Compressive Sensing in Communication Systems

Fyhn, Karsten

Compressive Sensing in Communication Systems

Ph.D. Dissertation

Karsten Fyhn

Aalborg University
Department of Electronic Systems
Fredrik Bajers Vej 7B
DK-9220 Aalborg

Abstract

Wireless communication is omnipresent today, but this development has led to frequency spectrum becoming a limited resource. Furthermore, wireless devices become more and more energy-limited, due to the demand for continual wireless communication of higher and higher amounts of information. The need for cheaper, smarter and more energy-efficient wireless devices is greater now than ever. This thesis addresses this problem and concerns the application of the recently developed sampling theory of compressive sensing in communication systems.

Compressive sensing is the merging of signal acquisition and compression. It allows for sampling a signal with a rate below the bound dictated by the celebrated Shannon-Nyquist sampling theorem. In some communication systems this necessary minimum sample rate, dictated by the Shannon-Nyquist sampling theorem, is so high it is at the limit of what the current technology can manage. Even if the sampling rate is within the bounds of what is currently possible, the electrical components for acquiring highly oscillating signals are very expensive and energy demanding. Compressive sensing may mitigate this challenge by lowering the bound on the sample rate. The compressive sensing research area is still in its infancy and has so far been mainly a theoretical field. However, hardware implementations and actual application examples in current communication technologies have begun to emerge.

The approach in this thesis has been to attack some of the current challenges with using compressive sensing in communication systems. The main contribution of this thesis is two-fold: 1) a new compressive sensing hardware structure for spread spectrum signals, which is simpler than the current state-of-the-art, and 2) a range of algorithms for parameter estimation for the class of translation-invariant signals, which outperform the current state-of-the-art algorithms for frequency and time delay estimation. Though the proposed methods and algorithms in this work have been designed for use in communication systems, they are also relevant outside this area.

Resumé

Trådløs kommunikation er allestedsnærværende i dag, men denne udvikling har ført til at frekvensspektret nu er en begrænset ressource. Derudover bliver trådløse enheder mere og mere energi-begrænsede, på grund af efterspørgslen efter kontinuerlig trådløs kommunikation af større og større mængder information. Behovet for billigere, smartere og mere energi-effektive trådløse enheder er større end nogensinde før. Denne afhandling adresserer dette problem og omhandler anvendelsen af den ny-udviklede samplingsteori compressive sensing i kommunikationssystemer.

Compressive sensing er sammenlægningen af signal akkvisition og komprimering. Dette tillader at måle et signal med en hastighed under den nedre grænse dikteret af den velkendte Shannon-Nyquist samplingsteori. I nogen kommunikationssystemer er den nødvendige samplingsrate, som dikteret af Shannon-Nyquists samplingsteori, så høj at den er på grænsen af det mulige med nutidens teknologi. Selv hvis samplingsraten er indenfor grænserne af det mulige er de elektriske komponenter, der skal til for at måle stærkt oscillerende signaler, meget dyre og meget energi-ineffektive. Compressive sensing kan være en løsning på dette problem ved at sænke den nedre grænse for samplingsraten. Compressive sensing forskningsfeltet er stadig ungt og har indtil videre primært været et teoretisk felt. Dog begynder der nu at fremkomme hardware implementationer og reelle applikationseksempler indenfor nutidige kommunikationsteknologier.

Tilgangen i denne afhandling har været at angribe nogle af de nutidige udfordringer indenfor anvendelsen af compressive sensing i kommunikationssystemer. Hovedbidraget i denne afhandling består af to dele: 1) en ny compressive sensing hardware struktur til spread spectrum systemer, som er simplere end den nutidige state-of-the-art og 2) en række af algoritmer til parameterestimering til translationsinvariante signaler, som udkonkurrerer de nutidige state-of-the-art algoritmer til frekvens- eller tidsforsinkelses-estimering. Selvom de foreslåede metoder og algoritmer i denne afhandling er designet til brug i kommunikationssystemer er de også relevante indenfor andre felter.

Contents

Thesis Details

Thesis Title: Compressive Sensing in Communication Systems
Ph.D. Student: Karsten Fyhn
Supervisors: Prof. Søren Holdt Jensen, Aalborg University
Prof. Torben Larsen, Aalborg University

The main body of this thesis consist of the following papers.

[A] **Karsten Fyhn**, Thomas Arildsen, Torben Larsen, Søren Holdt Jensen, "Demodulating Subsampled Direct Sequence Spread Spectrum Signals using Compressive Signal Processing", *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, 2012

[B] **Karsten Fyhn**, Tobias Lindstrøm Jensen, Torben Larsen, Søren Holdt Jensen, "Compressive Sensing for Spread Spectrum Receivers", *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 2334–2343, May 2013

[C] **Karsten Fyhn**, Hamid Dadkhahi, Marco F. Duarte, "Spectral Compressive Sensing with Polar Interpolation", *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, 2013

[D] **Karsten Fyhn**, Marco F. Duarte, Søren Holdt Jensen, "Compressive Time Delay Estimation using Interpolation", *Submitted for the 1st IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Austin, Texas, USA, 2013

[E] **Karsten Fyhn**, Marco F. Duarte, Søren Holdt Jensen, "Compressive Parameter Estimation for Sparse Translation-Invariant Signals Using Polar Interpolation", *Submitted for publication in IEEE Transactions on Signal Processing*, 2013

This thesis has been submitted for assessment in partial fulfillment of the PhD degree. The thesis is based on the submitted or published scientific papers which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

Preface

This thesis is submitted to the Doctoral School of Engineering and Science at Aalborg University in partial fulfilment of the requirements for the degree of Doctor of Philosophy. The work was carried out in the period from August 2010 to July 2013 at the Department of Electronic Systems at Aalborg University.

The topic of the thesis is the recently developed sampling theory of compressive sensing and its application in communication systems. The thesis is split in two parts: First, an introduction is given, which summarizes the general theory of compressive sensing and describes the challenges in communication systems that compressive sensing might alleviate. Second, a number of published and submitted papers are presented, which represents the main body of work in this thesis. The two first papers concern compressive sensing for spread spectrum communication systems and the three last papers concern parameter estimation using compressive sensing.

I would like to thank my two supervisors, Professor Søren Holdt Jensen and Professor Torben Larsen, for their help and support during my PhD. I am also very grateful for the supervision and inspiration given to me by Assistant Professor Marco F. Duarte from University of Massachusetts Amherst, USA, whom I visited for five months in the Fall of 2012. He has been a great inspiration to my work and a never-ending source of insight and support. It was under his supervision that I worked on parameter estimation using compressive sensing.

I would also like to thank the other members of the SparSig project at Aalborg University: Assistant Professor Thomas Arildsen, Postdocs Tobias Lindstrøm Jensen and Jacek Pierzchlewski and fellow Ph.D.-students Pawel Pankiewicz, Peng Li, Ruben Grigoryan and Hao Shen. Thank you for always being prepared for a quick discussion about a problem and for being part of my social life at the university.

Last and foremost, I would especially like to thank my wife Kirstine for her love, support and encouragement through these three years.

Karsten Fyhn
Aalborg University, July 31, 2013

Part I

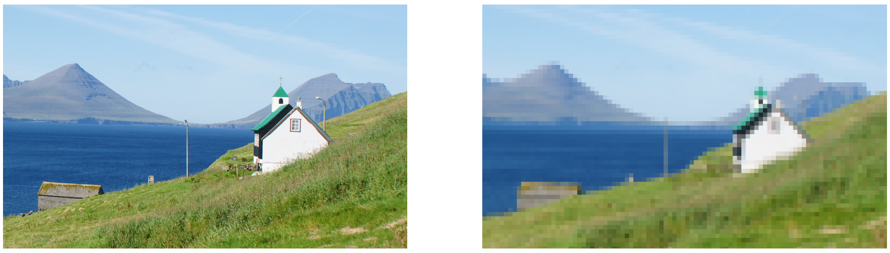
Compressive Sensing in Communications Systems

Chapter 1

Introduction and Thesis Overview

The world around us is an analog one. Most natural phenomena we can analyze may be in an infinite number of states, depending on temperature, density, velocity, voltage, etc. However, our processing platform of choice is often a digital unit, which cannot represent an infinite number of states. Hence we must *sample and quantize* natural phenomena. Sampling is a transition from an analog signal to a digital signal. From a continuous signal with an infinite number of possible values to a finite number of samples. This sampling is achieved using an Analog-to-Digital Converter (ADC), which is an electronics component present in any digital device that process an analog input signal. Furthermore, the ADC must quantize the input signal, meaning that every sample must be represented by a value from a finite set of possible values. To visualize this see Fig. 1.1 where the picture to the left is a close approximation of an analog picture — as humans we cannot clearly discern the discretization. However, the picture is actually just a refined version of the figure to the right. There we see the discrete matrix of colored squares or pixels, representing the actual image representation. Each pixel is only allowed to take on values from a finite set of possible colors. This is an example of how ADCs take measurements of an analog process and discretize it. In most cases a good approximation of the analog process is sufficient, so the ADC should take enough measurements, such that the amount of obtained information is sufficient for the task at hand. Similarly, ADCs are used to acquire speech or audio signals, communication signals or any other type of signal.

The ADC technology is very mature and has become a key corner stone in our infrastructure. However, the technology is limited with respect to some key parameters. There are limits to how fast an ADC may sample a signal and how well each sample may be quantized, i.e. how large a set each sample value may be represented from. There



(a) How we see an image. **(b)** How the image is actually composed.

Fig. 1.1: Example of sampling process.

are also other parameters but in this work, the focus is on these two. As described in [55, 98] the sample and quantization rate is improving but not as fast as the industry needs it. Especially Ultra-Wideband (UWB) radio may require very high sampling rate in the gigahertz range. Even though it may be possible to implement such ultra high sampling ADCs, such devices are expensive and consume a lot of energy, compared to similar devices with lower sampling rate and quantization rate [53]. To determine the necessary sample rate of a system the celebrated Shannon-Nyquist theorem is used, which gives a bound on the necessary sample rate to enable exact reconstruction of an analog signal from its discrete samples. However, the Shannon-Nyquist theorem, which is defined more clearly in the next section, is a very broad theorem, which holds for all bandlimited signals. If more a-priori knowledge about the signal is available it is possible to reduce the sample rate further. In [95] the authors show that it is possible to lower the necessary sampling rate and still attain a desired level of information if the *information rate* is much lower than the actual signal dimensionality. In this work the newly developed theory of Compressive Sensing (CS) [17, 30] is used to lower the sampling rate, while still attaining the relevant information in a given signal. CS may be used if the desired signals may be assumed *sparse* in some domain, which is the case in many scenarios.

In this work the focus has been on attacking some of the problems currently relevant in the CS field of research and on using the developed techniques for communication systems. The published and submitted papers show how CS may be used to lower the necessary sample rate in two different areas in communication: 1) spread spectrum systems and 2) parameter estimation which may be used for e.g. UWB-based communication and Frequency Modulated (FM) signals. We propose a new hardware structure for spread spectrum signals, highlight the problem of noise folding and demonstrate that quantization may make this problem less significant. Furthermore, several algorithms are proposed for parameter estimation for the class of translation-invariant signals, which outperform the current state-of-the-art algorithms for frequency and time delay estima-

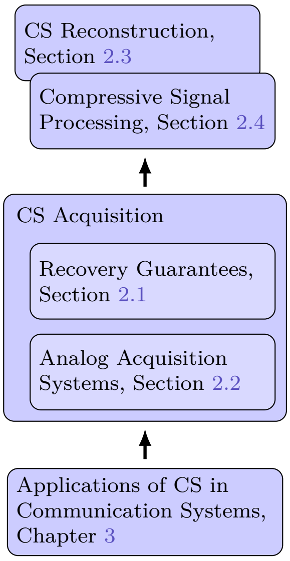

CS Reconstruction, Section 2.3
Compressive Signal Processing, Section 2.4
CS Acquisition
Recovery Guarantees, Section 2.1
Analog Acquisition Systems, Section 2.2
Applications of CS in Communication Systems, Chapter 3

Fig. 1.2: Thesis chapters overview. The arrows signify the processing flow, from the various communication signal applications, to the acquisition of the analog signal and finally the reconstruction or processing of the acquired signal.

tion, when the signal of interest is generated with a parametric signal model, rather than a dictionary-based one.

An overview of the structure of the two main introduction chapters is shown in Fig. 1.2. The remainder of this chapter rounds off with a small review of the Shannon-Nyquist sampling theorem, followed by a definition of the mathematical notation used in this work. In Chapter 2 an extensive summary of the main points in CS theory up until now is given. CS theory mainly consists of two aspects: signal acquisition and signal reconstruction and/or processing. Signal acquisition is split in two parts: one that treats the mathematical foundation for stable signal acquisition and the other which treats actual hardware structures for acquiring a signal. At the end of Chapter 2 some of the current challenges in CS which are addressed in this work are discussed. Then, in Chapter 3, examples of the use of CS in communications are given, especially for the two use cases listed in the above. The novelty and contribution of the individual papers presented in this Ph.D. thesis are listed in Chapter 4 and are summarized as follows:

- **Paper A and Paper B** concern the application of CS to spread spectrum systems for data decoding. The main contribution of these papers is the discovery that for spread spectrum systems it is possible to simplify the hardware front end. We propose the general Compressive Spread Spectrum (CSS) hardware design.

Another important contribution of these papers is that the numerical experiments indicate that taking quantization into account may remedy the drop in performance induced by noise folding.

- **Paper C, Paper D and Paper E** concern parameter estimation. We show that an interpolation scheme based on polar interpolation outperforms other state-of-the-art frequency and time delay estimation algorithms. The main novelty and contribution of the papers are a hybrid greedy and convex optimization based algorithm and the discovery in the numerical experiments that even though the reconstruction of a signal suffers from noise folding, the estimation of its parameters are not as heavily affected.

Finally, the paper is concluded in Chapter 5 and some possible directions for future work is given.

1.1 Shannon-Nyquist Theorem

What is denoted here as the Shannon-Nyquist Theorem was actually derived by several people, including Shannon and Nyquist. For more historical background on the theorem see [62]. Let $f(t)$ be a time domain function which is bandlimited in the frequency domain. Fourier analysis tells us that any such frequency domain function may be represented as a sum of exponentials, i.e. its Fourier series. This is the centerpiece in the theorem published by Shannon on the necessary sample rate of analog signals: *If a function $f(t)$ contains no frequencies higher than W Hertz, it is completely determined by giving its ordinates at a series of points spaced $1/(2W)$ seconds apart.* [85]

The proof of this theorem relies on $f(t)$ being bandlimited in frequency. Let $F(\omega)$ be the spectrum of $f(t)$ so that:

$$f(t) = \frac{1}{2\pi}\int_{-W}^{W} F(\omega)e^{j\omega t}\mathrm{d}\omega, \tag{1.1}$$

because $F(\omega)$ is zero outside $[-W, W]$. Then, define the sampling instants as:

$$t = \frac{n}{2W},\ n \in \mathbb{Z}. \tag{1.2}$$

The samples of $f(t)$ then becomes:

$$f\left(\frac{n}{2W}\right) = \frac{1}{2\pi}\int_{-W}^{W} F(\omega)e^{j\omega\frac{n}{2W}}\mathrm{d}\omega \tag{1.3}$$

With this formulation the right hand side may be recognized as the Fourier series expansion of $F(\omega)$. This means that the samples $f\left(\frac{n}{2W}\right)$ are identical to the Fourier coefficients in the expansion. These coefficients almost always uniquely represent the function and therefore retain all the information in the signal.

However, it can be shown that one must actually use a *higher* sampling rate than stated by Shannon, due to a few special cases. Define the signal:

$$f(t) = e^{j2\pi W_\mathrm{c} t}, \tag{1.4}$$

and let $W_\mathrm{c} = \frac{W_\mathrm{s}}{2}$ and $T_\mathrm{s} = \frac{1}{W_\mathrm{s}}$. Then:

$$f(nT_\mathrm{s}) = e^{j2\pi \frac{W_\mathrm{s}}{2} n \frac{1}{W_\mathrm{s}}} = e^{j\pi n} = (-1)^n \tag{1.5}$$

With this sampling frequency the ability to distinguish between $-W_\mathrm{c}$ and W_c is lost. If the signal is real, e.g. a sinusoid with a given frequency W_c and $W_\mathrm{c} = \frac{W_\mathrm{s}}{2}$ it may also happen that the signal is sampled exactly at the zero-crossings, which would lead to the zero vector as the sampled version of the sinusoid.

Therefore, the Shannon-Nyquist sampling theorem is often given as:

Theorem 1.1
If a bandlimited signal $f(t)$ has non-zero bandwidth B Hertz it is necessary to sample the signal with *more than* $2B$ Hertz.

This theorem is so powerful because of the assumption of bandlimitation. The fact that the signal is bandlimited in the frequency domain does not limit this theorem to be useful for signals analysed only in the Fourier domain. Instead, this bandlimitation only states the *rate of change* allowed in the discrete representation.

Any signal may be analyzed with respect to rate of change. In an image this corresponds to the resolution. The resolution is the measure for how many pixels are dedicated to represent the image. The more pixels dedicated, the better variations in the image are captured. This is one form of rate of change and any type of signal may be analyzed with respect to its rate of change by using the Fourier domain.

If a sampling system violates the Shannon-Nyquist theorem and samples a signal with a lower sampling rate than required, the system will experience *aliasing*, i.e. any frequency content higher than half the sampling rate may get aliases or images at lower and higher frequencies. This happens because when sampling a continuous function $x(t)$, with Fourier transform $X_\mathrm{c}(f)$, the spectrum of the sampled signal $X_\mathrm{s}(f)$ consists of periodically repeated copies of the Fourier transform of $X_\mathrm{c}(f)$. These copies are shifted by integer multiples of the sampling frequency [70]. This is visualized in Fig. 1.3.

It is worth noting that the Shannon-Nyquist theorem is mainly a mathematical theorem. It holds for bandlimited functions that are assumed to be infinite.

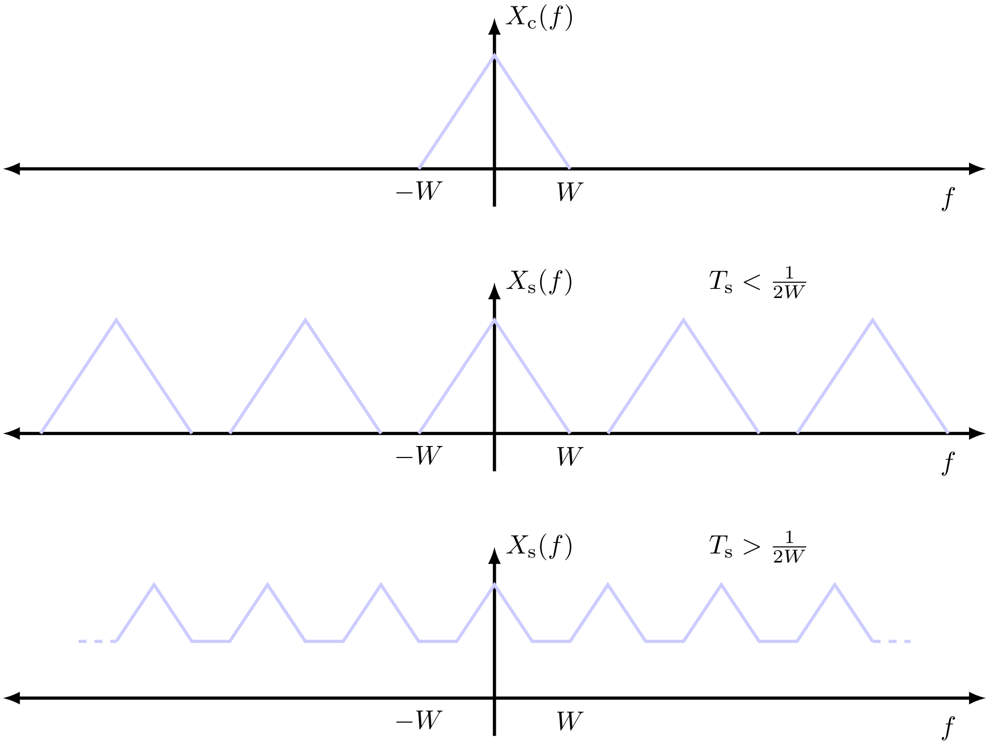


Fig. 1.3: Example of aliasing. In the top figure $X_c(f)$ is the spectrum that is sampled. In the middle figure $X_s(f)$ is the spectrum of the sampled signal when $T_s \leq \frac{1}{2W}$, i.e. follows the Shannon-Nyquist sampling theorem. Notice that the replicated versions are well-separated. In the bottom figure $X_s(f)$ is again the spectrum of the sampled signal, but now $T_s \geq \frac{1}{2W}$, i.e. the Shannon-Nyquist theorem is not followed. Here the spectrum slices are overlapping and aliasing occurs.

1.2 Notation

Before starting the introduction to CS the notation used throughout this work is introduced. Any continuous function is represented by a lower-case letter and its continuous parameters enclosed in parentheses, e.g. $f(t)$, where f is the function name and t is its parameter. For discrete functions brackets are used: $f[t]$, where t is now an index from a discrete set. When denoting the spectrum of a signal $f(t)$ this refers to the Fourier transform $F(W)$ of that signal, unless otherwise stated.

Vectors and matrices are denoted using lower- and uppercase bold letters $\mathbf{x}$ and $\mathbf{X}$, respectively. Individual entries in vectors are written as lowercase letters with a subscript signifying individual indexes: x_i which corresponds to the ith entry of the vector $\mathbf{x}$. Similarly, for matrices, individual column vectors are written as lowercase bold letters with a subscript signifying the column index: $\mathbf{x}_i$ corresponds to the ith column of $\mathbf{X}$. The only exception is when the subscript is used on sets of indices, i.e. $\mathbf{X}_S$ is a matrix composed of the columns from $\mathbf{X}$ corresponding to the indices in the set S and $\mathbf{x}_S$ is a vector composed of the entries from $\mathbf{x}$ corresponding to the indices in the set S. Operations on matrices include the transpose $\mathbf{X}^T$, the Hermitian transpose $\mathbf{X}^H$, the inverse $\mathbf{X}^{-1}$ and the Moore–Penrose pseudo inverse $\mathbf{X}^\dagger$.

Different norm functions are used throughout this work and are defined as a function that associates a length to a given vector in some vector space. Here the notion of ℓ_p-norms is used, which are defined as follows:

$$\|\mathbf{x}\|_p = \left(\sum_{n=1}^{N} |x_n|^p \right)^{1/p} \quad \text{for } p \geq 1 \tag{1.6}$$

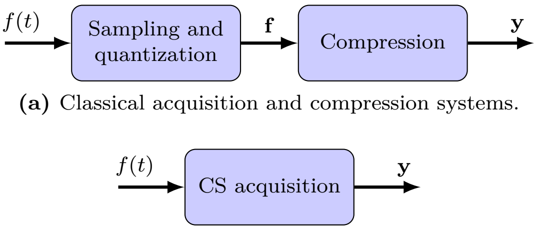
At times a quasinorm such as ℓ_0 is also used, which cannot be used in the above formula, but which corresponds only to counting all the non-zero entries in the vector $\mathbf{x}$.

Vectors and matrices are defined as possible points in vector spaces such as $\mathbf{x} \in \mathbb{R}^N$ for a vector in a real N-dimensional vector space or $\mathbf{X} \in \mathbb{C}^{N \times M}$ as a matrix in the complex vector space with dimensions $N \times M$.

Chapter 2

Compressive Sensing

Compressive Sensing (CS) is a relatively new field in signal processing. It was first proposed by Candes et al. [17] and Donoho [30] in 2004 and represents a new paradigm for sampling signals. The problem that CS attacks is based on the fact that many current signal acquisition systems consists of two essential, sequential steps: acquisition and compression, see Fig. 2.1. First an analog signal $f(t)$ is acquired by using some *sampling kernel* such as a matched filter or a Dirac comb, resulting in the Nyquist sampled $\mathbf{f}$. In this work sampling kernels are always linear functions of some input data. If the signal of interest is highly oscillatory the amount of acquired data may become difficult to acquire, store and process quickly. Also, in many applications the information rate of the signal is actually much lower than the Nyquist rate mandated by the Shannon-Nyquist theorem [95]. Therefore, compression often follows acquisition so that only the desired information is extracted from the signal, stored and processed, here denoted by the measurements $\mathbf{y}$. In CS the acquisition and compression step is combined into one step, hence the name: *compressive sensing*. This is often stated



(a) Classical acquisition and compression systems.

(b) Compressive sensing acquisition.

Fig. 2.1: Classical and compressive sensing acquisition and compression systems.

mathematically as:

$$\mathbf{y} = \mathbf{A}\mathbf{f}, \tag{2.1}$$

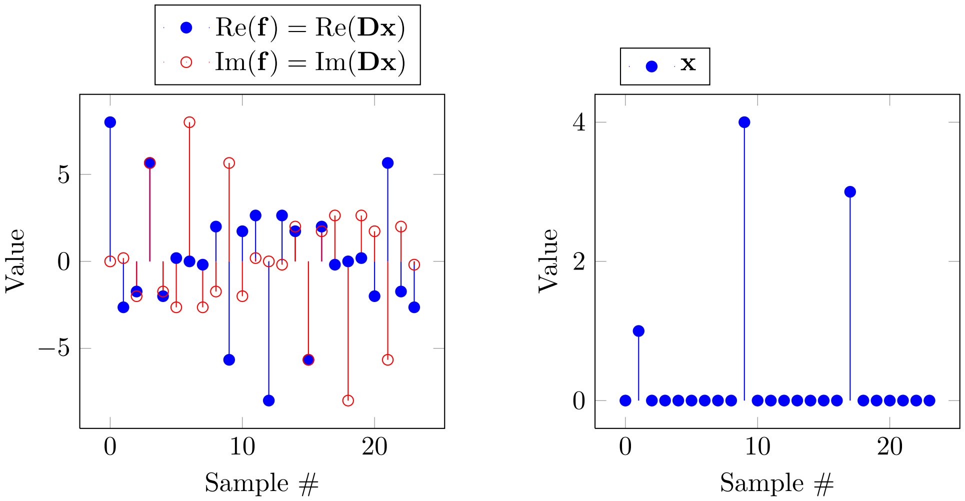
where $\mathbf{y} \in \mathbb{C}^{M\times 1}$ is the acquired, compressed measurement vector, $\mathbf{A} \in \mathbb{C}^{M\times N}$, $M \leq N$ denotes a measurement matrix or sampling kernel and $\mathbf{f} \in \mathbb{C}^{N\times 1}$ signifies the signal of interest sampled at the Nyquist rate. If $M < N$ $\mathbf{A}$ is dimensionality reducing or compressing. Here it should be noted that the notion of compression in CS is not always identical to the classical notion of compression in information theory. In information theory compression relates to the number of bits required to store the signal and methods for encoding such that a signal may be represented using as few bits as possible. The encoding may be either lossy or lossless. In CS the compression is often a dimensionality reduction, rather than a reduction in the number of bits. However, some recent work has merged CS and quantization to reduce the number of bits as much as possible [14, 15, 52].

The measurement matrix should model the analog front end and the sampling operation by the ADC, i.e. an analog component that outputs discrete measurements. To be able to reconstruct the actual transmitted signal, the task for an RF engineer is therefore to design a dual pair consisting of an analog front end and a discrete measurement matrix $\mathbf{A}$, such that the analog front end measures the signal so that the output discrete measurements $\mathbf{y}$ correspond to the result of the matrix-vector operation with $\mathbf{A}$ in Eqn. (2.1). If this is achieved, it is possible to reconstruct the Nyquist-rate sampled version of $f(t)$, $\mathbf{f}$, which is known by the Shannon-Nyquist theorem to contain all the information from the analog signal. An example is the matched filter, which is often used in communications when the transmitted signal is known, but its position in time is not. Then the analog implementation is filters that implement the matched filter followed by an ADC, while the measurement matrix consists of rows, where each row is the discrete version of the matched filter shifted in time, corresponding to the sampling rate of the ADC in the analog implementation.

In many classical systems the measurement matrix $\mathbf{A}$ is square and invertible. The CS measurement matrix $\mathbf{A}$ has fewer rows than columns, may be rank-deficient and does not constitute an isometry for all input vectors $\mathbf{x}$. However, *near-isometry* is possible if only a subset of input vectors is allowed. In CS this is achieved by assuming sparsity in some domain. Classically, the sparsity assumed by CS is dictionary-based, identical to what is known as transform coding in compression theory. An example of this is the following: Assume a signal $\mathbf{f}$ must be compressed. Often a signal may be represented as:

$$\mathbf{f} = \mathbf{D}\mathbf{x}, \tag{2.2}$$

where $\mathbf{D} \in \mathbb{C}^{N\times N}$ denotes a dictionary matrix and $\mathbf{x} \in \mathbb{C}^{N\times 1}$ is the transform representation of $\mathbf{f}$. If $\mathbf{x}$ is sparse for some dictionary $\mathbf{D}$ and for all possible choices of $\mathbf{f}$, i.e. only $K \ll N$ entries are non-zero, the signal $\mathbf{f}$ may be compressed by only storing



(a) The signal in the time domain.

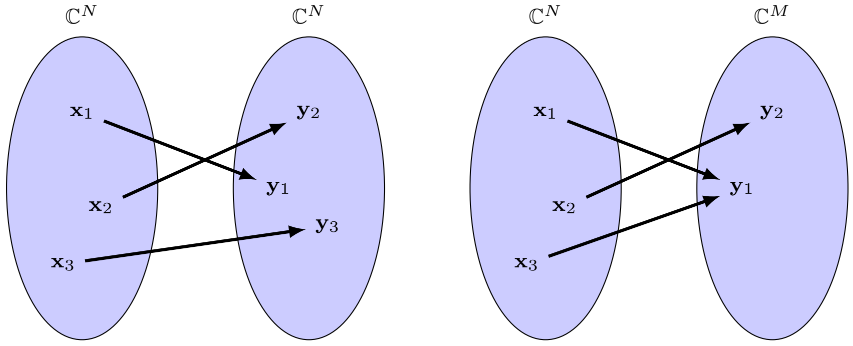
(b) The signal in the DFT domain.

Fig. 2.2: Example of transform coding with an inverse DFT dictionary matrix **D**.

the non-zero entries and their positions in the sparse transform domain representation **x**. This is the fundamental principle behind many forms of compression, e.g. JPEG for images. An example of transform coding for an inverse DFT dictionary matrix **D** is shown in Fig. 2.2. Classical CS requires this form of sparsity, but other techniques exist to handle cases with other kinds of sparsity, e.g. manifold models [9] or parametric models with a few parameters [44].

CS comprises methods for acquiring signals that are sparse in some known domain, such that the measurement matrix **A** constitutes an near-isometry for the class of sparse signals. However, Eqn. (2.1) only concerns the acquisition of the signal. Often there will also be a reconstruction stage. With the Shannon-Nyquist theorem the reconstruction is achieved by sinc interpolation which is a relatively simple operation. In CS one must instead solve an underdetermined system of linear equations, which may be assumed sparse. This is done using non-linear methods, but this may be a very heavy computational task.

The following sections first elaborates on the requirements on the measurement matrix **A** for CS to guarantee a unique mapping from the signal domain to the measurements domain, i.e. **A** must represent a *stable embedding*. Then a review of current dual pairs of analog front ends and discrete measurement matrices **A** that satisfy these requirements are given. This is followed by an overview of different reconstruction algorithms used for recovering the original Nyquist rate signal from the compressed measurements. Then the related field of *Compressive Signal Processing* is introduced,



(a) Linear operator with full column and row rank.

(b) Linear operator with less than full column rank and non-empty null-space.

Fig. 2.3: Linear operation by matrix $\mathbf{A}$.

where the signal is not reconstructed but instead processed directly in the compressed domain. Finally some of the current challenges in CS which are addressed in this collection of papers are described, e.g. quantization, noise folding and signals that are not sparsely representable in any specific dictionary matrix, but instead is sparse in some other model.

2.1 Recovery Guarantees

In this section the requirements for the measurement matrix $\mathbf{A}$ to form a near-isometry for sparse signals are examined. However, before talking of CS measurement matrices, that are underdetermined and constitute an ill-posed problem, a review is given of the properties of a well-defined system of linear equations: $\mathbf{y} = \mathbf{A}\mathbf{x}$, $\mathbf{A} \in \mathbb{C}^{N \times N}$. In such cases, if $\mathbf{y}$ and $\mathbf{A}$ are given and $\mathbf{A}$ has full rank, $\mathbf{x}$ is obtained by solving the system of linear equations, which is equivalent to multiplying $\mathbf{y}$ with the inverse of $\mathbf{A}$ denoted $\mathbf{A}^{-1}$. In this section the dictionary matrix is assumed to be the identity matrix $\mathbf{D} = \mathbf{I}$ or that the measurement matrix is the matrix-matrix product of the measurement matrix and the dictionary matrix. That the matrix $\mathbf{A}$ has full rank also means that its columns are linearly independent and span an N dimensional subspace. Therefore, the matrix $\mathbf{A}$ may be used to represent all possible signals in $\mathbb{C}^N$ and each signal has one unique representation by $\mathbf{A}$, i.e. $\mathbf{A}$ constitutes a bijective function. This is visualized in Fig. 2.3a where any vector $\mathbf{x}$ in $\mathbb{C}^N$ is mapped to a unique vector $\mathbf{y}$ also in $\mathbb{C}^N$. Such problems are often termed well-posed when they satisfy three conditions:

- A solution to the problem exists,

- The solution is unique, and
- Small changes in the input results only in small changes in the output.

When a problem does not fulfill these requirements it is termed ill-posed. Ill-posed problems are often encountered in inverse problems and is often separated into two groups: underdetermined and overdetermined systems of linear equations. In overdetermined systems of linear equations there are more equations than unknowns, which means there may be no solution to the problem, whereas in underdetermined systems of linear equations there are fewer equations than unknowns, so that there are potentially infinitely many solutions. Overdetermined problems may be solved using least squares to find the best fit of the data. Underdetermined problems, i.e. when the matrix $\mathbf{A}$ is fat and has more rows than columns has potentially infinitely many solutions. This is visualized in Fig. 2.3b, where the point $\mathbf{y}_1$ has two possible solutions: $\mathbf{x}_1$ and $\mathbf{x}_3$. This is the problem addressed by CS.

Let $\mathcal{N}(A) = \{\mathbf{z} : \mathbf{A}\mathbf{z} = \mathbf{0}\}$ denote the null space of $\mathbf{A}$. To ensure recovery of all possible signals $\mathbf{x} \in \mathbb{C}^N$ from the measurements $\mathbf{y} = \mathbf{A}\mathbf{x}$ it is important that for any pair of vectors $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{C}^N$, $\mathbf{x} \neq \hat{\mathbf{x}}$ their images must also be different, i.e. $\mathbf{A}\mathbf{x} \neq \mathbf{A}\hat{\mathbf{x}}$. Otherwise it is impossible to uniquely return to the original signal from the measurements. This can be defined using the null space by noting that if $\mathbf{A}\mathbf{x} = \mathbf{A}\hat{\mathbf{x}}$ then $\mathbf{A}(\mathbf{x} - \hat{\mathbf{x}}) = \mathbf{0}$, i.e. the signal $\mathbf{x} - \hat{\mathbf{x}}$ is in the null space of $\mathbf{A}$. If it can be ensured that no possible set $\mathbf{x}, \hat{\mathbf{x}}$ exists, such that $\mathbf{A}(\mathbf{x} - \hat{\mathbf{x}}) = \mathbf{0}$, unique recovery is ensured.

If restrictions are put on the possible input vectors $\mathbf{x}$ it is possible to make an underdetermined system well-posed for that limited set of input vectors. This is the basis for CS. First, assume that $\mathbf{x} \in \mathbb{C}^N$ is exactly K-sparse or less, i.e. only K or less out of the N entries of $\mathbf{x}$ are nonzero. Let $\Lambda_K = \{\mathbf{x} : \|\mathbf{x}\|_0 \leq K\}$ signify the set of K-sparse vectors and define the Spark of a matrix as:

Definition 2.1 (Reproduced from Definition 1.1 in [41])
The spark of a given matrix $\mathbf{A}$ is the smallest number of columns of $\mathbf{A}$ that are linearly dependent.

Then the condition on the null space may be relaxed to only hold for vectors in Λ_K and it can be shown that:

Theorem 2.1 (Reproduced from Corollary 4 in [31])
For any vector $\mathbf{y} \in \mathbb{C}^M$, there exist at most one signal $\mathbf{x} \in \Lambda_K \subset \mathbb{C}^N$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}$ if and only if $\text{Spark}(\mathbf{A}) > 2K$.

Proof. The assumption that $\text{Spark}(\mathbf{A}) > 2K$ means that for any $\mathbf{h} = \mathbf{x} - \hat{\mathbf{x}}$, $\mathbf{x}, \hat{\mathbf{x}} \in \Lambda_K$ $\mathbf{h}$ can at most have $2K$ nonzero entries and hence cannot pick any linearly dependent set of columns from $\mathbf{A}$. Therefore $\mathbf{h}$ cannot be in the null space of $\mathbf{A}$ unless $\mathbf{h} = \mathbf{0}$, which implies $\mathbf{x} = \hat{\mathbf{x}}$. □

Since $\text{Spark}(\mathbf{A}) \in [2, M + 1]$ there is a restriction on the minimum number of measurements $M \geq 2K$ for K-sparse signals using Theorem 2.1.

The above holds only for exactly K-sparse signals, which cannot always be assumed. Instead the signals will be *compressible*, meaning that the signal is well approximated by a K-sparse signal. To treat such vectors the *null space property* is used. First, define two sets: a subset of indices $\Lambda \subset \{1, 2, \ldots, N\}$ corresponding to the non-zero entries in a vector and the remaining indices $\Lambda_c \subset \{1, 2, \ldots, N\} \backslash \Lambda$. Hence, when writing $\mathbf{x}_\Lambda$ this signifies a length N vector with all entries indexed by Λ_c set to zero. Then define the following:

Definition 2.2 (Reproduced from Definition 1.2 of [41])
A matrix $\mathbf{A}$ satisfies the null-space property (NSP) of order K if there exists a constant $C > 0$ such that,

$$\|h_\Lambda\|_2 \leq C \frac{\|h_{\Lambda_c}\|_1}{\sqrt{K}} \tag{2.3}$$

holds for all $h \in \mathcal{N}(\mathbf{A})$ and for all Λ such that $|\Lambda| \leq K$.

The NSP ensures that vectors in the null space of $\mathbf{A}$ do not have their energy concentrated on only a few entries. One important example to note is the NSP for K-sparse signals in the null space of $\mathbf{A}$. If $\mathbf{h}$ is exactly K-sparse then $\|h_{\Lambda_c}\|_1 = 0$. This implies that $\mathbf{h}_\Lambda = 0$ as well, due to Eqn. (2.3). This means that if $\mathbf{A}$ satisfies the NSP the only exactly K-sparse vector in $\mathcal{N}(\mathbf{A})$ is $\mathbf{h} = \mathbf{0}$. However, this still does not show why the NSP is important for general compressible signals. The NSP is used to estimate performance of reconstruction algorithms for general non-sparse signals. Define the following:

$$\|\Delta(\mathbf{A}\mathbf{x}) - \mathbf{x}\|_2 \leq C \frac{\min\limits_{\hat{\mathbf{x}} \in \Sigma_K} \|\mathbf{x} - \hat{\mathbf{x}}\|_1}{\sqrt{K}} \tag{2.4}$$

for all $\mathbf{x}$ and where $\hat{\mathbf{x}} \in \Sigma_K \subset \mathbb{C}^N$ is any exactly K-sparse vector and $\Delta(\cdot) : \mathbb{C}^M \to \mathbb{C}^N$ is a reconstruction algorithm. Then the following is true:

Theorem 2.2 (Reproduced from Theorem 3.2 in [22])
Let $\mathbf{A} : \mathbb{C}^N \rightarrow \mathbb{C}^M$ denote a measurement matrix and $\Delta : \mathbb{C}^M \rightarrow \mathbb{C}^N$ denote a specific reconstruction algorithm. If the pair $(\mathbf{A}, \Delta)$ satisfies Eqn. (2.4) then $\mathbf{A}$ satisfies the NSP of order $2K$.

This theorem then sets a bound on the error by an arbitrary reconstruction algorithm. The bound ensures exact recovery of exactly K-sparse signals and ensures some robustness for compressible signals that directly corresponds to how well the compressible signal is approximated by an exactly K-sparse signal.

Thus, the NSP tells us something about the bound on the reconstruction error for a given measurement matrix and an arbitrary reconstruction algorithm.

2.1.1 Restricted Isometry Property

The NSP may be used for exactly K-sparse and compressible signals in the noise-free setting. If noise is included in the signal model it is necessary to use another model, which is called the Restricted Isometry Property:

Theorem 2.3 (Reproduced from Definition 1.1 in [18])
A matrix $\mathbf{A}$ satisfies the Restricted Isometry Property (RIP) of order K if there exists a δ_K such that

$$(1 - \delta_K)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta_K)\|\mathbf{x}\|_2^2 \tag{2.5}$$

holds for all $\mathbf{x} \in \Sigma_K$.

This theorem is very important because it means that if a matrix $\mathbf{A}$ can be found that satisfies the RIP of order $2K$ it will approximately preserve the distance between any pair of K-sparse input vectors with respect to some constant δ_K. This is more clear if the RIP is defined using the difference vector $\mathbf{h} = \mathbf{x} - \hat{\mathbf{x}}$ where $\mathbf{x}$ and $\hat{\mathbf{x}}$ are two arbitrary K-sparse vectors in Σ_K. Therefore, $\mathbf{h}$ is $2K$-sparse or less. Using $\mathbf{h}$ the RIP becomes:

$$(1 - \delta_{2K})\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 \leq \|\mathbf{A}\mathbf{x} - \mathbf{A}\hat{\mathbf{x}}\|_2^2 \leq (1 + \delta_{2K})\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 \tag{2.6}$$

Here it is clearly shown that the RIP signifies a distance preservation between any pair of K-sparse vectors before and after multiplication with the $\mathbf{A}$ matrix. This RIP then states that the distances does not change arbitrarily much as long as the matrix satisfies RIP of order $2K$. It is also important to note that if $\mathbf{A}$ satisfies RIP of order K with constant δ_K it also satisfies RIP of any order $K' < K$ with constant $\delta_{K'} < \delta_K$. The constant δ_K is very important and unless it is bounded somehow, the RIP cannot really be used for anything. This bound is further defined in Section 2.3. Using probabilistic analysis it has been shown that many forms of random matrices satisfy the RIP, such as Gaussian random matrices [18].

Furthermore, the RIP may also be used to find a necessary number of measurements to take for the RIP to hold.

Theorem 2.4 (Reproduced from Theorem 3.5 in [25])
Let $\mathbf{A}$ be an $M \times N$ matrix that satisfies the RIP of order $2K$ with constant $\delta_{2K} \in (0, \frac{1}{2}]$. Then

$$M \geq CK \log\left(\frac{N}{K}\right) \tag{2.7}$$

where $C = \frac{1}{2}\log\left(\sqrt{24}+1\right) \approx 0.28$.

2.1.2 An example of the null space condition, NSP and RIP

To better exemplify the effect of the NSP and the RIP an example is given: Define the measurement matrix as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \sqrt{1/2} \\ 0 & 1 & \sqrt{1/2} \end{bmatrix} \tag{2.8}$$

and assume that the signal of interest $\mathbf{x}$ is 1-sparse and that the dictionary matrix is the identity matrix $\mathbf{D} = \mathbf{I}$.

This matrix has $\text{Spark}(\mathbf{A}) = 3$ which is clearly more than $2K$ and hence fulfills Theorem 2.1. The NSP may then be used to find the reconstruction error for any compressible signal, i.e. if the input vector is not exactly 1-sparse, but is instead $\mathbf{x} = \begin{bmatrix} 1 & 0.2 & 0.1 \end{bmatrix}$ then the reconstruction error is upper bounded by:

$$\begin{aligned} \|\Delta(\mathbf{Ax}) - \mathbf{x}\|_2 &\leq C\frac{\min\limits_{\hat{\mathbf{x}}\in\Sigma_K} \|\mathbf{x}-\hat{\mathbf{x}}\|_1}{\sqrt{K}} \\ &\leq C\frac{0.3}{\sqrt{1}} = 0.3C \end{aligned} \tag{2.9}$$

Here the nominator is found as $\min_{\hat{\mathbf{x}}\in\Sigma_K} \|\mathbf{x}-\hat{\mathbf{x}}\|_1 = 0.3$ with $\hat{\mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$ the best sparse approximation of $\mathbf{x}$. This shows that the less sparse the signal is, the worse is the approximation with a K-sparse signal in the nominator at capturing the energy of the signal and the higher is the reconstruction error. For the noisy case the RIP is used to

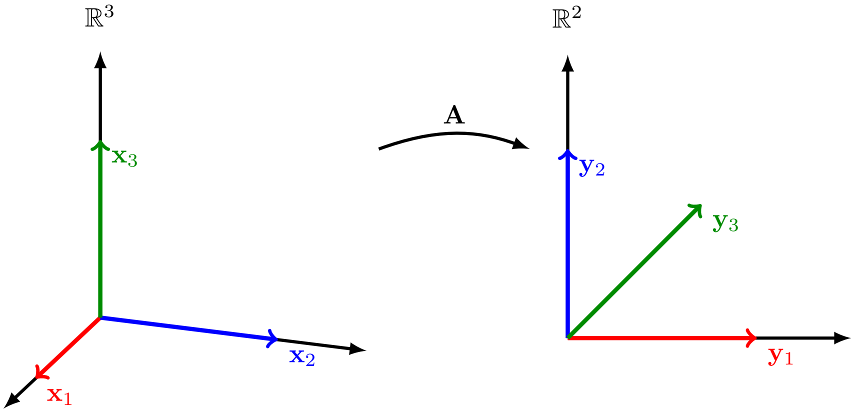


Fig. 2.4: The transform from $\mathbb{R}^3$ to $\mathbb{R}^2$ by $\mathbf{A}$.

verify whether the matrix $\mathbf{A}$ preserves distance for some choice of δ_{2K}. However, even for such a small matrix, this is not possible because it is not possible to test all possible choices of $\mathbf{h} = a\mathbf{x} - b\hat{\mathbf{x}}$. This is not possible as it would require testing for all possible choices of a and b, which are continuous variables. This illustrates the problem with the RIP, in that it is almost always impossible to verify, except through probabilistic analysis. The transform by $\mathbf{A}$ is visualized in Fig. 2.4.

2.1.3 Phase-Transition Plots

While the RIP is an important theoretical result, its practical appliance is not so straight forward, as demonstrated in the above. Verification that a given matrix satisfies the RIP must be done by probabilistic analysis and can often not be done for a specific instantiation, since it would require to test all possible combinations of K-sparse vectors. Furthermore, the RIP has been shown to give a less precise and more conservative boundary between reconstruction success and failure than other bounds, see e.g. the discussions in [11, 32]. Instead, Donoho-Tanner phase-transition or just phase-transition diagrams [32, 33] may be used to demonstrate empirically for which levels of sparsity a specific dictionary and (class of) measurement matrix are applicable. An example of a phase transition plot is shown in Fig. 2.5 which is a figure from Paper B. The parameters of the two-dimensional plot are the subsampling ratio $\delta = M/N$ and the sparsity ratio $\rho = S/M$, where $S = K$. Each curve in the plot is found from a surface plot of the rate of success based on Monte Carlo simulations. In each surface plot, a clear transition curve is evident and to condense the results only the transition curve where the probability of error crosses 0.5 is plotted. Each surface plot is generated so that new simulations are conducted until the mean square error between the ith and

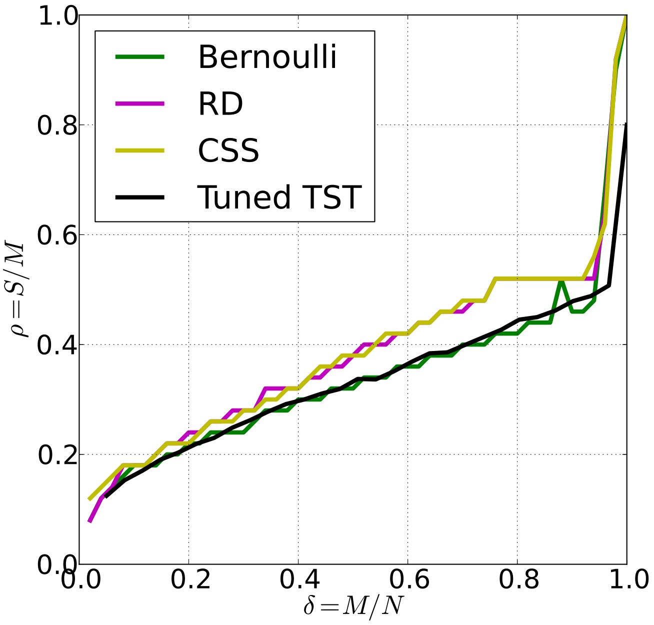


Fig. 2.5: Phase Transition Diagram from Paper B for three different measurement matrices (Rademacher, Random Demodulator and Compressive Spread Spectrum measurement matrix). The black line is the phase transition line for the Tuned Two Stage Thresholding (TST) algorithm from [60].

the $(i-1)$th plot is less than 10^{-5}. For each parameter set and in each simulation, an experiment is a success (1) if the mean squared error between the reconstructed and the received signal is less than 10^{-6} and a failure (0) otherwise. The three measurement matrices used are as follows:

- A Rademacher distributed measurement matrix, with a dense structure where entries are either -1 or 1,
- A Random Demodulator [54, 91] measurement matrix, with a banded structure, where entries are either -1 or 1 on the band and 0 outside, and
- A Compressive Spread Spectrum measurement matrix as proposed in Paper B.

The black line is for validation and is based on data from [60]. For more details on this specific phase transition plot we refer to Paper B. A phase transition plot then signifies two *regions*: a region below and to the right of the transition line, where the reconstruction is successful and a region above and to the left of the transition line, where the reconstruction fails. This may be used to evaluate for which levels of sparsity CS may be used in a given scenario and what is the empirically found lower bound on the number of measurements.

2.1.4 Coherence

Another often used metric for evaluating whether a given matrix or class of matrices are applicable for CS is coherence.

Definition 2.3 (Reproduced from Definition 1.5 in [41])
The coherence of a matrix $\mathbf{A}$, $\mu(\mathbf{A})$, is the largest absolute inner product between any two columns $\mathbf{a}_i, \mathbf{a}_j$ of $\mathbf{A}$:

$$\mu(\mathbf{A}) = \max_{1 \le i < j \le N} \frac{|\langle \mathbf{a}_i, \mathbf{a}_j \rangle|}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2}. \tag{2.10}$$

Using this definition the following upper bound on the sparsity is defined:

Theorem 2.5 (Reproduced from Theorem 12 in [31])
If

$$K < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{A})}\right) \tag{2.11}$$

then for each measurement vector $\mathbf{y} \in \mathbb{C}^M$ there exists at most one signal $\mathbf{x} \in \Sigma_K \subset \mathbb{C}^N$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}$.

This cannot directly give any notion of how many measurements must be taken, but does contain information about the allowable signals for a given matrix $\mathbf{A}$. The important aspect of this theorem is that the columns of $\mathbf{A}$ should be as incoherent as possible.

When using coherence in the example from Section 2.1.2 the bound on sparsity is:

$$K \le \frac{1}{2}\left(1 + \frac{1}{0.7071}\right) = 1.207, \tag{2.12}$$

which again shows that the measurement matrix $\mathbf{A}$ in that example constitutes a stable embedding.

The RIP and coherence is connected by the following theorem:

Theorem 2.6 (Reproduced from Lemma 1.5 in [41])
If $\mathbf{A}$ has unit-norm columns and coherence $\mu = \mu(\mathbf{A})$, then $\mathbf{A}$ satisfies the RIP of order K with $\delta_K = (K-1)\mu$ for all $K < 1/\mu$.

2.2 Analog Acquisition Systems

Until now the measurement system $\mathbf{y} = \mathbf{Ax}$ or $\mathbf{y} = \mathbf{Af}$ has been modelled by the measurement matrix $\mathbf{A}$, a discrete linear operator on the Nyquist rate sampled signal $\mathbf{f}$. However, the main benefit of CS is to reduce the necessary sampling rate of a system and hence, the Nyquist-rate sampled signal should never be directly acquired. Instead, analog front end designs are developed that directly maps the analog signal to a subspace that may be sampled at a low rate (compared to the Nyquist rate). In this section two CS analog sampling structures are reviewed:

- Random Demodulator
- Modulated Wideband Converter

However, first the mathematical framework for analog to digital conversion is introduced. Let $f(t)$ be an analog signal of interest and let $h_i(t)$ denote a *sampling kernel function* corresponding to the ith measurement of $f(t)$. Then discrete measurements $\mathbf{y} = [y_1, y_2, \ldots, y_M]$ are obtained as:

$$y_i = \int_{-\infty}^{\infty} f(t)h_i(t)\mathrm{d}t \tag{2.13}$$

Two examples of sampling kernels is the Dirac comb and the matched filter. The Dirac comb sampling kernel is defined as:

$$y_i = \int_{-\infty}^{\infty} f(t)\delta(t - iT_s)\mathrm{d}t \tag{2.14}$$

where $T_s = 1/f_s$ with f_s the sampling frequency at more than Nyquist rate. This corresponds to sampling according to the Nyquist rate and the Shannon-Nyquist theorem states that with this form of sampling the infinite, perfectly bandlimited analog signal is reduced to a finite representation without losing any information. If the signal is not perfectly bandlimited, the reconstruction experiences aliasing, which means that some energy from the higher frequencies above half the sampling frequency is brought down into the signal bandwidth. Therefore it is important to filter the signal before sampling, such that this energy is small compared to the in-band signal energy.

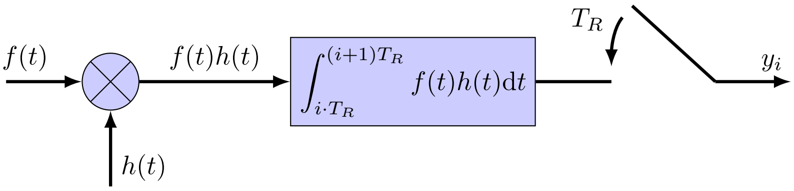


Fig. 2.6: Random Demodulator hardware structure.

In communications systems the matched filter [77] is often employed to maximize the SNR. When the transmitted signal is generated according to a known waveform, often to conform to some standard, the sampling kernel filter should be matched to that waveform, i.e. a time-reverse, conjugated version of the same waveform. In that case the matched filter is the optimal linear filter for maximizing SNR in the presence of additive white Gaussian noise. The matched filter is one way of reducing the sampling rate and is indeed the optimal way if only detection of a known waveform is desired. A single sample over the interval in which the waveform may be transmitted suffices to determine whether the waveform is there or not. We couple the concept of matched filters with CS in Papers A and B.

However, when there are more degrees of freedom, such as which waveform is sent or the waveform may have varying parameters, the matched filter may no longer be the optimal choice and Nyquist sampling is employed instead. However, if the number of degrees of freedom is small compared to the dimensions of the Nyquist-rate sampled signal, CS may enable a lower sampling rate without losing information in the conversion.

To incorporate CS into analog sampling the design task is to construct sampling kernel functions $h_i(t)$ such that the outputs y_i correspond to the linear operator $\mathbf{y} = \mathbf{Af}$ with some measurement matrix $\mathbf{A}$ and $\mathbf{f}$ the Nyquist-rate sampled version of $f(t)$. In the following two examples of such designs are given.

2.2.1 Random Demodulator

One of the first and simplest analog CS front ends is the Random Demodulator (RD) [54, 91]. It is based on multiplying the signal with a known pseudo-random noise sequence $h(t)$, which oscillates between -1 and 1. The output of the multiplier is integrated using a low-pass filter and sampled at a low rate T_R, as shown in Fig. 2.6. In the frequency domain this may be understood as smearing the signal out across the spectrum followed by low-pass filtering. The signal is therefore aliased down into the low end of the spectrum and sampled. In [91] the RD is described mathematically using the two matrices $\mathbf{D}$ and $\mathbf{H}$. First, let $\mathbf{h} = [h_1, h_2, \ldots, h_N] \in \{\pm 1\}^N$ be the chipping sequence sampled at the Nyquist rate of the signal $f(t)$. The linear operation $\mathbf{Df}$ signifies the modulation with the chipping sequence, where $\mathbf{D}$ is the diagonal matrix $\mathbf{D} = \mathrm{diag}\{\mathbf{h}\}$.

Second, the **H** matrix denotes the accumulate-and-dump action performed after mixing. Let M denote the number of samples taken and assume here that M divides N. Then each sample is the sum of N/M consecutive entries of the demodulated signal. The matrix performing this sampling action may therefore be defined as an $M \times N$ matrix, with N/M consecutive unit entries in the rth row starting in column $rN/M + 1$ for each $r = 0, 1, \ldots, M - 1$. An example with $M = 3$ and $N = 6$ is:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}. \tag{2.15}$$

The RD is therefore designed to sample an analog signal, so that in a discrete representation this is the equivalent to:

$$\mathbf{y} = \mathbf{HDf}, \tag{2.16}$$

where **f** is the Nyquist sampled input signal and **y** is the compressively measured output signal. Notice that though **f** refers to the received signal sampled at the Nyquist rate, the signal $f(t)$ is *never actually sampled* at that rate - $f(t)$ is sampled at a lower rate and **f** may then be reconstructed using CS so that it is equal to the vector that would be obtained if the signal had been sampled at the Nyquist-rate.

2.2.2 Modulated Wideband Converter

A problem with RD is that the signal components must lie on the grid specified by the dictionary. If the dictionary elements are not able to sparsely represent **f** the reconstruction performs poorly [21, 36, 65, 69].

This is a general problem for *dictionary-based* CS, i.e. CS acquisition and reconstruction systems that assume the presence of a finite dictionary. These systems assume that the signal of interest lies on the grid specified by the dictionary and if this assumption is violated the performance may be severely degraded.

An alternative CS structure is the Xampling framework [65] which uses a model called Union of Subspaces (UoS) instead of a dictionary. This model assumes that the signal lies in an infinite union of finite-dimensional subspaces. The finite-dimensionality comes from having knowledge of the number of generating parameters and the infinite union is due to those parameters being drawn from continuous intervals.

One example of an analog front end for Xampling is the Modulated Wideband Converter (MWC) [64] shown in Fig. 2.7. This hardware structure has P channels and uses P T_R-periodic signals $h_1(t), h_2(t), \ldots, h_P(t)$ to mix the input signal. These signals are often matched to the signal model of $f(t)$, meaning that they may be adaptive. Furthermore, their analog implementation may be more complex than the pseudo-random noise sequence generator from the RD. After multiplying the input signal with the mixing signals, each channel integrates and samples the signal over a period larger than the Nyquist rate period of the signal.

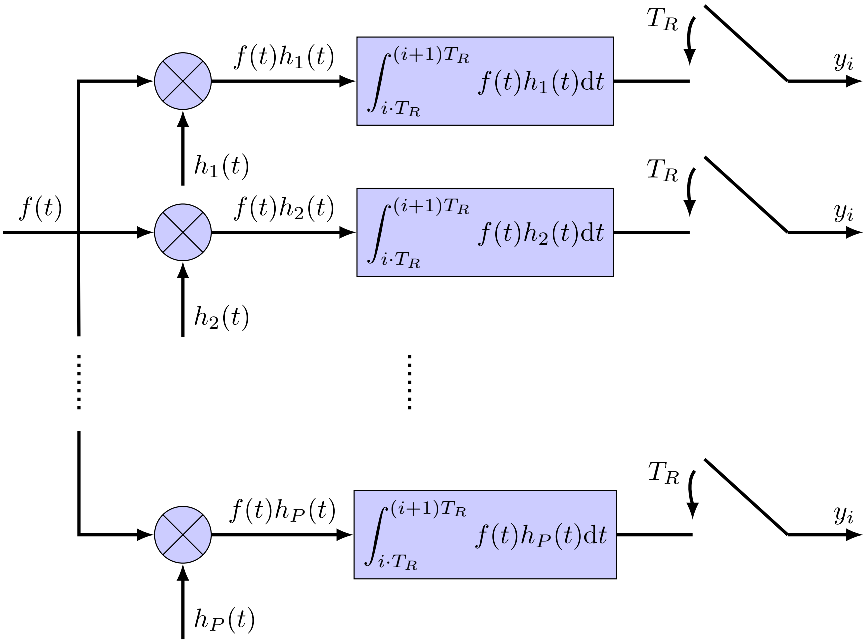

$f(t)$
$f(t)h_1(t)$
$h_1(t)$
$\int_{i \cdot T_R}^{(i+1)T_R} f(t)h_1(t)\mathrm{d}t$
T_R
y_i
$f(t)h_2(t)$
$h_2(t)$
$\int_{i \cdot T_R}^{(i+1)T_R} f(t)h_2(t)\mathrm{d}t$
T_R
y_i
$f(t)h_P(t)$
$h_P(t)$
$\int_{i \cdot T_R}^{(i+1)T_R} f(t)h_P(t)\mathrm{d}t$
T_R
y_i

Fig. 2.7: Modulated Wideband Converter hardware structure.

After acquisition the Xampling framework performs additional signal processing in the digital domain, sometimes without performing reconstruction. If only information or parameter estimation based on the signal is necessary, reconstruction is often not necessary.

The Xampling framework is advantageous when compared to the RD because it does not rely on a grid. However, the RD is a far simpler hardware implementation than the MWC because there is only one channel. A more in-depth comparison between the RD and MWC hardware structures is given in [56]. In our work we use the RD due to its simple structure and we attempt to alleviate some of its shortcomings. In Papers A and B we show that for spread spectrum systems the pseudo-random noise generator is not necessary and in Papers C, D and E we show how interpolation may remedy the problem of off the grid dictionary elements for dictionary-based CS algorithms.

2.3 Reconstruction Algorithms

Similarly to the Shannon-Nyquist theorem, successful acquisition is determined based on the ability to reconstruct the original analog signal from the taken measurements.

However, the actual analog signal is not reconstructed. Instead the Nyquist samples of the signal is reconstructed, since the Shannon-Nyquist theorem states that from these, the full bandlimited analog signal may be reconstructed perfectly.

In CS there are a multitude of reconstruction algorithms that take as input the measurements and some knowledge about the signal or system and then reconstructs the Nyquist samples $\mathbf{f}$. The reconstruction algorithms are often separated into two distinct groups: ℓ_1-minimization using convex optimization and greedy, iterative algorithms. The convex optimization algorithms often offers the best reconstruction performance for the lowest number of measurements, but suffer from potentially heavy computational complexity. The greedy algorithms on the other hand are often very fast and computationally light, but cannot always attain the same performance. In the following the first and most commonly used ℓ_1-minimization problem is described along with four different greedy algorithms: Matching Pursuit, Orthogonal Matching Pursuit, Compressive Sampling Matching Pursuit and Subspace Pursuit.

Another type of reconstruction algorithm, which is gaining fame is message passing algorithms, especially the Approximate Message Passing (AMP) algorithm [34]. These algorithms use belief propagation from graphical models to achieve performance comparable to that of ℓ_1-minimization algorithms, but with the computational complexity comparable to greedy algorithms. Because this field is still fairly new and has not been used in any of the papers supporting this thesis the message passing algorithms are not described in more detail.

For further details and analysis of reconstruction algorithms we refer to [41, 60, 93].

2.3.1 ℓ_1-synthesis and -analysis

As written in Section 2.1 to solve the ill-posed linear system of equations $\mathbf{y} = \mathbf{A}\mathbf{f}$ sparsity may be assumed. In the underdetermined, but sparse case, the following problem is posed:

$$\hat{\mathbf{x}} = \underset{\hat{\mathbf{x}} \in \mathbb{C}^N}{\operatorname{argmin}} \|\hat{\mathbf{x}}\|_0 \text{ s.t. } \mathbf{y} = \mathbf{A}\mathbf{D}\hat{\mathbf{x}} \tag{2.17}$$

Here the ℓ_0-norm objective function is used to enforce sparsity, while the constraint is used to make the model fit the measurements. The ℓ_0-norm is a quasinorm, since it does not fulfill the mathematical definition of an ℓ_p-norm, see Section 1.2. The ℓ_0-norm is of interest because it minimizes exactly the number of nonzero entries. However, with the ℓ_0-norm Eqn. (2.17) is an NP-hard program [67]. Instead, the sparsity-enforcing objective is replaced by the ℓ_1-norm. This norm is convex resulting in a linear programming problem. The incentive for using this norm, rather than e.g. the ℓ_2 norm is because it is also sparsity enforcing, which is demonstrated in Fig. 2.8. The figure demonstrates the behavior of five different ℓ_p norms and quasinorms when minimizing an objective function on the unit sphere in $\mathbb{R}^2$. The red line signifies the image by an underdetermined matrix $\mathbf{A}$. The objective function is then to find the sparsest solution among

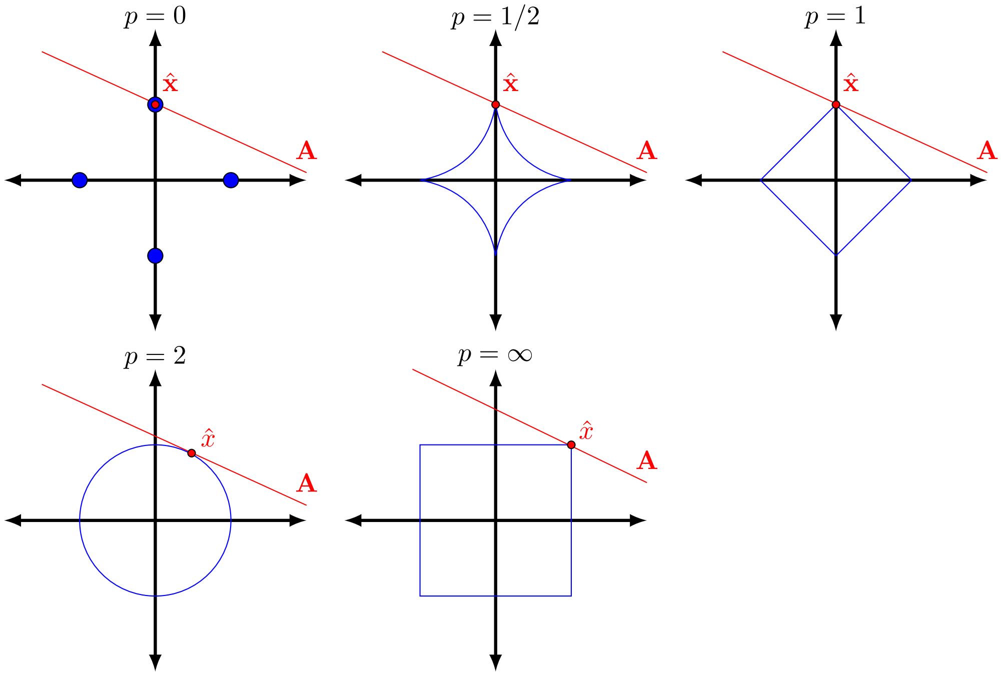


Fig. 2.8: Some ℓ_p norms on the unit sphere in $\mathbb{R}^2$.

the possible solutions on the line by expanding the spheres until they intersect the line. The ℓ_0 norm shows the desired solution — only the sparse solutions are allowed, where the sparse solution is a solution that runs along one of the axes in Fig. 2.8. $\ell_{1/2}$ and ℓ_1 also promote sparsity, whereas ℓ_2 and ℓ_∞ do not find the sparse solution. Since ℓ_1 is the only convex relaxation of the ℓ_0 norm that still promotes sparsity, this is the best choice for CS. The minimization problem to solve for noise-free recovery is therefore:

$$\hat{\mathbf{x}} = \underset{\hat{\mathbf{x}} \in \mathbb{C}^N}{\operatorname{argmin}} \|\hat{\mathbf{x}}\|_1 \text{ s.t. } \mathbf{y} = \mathbf{AD}\hat{\mathbf{x}} \tag{2.18}$$

This optimization problem is denoted Basis Pursuit (BP) [20] and has the following error bound:

Theorem 2.7 (Reproduced from Theorem 1.1 in [19])
Suppose that $\mathbf{A}$ satisfies the RIP of order $2K$ with $\delta_{2K} < \sqrt{2} - 1$ and we obtain

measurements of the form $\mathbf{y} = \mathbf{ADx}$. Then the solution $\hat{\mathbf{x}}$ from Eqn. (2.18) obeys

$$\|\hat{\mathbf{x}} - \mathbf{x}\|_2 \leq C \frac{\min\limits_{\tilde{\mathbf{x}} \in \Sigma_K} \|\mathbf{x} - \tilde{\mathbf{x}}\|_1}{\sqrt{K}}, \tag{2.19}$$

for all $\mathbf{x}$ and where $C = 2\frac{1-(1-\sqrt{2})\delta_{2K}}{1-(1+\sqrt{2})\delta_{2K}}$ and $\hat{\mathbf{x}} \in \Sigma_K \subset \mathbb{C}^N$ is any K-sparse vector.

This is an important theorem that states that it is possible to recover K-sparse signals exactly provided that $\mathbf{A}$ obeys the RIP. Furthermore, it gives a bound on the constant δ_{2K}. In the noisy case BP cannot be used. Instead the following optimization problem is solved:

$$\hat{\mathbf{x}} = \underset{\hat{\mathbf{x}} \in \mathbb{C}^N}{\operatorname{argmin}} \|\hat{\mathbf{x}}\|_1 \text{ s.t. } \|\mathbf{y} - \mathbf{AD}\hat{\mathbf{x}}\|_2^2 \leq \epsilon \tag{2.20}$$

where ϵ is a bound on the noise. Alternatively, the Basis Pursuit Denoising (BPDN) [20] formulation is used:

$$\hat{\mathbf{x}} = \underset{\hat{\mathbf{x}} \in \mathbb{C}^N}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{y} - \mathbf{AD}\hat{\mathbf{x}}\|_2^2 + \lambda\|\hat{\mathbf{x}}\|_1, \tag{2.21}$$

where λ is a sparsity/fidelity trade-off parameter.

There are also worst-case performance bounds on the recovery in the noisy case:

Theorem 2.8 (Reproduced from Theorem 1.1 in [19])
Suppose that $\mathbf{A}$ satisfies the RIP of order $2K$ with $\delta_{2K} < \sqrt{2}-1$ and let $\mathbf{y} = \mathbf{ADx}+\mathbf{w}$ where $\|\mathbf{w}\|_2 \leq \epsilon$. Then the solution $\hat{\mathbf{x}}$ from Eqn. (2.20) obeys

$$\|\hat{\mathbf{x}} - \mathbf{x}\|_2 \leq C_1 \frac{\min\limits_{\tilde{\mathbf{x}} \in \Sigma_K} \|\mathbf{x} - \tilde{\mathbf{x}}\|_1}{\sqrt{K}} + C_2\epsilon, \tag{2.22}$$

for all $\mathbf{x}$ and where $C_1 = 2\frac{1-(1-\sqrt{2})\delta_{2K}}{1-(1+\sqrt{2=})\delta_{2K}}$, $C_2 = 4\frac{\sqrt{1+\delta_{2K}}}{1-(1+\sqrt{2})\delta_{2K}}$ and $\hat{\mathbf{x}} \in \Sigma_K \subset \mathbb{C}^N$ is any K-sparse vector.

The BP and BPDN algorithms and similar convex optimization algorithms are often solved using e.g. CVX [46, 47]. There are other solvers available for specialized problems, but CVX is the most general and most widely used.

2.3.2 Greedy algorithms

An alternative to the ℓ_1 optimization algorithms is the plethora of greedy algorithms for CS. The term greedy is due to the hard decision made by the algorithms each iteration to only include some atoms. In recent years there have been many proposed greedy algorithms, but here the focus is only on a few popular choices:

- Matching Pursuit (MP),
- Orthogonal Matching Pursuit (OMP),
- Compressive Sampling Matching Pursuit (CoSaMP),
- Subspace Pursuit (SP)

The drawback of greedy algorithms is that they are not as easy to derive general performance bounds for. However, they are most often significantly faster and may rival the recovery performance of ℓ_1-minimization algorithms in many cases. Furthermore it is fairly easy to incorporate further structure, beyond dictionary-based sparsity, into the reconstruction with greedy algorithms than it is to do so with convex optimization algorithms [36]. Another significant difference between ℓ_1-based optimization and greedy algorithms is that greedy algorithms often assume known sparsity K. The sparsity may be estimated or assumed known, but this is nonetheless a drawback.

2.3.2.1 Matching Pursuit

The Matching Pursuit (MP) algorithm [61] was the first general algorithm for decomposing signals into linear expansions of waveforms from some dictionary. The algorithm is shown in Algorithm 1. The input to the algorithm is a set of measurements $\mathbf{y}$, a measurement matrix $\mathbf{A}$, a dictionary $\mathbf{D}$ and the size of the linear expansion, i.e. the sparsity K. The first step of the MP algorithm is the same for the other three algorithms as well: generate a *proxy* $\mathbf{p}$ for the signal. A proxy is a rough estimation of the sparse vector $\mathbf{x}$. It is used to choose the atom indexed by i_n from the dictionary that explains the most of the signal in the current iteration. The proxy is the correlation of the signal with each atom in the dictionary. The atom that has the highest absolute correlation is picked out. This atom's approximated linear expansion coefficient p_{i_n} is added to the vector $\hat{\mathbf{x}}$ which is constructed one atom per iteration. After adding the found atom expansion coefficient to $\hat{\mathbf{x}}$ a *residual* is updated. The residual contains the remaining energy of the received signal, after removing the signal components found so far. The residual then becomes the input for the following iteration. This continues until some stopping criterion is met, e.g. when the residual contains less energy than what corresponds to the noise floor. Notice that in the algorithms shown here the measurement and dictionary matrices $\mathbf{A}$ and $\mathbf{D}$ are assumed unit normalized. A drawback of MP is that if the received signal is not exactly sparsely representable on the dictionary grid used in

Algorithm 1 Matching Pursuit algorithm.

```
INPUT: $\mathbf{y}, \mathbf{A}, \mathbf{D}, K$
INITIALIZE: $\mathbf{r}^0 = \mathbf{y}, \hat{\mathbf{x}}^0 = \mathbf{0}, n = 0$
repeat
    $\mathbf{p} = |(\mathbf{AD})^H \mathbf{r}^n|$
    $i_n = \underset{j \in \{1,2,\ldots,N\}}{\text{argmax}} \; p_j$
    $\hat{x}_{i_n}^{n+1} = \hat{x}_{i_n}^{n} + p_{i_n}$
    $\mathbf{r}^{n+1} = \mathbf{r}^n - \mathbf{Ad}_{i_n}$
    $n = n + 1$
until Stopping criterion is met
OUTPUT: $\mathbf{r}^n, \hat{\mathbf{x}}^n$
```

MP, the algorithm may end up picking the same atoms again and again, because each iteration generates a large residual. This means there is no guaranteed convergence.

2.3.2.2 Orthogonal Matching Pursuit

An improvement over MP is the Orthogonal Matching Pursuit (OMP) [27, 28, 73, 92], which generates the estimate of the linear expansion coefficients based on information about all currently found atoms. The *projection* or approximation $\hat{\mathbf{x}}$ is updated every iteration by projecting $\mathbf{y}$ orthogonally onto the columns of $\mathbf{A}$ that belong in the set T^{n+1}. This step means that OMP never picks the same atom in subsequent iterations and that the residual in any given iteration is always orthogonal to all the currently selected atoms. The OMP algorithm is shown in Algorithm 2.

Algorithm 2 Orthogonal Matching Pursuit algorithm.

```
INPUT: $\mathbf{y}, \mathbf{A}, \mathbf{D}, K$
INITIALIZE: $\mathbf{r}^0 = \mathbf{y}, \hat{\mathbf{x}}^0 = \mathbf{0}, T^0 = \emptyset, n = 0$
repeat
    $\mathbf{p} = |(\mathbf{AD})^H \mathbf{r}^n|$
    $i_n = \text{argmax}_{j \in \{1,2,\ldots,N\}} \, p_j$
    $T^{n+1} = T^n \cup i_n$
    $\hat{\mathbf{x}}^{n+1} = (\mathbf{AD}_{T^{n+1}})^\dagger \mathbf{y}$
    $\mathbf{r}^{n+1} = \mathbf{y} - \mathbf{AD}\hat{\mathbf{x}}^{n+1}$
    $n = n + 1$
until Stopping criterion is met
OUTPUT: $\mathbf{r}^n, \hat{\mathbf{x}}^n$
```

2.3.2.3 Compressive Sampling Matching Pursuit

One drawback with MP and OMP is that once an atom has been included into the support set it cannot be removed again, if a better set of coefficients are found. This may be solved if the hard decision to pick only the atom with the highest absolute correlation in each iteration is replaced by a thresholding operation. This type of algorithm is sometimes referred to as a thresholding algorithm instead of a greedy algorithm. One such algorithm is the Compressive Sampling Matching Pursuit (CoSaMP) [68]. In CoSaMP the function $\text{supp}(\mathbf{x}, K)$ is used to find the K absolute largest entries in the vector $\mathbf{x}$. The support set T^n is initialized as the K absolute largest elements from the proxy. Then the support set is updated in two rounds in each iteration: First, the current support set is augmented by the $2K$ absolute largest elements from the proxy generated from the residual. This intermediate support set is used in a least-squares problem, where the measurements are fitted to the atoms corresponding to the $3K$ found strongest coefficients. The remaining coefficients are set to zero, where the notation $\overline{T^{n+0.5}}$ signifies all possible indices *except* those in $T^{n+0.5}$. Based on the approximation of $\hat{\mathbf{x}}$ the support set is again updated to contain only K entries. The CoSaMP algorithm is shown in Algorithm 3.

Algorithm 3 Compressive Sampling Matching Pursuit algorithm.

```
INPUT: $\mathbf{y}, \mathbf{A}, \mathbf{D}, K$
INITIALIZE: $T^0 = \text{supp}(\mathbf{A}^H\mathbf{y}, K), \hat{\mathbf{x}}^0 = \mathbf{0}, n = 0$
repeat
    $\mathbf{p} = (\mathbf{AD})^H(\mathbf{y} - \mathbf{AD}\hat{\mathbf{x}}^n)$
    $T^{n+0.5} = T^n \cup \text{supp}(\mathbf{p}, 2K)$
    $\hat{\mathbf{x}}^{n+0.5}_{T^{n+0.5}} = (\mathbf{AD}_{T^{n+0.5}})^\dagger\mathbf{y},\ \hat{\mathbf{x}}^{n+0.5}_{\overline{T^{n+0.5}}} = \mathbf{0}$
    $T^{n+1} = \text{supp}(\hat{\mathbf{x}}^{n+0.5}, K)$
    $\hat{\mathbf{x}}^{n+1}_{T^{n+1}} = \hat{\mathbf{x}}^{n+0.5}_{T^{n+1}},\ \hat{\mathbf{x}}^{n+1}_{\overline{T^{n+1}}} = \mathbf{0}$
    $n = n + 1$
until Stopping criterion is met
OUTPUT: $\mathbf{r}^n, \hat{\mathbf{x}}^n$
```

2.3.2.4 Subspace Pursuit

Finally, the Subspace Pursuit (SP) [24] algorithm is very similar to CoSaMP but solves two least squares problems per iteration and only takes in K new atoms in each iteration, so that the possible solution set in the first least squares problem is smaller. The SP algorithm was shown in [60] to perform better than the CoSaMP algorithm. The algorithm is shown in Algorithm 4. The stopping criterion used in SP is different from the others and was also proposed in [24]. It guarantees stability for the algorithm even when the RIP does not hold.

Algorithm 4 Subspace Pursuit algorithm.

```
INPUT: $\mathbf{y}, \mathbf{A}, \mathbf{D}, K$
INITIALIZE: $T^0 = \text{supp}(\mathbf{A}^H\mathbf{y}, K), \hat{\mathbf{x}}^0 = (\mathbf{AD}_{T^0})^\dagger\mathbf{y}, n = 0$
repeat
    $\mathbf{p} = (\mathbf{AD})^H(\mathbf{y} - \mathbf{AD}\hat{\mathbf{x}}^n)$
    $T^{n+0.5} = T^n \cup \text{supp}(\mathbf{p}, K)$
    $\hat{\mathbf{x}}^{n+0.5}_{T^{n+0.5}} = (\mathbf{AD}_{T^{n+0.5}})^\dagger\mathbf{y},\ \hat{\mathbf{x}}^{n+0.5}_{\overline{T^{n+0.5}}} = \mathbf{0}$
    $T^{n+1} = \text{supp}(\hat{\mathbf{x}}^{n+0.5}, K)$
    $\hat{\mathbf{x}}^{n+1} = (\mathbf{AD}_{T^{n+1}})^\dagger\mathbf{y}$
    $n = n + 1$
until $\|\mathbf{y} - \mathbf{AD}\hat{\mathbf{x}}^{n-1}\|_2 \leq \|\mathbf{y} - \mathbf{AD}\hat{\mathbf{x}}^n\|_2$
OUTPUT: $\mathbf{r}^{n-1}, \hat{\mathbf{x}}^{n-1}$
```

2.4 Compressive Signal Processing

Most CS literature concerns the reconstruction of the Nyquist-rate sampled signal. However, as was shown in [26] reconstruction is not always desired. In that work the authors introduce the concept of *compressive signal processing*, in which signals are processed in their compressed domain. The authors show how detection, classification, estimation and filtering may be performed in the compressed domain without ever reconstructing the original Nyquist-rate signal.

Detection is defined as the problem of distinguishing between two hypotheses:

$$\begin{aligned} \mathcal{H}_0 &: \mathbf{y} = \mathbf{Am}, \\ \mathcal{H}_1 &: \mathbf{y} = \mathbf{A}(\mathbf{f} + \mathbf{m}), \end{aligned} \tag{2.23}$$

where $\mathbf{A}$ is the measurement matrix, $\mathbf{f}$ the signal to detect and $\mathbf{m}$ is AWGN. If the signal $\mathbf{f}$ is known beforehand, the solution to this problem is to use a matched filter, which maximizes the SNR while minimizing the sample rate. However, if the signal may change over the lifetime of the device that is measuring it, the matched filter cannot be implemented in hardware. In such cases it may be better to use CS, which is non-adaptive and signal independent.

In classification the task is to distinguish between L hypotheses:

$$\mathcal{H}_i : \mathbf{y} = \mathbf{Af}_i,\ i \in \{1, 2, \ldots, L\}, \tag{2.24}$$

where $\mathbf{f}_i$ is one of L possible Nyquist-rate signals. This may be used in data decoding, where the decoding corresponds to looking up in a dictionary to determine the sent symbol or information. We treat this problem in Paper A and Paper B.

Estimation is defined in [26] as the estimation of a linear function of the Nyquist-rate signal $\langle \mathbf{l}, \mathbf{f} \rangle$, where $\mathbf{l}$ is some fixed test vector. The authors show that in some cases it is possible to assume $\langle \mathbf{Al}, \mathbf{y} \rangle = \langle \mathbf{l}, \mathbf{f} \rangle$, i.e. the estimated function may be determined directly in the compressed domain. This approach is limited to only 1-dimensional linear functions. The problem of estimation in CS is also attacked in Paper C, Paper D and Paper E. However, the approach used in those works is not limited to 1-dimensional linear functions.

Finally, filtering is defined as measuring the following signal:

$$\mathbf{y} = \mathbf{A}(\mathbf{f}_D + \mathbf{f}_I), \tag{2.25}$$

where $\mathbf{f}_D$ is the desired signal and $\mathbf{f}_I$ is an interfering signal. The goal of compressive filtering is then to remove the contribution of $\mathbf{f}_I$ from $\mathbf{y}$. This is mainly achieved by assuming that $\mathbf{f}_I$ lies in a known subspace and a measurement matrix $\mathbf{A}$ may be designed which has this subspace in its null space.

2.5 Challenges in Compressive Sensing

The concept of CS is still a new field of research that began only around 2004. Many researchers have worked on implementing CS in actual signal acquisition devices, but the main body of work is still conducted as numerical simulation experiments. These numerical experiments sometimes use a signal model which fits the CS concept very well, but oversimplifies the actual signal model significantly. This section looks as three such simplification problems: quantization, noise folding and off the grid components.

Quantization is the mapping from a measurement y_i in the real or complex domain $\mathbb{R}$ or $\mathbb{C}$ to a new measurement q_i from a finite set or codebook Ω, see Fig. 2.9. The Shannon-Nyquist theorem states that when sampling above the Nyquist frequency the analog signal may be reconstructed exactly, however this is not true if the signal is first quantized. Quantization is a lossy conversion since the signal is mapped from a continuous to a finite domain. Quantization in CS was originally treated as a kind of noise, i.e. the difference between a quantized signal and the true analog signal was modelled as a stochastic variable.

Another potential problem in CS is that the signal model employed often considers *measurement noise* only. This means that the signal model is:

$$\mathbf{y} = \mathbf{Af} + \mathbf{n}, \tag{2.26}$$

where $\mathbf{n} \in \mathbb{C}^M$ is e.g. AWGN. However, in e.g. communication systems, the *signal noise* is of equal importance, sometimes even more important. In this case the signal model should be:

$$\mathbf{y} = \mathbf{A}(\mathbf{f} + \mathbf{m}) + \mathbf{n} = \mathbf{Af} + \mathbf{Am} + \mathbf{n}, \tag{2.27}$$

where $\mathbf{m} \in \mathbb{C}^N$ might also be AWGN. If the latter signal model is the most accurate then the noise is also multiplied onto the measurement matrix. This corresponds to a

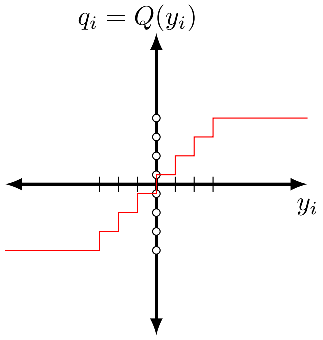


Fig. 2.9: Quantization effect.

noise folding effect which may decrease performance significantly. In [5, 90] the authors show that noise folding decreases the SNR of the signal by 3 dB per halving of the sampling rate. In [90] it is proposed to combat noise folding with quantization. The authors of that paper postulate that if a CS receiver may employ an ADC that samples at half the sample rate, this ADC may be assumed to quantize the signal better. If this assumption holds, there may be cases where the quantization noise would decrease the performance of the non-CS receiver more than noise folding decreases performance for the CS receiver. An example of such a case is given in Paper B.

A third assumption that is actually the foundation of CS is that the signal is generated according to a discrete dictionary and is sparse in that dictionary. However, as also described in Section 2.2.2, this assumption is challenged when the signal is better represented by a parametric model:

$$\mathbf{f} = \sum_{i=1}^{K} a_i \mathbf{g}(t, b_i), \tag{2.28}$$

where a_i is the amplitude coefficient, b_i is some other parameter, e.g. delay or frequency that must be estimated and $\mathbf{g}(t, b_i)$ is some parametric function. Such signals are sparse, in that there are only $2K$ unknown parameters, but they may not be sparse in a given dictionary. To solve such problems it is necessary to use knowledge of the signal's parametric model in e.g. high resolution algorithms such as MUSIC or ESPRIT or to use interpolation on the dictionary atoms. Problems such as these are treated in Paper C, Paper D and Paper E.

Chapter 3

Applications of Compressive Sensing in Communication Systems

Compressive Sensing was developed by mathematicians without any one application in mind, but rather with the general concept of subsampling as the goal. However, CS has afterwards been used in many areas such as medical imaging [29, 58], astronomy [12, 99], general imaging and video [37, 59]. For other general examples of applications of CS we refer to the references in Section 1.1 in [41] or the Compressive Sensing Resources page of Rice University `http://dsp.rice.edu/cs`. In this chapter the focus is on the application of CS in communications systems. A review is given of some of the different areas in communications where researchers have shown that CS may be applied with success. Finally, we show how the contributions of this Ph.D. thesis extend the current state-of-the-art.

The examples of CS in communications explained here are as follows:

- Sparse Channel Estimation,
- Wideband Sensing,
- Wireless Sensor Networks,
- Spread Spectrum Communications,
- Ultra-Wideband Communications,
- Radar Imaging, and

- General parameter estimation

This list is not exhaustive, but represents a broad slice of current research areas.

3.1 Sparse Channel Estimation

Channel estimation concerns the estimation of a channel impulse response. The impulse response is of interest in communications, because it may be used to equalize a received signal, so that e.g. intersymbol interference is limited. The channel estimation is often based on a transmitted pilot-sequence, which precedes the actual information transmission. Since a receiver knows the transmitted signal and the received signal, the channel impulse response is the only unknown and may be estimated. Often the channel impulse response may be assumed to be sparse [23, 74, 75]. In [23] the authors show how Matching Pursuit may be used to estimate a channel response, when the response is assumed sparse. In [8, 10] the authors show how this may be coupled with CS so that the channel impulse response may be reconstructed using less energy. Furthermore, [10] also states that an overcomplete dictionary may be better able to represent the channel impulse response sparsely. This is related to the problem of off the grid components, mentioned in Section 2.5 and to solve such problems the interpolation techniques proposed in Paper C, Paper D and Paper E may also be applicable.

3.2 Wideband Sensing

Wideband sensing is one of the main application areas for CS, since many researchers started looking into CS due to a solicitation from the US Defense Advanced Research Projects Agency (DARPA) for an ultra wideband *analog-to-information* RF converter. Such a receiver would be useful for e.g. cognitive radio. Both the analog radio front ends proposed in Section 2.2 were developed for this use [54, 65, 91]. Wideband sensing is a challenging task because in classical systems it would require either a very large filter bank, which each handles a small narrowband segment of the entire band of interest or a very fast ADC with very high dynamic range, i.e. a large number of quantization bits, to successfully acquire the signal. Currently it is not possible to produce ADCs that are fast enough for such a task. It is possible to create a filter bank for the task, but this is expensive and the required number of banks grows very large. In many scenarios the desired spectrum is actually sparse and hence researchers try to apply CS to lower the required sample rate or the required number of filter banks. A popular approach to this problem is to treat the CS problem as a Multiple Measurement Vector problem [66], which corresponds to using multiple filter banks to acquire a signal, rather than one fast ADC. Many researchers have worked on modifying and evaluating CS algorithms to work better for this type of problems [4, 48, 86]. A slightly related field is the work in [76], which exploits similar sparsity in the frequency domain to incorporate CS. This allows for less demanding filter requirements in direct down-conversion receivers.

3.3 Wireless Sensor Networks

Another area of research in CS is wireless sensor networks, where CS is used for data fusion and collection. Here, the aim is not to lower the sampling rate of a given ADC, but rather to limit the amount of sensors that must be "sampled", so that less energy is spent on transmissions between sensors. This is also called *distributed compressive sensing* [38, 39] and is developed to exploit both intra and inter sensor structure or correlation. This research area is tightly coupled with sensor fusion, encoding strategies and network coding [43]. Other notable works include [6, 78].

3.4 Spread Spectrum Communications

CS for spread spectrum systems is of interest as spread spectrum techniques are used in some wireless sensor technologies, such as the IEEE 802.15.4 standard, which is the physical layer standard for the Zigbee protocol stack. Spread spectrum techniques are also used in the Global Positioning System (GPS) to broadcast data to mobile terminals around the world. These systems do not operate at an extreme sampling rate and hence do not suffer from problems with getting a sufficiently fast ADC. Instead, it is of interest to lower the sampling rate of the ADC to reduce the monetary cost or the power requirements of the ADC component.

In the spread spectrum area, some researchers have studied the general use of CS for spread spectrum communication systems [1]. However, their work is mainly focused on using CS for fast multi-user detection, rather than subsampling. Another example is in [57], where the authors use CS to decrease the sampling rate of a GPS receiver by exploiting sparsity in the number of possible signal components at the receiver. Their receiver structure is based on possibly complicated hardware filters, which may make their implementation very difficult considering the impact of hardware filters to CS performance [71]. In [3] the authors treat a similar topic where they design spread spectrum codes to enable a base station to perform multi-user detection on a large number of users, of which only a few are active at a time. Their work focuses on simple on-off signalling, i.e. the existence of nodes, rather than communication with them, and solves the multi-user detection problem using an adapted convex optimization algorithm. Their motivation is on increasing the number of active users in a network, rather than decreasing the sampling rate of the ADC. In [100] the authors also solve a multiuser detection problem using compressive sensing, but in their work the focus is on the design of possibly complex analog filters. A comparison of different greedy algorithms for reconstruction of CS sampled Code Division Multiple Access (CDMA) signals is given in [82]. In [87] the authors investigate the sparsity trade-off point in CDMA communications between using CS or classical sampling.

Paper A is based on a signal model used in the IEEE 802.15.4 standard [89], in which a baseband signal with a Nyquist frequency of 200kHz must be sampled. In Paper B our motivation is not only to reduce the monetary cost, but also that by taking fewer samples we may be able to conserve power in the receiver, as can be seen in e.g. Eqn. 13 in [53].

3.5 Parameter Estimation

The final example of applications for CS is the very broad topic of parameter estimation in parametric models, which has uses in multiple fields. Two examples of parameter estimation problems 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 [7, 49], ultrasound [94, 97] and Ultra-Wideband (UWB) [45, 63, 72, 101]. 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 audio, speech and music processing, frequency tracking in communications [84], radar and sonar [94].

One problem in parameter estimation problems, when using a parametric model, is that the parameters for the model are often drawn from a continuous space. This causes reduced performance by the classical dictionary-based CS algorithms, as described in Section 2.5.

Some researchers have shown that this may be solved by using overcomplete frames, instead of a basis, as a dictionary. This allows for more degrees of freedom for the reconstruction algorithms. However, it also introduces ambiguity, since the coherence between atoms increases.

It was shown in [16] that as far as the recovery of signal $\mathbf{f}$, rather than the sparse vector $\mathbf{x}$, the coherence condition of the dictionary is not necessary, provided that the matrix $\mathbf{D}^H\mathbf{D}$ is sufficiently sparse.

Alternatively, one can take advantage of structured sparsity in spectral CS recovery by using a coherence inhibition model [35]. The resulting algorithm can recover the frequency-sparse signal with a DFT frame by avoiding dictionary elements with high coherence. A variation of this method uses a band-exclusion function to achieve the same avoidance [42].

More recently, it has been shown that one can recover a frequency-sparse signal from a random subset of its samples using atomic norm minimization [88].

Other work on the problem of sparsity in parametric dictionaries includes [51, 79], which uses 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 [2, 13, 50]. The simplest and most often used polynomial interpolation is fitting a parabola around the correlation peak. Such an 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 [96]. Interpolation-based algorithms improve the estimation precision but suffer 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 [40], one using a first- and second-order Taylor expansion, the other a form of polar interpolation. The authors show that polar interpolation outperforms Taylor expansion.

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 [83] or the Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) algorithm [80]. This approach exploits the fact that the dictionary matrix is cyclic. In [81], the TDE problem is converted to an FE problem and solved by means of the ESPRIT algorithm. A similar method has also been implemented using analog filters and using CS in [44].

The work in Paper C, Paper D and Paper E proposes three novel algorithms for estimating a *translation parameter* in a received signal. This translation parameter may be either time delay or frequency. The proposed algorithms are shown to outperform the current state-of-the-art algorithms in numerical experiments.

Chapter 4

Contributions

The contributions of this thesis work lie more in the field of compressive signal processing, rather than classical compressive sensing. In none of the papers supporting this thesis has the main focus been on reconstruction of the signal. Instead, the papers have always concerned some information processing from the signal. In Paper A and Paper B the main focus is data decoding, rather than reconstruction and in Paper C, Paper D and Paper E parameter estimation is the main motivation. The contributions of the individual papers are described in short below.

- **Paper A** concerns the application of compressive sensing to an IEEE 802.15.4 receiver for data decoding. The IEEE 802.15.4 PHY layer uses Direct Sequence Spread Spectrum (DSSS) codes, designed to limit the effect of interference from other wireless technologies. The redundancy introduced by this scheme may be exploited to reduce the required sample rate in the receiver. The numerical results verify that the DSSS signal may be subsampled and still accurately decoded, but also demonstrates the problem of noise folding, as introduced in Section 2.5. The main contribution of this paper, however, is the discovery that for spread spectrum systems it is possible to simplify the hardware front end. A Random Demodulator structure is used, which already is one of the simpler acquisition hardware structures, but because spread spectrum codes already spread the signal prior to transmission, the otherwise required psuedo-random noise generator may be excluded from the receiver's front end.

- **Paper B** builds upon the discovery in Paper A that in spread spectrum systems the hardware front end may be simplified. In this paper we propose the general Compressive Spread Spectrum (CSS) hardware design, which is a simplification of the Random Demodulator and can be used for all kinds of spread spectrum signals. In this paper we use Gold codes to show that the CSS measurement matrix is viable for subsampling, by generating Donoho-Tanner phase transition diagrams. Another important contribution of this paper is that the numerical experiments clearly show the effect of noise folding, but in one of the experiments it is shown that when quantization is taken into account, CS performs better than classical sampling [90].

- **Paper C** concerns parameter estimation, more specifically frequency estimation. In this paper we show that a new interpolation scheme based on polar interpolation [40] may be successfully adapted for use in CS and outperform other state-of-the-art frequency estimation algorithms, when the signals are corrupted with additive white Gaussian noise. The main novelty and contribution of the paper is a hybrid greedy and convex optimization based algorithm, which attains good performance comparable to the convex optimization algorithm from [40], but significantly faster.

- **Paper D** applies the same polar interpolation as in Paper C but for the TDE problem. The main novelty of this work is the proposal for a purely greedy algorithm based on polar interpolation, which is shown to outperform current state-of-the-art TDE algorithms.

- **Paper E** summarizes and extends the work from Paper C and Paper D. This paper generalizes the parameter estimation framework to hold for all translation-invariant signal classes and proposes a simple way of testing whether a given class is eligible for use in the proposed framework. Furthermore, the convex optimization algorithm from [40] is improved upon, such that more general classes of signals are allowed and also the hybrid algorithm from Paper C is replaced by a better version, which handles overlapping signals better. Another contribution of this work is a numerical experiment which shows that even though the reconstruction of a signal suffers from noise folding, the estimation of its parameters is not as heavily affected.

Chapter 5

Conclusion

This thesis has proposed a new hardware structure for spread spectrum signals, highlighted the problem of noise folding, but also shown that when quantization is taken into account, this problem may be less significant. Furthermore, several algorithms have been proposed for parameter estimation for the class of translation-invariant signals, which outperform the current state-of-the-art algorithms for frequency and time delay estimation when the signal of interest is generated with a parametric signal model, rather than a dictionary-based one. The approach in the thesis has been to attack the problems of CS as listed in Section 2.5 for various signal classes relevant in communication systems. However, the developed methods and algorithms are also relevant outside the area of communications.

In the CS research area most of the focus still remains on the dictionary-based, reconstruction oriented approach, where only measurement noise is assumed. However, as the field matures, the field of compressive signal processing will probably also gain more momentum and as more researchers use CS in laboratory experiments, the effect of noise folding and quantization will have to be addressed, as well as signals that follow parametric models, rather than a dictionary-based one.

While the work in this thesis has relied mainly on numerical experiments and synthesized signals, the natural next step is to evaluate the proposed methods on real systems and signals. This would verify whether the noise folding effect is modelled correctly and also highlight how robust the proposed algorithms are towards model inaccuracies. Furthermore, the polar interpolation approach should be further explored. The current convex optimization formulation, denoted CCBP in Paper E is very cumbersome and we believe it is possible to come up with a simpler, faster convex optimization problem formulation. Furthermore, the field of estimation theory within compressive sensing is still fairly unexplored. A natural next step would also be to compare the proposed estimation algorithms in this work with the Cramer Rao lower bound.

References

[1] V. Aggarwal, L. Applebaum, A. Bennatan, A. Calderbank, S. Howard, and S. Searle, “Enhanced CDMA communications using compressed-sensing reconstruction methods,” in *47th Annual Allerton Conf. on Communication, Control, and Computing*, Sep. 2009, pp. 1211–1215.

[2] 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.

[3] L. Applebaum, W. U. Bajwa, M. F. Duarte, and R. Calderbank, “Asynchronous code-division random access using convex optimization,” *Physical Communication*, vol. 5, no. 2, pp. 129–147, 2012.

[4] D. Ariananda and G. Leus, “Compressive wideband power spectrum estimation,” *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4775–4789, 2012.

[5] 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.

[6] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, “Compressive wireless sensing,” in *The Fifth International Conference on Information Processing in Sensor Networks (IPSN)*, 2006, pp. 134–142.

[7] W. Bajwa, K. Gedalyahu, and Y. Eldar, “Identification of parametric underspread linear systems and super-resolution radar,” *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2548–2561, 2011.

[8] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, “Compressed channel sensing: A new approach to estimating sparse multipath channels,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1058–1076, 2010.

[9] R. G. Baraniuk and M. B. Wakin, “Random projections of smooth manifolds,” *Found. Comput. Math.*, vol. 9, no. 1, pp. 51–77, Jan. 2009.

[10] C. Berger, Z. Wang, J. Huang, and S. Zhou, “Application of compressive sensing to sparse channel estimation,” *IEEE Communications Magazine*, vol. 48, no. 11, pp. 164–174, 2010.

[11] J. Blanchard, C. Cartis, and J. Tanner, “Compressed sensing: How sharp is the restricted isometry property?” *SIAM Review*, vol. 53, no. 1, pp. 105–125, 2011.

[12] J. Bobin, J.-L. Starck, and R. Ottensamer, “Compressed sensing in astronomy,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 5, pp. 718–726, 2008.

[13] 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.

[14] P. T. Boufounos, “Greedy sparse signal reconstruction from sign measurements,” in *Proceedings of the Asilomar Conference on Signals, Systems and Computing*, Nov. 2009, pp. 1305–1309.

[15] P. T. Boufounos and R. G. Baraniuk, “1-bit compressive sensing,” in *Proceedings of the Conference on Information Sciences and Systems (CISS)*, Mar. 2008.

[16] E. J. Candès, Y. C. Eldar, D. Needell, and P. Randall, “Compressed sensing with coherent and redundant dictionaries,” *Applied and Computational Harmonic Analysis*, vol. 31, no. 1, pp. 59 – 73, 2011.

[17] E. J. Candes, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[18] E. J. Candès and T. Tao, “Decoding by linear programming,” *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 4203–4215, 2005.

[19] E. J. Candès, “The restricted isometry property and its implications for compressed sensing,” *Compte Rendus de l’Academie des Sciences, Serie I*, vol. 346, no. 9-10, pp. 589–592, 2008.

[20] S. Chen, D. Donoho, and M. Saunders, “Atomic decomposition by basis pursuit,” *SIAM J. Sci. Comput.*, vol. 20, pp. 33–61, 1998.

[21] Y. Chi, L. Scharf, A. Pezeshki, and A. Calderbank, “Sensitivity to basis mismatch in compressed sensing,” *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2182–2195, 2011.

[22] A. Cohen, W. Dahmen, and R. DeVore, “Compressed sensing and best k-term approximation,” *Journal of the American Mathematical Society*, vol. 22, no. 1, pp. 211–231, 2009.

[23] S. Cotter and B. Rao, “Sparse channel estimation via matching pursuit with application to equalization,” *IEEE Transactions on Communications*, vol. 50, no. 3, pp. 374–377, 2002.

[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] M. A. Davenport, “Random observations on random observations: Sparse signal acquisition and processing,” Ph.D. dissertation, Rice University, 2010.

[26] M. Davenport, P. Boufounos, M. Wakin, and R. Baraniuk, “Signal processing with compressive measurements,” *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 445–460, Apr. 2010.

[27] M. Davenport and M. Wakin, “Analysis of orthogonal matching pursuit using the restricted isometry property,” *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4395–4401, 2010.

[28] G. M. Davis, S. G. Mallat, and Z. Zhang, “Adaptive time-frequency decompositions with matching pursuit,” pp. 402–413, 1994.

[29] A. Dixon, E. Allstot, D. Gangopadhyay, and D. Allstot, “Compressed sensing system considerations for ecg and emg wireless biosensors,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 2, pp. 156–166, 2012.

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

[31] 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.

[32] D. L. Donoho and J. Tanner, “Precise undersampling theorems,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 913–924, Jun. 2010.

[33] D. Donoho and J. Tanner, “Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 367, no. 1906, pp. 4273–4293, 2009.

[34] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

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

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

[37] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk, “Single-pixel imaging via compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, 2008.

[38] M. Duarte, S. Sarvotham, D. Baron, M. Wakin, and R. Baraniuk, “Distributed compressed sensing of jointly sparse signals,” in *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, 2005, pp. 1537–1541.

[39] M. Duarte, M. Wakin, D. Baron, and R. Baraniuk, “Universal distributed sensing via random projections,” in *The Fifth International Conference on Information Processing in Sensor Networks (ISPN)*, 2006, pp. 177–185.

[40] 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.

[41] Y. C. Eldar and G. Kutyniok, *Compressed sensing: Theory and Applications*, 2nd ed. Cambridge University Press, 2012.

[42] 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.

[43] S. Feizi and M. Medard, “A power efficient sensing/communication scheme: Joint source-channel-network coding by using compressive sensing,” in *49th Annual Allerton Conf. on Communication, Control, and Computing*, Sep. 2011, pp. 1048–1054.

[44] 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.

[45] S. Gishkori, G. Leus, and V. Lottici, “Compressive sampling based differential detection for UWB impulse radio signals,” *Physical Communication*, vol. 5, no. 2, pp. 185 – 195, 2012.

[46] M. Grant and S. Boyd, “Graph implementations for nonsmooth convex programs,” in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/~boyd/graph_dcp.html.

[47] ——, “CVX: Matlab software for disciplined convex programming, version 2.0 beta,” http://cvxr.com/cvx, Sep. 2012.

[48] R. Grigoryan, T. Arildsen, D. Tandur, and T. Larsen, *Performance Comparison of Reconstruction Algorithms in Discrete Blind Multi-Coset Sampling*. IEEE Press, 2012.

[49] M. Herman and T. Strohmer, “High-resolution radar via compressed sensing,” *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2275–2284, 2009.

[50] 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.

[51] 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.

[52] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, “Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors,” *IEEE Transactions on Information Theory*, vol. 59, no. 4, Apr. 2013.

[53] P. Kenington and L. Astier, “Power consumption of A/D converters for software radio applications,” *IEEE Transactions on Vehicular Technology*, vol. 49, no. 2, pp. 643–650, Mar. 2000.

[54] S. Kirolos, J. Laska, M. Wakin, M. Duarte, D. Baron, T. Ragheb, Y. Massoud, and R. Baraniuk, “Analog-to-information conversion via random demodulation,” in *IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, Oct. 2006, pp. 71–74.

[55] B. Le, T. Rondeau, J. Reed, and C. Bostian, “Analog-to-digital converters,” *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 69–77, 2005.

[56] M. A. Lexa, M. Davies, and J. Thompson, “Reconciling compressive sampling systems for spectrally sparse continuous-time signals,” *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 155–171, 2012.

[57] X. Li, A. Rueetschi, Y. C. Eldar, and A. Scaglione, “GPS Signal Acquisition via Compressive Multichannel Sampling,” *Physical Communication*, vol. 5, no. 2, pp. 173–184, 2012.

[58] M. Lustig, D. Donoho, and J. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging.” *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–95, 2007.

[59] J. Ma, G. Plonka, and M. Hussaini, “Compressive video sampling with approximate message passing decoding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 9, pp. 1354–1364, 2012.

[60] A. Maleki and D. L. Donoho, “Optimally tuned iterative reconstruction algorithms for compressed sensing,” *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 330–341, Apr. 2010.

[61] S. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[62] E. Meijering, “A chronology of interpolation: from ancient astronomy to modern signal and image processing,” *Proceedings of the IEEE*, vol. 90, no. 3, pp. 319–342, 2002.

[63] J. Meng, J. Ahmadi-Shokouh, H. Li, E. Charlson, Z. Han, S. Noghanian, and E. Hossain, “Sampling rate reduction for 60 ghz uwb communication using compressive sensing,” in *Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, 2009, pp. 1125–1129.

[64] M. Mishali and Y. C. Eldar, “From theory to practice: Sub-nyquist sampling of sparse wideband analog signals,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 375–391, Apr. 2010.

[65] ——, “Xampling: Signal acquisition and processing in a union of subspaces,” *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4719–4734, 2011.

[66] M. Mishali and Y. Eldar, “Blind multiband signal reconstruction: Compressed sensing for analog signals,” *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 993–1009, 2009.

[67] S. Muthukrishnan, “Data streams: Algorithms and applications,” *Foundations and Trends in Theoretical Computer Science*, vol. 1, no. 2, pp. 117–236, 2005.

[68] D. Needell and J. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,” *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301 – 321, 2009.

[69] 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.

[70] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time signal processing (2nd ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999.

[71] P. Pankiewicz, T. Arildsen, and T. Larsen, “Sensitivity of the Random Demodulation Framework to Filter Tolerances,” in *19th European Signal Processing Conf. (EUSIPCO)*, Barcelona, Spain, Aug. 2011.

[72] J. Paredes, G. Arce, and Z. Wang, “Ultra-wideband compressed sensing: Channel estimation,” *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 3, pp. 383–395, Oct. 2007.

[73] Y. Pati, R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition,” in *Conference Record of the 27th Asilomar Conference on Signals, Systems and Computers*, 1993.

[74] N. Pedersen, C. Manchón, and B. Fleury, “A fast iterative bayesian inference algorithm for sparse channel estimation,” *IEEE International Conference on Communications*, 2013.

[75] N. Pedersen, C. Manchón, D. Shutin, and B. Fleury, *Application of Bayesian Hierarchical Prior Modeling to Sparse Channel Estimation*, ser. IEEE International Conference on Communications, 2012, pp. 3487–3492.

[76] J. Pierzchlewski, T. Arildsen, and T. Larsen, *Compressed Sensing-Based Direct Conversion Receiver.* IEEE Press, 2012, pp. 804 – 809.

[77] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. McGraw Hill, 2008.

[78] G. Quer, R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi, “Sensing, compression, and recovery for WSNs: Sparse signal modeling and monitoring framework,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 10, pp. 3447–3461, 2012.

[79] D. Ramasamy, S. Venkateswaran, and U. Madhow, “Compressive Parameter Estimation in AWGN,” *Preprint (available at: http://arxiv.org/abs/1304.7539)*, 2013.

[80] 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.

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

[82] H. Schepker and A. Dekorsy, “Sparse multi-user detection for cdma transmission using greedy algorithms,” in *8th International Symposium on Wireless Communication Systems (ISWCS)*, 2011, pp. 291–295.

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

[84] S. Schnelle, J. P. Slavinsky, P. Boufounos, M. Davenport, and R. Baraniuk, “A compressive phase-locked loop,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 2885–2888.

[85] C. Shannon, “Communication in the presence of noise,” *Proceedings of the IEEE*, vol. 86, no. 2, pp. 447–457, 1998.

[86] H. Shen, T. Arildsen, and T. Larsen, “Blind multi-band spectrum signals reconstruction algorithms comparison,” *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pp. 353–357, 2011.

[87] B. Shim and B. Song, “Multiuser detection via compressive sensing,” *IEEE Communications Letters*, vol. 16, no. 7, pp. 972–974, 2012.

[88] G. Tang, B. Bhaskar, P. Shah, and B. Recht, “Compressed sensing off the grid,” *Preprint (available at: http://arxiv.org/abs/1207.6053)*, 2012. [Online]. Available: http://arxiv.org/abs/1207.6053

[89] *IEEE Std. 802.15.4-2006*, The Institute of Electrical and Electronics Engineers, Inc., 2006, available online: http://standards.ieee.com.

[90] J. Treichler, M. A. Davenport, J. N. Laska, and R. G. Baranuik, "Dynamic range and compressive sensing acquisition receivers," in *Defense Applications of Signal Processing (DASP)*, Coolum, Australia, Jul. 2011.

[91] 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.

[92] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[93] J. Tropp and S. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.

[94] R. Tur, Y. Eldar, and Z. Friedman, "Innovation rate sampling of pulse streams with application to ultrasound imaging," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1827–1842, 2011.

[95] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1417–1428, 2002.

[96] 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.

[97] N. Wagner, Y. Eldar, and Z. Friedman, "Compressed beamforming in ultrasound imaging," *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4643–4657, 2012.

[98] R. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 539–550, 1999.

[99] Y. Wiaux, L. Jacques, G. Puy, A. M. M. Scaife, and P. Vandergheynst, "Compressed sensing imaging techniques for radio interferometry," *Monthly Notices of the Royal Astronomical Society*, vol. 395, no. 3, pp. 1733–1742, 2009.

[100] X. Xie, Y. C. Eldar, and A. Goldsmith, "Reduced-dimension multiuser detection," 2011, submitted for IEEE Transactions on Information Theory, available on arXiv: http://arxiv.org/abs/1109.6303.

[101] P. Zhang, Z. Hu, R. Qiu, and B. Sadler, "A compressed sensing based ultra-wideband communication system," in *Proc. 2009 IEEE International Conf. on Communications*, Piscataway, NJ, USA, 2009, pp. 4239–4243.

Part II

Papers

Paper A

Demodulating Subsampled Direct Sequence Spread Spectrum Signals using Compressive Signal Processing

Karsten Fyhn, Thomas Arildsen, Torben Larsen, Søren Holdt Jensen

The paper has been published in the
Proceedings of the 20th European Signal Processing Conference (EUSIPCO)
Bucharest, Romania, 2012.

The layout has been revised.

Abstract

We show that to lower the sampling rate in a spread spectrum communication system using Direct Sequence Spread Spectrum (DSSS), compressive signal processing can be applied to demodulate the received signal. This may lead to a decrease in the power consumption or the manufacturing price of wireless receivers using spread spectrum technology. The main novelty of this paper is the discovery that in spread spectrum systems it is possible to apply compressive sensing with a much simpler hardware architecture than in other systems, making the implementation both simpler and more energy efficient. Our theoretical work is exemplified with a numerical experiment using the IEEE 802.15.4 standard's 2.4 *GHz band specification. The numerical results support our theoretical findings and indicate that compressive sensing may be used successfully in spread spectrum communication systems. The results obtained here may also be applicable in other spread spectrum technologies, such as Code Division Multiple Access (CDMA) systems.*

A.1 Introduction

The concept of compressive sensing [1, 2] is attracting more and more attention in the signal processing community. Where the classical Shannon-Nyquist sampling theorem requires a signal to be sampled at twice its signal bandwidth, compressive sensing samples the signal at its information rate, which may be much lower. Compressive sensing is used to reconstruct a signal to a full Nyquist rate representation, but if only inference about information in the signal is desired, compressive signal processing is better suited [3]. Compressive signal processing is used when inference about information in a signal is of interest, rather than the reconstruction of the signal itself. Compressive sensing and compressive signal processing samples the signal using a sampling scheme with typically a randomized structure and then exploits sparsity in the signal to enable subsampling. In DSSS systems the sparsity is in the selection of a code used for transmission of a given data sequence. In this work we show how compressive signal processing may be applied to a spread spectrum receiver to lower the sampling rate at the receiver. This may lower the overall energy consumption of the device and/or lower the price of the Analog to Digital Converter (ADC). To exemplify this consider the following: This work is based on a signal model used in the IEEE 802.15.4 standard [4], in which a baseband signal with a Nyquist frequency of 200kHz must be sampled. To show the benefit of lowering the sampling rate, we compare two ADCs from Analog Devices[1]: The AD7819 and the AD7813. The AD7819 is an 8-bit ADC with a maximum throughput of 200 kilosamples per second, whereas the AD7813 is an 8- or 10-bit ADC with a maximum throughput of 400 kilosamples per second. We are aware that 400 kilosamples

[1] `http://www.analog.com`

per second is the Nyquist rate of the system and the sampling rate should be higher than this to comply with the Shannon-Nyquist sampling theorem. However, we use these two ADCs as they are almost identical in every aspect except for the sampling rate, making them perfect for comparison. In present IEEE 802.15.4 compliant receivers, an ADC similar to the AD7813 must be used to comply with Shannon-Nyquist, but if compressive signal processing is able to lower the sampling rate by a factor of two, the AD7819 may be used instead. These two particular ADCs use the same amount of power so there are no energy savings, but where the AD7813 costs 2.98$, the AD7819 only costs 2.29$.

Previous work has studied the use of compressive sensing in Ultra-Wideband (UWB) systems for channel estimation where the sparsity of the signal lies in the time domain [5, 6]. Some researchers have studied the use of compressive sensing for spread spectrum communication systems [7]. However, this work is mainly theoretical and relies on second order Reed-Muller codes, which would be difficult to implement in hardware. A more practical approach is given in [8] where compressive sensing is used to decrease the sampling rate of a GPS receiver by exploiting sparsity in the number of possible signal components at the receiver. However, this approach also suffers from a complicated hardware implementation. In both works the receiver must use complicated hardware filters, which may make their implementation very difficult, considering the impact of hardware filters on compressive sensing performance [9]. In this work we apply compressive signal processing to a general DSSS system. We show that in a spread spectrum system it is possible to use simply a repeated version of the matched filter used in classic receivers instead of using a complicated filter structure to acquire random measurements. This greatly simplifies the implementation and makes compressive sensing feasible for implementation in spread spectrum wireless receiver systems. Our approach is not limited to DSSS but may also be applied in other spread spectrum technologies, such as CDMA.

One major obstacle in applying compressive sensing to any wireless system is the presence of noise folding, which occurs because the noise is not measurement noise, but noise added before measuring the signal. This severely impacts the receiver performance, which is also evident in our numerical experiments. We discuss how to mitigate this in Section A.6.

In the following we first define classic transmitter and receiver structures in Sections A.2 and A.3, respectively. Then we show how the classic receiver structure must be modified to incorporate compressive signal processing in Section A.4. Our theoretical work is exemplified with a numerical experiment using the IEEE 802.15.4 standard in Section A.5 followed by a discussion and conclusion in Section A.6.

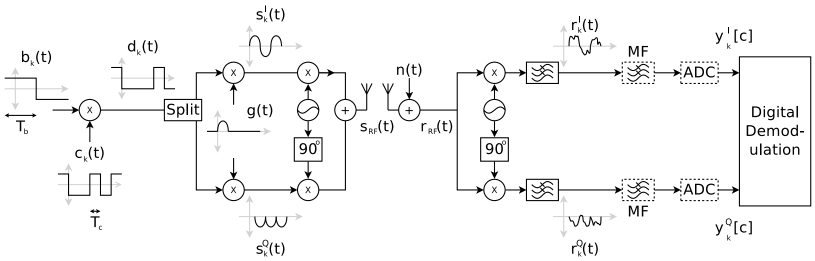


Fig. A.1: Transmitter and receiver structure for QPSK modulation/demodulation. The items drawn using dotted lines are hardware components that must be modified to enable compressive sensing.

A.2 Transmitter Structure

In both the transmitter and the receiver structure we treat the signal symbol-by-symbol, where each symbol may be a single bit of information or a block of bits. Let $\mathbf{b}_k \in \{\pm 1\}^{\mathrm{N}\times 1}$ be a binary vector, signifying the kth symbol to be transmitted and consisting of N information bits. Now define a binary pseudo-random noise (PRN) sequence as $\mathbf{c}_k \in \{\pm 1\}^{\mathrm{C}\times 1}$. These two binary vectors are the discrete equivalents of an information signal and a PRN signal, $b_k(t)$ and $c_k(t)$, respectively as shown in Fig. A.1 and are defined as:

$$b_k(t) = \sum_{n=0}^{N-1} \mathbf{b}_k[n]\mathrm{rect}\left(\frac{t-nT_b}{T_b}\right), \ \ 0 \le t < NT_b, \tag{A.1}$$

$$c_k(t) = \sum_{c=0}^{C-1} \mathbf{c}_k[c]\mathrm{rect}\left(\frac{t-cT_c}{T_c}\right), \ \ 0 \le t < CT_c, \tag{A.2}$$

where T_b and T_c are the bit and chip duration, respectively, and $NT_b = CT_c$. We define:

$$\mathrm{rect}(t) = \begin{cases} 1 & \text{if } 0 \le t < 1, \\ 0 & \text{otherwise.} \end{cases} \tag{A.3}$$

When multiplied, they form the spread spectrum data signal, $d_k(t) = b_k(t)c_k(t), \ 0 \le t < NT_b$.

The notation used in the above may in some cases be simplified, as the choice of a PRN sequence might be implemented as a mapping from one bit or a block of bits directly to a given sequence of chips, as done in e.g. IEEE 802.15.4 [4]. In the following, the signal model we define is based on the IEEE 802.15.4 standard's 2.4 GHz band specification. This means the encoding using DSSS may be written as a matrix-vector product, with $M = 2^N$ possible data signals:

$$d_k(t) = \boldsymbol{\Psi}(t)\boldsymbol{\alpha}_k, \text{ where} \tag{A.4}$$

$$\boldsymbol{\Psi}(t) = \begin{bmatrix} d_1(t) \\ d_2(t) \\ \vdots \\ d_M(t) \end{bmatrix}^{\mathrm{T}}, \ 0 \leq t < NT_b, \tag{A.5}$$

where $\boldsymbol{\Psi}(t)$ is a dictionary of possible data signals and $\boldsymbol{\alpha}_k \in \{0,1\}^{\mathrm{M}\times 1}$ is a sparse vector with only one non-zero entry, namely the entry that selects a given PRN sequence from the dictionary. It may also be considered a symbol vector as it corresponds to the kth symbol being transmitted. The sparsity of $\boldsymbol{\alpha}_k$ is what enables us to use compressive sensing for demodulation. The sparsity of the signal lies in which PRN sequence is chosen for transmission.

The IEEE 802.15.4 2.4 GHz band specification is based on QPSK and therefore the output sequence is split up, so that even-indexed chips in $d_k(t)$ are transmitted in the in-phase path and odd-indexed chips in the quadrature-phase path. In the following we only state the equations for the in-phase path, but similar expressions may be derived for the quadrature-phase part. The resulting data signals are then used to modulate some pulse shape function, $g(t)$:

$$s_k^I(t) = \boldsymbol{\Psi}^I(t)\boldsymbol{\alpha}_k, \text{ where} \tag{A.6}$$

$$\boldsymbol{\Psi}^I(t) = \begin{bmatrix} \sum\limits_{c\in S} d_1(t) g\left(t - cT_c\right) \\ \sum\limits_{c\in S} d_2(t) g\left(t - cT_c\right) \\ \vdots \\ \sum\limits_{c\in S} d_M(t) g\left(t - cT_c\right) \end{bmatrix}^{\mathrm{T}}, \ S = \{0, 2, \ldots, C\} \tag{A.7}$$

Here the dictionary matrix has been recast into an in-phase version, with pulse shape function included. Notice that $g(t)$ here and as depicted in Fig. A.1 is assumed to be a half-sine pulse, which is the pulse shaping function used in IEEE 802.15.4. This pulse shape has limited support in the time domain, which is not true for e.g. a raised cosine pulse shape. The equations in this work are defined for the half-sine pulse shape, but are easily changed to apply to other pulse shape functions.

A.3 Classic Receiver Structure

Before introducing our compressive sensing receiver structure, we first define a classic Nyquist sampling receiver structure. At the receiver, the received signal is:

$$r_k(t) = s_k(t) + n(t), \tag{A.8}$$

where $n(t)$ is additive white Gaussian noise.

The in-phase and quadrature-phase analog signals are sampled according to the chip rate using a matched filter to the pulse shape used at the transmitter and an ADC. Here, we assume a coherent receiver with perfect synchronization, performed prior to data decoding using e.g. a pilot sequence. The sampling may be represented using a measurement matrix, $\mathbf{\Theta}_1(t)$:

$$y_k^I[\ell] = \int_{\ell T_c}^{(\ell+1)T_c} \theta_\ell(t) r_k^I(t) \mathrm{d}t, \text{ where} \tag{A.9}$$

$$\mathbf{\Theta}_1(t) = \begin{bmatrix} \theta_0(t) \\ \theta_1(t) \\ \vdots \\ \theta_{C-1}(t) \end{bmatrix}, \quad \begin{matrix} \theta_i(t) = g(t - iT_c), \\ 0 \le t < CT_c \end{matrix} \tag{A.10}$$

The measurement matrix is denoted $\mathbf{\Theta}_1$ because it samples every $T_c/1$, i.e. at Nyquist rate.

This means that for every received symbol $2C$ samples must be taken for the in-phase and quadrature-phase signals in total. These samples then form the received signal vectors, $\mathbf{y}_k^I$ and $\mathbf{y}_k^Q$, which are used to demodulate the signal and find an estimate of the transmitted symbol, represented as $\boldsymbol{\alpha}_k$, using a least squares estimator.

Due to the simple design of this signalling scheme and the matched filter, it is possible to perform the demodulation process as a least squares estimation with simple purely binary versions of the analog dictionary and measurement matrices, $\mathbf{\Psi}^I(t), \mathbf{\Psi}^Q(t)$ and $\mathbf{\Theta}_1(t)$, respectively.

Define $\mathbf{y}_k = \mathbf{y}_k^I + j\mathbf{y}_k^Q$ and define M *signal candidates* as $\mathbf{s}_m = \mathbf{\Theta}_1 \left(\mathbf{\Psi}^I \boldsymbol{\alpha}_m + j\mathbf{\Psi}^Q \boldsymbol{\alpha}_m\right)$, where $\mathbf{\Theta}_1 = I$ is now simply the $\mathrm{C} \times \mathrm{C}$ identity matrix and $\mathbf{\Psi}^I \in \{\pm 1\}^{\mathrm{C}\times\mathrm{M}}$ and $\mathbf{\Psi}^Q \in \{\pm 1\}^{\mathrm{C}\times\mathrm{M}}$ are the discrete in-phase and quadrature-phase dictionary matrices with each entry signifying either a positive (1) or negative (-1) pulse in the analog versions of the dictionary matrices. With these definitions in order the least squares estimate can be found as:

$$\tilde{\boldsymbol{\alpha}}_{k,\mathrm{idx}} = \underset{m}{\operatorname{argmin}} \Big(\mathbf{y}_k - \mathbf{s}_m\Big)^{\mathrm{H}} \Big(\mathbf{y}_k - \mathbf{s}_m\Big) \tag{A.11}$$

where $(\cdot)^{\mathrm{H}}$ denotes Hermitian transpose, $\tilde{\alpha}_{k,\mathrm{idx}}$ is the estimate of the index in the $\boldsymbol{\alpha}_k$ vector which is non-zero, i.e. the index corresponding to the symbol that has been transmitted.

A.4 Compressive Sensing Receiver Structure

In hardware compressive sensing sampling structures, such as the Random Demodulator [10], a PRN sequence is mixed with the received signal followed by low-pass filtering. Due to the presence of a PRN sequence in a spread spectrum transmitter, which spreads the data signal, a compressive sensing-enabled receiver may merely use a repeated version of its matched filter, subsample the received signal and still demodulate the information. Before sampling, the matched filter must be modified to contain not only a single chip pulse shape but as many chip pulse shapes as shall be contained per sample. This received signal vector may then be written as:

$$\mathbf{y}_k^I[\ell] = \int_{\ell T_c/\kappa}^{(\ell+1)T_c/\kappa} \theta_\ell(t) r_k^I(t)\mathrm{d}t, \text{ where} \tag{A.12}$$

$$\mathbf{\Theta}_{1/\kappa}(t) = \begin{bmatrix} \theta_0(t) \\ \theta_1(t) \\ \vdots \\ \theta_{L-1}(t) \end{bmatrix}, \quad \begin{array}{l} \theta_j(t) = \sum\limits_{c=j/\kappa}^{(j+1)/\kappa} g(t - cT_c), \\ 0 \leq t \leq CT_c \end{array} \tag{A.13}$$

Here each value of $\ell = 0, 1, \ldots, L$ signifies a collection of chips due to the subsampling where $L = C\kappa$ is the number of samples taken per symbol. $\kappa = \frac{L}{C} \in]0, 1]$ is the undersampling ratio in the compressive sensing system and signifies the ratio between taken samples and Nyquist samples. In this work we limit ourselves to scenarios where $1/\kappa$ is an integer number, i.e. only an integer number of Nyquist samples are compressed together into one sample.

To verify that the use of an additional PRN sequence at the receiver is unnecessary, we may look at the outcome of the subsampling ADC in Fig. A.1. Assuming a noise-free setting ($n(t) = 0$), the outcome becomes:

$$\begin{aligned} y_k^I[\ell] &= \sum_{c=\ell/\kappa}^{(\ell+1)/\kappa} \int_{cT_c}^{(c+1)T_c} r_k^I(t) p_{\mathrm{PRN}}(t)\mathrm{d}t \\ &= \sum_{c=\ell/\kappa}^{(\ell+1)/\kappa} \int_{cT_c}^{(c+1)T_c} \sum_{c'=0}^{C/2-1} b_k(t + c'T_c) c_k(t + c'T_c) \\ &\quad \cdot g(t - nT_c) p_{\mathrm{PRN}}(t)\mathrm{d}t \end{aligned} \tag{A.14}$$

Notice that the up and down-conversions have been assumed perfect and $p_{\mathrm{PRN}}(t)$ is a new PRN sequence, added at the receiver as is done in the Random Demodulator receiver structure [10]. The symbol c' denotes a chip picked out in $d_k(t)$ at the transmitter and used to avoid confusion with c, the chips added together into a sample at the receiver. The special indexing with T_c in connection with $b_k(t)$ and $c_k(t)$ is to pick out the chips in the in-phase path only, similar to what was done in (A.7). Because everything is multiplicative, it can be seen that $c_k(t + nT_c)$ and $p_{\text{PRN}}(t)$ are synchronized and have the same chip rate, i.e. they may be viewed as a single PRN sequence. It follows that the multiplication of a PRN sequence at the receiver is unnecessary here.

Because we wish to demodulate a signal, which is equivalent to a classification problem, it is not necessary for us to reconstruct the full original signal as is done in compressive sensing. Instead we use the recently introduced concept of compressive signal processing [3] to perform classification in the compressed domain. By classification, we mean to classify which of the signal candidates in the dictionary $\mathbf{\Psi}^I$ and $\mathbf{\Psi}^Q$ has been transmitted. This does not require reconstruction of the signal itself and may therefore be done with less computational complexity by using compressive signal processing, rather than classic compressive sensing algorithms, that reconstruct the full signal.

To demodulate the data at the receiver using the two subsampled chip sequences, $\mathbf{y}_k^I$ and $\mathbf{y}_k^Q$, the classification rule in (A.11) is used again with $\mathbf{\Theta}_{1/\kappa} \in \{0,1\}^{\text{L}\times\text{C}}$ instead of $\mathbf{\Theta}_1 \in \{0,1\}^{\text{C}\times\text{C}}$. In [3] a prewhitening matrix, $\mathbf{W}$, is introduced to counter noise coloring by the measurement matrix. However, as our proposed measurement matrix, $\mathbf{\Theta}_{1/\kappa}$, has no overlapping rows, the noise remains white in our case. This prewhitening matrix is therefore not necessary here, but if e.g. a Gaussian or Bernoulli measurement matrix is used instead, it must be included.

A.5 Numerical Results

To demonstrate the performance of our proposed receiver structure, we have performed a numerical experiment in which we compare the Bit Error Rate (BER) of a classical receiver to that of a compressive sensing-enabled receiver. This is done for a range of Signal-to-Noise-Ratio (SNR) levels. The system used for this experiment is our MATLAB implementation of the physical layer of the IEEE 802.15.4 2450 MHz OQPSK radio band specification [4]. Each block of four bits is mapped into one of 16 binary chip sequences[2], according to the mapping in [4]. The chip sequence is then modulated using Offset Quadrature Phase Shift Keying (OQPSK). This standard has been chosen due to its widespread use, having been deployed already in many applications around the world and because it is a known standard to many scientists and engineers.

The experiment is repeated for a range of SNRs or more specifically energy per bit per noise spectral density (Eb/N0). The noise is added in a bandwidth corresponding to that of the baseband signal, i.e. 2MHz [4]. Our experiment is conducted by transmitting randomly generated data packets of length $127 \times 8 = 1016$ bits each (the maximum size of an IEEE 802.15.4 data packet). For each of the two tested methods and for each Eb/N0 level, bits are transmitted until at least 1000 bits have been received in error. All MATLAB code developed for this paper is published following the principle

[2] In the published version of this paper we write 32. The correct number is 16.

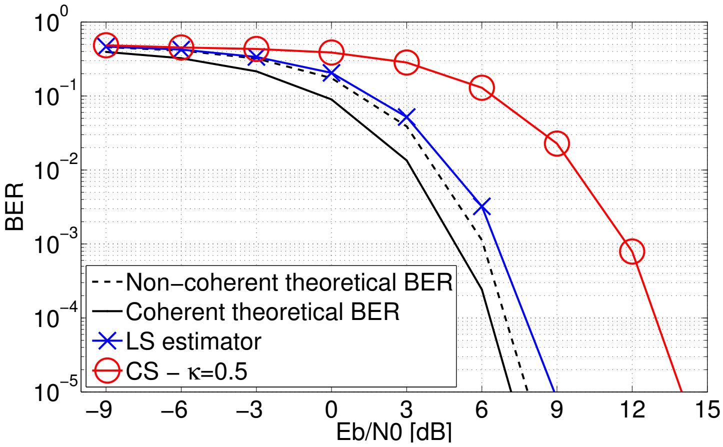


Fig. A.2: The BER versus Eb/N0 for a classical receiver implementation using least squares compared to that of a compressive sensing enabled receiver with $\kappa = 0.5$. The full black curve signifies theoretical BER per Eb/N0 for coherent MFSK and the dashed curve is theoretical BER per Eb/N0 for non-coherent MFSK.

of Reproducible Research [11] and is freely available at `http://www.sparsesampling.com/cspdsss2012`.

To validate the implementation of the compressive sensing framework, we have conducted a numerical experiment in which we added a constant to the transmitted signal, rather than additive white Gaussian noise (AWGN). The results for both the classical least squares and the compressive sensing implementation follow the expected results as found through mathematical calculations, thereby indicating that the implementation performs as expected.

The result of the BER versus Eb/N0 experiment with AWGN is shown in Fig. A.2. Also shown is the theoretical BER versus Eb/N0 for coherent MFSK [12], numerically evaluated:

$$P_b = \frac{8}{15}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\left[1-(1-Q(x))^{15}\right]e^{-\frac{\left(x-\sqrt{8\frac{E_b}{N_0}}\right)^2}{2}}\mathrm{d}x. \tag{A.15}$$

We have also included the theoretical curve for non-coherent MFSK, as it is stated in the IEEE 802.15.4 standard [4]:

$$P_b = \frac{8}{15}\frac{1}{16}\sum_{m=2}^{16}(-1)^m\begin{pmatrix}16\\m\end{pmatrix}e^{4\frac{E_b}{N_0}\left(\frac{1}{m}-1\right)} \tag{A.16}$$

The classical implementation does not follow the theoretical bound exactly because the PRN sequences in [4] are not orthogonal and due to the short code lengths.

For $\kappa = 0.5$ the compressive sensing receiver performs worse than a classical receiver by $\approx$ 4-5 dB, which is supported by the results on noise folding in [13].

A.6 Discussion and Conclusion

We have shown that compressive sensing enables subsampling of a DSSS signal. This has been demonstrated by means of IEEE 802.15.4 2.4GHz OQPSK signals, which we successfully subsampled with half the Nyquist rate. This subsampling may lead to a decrease in energy consumption or a lowering of the manufacturing price. The penalty is the expected drop in performance due to noise folding. This penalty has not been further treated in this work but in [13] the authors suggest to incorporate the effect of quantization, which should favor compressive sensing over a classical receiver as a compressive sensing enabled receiver is able to quantize the signal at a higher resolution, due to the lower sample rate.

An undersampling of $\kappa = 0.5$ is not a large undersampling rate. This is mainly due to the effect of noise folding and because the IEEE 802.15.4 standard spread spectrum codes are only 16 chips long in each channel (I and Q). For more complex spread spectrum systems with longer chipping sequences (and therefore more potential sparsity) and multiple users and if quantization is included in the signal model, we strongly believe there are cases where the sampling rate may be decreased, while still attaining the same or better BER performance than a classical receiver. This would make compressive signal processing in such systems more attractive.

The main result of this paper is the observation that in a spread spectrum receiver it is possible to use compressive sensing without generating a PRN sequence and mixing it with the received signal. This is possible because a spread spectrum signal has already been spread by the transmitter.

References

[1] E. J. Candes, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

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

[3] M. Davenport, P. Boufounos, M. Wakin, and R. Baraniuk, “Signal processing with compressive measurements,” *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 445–460, Apr. 2010.

[4] *IEEE Std. 802.15.4-2006*, The Institute of Electrical and Electronics Engineers, Inc., 2006, available online: http://standards.ieee.com.

[5] J. Paredes, G. Arce, and Z. Wang, "Ultra-wideband compressed sensing: Channel estimation," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 3, pp. 383–395, Oct. 2007.

[6] P. Zhang, Z. Hu, R. Qiu, and B. Sadler, "A compressed sensing based ultra-wideband communication system," in *Proc. 2009 IEEE International Conf. on Communications*, Piscataway, NJ, USA, 2009, pp. 4239–4243.

[7] V. Aggarwal, L. Applebaum, A. Bennatan, A. Calderbank, S. Howard, and S. Searle, "Enhanced CDMA communications using compressed-sensing reconstruction methods," in *47th Annual Allerton Conf. on Communication, Control, and Computing*, Sep. 2009, pp. 1211–1215.

[8] X. Li, A. Rueetschi, Y. C. Eldar, and A. Scaglione, "GPS Signal Acquisition via Compressive Multichannel Sampling," *Physical Communication*, vol. 5, no. 2, pp. 173–184, 2012.

[9] P. Pankiewicz, T. Arildsen, and T. Larsen, "Sensitivity of the Random Demodulation Framework to Filter Tolerances," in *19th European Signal Processing Conf. (EUSIPCO)*, Barcelona, Spain, Aug. 2011.

[10] 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.

[11] 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.

[12] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. McGraw Hill, 2008.

[13] M. Davenport, J. Laska, J. Treichler, and R. Baraniuk, "The pros and cons of compressive sensing for wideband signal acquisition: Noise folding versus dynamic range," *Signal Processing, IEEE Transactions on*, vol. 60, no. 9, pp. 4628–4642, 2012.

Paper B

Compressive Sensing for Spread Spectrum Receivers

Karsten Fyhn, Tobias Lindstrøm Jensen, Torben Larsen, Søren Holdt Jensen

The paper has been published in the
IEEE Transactions on Wireless Communication,
vol. 12, no. 5, pp. 2334–2343, May 2013

Abstract

With the advent of ubiquitous computing there are two design parameters of wireless communication devices that become very important: power efficiency and production cost. Compressive sensing enables the receiver in such devices to sample below the Shannon-Nyquist sampling rate, which may lead to a decrease in the two design parameters. This paper investigates the use of Compressive Sensing (CS) in a general Code Division Multiple Access (CDMA) receiver. We show that when using spread spectrum codes in the signal domain, the CS measurement matrix may be simplified. This measurement scheme, named Compressive Spread Spectrum (CSS), allows for a simple, effective receiver design. Furthermore, we numerically evaluate the proposed receiver in terms of bit error rate under different signal to noise ratio conditions and compare it with other receiver structures. These numerical experiments show that though the bit error rate performance is degraded by the subsampling in the CS-enabled receivers, this may be remedied by including quantization in the receiver model. We also study the computational complexity of the proposed receiver design under different sparsity and measurement ratios. Our work shows that it is possible to subsample a CDMA signal using CSS and that in one example the CSS receiver outperforms the classical receiver.

B.1 Introduction

As wireless communication devices are becoming more and more widespread and ubiquitous, the need for power efficiency and low production cost becomes paramount. A power costly operation in wireless communication is the conversion from analog to digital signals - the Analog to Digital Converter (ADC). The classic ADC uses the Shannon-Nyquist sampling theorem to represent an analog signal in digital form. The Shannon-Nyquist sampling theorem states that to perfectly represent an analog signal, it must be sampled at a frequency higher than twice the signal's bandwidth. When this theorem is obeyed, the original analog signal may be reconstructed perfectly from its sampled representation. The Shannon-Nyquist sampling theorem has been the foundation of digital signal processing for more than half a century and is considered a fundamental building block of digital signal processing systems. Recently, a new concept termed Compressive Sensing (CS) [1, 2] has been attracting more and more attention in the signal processing community as it provides an exception to the lower bound on the sampling rate by exploiting sparsity in the signal. If a signal is sparse in some arbitrary basis, it may be sampled at a rate lower than the Nyquist frequency. Sparsity in CS is when a signal is comprised of only a few atoms from a given basis. The sampled signal must be acquired through some linear measurement scheme. Examples of these are random Gaussian, Bernoulli and Rademacher measurement schemes, as well as the Random Demodulator (RD) [3, 4] and the Modulated Wideband Converter [5].

Compressive sensing has primarily been studied in the general signal processing area, and relatively few researchers have looked into its application in communication systems. In [6, 7] the authors examine the use of CS in Ultra-Wideband (UWB) communication systems for channel estimation where the sparsity of the signal lies in the time domain. Others have used compressive sensing for source coding in communication networks, together with network coding [8]. In the spread spectrum area, some researchers have studied the general use of CS for spread spectrum communication systems [9]. However, their work is mainly focused on using CS for fast multi-user detection, rather than subsampling. Another example is in [10], where the authors use CS to decrease the sampling rate of a GPS receiver by exploiting sparsity in the number of possible signal components at the receiver. Their receiver structure is based on possibly complicated hardware filters, which may make their implementation very difficult considering the impact of hardware filters to CS performance [11]. In [12] the authors treat a similar topic where they design spread spectrum codes to enable a base station to perform multi-user detection on a large number of users, of which only a few are active at a time. Their work focuses on simple on-off signalling, i.e. the existence of nodes, rather than communication with them, and solves the multi-user detection problem using an adapted convex optimization algorithm. Their motivation is on increasing the number of active users in a network, rather than decreasing the sampling rate of the ADC. A more applied approach is taken in [13] where compressive signal processing [14] is applied to enable subsampling of an IEEE 802.15.4 Direct Sequence Spread Spectrum (DSSS) signal. In [15] the authors also solve a multiuser detection problem using compressive sensing, but in their work the focus is on the design of possibly complex analog filters. For this paper we focus on keeping the analog part as simple as possible and process the signals in the digital domain instead.

In our work we apply CS to a general CDMA system. We show that a RD implementation may be used to subsample the CDMA signal, but we also develop a simplified version of the RD which performs equally well for CDMA signals but is simpler and cheaper to implement. Our motivation is that by taking fewer samples we may be able to conserve power in the receiver, as can be seen in e.g. Eqn. 13 in [16]. We show the performance of the proposed receiver structure for the simple discrete case, when compared to a classic receiver structure and an RD receiver structure. Then we extend our results to a full RF numerical simulation and demonstrate that the performance is identical in this setting. Due to noise folding the CS approach suffers a penalty for downsampling, but we show that if quantization is taken into account CS outperforms the classic receiver in some cases. We finally investigate the complexity of the developed algorithms and compare the computational cost of the numerical experiments with the theoretically calculated computation cost. Following the paradigm of Reproducible Research [17], all our results and code are made available at `http://www.sparsesampling.com/css`.

To define our notation, let all vectors and matrices be denoted using lower and upper case letters in bold, $\mathbf{x}$ and $\mathbf{X}$, respectively. The Penrose-Moore pseudo-inverse is denoted as $\mathbf{X}^{\dagger}$, the transpose of a real matrix as $\mathbf{X}^{\mathrm{T}}$ and the conjugate transpose of a matrix as $\mathbf{X}^{*}$. The expectation operator is denoted by $\mathbb{E}[\cdot]$.

In the following, we first develop a simple signal model in Section B.2, based on a dictionary of Gold sequences. We then elaborate on what CS is and which reconstruction algorithm we use in the numerical experiments in Section B.3. Furthermore, we define a novel measurement matrix design for spread spectrum receivers and demonstrate numerically how this measurement matrix performs with a Gold dictionary and the Subspace Pursuit reconstruction algorithm. This performance is compared to that of a Rademacher measurement matrix and a RD measurement matrix. This is followed by Section B.4, which includes a simple numerical experiment of the different receiver structures. In Section B.5 we extend the experiment to a full RF simulation with and without quantization. We then analyze the computational complexity of the proposed method in Section B.6, after which we conclude the paper in Section B.7.

B.2 Signal Model

First, we consider a purely discrete model of a spread spectrum communication system. Uncoded information bits are sent in a slotted fashion, with each slot containing a single CDMA signal. The system is assumed to be synchronized, which may be obtained by e.g. having a central node or base station transmit beacons, which signify the beginning and end of slots. This is how mobile phone networks and some wireless sensor networks operate. The receiver is considered non-coherent, as information is encoded in the phase, but we do assume that there is no carrier frequency offset between the transmitter and receiver oscillators. This is of course not a realistic assumption but it is done to keep the system simple. Future work should investigate the impact of oscillator drift on performance. Each slot contains an independent CDMA signal and the slots are decoded sequentially and independently of each other.

For one slot, define a discrete QPSK baseband signal, $\mathbf{x} \in \mathbb{C}^{\mathrm{N}\times 1}$ as:

$$\mathbf{x} = \boldsymbol{\Psi}\boldsymbol{\alpha}, \tag{B.1}$$

where $\boldsymbol{\Psi} \in \mathcal{S}_{\boldsymbol{\Psi}} \subset \{\pm 1\}^{\mathrm{N}\times\mathrm{N}}$ is an orthogonal or near-orthogonal dictionary, containing spreading waveforms for transmission, $\mathcal{S}_{\boldsymbol{\Psi}}$ is the subset of $\{\pm 1\}^{\mathrm{N}\times\mathrm{N}}$ that contains orthogonal or near-orthogonal dictionaries and $\boldsymbol{\alpha} \in \{\pm 1 \pm j, 0\}^{\mathrm{N}\times 1}$ is a sparse vector, that selects which spreading waveform(s) and what QPSK constellation point(s) to send. $\boldsymbol{\alpha}$ is a vector here because we only process one slot at a time and we assume that within a slot, the signal amplitude for each user is constant. That $\boldsymbol{\alpha}$ is assumed to be sparse is justified in some scenarios, which is demonstrated shortly.

An example of a system using the above signal model could be a wireless sensor network in which one node must gather information from any possible neighbors. Each node has a unique CDMA sequence assigned, which it uses to transfer information and each node does not know which neighbors it has, but it knows all possible CDMA sequences. Note that in this signal model $\boldsymbol{\alpha}$ is defined so that all users have identical amplitude. This is not realistic as the distance between nodes might vary a lot, resulting in differences between amplitude in the received signal components. We choose this simplification here but the reconstruction algorithm is not limited by this and also works for sparse vectors with different amplitude components.

In cases where the number of active nodes or users in a network is smaller than the total number of possible users, the vector $\boldsymbol{\alpha}$ may be assumed sparse, which is the enabling factor for CS . Cases such as these arise in e.g. mobile phone networks and wireless sensor networks, where the number of surrounding nodes *may* be large, but is *often* small.

At the receiver the following signal is observed:

$$\mathbf{y} = \boldsymbol{\Theta}\left(\mathbf{x} + \mathbf{w}\right) = \boldsymbol{\Theta}\boldsymbol{\Psi}\boldsymbol{\alpha} + \boldsymbol{\Theta}\mathbf{w}, \tag{B.2}$$

where $\boldsymbol{\Theta}$ is a measurement matrix, which we shall treat later, and $\mathbf{w} \in \mathbb{C}^{\mathrm{N}\times 1}$ is Additive White Gaussian Noise (AWGN) . Notice here that we take into account noise folding as the noise is folded down into the compressed domain together with the signal. This makes the noise colored and has an impact on the demodulation performance, because each time the sampling rate is reduced by one half, the Signal to Noise Ratio (SNR) is decreased by 3 dB [18, 19].

B.2.1 Spread Spectrum Dictionary of Gold Sequences

In spread spectrum signals, a possible dictionary $\boldsymbol{\Psi}$ is a set of Gold sequences, as used in e.g. GPS technology [20]. A set of Gold sequences is a special dictionary of binary sequences with very low auto and cross-correlation properties [21]. To generate a Gold dictionary, two maximum length sequences must be generated by two linear feedback shift registers (LFSR). A maximum length sequence is often denoted an m-sequence (it has m elements), and is a special kind of pseudo-random noise sequence generated by a LFSR, such that it is periodic and produces a sequence of length $2^m - 1$. It is called a maximum length sequence as its period is at maximum length. The reason for the length being $2^m - 1$ rather than 2^m is that the state where all cells are zero must be avoided. To obtain an m-sequence, the LFSR must be carefully chosen as there is no algorithm for ensuring maximum length. However, there are many known LFSR setups for varying choices of m. Furthermore, the two m sequences must be chosen so that their periodic cross-correlation is three-valued and takes on only the values $\{-1, -t, t-2\}$, where:

$$t = \begin{cases} 2^{(m+1)/2} + 1 & \text{for odd } m \text{ and} \\ 2^{(m+2)/2} + 1 & \text{for even } m. \end{cases} \tag{B.3}$$

The set of Gold sequences are then generated using two m-sequences: $\mathbf{g}_1$ and $\mathbf{g}_2$, both of length $N = 2^m - 1$. Each Gold sequence in the set is generated as $\mathbf{g}_1 \oplus \mathbf{g}_i$ (exclusive or), where $\mathbf{g}_i$ is $\mathbf{g}_2$ cyclically shifted by the parameter i. As i can take on values between $1 \leq i \leq 2^m - 1$, each shift constitutes a candidate for the set, resulting in a dictionary as follows: Define a $N \times N$ dictionary of Gold sequences as $\mathbf{\Psi}$, with each column signifying a possible code sequence.

When using such a CDMA dictionary, the received signal must be sampled at a rate corresponding to the chip rate, where a chip is one entry in the received Gold sequences. If $\boldsymbol{\alpha}$ is sparse the information rate of the signal is much lower and it may be possible to decrease the sampling rate by using CS.

In this paper, we use three Gold dictionary sizes: $m = 5, m = 7$ and $m = 10$. The m-sequence feedback sets used to generate these are:

- $m = 5$: $X^5 + X^2 + 1$ and $X^5 + X^4 + X^3 + X^2 + 1$
- $m = 7$: $X^7 + X^6 + 1$ and $X^7 + X^4 + 1$
- $m = 10$: $X^{10} + X^3 + 1$ and $X^{10} + X^9 + X^8 + X^6 + X^3 + X^2 + 1$

The chosen polynomials may be validated by calculating the auto and cross-correlation of the generated dictionaries and verifying that they follow the structure listed in the above.

B.3 Compressive Sensing

CS is a novel sampling scheme, developed to lower the number of samples required to obtain some desired signal. At the heart of CS is the linear sampling scheme, called the measurement matrix. In classic receivers the measurement matrix $\mathbf{\Theta}$ may be modelled as the identity matrix, such that $\mathbf{x}$ is sampled at the chip rate of each channel (I and Q). Here, we shall denote a classic receiver using $\mathbf{\Theta}_1 = \mathbf{I}$, where the subscript 1 denotes no subsampling and $\mathbf{I}$ is the identity matrix of size $\mathrm{N} \times \mathrm{N}$. In CS another measurement matrix is used. Denote by $\mathbf{\Theta}_\kappa \in \mathbb{R}^{\mathrm{M} \times \mathrm{N}}$ a CS measurement matrix, where $\kappa \in \mathbb{N}_1$ is the subsampling ratio when compared to the Nyquist rate and $M = N/\kappa$ (If κ does not divide N, M is rounded to the nearest integer). This measurement matrix is then responsible for mapping the N-dimensional signal $\mathbf{x}$ to a M-dimensional signal $\mathbf{y}$. Normally this would make it impossible to recover the original signal, but under the assumption that $\mathbf{x}$ is sparse in some basis, it is possible to reconstruct the original signal from the sampled, M-dimensional signal $\mathbf{y}$ [1, 2].

Notice that we are not interested in the reconstructed signal, $\mathbf{x}$, but in the sparse vector $\boldsymbol{\alpha}$, which allows us to demodulate the data in the signal. We may obtain an estimate of $\boldsymbol{\alpha}$ by reconstructing the signal from $\mathbf{y}$. Such a reconstruction may be obtained using e.g. a convex optimization problem solver or a greedy algorithm. For this work, we choose the greedy algorithm Subspace Pursuit [22]. This algorithm is chosen due to its good performance in terms of both reconstruction accuracy and running time, as shown in Section B.3.2.

Before explaining the reconstruction algorithm, we return to the measurement matrix and introduce a new measurement scheme which is enabled by the use of CDMA. This new measurement scheme is easier to implement than the RD, but performs almost identically for spread spectrum systems. We call this a Compressive Spread Spectrum (CSS) measurement matrix and explain it further in the following.

B.3.1 Compressive Spread Spectrum Measurement Matrix

In most CS literature a choice of measurement matrix or structure must be made. The Bernoulli or Rademacher distributed measurement matrix is often seen in the theoretical literature, but it is not well suited for practical implementation in a wireless receiver. The Random Demodulator (RD) sampling structure [3, 4] is one of the most well-known measurement matrix structures developed, which is well suited for practical implementation. In the RD a Pseudo-Random Noise (PRN) sequence is mixed with the received signal followed by low-pass filtering. Because a spread spectrum transmitter has already spread the signal before transmission, we show that the RD structure can be improved so that the mixing with a PRN sequence at the receiver may be skipped. This is similar to what is done in [13] with IEEE 802.15.4 signals, which uses Direct-Sequence-Spread-Spectrum (DSSS) signals. These can be viewed as a special class of CDMA signals, which are used to counter interference from blockers in the same frequency band, rather than to distinguish between users or signals.

The proposed measurement matrix may therefore be defined similarly to the definition of the RD matrix in [4]. In their work, the measurement matrix is based on two matrices, **D** and **H**. First, let $\epsilon_0, \epsilon_1, \ldots, \epsilon_N \in \{\pm 1\}$ be the chipping sequence used in the RD for a signal of length N. The mapping $\mathbf{x} \rightarrow \mathbf{Dx}$ signifies the demodulation mapping with the chipping sequence, where **D** is the diagonal matrix:

$$\mathbf{D} = \begin{bmatrix} \epsilon_0 & & & \\ & \epsilon_1 & & \\ & & \ddots & \\ & & & \epsilon_N \end{bmatrix}. \tag{B.4}$$

Second, the **H** matrix denotes the accumulate-and-dump action performed after mixing. Let M denote the number of samples taken and assume here that M divides N. Then each sample is the sum of N/M consecutive entries of the demodulated signal. The matrix performing this sampling action may therefore be defined as an $M \times N$ matrix, with N/M consecutive unit entries in the rth row starting in column $rN/M + 1$

for each $r = 0, 1, \ldots, M-1$. An example with $M = 3$ and $N = 6$ is:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & & & & \\ & & 1 & 1 & & \\ & & & & 1 & 1 \end{bmatrix}. \tag{B.5}$$

The RD is therefore designed to sample an analog signal, so that in a discrete representation this is the equivalent to:

$$\mathbf{y} = \mathbf{HDx}, \tag{B.6}$$

where $\mathbf{x}$ is the Nyquist sampled input signal and $\mathbf{y}$ is the compressively sampled output signal.

The reason for applying a chipping sequence is to spread the signal across the frequency spectrum, so that information is aliased down into the lower frequency area, which is left untouched by the low-pass filtering. In the proposed receiver this mixing is unnecessary because the signal has already been spread at the transmitter. The proposed receiver may therefore be simplified to:

$$\mathbf{y} = \mathbf{Hx}. \tag{B.7}$$

This is significantly simpler to implement in hardware than the RD. Comparing to the notation introduced for the measurement matrix in Section B.2 we therefore have: $\mathbf{\Theta}_\kappa = \mathbf{H}$.

To justify the use of no PRN sequence in the measurement matrix, consider the following. The use of a CDMA dictionary introduces a random-like dictionary matrix, which spreads the signal out so that each sample contains a little bit of the original information signal. This is similar to what the measurement matrix does in CS. Therefore, the sampling process may be rewritten as:

$$\mathbf{y} = \mathbf{Hx} = \mathbf{H\Psi}\boldsymbol{\alpha} = \mathbf{\Theta I}\boldsymbol{\alpha}. \tag{B.8}$$

Here, the measurement matrix becomes $\mathbf{\Theta} = \mathbf{H\Psi}$, i.e. the subsampling matrix and the CDMA codes. The dictionary then becomes the identity matrix. When viewed like this, it is clear that $\mathbf{\Theta}$ and $\mathbf{I}$ are incoherent as the identity matrix only takes out one element in $\mathbf{\Theta}$. Another common mathematical tool for verifying the validity of a measurement scheme for compressive sensing is Restricted-Isometry-Property (RIP). However, the RIP gives a less precise and more conservative boundary between reconstruction success and failure than other bounds, see e.g. the discussions in [23, 24]. Instead, phase-transition diagrams [23] may be used to demonstrate empirically for which levels of sparsity the dictionary and measurement matrix are applicable. In the following, we first define the Subspace Pursuit algorithm and then use phase-transition diagrams to show that the proposed CSS measurement matrix has transitions that are very close to those of the Rademacher and RD measurement matrices for dictionary matrices using Gold sequences.

B.3.2 Subspace Pursuit

To reconstruct the signal a reconstruction algorithm must be chosen. Many different approaches have been developed, but two main classes of reconstruction algorithms are in widespread use: ℓ_1 minimization and greedy algorithms. Often, ℓ_1 minimization provides the best solution, but if the matrices $\mathbf{\Psi}$ and $\mathbf{\Theta}$ are very large, it is much more efficient to use the simpler greedy algorithms [25]. Therefore, we choose to use greedy algorithms in this work.

In [25] an extensive numerical comparison between reconstruction algorithms is performed based on phase transition plots. Their results show that the best performance is attained using ℓ_1 (at least theoretically). Second best is the least angle regression (LARS) algorithm. However, as shown in Table VII in [25], the LARS algorithm is quite slow. A better choice is a Tuned Two Stage Thresholding algorithm, which has good performance and is very fast. In [25], two algorithms in particular are mentioned: CoSaMP and the Subspace Pursuit algorithm. The Subspace Pursuit algorithm from [22] is shown to perform best of the two.

Recall that $\mathbf{\Theta}_\kappa$ is a measurement matrix with N columns and N/κ rows and define $\mathbf{A} = \mathbf{\Theta}_\kappa \mathbf{\Psi}$. Then we define the Subspace Pursuit algorithm as in Algorithm 5[1]. In each algorithm iteration, the pseudo-inverse is calculated as the least-squares solution as this is less computationally demanding.

To demonstrate the performance of the Subspace Pursuit algorithm with the Gold dictionary, we have performed numerical experiments to find the phase transition in the noise-less case for various choices of measurement matrices. The size of Gold dictionary used is $m = 10$, i.e. the dictionary matrix $\mathbf{\Psi}$ is of size 1023×1023. The results are shown in Fig. B.1. For each curve, we generate a surface plot of the rate of success, based on Monte Carlo simulations. In this surface plot, a clear transition curve is evident and to condense the results we only plot the transition curve where the probability of error crosses 0.5. Each surface plot is generated so that new simulations are conducted until the Mean Squared Error (MSE) between the ith and the $(i-1)$th figure is less than 10^{-5}. For each parameter set and in each simulation, an experiment is a success (1) if the MSE between the reconstructed and the received signal is less than 10^{-6} and a failure (0) otherwise. The three measurement matrices used are as follows:

- A Rademacher distributed measurement matrix, with a dense structure where entries are either -1 or 1,
- A RD measurement matrix, with a banded structure, where entries are either -1 or 1 on the band and 0 outside, and

[1] In the first initialization step we choose to take the transpose of $\mathbf{A}$, rather than the Penrose–Moore pseudo-inverse. If instead the Penrose–Moore pseudo-inverse is used, the performance at high values of δ and ρ is increased in Fig. B.1, but so is the computational complexity. This issue is not treated in more detail here, since our problems are assumed to always have low ρ.

Algorithm 5 Subspace Pursuit Algorithm [22]

```
Input:
Sparsity $S$, measurement and dictionary matrices combined $\mathbf{A}$ and received, sampled signal $\mathbf{y}$
Initialization:
$T^0 = \{$indices of the $S$ largest absolute magnitude entries
      in the vector $\mathbf{A}^{\mathrm{T}}\mathbf{y}\}$
$\mathbf{y}_r^0 = \mathbf{y} - \mathbf{A}_{T^0}\mathbf{A}_{T^0}^{\mathrm{T}}\mathbf{y}$
$\ell = 0$
repeat
  $\ell \leftarrow \ell + 1$
  $\tilde{T}^\ell \leftarrow T^{\ell-1} \cup \{$indices of the $S$ largest absolute magnitude
        entries in the vector $\mathbf{A}^{\mathrm{T}}\mathbf{y}_r^{\ell-1}\}$
  $T^\ell \leftarrow \{$indices of the $S$ largest absolute magnitude
        entries in the vector $\mathbf{A}_{\tilde{T}^\ell}^{\dagger}\mathbf{y}\}$
  $\mathbf{y}_r^\ell \leftarrow \mathbf{y} - \mathbf{A}_{T^\ell}\mathbf{A}_{T^\ell}^{\dagger}\mathbf{y}$
until $\|\mathbf{y}_r^\ell\|_2 > \|\mathbf{y}_r^{\ell-1}\|_2, \ell \geq S$
```

- the proposed CSS measurement matrix.

To validate the above results, we have inserted the phase transition line for the Tuned Two Stage Threshold (TST) algorithm from [25] in Fig. B.1[2]. As can be seen our implementation corresponds well with their results and it is clear that the proposed CSS measurement matrix performs close to identically to the RD measurement matrix and that, as previously argued, the $\mathbf{D}$ matrix is unnecessary. Notice also the clear horizontal line in the graph around $\delta = 0.9$ and $\rho = 0.5$. We analyze this irregularity more in Section B.6.

B.4 Discrete Numerical Experiment

In the above analysis, we have focused on the noise-less case and have shown that the presented dictionary and measurement matrix setup does enable CS for certain levels of sparsity. We therefore now return to the original signal model in Eqn. (B.2) and investigate the noisy case by carrying out Bit Error Rate (BER) experiments. In Fig. B.2 a flow chart of the numerical experiment is shown. First, we encode a randomly generated bit sequence $\mathbf{b}$ to form the sparse vector $\boldsymbol{\alpha}$ from Eqn. (B.1). The non-zero positions are chosen randomly from a uniform distribution. Each non-zero position contains a QPSK symbol. Then, $\boldsymbol{\alpha}$ is used to create a CDMA signal using the Gold

[2] Data from http://sparselab.stanford.edu/OptimalTuning/main.htm

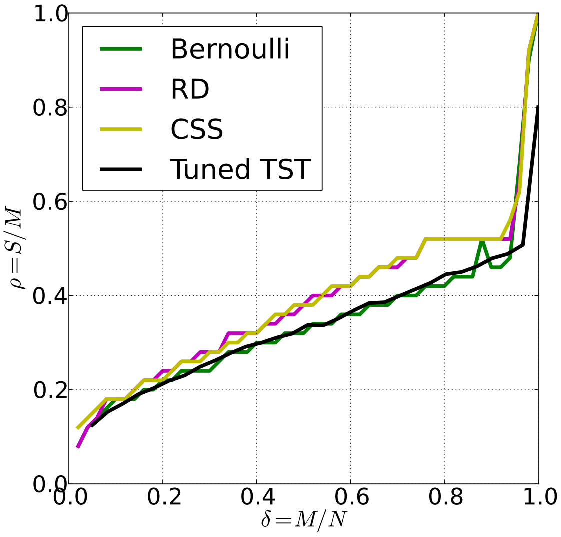


Fig. B.1: Phase Transition Diagrams for the three different measurement matrices (Rademacher, RD and CSS measurement matrix) with dictionary size $m = 10$. The black line is the phase transition line for the Tuned Two Stage Thresholding (TST) algorithm from [25].

dictionary as $\mathbf{x} = \boldsymbol{\Psi}\boldsymbol{\alpha}$. This signal is then corrupted by additive white Gaussian noise, generated according to a chosen SNR value. Here, SNR is defined as follows:

$$\text{SNR} = \mathbb{E}\left[\frac{\|\mathbf{x}\|_2^2}{\|\mathbf{w}\|_2^2}\right] = \frac{\|\mathbf{x}\|_2^2}{N\sigma^2}, \tag{B.9}$$

where $w \sim \mathcal{N}(0, \sigma^2 I)$ with σ^2 the variance of the noise.

At the receiver, the sampling is modelled as in Eqn. (B.2) with multiplication by a measurement matrix. In the simulations we use $\kappa = 2$ or $\kappa = 4$. As is shown in the phase transition plots previously, the method also works for other choices of κ in the noise-less case. However, to clearly demonstrate that our implementation produces the expected 3 dB drop in performance per doubling of κ due to noise folding, we have chosen these two values. A measurement matrix based on samples obtained from a Rademacher distribution introduces colored noise. This decreases the performance, unless the signal is prewhitened before the reconstruction algorithm. This coloring occurs because the rows in the Rademacher matrix are not orthogonal. In the RD and CSS measurement matrices the rows are orthogonal and prewhitening is therefore unnecessary. The prewhitening is achieved by multiplying the received $\mathbf{y}$ vector with a new matrix $\mathbf{P}$ to obtain $\tilde{\mathbf{y}} = \mathbf{P}\mathbf{y}$. By setting $\mathbf{P} = \mathbf{C}^{-1}$, where $\mathbf{C}$ is e.g. the Cholesky factor-

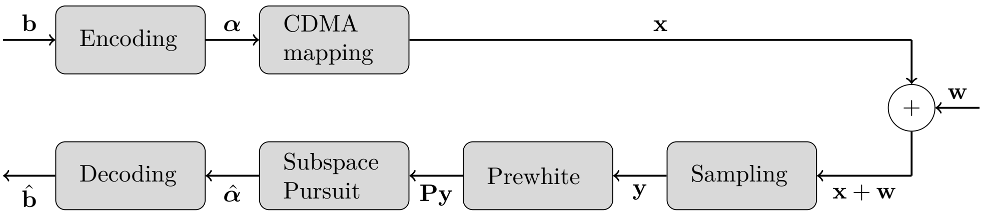
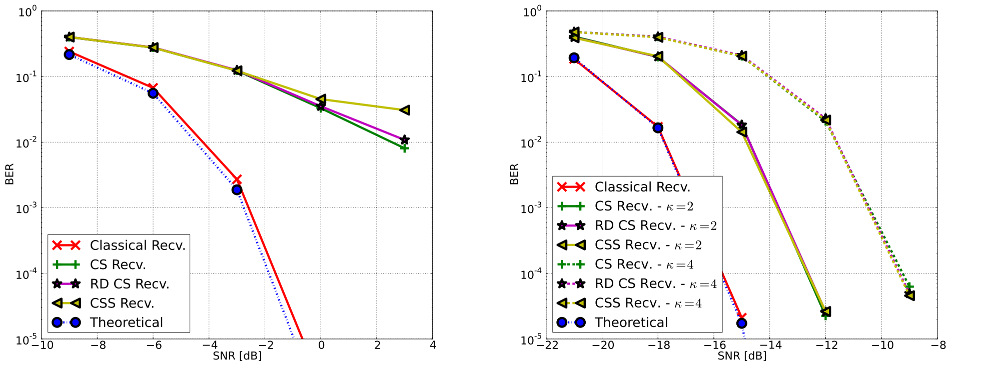


Fig. B.2: Flow chart of the discrete numerical experiment.

ization ($\mathbf{C}\mathbf{C}^T = \mathbf{\Theta}_\kappa \mathbf{\Theta}_\kappa^T$), the variance of the noise term $\tilde{\mathbf{w}} = \mathbf{P}\mathbf{\Theta}_\kappa \mathbf{w}$ from Eqn. (B.2) becomes:

$$\mathbb{E}[\mathbf{P}\mathbf{\Theta}_\kappa \mathbf{w}\mathbf{w}^T \mathbf{\Theta}_\kappa^T \mathbf{P}^T] = \sigma^2 \mathbf{C}^{-1}\mathbf{C}\mathbf{C}^T(\mathbf{C}^{-1})^T = \sigma^2 \mathbf{I}. \tag{B.10}$$

After prewhitening, we reconstruct the sparse vector $\hat{\boldsymbol{\alpha}}$ using the Subspace Pursuit algorithm, which now also must include the $\mathbf{P}$ matrix, i.e. $\mathbf{A} = \mathbf{P}\mathbf{\Theta}_\kappa \mathbf{\Psi}$. It is clear that this extra step increases complexity, but note that this step is only performed for the Rademacher measurement matrix. The $\mathbf{A}$ matrix must be generated anew for each slot because a new measurement matrix $\mathbf{\Theta}_\kappa$ is generated. The RD and CSS measurement matrices skip this step as their rows are orthogonal. After obtaining the sparse vector $\hat{\boldsymbol{\alpha}}$, we are able to decode the original bit sequence, $\hat{\mathbf{b}}$.



(a) $m = 5$, $\kappa = 2$.

(b) $m = 10$, solid lines are for $\kappa = 2$, dashed are $\kappa = 4$.

Fig. B.3: BER versus SNR for different dictionary sizes and choices of κ. CS here is the Rademacher measurement scheme. Simulations were run until 100 bit errors were found for each SNR point.

To validate the obtained results, we compare the numerical results with the theoretical performance for non-coherent MFSK [26]:

$$P_b = \frac{N}{2(N-1)} \frac{1}{N} \sum_{k=2}^{N} (-1)^k \binom{N}{k} \cdot \\ \exp\left(N \cdot \mathrm{SNR} \left(\frac{1}{k} - 1 \right) \right), \tag{B.11}$$

where SNR is the signal to noise ratio. We use the non-coherent formula because the CDMA codes are QPSK modulated. This corresponds to a phase shift of the original codes, which makes the receiver non-coherent. Furthermore, for the above result to hold, we must fix $S = 1$, i.e. the CDMA signal is 1-sparse. Then, instead of performing reconstruction of the sparse $\boldsymbol{\alpha}$, we may instead perform classification as in [13, 14]. This would replace the Subspace Pursuit algorithm with a simpler estimation framework. However, to conserve generality and because we use $S = 10$ later, we continue using the general CS framework and the Subspace Pursuit algorithm.

As dictionary we use Gold sequences with $m \in \{5, 10\}$. This reveals the performance for different dictionary sizes and especially $m = 10$ is interesting as it is the LFSR length used in e.g. GPS. The results of the numerical experiments are shown in Fig. B.3.

As can be seen, for $m = 5$ both the Rademacher, RD and especially the CSS measurement matrix seems to perform poorly. For high SNR values there is more than the expected 3 dB loss per octave due to noise folding. At $m = 10$ the loss is almost exactly 3 dB per halving of the sampling rate. For $m = 10$ we have also included the result for $\kappa = 4$ to show the performance when the sampling rate is reduced to a quarter of the Nyquist sampling rate. Again, the curve follows the previous results for noise folding, as the performance degrades by approximately 3 dB more for all the CS-enabled receiver structures. These results show that the CSS measurement matrix, though simpler than all the other measurement matrices, performs equally well in the above experiments for $m = 10$. For small dictionary sizes, its performance is worse.

B.5 RF Numerical Experiment

To obtain more realistic communication-relevant results, we have extended the above discrete numerical experiment to a full transmitter/receiver simulation with RF up and down-conversion and with root raised cosine pulse shaping and matched filter. This we have done to demonstrate that the results from Fig. B.3 translate to a realistic transmitter/receiver system. The construction of the experiment is visualized in Fig. B.4. This conceptual flow chart also visualizes how the ADC process must be incorporated in a receiver structure to implement the proposed CSS method. The experiment we have conducted is based on a QPSK signal with a chip rate of 10^6 chips per second using a root raised cosine pulse shaping filter with a roll-off factor of 1. This signal is represented in the simulation as sampled at 10 times that rate, to emulate an analog signal.

The signal is up-converted to an RF frequency of 3 MHz, i.e. 3 times the chip rate. The RF signal is sampled at 12 MHz, again to emulate an analog signal. Here, AWGN is added followed by down-conversion again. The down-conversion is implemented as perfect direct down-conversion. This is accomplished by first multiplying with a complex exponential, followed by taking an FFT of the signal. In the output from the FFT, all values above the chip rate are set to 0, after which the inverse FFT is taken. At baseband, the sampling is done by a matched filter based on the same root raised cosine that is used for pulse shaping. The samples are then input to the Subspace Pursuit algorithm, similar to the discrete numerical experiment.

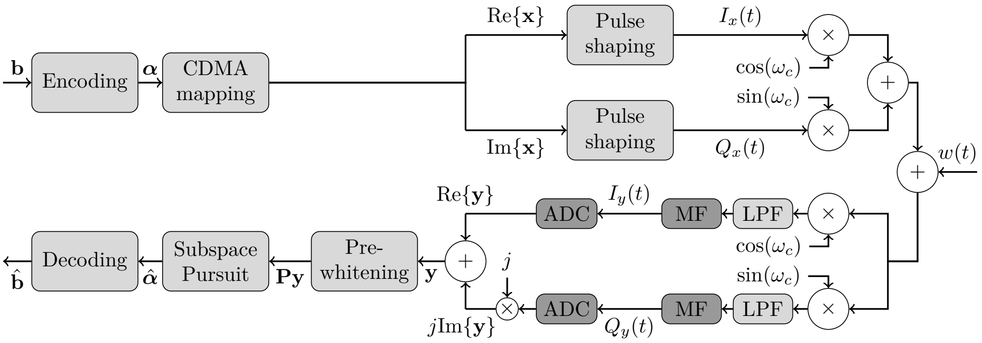
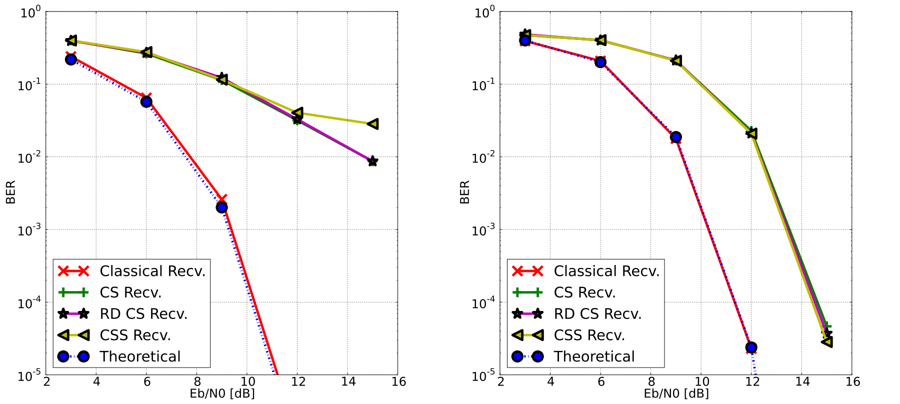


Fig. B.4: Conceptual flow chart of the RF numerical experiment. Note that all continuous variables here are only conceptual. In the numerical experiments they are represented as discrete, oversampled sequences. Here, MF is a matched filter and LPF is a low–pass filter. Dark boxes signify components that must be changed compared to a traditional architecture to enable the CS subsampling described in this work.

The results of the experiment are shown in Fig. B.5. The theoretical curve is calculated using a modified version of the non-coherent MFSK equation used before:

$$P_b = \frac{M}{2(M-1)}\frac{1}{M}\sum_{k=2}^{M}(-1)^k\binom{M}{k}\cdot \\ \exp\left(\log_2(4)\frac{E_b}{N_0}\left(\frac{1}{k}-1\right)\right), \tag{B.12}$$

where E_b/N_0 is the energy per bit per noise spectral density and we multiply E_b/N_0 with $\log_2(4)$ because there are 4 constellation points in QPSK. As can be seen, the results here are close to identical with those for the simpler discrete numerical experiment. Noise folding still gives rise to a penalty, which makes CS a trade-off between sampling rate and BER performance. However, previous work has suggested that quantization may shift the trade-off point, so that CS obtains both the low sampling rate and a better performance than a classical receiver [18]. We investigate this in the following.



(a) $m = 5$, 100 errors found for each point, $\kappa = 2$.

(b) $m = 10$, 100 errors found for each point, $\kappa = 2$.

Fig. B.5: BER versus Eb/N0 for different dictionary sizes. CS here is the Rademacher measurement scheme. Simulations were run until 100 bit errors were found for each Eb/N0 point.

B.5.1 RF Numerical Experiment with Quantization

In [18], it is proposed to combat noise folding with quantization as a CS receiver is able to quantize the sampled signal better, since it takes fewer measurements. By better quantization we mean that if the CS receiver takes half as many samples, it may quantize twice as well at no additional cost. We have investigated this by applying uniform quantization to the RF experiment performed in the previous section. However, as simple QPSK modulation is used, only the sign matters for demodulation and therefore quantization has no effect in the simple case of $S = 1$ used so far. Therefore, we investigate $S = 10$ instead and used 2 bits of quantization per sample (i.e. 4 bits of quantization for CSS as $\kappa = 2$). This is merely intended as an example study to show that when taking into account quantization, CS may perform better than a classical receiver. The result of the numerical experiment is shown in Fig. B.6. As can be seen, quantization makes CS a better alternative in this scenario. The CS approach becomes significantly better for high E_b/N_0 values, because the classical receiver is not able to quantize the signal properly. For comparison, we have also included the same result for a classical receiver with 4 bits of quantization, i.e. the same level of quantization as the CSS receiver. Then it becomes clear that the classical receiver again is the best choice, but remember that it operates at twice the sampling frequency. A CS-enabled receiver can therefore be seen as a trade-off point between sampling rate and dynamic range.

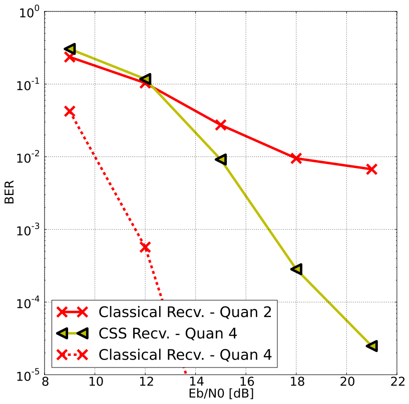


Fig. B.6: BER versus Eb/N0 for a classical receiver and a CSS receiver, both with quantization. $m = 7$, $S = 10$, $\kappa = 2$ and 100 errors found for each E_b/N_0 point. The dotted line is for a classical receiver with 4 bits of quantization per sample.

B.6 Complexity Analysis

To evaluate the Subspace Pursuit algorithm, we investigate the computational complexity of the algorithm, shown in Tab. B.1, where K is the number of iterations used in the Subspace Pursuit algorithm, S is the sparsity, M is the number of measurements taken and N is the number of Nyquist samples.

Table B.1: Computational cost of the Subspace Pursuit algorithm.

Action	**Approx. cost**
Initialization:	
• 1 computation of $\mathbf{A}^{\mathrm{T}}\mathbf{y}$	$4MN$
• 1 computation of $\mathbf{y} - \mathbf{A}_{T^0}\mathbf{A}_{T_0}^{\mathrm{T}}\mathbf{y}$	$2M + 8MS$
Loop:	
• K computations of $\mathbf{A}^T\mathbf{y}$	$4KMN$
• K least squares problems $(\mathbf{A}_{\tilde{T}}^{\dagger}\mathbf{y})$ • K computations of $\mathbf{y} - \mathbf{A}_{T^0}\mathbf{A}_{T_0}^{\dagger}\mathbf{y}$	$K(2M(2S)^2 + 11(2S)^3)$ $K(2M + 4MS + 2MS^2 + 11S^3)$
Total:	$99KS^3 + 4(K+1)MN$ $+2(K+1)M + 4(K+2)MS + 10MKS^2$

The matrix $\mathbf{A}$ is real, but since $\mathbf{y}$ is complex this affects the matrix-vector computations. A matrix-vector product then costs $4MN$ and calculating a residual costs

$2M + 8MS$.

The pseudo-inverse is never calculated, instead a linear least-squares problem is solved using the Singular Value Decomposition (SVD). Solving a least-squares problem with S variables and M observations using the SVD costs [27]:

$$\text{Cost}_{\text{LS with SVD}} \sim 2MS^2 + 11S^3. \tag{B.13}$$

Notice that the first least square problem in the loop takes in $2S$ atoms from the dictionary. The cost of sorting and locating entries is not taken into account here, as those algorithms are more memory then computationally demanding.

It is also important to notice that the problem sizes involved here are very small. Compressive sensing only works for sparse signals, so S is often small compared to M and N. In the examples given here, $N = 1023$ is the largest dimension we have worked with. Because of this, the mathematical model in Tab. B.1 is not adequate, as the computational complexity is instead dominated by programming language overhead, such as the cost of calling different functions. Therefore, it is important to include an extra term: cK, where K is the number of iterations performed and c is some constant that depend on system and programming language overhead.

It is of interest to investigate the required number of iterations, K, of the Subspace Pursuit algorithm, to better understand the cost of using CSS . In Fig. B.7 we show the number of iterations used to generate the results in Fig. B.1. The horizontal line through $\rho = 0.5$ is interesting and unexpected. If we change the input sparsity to the Subspace Pursuit algorithm from S to $2S$, the line moves from $\rho = 0.5$ to $\rho = 0.25$, which means it is related to the number of atoms available to the Subspace Pursuit algorithm in each iteration. It is not related to the dictionary type, as we have obtained

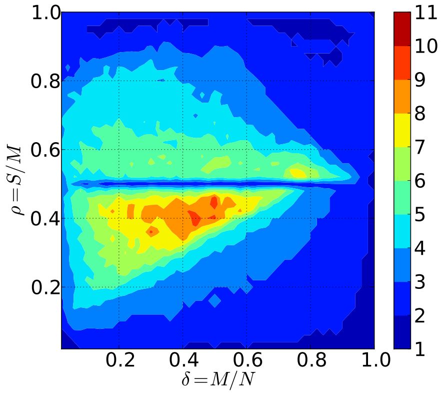


Fig. B.7: Number of Subspace Pursuit iterations for the CSS measurement matrix and Gold Dictionary size $m = 10$.

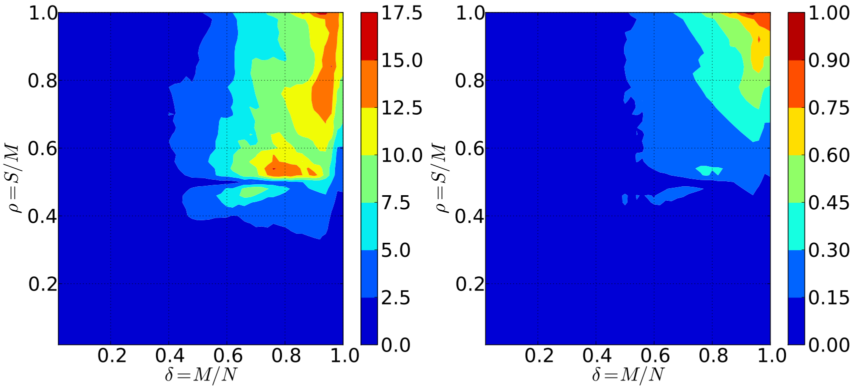


Fig. B.8: Computational cost of the Subspace Pursuit algorithm for the CSS measurement matrix and Gold Dictionary size $m = 10$. The figure to the left shows measured execution time in seconds from the numerical experiment conducted in Fig. B.1 and the figure to the right is generated using the formula for the total computational cost found in Tab. B.1 plus the term cK with $c = 3 \cdot 10^9$. For the figure to the right, the number of iterations of the Subspace Pursuit algorithm, K, is taken from Fig. B.7 and the numbers are normalized.

exactly the same phase transition diagrams and iteration counts with a Haar wavelet packet dictionary. Furthermore, it is not due to a "lucky" initial guess, as the line first emerges in the third iteration of the algorithm. It seems to be an overlooked property of the algorithm, which has gone unnoticed so far because the line in Fig. B.7 lies in the region of Fig. B.1, where the algorithm cannot find the correct solution anyway.

Finally, we have measured the computation time for running the Subspace Pursuit algorithm for the CSS numerical experiment in Fig. B.1. These are compared to the theoretical values obtained by using Tab. B.1. The constant c has been set to $3 \cdot 10^9$, which is a value found to give a good accordance with the numerically found values. It is important to note that this choice of c is very much a function of the algorithm, problem size, programming language and the machine on which the experiment is conducted and should therefore not be seen as a general choice. The result is shown in Fig. B.8. The values in the figure on the right are normalized to one, as they are completely dependent on machine power and are only shown here to visualize how much the computational requirements change with the parameters. As can be seen, the numerically obtained computation times seem to correspond fairly well to the mathematical model. Each point in the above numerical experiment has been run as a simulation on 1 out of 16 threads on computation nodes with 2x Intel Xeon X5570 CPUs and 48GB memory.

B.7 Conclusion

In this work we apply CS to a general CDMA system and we show that it is possible to use a very simple measurement scheme at the receiver side to enable subsampling of the CDMA signal. We show that the performance of the proposed receiver scheme is affected negatively in BER performance, similar to other CS schemes. However, we also show that when taking quantization into account, the proposed receiver model performs better in our example than a classical receiver with the same quantized bit rate. Finally, we investigate the complexity of the developed algorithms and compare the computational cost of the numerical experiments with the theoretically calculated computation cost.

Our work here has shown that CS used in spread spectrum receivers allows for a simplified front-end compared to other state-of-the-art CS sampling designs. Furthermore, we have shown that the problem of noise folding may be remedied in some cases by using quantization. Future work should investigate further which scenarios may benefit from CS and also perform laboratory experiments with the CSS receiver structure. Furthermore, the premise of this work is that taking fewer samples conserves power. This must be validated through laboratory experiments and the power efficiency of the CSS receiver structure should be better evaluated.

References

[1] E. J. Candes, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

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

[3] S. Kirolos, J. Laska, M. Wakin, M. Duarte, D. Baron, T. Ragheb, Y. Massoud, and R. Baraniuk, “Analog-to-information conversion via random demodulation,” in *IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, Oct. 2006, pp. 71–74.

[4] 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.

[5] M. Mishali and Y. C. Eldar, “From theory to practice: Sub-nyquist sampling of sparse wideband analog signals,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 375–391, Apr. 2010.

[6] J. Paredes, G. Arce, and Z. Wang, "Ultra-wideband compressed sensing: Channel estimation," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 3, pp. 383–395, Oct. 2007.

[7] P. Zhang, Z. Hu, R. Qiu, and B. Sadler, "A compressed sensing based ultra-wideband communication system," in *Proc. 2009 IEEE International Conf. on Communications*, Piscataway, NJ, USA, 2009, pp. 4239–4243.

[8] S. Feizi and M. Medard, "A power efficient sensing/communication scheme: Joint source-channel-network coding by using compressive sensing," in *49th Annual Allerton Conf. on Communication, Control, and Computing*, Sep. 2011, pp. 1048–1054.

[9] V. Aggarwal, L. Applebaum, A. Bennatan, A. Calderbank, S. Howard, and S. Searle, "Enhanced CDMA communications using compressed-sensing reconstruction methods," in *47th Annual Allerton Conf. on Communication, Control, and Computing*, Sep. 2009, pp. 1211–1215.

[10] X. Li, A. Rueetschi, Y. C. Eldar, and A. Scaglione, "GPS Signal Acquisition via Compressive Multichannel Sampling," *Physical Communication*, vol. 5, no. 2, pp. 173–184, 2012.

[11] P. Pankiewicz, T. Arildsen, and T. Larsen, "Sensitivity of the Random Demodulation Framework to Filter Tolerances," in *19th European Signal Processing Conf. (EUSIPCO)*, Barcelona, Spain, Aug. 2011.

[12] L. Applebaum, W. U. Bajwa, M. F. Duarte, and R. Calderbank, "Asynchronous code-division random access using convex optimization," *Physical Communication*, vol. 5, no. 2, pp. 129–147, 2012.

[13] K. Fyhn, T. Arildsen, T. Larsen, and S. H. Jensen, "Demodulating subsampled direct sequence spread spectrum signals using compressive signal processing," in *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, Aug. 2012, pp. 2556–2560.

[14] M. Davenport, P. Boufounos, M. Wakin, and R. Baraniuk, "Signal processing with compressive measurements," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 445–460, Apr. 2010.

[15] X. Xie, Y. C. Eldar, and A. Goldsmith, "Reduced-dimension multiuser detection," 2011, submitted for IEEE Transactions on Information Theory, available on arXiv: http://arxiv.org/abs/1109.6303.

[16] P. Kenington and L. Astier, "Power consumption of A/D converters for software radio applications," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 2, pp. 643–650, Mar. 2000.

[17] 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.

[18] J. Treichler, M. A. Davenport, J. N. Laska, and R. G. Baranuik, “Dynamic range and compressive sensing acquisition receivers,” in *Defense Applications of Signal Processing (DASP)*, Coolum, Australia, Jul. 2011.

[19] 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.

[20] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements and Performance*, 2nd ed. Ganga-Jamuna Press, 2010.

[21] R. Gold, “Optimal binary sequences for spread spectrum multiplexing (Corresp.),” *IEEE Trans. Inf. Theory*, vol. 13, no. 4, pp. 619–621, Oct. 1967.

[22] 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.

[23] D. L. Donoho and J. Tanner, “Precise undersampling theorems,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 913–924, Jun. 2010.

[24] J. Blanchard, C. Cartis, and J. Tanner, “Compressed sensing: How sharp is the restricted isometry property?” *SIAM Review*, vol. 53, no. 1, pp. 105–125, 2011.

[25] A. Maleki and D. L. Donoho, “Optimally tuned iterative reconstruction algorithms for compressed sensing,” *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 330–341, Apr. 2010.

[26] B. Sklar, *Digital Communications: Fundamentals and Applications*, ser. Prentice Hall Communications Engineering and Emerging Technologies Series. Prentice-Hall PTR, 2001.

[27] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.

Paper C

Spectral Compressive Sensing with Polar Interpolation

Karsten Fyhn, Hamid Dadkhahi, Marco F. Duarte

The paper has been published in the
Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP) Vancouver, Canada, 2013.

Abstract

Existing approaches to compressive sensing of frequency-sparse signals focuses on signal recovery rather than spectral estimation. Furthermore, the recovery performance is limited by the coherence of the required sparsity dictionaries and by the discretization of the frequency parameter space. In this paper, we introduce a greedy recovery algorithm that leverages a band-exclusion function and a polar interpolation function to address these two issues in spectral compressive sensing. Our algorithm is geared towards line spectral estimation from compressive measurements and outperforms most existing approaches in fidelity and tolerance to noise.

C.1 Introduction

One of the most popular thrusts in compressive sensing (CS) research has focused on the recovery of signals that are spectrally sparse (i.e., that have a sparse frequency-domain representation) from a reduced number of measurements [1–5]. Such *frequency-sparse signals* bring up a novel issue in the formulation of the CS recovery problem: frequency-domain representations have a continuous parameter space, while CS is inherently rooted on discretized signal representations.

Aiming for an increasingly dense sampling of the frequency parameter space introduces performance issues in sparsity-leveraging algorithms. In particular, increasing the resolution of the parameter sampling worsens the coherence of the dictionary that provides sparsity for relevant signals. This both prevents certain algorithms from finding the sparse representation successfully and introduces ambiguity on the choice of representations available for a signal in the dictionary. Initial contributions address such issues by modifying the sparsity prior, the recovery algorithm, or both, to be tailored to the intricacies of the signal representation [5–8].

Interestingly, CS recovery of frequency-sparse signals can be formalized in two different ways: recovery of the signal samples, and recovery of the signal's component frequencies. Previous contributions have almost exclusively focused on the former; their performance for the latter goal is limited by the representation leveraged during CS. Particularly, the required discretization of the parameter space explicitly limits the performance of compressive frequency estimation.

In this paper, we improve over existing approaches by introducing interpolation steps within CS recovery algorithms that break the discretization barrier implicit in CS and are able to improve the quality of frequency parameter estimation. While such interpolation is considered briefly and integrated to a simple recovery algorithm in [5], we introduce a novel polar interpolation approach that leverages the fact that frequency-sparse signals are translation-invariant in the frequency domain. We couple polar interpolation with a more sophisticated CS greedy recovery approach to improve the performance of spectral CS over existing algorithms. We provide experimental evidence that shows improved frequency estimation performance against approaches previously proposed for spectral CS signal recovery: in some cases, our estimates are more precise than those from the baseline approaches, while in other cases we match the precision of the baseline with greatly reduced computational complexity.

C.2 Background and Related Work

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 $\mathbf{A}$ is an $M \times N$ sensing matrix and $M \ll N$. In practice, the measurements are acquired in the presence of noise $\mathbf{z}$, in which case we have $\mathbf{y} = \mathbf{A}\mathbf{f} + \mathbf{z}$.

In many applications, the signal $\mathbf{f}$ is not sparse but has a sparse representation in some dictionary $\mathbf{D}$. In other words, we have $\mathbf{f} = \mathbf{D}\mathbf{x}$, where $\mathbf{x}$ is K-sparse (i.e. $||\mathbf{x}||_0 \leq K$). Under certain conditions on the matrix $\mathbf{A}$ [9, 10], we can recover $\mathbf{x}$ from the measurements $\mathbf{y}$ through the following ℓ_1-minimization problem (which we refer to as ℓ_1-*synthesis*):

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

where ϵ is an upper bound on the noise level $||\mathbf{z}||_2$. Note that optimal recovery of $\mathbf{x}$ from the optimization in (C.1) is feasible only when the elements of the dictionary $\mathbf{D}$ form an orthonormal basis, and thus are incoherent [1, 11]. However, in many applications, the signal of interest is sparse in an overcomplete dictionary or a frame, rather than in a basis.

This paper focuses on frequency-sparse signals, which can be modeled as a superposition of K complex sinusoids with arbitrary frequencies $\tilde{\boldsymbol{\omega}} = \{\omega_1, \omega_2, \ldots, \omega_K\}$. The signal $\mathbf{f} = \begin{bmatrix} f_1 & f_2 & \ldots f_N \end{bmatrix}^T$ is given by

$$f_n = \sum_{k=1}^{K} x_k e^{j2\pi\tilde{\omega}_k n}, \ \tilde{\omega}_k \in [0, 1], \ n \in \{1, 2, \ldots, N\}. \tag{C.2}$$

Such signals are sparse in the discrete-time Fourier transform (DTFT), when defined using an infinite dictionary. In practice, a finite-length representation of the signal is required, and the transform of choice is the discrete Fourier transform (DFT). Unfortunately, the DFT coefficients for such a frequency-sparse signal are sparse only when the frequencies of the constituent sinusoids are integral. One way to remedy this problem would be to employ a dictionary corresponding to a finer discretization of the Fourier representation. We call such a dictionary a DFT frame of redundancy $c \in \mathbb{N}$, containing

$P = c \cdot N$ elements, defined as:

$$\mathbf{D} = \begin{bmatrix}\mathbf{d}(\omega_1) & \mathbf{d}(\omega_2) & \cdots & \mathbf{d}(\omega_P)\end{bmatrix}, \ \omega_p = \frac{p}{P},$$
$$\mathbf{d}(\omega_p) = \begin{bmatrix}d_1(\omega_p) & d_2(\omega_p) & \ldots & d_N(\omega_p)\end{bmatrix}^T, \tag{C.3}$$

where $d_n(\omega) = \frac{1}{\sqrt{N}} e^{j2\pi\omega n}$. However, the DFT frame violates the incoherence requirement for the dictionary [5].

It has recently been shown in [6] that as far as the recovery of signal $\mathbf{f}$ (instead of the sparse coefficient vector $\mathbf{x}$) is concerned, the coherence condition of the dictionary is not necessary, provided that the matrix $\mathbf{D}^H\mathbf{D}$ is sufficiently sparse, where $(\cdot)^H$ designates the Hermitian operation. In this case, the signal $\mathbf{f}$ can be recovered via ℓ_1-*analysis*. However, the matrix $\mathbf{D}^H\mathbf{D}$ is not sufficiently sparse for DFT frames.

Alternatively, one can take advantage of structured sparsity in spectral CS recovery by using a coherence inhibition model [5]. The resulting structured iterative hard thresholding (SIHT) algorithm can recover the frequency-sparse signal with a DFT frame by avoiding dictionary elements with high coherence. A variation of this method uses a band-exclusion function to achieve the same avoidance [8]. We can define the η-coherence band of the index set S as

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

where $\mu(i,k) = |\langle \mathbf{d}(\omega_i), \mathbf{d}(\omega_k)\rangle|$ is the coherence between two atoms in the dictionary. The authors use the band-exclusion function to avoid selecting coherent dictionary elements in various greedy algorithms, including Band-excluded Orthogonal Matching Pursuit (BOMP).

More recently, it has been shown that one can recover a frequency-sparse signal from a random subset of its samples using atomic norm minimization [7]. The atomic norm of $\mathbf{f}$ is defined as the size of the smallest scaled convex hull of a continuous dictionary of complex exponentials. Thus, the recovery procedure searches over a continuous dictionary rather than a discretized one. The atomic norm minimization can be implemented as a semidefinite program (SDP), which can be computationally expensive. In addition, this formulation does not account for measurement noise, and it is not clear if guarantees can be given for arbitrary measurement settings. Nonetheless, [7] motivates our formulation of algorithms that push past the discretization of the frequency parameter space.

C.3 Polar Interpolation for Frequency Estimation

One way to remedy the discretization of the frequency parameter space implicit in CS is to use interpolation. In [12], 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 closely approximated by a point in such arc.

The complex exponentials that compose a DFT frame also form a manifold over a hypersphere, and thus can be approximated by an arc of a circle. This is motivated by the fact that complex exponentials have translation-invariant Fourier transforms, which correspond to an isometric rotation of the time-domain vectors. In this case, the DFT frame samples the frequency parameter space with a steps size $\Delta = 1/c$, and we approximate a segment of the manifold $\mathbf{d}(\tilde{\omega}_i) : \tilde{\omega}_i \in [\omega_p - \frac{\Delta}{2}, \omega_p + \frac{\Delta}{2}]$ by a circular arc containing the three exponentials $\{\mathbf{d}(\omega_p - \frac{\Delta}{2}), \mathbf{d}(\omega_p), \mathbf{d}(\omega_p + \frac{\Delta}{2})\}$. Making use of trigonometric identities, the polar interpolator approximates exponentials $\mathbf{d}(\tilde{\omega}_i)$, $\tilde{\omega}_i \in [\omega_p - \frac{\Delta}{2}, \omega_p + \frac{\Delta}{2}]$, using linear combinations of the three exponentials [12]:

$$\mathbf{d}(\tilde{\omega}_i) \approx \mathbf{c}(\omega_p) + r\cos\left(\frac{2\tilde{\omega}}{\Delta}\theta\right)\mathbf{u}(\omega_p) + r\sin\left(\frac{2\tilde{\omega}}{\Delta}\theta\right)\mathbf{v}(\omega_q),$$

$$\begin{bmatrix} \mathbf{c}(\omega_p)^T \\ \mathbf{u}(\omega_p)^T \\ \mathbf{v}(\omega_p)^T \end{bmatrix} = \begin{bmatrix} 1 & r\cos(\theta) & -r\sin(\theta) \\ 1 & r & 0 \\ 1 & r\cos(\theta) & r\sin(\theta) \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{d}(\omega_q - \frac{\Delta}{2})^T \\ \mathbf{d}(\omega_p)^T \\ \mathbf{d}(\omega_p + \frac{\Delta}{2})^T \end{bmatrix},$$

where r is the ℓ_2 norm of each element of the dictionary and θ is the angle between $\mathbf{d}(\omega_p)$ and $\mathbf{d}(\omega_p - \frac{\Delta}{2})$. In order to extend the above approximation to sums of J exponentials with frequencies $\Omega = \{\omega_1, \omega_2, \ldots, \omega_J\}$, we define:

$$\tilde{\mathbf{f}} = \mathbf{C}(\Omega)\boldsymbol{\alpha} - \mathbf{U}(\Omega)\boldsymbol{\beta} - \mathbf{V}(\Omega)\boldsymbol{\gamma}, \tag{C.5}$$

$$\begin{aligned} \mathbf{C}(\Omega) &= \begin{bmatrix} \mathbf{c}(\omega_1) & \mathbf{c}(\omega_2) & \cdots & \mathbf{c}(\omega_J) \end{bmatrix}, \\ \mathbf{U}(\Omega) &= \begin{bmatrix} \mathbf{u}(\omega_1) & \mathbf{u}(\omega_2) & \cdots & \mathbf{u}(\omega_J) \end{bmatrix}, \\ \mathbf{V}(\Omega) &= \begin{bmatrix} \mathbf{v}(\omega_1) & \mathbf{v}(\omega_2) & \cdots & \mathbf{v}(\omega_J) \end{bmatrix}, \end{aligned} \tag{C.6}$$

where $\boldsymbol{\alpha}$ represents the amplitude of the signal and $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ controls the frequency translations. The three coefficient vectors can be estimated using the following constrained convex optimization problem [12]:

$$\begin{aligned}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) &= \mathrm{T}(\mathbf{y}, \mathbf{A}, \Omega) \\ &= \operatorname*{argmin}_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}} \frac{1}{2\sigma^2} ||\mathbf{y} - \mathbf{A}\tilde{\mathbf{f}}||_2^2 + ||\boldsymbol{\alpha}||_1 \\ &\text{s.t.} \left\{ \begin{matrix} \alpha_j \geq 0, \\ \sqrt{\beta_j^2 + \gamma_j^2} \leq \alpha_j^2 r^2, \\ \alpha_j r \cos(\theta) \leq \beta_j \leq \alpha_j r, \end{matrix} \right\} \text{ for } j = 1, \ldots, J,\end{aligned} \tag{C.7}$$

where $\mathbf{A}$ is the measurement matrix, and $\mathbf{y}$ is the received compressed signal. 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 $\mathbf{d}(\omega_j)$ is restricted to the interval $[0, \theta]$. It is necessary to scale $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ after the optimization problem [12]:

$$(\beta_j, \gamma_j) \leftarrow \left(\frac{\beta_j \alpha_j r}{\sqrt{\beta_j^2 + \gamma_j^2}}, \frac{\gamma_j \alpha_j r}{\sqrt{\beta_j^2 + \gamma_j^2}} \right). \tag{C.8}$$

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 (C.6) and the frequency estimates using the one-to-one relation

$$\alpha_j \mathbf{c}(\omega_j) + \beta_j \mathbf{u}(\omega_j) + \gamma_j \mathbf{v}(\omega_j) = \alpha_j \mathbf{d} \left(\omega_j + \tfrac{\Delta}{2\theta} \tan^{-1}(\tfrac{\gamma_j}{\beta_j}) \right). \tag{C.9}$$

The optimization (C.7), when applied with all parameter values used in the dictionary D, is named continuous basis pursuit (CBP) in [12]:

$$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \mathrm{T}(\mathbf{y}, \mathbf{A}, \Omega_{CBP}), \tag{C.10}$$

where $\Omega_{CBP} = \{\omega_1, \omega_2, \ldots, \omega_P\}$ is the set of all frequencies that appear in the DFT frame for our application of interest. As posed, CBP has a high computational complexity: it operates on matrices of size $3N$, whereas other CS algorithms operate on matrices of size N. However, its interpolation step has one important advantage: translation-invariance and interpolation enables CBP to reconstruct arbitrary frequency-sparse signal while requiring only a small subset of 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.

C.4 Band-Excluded Interpolating Subspace Pursuit

We incorporate the convex optimization (C.7) and band-exclusion (C.4) in a Subspace Pursuit algorithm [13]. We call this algorithm Band-Excluded Interpolating Subspace Pursuit (BISP), which is shown in Algorithm 6.

In the algorithm initialization, the best K correlating atoms are found and stored in S^n by generating a proxy for the sparse signal. The K atoms are found iteratively, which deviates from the original Subspace Pursuit algorithm where the K atoms are found in one step. In each iteration, we trim the proxy based on the found atom and the band exclusion function $B_\eta(S)$, as defined in (C.4). In the main loop, we find the K best atom indices and add them to S^n. From S^n, we form a set Ω consisting of all frequencies corresponding to the indices in S^n along with all adjacent indices. This is necessary because the frequencies present in $\mathbf{y}$ 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 (C.7) along with the measurement matrix and the received signal.

In practice, we found that for noisy measurements it is often preferable to move the minimization objective $||\mathbf{y} - \mathbf{A}\tilde{\mathbf{f}}||_2^2$ in (C.7) into a constraint. Moving this fidelity measure from the objective function to a constraint causes the optimization to return the sparsest set of coefficients that yields measurements within the noise range of the observation. If the output is non-existent or trivial, we move the fidelity metric from the objective function to the constraint (or vice versa).

C.5 Numerical Experiments

To evaluate Algorithm 6, we have performed two numerical experiments.[1] We generated frequency-sparse signals of length $N = 100$ containing $K = 4$ complex sinusoids with frequencies selected uniformly at random. We used a DFT frame with $c = 5$ ($\Delta = 0.2$Hz), and considered well-separated tones so that no two tones are closer than 1Hz of each other. We performed Monte Carlo experiments and averaged over 30 experiments. As measurement matrix[2] we used a Gaussian matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$. We set $M = \kappa N$, where $\kappa \in (0, 1]$ is the CS subsampling rate. We compare our proposed Algorithm 6 with six state-of-the-art methods: ℓ_1-synthesis, ℓ_1-analysis, SIHT, SDP, BOMP, and CBP. As performance measure, we use the Hungarian algorithm [15, 16] to find the best matching between the estimated and true frequencies. For the algorithms that return a dense DFT coefficient vector or a reconstructed signal (ℓ_1-synthesis, ℓ_1-analysis, SIHT,

[1] The documentation and code for these experiments are made freely available at `http://www.sparsesampling.com/scspi`, following the principle of Reproducible Research [14].

[2] For the SDP algorithm we used a random subsampling matrix, as the algorithm is only defined for such a measurement matrix. The authors would like to thank Gongguo Tang for providing the implementation of SDP.

Algorithm 6 BISP

```
INPUTS: Compressed signal $\mathbf{y}$, sparsity $K$, measurement matrix $\mathbf{A}$ and spacing between dictionary elements $\Delta$.
OUTPUTS: Reconstructed signal $\tilde{\mathbf{f}}$ and frequency estimates $\tilde{\boldsymbol{\omega}}$.
INITIALIZE: $\boldsymbol{\Phi} = \mathbf{AD}$, $i = 1, S^0 = \emptyset$
while $i \leq K$ do
    $S^0 = S^0 \cup \arg\max_i |\langle \mathbf{y}, \boldsymbol{\Phi}_i \rangle|,\ i \notin B_0(S^0),\ i = i + 1$
end while
$\mathbf{y}_r^0 = \mathbf{y} - \boldsymbol{\Phi}_{S^0} \boldsymbol{\Phi}_{S^0}^{\dagger} \mathbf{y},\ \ n = 1$
LOOP:
repeat
    $i = 1, S^n = S^{n-1}$
    while $i \leq K$ do
        $S^n = S^n \cup \arg\max_i |\langle \mathbf{y}, \boldsymbol{\Phi}_i \rangle|,\ i \notin B_0(S^n),\ i = i + 1$
    end while
    $\mathbf{a} = (\boldsymbol{\Phi}_{S^n})^{\dagger} \mathbf{y}$
    $S^n = \mathrm{supp}(\mathrm{thresh}(\mathbf{a}, K))$
    $\Omega = \cup \{\Delta(s-1), \Delta s, \Delta(s+1) | s \in S^n\}$
    From $\mathrm{T}(\mathbf{y}, \mathbf{A}, \Omega)$ obtain $\tilde{\mathbf{f}}$ and $\tilde{\boldsymbol{\omega}}$ using (C.9) and (C.6)
    $\mathbf{y}_r^n = \mathbf{y} - \mathbf{A}\tilde{\mathbf{f}},\ n = n + 1$
until $||\mathbf{y}_r^n||_2 > ||\mathbf{y}_r^{n-1}||_2 \vee n \leq K$
```

and SDP), we apply the MUSIC algorithm [17] on the reconstructed signal to estimate its frequencies. In the BISP and BOMP algorithms, we exclude atoms with coherence $\eta > 0.25$ using (C.4).

For the first experiment, we explore a range of subsampling ratios κ with noiseless measurements to verify the level of compression that allows for successful estimation. We set $\epsilon = 10^{-10}$ for the relevant algorithms. The result of the numerical experiment is shown in Figure C.1. In the noiseless case, SDP obtains the best result. The polar interpolation algorithms (CBP and BISP) both converge to a given estimation precision, which corresponds to the level of approximation error. When the number of measurements M is sufficiently small, CBP outperforms ℓ_1-synthesis. The performance of BOMP and SIHT is worst among the algorithms tested. Surprisingly, while the DFT coefficients $\mathbf{x}$ found by ℓ_1-synthesis are not sparse and do not match the original frequencies, the signal $\mathbf{f}$ is still reconstructed accurately, and so the MUSIC algorithm recovers the frequencies adequately.

For the second experiment, we include measurement noise in the signal model. We fix $\kappa = 0.5$ and vary the signal-to-noise ratio (SNR) from 0 to 20 dB. In the noisy case, the polar interpolation algorithms perform best. This is because their interpolation step relies less on the sparsity of the signal and more on the known signal model and

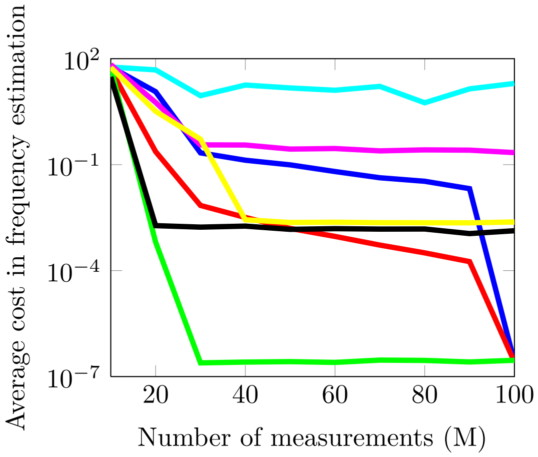


Fig. C.1: Frequency estimation performance in noise-less case. The legend is shown in Fig. C.2.

	Noiseless	Noisy
ℓ_1-analysis	9.5245	8.8222
ℓ_1-synthesis	2.9082	2.7340
SIHT	0.2628	0.1499
SDP	8.2355	9.9796
BOMP	0.0141	0.0101
CBP	46.9645	40.3477
BISP	5.4265	1.4060

Table C.1: Average computation times in seconds.

the fitting to a circle on the manifold. Additionally, the presence of noise renders the measurements non-sparse in the dictionaries used by the non-interpolating algorithms, hindering their performance.

The computation time of the algorithms is also of importance, and we have listed the average computation times in Table C.1. We observed that most algorithms exhibit computation time roughly independent of M, with the exception of ℓ_1-synthesis and CBP[3]. The table shows that the excellent performance of SDP in Figure C.1 is tempered by its high computational complexity, as well as its lack of flexibility on the measurement scheme. Moreover, the relaxation in BISP that accounts for the presence of noise reduces its computation time, increasing its performance advantage over SDP and CBP.

[3] See results at http://www.sparsesampling.com/scspi.

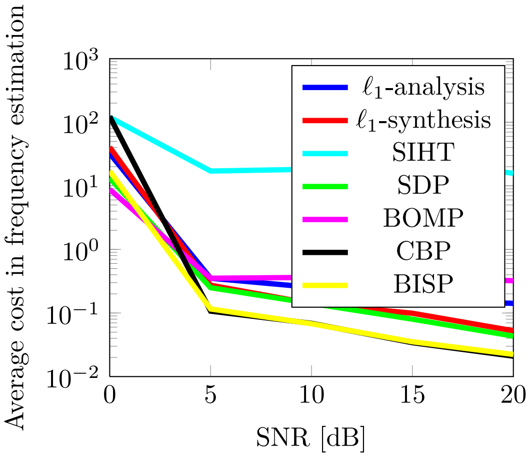


Fig. C.2: Frequency estimation performance in noisy case.

References

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

[2] 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.

[3] M. Mishali and Y. C. Eldar, “From theory to practice: Sub-nyquist sampling of sparse wideband analog signals,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 375–391, Apr. 2010.

[4] M. Wakin, S. Becker, E. Nakamura, M. Grant, E. Sovero, D. Ching, J. Yoo, J. Romberg, A. Emami-Neyestanak, and E. Candès, “A nonuniform sampler for wideband spectrally-sparse environments,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2012, to appear.

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

[6] E. J. Candès, Y. C. Eldar, D. Needell, and P. Randall, “Compressed sensing with coherent and redundant dictionaries,” *Applied and Computational Harmonic Analysis*, vol. 31, no. 1, pp. 59 – 73, 2011.

[7] G. Tang, B. Bhaskar, P. Shah, and B. Recht, “Compressed sensing off the grid,” *Preprint (available at: http://arxiv.org/abs/1207.6053)*, 2012. [Online]. Available: http://arxiv.org/abs/1207.6053

[8] 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.

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

[10] E. J. Candes, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] H. W. Kuhn and B. Yaw, “The hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, pp. 83–97, 1955.

[16] J. Munkres, “Algorithms for the assignment and transportation problems,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[17] P. Stoica and R. L. Moses, *Introduction to spectral analysis.* Upper Saddle River, NJ: Prentice Hall, 1997.

Paper D

Compressive Time Delay Estimation using Interpolation

Karsten Fyhn, Marco F. Duarte, Søren Holdt Jensen

The paper has been submitted to the
1st IEEE Global Conference on Signal and Information Processing (GlobalSIP) Austin, Texas, USA, 2013.

Abstract

We show that compressive sensing (CS) applied to time delay estimation (TDE) simultaneously enables a reduction in the sampling frequency and preserves good estimation precision. With CS, we seek to recover signals and parameters from an under-determined system of linear equations by assuming sparsity in a known dictionary. A common problem in CS is that the observed signals may not be sparsely representable in the dictionary. This problem also occurs in TDE as the delay parameter is a continuous parameter. We remedy this issue by combining CS with interpolation.

D.1 Problem Formulation

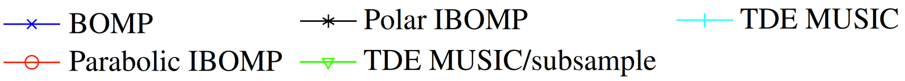
Let the received time-domain analog signal be defined as

$$f(t;\boldsymbol{\alpha},\boldsymbol{\tau}) = \sum_{i=1}^{K} \alpha_i \cdot g(t-\tau_i) + n(t), \tag{D.1}$$

where $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \cdots, \alpha_K\}$ are the unknown signal amplitudes, $\boldsymbol{\tau} = \{\tau_1, \tau_2, \cdots, \tau_K\}$ are the unknown signal delays in time, $g(t)$ is a known signal waveform and $n(t)$ is the noise. The task of the estimation algorithm is then to estimate $\boldsymbol{\alpha}$ and $\boldsymbol{\tau}$ from a sampled version of Eqn. (D.1). Depending on the bandwidth of $g(t)$, the required sampling rate to estimate the delays to a desired precision may be high. If we assume that only a few signal components are active, i.e. K is small, we may use CS to achieve the desired precision at a lower sampling rate. With a CS receiver the received signal is $\mathbf{y} = \boldsymbol{\Phi}\mathbf{f}$, where $\mathbf{f} \in \mathbb{C}^N$ is the Nyquist sampled version of Eqn. (D.1), $\mathbf{y} \in \mathbb{C}^M$ is the received signal and $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$ is the CS measurement matrix.

To enable reconstruction CS requires a sparsifying dictionary $\boldsymbol{\Psi} \in \mathbb{C}^{N \times N}$. In the case of TDE the dictionary is a circulant matrix of delayed waveforms. Since the delay parameter is continuous the received signal may not be sparsely representable by the dictionary, which may lower performance.

Our contribution is bridging the work on CS and interpolation to improve estimator precision in TDE while keeping the sampling frequency low. This is achieved by incorporating an interpolation step in a greedy algorithm. In each iteration of the algorithm, after finding the strongest correlating atom in the dictionary, we propose to use an interpolation function to improve the estimation precision. There are many possible choices of interpolation functions. In this work we compare two such functions: second order polynomial and polar interpolation based on a manifold model.



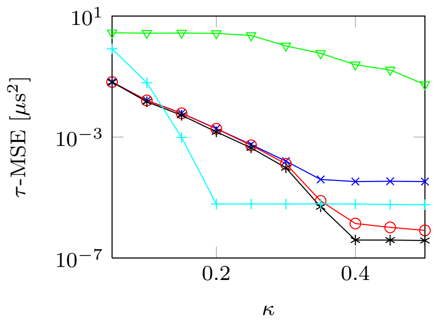


Fig. D.1: τ-MSE vs. κ.

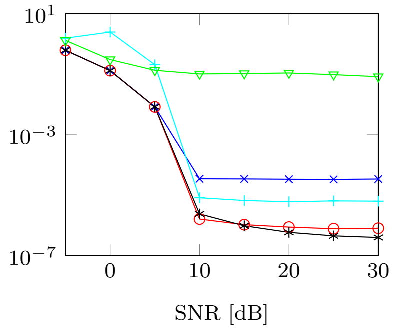


Fig. D.2: τ-MSE vs. SNR, $\kappa = 0.5$.

D.2 Numerical Simulations

We compare five delay estimators: 1) *BOMP* is an existing greedy algorithm without interpolation, 2) *PaIBOMP* adds to BOMP parabolic interpolation, 3) *PoIBOMP* uses polar interpolation, 4) *TDE MUSIC* reconstructs the signal using ℓ_1-minimization and then estimates the delays using the MUSIC algorithm, and 5) *TDE MUSIC/subsample* directly downsamples the signal by a factor of N/M and estimates the delays using the MUSIC algorithm. The last algorithm shows that direct downsampling fails due to aliasing.

In the first experiments we assume a noise-free signal and vary the number of measurements $M = \kappa N$, where $\kappa \in [0, 1)$ is the CS subsampling rate. Fig. D.1 shows the performance of the five estimators by computing the time delay mean squared error (τ-MSE) between the true and estimated value of the time delay. This corresponds to the sample variance of the estimators and is a measure of estimator precision. All four CS estimators allow for subsampling while maintaining good estimation precision. TDE MUSIC performs best for low κ, while the interpolation algorithms perform best as κ increases.

For the second experiment we include additive white Gaussian measurement noise in the signal model. We fix $\kappa = 0.5$ and vary the signal-to-noise ratio (SNR). Fig. D.2 shows that the algorithms are affected by noise, but as SNR increases they converge towards the results for $\kappa = 0.5$ in Fig. D.1.

These numerical results show that CS coupled with interpolation enables subsampling while maintaining a desired estimation precision. For full details, see our technical report [1] and our website `www.sparsesampling.com/tde`.

References

[1] K. Fyhn, M. F. Duarte, and S. H. Jensen, "Compressive time delay estimation using interpolation," Joint work by Aalborg University, Denmark and University of Massachusetts Amherst, USA, Tech. Rep., 2013, available at http://arxiv.org/abs/1306.2434.

Paper E

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

Karsten Fyhn, Marco F. Duarte, Søren Holdt Jensen

The paper has been submitted to
IEEE Transactions on Signal Processing, 2013.

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.

E.1 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 recover $\mathbf{x}$ from the measurements $\mathbf{y}$ through the following ℓ_1-minimization problem (which we refer to as ℓ_1-*synthesis*):

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

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. (E.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 Euclidean 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:

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

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 E.3 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 E.4 in which we introduce an iterative, greedy algorithm based on interpolation. In Section E.5 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 E.6 concludes the paper.

E.2 Previous Work

Prior work on the problem of sparsity in parametric dictionaries includes [7, 8], which uses 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:

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

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

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

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, $\mathbf{f} = \mathbf{D}\mathbf{x}$:

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

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. (E.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. (E.1) followed by estimation.

E.3 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:

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

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. (E.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 $b_p = [\![b_n]\!] = \operatorname{round}\left(\frac{b_n}{\Delta}\right)\Delta$, 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]:

$$\tilde{\mathbf{g}}(b_n) = \mathbf{c}(b_p) + r\cos\left(\frac{2\Delta_n}{\Delta}\theta\right)\mathbf{u}(b_p) + r\sin\left(\frac{2\Delta_n}{\Delta}\theta\right)\mathbf{v}(b_p),$$

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

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

$$r = \|\mathbf{g}(b_p)\|_2,$$
$$\theta = \frac{\mathrm{Re}\{\langle \mathbf{g}(b_p), \mathbf{g}(b_p - \frac{\Delta}{2})\rangle\}}{\|\mathbf{g}(b_p)\|_2 \cdot \|\mathbf{g}(b_p - \frac{\Delta}{2})\|_2}$$

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 $[\![\Omega_J]\!] = \{[\![b_1]\!], [\![b_2]\!], \dots, [\![b_J]\!]\}$:

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

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

E.3.1 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]:

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

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]:

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

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. (E.8) and the frequency estimates using the one-to-one relation:

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

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.

E.3.2 Advanced convex optimization problem

One of the contributions of this paper is an improved convex optimization formulation of Eqn. (E.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:

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

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

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

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

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:

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

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. (E.9) using variable substitution:

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

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:

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

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):

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

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. (E.10-E.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.

E.4 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 (E.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 7. 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. (E.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 (E.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 $\tilde{\mathbf{b}}$.

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. (E.2) as follows:

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

Algorithm 7 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 $\xi$.

**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. (E.17) to obtain $\mathbf{x}$
Obtain $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{b}}$ using (E.11) and (E.8)
```

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

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

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. (E.7). We reformulate those equations to a linear least squares problem:

$$\mathbf{y}_{res,n} \approx \mathbf{A}\left[\mathbf{g}(b_p - \tfrac{\Delta}{2}) \quad \mathbf{g}(b_p) \quad \mathbf{g}(b_p + \tfrac{\Delta}{2})\right] \left(\begin{bmatrix} 1 & r\cos(\theta) & -r\sin(\theta) \\ 1 & r & 0 \\ 1 & r\cos(\theta) & r\sin(\theta) \end{bmatrix}^{-1} \right)^{T} \mathbf{x},$$

$$\mathbf{x} = \begin{bmatrix} a_i \\ a_i r\cos\left(\frac{2\Delta_n\theta}{\Delta}\right) \\ a_i r\sin\left(\frac{2\Delta_n\theta}{\Delta}\right) \end{bmatrix}. \tag{E.21}$$

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. (E.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:

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

We term the interpolation function in Eqn. (E.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. (E.19) or polar interpolation as in Eqn. (E.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.

E.5 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. (E.6).

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

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

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:

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

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 $\mathbf{g}(b_n)$:

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

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

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

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

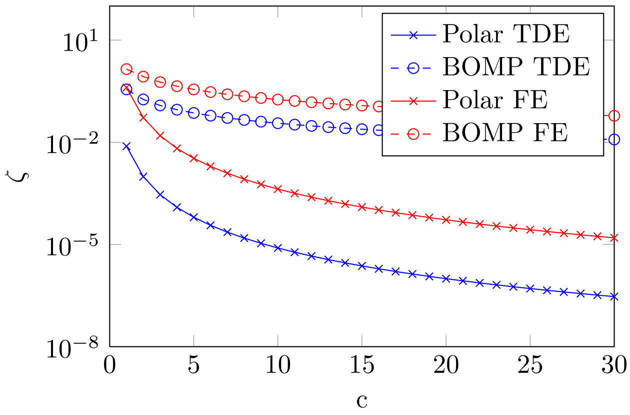


Fig. E.1: Polar estimation approximation error analysis.

E.5.1 ζ and λ analysis

We first investigate the approximation error parameter, ζ. We have conducted numerical experiments on the bound in Eqn. (E.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. (E.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. (E.15). The result of the experiment is shown in Fig. E.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. (E.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, $i = 1, \ldots, N$:

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

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

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:

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

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

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:

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

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

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

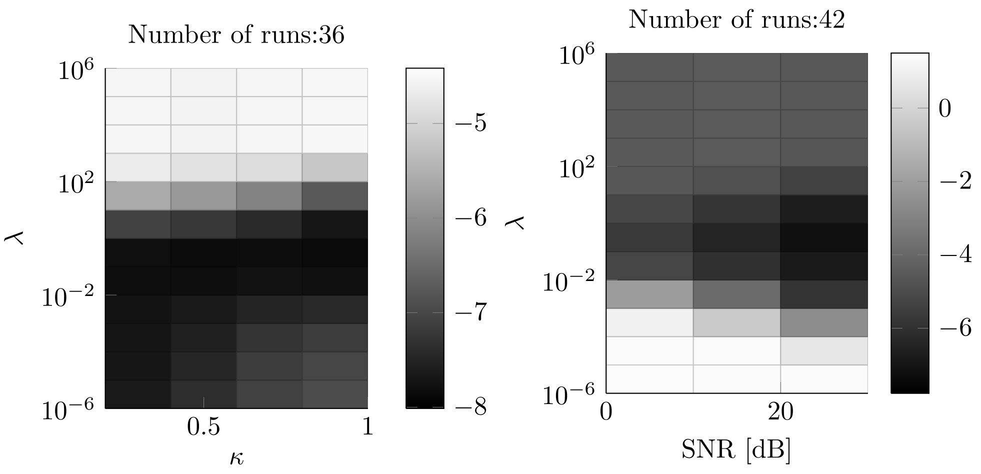


Fig. E.2: λ analysis for CCBP. Left figure is with SNR 1000 and right figure is with $\lambda = 1$. The z-axis is the mean squared error of the parameter estimate **b**-MSE. The scaling on the z-axis is microseconds squared on a logarithmic scale.

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 **b** parameter, termed the **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. (E.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. E.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. E.2. The colorbar signifies the **b**-MSE 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.

E.5.2 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], $\boldsymbol{\Psi} \in \{-1, 0, 1\}^{M\times N}$. 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. (E.1) after which the problem is converted to a frequency estimation problem that is solved using the MUSIC algorithm, as explained in Eqn. (E.5),
- **PaIBOMP** - BOMP with parabolic interpolation,
- **CCBP** - The CCBP algorithm in Eqn. (E.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. E.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.

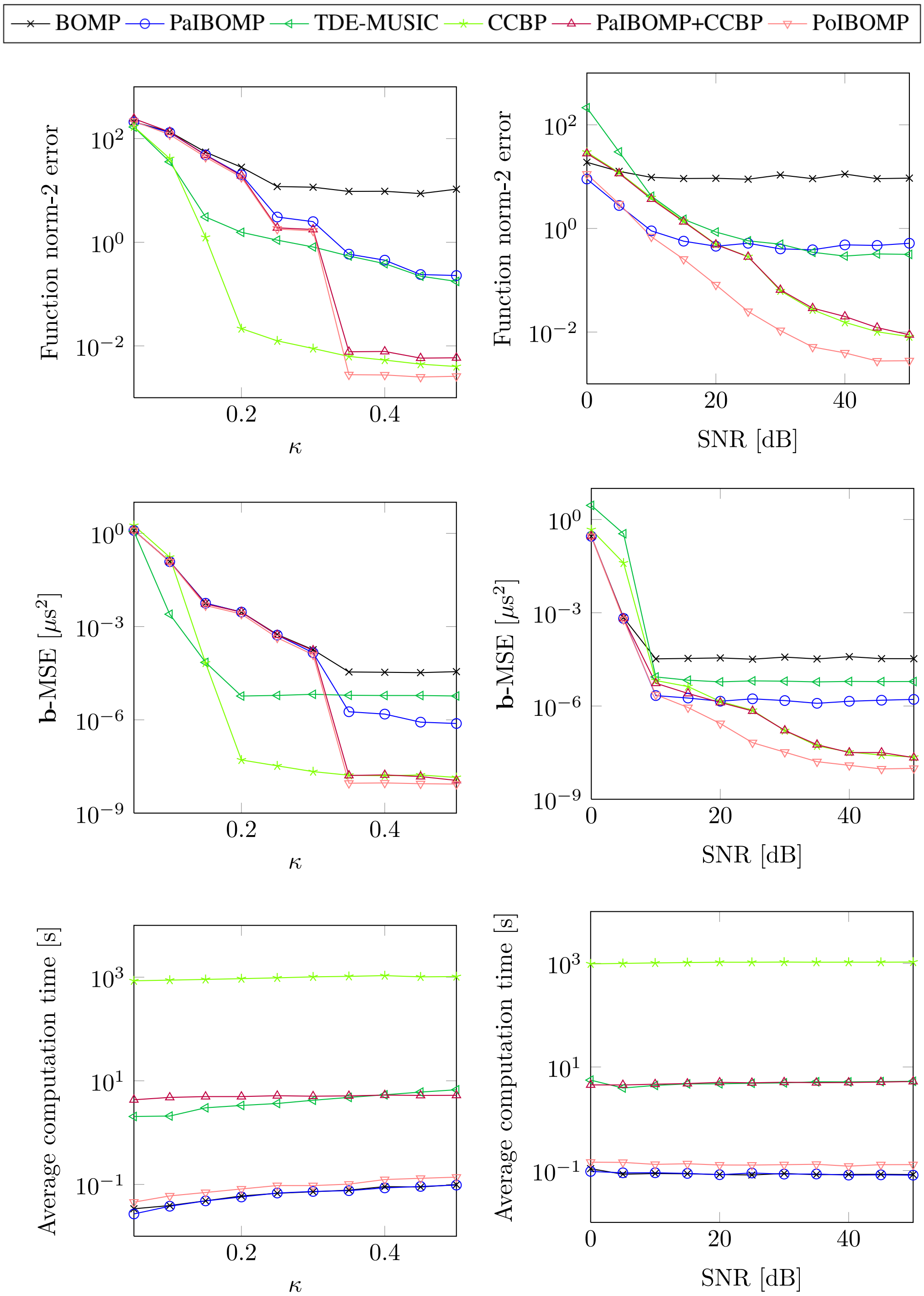


Fig. E.3: The estimator precision with non-overlapping pulses for the TDE problem. The left figures are noise-less experiments for varying choices of subsampling ratios, κ, while the right figures are for $\kappa = 0.4$ and with varying SNR levels. The top figures is signal reconstruction quality, the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

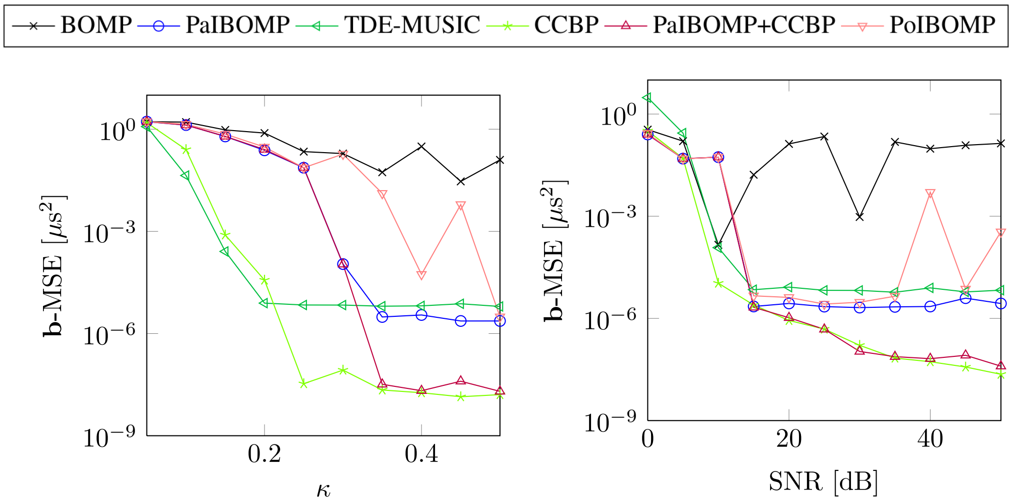


Fig. E.4: The estimator precision with overlapping pulses for the TDE problem. The left side is the noise-less case with varying subsampling ratios and the right side is with $\kappa = 0.4$ and varying SNR levels.

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. E.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

[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.

as CCBP than in Fig. E.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. E.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. (E.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. E.5. As can be seen in the top two figures all the estimators' signal reconstruction are affected by noise folding when CS is used. 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.

E.6 Conclusion

With our numerical experiments, we show that the proposed CCBP and, with high enough sampling frequency, PaIBOMP+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

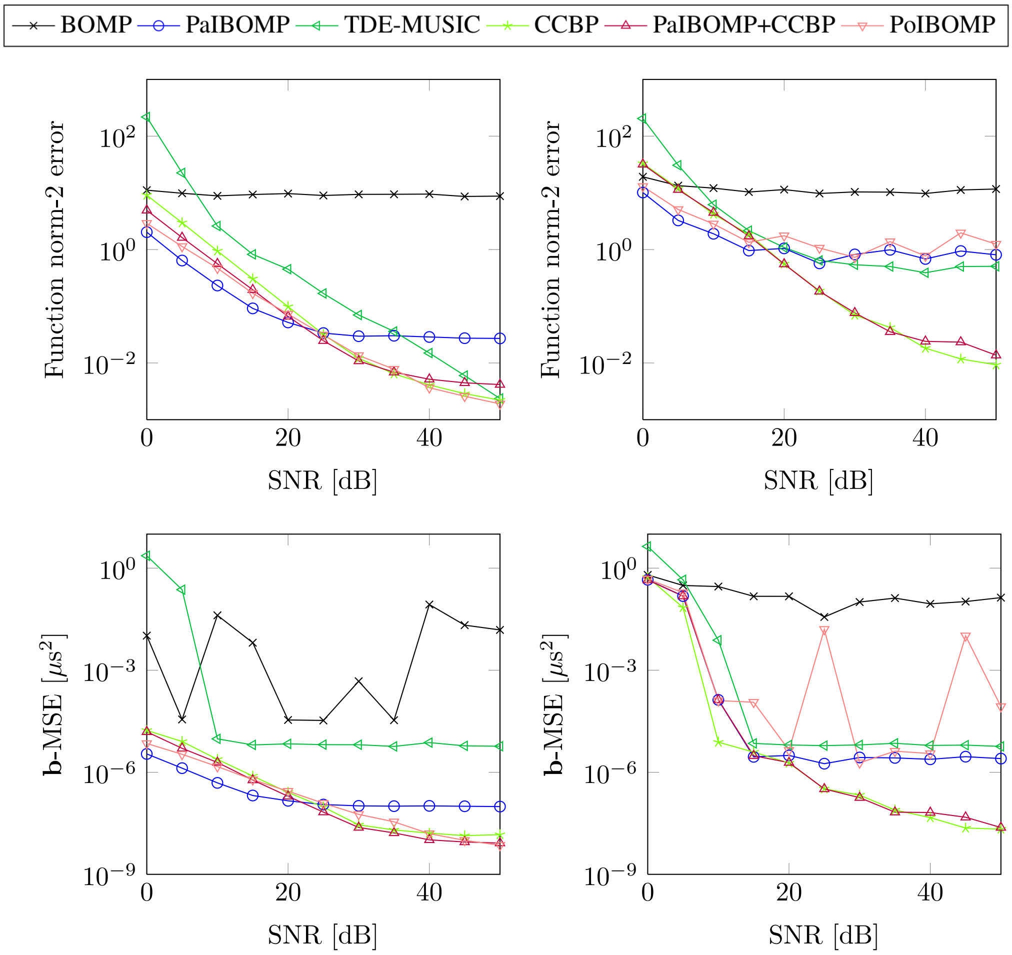


Fig. E.5: The function error and estimator precision with overlapping pulses and noise folding for the TDE problem. The left side is for $\kappa = 1$ and the right side is for $\kappa = 0.4$.

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. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[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. Chen, D. Donoho, and M. Saunders, “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, M. A. Davenport, J. N. Laska, and R. G. Baranuik, “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.