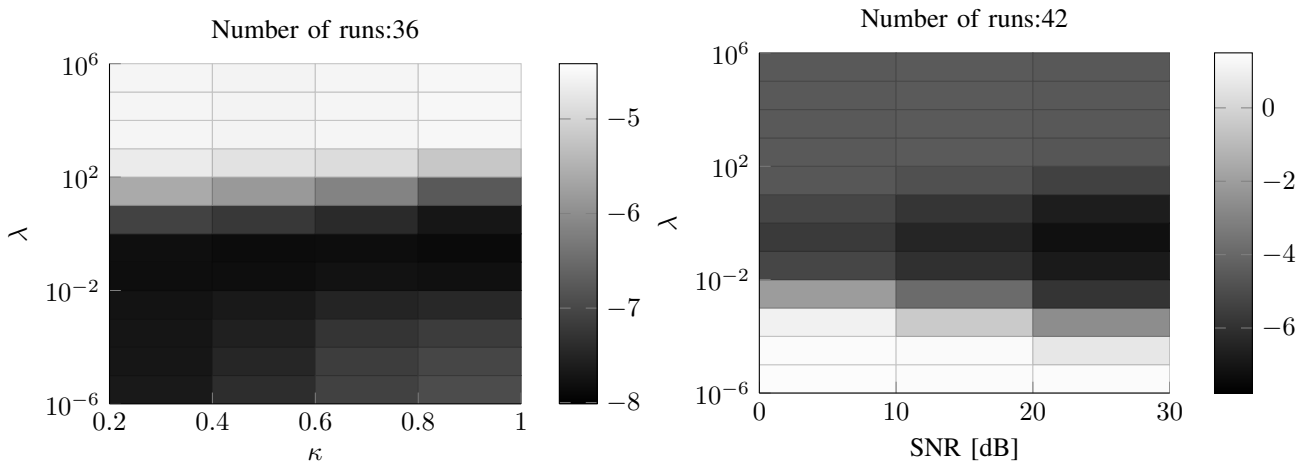

Fig. 2: λ analysis for CCBP. Left figure is with SNR 1000 and right figure is with $\lambda = 1$. The z-axis is the mean squared error of the parameter estimate $\mathbf{b}$-MSE. The scaling on the z-axis is microseconds squared on a logarithmic scale.

in microseconds squared on a logarithmic scale. As λ increases the ℓ_1-norm of the solution vector $\mathbf{x}$ decreases and eventually becomes the zero-vector. When this happens the CCBP algorithm falls back to the BOMP algorithm. As can be seen, $\lambda = 1$ is a good choice for TDE estimation.

B. Performance Evaluation of the estimators

Now the optimization variables for the TDE problem have been chosen and the next experiment shall evaluate the estimation performance of the three proposed algorithms versus other TDE algorithms when CS is applied. We evaluate the estimators in three different scenarios:

- Case A: Experiments for well-spaced pulses with and without measurement noise
- Case B: Experiments for overlapping pulses with and without measurement noise
- Case C: Experiments for overlapping pulses with signal noise

The two first cases evaluate how much the signal may be subsampled using CS and still attain good estimation precision. The last case evaluates the effect of noise folding, when the noise is added before the measurement matrix $\mathbf{A}$ is multiplied on.

For all the experiments we use a Random Demodulator CS measurement matrix [27], $\mathbf{\Psi} \in \{-1, 0, 1\}^{M \times N}$. We set $M = \kappa N$, where $\kappa \in [0, 0.5)$ is the CS subsampling rate. We evaluate the performance of the three estimators by computing the translation parameter mean squared error ($\mathbf{b}$-MSE) between the true value of the time delay and the estimated value. Each point in the plot is the result of more than 100 Monte Carlo experiments. The algorithms we evaluate are as follows:

- **BOMP** - a greedy algorithm proposed in [16] with no interpolation,
- **TDE-MUSIC** - an algorithm that reconstructs the received signal using Eqn. (1) after which the problem is converted to a frequency estimation problem that is solved using the MUSIC algorithm, as explained in Eqn. (5),
- **PaIBOMP** - BOMP with parabolic interpolation,
- **CCBP** - The CCBP algorithm in Eqn. (18),
- **PoIBOMP** - BOMP with polar interpolation, and

- **PaIBOMP+CCBP** - BOMP with parabolic interpolation, where the estimates are refined using the CCBP algorithm.

The reason why we use parabolic interpolation in the PaIBOMP+CCBP algorithm, instead of polar interpolation is that the parabolic interpolation is more stable when the pulses are overlapping. This is shown in the numerical experiments. For the PaIBOMP+CCBP algorithm we set $\xi = 0$, as we have rarely observed that the greedy algorithms chooses the wrong atom for the TDE problem.

Case A: Well-spaced pulses: This experiment is performed with $K = 3$ well-spaced pulses, i.e. $\eta = 0$. The minimum separation between pulses is set to 10^{-6} seconds, i.e. exactly the width of a pulse. This means there is no overlap anywhere between pulses. The result of our comparison is shown in Fig. 3. As shown, the polar interpolation algorithms outperform all the other algorithms. With $\xi = 0$ PaIBOMP+CCBP has the same computational complexity as TDE-MUSIC, whereas CCBP is significantly more computationally heavy.

Also note that PoIBOMP outperforms both CCBP and PaIBOMP+CCBP while also being significantly less computational complex. This is because the pulses are well separated. In the next experiment, we use overlapping pulses which affects the purely greedy algorithms more than the pure and hybrid convex optimization algorithms.

Case B: Overlapping pulses: For this experiment we use the same parameter values, except that the minimum pulse separation is now set to $5 \cdot T_s$, i.e. five times the sampling rate. The reason why we do not set the separation to 0 is that if two identical pulses are received, there is no possibility of correctly decoding these without introducing further assumptions. Therefore, we introduce this minimum spacing. We set $\eta = 1$, i.e. we disable the band exclusion, such that there is no restriction on which dictionary atoms are used in each iteration. The result is shown in Fig. 4.[1] As can be seen the greedy algorithm are heavily affected by this, especially BOMP and PoIBOMP. The PaIBOMP+CCBP algorithm is also affected in that it requires a little higher κ before it attains the same performance as CCBP than in Fig. 3. This figure also shows why we use parabolic interpolation in the PaIBOMP+CCBP algorithm, instead of the polar interpolation from PoIBOMP. When the pulses are overlapping the polar interpolation has erratic instability issues. It is important to note that the irregularities for PoIBOMP in Fig. 4 are due to a single Monte Carlo simulation in which the estimation fails. Polar interpolation relies on the value of all N dimensions of the signal for the hypersphere assumption and when pulses are overlapping this assumption is incorrect. As shown in Eqn. (22) the translation estimate for PoIBOMP relies on finding the inverse tangent and if $\hat{x}_2$ is erroneous this may result in a large error. In contrast, the parabolic interpolation uses only three points from the cross correlation function. Hence, with overlapping pulses the PoIBOMP algorithm suffers more from the interference from other pulses than PaIBOMP.

Case C: Noise folding: In our final numerical experiment we investigate the effect of signal noise in the received signal, rather than measurement noise. Signal noise introduces noise folding [28], [29], which decreases reconstruction performance. Signal noise occurs when the signal models is: $\mathbf{y} = \mathbf{A}(\mathbf{D}\mathbf{x} + \mathbf{n}) + \mathbf{w}$, where $\mathbf{n}$ is the signal noise and $\mathbf{w}$ is the measurement noise. In the experiments so far we have only considered measurement noise, but now we focus on the signal noise and set the measurement noise to zero. The estimator performance for $\kappa = 1$, i.e. no subsampling, and $\kappa = 0.4$ is shown in Fig. 5. As can be seen in the top two figures all the estimators' signal reconstruction are affected by noise folding when CS is used.

[1] In the experiment generating the figure to the right we have removed one spurious experiment from the results. For $\kappa = 0.5$ in one out of 118 Monte Carlo simulations the generated signal and the instantiation of the Random Demodulator correlated in such a way that the proxy in the greedy algorithms contained two peaks instead of one. This resulted in a significant estimation error, but to illustrate that this happened in only one individual experiment, we have removed it from the plots. The original data set is available for inspection at www.sparsesampling.com/cpe.

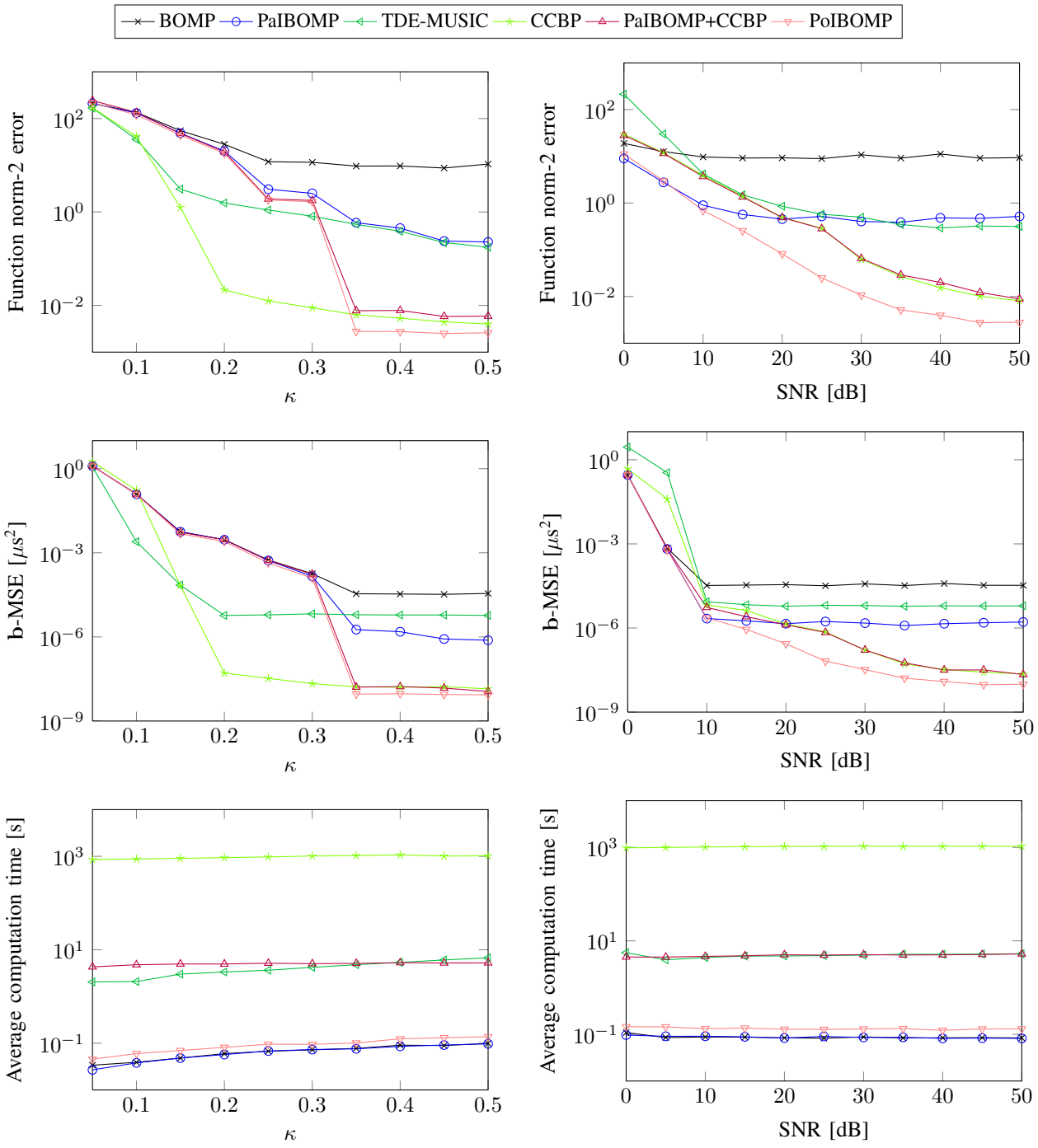

Fig. 3: The estimator precision with non-overlapping pulses for the TDE problem. The left figures are noise-less experiments for varying choices of subsampling ratios, κ, while the right figures are for $\kappa = 0.4$ and with varying SNR levels. The top figures is signal reconstruction quality, the middle row is translation parameter estimation precision and the bottom row is average computation time.

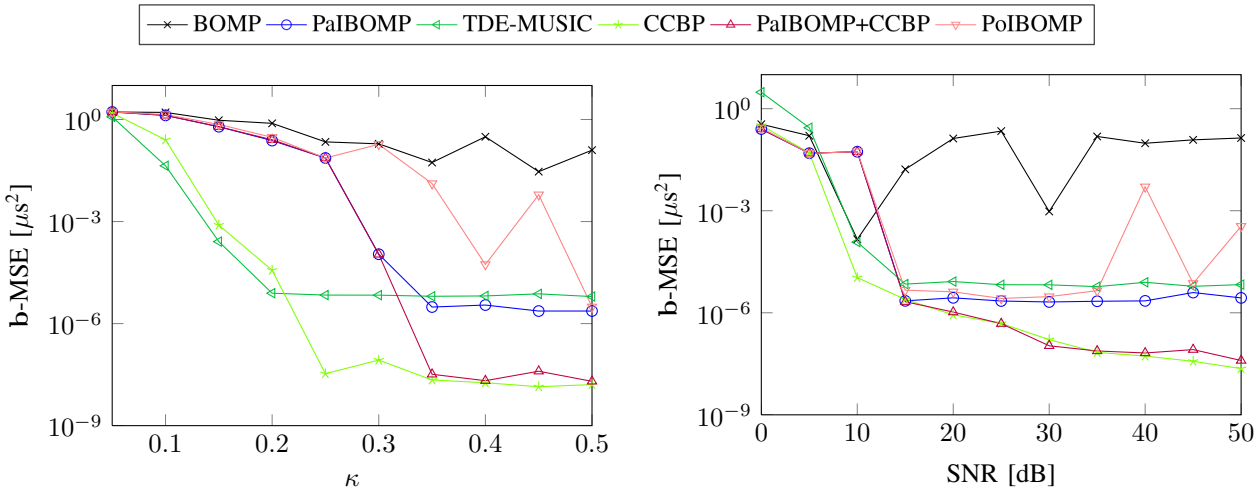

Fig. 4: The estimator precision with overlapping pulses for the TDE problem. The left side is the noise-less case with varying subsampling ratios and the right side is with $\kappa = 0.4$ and varying SNR levels.

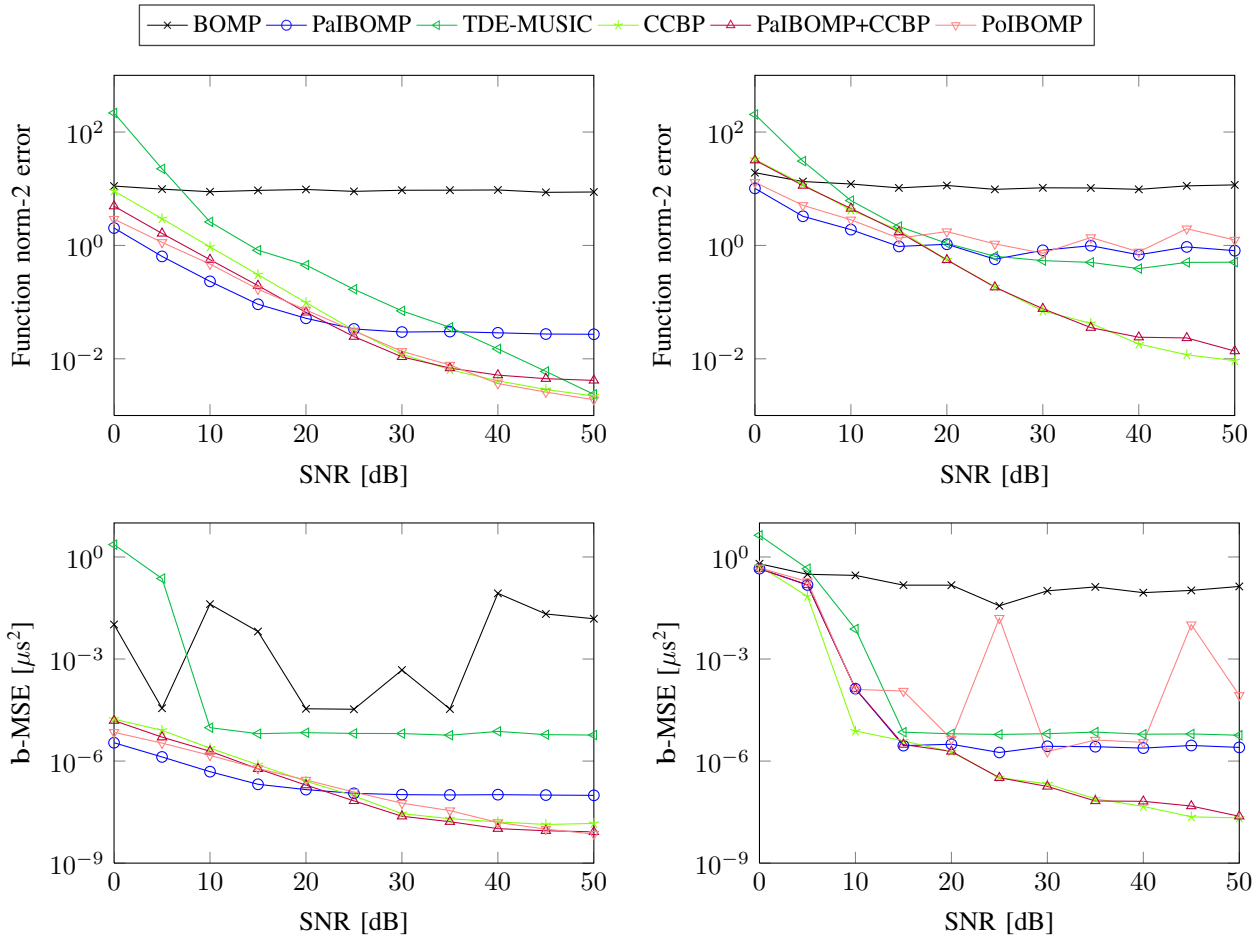

Fig. 5: The function error and estimator precision with overlapping pulses and noise folding for the TDE problem. The left side is for $\kappa = 1$ and the right side is for $\kappa = 0.4$.

However, in the bottom two figures we see that for the parameter estimation some of the algorithms are still able to estimate the translation parameter at a similar precision as without subsampling. The greedy algorithms are most heavily affected by the noise folding, whereas the convex optimization based algorithms are less affected. As shown in [29] noise folding may be remedied by using quantization. In that work, the authors postulate that when a receiver uses e.g. half as many samples as a classical receiver, it may instead use twice as many bits for quantization. This is also demonstrated in [30] for a spread spectrum receiver.

VI. Conclusion

With our numerical experiments, we show that the proposed CCBP and, with high enough sampling frequency, PaI-BOMP+CCBP algorithm outperform all the other algorithms in terms of estimation precision. If the pulses are known to be well separated, the PoIBOMP algorithm attains the best estimation precision, while having very low computational complexity. If the pulses cannot be assumed well separated it is better to use the pure convex optimization algorithm CCBP or the hybrid PaIBOMP+CCBP to attain the best estimation precision. At lower subsampling ratios the algorithms that achieve the best performance are CCBP or TDE-MUSIC. In our experiments CCBP attained the best estimation precision; however, it is also significantly more computationally complex. It may be possible to reduce this complexity through a more judicious formulation of solvers for the proposed optimization. The proposed modified optimization problem introduces many new variables to be able to capture the full signal information, but it may be possible to decrease this number with a smarter problem formulation.

In the last numerical experiment, we investigated the estimators' performance when the observations feature signal noise instead of measurement noise. This results in noise folding which has been shown before to severely affect signal reconstruction. In our experiments we see that the greedy algorithms are highly sensitive to such noise folding, while TDE-MUSIC, CCBP and PaIBOMP+CCBP are less sensitive.

The work shows that compressive sensing for the class of sparse translation-invariant signals allows for a lower sampling rate and that the use of polar interpolation increases the estimation precision. The cost in terms of computational complexity is a trade-off in terms of the desired estimation precision and whether it is known if the signal pulses are well-separated or not.

References

[1] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] E. J. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, pp. 1207–1223, 2005.

[3] E. J. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, no. 3, pp. 969–985, June 2007.

[4] H. Rauhut, K. Schnass, and P. Vandergheynst, "Compressed sensing and redundant dictionaries," *IEEE Transactions on Information Theory*, vol. 54, no. 9, pp. 2210–2219, Sep. 2008.

[5] J. K. Nielsen, M. G. Christensen, and S. H. Jensen, "On compressed sensing and the estimation of continuous parameters from noisy observations," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 3609–3612.

[6] K. Fyhn, H. Dadkhahi, and M. F. Duarte, "Spectral compressive sensing with polar interpolation," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2013, to appear.

[7] L. Jacques and C. De Vleeschouwer, "A geometrical study of matching pursuit parametrization," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2835–2848, Jul. 2008.

[8] D. Ramasamy, S. Venkateswaran, and U. Madhow, "Compressive parameter estimation in awgn," *Preprint (available at: http://arxiv.org/abs/1304.7539)*, 2013.

[9] R. Boucher and J. Hassab, "Analysis of discrete implementation of generalized cross correlator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 3, pp. 609–611, Jun. 1981.

[10] G. Jacovitti and G. Scarano, "Discrete time techniques for time delay estimation," *IEEE Trans. Signal Process.*, vol. 41, no. 2, pp. 525–533, Feb. 1993.

[11] D. Aiordachioaie and V. Nicolau, "On time delay estimation by evaluation of three time domain functions," in *3rd International Symposium on Electrical and Electronics Engineering (ISEEE)*, Sep. 2010, pp. 281–286.

[12] F. Viola and W. Walker, "A spline-based algorithm for continuous time-delay estimation using sampled data," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 52, no. 1, pp. 80–93, Jan. 2005.

[13] C. Ekanadham, D. Tranchina, and E. P. Simoncelli, "Recovery of sparse translation-invariant signals with continuous basis pursuit," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4735–4744, Oct. 2011.

[14] M. F. Duarte, "Localization and bearing estimation via structured sparsity models," in *IEEE Statistical Signal Processing Workshop (SSP)*, Ann Arbor, MI, USA, 2012.

[15] M. F. Duarte and R. G. Baraniuk, "Spectral compressive sensing," *Applied and Computational Harmonic Analysis*, 2012, to appear.

[16] A. Fannjiang and W. Liao, "Coherence pattern-guided compressive sensing with unresolved grids," *SIAM Journal on Imaging Sciences*, vol. 5, no. 1, pp. 179–202, Feb. 2012.

[17] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[18] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.

[19] H. Saarnisaari, "TLS-ESPRIT in a time delay estimation," in *IEEE 47th Vehicular Technology Conference*, vol. 3, 1997, pp. 1619–1623.

[20] C. Le Bastard, V. Baltazart, and Y. Wang, "Modified ESPRIT (M-ESPRIT) algorithm for time delay estimation in both any noise and any radar pulse context by a GPR radar," *Signal Processing*, vol. 90, no. 1, pp. 173–179, 2010.

[21] J. Li and R. Wu, "An efficient algorithm for time delay estimation," *IEEE Transactions on Signal Processing*, vol. 46, no. 8, pp. 2231–2235, 1998.

[22] K. Gedalyahu and Y. Eldar, "Time-delay estimation from low-rate samples: A union of subspaces approach," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3017–3031, 2010.

[23] S. S. Chen *et al.*, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, pp. 33–61, 1998.

[24] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.

[25] P. Vandewalle, J. Kovacevic, and M. Vetterli, "Reproducible research in signal processing – What, why, and how," *IEEE Signal Processing Magazine*, vol. 26, no. 3, pp. 37–47, May 2009.

[26] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.

[27] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk, "Beyond Nyquist: Efficient sampling of sparse bandlimited signals," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 520–544, Jan. 2010.

[28] E. Arias-Castro and Y. C. Eldar, "Noise folding in compressed sensing," *IEEE Signal Process. Lett.*, vol. 18, no. 8, pp. 478–481, Aug. 2011.

[29] J. Treichler *et al.*, "Dynamic range and compressive sensing acquisition receivers," in *Defense Applications of Signal Processing (DASP)*, Coolum, Australia, Jul. 2011.

[30] K. Fyhn, T. L. Jensen, T. Larsen, and S. H. Jensen, "Compressive sensing for spread spectrum recievers," *Preprint (available at: http://arxiv.org/abs/1302.6703)*, 2013.