EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

REVIEW Open Access

A unified approach to sparse signal processing

Farokh Marvasti[1*], Arash Amini[1], Farzan Haddadi[2], Mahdi Soltanolkotabi[1], Babak Hossein Khalaj[1], Akram Aldroubi[3], Saeid Sanei[4] and Janathon Chambers[5]

Abstract

A unified view of the area of sparse signal processing is presented in tutorial form by bringing together various fields in which the property of sparsity has been successfully exploited. For each of these fields, various algorithms and techniques, which have been developed to leverage sparsity, are described succinctly. The common potential benefits of significant reduction in sampling rate and processing manipulations through sparse signal processing are revealed. The key application domains of sparse signal processing are sampling, coding, spectral estimation, array processing, component analysis, and multipath channel estimation. In terms of the sampling process and reconstruction algorithms, linkages are made with random sampling, compressed sensing, and rate of innovation. The redundancy introduced by channel coding in finite and real Galois fields is then related to over-sampling with similar reconstruction algorithms. The error locator polynomial (ELP) and iterative methods are shown to work quite effectively for both sampling and coding applications. The methods of Prony, Pisarenko, and MUltiple SIgnal Classification (MUSIC) are next shown to be targeted at analyzing signals with sparse frequency domain representations. Specifically, the relations of the approach of Prony to an annihilating filter in rate of innovation and ELP in coding are emphasized; the Pisarenko and MUSIC methods are further improvements of the Prony method under noisy environments. The iterative methods developed for sampling and coding applications are shown to be powerful tools in spectral estimation. Such narrowband spectral estimation is then related to multi-source location and direction of arrival estimation in array processing. Sparsity in unobservable source signals is also shown to facilitate source separation in sparse component analysis; the algorithms developed in this area such as linear programming and matching pursuit are also widely used in compressed sensing. Finally, the multipath channel estimation problem is shown to have a sparse formulation; algorithms similar to sampling and coding are used to estimate typical multicarrier communication channels.

1 Introduction

There are many applications in signal processing and communication systems where the discrete signals are sparse in some domain such as time, frequency, or space; i.e., most of the samples are zero, or alternatively, their transforms in another domain (normally called "*frequency coefficients*") are sparse (see Figures 1 and 2). There are trivial sparse transformations where the sparsity is preserved in both the "time" and "frequency" domains; the identity transform matrix and its permutations are extreme examples. Wavelet transformations that preserve the local characteristics of a sparse signal can be regarded as "almost" sparse in the "frequency" domain; in general, for sparse signals, the more similar the transformation matrix is to an identity matrix, the sparser the signal is in the transform domain. In any of these scenarios, sampling and processing can be optimized using sparse signal processing. In other words, the sampling rate and the processing manipulations can be significantly reduced; hence, a combination of data compression and processing time reduction can be achieved.[a]

Each field has developed its own tools, algorithms, and reconstruction methods for sparse signal processing. Very few authors have noticed the similarities of these fields. It is the intention of this tutorial to describe these methods in each field succinctly and show that these methods can be used in other areas and applications often with appreciable improvements. Among these fields are 1—*Sampling*: random sampling of bandlimited signals [1], compressed sensing (CS) [2], and sampling with finite rate of innovation [3]; 2—*Coding*: Galois [4,5] and real-field error correction

*Correspondence: marvasti@sharif.edu
[1]Electrical Engineering Department, Advanced Communication Research Institute (ACRI), Sharif University of Technology, Tehran, Iran
Full list of author information is available at the end of the article

Springer

© 2012 Marvasti et al.; licensee Springer. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

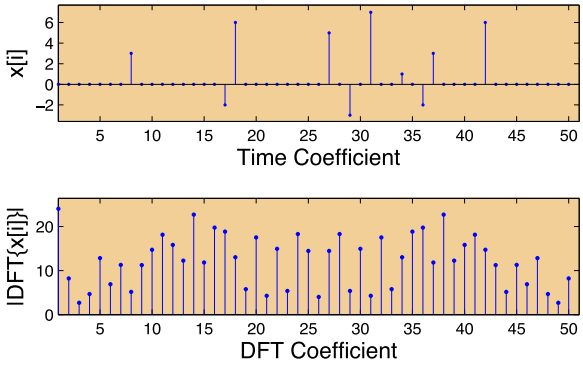

Figure 1 Sparse discrete time signal with its DFT.

codes [6]; 3—*Spectral Estimation* [7-10]; 4—*Array Processing*: Multi-source location (MSL) and direction of arrival (DOA) estimation [11,12], sparse array processing [13], and sensor networks [14]; 5—*Sparse Component Analysis* (SCA): blind source separation [15-17] and dictionary representation [18-20]; 6—*Channel Estimation* in Orthogonal Frequency Division Multiplexing (OFDM) [21-23]. The sparsity properties of these fields are summarized in Tables 1, 2, and 3.[b] The details of most of the major applications will be discussed in the next sections but the common traits will be discussed in this introduction.

The columns of Table 1 consist of 0—category, 1—topics, 2—sparsity domain, 3—type of sparsity, 4—information domain, 5—type of sampling in information domain, 6—minimum sampling rate, 7—conventional reconstruction methods, and 8—applications. The first rows (2–7) of column 1 are on sampling techniques. The 8–9th rows are related to channel coding, row 10 is on spectral estimation and rows 11–13 are related to array processing. Rows 14–15 correspond to SCA and finally, row 16 covers multicarrier channel estimation, which is a rather new topic. As shown in column 2 of the table, depending on the topics, sparsity is defined in the time, space, or "frequency" domains. In some applications, the

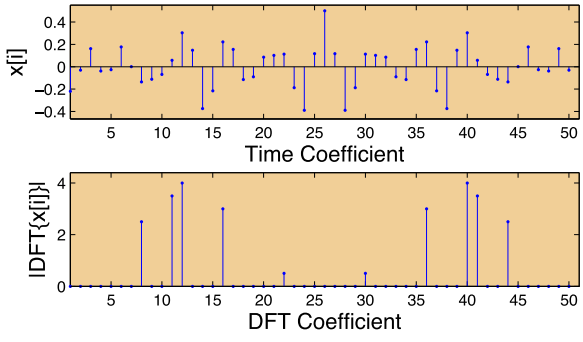

Figure 2 Sparsity is manifested in the frequency domain.

sparsity is defined as the number of polynomial coefficients (which in a way could be regarded as "frequency"), the number of sources (which may depend on location or time sparsity for the signal sources), or the number of "words" (signal bases) in a dictionary. The type of sparsity is shown in column 3; for sampling schemes, it is usually low-pass, band-pass, or multiband [24], while for compressed sensing, and most other applications, it is random. Column 4 represents the information domain, where the order of sparsity, locations, and amplitudes can be determined by proper sampling (column 5) in this domain. Column 7 is on traditional reconstruction methods; however, for each area, any of the reconstruction methods can be used. The other columns are self explanatory and will be discussed in more details in the following sections.

The rows 2–4 of Table 1 are related to the sampling (uniform or random) of signals that are bandlimited in the Fourier domain. Band-limitedness is a special case of sparsity where the nonzero coefficients in the frequency domain are consecutive. A better assumption in the frequency domain is to have random sparsity [25-27] as shown in row 5 and column 3. A generalization of the sparsity in the frequency domain is sparsity in any transform domain such as Discrete Cosine and Wavelet Transforms (DCT and DWT); this concept is further generalized in CS (row 6) where sampling is taken by a linear combination of time domain samples [2,28-30]. Sampling of signals with finite rate of innovation (row 7) is related to piecewise smooth (polynomial based) signals. The positions of discontinuous points are determined by annihilating filters that are equivalent to error locator polynomials in error correction codes and the Prony's method [10] as discussed in Sections 4 and 5, respectively.

Random errors in a Galois field (row 8) and the additive impulsive noise in real-field error correction codes (row 9) are sparse disturbances that need to be detected and removed. For erasure channels, the impulsive noise can be regarded as the negative of the missing sample value [31]; thus the missing sampling problem, which can also be regarded as a special case of nonuniform sampling, is also a special case of the error correction problem. A subclass of impulsive noise for 2-D signals is salt and pepper noise [32]. The information domain, where the sampling process occurs, is called the syndrome which is usually in a transform domain. Spectral estimation (row 10) is the dual of error correction codes, i.e., the sparsity is in the frequency domain. MSL (row 11) and multi-target detection in radars are similar to spectral estimation since targets act as spatial sparse mono-tones; each target is mapped to a specific spatial frequency regarding its line of sight direction relative to the receiver. The techniques developed for this branch of science is unique; with examples such as MUSIC [7], Prony [8], and Pisarenko [9]. We shall see that the techniques used in real-field error correction

Table 1 Various topics and applications with sparsity properties: the sparsity, which may be in the time/space or "frequency" domains, consists of unknown samples/coefficients that need to be determined

1	0 Category	1 Topics	2 Sparsity domain	3 Type of sparsity	4 Information domain	5 Type of sampling in info. domain	6 Min number of required samples	7 Conventional reconstruction methods	8 Applications
2	Sampling	Uniform sampling	Frequency	Lowpass	Time/space	Uniform	$2 \times BW - 1$	Lowpass filtering/ Interpolation	A/D
3		Nonuniform sampling	Frequency	Lowpass	Time/space	Missing samp--les/jitter/per--iodic/random	$2 \times BW - 1$ (in some cases even BW)	Iterative metho--ds/filter banks/ spline interp.	Seismic/ MRI/CT/ FM/ PPM
4		Sampling of multiband signals	Frequency	Union of disjoint intervals	Time/pace	Uniform/jit--ter/periodic/ random	$2 \times \sum BW$	Iterative metho--ds/filter banks/ interpolation	Data compression/ radar
5		Random sampling	Frequency	Random	Time/space	Random/ uniform	$2 \times \sum$ # coeff.	Iterative methods: adapt. thresh. RDE/ELP	Missing samp. recovery/ data comp.
6		Compressed sensing	An arbitrary orthonormal transform	Random	Random mapping of time/space	Random mixtures of samples	$c \cdot k \cdot \log(\frac{n}{k})$	Basis pursuit/ matching pursuit	Data compression
7		Finite rate of innovation	Time and polynomial coeff.	Random	Filtered time domain	Uniform	# Coeff. + 1 + 2 · (# discont. epochs)	Annihilating filter (ELP)	ECG/ OCT/ UWB
8	Channel coding	Galois field codes	Time	Random	Syndrome	Uniform or random	2 × # errors	Berlekamp -Massey/Viterbi/ belief prop.	Digital communic--tion
9		Real field codes	Time	Random	Transform domain	Uniform or random	2 × # impulsive noise	Adaptive thresholding RDE/ELP	Fault tolerant system
10	Spectral estimation	Spectral estimation	Frequency	Random	Time/ autocor--relation	Uniform	2 × # tones −1	MUSIC/ pisarenko/ prony/MDL	Military/ radars

Table 1 Various topics and applications with sparsity properties: the sparsity, which may be in the time/space or "frequency" domains, consists of unknown samples/coefficients that need to be determined *(Continued)*

11	Array processing	MSL/ DOA estimation	Space	Random	Space/ autocor--relation	Uniform	2× # sources	MDL+ MUSIC/ ESPRIT	Radars/ sonar/ ultrasound
12		Sparse arr--ay beam--forming	Space	Random/ missing elements	Space	Peaks of sidelobes/ [non]uniform	2 × # desired array elements	Optimiz--ation: LP/ SA/GA	Radars/sonar/ ultrasound/ MSL
13		Sensor networks	Space	Random	Space	Uniform	2× BW of random field	Similar to row 5	Seismic/ meteorology/ environmental
14	SCA	BSS	Active source/time	Random	Time	Uniform	2 × # active sources	ℓ_1/ℓ_2/ SL0	Biomedical
15		SDR	Dictionary	Uniform/ random	Linear mix--ture of time samples	Random	2×# sparse words	ℓ_1/ℓ_2/ SL0	Data compression
16	Channel estimation	Multipath channels	Time	Random	Frequency or time	Uniform/ nonuniform	2 × # Spa--rse channel components	ℓ_1/ MIMAT	Channel equaliz--ation/OFDM

The information domain consists of known samples/coefficients in the "frequency" or time/space domain (the complement of the sparse domain). A list of acronyms is given in Table 2 at the end of this section; also, a list of common notations is presented in Table 3. For definition of ESPRIT on row 11 and column 7, see the footnote on page 41.

Table 2 List of acronyms

ADSL	Asynchronous Digital Subscriber Line	AIC	Akaike Information Criterion
ARMA	Auto-Regressive Moving Average	AR	Auto-Regressive
BW	BandWidth	BSS	Blind Source Separation
CFAR	Constant False Alarm Rate	CAD	Computer Aided Design
CS	Compressed Sensing	CG	Conjugate Gradient
DAB	Digital Audio Broadcasting	CT	Computer Tomography
DCT	Discrete Cosine Transform	DC	Direct Current: Zero-Frequency Coefficient
DFT	Discrete Fourier Transform	DHT	Discrete Hartley Transform
DOA	Direction Of Arrival	DST	Discrete Sine Transform
DT	Discrete Transform	DVB	Digital Video Broadcasting
DWT	Discrete Wavelet Transform	EEG	ElectroEncephaloGraphy
ELP	Error Locator Polynomial	ESPRIT	Estimation of Signal Parameters via
FDTD	Finite-Difference Time-Domain		Rotational Invariance Techniques
FETD	Finite-Element Time-Domain	FOCUSS	FOCal Under-determined System Solver
FPE	Final Prediction Error	GPSR	Gradient Projection Sparse Reconstruction
GA	Genetic Algorithm	ICA	Independent Component Analysis
HNQ	Hannan and Quinn method	IDT	Inverse Discrete Transform
IDE	Iterative Detection and Estimation	ISTA	Iterative Shrinkage-Threshold Algorithm
IMAT	Iterative Methods with Adaptive Thresholding	KLT	Karhunen Loeve Transform
ℓ_1	Absolute Summable Discrete Signals	ℓ_2	Finite Energy Discrete Signals
LDPC	Low Density Parity Check	LP	Linear Programming
MA	Moving Average	MAP	Maximum A Posteriori probability
MDL	Minimum Description Length	ML	Maximum Likelihood
MIMAT	Modified IMAT	MSL	Multi-Source Location
MMSE	Minimum Mean Squared Error	NP	Non-Polynomial time
MUSIC	MUltiple SIgnal Classification	OFDM	Orthogonal Frequency Division Multiplex
OCT	Optical Coherence Tomography	OMP	Orthogonal Matching Pursuit
OFDMA	Orthogonal Frequency Division Multiple Access	PCA	Principle Component Analysis
OSR	Over Sampling Ratio	PHD	Pisarenko Harmonic Decomposition
PDF	Probability Density Function	PPM	Pulse-Position Modulation
POCS	Projection Onto Convex Sets	RIP	Restricted Isometry Property
RDE	Recursive Detection and Estimation	RV	Residual Variance
RS	Reed-Solomon	SCA	Sparse Component Analysis
SA	Simulated Annealing	SDFT	Sorted DFT
SDCT	Sorted DCT	SER	Symbol Error Rate
SDR	Sparse Dictionary Representation	SL0	Smoothed ℓ_0-norm
SI	Shift Invariant	ULA	Uniform Linear Array
SNR	Signal-to-Noise Ratio	WIMAX	Worldwide Inter-operability for Microwave Access
UWB	Ultra Wide Band	WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network		

codes such as iterative methods (IMAT) can also be used in this area.

The array processing category (rows 11–13) consists of three separate topics. The first one covers MSL in radars, sonars, and DOA. The techniques developed for this field are similar to the spectral estimation methods with emphasis on the minimum description length (MDL) [33]. The second topic in the array processing category is related to the design of sparse arrays where some of the array elements are missing; the remaining nodes form a nonuniform sparse grid. In this case, one of the optimization problems is to find the sparsest array (number, locations, and weights of elements) for a given beampattern. This problem has some resemblance to the missing sampling problem but will not be discussed in this article. The third topic is on sensor networks (row 13). Distributed sampling and recovery of a physical field using an array of sparse sensors is a problem of increasing interest in environmental and seismic monitoring applications of sensor networks [34]. Sensor fields may be bandlimited or non-bandlimited. Since the power consumption is the most restricting issue in sensors, it is vital to use the lowest possible number of sensors (sparse sensor networks) with the minimum processing computation; this topic also will not be discussed in this article.

In SCA, the number of observations is much less than the number of sources (signals). However, if the sources are sparse in the time domain, then the active sources and their amplitudes can be determined; this is equivalent to error correction codes. Sparse dictionary representation (SDR) is another new area where signals are represented by the sparsest number of words (signal bases) in a dictionary of finite number of words; this sparsity may result in a tremendous amount of data compression. When the dictionary is over complete, there are many ways to represent the signal; however, we are interested in the sparsest representation. Normally, for extraction of statistically independent sources, independent component analysis (ICA) is used for a complete set of linear mixtures. In the case of a non-complete (underdetermined) set of linear mixtures, ICA can work if the sources are also sparse; for this special case, ICA analysis is synonymous with SCA.

Finally, channel estimation is shown in row 16. In mobile communication systems, multipath reflections create a channel that can be modeled by a sparse FIR filter. For proper decoding of the incoming data, the channel characteristics should be estimated before they can be equalized. For this purpose, a training sequence is inserted within the main data, which enables the receiver to obtain the output of the channel by exploiting this training sequence. The channel estimation problem becomes a deconvolution problem under noisy environments. The sparsity criterion of the channel greatly improves the channel estimation; this is where the algorithms for extraction of a sparse signal could be employed [21,22,35].

When sparsity is random, further signal processing is needed. In this case, there are three items that need to be considered. 1—Evaluating the number of sparse coefficients (or samples), 2—finding the positions of sparse coefficients, and 3—determining the values of these coefficients. In some applications, only the first two items are needed; e.g., in spectral estimation. However, in almost all the other cases mentioned in Table 1, all the three items should be determined. Various types of linear programming (LP) and some iterative algorithms, such as the IMAT with adaptive thresholding (IMAT), determine the number, positions, and values of sparse samples at the same time. On the other hand, the minimum description length (MDL) method, used in DOA/MSL and spectral estimation, determines the number of sparse source locations or frequencies. In the subsequent sections, we shall describe, in more detail, each algorithm for various areas and applications based on Table 1.

Finally, it should be mentioned that the signal model for each topic or application may be deterministic or stochastic. For example, in the sampling category for rows 2–4 and 7, the signal model is typically deterministic although stochastic models could also be envisioned [36]. On the other hand, for random sampling and CS (rows 5–6), the signal model is stochastic although deterministic models may also be envisioned [37]. In channel coding and estimation (rows 8–9 and 16), the signal model is normally deterministic. For Spectral and DOA estimation (rows 10–11), stochastic models are assumed, whereas for array beam-forming (row 12), deterministic models are used. In sensor networks (row 13), both deterministic and stochastic signal models are employed. Finally, in SCA (rows 14–15), statistical independence of sources may be necessary and thus stochastic models are applied.

Table 3 Common notations used throughout the article

n	Length of original vector
k	Order of sparsity
m	Length of observed vector
$\mathbf{x}$	Original vector
$\mathbf{s}$	Corresponding sparse vector
$\mathbf{y}$	Observed vector
$\boldsymbol{\nu}$	Noise vector
$\mathbf{A}$	Transformation matrix relating $\mathbf{s}$ to $\mathbf{y}$
$\|\mathbf{u}_{n\times 1}\|_{\ell_p}$	$\left(\sum_{i=1}^{n} \|u_i\|^p\right)^{\left(\frac{1}{p}\right)}$

2 Underdetermined system of linear equations

In most of the applications where sparsity constraint plays a significant role, we are dealing with under-determined system of linear equations; i.e., a sparse vector $\mathbf{s}_{n\times 1}$ is

observed through a linear mixing system denoted by $\mathbf{A}_{m\times n}$ where $m < n$:

$$\mathbf{x}_{m\times 1} = \mathbf{A}_{m\times n} \cdot \mathbf{s}_{n\times 1} \tag{1}$$

Since $m < n$, the vector $\mathbf{s}_{n\times 1}$ cannot be uniquely recovered by observing the measurement vector $\mathbf{x}_{m\times 1}$; however, among the infinite number of solutions to (1), the sparsest solution may be unique. For instance, if no $2k$ columns of $\mathbf{A}_{m\times n}$ are linearly dependent, the null-space of $\mathbf{A}_{m\times n}$ does not include any $2k$-sparse vector (at most $2k$ nonzero elements) and therefore, the measurement vectors ($\mathbf{x}_{m\times n}$) of different k-sparse vectors are different. Thus, if $\mathbf{s}_{n\times 1}$ is sparse enough (k-sparse), the sparsest solution of (1) is unique and coincides with $\mathbf{s}_{n\times 1}$; i.e., perfect recovery. Unfortunately, there are two obstacles here: (1) the vector $\mathbf{x}_{m\times 1}$ often includes an additive noise term, and (2) finding the sparsest solution of a linear system is an NP problem in general.

Since in the rest of the article, we are frequently dealing with the problem of reconstructing the sparsest solution of (1), we first review some of the important reconstruction methods in this section.

2.1 Greedy methods

Mallat and Zhang [38] have developed a general iterative method for approximating sparse decomposition. When the dictionary is orthogonal and the signal x is composed of $k \ll n$ atoms, the algorithm recovers the sparse decomposition exactly after n steps. The introduced method which is a greedy algorithm [39], is usually referred to as *Matching Pursuit*. Since the algorithm is myopic, in some certain cases, wrong atoms are chosen in the first few iterations, and thus the remaining iterations are spent on correcting the first few mistakes. The concepts of this method are the basis of other advanced greedy methods such as OMP [40] and CoSaMP [41]. The algorithms of these greedy methods (MP, OMP, and CoSaMP) are shown in Table 4.

2.2 Basis pursuit

The mathematical representation of counting the number of sparse components is denoted by ℓ_0. However, ℓ_0 is not a proper norm and is not computationally tractable. The closest convex norm to ℓ_0 is ℓ_1. The ℓ_1 optimization of an overcomplete dictionary is called Basis Pursuit. However the ℓ_1-norm is non-differentiable and we cannot use gradient methods for optimal solutions [42]. On the other hand, the ℓ_1 solution is stable due to its convexity (the global optimum is the same as the local one) [20].

Formally, the Basis Pursuit can be formulated as:

$$\min \|\mathbf{s}\|_{\ell_1} \quad \text{s.t.} \quad \mathbf{x} = \mathbf{A}\cdot\mathbf{s} \tag{2}$$

Table 4 Greedy algorithms

1. Let $\hat{\mathbf{s}} = \mathbf{0}_{n\times 1}$, $\mathbf{r}^{(0)} = \mathbf{x}$, $\boldsymbol{\mathcal{S}}^{(0)} = \emptyset$ and $i = 1$.
2. Evaluate $c_j = \langle \mathbf{r}^{(i-1)}, \mathbf{a}_j \rangle$ for $j = 1, \ldots, n$ where $\mathbf{a}_j$'s are the columns of the mixing matrix $\mathbf{A}$ (atoms) and sort c_j's as $|c_{j_1}| \geq \cdots \geq |c_{j_n}|$.
3. - MP: Set $\boldsymbol{\mathcal{S}}^{(i)} = \boldsymbol{\mathcal{S}}^{(i-1)} \cup j_1$.
   - OMP: Set $\boldsymbol{\mathcal{S}}^{(i)} = \boldsymbol{\mathcal{S}}^{(i-1)} \cup j_1$ and $\mathbf{A}_{m\times|\boldsymbol{\mathcal{S}}^{(i)}|} = \left[\mathbf{a}_j\right]_{j\in\boldsymbol{\mathcal{S}}^{(i)}}$.
   - CoSaMP: Set $\boldsymbol{\mathcal{S}}^{(i)} = \boldsymbol{\mathcal{S}}^{(i-1)} \cup \{j_1, \ldots, j_{2k}\}$ and $\mathbf{A}_{m\times|\boldsymbol{\mathcal{S}}^{(i)}|} = \left[\mathbf{a}_j\right]_{j\in\boldsymbol{\mathcal{S}}^{(i)}}$.
4. - MP: − − −
   - OMP & CoSaMP: Find $\tilde{\mathbf{s}}$ that $\mathbf{A}^{(i)} \cdot \tilde{\mathbf{s}} = \mathbf{x}$.
5. - MP & OMP: − − −
   - CoSaMP: Sort the values of $\tilde{\mathbf{s}}$ as $|\tilde{s}_{t_1}| \geq |\tilde{s}_{t_2}| \geq \ldots$ and redefine $j_1, \ldots, j_k$ as the indices of the columns in $\mathbf{A}$ that correspond to the columns $t_1, \ldots, t_k$ in $\mathbf{A}^{(i)}$. Also set $\boldsymbol{\mathcal{S}}^{(i)} = \{j_1, \ldots, j_k\}$.
6. - MP: Set $\hat{s}_{j_1} = c_{j_1}$.
   - OMP & CoSaMP: Set $\hat{s}_{j_l} = \tilde{s}_l$ for $l = 1, \ldots, k$ and $\hat{s}_l = 0$ where $l \notin \boldsymbol{\mathcal{S}}^{(i)}$.
7. Set $\mathbf{r}^{(i)} = \mathbf{x} - \mathbf{A}\cdot\hat{\mathbf{s}}$.
8. Stop if $\|\mathbf{r}^{(i)}\|_{\ell_2}$ is smaller than a desired threshold or when a maximum number of iterations is reached; otherwise, increase i and go to step 2.

We now explain how the Basis Pursuit is related to LP. The standard form of LP is a constrained optimization problem defined in terms of variable $\mathbf{x} \in \mathbb{R}^n$ by:

$$\min \mathbf{C}^T\mathbf{x} \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \qquad \forall i:\ x_i \geq 0 \tag{3}$$

where $\mathbf{C}^T\mathbf{x}$ is the objective function, $\mathbf{A}\mathbf{x} = \mathbf{b}$ is a set of equality constraints and $\forall i:\ x_i \geq 0$ is a set of bounds. Table 5 shows this relationship. Thus, the solution of (2) can be obtained by solving the equivalent LP. The Interior Point methods are the main approaches to solve LP.

2.3 Gradient projection sparse reconstruction (GPSR)

The GPSR technique [44] is considered as one of the fast variations of the ℓ_1-minimization method and consists of solving the following minimization problem:

$$\arg\min_{\mathbf{s}}\ J(\mathbf{s}) = \frac{1}{2}||\mathbf{x} - \mathbf{A}\mathbf{s}||_{\ell_2}^2 + \tau||\mathbf{s}||_{\ell_1} \tag{4}$$

Note that $J(\mathbf{s})$ is almost the Lagrange form of the constraint problem in (2) where the Lagrange multiplier is defined as $\frac{1}{2\tau}$, with the difference that in (4), the minimization procedure is performed exclusively on $\mathbf{s}$ and not on τ. Thus, the outcome of (4) coincides with that of (2)

Table 5 Relation between LP and basis pursuit (the notation for LP is from [43])

Basis pursuit	Linear programming
m	$2p$
$\mathbf{s}$	$\mathbf{x}$
$(1, \ldots, 1)_{1\times m}$	$\mathbf{C}$
$\pm\mathbf{A}$	$\mathbf{A}$
$\mathbf{x}$	$\mathbf{b}$

only when the proper τ is used. For a fast implementation of (4), the positive and negative elements of $\mathbf{s}$ are treated separately, i.e.,

$$\mathbf{s} = \mathbf{u} - \mathbf{v}, \quad \forall j : \; u_j \, , \, v_j \; \geq 0$$

Now by assuming that all the vectors and matrices are real, it is easy to check that the minimizer of the following cost function (F) corresponds to the minimizer of $J(\mathbf{s})$:

$$F(\mathbf{z}) = \mathbf{c}^T\mathbf{z} + \frac{1}{2}\mathbf{z}^T\mathbf{B}\mathbf{z} \quad \text{s.t.} \quad \mathbf{z} \geq 0 \tag{5}$$

where

$$\mathbf{z} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \qquad \mathbf{c} = \tau \mathbf{1}_{2n\times 1} + \begin{bmatrix} -\mathbf{A}^T\mathbf{x} \\ \mathbf{A}^T\mathbf{x} \end{bmatrix}$$
$$\mathbf{B} = \begin{bmatrix} \mathbf{A}^T\mathbf{A} & -\mathbf{A}^T\mathbf{A} \\ -\mathbf{A}^T\mathbf{A} & \mathbf{A}^T\mathbf{A} \end{bmatrix} \tag{6}$$

In GPSR, the latter cost function is iteratively minimized by moving in the opposite direction of the gradient while respecting the condition $\mathbf{z} \geq 0$. There step-wise explanation of the basic GPSR method is given in Table 6. In this table, $(a)_+$ denotes the value $\max\{a, 0\}$ while $(\mathbf{a})_+$ indicates the element-wise action of the same function on the vector $\mathbf{a}$. There is another adaptation of this method known as *Barzilai-Borwein* (BB) GPSR which is not discussed here.

2.4 Iterative shrinkage-threshold algorithm (ISTA)

Instead of using the gradient method for solving (4), it is possible to approximate the cost function. To explain this idea, let $\mathbf{s}^{(0)}$ be an estimate of the minimizer of (4) and let $\tilde{J}(\mathbf{s})$ be a cost function that satisfies:

$$\forall \, \mathbf{s} : \; \tilde{J}(\mathbf{s}) \geq J(\mathbf{s}) \quad \& \quad \tilde{J}(\mathbf{s}^{(0)}) = J(\mathbf{s}^{(0)}) \tag{7}$$

Now if $\mathbf{s}^{(1)}$ is the minimizer of $\tilde{J}(.)$, we should have $J(\mathbf{s}^{(1)}) \leq J(\mathbf{s}^{(1)})$; i.e., $\mathbf{s}^{(1)}$ better estimates the minimizer of $J(.)$ than $\mathbf{s}^{(0)}$. This technique is useful only when finding the minimizer of $\tilde{J}(.)$ is easier than solving the original problem. In ISTA [45], at the kth iteration and by having the estimate $\mathbf{s}^{(i)}$, the following alternative cost function is used:

$$\tilde{J}_i(\mathbf{s}) = J(\mathbf{s}) + \frac{\beta}{2}\|\mathbf{s} - \mathbf{s}^{(i)}\|_{\ell_2}^2 - \frac{1}{2}\|\mathbf{A}(\mathbf{s} - \mathbf{s}^{(i)})\|_{\ell_2}^2 \tag{8}$$

where β is a scalar larger than all squared singular values of $\mathbf{A}$ to ensure (7). By modifying the constant terms and rewriting the above cost function, one can check that the minimizer of $\tilde{J}_i(.)$ is essentially the same as

$$\arg\min_{\mathbf{s}} \frac{\beta}{2}\|\mathbf{s} - \mathbf{z}^{(i)}\|_{\ell_2}^2 + \tau\|\mathbf{s}\|_{\ell_1} \tag{9}$$

where

$$\mathbf{z}^{(i)} = \mathbf{s}^{(i)} + \frac{1}{\beta}\mathbf{A}^H\left(\mathbf{x} - \mathbf{A}\mathbf{s}^{(i)}\right) \tag{10}$$

Note that the minimization problem in (9) is separable with respect to the elements of $\mathbf{s}$ and we just need to find the minimizer of the single-variable cost function $\frac{\beta}{2}(s - z)^2 + \tau|s|$, which is the well-known shrinkage-threshold operator:

$$\mathcal{S}_{[\beta,\tau]}(z) = \begin{cases} z - \dfrac{\tau}{\beta} & z > \dfrac{\tau}{\beta} \\ 0 & |z| \leq \dfrac{\tau}{\beta} \\ z + \dfrac{\tau}{\beta} & z < -\dfrac{\tau}{\beta} \end{cases} \tag{11}$$

The steps of the ISTA algorithm are explained in Table 7.

2.5 FOCal underdetermined system solver (FOCUSS)

FOCal underdetermined system solver is a non-parametric algorithm that consists of two parts [46]. It starts by finding a low resolution estimation of the sparse signal, and then pruning this solution to a sparser signal representation through several iterations. The solution at each iteration step is found by taking the pseudo-inverse of a modified weighted matrix. The pseudo-inverse of the modified weighted matrix is defined by $(\mathbf{AW})^+ = (\mathbf{AW})^H(\mathbf{AW} \cdot (\mathbf{AW})^H)^{-1}$. This iterative

Table 6 Basic GPSR algorithm

1. Initialize $\beta \in (0, 1)$, $\mu \in (0, \frac{1}{2})$, α_0 and $\mathbf{z}^{(0)}$. Also set $i = 0$.
2. Choose $\alpha^{(i)}$ to be the largest number of the form $\alpha_0\beta^j, j \geq 0$, such that
$$F\left(\left(\mathbf{z}^{(i)} - \alpha^{(i)}\nabla F(\mathbf{z}^{(i)})\right)_+\right) \leq F(\mathbf{z}^{(i)}) -$$
$$\mu\nabla F(\mathbf{z}^{(i)})^T\left(\mathbf{z}^{(i)} - \left(\mathbf{z}^{(i)} - \alpha^{(i)}\nabla F(\mathbf{z}^{(i)})\right)_+\right)$$
3. Set $\mathbf{z}^{(i+1)} = \left(\mathbf{z}^{(i)} - \alpha^{(i)}\nabla F(\mathbf{z}^{(i)})\right)_+$.
4. Check the termination criterion. If neither the maximum number of iterations has passed nor a given stopping condition is fulfilled, increase i and return to the 2nd step.

Table 7 ISTA algorithm

1. Choose the scalar β larger than all the singular values of $\mathbf{A}$ and set $i = 0$. Also initialize $\mathbf{s}^{(0)}$, e.g, $\mathbf{s}^{(0)} = \mathbf{A}^+\mathbf{x}$.
2. Set $\mathbf{z}^{(i)} = \mathbf{s}^{(i)} + \frac{1}{\beta}\mathbf{A}^H(\mathbf{x} - \mathbf{A}\mathbf{s}^{(i)})$.
3. Apply the shrinkage-threshold operator defined in (11):
$$s_j^{(i+1)} = \boldsymbol{\mathcal{S}}_{[\beta,\tau]}(z_j^{(i)}), \quad 1 \leq j \leq n$$
4. Check the termination criterion. If neither the maximum number of iterations has passed nor a given stopping condition is fulfilled, increase i and return to the 2nd step.

Table 8 FOCUSS (Basic)

- Step 1: $\mathbf{W}_{p_i} = diag(\mathbf{s}_{i-1})$
- Step 2: $\mathbf{q}_i = \left(\mathbf{A}\mathbf{W}_{p_i}\right)^{+}\mathbf{x}$
- Step 3: $\mathbf{s}_i = \mathbf{W}_{p_i} \cdot \mathbf{q}_i$

algorithm is the solution of the following optimization problem:

$$\text{Find } \mathbf{s} = \mathbf{W}\mathbf{q}, \text{ where: } \min \|\mathbf{q}\|_{\ell_2} \text{ s.t. } \mathbf{x} = \mathbf{A}\mathbf{W}\mathbf{q} \tag{12}$$

Description of this algorithm is given in Table 8 and an extended version is discussed in [46].

2.6 Iterative detection and estimation (IDE)

The idea behind this method is based on a geometrical interpretation of the sparsity. Consider the elements of vector $\mathbf{s}$ are i.i.d. random variables. By plotting a sample distribution of vector $\mathbf{s}$, which is obtained by plotting a large number of samples in the S-space, it is observed that the points tend to concentrate first around the origin, then along the coordinate axes, and finally across the coordinate planes. The algorithm used in IDE is given in Table 9. In this table, $\mathbf{s}_i$s are the inactive sources, $\mathbf{s}_a$s are the active sources, $\mathbf{A}_i$ is the column of $\mathbf{A}$ corresponding to the inactive $\mathbf{s}_i$ and $\mathbf{A}_a$ is the column of $\mathbf{A}$ corresponding to the active $\mathbf{s}_a$. Notice that IDE has some resemblances to the RDE method discussed in Section 4.1.2, IMAT mentioned in Section 4.1.2, and MIMAT explained in Section 8.1.2.

2.7 Smoothed ℓ_0-norm (SL0) method

As discussed earlier, the criterion for sparsity is the ℓ_0-norm; thus our minimization is

$$\min \|\mathbf{s}\|_{\ell_0} \quad \text{s.t.} \quad \mathbf{A} \cdot \mathbf{s} = \mathbf{x} \tag{13}$$

Table 9 IDE steps

- Detection Step: Find indices of inactive sources:

$$I^l = \left\{ 1 \le i \le m : \ \left| \mathbf{a}_i^T \cdot \mathbf{x} - \sum_{j \ne i}^{m} \hat{s}_j^l \mathbf{a}_i^T \cdot \mathbf{a}_j \right| < \epsilon^l \right\}$$

- Estimation Step: Find the following projection as the new estimate:

$$\mathbf{s}^{l+1} = \text{argmin}_{\mathbf{s}} \sum_{i \in I^l} s_i^2 \text{ s.t. } \mathbf{x}(t) = \mathbf{A} \cdot \mathbf{s}(t)$$

  The solution is derived from Karush-Kuhn-Tucker system of equations. At the $(l+1)^{th}$ iteration

$$\mathbf{s}_i = \mathbf{A}_i^T \cdot \mathbf{P}\left(\mathbf{x} - \mathbf{A}_a \cdot \mathbf{s}_a\right)$$

$$\mathbf{s}_a = \left(\mathbf{A}_a^T \mathbf{P} \mathbf{A}_a\right)^{-1} \mathbf{A}_a^T \mathbf{P} \cdot \mathbf{x}$$

  where the matrices and vectors are partitioned into inactive/active parts as $\mathbf{A}_i, \mathbf{A}_a, \mathbf{s}_i, \mathbf{s}_a$ and $\mathbf{P} = \left(\mathbf{A}_i \mathbf{A}_i^T\right)^{-1}$
- Stop after a fixed number of iterations.

The ℓ_0-norm has two major drawbacks: the need for a combinatorial search, and its sensitivity to noise. These problems arise from the fact that the ℓ_0-norm is discontinuous. The idea of SL0 is to approximate the ℓ_0-norm with functions of the type [47]:

$$f_\sigma(s) \triangleq e^{-\frac{s^2}{2\sigma^2}} \tag{14}$$

where σ is a parameter which determines the quality of the approximation. Note that we have

$$\lim_{\sigma \to 0} f_\sigma(s) = \begin{cases} 1 & \text{if } s = 0 \\ 0 & \text{if } s \ne 0 \end{cases} \tag{15}$$

For the vector $\mathbf{s}$, we have $\|\mathbf{s}\|_{\ell_0} \approx n - F_\sigma(s)$, where $F_\sigma(s) = \sum_{i=1}^{n} f_\sigma(s_i)$. Now minimizing $\|\mathbf{s}\|_{\ell_0}$ is equivalent to maximizing $F_\sigma(s)$ for some appropriate values of σ. For small values of σ, $F_\sigma(s)$ is highly non-smooth and contains many local maxima, and therefore its maximization over $\mathbf{A} \cdot \mathbf{s} = \mathbf{x}$ may not be global. On the other hand, for larger values of σ, $F_\sigma(s)$ is a smoother function and contains fewer local maxima, and its maximization may be possible (in fact there are no local maxima for large values of σ [47]). Hence we use a decreasing sequence for σ in the steepest ascent algorithm and may escape from getting trapped into local maxima and reach the actual maximum for small values of σ, which gives the minimum ℓ_0-norm solution. The algorithm is summarized in Table 10.

Table 10 SL0 steps

- Initialization:
  1. Set $\hat{\mathbf{s}}_0$ equal to the minimum ℓ_2-norm solution of $\mathbf{A}\mathbf{s} = \mathbf{x}$, obtained by pseudo-inverse of $\mathbf{A}$.
  2. Choose a suitable decreasing sequence for σ, $[\sigma_1, \ldots, \sigma_K]$.
- For $i = 1, \ldots, K$:
  1. Set $\sigma = \sigma_i$,
  2. Maximize the function F_σ on the feasible set $\mathcal{S} = \{\mathbf{s} | \mathbf{A}\mathbf{s} = \mathbf{x}\}$ using L iterations of the steepest ascent algorithm (followed by projection onto the feasible set):
     - Initialization: $\mathbf{s} = \hat{\mathbf{s}}_{i-1}$.
     - for $j = 1, \ldots, L$ (loop L times):
       (a) Let: $\Delta\mathbf{s} = [s_1 e^{-\frac{s_1^2}{2\sigma^2}}, \ldots, s_n e^{-\frac{s_n^2}{2\sigma^2}}]^T$.
       (b) Set $\mathbf{s} \leftarrow \mathbf{s} - \mu\Delta\mathbf{s}$ (where μ is a small positive constant).
       (c) Project $\mathbf{s}$ back onto the feasible set $\mathcal{S}$:

$$\mathbf{s} \leftarrow \mathbf{s} - \mathbf{A}^T \left(\mathbf{A}\mathbf{A}^T\right)^{-1} (\mathbf{A}\mathbf{s} - \mathbf{x})$$

  3. Set $\hat{\mathbf{s}}_i = \mathbf{s}$.
- Final answer is $\hat{\mathbf{s}} = \hat{\mathbf{s}}_K$

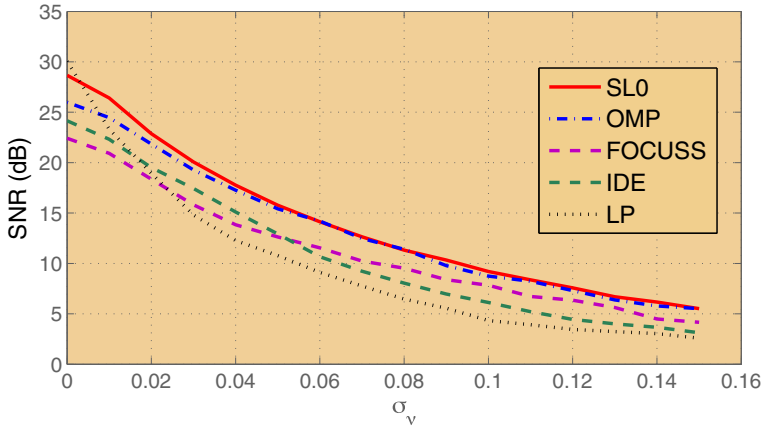

Figure 3 Performance of various methods with respect to the standard deviation when $n = 1,000$, $m = 400$, and $k = 100$.

2.8 Comparison of different techniques

The above techniques have been simulated and the results are depicted in Figure 3. In order to compare the efficiency and computational complexity of these methods, we use a fixed synthetic mixing matrix and source vectors. The elements of the mixing matrix are obtained from zero mean independent Gaussian random variables with variance $\sigma^2 = 1$. Sparse sources have been artificially generated using a Bernoulli–Gaussian model: $s_i = pN(0,\sigma_{on}) + (1-p)N(0,\sigma_{off})$. We set $\sigma_{off} = 0.01, \sigma_{on} = 1$ and $p = 0.1$. Then, we compute the noisy mixture vector $\mathbf{x}$ from $\mathbf{x} = \mathbf{As} + \boldsymbol{\nu}$, where $\boldsymbol{\nu}$ is the noise vector. The elements of the vector $\boldsymbol{\nu}$ are generated according to independent zero mean Gaussian random variables with variance σ_ν^2. We use orthogonal matching pursuit (OMP) which is a variant of Matching Pursuit [38]. OMP has a better performance in estimating the source vector in comparison to Matching Pursuit. Figure 4 demonstrates the time needed for each algorithm to estimate the vector $\mathbf{s}$ with respect to the number of sources. This figure shows that IDE and SL0 have the lowest complexity.

Figures 5 and 6 illustrate a comparison of several sparse reconstruction methods for sparse DFT signals and sparse random transformations, respectively. In all the simulations, the block size of the sparse signal is 512 while the number of sparse signal components in the frequency domain is 20. The compression rate is 25% which leads to a selection of 128 time domain observation samples.

In Figure 5, the greedy algorithms, COSAMP and OMP, demonstrate better performances than ISTA and GPSR, especially at lower input signal SNRs. IMAT shows a better performance than all other algorithms; however its performance in the higher input signal SNRs is almost similar to OMP and COSAMP. In Figure 6, OMP and COSAMP have better performances than the other ones while ISTA, SL0, and GPSR have more or less the same performances. In sparse DFT signals, the complexity of the IMAT algorithm is less than the others while ISTA is

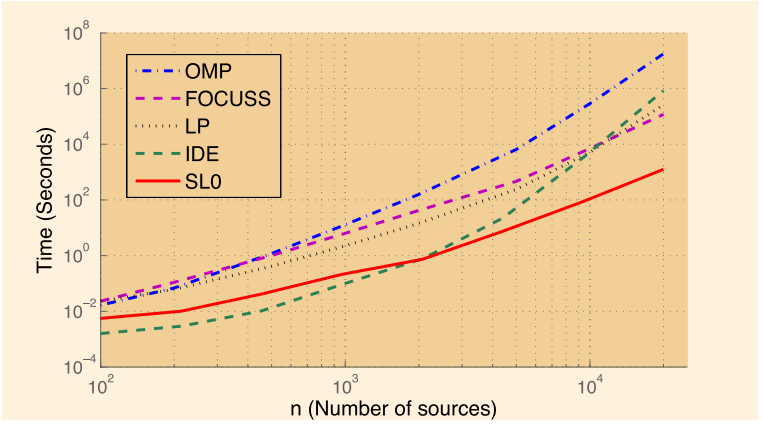

Figure 4 Computational time (complexity) versus the number of sources for $m = 0.4\ n$ and $k = 0.1\ n$.

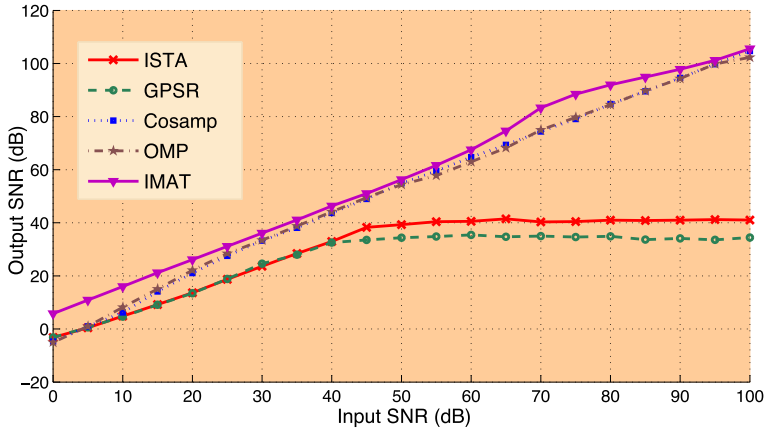

Figure 5 Performance comparison of some reconstruction techniques for DFT sparse signals.

the most complex algorithm. Similarly in Figure 6, SL0 has the least complexity.

3 Sampling: uniform, nonuniform, missing, random, compressed sensing, rate of innovation

Analog signals can be represented by finite rate discrete samples (uniform, nonuniform, or random) if the signal has some sort of redundancies such as band-limitedness, finite polynomial representation (e.g., periodic signals that are represented by a finite number of trigonometric polynomials), and nonlinear functions of such redundant functions [48,49]. The minimum sampling rate is the Nyquist rate for uniform sampling and its generalizations for nonuniform [1] and multiband signals [50]. When a signal is discrete, the equivalent discrete representation in the "frequency" domain (DFT, DCT, DWT, Discrete Hartley Transform (DHT), Discrete Sine Transform (DST)) may be sparse, which is the discrete version of bandlimited or multiband analog signals where the locations of the bands are unknown.

For discrete signals, if the nonzero coefficients ("frequency" sparsity) are consecutive, depending on the location of the zeros, they are called lowpass, bandpass, or multiband discrete signals; if the locations of the nonzero coefficients do not follow any of these patterns, the "frequency" sparsity is random. The number of discrete time samples needed to represent a frequency-sparse signal with known sparsity pattern follows the law of algebra, i.e., the number of time samples should be equal to the number of coefficients in the "frequency" domain; since the two domains are related by a full rank transform matrix, recovery from the time samples is equivalent to solving an invertible $k \times k$ system of linear equations where k is the number of sparse coefficients. For band-limited real signals, the Fourier transform (sparsity domain) consists of similar nonzero patterns in both negative and positive frequencies where only the positive part is counted

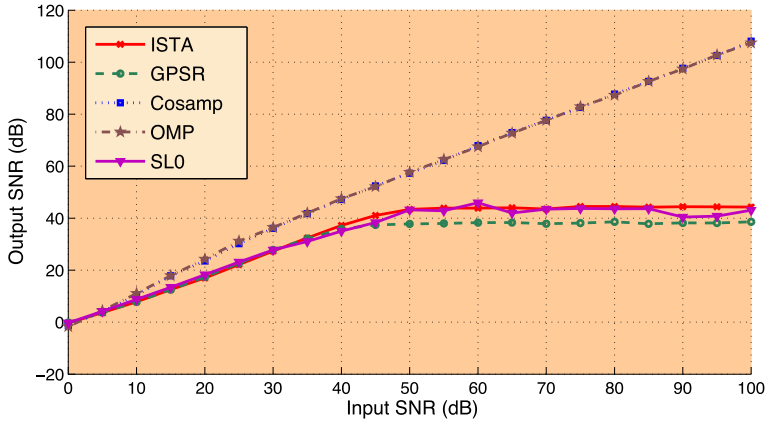

Figure 6 Performance comparison of some reconstruction techniques for sparse random trasnformations.

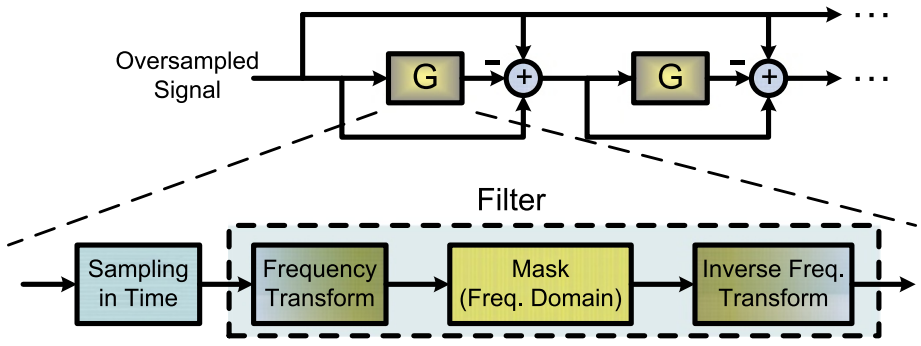

Figure 7 Block diagram of the iterative reconstruction method. The mask is an appropriate filter with coefficients of 1's and 0's depending on the type of sparsity in the original signal.

as the bandwidth; thus, the law of algebra is equivalent to the Nyquist rate, i.e., twice the bandwidth (for discrete signals with DC components it is twice the bandwidth minus one). The dual of frequency-sparsity is time-sparsity, which can happen in a burst or a random fashion. The number of "frequency" coefficients needed follows the Nyquist criterion. This will be further discussed in Section 4 for sparse additive impulsive noise channels.

3.1 Sampling of sparse signals

If the sparsity locations of a signal are known in a transform domain, then the number of samples needed in the time (space) domain should be at least equal to the number of sparse coefficients, i.e., the so-called Nyquist rate. However, depending on the type of sparsity (lowpass, bandpass, or random) and the type of sampling (uniform, periodic nonuniform, or random), the reconstruction may be unstable and the corresponding reconstruction matrix may be ill-conditioned [51,52]. Thus in many applications discussed in Table 1, the sampling rate in column 6 is higher than the minimum (Nyquist) rate.

When the location of sparsity is not known, by the law of algebra, the number of samples needed to specify the sparsity is at least twice the number of sparse coefficients. Again for stability reasons, the actual sampling rate is higher than this minimum figure [1,50]. To guarantee stability, instead of direct sampling of the signal, a combination of the samples can be used. Donoho has recently shown that if we take linear combinations of the samples, the minimum stable sampling rate is of the order $O(k\log(\frac{n}{k}))$, where n and k are the frame size and the sparsity order, respectively [29].

3.1.1 Reconstruction algorithms

There are many reconstruction algorithms that can be used depending on the sparsity pattern, uniform or random sampling, complexity issues, and sensitivity to quantization and additive noise [53,54]. Among these methods are LP, lagrange interpolation [55], time varying method [56], spline interpolation [57], matrix inversion [58], error locator polynomial (ELP) [59], iterative techniques [52,60-65], and IMAT [25,31,66,67]. In the following, we will only concentrate on the last three methods as well as the first (LP) that have been proven to be effective and practical.

Iterative methods when the location of sparsity is known The reconstruction algorithms have to recover the original sparse signal from the information domain and the type of sparsity in the transform domain. We know the samples in the information domain (both position and amplitude) and we know the location of sparsity in the transform domain. An iteration between these two domains (Figure 7 and Table 11) or consecutive Projections Onto Convex Sets (POCS) should yield the original signal [51,61,62,65,68-71].

In the case of the usual assumption that the sparsity is in the "frequency" domain and for the uniform sampling case of lowpass signals, one projection (bandlimiting in the frequency domain) suffices. However, if the frequency sparsity is random, the time samples are nonuniform, or the "frequency" domain is defined in a domain other than the DFT, then we need several iterations to have a good replica of the original signal. In general, this iterative method converges if the "Nyquist" rate is satisfied, i.e., the number of samples per block is greater than or equal to the number of coefficients. Figure 8 shows the improvement

Table 11 The iterative algorithm based on the block diagram of Figure 7

1. Take the transform (e.g. the Fourier transform) of the input to the i^{th} iteration ($\mathbf{x}^{(i)}$) and denote it as $X^{(i)}$; $\mathbf{x}^{(0)}$ is normally the initial received signal.
2. Multiply $X^{(i)}$ by a mask (for instance a band-limiting filter).
3. Take the inverse transform of the result in step 2 to get $\mathbf{r}^{(i)}$.
4. Set the new result as: $\mathbf{x}^{(i+1)} = \mathbf{x}^{(0)} + \mathbf{x}^{(i)} - \mathbf{r}^{(i)}$.
5. Repeat for a given number of iterations.
6. Stop when $\|\mathbf{x}^{(i+1)} - \mathbf{x}^{(i)}\|_{\ell_2} < \epsilon$.

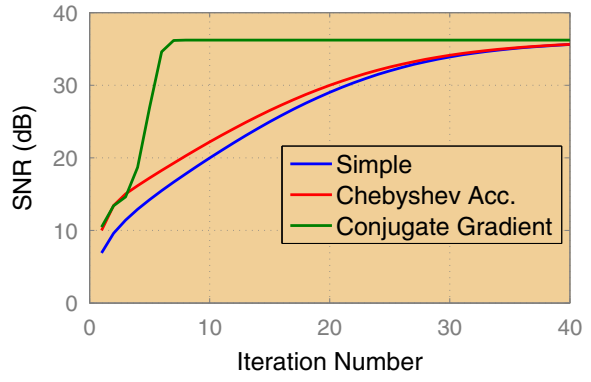

Figure 8 SNR improvement vs. the no. of iterations for a random sampling set at the Nyquist rate (OSR = 1) for a bandpass signal.

in dB versus the number of iterations for a random sampling set for a bandpass signal. In this figure, besides the standard iterative method, accelerated iterations such as Chebyshev and conjugate gradient methods are also used (please see [72] for the algorithms).

Iterative methods are quite robust against quantization and additive noise. In fact, we can prove that the iterative methods approach the pseudo-inverse (least squares) solution for a noisy environment; specially, when the matrix is ill-conditioned [50].

Iterative method with adaptive threshold (IMAT) for unknown location of sparsity As expected, when sparsity is assumed to be random, further signal processing is needed. We need to evaluate the number of sparse coefficients (or samples), the position of sparsity, and the values of the coefficients. The above iterative method cannot work since projection (the masking operation in Figure 7) onto the "frequency" domain is not possible without the knowledge of the positions of sparse coefficients. In this scenario, we need to use the knowledge of sparsity in some way. The introduction of an adaptive nonlinear threshold in the iterative method can do the trick and thus the name, IMAT; the block diagram and the pseudo-code are depicted in Figure 9 and Table 12, respectively. The algorithms in [23,25,31,73] are variations of this method. Figure 9 shows that by alternate projections between information and sparsity domains (adaptively lowering or raising the threshold levels in the sparsity domain), the sparse coefficients are gradually picked up after several iterations. This method can be considered as a modified version of Matching Pursuit as described in Section 2.1; the results are shown in Figure 10. The sampling rate in the time domain is twice the number of unknown sparse coefficients. This is called the full capacity rate; this figure shows that after approximately 15 iterations, the SNR reaches its peak value. In general, the higher the sampling rate relative to the full capacity, the faster is the convergence rate and the better the SNR value.

Matrix solutions When the sparse nonzero locations are known, matrix approaches can be utilized to determine the values of sparse coefficients [58]. Although these methods are rather straightforward, they may not be robust against quantization or additive noise when the matrices are ill conditioned.

There are other approaches such as Spline interpolation [57], nonlinear/time varying methods [58], Lagrange interpolation [55] and error locator polynomial (ELP) [74] that will not be discussed here. However, the ELP approach will be discussed in Section 4.1; variations of this method are called the annihilating filter in sampling with finite rate of innovation (Section 3.3) and Prony's method in spectral and DOA estimation (Section 5.1). These methods work quite well in the absence of additive noise but they may not be robust in the presence of noise. In the case of additive noise, the extensions

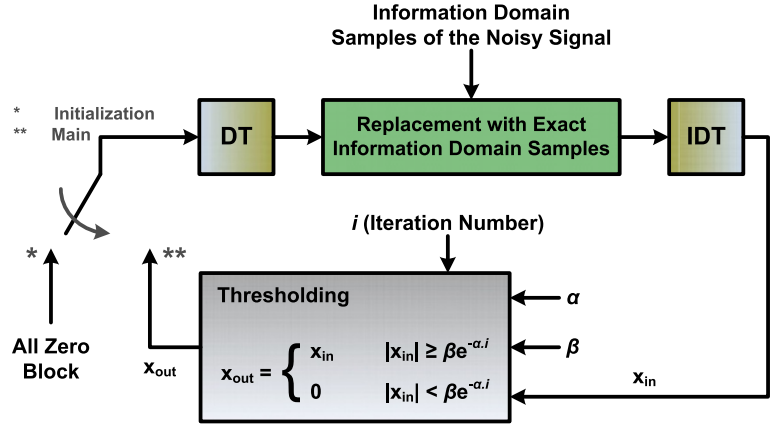

Figure 9 The IMAT for detecting the number, location, and values of sparsity.

Table 12 Generic IMAT of Figure 9 for any sparsity in the DT, which is typically DFT

1. Use the all-zero block as the initial value of the sparse domain signal (0^{th} iteration)
2. Convert the current estimate of the signal in the sparse domain into the information domain (for instance the time domain into the Fourier domain)
3. Where possible, replace the values with the known samples of the signal in the information domain.
4. Convert the signal back to the sparse domain.
5. Use adaptive hard thresholding to distinguish the original nonzero samples.
6. If neither the maximum number of iterations has past nor a given stopping condition is fulfilled, return to the 2nd step.

of the Prony method (ELP) such as Pisarenko harmonic decomposition (PHD), MUSIC and Estimation of signal parameters via rotational invariance techniques (ESPRIT) will be discussed in Sections 5.2, 5.3, and 6.

3.2 Compressed sensing (CS)

The relatively new topic of CS (Compressive) for sparse signals was originally introduced in [29,75] and further extended in [30,76,77]. The idea is to introduce sampling schemes with low number of required samples which uniquely represent the original sparse signal; these methods have lower computational complexities than the traditional techniques that employ oversampling and then apply compression. In other words, compression is achieved exactly at the time of sampling. Unlike the classical sampling theorem [78] based on the Fourier transform, the signals are assumed to be sparse in an arbitrary transform domain. Furthermore, there is no restricting assumption for the locations of nonzero coefficients in the sparsity domain; i.e., the locations should not follow a specific pattern such as lowpass or multiband structure.

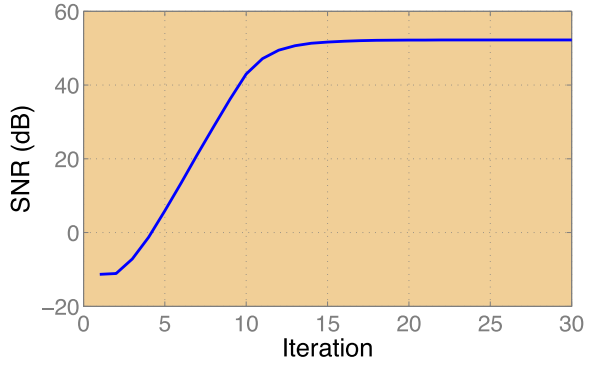

Figure 10 SNR vs. the no. of iterations for sparse signal recovery using the IMAT (Table 12).

Clearly, this assumption includes a more general class of signals than the ones previously studied.

Since the concept of sparsity in a transform domain is more convenient to study for discrete signals, most of the research in this field is focused along discrete type signals [79]; however, recent results [80] show that most of the work can be generalized to continuous signals in shift-invariant subspaces (a subclass of the signals which are represented by Riesz basis).[c] We first study discrete signals and then briefly discuss the extension to the continuous case.

3.2.1 CS mathematical modeling

Let the vector $\mathbf{x} \in \mathbb{R}^n$ be a finite length discrete signal which has to be under-sampled. We assume that $\mathbf{x}$ has a sparse representation in a transform domain denoted by a unitary matrix $\mathbf{\Psi}_{n\times n}$; i.e., we have:

$$\mathbf{x} = \mathbf{\Psi} \cdot \mathbf{s} \tag{16}$$

where $\mathbf{s}$ is an $n\times 1$ vector which has at most k non-zero elements (k-sparse vectors). In practical cases, $\mathbf{s}$ has at most k significant elements and the insignificant elements are set to zero which means $\mathbf{s}$ is an almost k-sparse vector. For example, $\mathbf{x}$ can be the pixels of an image and $\mathbf{\Psi}$ can be the corresponding IDCT matrix. In this case, most of the DCT coefficients are insignificant and if they are set to zero, the quality of the image will not degrade significantly. In fact, this is the main concept behind some of the lossy compression methods such as JPEG. Since the inverse transform on $\mathbf{x}$ yields $\mathbf{s}$, the vector $\mathbf{s}$ can be used instead of $\mathbf{x}$, which can be succinctly represented by the locations and values of the nonzero elements of $\mathbf{s}$. Although this method efficiently compresses $\mathbf{x}$, it initially requires all the samples of $\mathbf{x}$ to produce $\mathbf{s}$, which undermines the whole purpose of CS.

Now let us assume that instead of samples of $\mathbf{x}$, we take m linear combinations of the samples (called generalized samples). If we represent these linear combinations by the matrix $\mathbf{\Phi}_{m\times n}$ and the resultant vector of samples by $\mathbf{y}_{m\times 1}$, we have

$$\mathbf{y}_{m\times 1} = \mathbf{\Phi}_{m\times n} \cdot \mathbf{x}_{n\times 1} = \mathbf{\Phi}_{m\times n} \cdot \mathbf{\Psi}_{n\times n} \cdot \mathbf{s}_{n\times 1} \tag{17}$$

The question is how the matrix $\mathbf{\Phi}$ and the size m should be chosen to ensure that these samples uniquely represent the original signal $\mathbf{x}$. Obviously, the case of $\mathbf{\Phi} = \mathbf{I}_{n\times n}$ where $\mathbf{I}_{n\times n}$ is an $n \times n$ identity matrix yields a trivial solution (keeping all the samples of $\mathbf{x}$) that does not employ the sparsity condition. We look for $\mathbf{\Phi}$ matrices with as few rows as possible which can guarantee the invertibility, stability, and robustness of the sampling process for the class of sparse inputs.

To solve this problem, we introduce probabilistic measures; i.e., instead of exact recovery of signals, we focus on the probability that a random sparse signal (according

to a given probability density function) fails to be reconstructed using its generalized samples. If the probability δ of failure can be made arbitrarily small, then the sampling scheme (the joint pair of $\mathbf{\Psi}, \mathbf{\Phi}$) is successful in recovering $\mathbf{x}$ with probability $1-\delta$, i.e., with high probability.

Let us assume that $\mathbf{\Phi}^{(m)}$ represents the submatrix formed by m random (uniform) rows of an orthonormal matrix $\mathbf{\Phi}_{n\times n}$. It is apparent that if we use $\{\mathbf{\Phi}^{(m)}\}_{m=0}^{n}$ as the sampling matrices for a given sparsity domain, the failure probabilities for $\mathbf{\Phi}^{(0)}$ and $\mathbf{\Phi}^{(n)}$ are, respectively, one and zero, and as the index m increases, the failure probability decreases. The important point shown in [81] is that the decreasing rate of the failure probability is exponential with respect to $\frac{m}{k}$. Therefore, we expect to reach an almost zero failure probability much earlier than $m = n$ despite the fact that the exact rate highly depends on the mutual behavior of the two matrices $\mathbf{\Psi}, \mathbf{\Phi}$. More precisely, it is shown in [81] that

$$P_{\text{failure}} < n \cdot e^{-\frac{c}{\mu^2(\mathbf{\Psi},\mathbf{\Phi})}\cdot\frac{m}{k}} \tag{18}$$

where P_{failure} is the probability that the original signal cannot be recovered from the samples, c is a positive constant, and $\mu(\mathbf{\Psi}, \mathbf{\Phi})$ is the maximum coherence between the columns of $\mathbf{\Psi}$ and rows of $\mathbf{\Phi}$ defined by [82]:

$$\mu(\mathbf{\Psi}, \mathbf{\Phi}) = \max_{1\le a,b\le n} \left|\langle \boldsymbol{\psi}_a\,,\, \boldsymbol{\phi}_b\rangle\right| \tag{19}$$

where $\boldsymbol{\psi}_a, \boldsymbol{\phi}_b$ are the a^{th} column and the b^{th} row of the matrices $\mathbf{\Psi}$ and $\mathbf{\Phi}$, respectively. The above result implies that the probability of reconstruction is close to one for

$$m \ge \underbrace{\mu^2(\mathbf{\Psi}, \mathbf{\Phi})}_{\ge\, \mu^2(\mathbf{\Psi},\mathbf{\Phi}^{(m)})} \frac{k\cdot \ln n}{c} \tag{20}$$

The above derivation implies that the smaller the maximum coherence between the two matrices, and the lower is the number of required samples. Thus, to decrease the number of samples, we should look for matrices $\mathbf{\Phi}$ with low coherence with $\mathbf{\Psi}$. For this purpose, we use a random $\mathbf{\Phi}$. It is shown that the coherence of a random matrix with i.i.d. Gaussian distribution with any unitary $\mathbf{\Psi}$ is considerably small [29], which makes it a proper candidate for the sampling matrix. Investigation of the probability distribution has shown that the Gaussian PDF is not the only solution (for example binary Bernouli distribution and other types are considered in [83]) but may be the simplest to analyze.

For the case of random matrix with i.i.d. Gaussian distribution (or more general distributions for which the concentration inequality holds [83]), a stronger inequality compared with (20) is valid; this implies that for the reconstruction with a probability of almost one, the following condition for the number of samples m suffices [2,79]:

$$m \ge c'\, k \log\left(\frac{n}{k}\right) \tag{21}$$

Notice that the required number of samples given in (20) is for random sampling of an orthonormal basis while (21) represents the required number of samples with i.i.d. Gaussian distributed sampling matrix. Typically, the number in (21) is less than that of (20).

3.2.2 Reconstruction from compressed measurements

In this section, we consider reconstruction algorithms and the stability robustness issues. We briefly discuss the following three methods: a—geometric, b—combinatorial, and c—information theoretic. The first two methods are standard while the last one is more recent.

Geometric methods The oldest methods for reconstruction from compressed sampling are geometric, i.e., ℓ_1 minimization techniques for finding a k-sparse vector $\mathbf{s} \in \mathbb{R}^n$ from a set of $m = O\left(k\log(n)\right)$ measurements (y_is); see e.g., [29,81,84-86]. Let us assume that we have applied a suitable $\mathbf{\Phi}$ which guarantees the invertibility of the sampling process. The reconstruction method should be a technique to recover a k-sparse vector $\mathbf{s}_{n\times 1}$ from the observed samples $\mathbf{y}_{m\times 1} = \mathbf{\Phi}_{m\times n}\cdot\mathbf{\Psi}_{n\times n}\cdot\mathbf{s}_{n\times 1}$ or possibly $\mathbf{y}_{m\times 1} = \mathbf{\Phi}_{m\times n}\cdot\mathbf{\Psi}_{n\times n}\cdot\mathbf{s}_{n\times 1} + \boldsymbol{\nu}_{m\times 1}$, where $\boldsymbol{\nu}$ denotes the noise vector. Suitability of $\mathbf{\Phi}$ implies that $\mathbf{s}_{n\times 1}$ is the only k-sparse vector that produces the observed samples; therefore, $\mathbf{s}_{n\times 1}$ is also the sparsest solution for $\mathbf{y} = \mathbf{\Phi}\cdot\mathbf{\Psi}\cdot\mathbf{s}$. Consequently, $\mathbf{s}$ can be found using

$$\text{minimize } \|\mathbf{s}\|_{\ell_0} \quad \text{subject to } \mathbf{y} = \mathbf{\Phi}\cdot\mathbf{\Psi}\cdot\mathbf{s} \tag{22}$$

Good methods for the minimization of an ℓ_0-norm (sparsity) do not exist. The ones that are known are either computationally prohibitive or are not well behaved when the measurements are corrupted with noise. However, it is shown in [82] and later in [76,87] that minimization of an ℓ_1-norm results in the same vector $\mathbf{s}$ for many cases:

$$\text{minimize } \|\mathbf{s}\|_{\ell_1} \quad \text{subject to } \mathbf{y} = \mathbf{\Phi}\cdot\mathbf{\Psi}\cdot\mathbf{s} \tag{23}$$

The interesting part is that the number of required samples to replace ℓ_0 with ℓ_1-minimization has the same order of magnitude as the one for the invertibility of the sampling scheme. Hence, $\mathbf{s}$ can be derived from (22) using ℓ_1-minimization. It is worthwhile to mention that replacement of ℓ_1-norm with ℓ_2-norm, which is faster to implement, does not necessarily produce reasonable solutions. However, there are greedy methods (Matching Pursuit as discussed in Section 7 on SCA [40,88]) which iteratively approach the best solution and compete with the ℓ_1-norm optimization (equivalent to Basis Pursuit methods as discussed in Section 7 on SCA).

To show the performance of the BP method, we have reported the famous phase transition diagram from [89] in Figure 11; this figure characterizes the perfect reconstruction region with respect to the parameters k/m and m/n. In fact, the curve represents the points for which the BP method recovers the sparse signal measured through a Gaussian random matrix with probability 50%. The interesting point is that the transition from the high-probability region (below the curve) to the low-probability one (above the curve) is very sharp and when $n \to \infty$ the plotted curve separates the regions for probabilities 0 and 100%. The empirical results show that by deviating from the Gaussian distribution, the curve does not change while it is yet to be proved [89].

A sufficient condition for these methods to work is that the matrix $\mathbf{\Phi} \cdot \mathbf{\Psi}$ must satisfy the so-called *restricted isometric property* (RIP) [75,83,90]; which will be discussed in the following section.

Restricted isometric property It is important to note that the ℓ_1-minimization algorithm produces almost optimal results for signals that are not k-sparse. For example, almost sparse signals (*compressible signals*) are more likely to occur in applications than exactly k-sparse vectors, (e.g., the wavelet transform of an image consists mostly of small coefficients and a few large coefficients). Moreover, even exactly k-sparse signals may be corrupted by additive noise. This characteristic of ℓ_1-minimization algorithms is called *stability*. Specifically, if we let $\beta_k(\mathbf{s})$ denote the smallest possible error (in the ℓ_1-norm) that can be achieved by approximating a signal $\mathbf{s}$ by a k-sparse vector $\mathbf{z}$

$$\beta_k(\mathbf{s}) := \inf\{\|\mathbf{s}-\mathbf{z}\|_{\ell_1}, \|\mathbf{z}\|_{\ell_0} \le k\},$$

then the vector $\hat{\mathbf{s}}$ produced by the ℓ_1-reconstruction method is almost optimal in the sense that $\|\mathbf{s}-\hat{\mathbf{s}}\|_{\ell_1} \le C\beta_k(\mathbf{s})$ for some constant C independent of $\mathbf{s}$. An implication of stability is that small perturbations in the signal caused by noise result in small distortions in the output solution. The previous result means that if $\mathbf{s}$ is not k-sparse, then $\hat{\mathbf{s}}$ is close to the k-sparse vector $\mathbf{s}_k^o$ that has the k-largest components of $\mathbf{s}$. In particular, if $\mathbf{s}$ is k-sparse, then $\mathbf{s}_k^o = \mathbf{s}$. This stability property is different from the so-called *robustness* which is another important characteristic that we wish to have in any reconstruction algorithm. Specifically, an algorithm is *robust* if small perturbations in the measurements are reflected in small errors in the reconstruction. Both stability and robustness are achieved by the ℓ_1-minimization algorithms (after a slight modification of (22), see [83,91]). Although the two concepts of robustness and stability can be related, they are not the same.

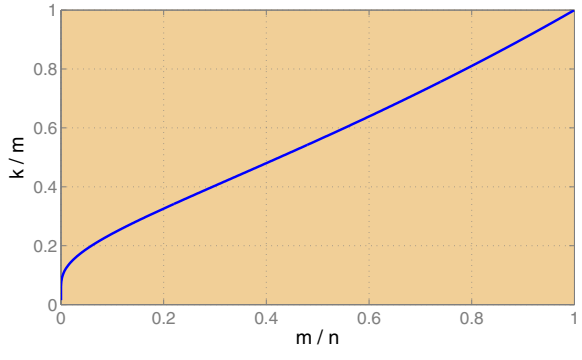

Figure 11 The phase transition of the BP method for reconstruction of the sparse vector from Gaussian random measurement matrices; the probability of perfect reconstruction for the pairs of $\frac{k}{m}$ and $\frac{m}{n}$ that stand above and below the curve are, respectively, 0 and 1 asymptotically.

In compressed sensing, the degree of stability and robustness of the reconstruction is determined by the characteristics of the sampling matrix $\mathbf{\Phi}$. We say that the matrix $\mathbf{\Phi}$ has RIP of order k, when for all k-sparse vectors $\mathbf{s}$, we have [30,76]:

$$1-\delta_k \le \frac{\|\mathbf{\Phi}\cdot\mathbf{s}\|_{\ell_2}^2}{\|\mathbf{s}\|_{\ell_2}^2} \le 1+\delta_k \tag{24}$$

where $0 \le \delta_k < 1$ (isometry constant). The RIP is a sufficient condition that provides us with the maximum and minimum power of the samples with respect to the input power and ensures that none of the k-sparse inputs fall in the null space of the sampling matrix. The RIP property essentially states that every k columns of the matrix $\mathbf{\Phi}_{m\times n}$ must be almost orthonormal (these submatrices preserve the norm within the constants $1 \pm \delta_k$). The explicit construction of a matrix with such a property is difficult for any given n, k and $m \approx k \log n$; however, the problem has been studied in some cases [37,92]. Moreover, given such a matrix $\mathbf{\Phi}$, the evaluation of $\mathbf{s}$ (or alternatively $\mathbf{x}$) via the minimization problem involves numerical methods (e.g., linear programming, GPSR, SPGL1, FPC [44,93]) for n variables and m constraints which can be computationally expensive.

However, probabilistic methods can be used to construct $m \times n$ matrices satisfying the RIP property for a given n, k and $m \approx k \log n$. This can be achieved using Gaussian random matrices. If $\mathbf{\Phi}$ is a sample of a Gaussian random matrix with the number of rows satisfying (20), $\mathbf{\Phi}\cdot\mathbf{\Psi}$ is also a sample of a Gaussian random matrix with the same number of rows and thus it satisfies RIP with high probability. Using matrices with the appropriate RIP property in the ℓ_1-minimization, we guarantee exact recovery of k-sparse signals that are stable and robust against additive noise.

Without loss of generality, assume that Ψ is equal to the identity matrix I, and that instead of $\mathbf{\Phi}\cdot\mathbf{s}$, we measure

$\mathbf{\Phi}\cdot\mathbf{s}+\boldsymbol{\nu}$, where $\boldsymbol{\nu}$ represents an additive noise vector. Since $\mathbf{\Phi}\cdot\mathbf{s}+\boldsymbol{\nu}$ may not belong to the range space of $\mathbf{\Phi}$ over k-sparse vectors, the ℓ_1 minimization of (25) is modified as follows:

$$\text{minimize } \|\mathbf{s}\|_{\ell_1} \quad \text{subject to } \|\mathbf{y}-\mathbf{\Phi}\cdot\mathbf{s}\|_{\ell_2}\le\epsilon \tag{25}$$

where ϵ^2 is the maximum noise power. Let us denote the result of the above minimization for $\mathbf{y}=\mathbf{\Phi}\cdot\mathbf{s}+\boldsymbol{\nu}$ by $\hat{\mathbf{s}}$. With the above algorithm, it can be shown that

$$\|\hat{\mathbf{s}}-\mathbf{s}\|_{\ell_2}\le\epsilon \tag{26}$$

This shows that small perturbations in the measurements cause small perturbations in the output of the ℓ_1-minimization method (robustness).

Combinatorial Another standard approach for reconstruction of compressed sampling is combinatorial. As before, without loss of generality $\Psi = I$. The sampling matrix $\mathbf{\Phi}$ is found using a bipartite graph which consists of binary entries, i.e., entries that are either 1 or 0. Binary search methods are then used to find an unknown k-sparse vector $\mathbf{s}\in\mathbb{R}^n$, see, e.g., [84,94-100] and the references therein. Typically, the binary matrix $\mathbf{\Phi}$ has $m = O(k\log n)$ rows, and there exist fast algorithms for finding the solution $\mathbf{x}$ from the m measurements (typically a linear combination). However, the construction of $\mathbf{\Phi}$ is also difficult.

Information theoretic A more recent approach is adaptive and information theoretic [101]. In this method, the signal $\mathbf{s}\in\mathbb{R}^n$ is assumed to be an instance of a vector random variable $\mathbf{s}=(s_1,\ldots,s_n)^t$, where $(.)^t$ denotes transpose operator, and the *ith* row of $\mathbf{\Phi}$ is constructed using the value of the previous sample y_{i-1}. Tools from the theory of Huffman coding are used to develop a deterministic construction of a sequence of binary sampling vectors (i.e., their components consist of 0 or 1) in such a way as to minimize the average number of samples (rows of $\mathbf{\Phi}$) needed to determine a signal. In this method, the construction of the sampling vectors can always be obtained. Moreover, it is proved that the expected total cost (number of measurements and reconstruction combined) needed to sample and reconstruct a k-sparse vector in R^n is no more than $k\log n+2k$.

3.3 Sampling with finite rate of innovation

The classical sampling theorem states that

$$x(t)=\sum_{i\in\mathbb{Z}} x\left(\frac{i}{2B}\right)\cdot\operatorname{sinc}(2Bt-i) \tag{27}$$

where B is the bandwidth of $x(t)$ with the Nyquist interval $T_s=1/2B$. These uniform samples can be regarded as the degrees of freedom of the signal; i.e., a lowpass signal with bandwidth B has one degree of freedom in each Nyquist interval T_s. Replacing the *sinc* function with other kernels in (27), we can generalize the sparsity (bandlimitedness) in the Fourier domain to a wider class of signals known as the shift invariant (SI) spaces:

$$x(t)=\sum_{i\in\mathbb{Z}} c_i\cdot\varphi\left(\frac{t}{T_s}-i\right) \tag{28}$$

Similarly, the above signals have one degree of freedom in each T_s period of time (the coefficients c_i). A more general definition for the degree of freedom is introduced in [3] and is named the *Rate of Innovation*. For a given signal model, if we denote the degree of freedom in the time interval of $[t_1,t_2]$ by $C_x(t_1,t_2)$, the local rate of innovation is defined by $\frac{1}{t_2-t_1}C_x(t_1,t_2)$ and the global rate of innovation (ρ) is defined as

$$\rho=\lim_{\tau\to\infty}\frac{1}{2\tau}C_x(t-\tau,t+\tau) \tag{29}$$

provided that the limit exists; in this case, we say that the signal has finite rate of innovation [3,27,102,103]. As an example, for the lowpass signals with bandwidth B we have $\rho=2B$, which is the same as the Nyquist rate. In fact by proper choice of the sampling process, we are extracting the innovations of the signal. Now the question that arises is whether the uniform sampling theorems can be generalized to the signals with finite rate of innovation. Answer is positive for a class of non-bandlimited signals including the SI spaces. Consider the following signals:

$$x(t)=\sum_{i\in\mathbb{Z}}\sum_{r=1}^{R} c_{i,r}\cdot\varphi_r\left(\frac{t-t_i}{T_s}\right) \tag{30}$$

where $\{\varphi_r(t)\}_{r=1}^{k}$ are arbitrary but known functions and $\{t_i\}_{i\in\mathbb{Z}}$ is a realization of a point process with mean μ. The free parameters of the above signal model are $\{c_{i,r}\}$ and $\{t_i\}$. Therefore, for this class of signals we have $\rho=\frac{2}{\mu}$; however, the classical sampling methods cannot reconstruct these kinds of signals with the sampling rate predicted by ρ. There are many variations for the possible choices of the functions $\varphi_r(t)$; nonetheless, we just describe the simplest version. Let the signal $x(t)$ be a finite mixture of sparse Dirac functions:

$$x(t)=\sum_{i=1}^{k} c_i\cdot\delta(t-t_i) \tag{31}$$

where $\{t_i\}$ is assumed to be an increasing sequence. For this case, since there are k unknown time instants and k unknown coefficients, we have $C_x(t_1,t_k)=2k$. We intend to show that the samples generated by proper sampling kernels $\varphi(t)$ can be used to reconstruct the sparse Dirac

functions. In fact, we choose the kernel $\varphi(t)$ to satisfy the so called Strang-Fix condition of order $2k$:

$$\forall\, 0 \le r \le 2k-1,\ \ \exists\, \{\alpha_{r,i}\}_{i\in\mathbb{Z}}: \quad \sum_{i\in\mathbb{Z}} \alpha_{r,i}\varphi(t-i) = t^r \tag{32}$$

The above condition for the Fourier domain becomes

$$\begin{cases} \Phi\big(\Omega = 0\big) \neq 0 \\ \Phi^{(r)}\big(\Omega = 2\pi i\big) = 0, \quad \forall\, i \neq 0 \in \mathbb{Z} \end{cases} \qquad r = 0, \ldots, 2k-1 \tag{33}$$

where $\Phi(\Omega)$ denotes the Fourier transform of $\varphi(t)$, and the superscript (r) represents the r^{th} derivative. It is also shown that such functions are of the form $\varphi(t) = f(t) * \beta_{2k}(t)$, where $\beta_{2k}(t)$ is the B-spline of order $2k^{th}$ and $f(t)$ is an arbitrary function with nonzero DC frequency [102]. Therefore, the function $\beta_{2k}(t)$ is itself among the possible options for the choice of $\varphi(t)$.

We can show that for the sampling kernels which satisfy the Strang-Fix condition (32), the innovations of the signal $x(t)$ (31) can be extracted from the samples $(y[j])$:

$$y[j] = \left(x(t) * \varphi\left(-\frac{t}{T_s}\right)\right)\Big|_{t=j\cdot T_s} = \sum_{i=1}^{k} c_i \varphi(t_i - j) \tag{34}$$

Thus,

$$\begin{aligned} \tau_r &\triangleq \sum_{j\in\mathbb{Z}} \alpha_{r,j} y[j] \\ &= \sum_{i=1}^{k} c_i \sum_{j\in\mathbb{Z}} \alpha_{r,j}\varphi(t_i - j) = \sum_{i=1}^{k} c_i t_i^r \end{aligned} \tag{35}$$

In other words, we have filtered the discrete samples $(y[j])$ in order to obtain the values τ_r; (35) shows that these values are only a function of the innovation parameters (amplitudes c_i and time instants t_i). However, the values τ_r are nonlinearly related to the time instants and therefore, the innovations cannot be extracted from τ_r using linear algebra.[d] However, these nonlinear equations form a well-known system which was studied by *Prony* in the field of spectral estimation (see Section 5.1) and its discrete version is also employed in both real and Galois field versions of Reed-Solomon codes (see Section 4.1). This method which is called the *annihilating filter* is as follows:

The sequence $\{\tau_r\}$ can be viewed as the solution of a recursive equation. In fact if we define $H(z) = \sum_{i=0}^{k} h_i z^i = \prod_{i=1}^{k}(z - t_i)$, we will have (see Section 4.1 and Appendices 1, 2 for the proof of a similar theorem):

$$\forall\, r: \quad \tau_{r+k} = -\sum_{i=1}^{k} h_i \cdot \tau_{r+i-1} \tag{36}$$

In order to find the time instants t_i, we find the polynomial $H(z)$ (or the coefficients h_i) and we look for its roots. A recursive relation for τ_r becomes

$$\begin{bmatrix} \tau_1 & \tau_2 & \ldots & \tau_k \\ \tau_2 & \tau_3 & \ldots & \tau_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_k & \tau_{k+1} & \ldots & \tau_{2k-1} \end{bmatrix} \cdot \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \end{bmatrix} = - \begin{bmatrix} \tau_{k+1} \\ \tau_{k+2} \\ \vdots \\ \tau_{2k} \end{bmatrix} \tag{37}$$

By solving the above linear system of equations, we obtain coefficients h_i (for a discussion on invertibility of the left side matrix see [102,104]) and consequently, by finding the roots of $H(z)$, the time instants will be revealed. It should be mentioned that the choice of $\tau_1, \ldots, \tau_{2k}$ in (37) can be replaced with any $2k$ consecutive terms of $\{\tau_i\}$. After determining $\{t_i\}$, (35) becomes a linear system of equations with respect to the values $\{c_i\}$ which could be easily solved.

This reconstruction method can be used for other types of signals satisfying (30) such as the signals represented by piecewise polynomials [102] (for large enough n, the n^{th} derivative of these signals become delta functions). An important issue in nonlinear reconstruction is the noise analysis; for the purpose of denoising and performance under additive noise the reader is encouraged to see [27].

A nice application of sampling theory and the concept of sparsity is error correction codes for real and complex numbers [105]. In the next section, we shall see that similar methods can be employed for decoding block and convolutional codes.

4 Error correction codes: Galois and real/complex fields

The relation between sampling and channel coding is the result of the fact that over-sampling creates redundancy [105]. This redundancy can be used to correct for "sparse" impulsive noise. Normally, the channel encoding is performed in finite Galois fields as opposed to real/complex fields; the reason is the simplicity of logic circuit implementation and insensitivity to the pattern of errors. On the other hand, the real/complex field implementation of error correction codes has stability problems with respect to the pattern of impulsive, quantization and additive noise [52,59,74,106-109]. Nevertheless, such implementation has found applications in fault tolerant computer systems [110-114] and impulsive noise removal from 1-D and 2-D signals [31,32]. Similar to finite Galois fields, real/complex field codes can be implemented in both block and convolutional fashions.

A discrete real-field block code is an oversampled signal with n samples such that, in the transform domain (e.g., DFT), a contiguous number of high-frequency components are zero. In general, the zeros do not have to be the high-frequency components or contiguous. However,

if they are contiguous, the resultant m equations (from the syndrome information domain) and m unknown erasures form a Vandermonde matrix, which ensures invertibility and consequently erasure recovery. The DFT block codes are thus a special case of Reed-Solomon (RS) codes in the field of real/complex numbers [105].

Figure 12 represents convolutional encoders of rate 1/2 of finite constraint length [105] and infinite precision per symbol. Figure 12a is a systematic convolutional encoder and resembles an oversampled signal discussed in Section 3 if the FIR filter acts as an ideal interpolating filter. Figure 12b is a non-systematic encoder used in the simulations to be discussed subsequently. In the case of additive impulsive noise, errors could be detected based on the side information that there are frequency gaps in the original oversampled signal (syndrome). In the following subsections, various algorithms for decoding along with simulation results are given for both block and convolutional codes. Some of these algorithms can be used in other applications such as spectral and channel estimation.

4.1 Decoding of block codes—ELP method

Iterative reconstruction for an erasure channel is identical to the missing sampling problem [115] discussed in Section 3.1.1 and therefore, will not be discussed here. Let us assume that we have a finite discrete signal $x_{\text{orig}}[\,i]$, where $i = 1,\ldots,l$. The DFT of this sequence yields l complex coefficients in the frequency domain $(X_{\text{orig}}[\,j]\,,\ j = 1,\ldots,l)$. If we insert p consecutive zeros[e] to get $n = l + p$ samples $(X[\,j]\,,\ j = 1,\ldots,n)$ and take its inverse DFT, we end up with an oversampled version of the original signal with n complex samples $(x[\,i]\,,\ i = 1,\ldots,n)$. This oversampled signal is real if Hermitian symmetry (complex conjugate symmetry) is preserved in the frequency domain, e.g., the set Θ of p zeros is centered at $\frac{n}{2}$. For erasure channels, the sparse missing samples are denoted by $e[\,i_m] = x[\,i_m]$, where i_ms denote the positions of the lost samples; consequently, for $i \neq i_m$, $e[\,i] = 0$. The Fourier transform of $e[\,i]$ (called $E[\,j]\,,\ j = 1,\ldots,n$) is known for the syndrome positions Θ. The remaining values of $E[\,j]$ can be found from the following recursion (see Appendix 1):

$$E[\,r] = -\frac{1}{h_k}\sum_{t=1}^{k} E[\,r+t]\,h_{k-t} \tag{38}$$

where h_ks are the ELP coefficients as defined in (36) and Appendix 1, r is a member of the complement of Θ, and the index additions are in $mod(n)$. After finding $E[\,j]$ values, the spectrum of the recovered oversampled signal $X[\,j]$ can be found by removing $E[\,j]$ from the received signal (see (99) in Appendix 1). Hence the original signal can be recovered by removing the inserted zeros at the syndrome positions of $X[\,j]$. The above algorithm, called the ELP algorithm, is capable of correcting any combination of erasures. However, if the erasures are bursty, the above algorithm may become unstable. To combat bursty erasures, we can use the Sorted DFT (SDFT[f]) [1,59,116,117] instead of the conventional DFT. The simulation results for block codes with erasure and impulsive noise channels are given in the following two subsections.

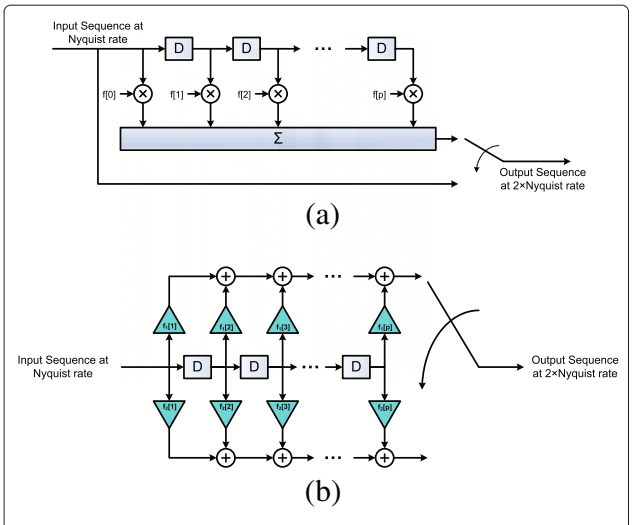

Figure 12 Convolutional encoders. (a) A real-field systematic convolutional encoder of rate $\frac{1}{2}$; $f[\,i]$s are the taps of an FIR filter. **(b)** A non-systematic convolutional encoder of rate $\frac{1}{2}$, $f_1[\,i]$s and $f_2[\,i]$s are the taps of 2 FIR filters.

4.1.1 *Simulation results for erasure channels*

The simulation results for the ELP decoding implementation for $n = 32$, $p = 16$, and $k = 16$ erasures (a burst of 16 consecutive missing samples from position 1 to 16) are shown in Figure 13; this figure shows we can have perfect reconstruction up to the capacity of the code (up to the finite computer precision which is above 320 dB; this is also true for Figures 14 and 15). By capacity we mean the maximum number of erasures that a code is capable of correcting.

Since consecutive sample losses represent the worst case [59,116], the proposed method works better for random samples. In practice, the error recovery capability of this technique degrades with the increase of the block and/or burst size due to the accumulation of round-off errors. In order to reduce the round-off error, instead of the DFT, a transform based on the SDFT, or Sorted DCT (SDCT) can be used [1,59,116]. These types of transformations act as an interleaver to break down the bursty erasures.

4.1.2 *Simulation results for random impulsive noise channel*

There are several methods to determine the number, locations, and values of the impulsive noise samples, namely Modified Berlekamp-Massey for real fields [118,119],

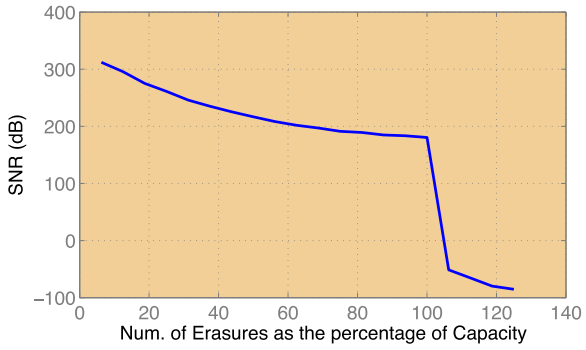

Figure 13 Recovery of a burst of 16 sample losses.

ELP, IMAT, and constant false alarm rate with recursive detection estimation (CFAR-RDE). The Berlekamp-Massey method for real numbers is sensitive to noise and will not be discussed here [118]. The other methods are discussed below.

ELP method [104] When the number and positions of the impulsive noise samples are not known, h_t in (38) is not known for any t; therefore, we assume the maximum possible number of impulsive noise samples per block, i.e., $k = \lfloor \frac{n-l}{2} \rfloor$ as given in (96) in Appendix 1. To solve for h_t, we need to know only $n - l$ samples of E in the positions where zeros are added in the encoding procedure. Once the values of h_t are determined from the pseudo-inverse [104], the number and positions of impulsive noise can be found from (98) in Appendix 1. The actual values of the impulsive noise can be determined from (38) as in the erasure channel case. For the actual algorithm, please refer to Appendix 2. As we are using the above method in the field of real numbers, exact zeros of $\{H_k\}$, which are the DFT of $\{h_i\}$, are rarely observed; consequently, the zeros can be found by thresholding the magnitudes of H_k. Alternatively, the magnitudes of H_k can be used as a mask for soft-decision; in this case, thresholding is not needed.

CFAR-RDE and IMAT methods [31] The CFAR-RDE method is similar to the IMAT with the additional inclusion of the CFAR module to estimate the impulsive noise; CFAR is extensively used in radars to detect and remove clutter noise from data. In CFAR, we compare the noisy signal with its neighbors and determine if an impulsive (sparse) noise is present or not (using soft decision [31]).[g] After removing the impulsive noise in a "soft" fashion, we estimate the signal using the iterative method for an erasure channel as described in Section 3.1.1 for random sampling or using the ELP method. The impulsive noise

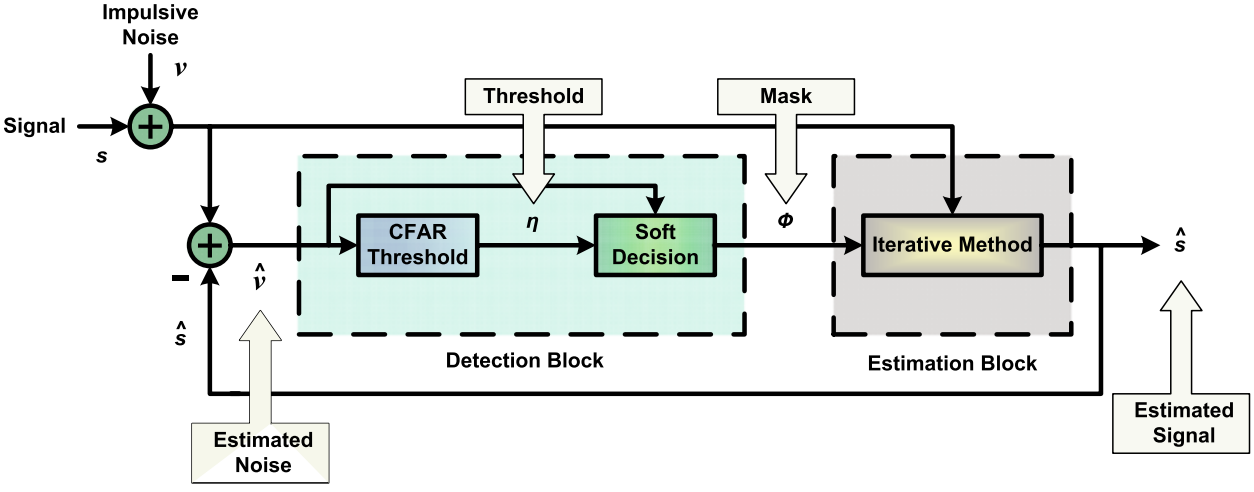

Figure 14 Simulation results of a convolutional decoder, using the iterative method with the generator matrix, after 30 CG iterations (see [72]); SNR versus the relative rate of erasures (w.r.t. full capacity) in an erasure channel.

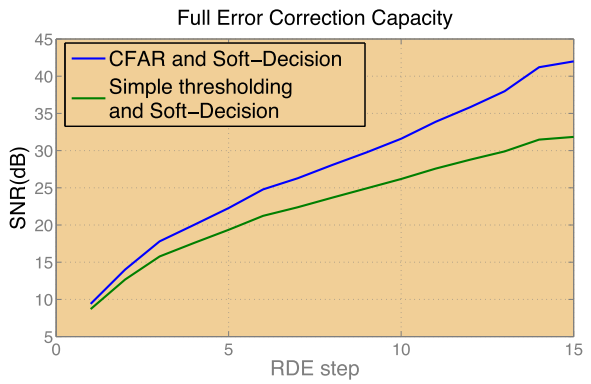

Figure 15 Simulation results by using the IMAT method for detecting the location and amplitude of the impulsive noise, $\lambda = 1.9$.

and signal detection and estimation go through several iterations in a recursive fashion as shown in Figure 16. As the number of recursions increases, the certainty about the detection of impulsive noise locations also increases; thus, the soft decision is designed to act more like the hard decision during the later parts of the iteration steps, which yields the error locations. Meanwhile, further iterations are performed to enhance the quality of the original signal since suppression of the impulsive noise also suppresses the original signal samples at the location of the impulsive noise. The improvement of using CFAR-RDE over a simple soft decision RDE is shown in Figure 17.

4.2 Decoding for convolutional codes

The performance of convolutional decoders depends on the coding rate, the number and values of FIR taps for the encoders, and the type of the decoder. Our simulation results are based on the structure given in Figure 12b, and the taps of the encoder are

$$\begin{aligned} f_1 &= [1, 2, 3, 4, 5, 16], \\ f_2 &= [16, 5, 4, 3, 2, 1] \end{aligned} \tag{39}$$

The input signal is taken from a uniform random distribution of size 50 and the simulations are run 1,000 times and then averaged. The following subsections describe the simulation results for erasure and impulsive noise channels.

4.2.1 *Decoding for erasure channels*

For the erasure channels, we derive the generator matrix of a convolutional encoder (Figure 12b with taps given in (39)) as shown below [4]

$$G = \begin{bmatrix} f_1[1] & 0 & \dots \\ f_2[1] & 0 & \dots \\ f_1[2] & f_1[1] & \dots \\ f_2[2] & f_2[1] & \dots \\ \vdots & \vdots & \dots \\ f_1[n] & f_1[n-1] & \dots \\ f_2[n] & f_2[n-1] & \dots \\ 0 & f_1[n] & \dots \\ 0 & f_2[n] & \dots \\ 0 & 0 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}^T \tag{40}$$

An iterative decoding scheme for this matrix representation is similar to that of Figure 7 except that the operator G consists of the generator matrix, a mask (erasure operation), and the transpose of the generator matrix. If the rate of erasure does not exceed the encoder full capacity, the

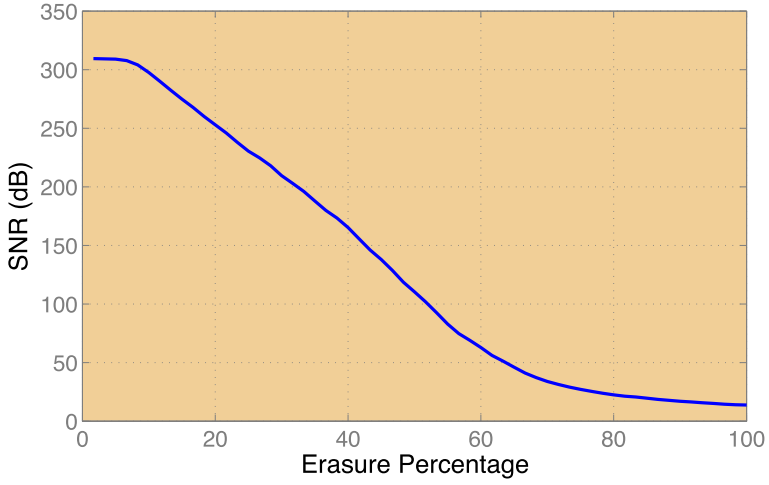

Figure 16 CFAR-RDE method with the use of adaptive soft thresholding and an iterative method for signal reconstruction.

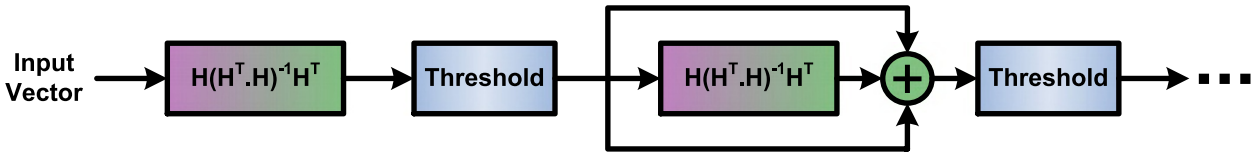

Figure 17 Comparison of CFAR-RDE and a simple soft decision RDE for DFT block codes.

matrix form of the operator $\mathbf{G}$ can be shown to be a non-negative definite square matrix and therefore its inverse exists [51,60].

Figure 14 shows that the SNR values gradually decrease as the rate of erasure reaches its maximum (capacity).

4.2.2 Decoding for impulsive noise channels

Let us consider $\mathbf{x}$ and $\mathbf{y}$ as the input and the output streams of the encoder, respectively, related to each other through the generator matrix $\mathbf{G}$ as $\mathbf{y} = \mathbf{G}\mathbf{x}$.

Denoting the observation vector at the receiver by $\hat{\mathbf{y}}$, we have $\hat{\mathbf{y}} = \mathbf{y} + \boldsymbol{\nu}$, where $\boldsymbol{\nu}$ is the impulsive noise vector. Multiplying $\hat{\mathbf{y}}$ by the transpose of the parity check matrix $\mathbf{H}^T$, we get

$$\mathbf{H}^T\hat{\mathbf{y}} = \mathbf{H}^T\boldsymbol{\nu} \tag{41}$$

Multiplying the resultant by the right pseudo-inverse of the $\mathbf{H}^T$, we derive

$$\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\hat{\mathbf{y}} = \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\boldsymbol{\nu} = \tilde{\boldsymbol{\nu}} \tag{42}$$

Thus by multiplying the received vector by $\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ (projection matrix into the range space of $\mathbf{H}$), we obtain an approximation of the impulsive noise. In the IMAT method, we apply the operator $\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ in the iteration of Figure 9; the threshold level is reduced exponentially at each iteration step. The block diagram of IMAT in Figure 9 is modified as shown in Figure 18.

For simulation results, we use the generator matrix shown in (40), which can be calculated from [4].

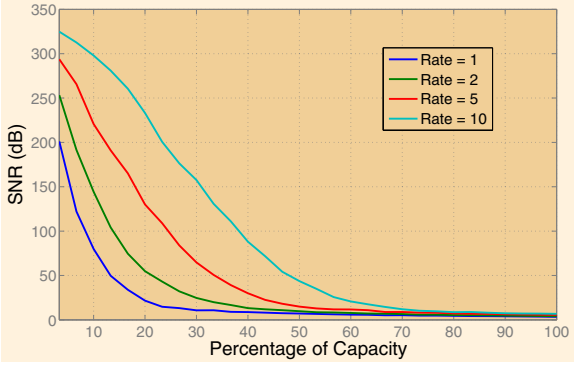

Figure 18 The modified diagram of the IMAT method from Figure 9.

In our simulations, the locations of the impulsive noise samples are generated randomly and their amplitudes have Gaussian distributions with zero mean and variance equal to 1, 2, 5, and 10 times the variance of the encoder output. The results are shown in Figure 15 after 300 iterations. This figure shows that the high variance impulsive noise has a better performance.

5 Spectral estimation

In this section, we review some of the methods which are used to evaluate the frequency content of data [7-10]. In the field of signal spectrum estimation, there are several methods which are appropriate for different types of signals. Some methods are more suitable to estimate the spectrum of wideband signals, whereas some others are better for the extraction of narrow-band components. Since our focus is on sparse signals, it would be reasonable to assume sparsity in the frequency domain, i.e., we assume the signal to be a combination of several sinusoids plus white noise.

Conventional methods for spectrum analysis are nonparametric methods in the sense that they do not assume any model (statistical or deterministic) for the data, except that it is zero or periodic outside the observation interval. For example, the periodogram $\hat{P}_{\text{per}}(f)$ is a well-known nonparametric method that can be computed via the FFT algorithm:

$$\hat{P}_{\text{per}}(f) = \frac{1}{mT_s}\left|T_s\sum_{r=0}^{m-1}x_r e^{-j2\pi f r T_s}\right|^2 \tag{43}$$

where m is the number of observations, T_s is the sampling interval (usually assumed as unity), and x_r is the signal. Although non-parametric methods are robust with low computational complexity, they suffer from fundamental limitations. The most important limitation is their resolution; too closely spaced harmonics cannot be distinguished if the spacing is smaller than the inverse of the observation period.

To overcome this resolution problem, parametric methods are devised. Assuming a statistical model with some unknown parameters, we can increase resolution by estimating the parameters from the data at the cost of more computational complexity. Theoretically, in parametric methods, we can resolve closely spaced harmonics with limited data length if the SNR goes to infinity.[h]

In this section, we shall discuss three parametric approaches for spectral estimation: the Pisarenko, the Prony, and the MUSIC algorithms. The first two are mainly used in spectral estimation, while the MUSIC algorithm was first developed for array processing and later has been extended to spectral estimation. It should be noted that the parametric methods unlike the non-parametric approaches require prior knowledge of the model order (the number of tones). This can be decided from the data using the minimum discription length (MDL) method discussed in the next section.

5.1 Prony method

The Prony method was originally proposed for modeling the expansion of gases [120]; however, now it is known as a general spectral estimation method. In fact, Prony tried to fit a weighted mixture of k damped complex exponentials to $2k$ data measurements. The original approach is related to the noiseless measurements; however, it has been extended to produce the least squared solutions for noisy measurements. We focus only on the noiseless case here. The signal is modeled as a weighted mixture of k complex exponentials with complex amplitudes and frequencies:

$$x_r = \sum_{i=1}^{k} b_i z_i^r \tag{44}$$

where x_r is the noiseless discrete sparse signal consisting of k exponentials with parameters

$$\begin{aligned} b_i &= a_i e^{j\theta_i} \\ z_i &= e^{j2\pi f_i T_s} \end{aligned} \tag{45}$$

where a_i, θ_i, f_i represent the amplitude, phase, and the frequency (f_i is a complex number in general), respectively. Let us define the polynomial $H(z)$ such that its roots represent the complex exponential functions related to the sparse tones (see Section 3.3 on FRI, (38) on ELP and Appendix 1):

$$H(z) = \prod_{i=1}^{k} (z - z_i) = \sum_{i=0}^{k} h_i z^{k-i} \tag{46}$$

By shifting the index of (44) and multiplying by the parameter h_j and summing over j we get

$$\sum_{j=0}^{k} h_j x_{r-j} = \sum_{i=1}^{k} b_i z_i^{r-k} \sum_{j=0}^{k} h_j z_i^{k-j} = 0 \tag{47}$$

where r is indexed in the range $k+1 \le r \le 2k$. This formula implies a recursive equation to solve for h_is [8]. After the evaluation of the h_is, the roots of (46) yield the frequency components. Hence, the amplitudes of the exponentials can be evaluated from a set of linear equations given in (44). The basic Prony algorithm is given in Table 13.

The Prony method is sensitive to noise, which was also observed in the ELP and the annihilating filter methods discussed in Sections 3.3 and 4.1. There are extended Prony methods that are better suited for noisy measurements [10].

5.2 Pisarenko harmonic decomposition (PHD)

The PHD method is based on the polynomial of the Prony method and utilizes the eigen-decomposition of the data covariance matrix [10]. Assume k complex tones are present in the spectrum of the signal. Then, decompose the covariance matrix of $k+1$ dimensions into a k-dimensional signal subspace and a 1-dimensional noise subspace that are orthogonal to each other. By including the additive noise, the observations are given by

$$y_r = x_r + \nu_r \tag{48}$$

where y is the observation sample and ν is a zero-mean noise term that satisfies $E\{\nu_r \nu_{r+i}\} = \sigma^2 \delta[i]$. By replacing $x_r = y_r - \nu_r$ in the difference equation (47), we get

$$\sum_{i=0}^{k} h_i y_{r-i} = \sum_{i=0}^{k} h_i \nu_{r-i} \tag{49}$$

which reveals the auto-regressive moving average (ARMA) structure (order (k,k)) of the observations y_r as a random process. To benefit from the tools in linear algebra, let us define the following vectors:

$$\begin{aligned} \mathbf{y} &= [y_r, \ \ldots, \ y_{r-k}]^T \\ \mathbf{h} &= [1, \ h_1, \ \ldots, \ h_k]^T \\ \boldsymbol{\nu} &= [\nu_r, \ \ldots, \ \nu_{r-k}]^T \end{aligned} \tag{50}$$

Now (49) can be written as

$$\mathbf{y}^H \mathbf{h} = \boldsymbol{\nu}^H \mathbf{h} \tag{51}$$

Multiplying both sides of (51) by $\mathbf{y}$ and taking the expected value, we get $E\{\mathbf{y}\mathbf{y}^H\}\mathbf{h} = E\{\mathbf{y}\boldsymbol{\nu}^H\}\mathbf{h}$. Note that

$$E\{\mathbf{y}\mathbf{y}^H\} = \mathbf{R}_{\mathbf{yy}} \tag{52}$$

$$E\{\mathbf{y}\boldsymbol{\nu}^H\} = E\{(\mathbf{x}+\boldsymbol{\nu})\boldsymbol{\nu}^H\} = E\{\boldsymbol{\nu}\boldsymbol{\nu}^H\} = \sigma^2 \mathbf{I} \tag{53}$$

We thus have an eigen-equation

$$\mathbf{R}_{\mathbf{yy}} \mathbf{h} = \sigma^2 \mathbf{h} \tag{54}$$

Table 13 Basic prony algorithm

1. Solve the recursive equation in (47) to evaluate h_is.
2. Find the roots of the polynomial represented in (46); these roots are the complex exponentials defined as z_i in (44).
3. Solve (44) to obtain the amplitudes of the exponentials (b_is).

which is the key equation of the Pisarenko method. The eigen-equation of (54) states that the elements of the eigenvector of the covariance matrix, corresponding to the smallest eigenvalue (σ^2), are the same as the coefficients in the recursive equation of x_r (coefficients of the ARMA model in (49)). Therefore, by evaluating the roots of the polynomial represented in (46) with coefficients that are the elements of this vector, we can find the tones in the spectrum.

Although we started by eigen-decomposition of $\mathbf{R_{yy}}$, we observed that only one of the eigenvectors is required; the one that corresponds to the smallest eigenvalue. This eigenvector can be found using simple approaches (in contrast to eigen-decomposition) such as power method. The PHD method is briefly shown in Table 14.

A different formulation of the PHD method with linear programming approach (refer to Section 2.2 for description of linear programming) for array processing is studied in [121]. The PHD method is shown to be equivalent to a geometrical projection problem which can be solved using ℓ_1-norm optimization.

5.3 MUSIC

MUltiple SIgnal Classification (MUSIC), is a method originally devised for high-resolution source direction estimation in the context of array processing that will be discussed in the next section [122]. The inherent equivalence of array processing and time series analysis paves the way for the employment of this method in spectral estimation. MUSIC can be understood as a generalization and improvement of the Pisarenko method. It is known that in the context of array processing, MUSIC can attain the statistical efficiency[i] in the limit of asymptotically large number of observations [11].

In the PHD method, we construct an autocorrelation matrix of dimension $k+1$ under the assumption that its smallest eigenvalue (σ^2) belongs to the noise subspace. Then we use the Hermitian property of the covariance matrix to conclude that the noise eigenvector should be orthogonal to the signal eigenvectors. In MUSIC, we extend this method using a noise subspace of dimension greater than one to improve the performance. We also use some kind of averaging over noise eigenvectors to obtain a more reliable signal estimator.

Table 14 PHD algorithm

1. Given the model order k (number of sinusoids), find the autocorrelation matrix of the noisy observations with dimension $k+1$ ($\mathbf{R_{yy}}$).
2. Find the smallest eigenvalue (σ^2) of $\mathbf{R_{yy}}$ and the corresponding eigenvector ($\mathbf{h}$).
3. Set the elements of the obtained vector as the coefficients of the polynomial in (46). The roots of this polynomial are the estimated frequencies.

The data model for the sum of exponentials plus noise can be written in the matrix form as

$$\mathbf{y}_{m\times 1} = \mathbf{A}_{m\times k}\mathbf{b}_{k\times 1} + \boldsymbol{\nu}_{m\times 1} \tag{55}$$

where the length of data is taken as $m > k$ and the elements of $\mathbf{A}$ are

$$a_{p,q} \triangleq e^{j(p-1)\omega_q} \quad \text{for} \quad 1 \le p \le m\,,\ 1 \le q \le k \tag{56}$$

where $\boldsymbol{\nu}$ represents the noise vector. Since the frequencies are different, $\mathbf{A}$ is of rank k and the first term in (55) forms a k-dimensional signal subspace, while the second term is randomly distributed in both signal and noise subspaces; i.e., unlike the first term, it is not confined to a subspace of lower dimension. The correlation matrix of the observations is given by

$$\mathbf{R} = \mathbf{A}\mathbf{b}\mathbf{b}^H\mathbf{A}^H + \sigma^2\mathbf{I} \tag{57}$$

where the noise is assumed to be white with variance σ^2. If we decompose $\mathbf{R}$ into its eigenvectors, k eigenvalues corresponding to the k-dimensional subspace of the first term of (57) are essentially greater than the remaining $m-k$ values, σ^2, corresponding to the noise subspace; thus, by sorting the eigenvalues, the noise and signal subspaces can be determined. Assume ω is an arbitrary frequency and $\mathbf{e}(\omega) = [1, e^{j\omega}, \ldots, e^{j(m-1)\omega}]$. The MUSIC method estimates the spectrum content of the signal at frequency ω by projecting the vector $\mathbf{e}(\omega)$ into the noise subspace. When the projected vector is zero, the vector $\mathbf{e}(\omega)$ falls in the signal subspace and most likely, ω is among the spectral tones. In fact, the frequency content of the spectrum is inversely proportional to the ℓ_2-norm of the projected vector:

$$P_{MU}(\omega) = \frac{1}{\mathbf{e}^H(\omega)\boldsymbol{\Pi}^{\perp}\mathbf{e}(\omega)} \tag{58}$$

$$\boldsymbol{\Pi}^{\perp} = \sum_{i=k+1}^{m} \mathbf{v}_i\mathbf{v}_i^H \tag{59}$$

where $\mathbf{v}_i$s are eigenvectors of $\mathbf{R}$ corresponding to the noise subspace.

The k peaks of $P_{MU}(\omega)$ are selected as the frequencies of the sparse signal. The determination of the number of frequencies (model order) in MUSIC is based on the MDL and Akaike information criterion (AIC) methods to be discussed in the next section. The MUSIC algorithm is briefly explained in Table 15.

Figure 19 compares the results (in the order of improved performance) for various spectral line estimation methods. The first upper figure shows the original spectral lines, and the four other figures show the results for Prony, PHD, MUSIC, and IMAT methods. We observe that the Prony method (which is similar to ELP and annihilating filter of Section 3.3 and (38)) does not yield good results due to its sensitivity to noise, while the IMAT method is the best. The application of IMAT to spectral estimation

Table 15 MUSIC algorithm

1. Find the autocorrelation matrix of the noisy observations ($\mathbf{R_{yy}}$) with the available size as shown in (57).
2. Using a given value of k or a method to determine k (such as MDL), separate the $m-k$ smallest eigenvalues of $\mathbf{R_{yy}}$ and the corresponding eigenvectors ($\mathbf{v}_{k+1},\ldots,\mathbf{v}_m$).
3. Use (58) to estimate the spectral content at frequency ω.

is a clear confirmation of our contention that we can apply tools developed in some areas to other areas for better performance.

6 Sparse array processing

There are three types of array processing: 1—estimation of multi-source location (MSL) and Direction of Arrival (DOA), 2—sparse array beam-forming and design, and 3—sparse sensor networks. The first topic is related to estimating the directions and/or the locations of multiple targets; this problem is very similar to the problem of spectral estimation dealt with in the previous section; the relations among sparsity, spectral estimation, and array processing were discussed in [123,124]. The second topic is related to the design of sparse arrays with some missing and/or random array sensors. The last topic, depending on the type of sparsity, is either similar to the second topic or related to CS of sparse signal fields in a network. In the following, we will only consider the first kind.

6.1 Array processing for MSL and DOA estimation

Among the important fields of active research in array processing are MSL and DOA estimation [122,125,126]. In such schemes, a passive or active array of sensors is used to locate the sources of narrow-band signals. Some applications may assume far-field sources (e.g., radar signal processing) where the array is only capable of DOA estimation, while other applications (e.g. biomedical imaging systems) assume near-field sources where the array is capable of locating the sources of radiation. A closely related field of study is spectral estimation due to similar linear statistical models. The stochastic sparse signals pass through a partially known linear transform (e.g., array response or inverse Fourier transform) and are observed in a noisy environment.

In the array processing context, the common temporal frequency of the source signals is known. Spatial sampling of the signal is used to extract the direction of the signal (spatial frequency). As a far-field approximation, the signal wavefronts are assumed to be planar. Consider a signal arriving with angle φ as in Figure 20. Simultaneous sampling of this wavefront on the array will exhibit a phase change of the signal from sensor to sensor. In this way, discrete samples of a complex exponential are obtained, where its frequency can be translated to the direction of the signal source. The response of a uniform linear array (ULA) to a wavefront impinging on the array from direction φ is

$$\mathbf{a}(\varphi) = [\,1,\ e^{j2\pi\frac{d}{\lambda}\sin(\varphi)},\ \ldots,\ e^{j(n-1)2\pi\frac{d}{\lambda}\sin(\varphi)}] \tag{60}$$

where d is the inter-element spacing of the array, λ is the wavelength, and n is the number of sensors in the array. When multiple sources are present, the observed vector is the sum of the response (sweep) vectors and noise. This

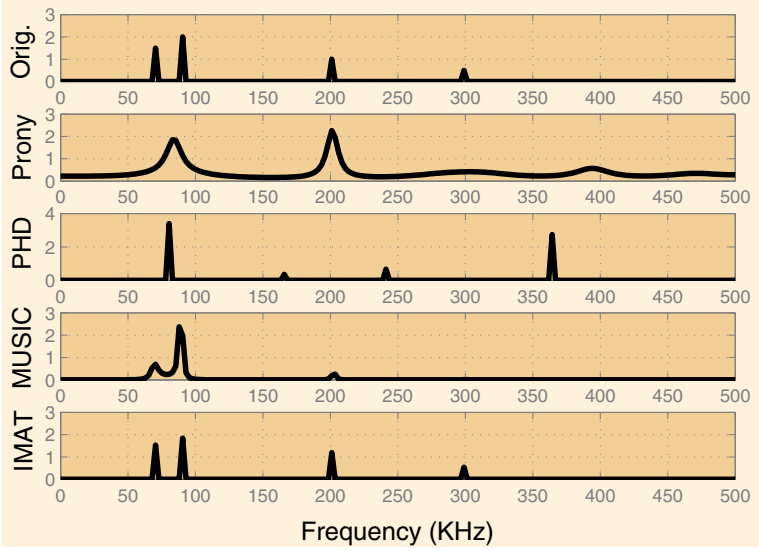

Figure 19 A comparison of various spectral estimation methods for a sparse mixture of sinusoids (the top figure) using Prony, Pisarenko, MUSIC, and IMAT methods (in the order of improved performance); input SNR is $5dB$ and 256 time samples are used.

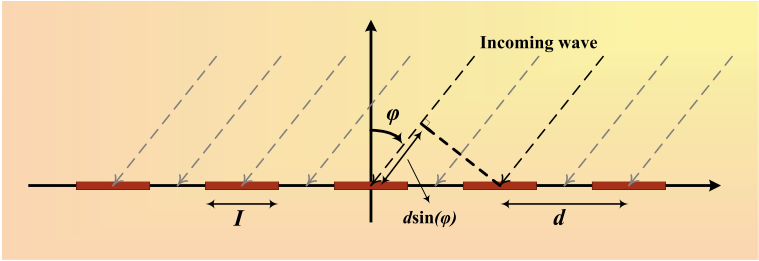

Figure 20 Uniform linear array with element distance *d*, element length *l*, and a wave arriving from direction φ.

resembles the spectral estimation problem with the difference that sampling of the array elements is not limited in time. In fact in array processing, an additional degree of freedom (the number of elements) is present; thus, array processing is more general than spectral estimation.

Two main fields in array processing are MSL and DOA for estimating the source locations and directions, respectively; for both purposes, the angle of arrival (azimuth and elevation) should be estimated while for MSL an extra parameter of range is also needed. The simplest case is the 1-D ULA (azimuth-only) for DOA estimation.

For the general case of k sources with angles $\varphi_1, \ldots, \varphi_k$ with respect to the array, the ULA response is given by the matrix $\mathbf{A}(\boldsymbol{\varphi}) = [\mathbf{a}(\varphi_1), \ \ldots, \ \mathbf{a}(\varphi_k)]$, where the vector $\boldsymbol{\varphi}$ of DOA's is defined as $\boldsymbol{\varphi} = [\varphi_1, \ \ldots, \ \varphi_k]$. In the above notation, $\mathbf{A}$ is a matrix of size $n \times k$ and $\mathbf{a}(\varphi_i)$s are column vectors. Now, the vector of observations at array elements ($\mathbf{y}[i]$) is given by

$$\mathbf{y}[i] = \mathbf{A}\mathbf{s}[i] + \boldsymbol{\nu}[i] \tag{61}$$

where the vector $\mathbf{s}[i]$ represents the multi-source signals and $\boldsymbol{\nu}[i]$ is the white Gaussian noise vector. Source signals and additive noise are assumed to be zero-mean and i.i.d. normal processes with covariance matrices $\mathbf{P}$ and $\sigma^2\mathbf{I}$, respectively. With these assumptions, the observation vector $\mathbf{y}[i]$ will also follow an n-dimensional zero-mean normal distribution with the covariance matrix

$$\mathbf{R} = E\{\mathbf{y}\mathbf{y}^H\} = \mathbf{A}\mathbf{P}\mathbf{A}^H + \sigma^2\mathbf{I} \tag{62}$$

In the field of DOA estimation, extensive research has been accomplished in (1) source enumeration, and (2) DOA estimation methods. Both of the subjects correspond to the determination of parameters k and $\boldsymbol{\varphi}$.

Although some methods are proposed for simultaneous detection and estimation of the model statistical characteristics [127], most of the literature is devoted to two-stage approaches; first, the number of active sources is detected and then their directions are estimated by techniques such as estimation of signal parameters via rotational invariance techniques (ESPRIT)[j] [128-132]. Usually, the joint detection-estimation methods outperform the two-stage approaches with the cost of higher computational complexity. In the following, we will describe Minimum Description Length (MDL) as a powerful tool to detect the number of active sources.

6.1.1 Minimum description length

One of the most successful methods in array processing for source enumeration is the use of the MDL criterion [133]. This technique is very powerful and outperforms its older versions including AIC [134-136]. Hence, we confine our discussion to MDL algorithms.

6.1.2 Preliminaries

Minimum description length is an optimum method of finding the model order and parameters for the most compressed representation of the observed data. For the purpose of statistical modeling, the MAP probability or the suboptimal criterion of ML is used; more precisely, conditioned on the observed data, the maximum probability among the possible options is found (hypotheses testing) [137]. When the model parameters are not known, the MAP and ML criteria result in the most complex approach; consider fitting a finite sequence of data to a polynomial of unknown degree [33]:

$$y(t_i) = P(t_i) + \nu(t_i), \quad i = 1, \ \ldots, \ m \tag{63}$$

where $P(t) = a_0 + a_1 t + \cdots + a_k t^k$, $\nu(t)$ is the observed Gaussian noise and k is the unknown model order (degree of the polynomial $P(t)$) which determines the complexity. Clearly, $m - 1$ is the maximum required order for unique description of the data (m observed samples), and the ML criterion always selects this maximum value ($\hat{k}_{ML} = m - 1$); i.e., the ML method forces the polynomial $P(t)$ to pass through all the points. MDL, on the other hand, yields a sparser solution ($\hat{k}_{MDL} < m - 1$).

Due to the existence of additive noise, it is quite rational to look for a polynomial with degree less than m which also takes the complexity order into account. In MDL, the idea of how to consider the complexity order is borrowed from information theory: given a specific statistical distribution, we can find an optimum source coding scheme (e.g., Huffman coding) which attains the lowest average

code length for the symbols. Furthermore, if p_s is the distribution of the source s and q_s is another distribution, we have [138]:

$$H(s) = -\int (p_s \log p_s)\mathrm{d}s \leq -\int (p_s \log q_s)\mathrm{d}s \tag{64}$$

where $H(s)$ is the entropy of the signal. This implies that the minimum average code length is obtained only for the correct source distribution (model parameters); in other words, the choice of wrong model parameters (distribution function) leads to larger code lengths. When a particular model with the set of parameters $\boldsymbol{\theta}$ is assumed for the data a priori, each time a sequence y is received, the parameters should first be estimated. The optimum estimation method is usually the ML estimator which results in $\hat{\boldsymbol{\theta}}_{ML}$. Now, the probability distribution for a received sequence y becomes $p\left(y|\hat{\boldsymbol{\theta}}_{ML}\right)$ which according to information theory, requires an average code length of $-\log\left(p\left(y|\hat{\boldsymbol{\theta}}_{ML}\left(y\right)\right)\right)$ bits. In addition to the data, the model parameters should also be encoded which in turn requires $\frac{\kappa}{2}\log(m)$ bits where κ is the number of independent parameters to be encoded in the model and m is the number of data points.[k] Thus, the *two-part MDL* selects the model that minimizes the whole required code length which is given by [139]:

$$-\log\left(p\left(y|\hat{\theta}_{ML}\right)\right) + \frac{\kappa}{2}\log(m) \tag{65}$$

The first term is the ML term for data encoding, and the second term is a penalty function that inhibits the number of free parameters of the model to become very large.

Example of using MDL in spectral estimation An example from spectral estimation can help clarify how the MDL method works (for more information refer to the previous section on spectral estimation). The mathematical formulation of the problem is as follows: If there are k (unknown) sinusoids with various frequencies, amplitudes, and phases ($3k$ unknown parameters) observed in a noisy data vector x (sampled at n distinct time slots), the maximum likelihood function for this observed data with additive Gaussian noise is as follows:

$$L(\theta_k, x) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}}\prod_{t=1}^{n} e^{-\frac{\left(x_t - \Sigma_{j=1}^{k} a_j \sin(\omega_i t+\phi_i)\right)^2}{2\sigma^2}}, \tag{66}$$

here $\theta_k = \{a_j, \omega_j, \phi_j\}_{j=1}^{k}$ are the unknown sinusoidal parameters to be estimated to compute the likelihood term in (65), which in this case is computed from (66). The $3k$ unidentified parameters are estimated by the grid search, i.e., all possible values of frequency and phase (amplitude can be estimated using the assumed frequency and phase by using this relation; $\hat{a_j} = \frac{\Sigma_t x(t)\sin(\hat{\omega_j}t+\hat{\phi})}{\Sigma_t (x(t)\sin(\hat{\omega_j}t+\hat{\phi}))^2}$ [140] are tested and the one maximizing the likelihood function (66) is selected as the best estimate.

To find the number of embedded sinusoids in the noisy observed data, it is initially assumed that $k = 0$ and (65) is calculated, then k is increased and by using the grid search, the maximum value of the likelihood for the assumed k is calculated from (66), and this calculated value is then used to compute (65). This procedure should be followed as long as (65) decreases and consequently aborted when it starts to rise. The k minimizing (65) is the k selected by MDL method and hopefully reveals the true number of the sinusoids in the noisy observed data. It is obvious that the sparsity condition, i.e., $k << n$, is necessary for the efficient operation of MDL. In addition to the number of sinusoids, MDL has apparently estimated the frequency, amplitude, and phase of the embedded sinusoids. This should make it clear why such methods are called detection–estimation algorithms.

The very same method can be used to find the number, position, and amplitude of an impulsive noise added to a low-pass signal in additive noise. If the samples of the added impulsive noise are statistically independent from each other, the high-pass samples of the discrete fourier transform (DFT) of the noisy observed data with impulsive noise should be taken and the same method applied.

MDL source enumeration In the source enumeration problem, our model is a multivariate Gaussian random process with zero mean and covariance of the type shown in (62), where the number of active sources is unknown. In some enumeration methods (other than MDL), the exact form of (62) is employed which results in high computational complexity. In the conventional MDL method, it is assumed that the model is a covariance matrix with a spherical subspace[l] of dimension $n - k$. Suppose the sample covariance matrix is

$$\hat{\mathbf{R}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_i \mathbf{x_i}^H \tag{67}$$

and assume the ordered eigenvalues of $\hat{\mathbf{R}}$ are $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_n$, while the ordered eigenvalues of the exact covariance matrix $\mathbf{R}$ are $\lambda_1 \geq \cdots \geq \lambda_k \geq \lambda_{k+1} = \cdots = \lambda_n = \sigma^2$. The normal distribution function of the received complex data $\mathbf{x}$ is [129]

$$p(\mathbf{x};\mathbf{R}) = \frac{1}{\det(\pi\mathbf{R})^m} e^{-tr\{\mathbf{R}^{-1}\hat{\mathbf{R}}\}} \tag{68}$$

where $tr(.)$ stands for the trace operator. The ML estimate of signal eigenvalues in $\mathbf{R}$ are $\hat{\lambda}_i$, $i = 1, \ldots, k$ with the respective eigenvectors $\{\hat{\mathbf{v}}_i\}_{i=1}^{k}$. Since $\lambda_{k+1} = \cdots = \lambda_n = \sigma^2$, the ML estimate of the noise eigenvalue

is $\hat{\sigma}^2_{ML} = \frac{1}{n-k}\sum_{i=k+1}^{n}\hat{\lambda}_i$ and $\{\hat{\mathbf{v}}_i\}_{i=k+1}^{n}$ are all noise eigenvectors. Thus, the ML estimate of $\mathbf{R}$ given $\hat{\mathbf{R}}$ is

$$\mathbf{R}_{ML} = \sum_{i=1}^{k}\hat{\lambda}_i\hat{\mathbf{v}}_i\hat{\mathbf{v}}_i^H + \hat{\sigma}^2_{ML}\sum_{i=k+1}^{n}\hat{\mathbf{v}}_i\hat{\mathbf{v}}_i^H \tag{69}$$

In fact, since we know that $\mathbf{R}$ has a spherical subspace of dimension $n-k$, we correct the observed $\hat{\mathbf{R}}$ to obtain $\mathbf{R}_{ML}$.

Now, we calculate $-\log\left(p(\mathbf{x}|\mathbf{R}_{ML})\right)$; it is easy to show that

$$tr\left\{\mathbf{R}_{ML}^{-1}\hat{\mathbf{R}}\right\} = n \tag{70}$$

which is independent of k and can be omitted in the minimization of (65). Thus, for the first term of (65) we only need the determinant $|\mathbf{R}_{ML}|$ which is the product of the eigenvalues, and the MDL criterion becomes

$$m\sum_{i=1}^{k}\log(\hat{\lambda}_i) + m(n-k)\log\left(\frac{1}{n-k}\sum_{i=k+1}^{n}\hat{\lambda}_i\right) + \frac{\kappa}{2}\log(m) \tag{71}$$

where κ is the number of free parameters in the distribution. This expression should be computed for different values of $0 \le k \le n-1$ and its minimum point should be $\hat{k}_{MDL}$. Note that we can subtract the term $m\sum_{i=1}^{n}\log(\hat{\lambda}_i)$ from the expression, which is not dependent on k to get the well-known MDL criterion [129]:

$$m(n-k)\log\left[\frac{\frac{1}{n-k}\sum_{i=k+1}^{n}\hat{\lambda}_i}{\prod_{i=k+1}^{n}\hat{\lambda}_i^{\frac{1}{n-k}}}\right] + \frac{\kappa}{2}\log(m) \tag{72}$$

where the first term is the likelihood ratio for the sphericity test of the covariance matrix. This likelihood ratio is a function of arithmetic and geometric means of the noise subspace eigenvalues [141]. Figure 21 is an example of MDL performance in determining the number of sources in array processing. It is evident that in low SNRs, the MDL has a strong tendency to underestimate the number of sources, while as SNR increases, it gives a consistent estimate. Also at high SNRs, underestimation is more probable than overestimation.

Now we compute the number of independent parameters (κ) in the model. Since the noise subspace is spherical, the choice of eigenvectors in this subspace can accept any arbitrary orthonormal set; i.e., no information is revealed when these vectors are known. Thus, the set of parameters is $\{\lambda_1, \ldots, \lambda_k, \sigma^2, \mathbf{v}_1, \ldots, \mathbf{v}_k\}$. The eigenvalues of a hermitian matrix (correlation matrix) are all real while the eigenvectors are normal complex vectors. Therefore, the eigenvalues (including σ^2) introduce $k+1$ degrees of freedom. The first eigenvector has $2n-2$ degrees of freedom (since its first nonzero element can be adjusted to unity), while the second, due to its orthogonality to the first eigenvector, has $2n-4$ degrees of freedom. With the same argument, it can be shown that there are $2(n-i)$ free parameters in the i^{th} eigenvector; hence

$$\kappa = 1 + k + \sum_{i=1}^{k}2(n-i) = n(2n-k)+1 \tag{73}$$

where the last integer 1 can be omitted since it is independent of k.

The two-part MDL, despite its very low computational complexity, is among the most successful methods for source enumeration in array processing. Nonetheless, this method does not reach the best attainable performance

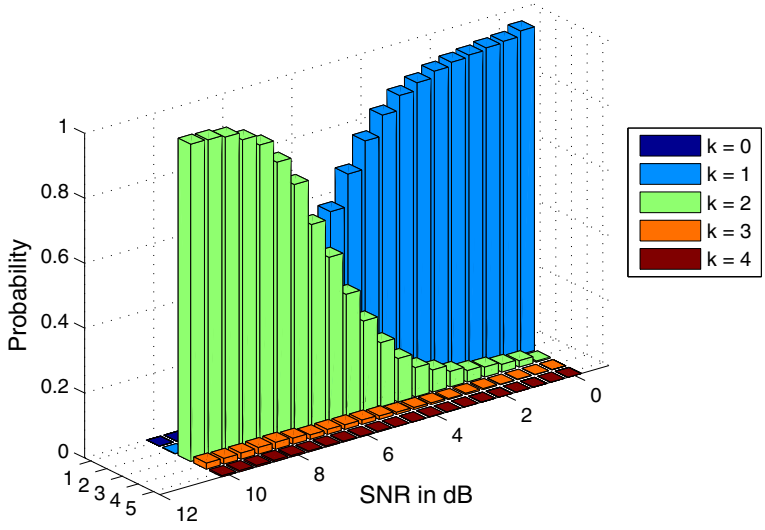

Figure 21 An MDL example; the vertical axis is the probability of order detection. And the other two axes are the number of sources and the SNR values. The MDL method estimates the number of active sources (which is 2) correctly when the SNR value is relatively high.

for finite number of measurements [142]. The new version of MDL, called *one-part* or *Refined MDL* has improved the performance for the cases of finite measurements which has not been applied to the array processing problem [33].

6.2 Sparse sensor networks

Wireless sensor networks typically consist of a large number of sensor nodes, spatially distributed over a region of interest, that observe some physical environment including acoustic, seismic, and thermal fields with applications in a wide range of areas such as health care, geographical monitoring, homeland security, and hazard detection. The way sensor networks are used in practical applications can be divided into two general categories:

(1) There exists a central node known as the fusion center (FC) that retrieves relevant field information from the sensor nodes and communication from the sensor nodes to FC generally takes place over a power- and bandwidth-constrained wireless channel.
(2) Such a central node does not exist and the nodes take specific decisions based on the information they obtain and exchange among themselves. Issues such as distributed computing and processing are of high importance in such scenarios.

In general, there are three main tasks that should be implemented efficiently in a wireless sensor network: sensing, communication, and processing. The main challenge in design of practical sensor networks is to find an efficient way of jointly performing these tasks, while using the minimum amount of system resources (computation, power, bandwidth) and satisfying the required system design parameters (such as distortion levels). For example, one such metric is the so-called energy-distortion tradeoff which determines how much energy the sensor network consumes in extracting and delivering relevant information up to a given distortion level. Although many theoretical results are already available in the case of point-to-point links in which separation between source and channel coding can be assumed, the problem of efficiently transmitting or sharing information among a vast number of distributed nodes remains a great challenge. This is due to the fact that well-developed theories and tools for distributed signal processing, communications, and information theory in large-scale networked systems are still under development. However, recent results on distributed estimation or detection indicate that joint optimization through some form of source-channel matching and local node cooperation can result in significant system performance improvement [143-147].

6.2.1 How sparsity can be exploited in a sensor network

Sparsity appears in many applications for which sensor networks are deployed, e.g., localization of targets in a large region or estimation of physical phenomena such as temperature fields that are sparse under a suitable transformation. For example, in radar applications, under a far-field assumption, the observation system is linear and can be expressed as a matrix of steering vectors [148,149]. In general, sparsity can arise in a sensor network from two main perspectives:

(1) Sparsity of node distribution in spatial terms
(2) Sparsity of the field to be estimated

Although nodes in a sensor network can be assumed to be regularly deployed in a given environment, such an assumption is not valid in many practical scenarios. Therefore, the non-uniform distribution of nodes can lead to some type of sparsity in spatial domain that can be exploited to reduce the amount of sensing, processing, and/or communication. This issue is subsequently related to extensions of the nonuniform sampling techniques to two-dimensional domains through proper interpolation and data recovery when samples are spatially sparse [34, 150]. The second scenario that provides a proper basis for exploiting the sparsity concepts arises when the field to be estimated is a sparse multi-dimensional signal. From this point of view, ideas such as those presented earlier in the context of compressed sensing (Section 3.2) provide the proper framework to address the sparsity in such fields.

Spatial sparsity and interpolation in sensor networks

Although general 2-D interpolation techniques are well-known in various branches of statistics and signal processing, the main issue in a sensor network is exploring proper spatio/temporal interpolation such that communication and processing are also efficiently accomplished. While there is a wide range of interpolation schemes (polynomial, Fourier, and least squares [151]), many of these schemes are not directly applicable for spatial interpolation in sensor networks due to their communication complexity.

Another characteristic of many sensor networks is the non-uniformity of node distribution in the measurement field. Although non-uniformity has been dealt with extensively in contexts such as signal processing, geo-spatial data processing, and computational geometry [1], the combination of irregular sensor data sampling and intra-network processing is a main challenge in sensor networks. For example, reference [152] addresses the issue of spatio-temporal non-uniformity in sensor networks and how it impacts performance aspects of a sensor network such as compression efficiency and routing overhead. In order to reduce the impact of non-uniformity, the authors in [152] propose using a combination of spatial data interpolation and temporal signal segmentation. A simple interpolation wavelet transform for irregular sampling

which is an extension of the 2-D irregular grid transform to 3-D spatio-temporal transform grids is also proposed in [153]. Such a multi-scale transform extends the approach in [154] and removes the dependence on building a distributed mesh within the network. It should be noted that although wavelet compression allows the network to trade reconstruction quality for communication energy and bandwidth usage, such energy savings are naturally offset by the overhead cost of computing the wavelet coefficients.

Distributed wavelet processing within sensor networks is yet another approach to reduce communication energy and wireless bandwidth usage. Use of such distributed processing makes it possible to trade long-haul transmission of raw data to the FC for less costly local communication and processing among neighboring nodes [153]. In addition, local collaboration among nodes decorrelates measurements and results in a sparser data set.

Compressive sensing in sensor networks Most natural phenomena in SNs are compressible through representation in a natural basis [86]. Some examples of these applications are imaging in a scattering medium [148], MIMO radar [149], and geo-exploration via underground seismic data. In such cases, it is possible to construct a highly compressed version of a given field, in a decentralized fashion. If the correlations between data at different nodes are known a-priori, it is possible to use schemes that have very favorable power-distortion-latency tradeoffs [143,155,156]. In such cases, distributed source coding techniques, such as Slepian-Wolf coding, can be used to design compression schemes without collaboration between nodes (see [155] and the references therein). Since prior knowledge of such correlations is not available in many applications, collaborative, intra-network processing and compression are used to determine unknown correlations and dependencies through information exchange between network nodes. In this regard, the concept of compressive wireless sensing has been introduced in [147] for energy-efficient estimation at the FC of sensor data, based on ideas from wireless communications [143,145,156-158] and compressive sampling theory [29,75,159]. The main objective in such an approach is to combine processing and communications in a single distributed operation [160-162].

Methods to obtain the required sparsity in a SN While transform-based compression is well-developed in traditional signal and image processing domains, the understanding of sparse transforms for networked data is not as trivial [163]. There are methods such as associating a graph with a given network, where the vertices of the graph represent the nodes of the network, and edges between vertices represent relationships among data at adjacent nodes. The structure of the connectivity is the key to obtaining effective sparse transformations for networked data [163]. For example, in the case of uniformly distributed nodes, tools such as DFT or DCT can be adopted to exploit the sparsity in the frequency domain. In more general settings, wavelet techniques can be extended to handle the irregular distribution of sampling locations [153]. There are also scenarios in which standard signal transforms may not be directly applicable. For example, network monitoring applications rely on the analysis of communication traffic levels at the network nodes where network topology affects the nature of node relationships in complex ways. Graph wavelets [164] and diffusion wavelets [165] are two classes of transforms that have been proposed to address such complexities. In the former case, the wavelet coefficients are obtained by computing the digital differences of the data at different scales. The coefficients at the first scale are differences between neighboring data points, and those at subsequent spatial scales are computed by first aggregating data in neighborhoods and then computing differences between neighboring aggregations. The resulting graph wavelet coefficients are then defined by aggregated data at different scales and computing differences between the aggregated data [164]. In the latter scheme, diffusion wavelets are based on construction of an orthonormal basis for functions supported on a graph and obtaining a custom-designed basis by analyzing eigenvectors of a diffusion matrix derived from the graph adjacency matrix. The resulting basis vectors are generally localized to neighborhoods of varying size and may also lead to sparse representations of data on a graph [165]. One example of such an approach is where the node data correspond to traffic rates of routers in a computer network.

Implementation of CS in a wireless SN Two main approaches to implement random projections in a SN are discussed in the literature [163]. In the first approach, the CS projections are simultaneously calculated through superposition of radio waves and communicated using amplitude-modulated coherent transmissions of randomly weighted values directly from the nodes in the network to the FC (Figure 22). This scheme, introduced in [147,157] and further refined in [166], is based on the notion of the so-called *matched source-channel communication* [156,157]. Although the need for complex routing, intra-network communications, and processing are alleviated, local phase synchronization among nodes is an issue to be addressed properly in this approach.

In the second approach, the projections can be computed and delivered to every subset of nodes in the network using gossip/consensus techniques, or be delivered to a single point using clustering and aggregation. This approach is typically used for networked data storage and

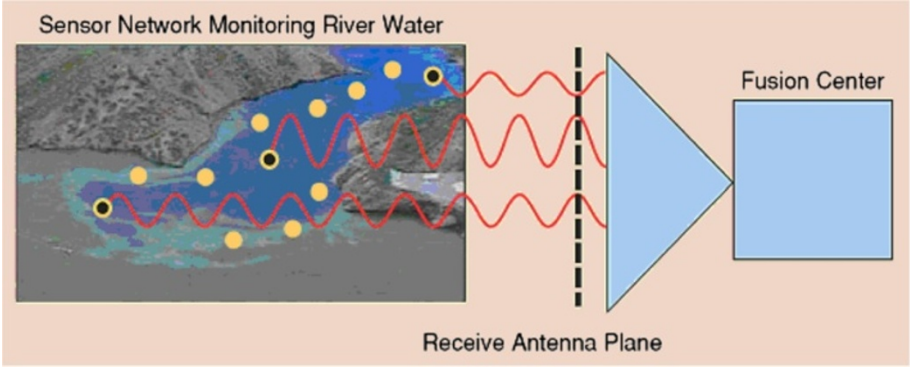

Figure 22 Computation of CS projections through superposition of radio waves of randomly weighted values directly from the nodes in the network to the FC (from [163]).

retrieval applications. In this method, computation and distribution of each CS sample is accomplished through two simple steps [163]. In the first step, each of the sensors multiplies its data with the corresponding element of the compressing matrix. Then, in the second step, the resulting local terms are simultaneously aggregated and distributed across the network using randomized gossip [167], which is a simple iterative decentralized algorithm for computing linear functions. Because each node only exchanges information with its immediate neighbors in the network, gossip algorithms are more robust to failures or changes in the network topology and cannot be easily compromised by eliminating a single server or fusion center [168].

Finally, it should be noted that in addition to the encoding process, the overall system performance is significantly affected by the decoding process [44,88,169]; this study and its extensions to sparse SNs remain as challenging tasks.

6.2.2 Sensing capacity

Despite wide-spread development of SN ideas in recent years, understanding of fundamental performance limits of sensing and communication between sensors is still under development. One of the issues that has recently attracted attention in theoretical analysis of sensor networks is the concept of sensor capacity. The sensing capacity was initially introduced for discrete alphabets in applications such as target detection [170] and later extended in [14,171,172] to the continuous case. The questions in this area are related to the problem of sampling of sparse signals, [29,76,159] and sampling with finite rate of innovation [3,103]. In the context of the CS, sensing capacity provides bounds on the maximum signal dimension or complexity per sensor measurement that can be recovered to a pre-defined degree of accuracy. Alternatively, it can be interpreted as the minimum number of sensors necessary to monitor a given region to a desired degree of fidelity based on noisy sensor measurements. The inverse of sensing capacity is the compression rate; i.e., the ratio of the number of measurements to the number of signal dimensions which characterizes the minimum rate to which the source can be compressed. As shown in [14], sensing capacity is a function of SNR, the inherent dimensionality of the information space, sensing diversity, and the desired distortion level.

Another issue to be noted with respect to the sensing capacity is the inherent difference between sensor network and CS scenarios in the way in which the SNR is handled [14,172]. In sensor networks composed of many sensors, fixed SNR can be imposed for each individual sensor. Thus, the sensed SNR per location is spread across the field of view leading to a row-wise normalization of the observation matrix. On the other hand, in CS, the vector-valued observation corresponding to each signal component is normalized by each column. This difference has led to different regimes of compression rate [172]. In SN, in contrast to the CS setting, sensing capacity is generally small and correspondingly the number of sensors required does not scale linearly with the target sparsity. Specifically, the number of measurements is generally proportional to the signal dimension and is weakly dependent on target density sparsity. This issue has raised questions on compressive gains in power-limited SN applications based on sparsity of the underlying source domain.

7 Sparse component analysis: BSS and SDR

7.1 Introduction

Recovery of the original source signals from their mixtures, without having a priori information about the sources and the way they are mixed, is called blind source separation (BSS). This process is impossible if no assumption about the sources can be made. Such an assumption on the sources may be uncorrelatedness, statistical independence, lack of mutual information, or disjointness in some space [18,19,49].

The signal mixtures are often decomposed into their constituent principal components, independent components, or are separated based on their disjoint characteristics described in a suitable domain. In the latter case, the original sources should be sparse in that domain. Independent component analysis (ICA) is often used for separation of the sources in the former case, whereas SCA is employed for the latter case. These two mathematical tools are described in the following sections followed by some results and illustrations of their applications.

7.2 Independent component analysis (ICA)

The main assumption in ICA is the statistical independence of the constituent sources. Based on this assumption, ICA can play a crucial role in the separation and denoising of signals (BSS).

There has been recent research interest in the field of BSS due to its practicality in a wide range of problems. For example, BSS of acoustic signals measured in a room is often referred to as the Cocktail Party problem, which means separation of individual sounds from a number of recordings in an echoic and noisy environment. Figure 23 illustrates the BSS concept, wherein the mixing block represents the multipath propagation model between the original sources and the microphone measurements.

Generally, BSS algorithms make assumptions about the environment in order to make the problem more tractable. There are typically three assumptions about the mixing medium. The most simple but widely used case is the instantaneous case, where the source signals arrive at the sensors at the same time. This has been considered for separation of biological signals such as the EEG where the signals have narrow bandwidths and the sampling frequency is normally low [173]. The generative model for BSS in this case can be easily formulated as

$$\mathbf{x}[i] = \mathbf{H} \cdot \mathbf{s}[i] + \boldsymbol{\nu}[i] \tag{74}$$

where $\mathbf{s}[i]$, $\mathbf{x}[i]$, and $\boldsymbol{\nu}[i]$ denote, respectively, the vector of source signals, size $n \times 1$, observed signal size $m \times 1$, and noise signal size $m \times 1$. $\mathbf{H}$ is the mixing matrix of size $m \times n$. Generally, the mixing process can be nonlinear (due to inhomogenity of the environment and that the medium can change with respect to the source signal variations; e.g., stronger vibration of a drum as a medium, with louder sound). However, in an instantaneous linear case where the above problems can be avoided or ignored, the separation is performed by means of a separating matrix, $\mathbf{W}$ of size $n \times m$, which uses only the information contained in $\mathbf{x}[i]$ to reconstruct the original source signals (or the independent components) as

$$\mathbf{y}[i] = \mathbf{W} \cdot \mathbf{x}[i] \tag{75}$$

where $\mathbf{y}[i]$ is the estimate for the source signal $s[i]$. The early approaches in instantaneous BSS started from the work by Herault and Jutten [174] in 1986. In their approach, they considered non-Gaussian sources with equal number of independent sources and mixtures. They proposed a solution based on a recurrent artificial neural network for separation of the sources.

In the cases where the number of sources is known, any ambiguity caused by false estimation of the number of sources can be avoided. If the number of sources is unknown, a criterion may be established to estimate the number of sources beforehand. In the context of model identification, this is referred to as *Model Order Selection* and methods such as the final prediction error (FPE), AIC, residual variance (RV), MDL and Hannan and Quinn (HNQ) methods [175] may be considered to solve this problem.

In acoustic applications, however, there are usually time lags between the arrival times of the signals at the sensors. The signals also may arrive through multiple paths. This type of mixing model is called a convolutive model [176]. The convolutive mixing model can also be classified into two subcategories: anechoic and echoic. In both cases, the vector representations of mixing and separating processes are modified as $\mathbf{x}[i] = \mathbf{H}[i] * \mathbf{s}[i] + \boldsymbol{\nu}[i]$ and $\mathbf{y}[i] = \mathbf{W}[i] * \mathbf{x}[i]$, respectively, where $*$ denotes the convolution operation. In an anechoic model, however, the expansion of the mixing process may be given as

$$x_r[i] = \sum_{j=1}^{n} h_{r,j} s_j[i - \delta_{r,j}] + \nu_r[i], \quad \text{for } r = 1, \ldots, m \tag{76}$$

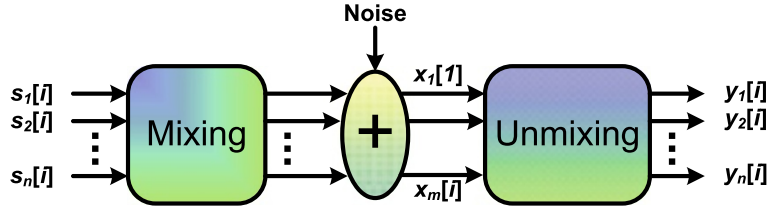

Figure 23 The BSS concept; the unobservable sources $s_1[i], \ldots, s_n[i]$ are mixed and corrupted by additive zero mean noise to generate the observations $x_1[i], \ldots, x_m[i]$. The target of BSS is to estimate an unmixing system to recover the original sources in $y_1[i], \ldots, y_n[i]$.

where the attenuation, $h_{r,j}$, and delay $\delta_{r,j}$ of source j to sensor r would be determined by the physical position of the source relative to the sensors. Then the unmixing process to estimate the sources will be given as

$$y_j[i] = \sum_{r=1}^{m} w_{j,r} x_r[i - \delta_{j,r}], \quad \text{for } j = 1, \ldots, n \tag{77}$$

where the $w_{j,r}$s are the elements of $\mathbf{W}$. In an echoic mixing environment, it is expected that the signals from the same sources reach the sensors through multiple paths. Therefore, the expansion of the mixing and separating models will be changed to

$$x_r[i] = \sum_{j=1}^{n} \sum_{l=1}^{L} h_{r,j}^{l} s_j[i - \delta_{r,j}^{l}] + v_r[i], \quad r = 1, \ldots, m \tag{78}$$

where L denotes the maximum number of paths for the sources, $v_r[i]$ is the accumulated noise at sensor r, and $(.)^l$ refers to the l^{th} path. The unmixing process will be formulated similarly to the anechoic one. For a known number of sources, an accurate result may be expected if the number of paths is known; otherwise, the overall number of observations in an echoic case is infinite.

The aim of BSS using ICA is to estimate an unmixing matrix $\mathbf{W}$ such that $\mathbf{Y} = \mathbf{WX}$ best approximates the independent sources $\mathbf{S}$, where $\mathbf{Y}$ and $\mathbf{X}$ are respectively matrices with columns $\mathbf{y}[i] = \left[y_1[i], y_2[i], \ldots, y_n[i]\right]^T$ and $\mathbf{x}[i] = [x_1[i], x_2[i], \ldots, x_m[i]]^T$. Thus the ICA separation algorithms are subject to permutation and scaling ambiguities in the output components, i.e. $\mathbf{W} = \mathbf{PDH}^{-1}$, where $\mathbf{P}$ and $\mathbf{D}$ are the permutation and scaling (diagonal) matrices, respectively. Permutation of the outputs is troublesome in places where either the separated segments of the signals are to be joined together or when a frequency-domain BSS is performed.

Mutual information is a measure of independence and maximizing the non-Gaussianity of the source signals is equivalent to minimizing the mutual information between them [177].

In those cases where the number of sources is more than the number of mixtures (underdetermined systems), the above BSS schemes cannot be applied simply because the mixing matrix is not invertible, and generally the original sources cannot be extracted. However, when the signals are sparse, the methods based on disjointness of the sources in some domain may be utilized. Separation of the mixtures of sparse signals is potentially possible in the situation where, at each sample instant, the number of nonzero sources is not more than a fraction of the number of sensors (see Table 1, row and column 6). The mixtures of sparse signals can also be instantaneous or convolutive.

7.3 Sparse component analysis (SCA)

While the independence assumption for the sources is widely exploited in the design of BSS algorithms, the possible disjointness of the sources in some domain has not been considered. In SCA, this property is directly employed. Blind source separation by sparse decomposition has been addressed by Zibulevsky and Pearlmutter [178] for both over-determined/exactly-determined and underdetermined systems using the maximum a posteriori approach. One way of formulating SCA is by representing the sources using a proper signal dictionary:

$$s_r[i] = \sum_{l=1}^{n} c_{r,l} \boldsymbol{\phi}_l[i] \tag{79}$$

where $r = 1, \ldots, m$ and n is the number of basis functions in the dictionary. The functions $\boldsymbol{\phi}_l[i]$ are called atoms or elements of the dictionary. These atoms do not have to be linearly independent and may form an overcomplete dictionary. The sparsity property requires that only a small number of the coefficients $c_{r,l}$ differ significantly from zero. Based on this definition, the mixing and unmixing systems are modeled as follows:

$$\begin{aligned} \mathbf{x}[i] &= \mathbf{A}\mathbf{s}[i] + \boldsymbol{\nu}[i] \\ \mathbf{s}[i] &= \mathbf{C}\boldsymbol{\Phi}[i] \end{aligned} \tag{80}$$

where $\boldsymbol{\nu}[i]$ is an $m \times 1$ vector. $\mathbf{A}$ and $\mathbf{C}$ can be determined by optimization of a cost function based on an exponential distribution for $c_{i,j}$ [178]. In places where the sources are sparse and at each time instant, at most one of the sources has significant nonzero value, the columns of the mixing matrix may be calculated individually, which makes the solution to the underdetermined case possible.

The SCA problem can be stated as a clustering problem since the lines in the scatter plot can be separated based on their directionalities by means of clustering. A number of works on this method have been reported [18,179,180]. In the work by Li et al. [180], the separation has been performed in two different stages. First, the unknown mixing matrix is estimated using the k-means clustering method. Then, the source matrix is estimated using a standard linear programming algorithm. The line orientation of a data set may be thought of as the direction of its greatest variance. One way is to perform eigenvector decomposition on the covariance matrix of the data, the resultant principal eigenvector, i.e., the eigenvector with the largest eigenvalue, indicates the direction of the data, since it has the maximum variance. In [179], GAP statistics as a metric which measures the distance between the total variance and cluster variances, has been used to estimate the number of sources followed by a similar method to Li's algorithm explained above. In line with this approach, Bofill and Zibulevsky [15] developed a potential function method for estimating the mixing matrix followed by

ℓ_1-norm decomposition for the source estimation. Local maxima of the potential function correspond to the estimated directions of the basis vectors. After the mixing matrix is identified, the sources have to be estimated. Even when **A** is known, the solution is not unique. So, a solution is found for which the ℓ_1-norm is minimized. Therefore, for $\mathbf{x}[i] = \sum \mathbf{a}_j s_j[i]$, $\sum_j |s_j|$ is minimized using linear programming.

Geometrically, for a given feasible solution, each source component is a segment of length $|s_j|$ in the direction of the corresponding $\mathbf{a}_j$ and, by concatenation, their sum defines a path from the origin to $x[i]$. Minimizing $\sum_j |s_j|$ amounts therefore to finding the shortest path to $x[i]$ over all feasible solutions $j = 1, \ldots, n$, where n is the dimension of space of the independent basis vectors [18]. Figure 24 shows the scatter plot and the shortest path from the origin to the data point $x[i]$.

There are many cases for which the sources are disjoint in other domains, rather than the time-domain, or when they can be represented as sum of the members of a dictionary which can consist for example of wavelets or wavelet packets. In these cases the SCA can be performed in those domains more efficiently. Such methods often include transformation to time-frequency domain followed by a binary masking [181] or a BSS followed by binary masking [176]. One such approach, called degenerate unmixing estimation technique (DUET) [181], transforms the anechoic convolutive observations into the time-frequency domain using a short-time Fourier transform and the relative attenuation and delay values between the two observations are calculated from the ratio of corresponding time-frequency points. The regions of significant amplitudes (atoms) are then considered to be the source components in the time-frequency domain. In this method only two mixtures have been considered and as a major limit of this method, only one source has been considered active at each time instant.

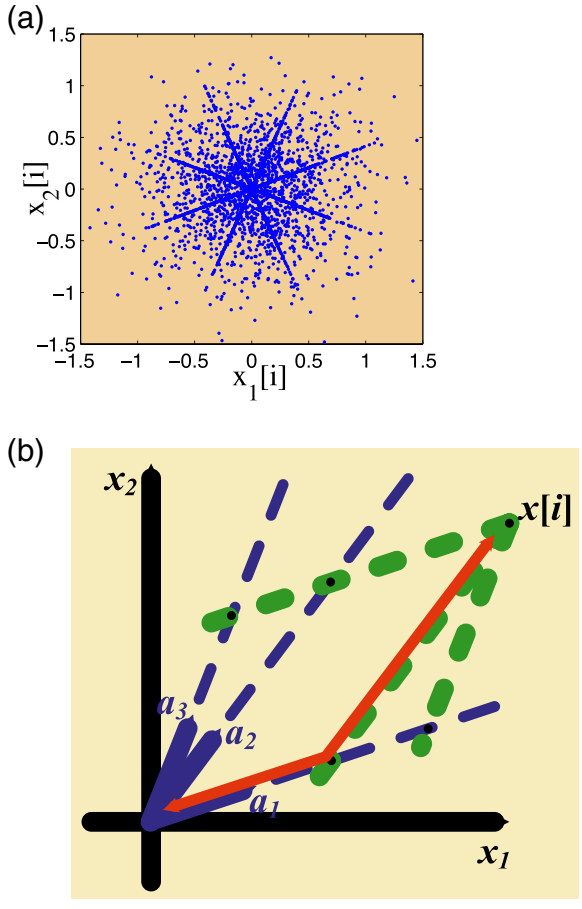

Figure 24 Measurement points for data structures consisting of multiple lower dimensional subspaces. (a) the scatter plot and **(b)** the shortest path from the origin to the data point, $x[i]$, extracted from [15].

For instantaneous separation of sparse sources, the common approach used by most researchers is to attempt to maximize the sparsity of the extracted signals at the output of the separator. The columns of the mixing matrix **A** assign each observed data point to only one source based on some measure of proximity to those columns [182], i.e., at each instant only one source is considered active. Therefore the mixing system can be presented as:

$$x_r[i] = \sum_{j=1}^{n} a_{j,r} s_j[i] \quad r = 1, \ldots, m \tag{81}$$

where in an ideal case, $a_{j,r} = 0$ for $r \neq j$. Minimization of the ℓ_1-norm is one of the most logical methods for estimation of the sources as long as the signals can be considered sparse. ℓ_1-norm minimization is a piecewise linear operation that partially assigns the energy of $\mathbf{x}[i]$ to the m columns of **A** around $\mathbf{x}[i]$ in $\mathbb{R}^n$ space. The remaining $n-m$ columns are assigned zero coefficients, therefore the ℓ_1-norm minimization can be manifested as:

$$\min \|\mathbf{s}[i]\|_{\ell_1} \text{ subject to } \mathbf{A} \cdot \mathbf{s}[i] = \mathbf{x}[i] \tag{82}$$

A detailed discussion of signal recovery using ℓ_1-norm minimization is presented by Takigawa et al. [183] and described below. As mentioned above, it is important to choose a domain that sparsely represents the signals.

On the other hand, in the method developed by Pedersen et al. [176], as applied to stereo signals, the binary masks are estimated after BSS of the mixtures and then applied to the microphone signals. The same technique has been used for convolutive sparse mixtures after the signals are transformed to the frequency domain.

In another approach [184], the effect of outlier noise has been reduced using median filtering then hybrid fast ICA filtering, and ℓ_1-norm minimization have been used for separation of temporomandibular joint sounds. It has

been shown that for such sources, this method outperforms both DUET and Li's algorithms. The authors of [185] have recently extended the DUET algorithm to separation of more than two sources in an echoic mixing scenario in the time-frequency domain.

In a very recent approach, it has been considered that brain signal sources in the space-time frequency domain are disjoint. Therefore, clustering the observation points in the space-time-frequency-domain can be effectively used for separation of brain sources [186].

As it can be seen, generally, BSS exploits independence of the source signals, whereas SCA benefits from the disjointness property of the source signals in some domain. While the BSS algorithms mostly rely on ICA with statistical properties of the signals, SCA uses their geometrical and behavioral properties. Therefore, in SCA, either a clustering approach or a masking procedure can result in estimation of the mixing matrix. Often, an ℓ_1-norm is used to recover the source signals. Generally, in places where the source signals are sparse, the SCA methods often result in more accurate estimation of the signals with less ambiguities in the estimation.

7.4 SCA algorithms

There are three main steps for the solution of an SCA problem as shown in Table 16 [187]. The first step of Table 16 shows a linear model for the SCA problem, the second step consists of estimating the mixing matrix $\mathbf{A}$ using sparsity information, and finally the third step is to estimate the sparse source representation based on the estimate of $\mathbf{A}$ [17].

A brief review of major approaches that are suggested for the third step was given in Section 2.

7.5 Sparse dictionary representation (SDR) and signal modeling

A signal $\mathbf{x} \in \mathbb{R}^n$ may be sparse in a given basis but not sparse in a different basis. For example, an image may be sparse in a wavelet basis (i.e., most of the wavelet coefficients are small) even though the image itself may not be sparse (i.e., many of the gray values of the image are relatively large). Thus, given a class $\mathcal{S} \subset \mathbb{R}^n$, an important problem is to find a basis or a frame in which all signals in $\mathcal{S}$ can be represented sparsely. More specifically, given a class of signals $\mathcal{S} \subset \mathbb{R}^n$, it is important to find a basis (or a frame) $D = \{w_j\}_{j=1}^d$ (if it exists) for $\mathbb{R}^n$ such that every data vector $\mathbf{x} \in \mathcal{S}$ can be represented by at most $k \ll n$ linear combinations of elements of D. The dictionary design problem has been addressed in [18-20,40,75,190]. A related problem is the signal modeling problem in which the class $\mathcal{S}$ is to be modeled by a union of subspaces $\mathcal{M} = \bigcup_{i=1}^l V_i$ where each V_i is a subspace of $\mathbb{R}^n$ with the dimension of $V_i \leq k$ where $k \ll n$ [49]. If the subspaces V_i are known, then it is possible to pick a basis $E^i = \{e_j^i\}_j$ for each V_i and construct a dictionary $D = \bigcup_{i=1}^l E^i$ in which every signal of $\mathcal{S}$ has sparsity k (or is almost k sparse). The model $\mathcal{M} = \bigcup_{i=1}^l V_i$ can be found from an observed set of data $F = \{f_1, \ldots, f_m\} \subset \mathcal{S}$ by solving (if possible) the following non-linear least squares problem:

Find subspaces $V_1, \ldots, V_l$ of $\mathbb{R}^n$ that minimize the expression

$$e(F, \{V_1, \ldots, V_l\}) = \sum_{i=1}^{m} \min_{1 \leq j \leq l} d^2(f_i, V_j) \tag{83}$$

over all possible choices of l subspaces with dimension of $V_i \leq k < n$. Here d denotes the Euclidian distance in $\mathbb{R}^n$ and k is an integer with $1 \leq k < n$ for $i = 1, \ldots, l$. Note that $e(F, \{V_1, \ldots, V_l\})$ is calculated as follows: for each $f_i \in F$ and fixed $\{V_1, \ldots, V_l\}$, the subspace $V_j \in \{V_1, \ldots, V_l\}$ closest to f_i is found and the distance $d^2(f_i, V_j)$ is computed. This process is repeated for all $f_i \in F$ and the squares of the distances are added together to find $e(F, \{V_1, \ldots, V_l\})$. The optimal model is then obtained as the union $\mathcal{M} = \bigcup_i V_i^o$, where $\{V_1^o, \ldots, V_l^o\}$ minimize the expression (83). When $l = 1$ this problem reduces to the classical least squares problem. However, when $l > 1$ the set $\bigcup_i V_i$ is a nonlinear set and the problem is fully non-linear (see Figure 25). A more general nonlinear least squares problem has been studied for finite and infinite Hilbert spaces [49]. In that general setting, the existence of solutions is proved and a meta-algorithm for searching for the solution is described.

For the special finite dimensional case of $\mathbb{R}^n$ in (83), the search algorithm is an iterative algorithm that alternates between data partition and the optimization of a simpler least squares problem. This algorithm, which is equivalent to the k-means algorithm, is summarized in Table 17.

In some new attempts sparse representation and the compressive sensing concept have been extended to solving multichannel source separation [191-194]. In [191,192] separation of sparse sources with different

Table 16 SCA steps

1. Consider the model $\mathbf{x} = \mathbf{A} \cdot \mathbf{s}$; we need a linear transformation that applies to both sides of the equation to yield a new sparse source vector.
2. Estimate the mixing matrix $\mathbf{A}$. Several approaches are presented for this step, such as natural gradient ICA approaches, and clustering techniques with variants of k-means algorithm [18,187].
3. Estimate the source representation based on the sparsity assumption. A majority of proposed methods are primarily based on minimizing some norm or pseudo-norm of the source representation vector. The most effective approaches are Matching Pursuit [38,187], Basis Pursuit, [85,178,188,189], FOCUSS [46], IDE [73] and Smoothed ℓ_0-norm [47].

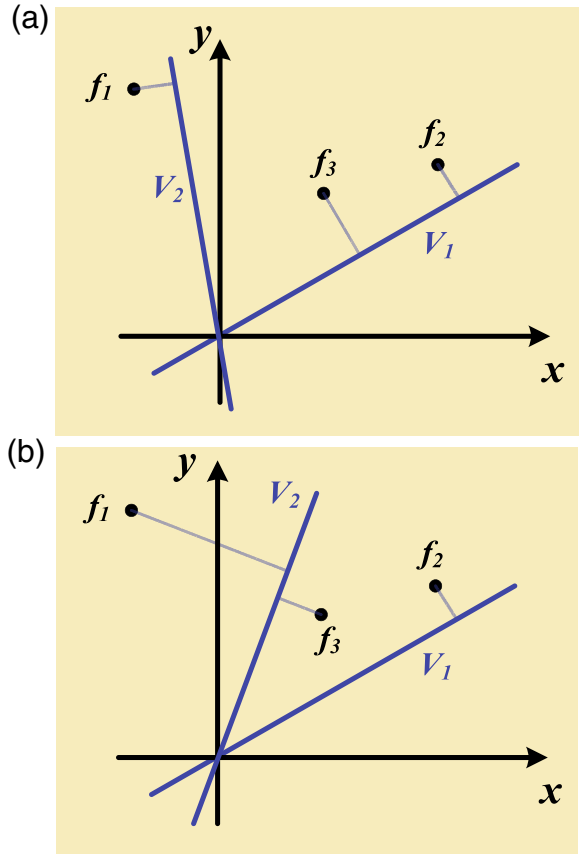

Figure 25 Objective function. (a) $e = d^2(f_1, V_2) + d^2(f_2, V_1) + d^2(f_3, V_1)$ and **(b)** $e = d^2(f_1, V_2) + d^2(f_3, V_2) + d^2(f_2, V_1)$. Configuration of V_1, V_2 in a) creates the partition $P_1 = \{f_1\}$ and $P_2 = \{f_2, f_3\}$ while the configuration in (b) causes the partition $P_1 = \{f_1, f_3\}$ and $P_2 = \{f_2\}$.

morphologies has been presented by developing a multichannel morphological component analysis approach. In this scheme, the signals are considered as combination of features from different dictionaries. Therefore, different dictionaries are assumed for different sources. In [193] inversion of a random field from pointwise measurements collected by a sensor network is presented. In this article, it is assumed that the field has a sparse representation in a known basis. To illustrate the approach, the inversion of an acoustic field created by the superposition of a discrete number of propagating noisy acoustic sources is considered. The method combines compressed sensing (sparse reconstruction by ℓ_1-constrained optimization) with distributed average consensus (mixing the pointwise sensor measurements by local communication among the sensors). [194] addresses source separation from a linear mixture under source sparsity and orthogonality of the mixing matrix assumptions. A two-stage separation process is proposed. In the first stage recovering a sparsity pattern of the sources is tried by exploiting the orthogonality prior. In the second stage, the support is used to reformulate the recovery task as an optimization problem. Then a solution based on alternating minimization for solving the above problems is suggested.

8 Multipath channel estimation

In wireless systems, channel estimation is required for the compensation of channel distortions. The transmitted signal reflects off different objects and arrives at the receiver from multiple paths. This phenomenon causes the received signal to be a mixture of reflected and scattered versions of the transmitted signal. The mobility of the transmitter, receiver, and scattering objects results in rapid changes in the channel response, and thus the channel estimation process becomes more complicated. Due to the sparse distribution of scattering objects, a multipath channel is sparse in the time domain as shown in Figure 26. By taking sparsity into consideration, channel estimation can be simplified and/or made more accurate. The sparse time varying multipath channel is modeled as

$$h(t,\tau) = \sum_{l=0}^{k-1} \alpha_l(t)\delta(\tau - \tau_l(t)) \tag{84}$$

where k is the number of taps, α_l is the l^{th} complex path gain, and τ_l is the corresponding path delay. At time t, the transfer function is given by

$$H(t,f) = \int_{-\infty}^{+\infty} h(t,\tau)e^{-j2\pi f\tau}\mathrm{d}\tau \tag{85}$$

Table 17 Search algorithm

- **Input**:
  - initial partition $\{F_1^1, \ldots, F_l^1\}$
  - Data set $\mathcal{F}$
- **Iterations**:
  1. Use the SVD to find $\{V_1^1, \ldots, V_l^1\}$ by minimizing $e\left(F_i^1, V_i^1\right)$ for each i, and compute $\Gamma_1 = \sum_i e\left(F_i^1, V_i^1\right)$;
  2. Set $j = 1$;
  3. **While** $\Gamma_j = \sum_i e\left(F_i^j, V_i^j\right) > e\left(\mathcal{F}, \{V_1^j, \ldots, V_l^j\}\right)$
  4. Choose a new partition $\left\{F_1^{j+1}, \ldots, F_l^{j+1}\right\}$ that satisfies, $f \in F_k^{j+1}$ implies that $d\left(f, V_k^j\right) \leq d\left(f, V_h^j\right), h = 1, \ldots, l$;
  5. Use SVD to find and choose $\{V_1^{j+1}, \ldots, V_l^{j+1}\}$, by minimizing $e\left(F_i^{j+1}, V_i\right)$ for each i, and compute $\Gamma_{j+1} = \sum_i e\left(F_i^{j+1}, V_i^{j+1}\right)$;
  6. Increment j by 1, i.e., $j \rightarrow j + 1$;
  7. **End while**
- **Output**:
  - $\{F_1^j, \ldots, F_l^j\}$ and $\{V_1^j, \ldots, V_l^j\}$.

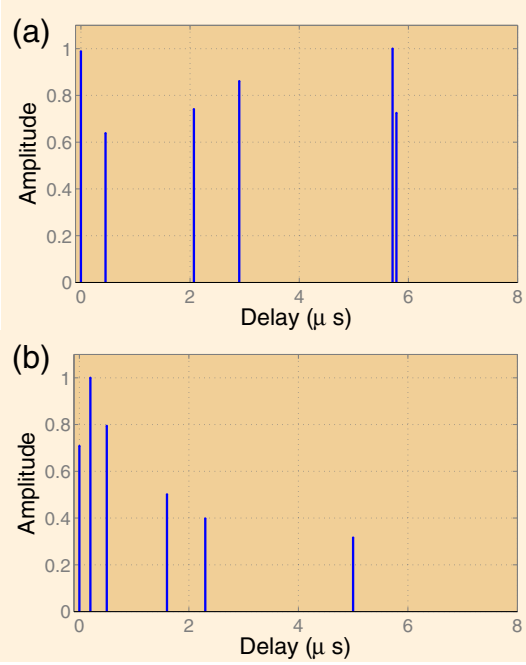

Figure 26 The impulse response of two typical multipath channels. (a) Brazil-D and **(b)** TU6 channel profiles.

The estimation of the multipath channel impulse response is very much similar to the determination of analog epochs and amplitudes of discontinuities for finite rate of innovation as shown in (31). Essentially, if a known train of impulses is transmitted and the received signal from the multipath channel is filtered and sampled (information domain as discussed in Section 3.3), the channel impulse response can be estimated from these samples using an annihilating filter (the Prony or ELP method) [27] defined with the $\mathcal{Z}$-transform and a pseudo-inverse matrix inversion, in principle.[m] Once the channel impulse response is estimated, its effect is compensated; this process can be repeated according to the dynamics of the time varying channel.

A special case of multipath channel is an OFDM channel, which is widely used in ADSL, DAB, DVB, WLAN, WMAN, and WIMAX.[n] OFDM is a digital multicarrier transmission technique where a single data stream is transmitted over several sub-carrier frequencies to achieve robustness against multipath channels as well as spectral efficiency [195]. Channel estimation for OFDM is relatively simple; the time instances of channel impulse response is now quantized and instead of an annihilating filter defined in the $\mathcal{Z}$-transform, we can use DFT and ELP of Section 4.1. Also, instead of a known train of impulses, some of the available sub-carriers in each transmitted symbol are assigned to predetermined patterns, which are usually called comb-type pilots. These pilot tones help the receiver to extract some of the DFT samples of the discrete time varying channel (84) at the respective frequencies in each transmitted symbol. These characteristics make the OFDM channel estimation similar to unknown sparse signal recovery of Section 3.1.1 and the impulsive noise removal of Section 4.1.2. Because of these advantages, our main example and simulations are related to OFDM channel estimation.

8.1 OFDM channel estimation

For OFDM, the discrete version of the time varying channel of (85) in the frequency domain becomes

$$H[r,i] \triangleq H(rT_f, i\Delta f) = \sum_{l=0}^{n-1} h[r,l]\, e^{-\frac{j2\pi il}{n}} \tag{86}$$

where

$$h[r,l] = h(rT_f, lT_s) \tag{87}$$

where T_f and n are the symbol length (including cyclic prefix) and number of sub-carriers in each OFDM symbol, respectively. Δf is the sub-carrier spacing, and $T_s = \frac{1}{\Delta f}$ is the sample interval. The above equation shows that for the r^{th} OFDM symbol, $H[r,i]$ is the DFT of $h[r,l]$.

Two major methods are used in the equalization process [196]: (1) zero forcing and (2) minimun mean squared error (MMSE). In the zero forcing method, regardless of the noise variance, equalization is obtained by dividing the received OFDM symbol by the estimated channel frequency response; while in the MMSE method, the approximation is chosen such that the MSE of the transmitted data vector $\left(E\left[\|\mathbf{X}-\hat{\mathbf{X}}\|^2\right]\right)$ is minimized, which introduces the noise variance in the equations.

8.1.1 Statement of the problem

The goal of the channel estimation process is to obtain the channel impulse response from the noisy values of the channel transfer function in the pilot positions. This is equivalent to solving the following equation for $\mathbf{h}$.

$$\tilde{\mathbf{H}}_{i_p} = \mathbf{F}_{i_p}\mathbf{h} + \boldsymbol{\nu}_{i_p} \tag{88}$$

where i_p is an index vector denoting the pilot positions in the frequency spectrum, $\tilde{\mathbf{H}}_{i_p}$ is a vector containing the noisy value of the channel frequency spectrum in these pilot positions and $\mathbf{F}_{i_p}$ denotes the matrix obtained from taking the rows of the DFT matrix pertaining to the pilot positions. $\boldsymbol{\nu}_{i_p}$ is the additive noise on the pilot points in the frequency domain. Thus, the channel estimation problem is equivalent to finding the sparse vector $\mathbf{h}$ from the above set of equations for a set of pilots. Various channel estimation methods [197] have been used with the usual tradeoffs of optimality and complexity. The least square (LS) [197], ML [198], MMSE [199-201], and Linear Minimum Mean Squared Error (LMMSE) [198,199,202]

techniques are among some of these methods. However, none of these techniques use the inherent sparsity of the multipath channel **h**, and thus, they are not as accurate.

8.1.2 Sparse OFDM channel estimation

In the following, we present two methods that utilize this sparsity to enhance the channel estimation process.

CS-based channel estimation The idea of using time-domain sparsity in OFDM channel estimation has been proposed by [203-205]. There are two main advantages in including the sparsity constraint of the channel impulse response in the estimation process:

(1) Decrease in the MSE: By applying the sparsity constraint, the energy of the estimated channel impulse response will be concentrated into a few coefficients while in the conventional methods, we usually observe a leakage of the energy to the neighboring coefficients of the nonzero taps. Thus, if the sparsity-based methods succeed in estimating the support of the channel impulse response, the MSE will be improved by prevention of the leakage effect.
(2) Reduction in the overhead: The number of pilot sub-carriers is in fact, the number of (noisy) samples that we obtain from the channel frequency response. Since the pilot sub-carriers do not convey any data, they are considered as the overhead imposed to enhance the estimation process. The theoretical results in [203] indicate that by means of sparsity-based methods, the perfect estimation can be achieved with an overhead proportional to the number of non-zero channel taps (which is considerably less than that of the current standards).

In the sequel, we present two iterative methods which exploit the inherent sparsity of the channel impulse response to improve the channel estimation task in OFDM systems.

8.1.3 Iterative method with adaptive thresholding (IMAT) for OFDM channel estimation [206]

Here we apply a similar iterative method as in Section 4.2 for the channel estimation problem in (88). The main goal is to estimate **h** from $\tilde{\mathbf{H}}_{i_p}$ given that **h** has a few non-zero coefficients. To obtain an initial estimate $\hat{\mathbf{h}}_0$, we use the Moore-Penrose pseudo-inverse of $\mathbf{F}_{i_p}$ which yields a solution with minimum ℓ_2-norm:

$$\begin{aligned}\hat{\mathbf{h}}_0 &= \mathbf{F}_{i_p}^{+}\tilde{\mathbf{H}}_{i_p} = \mathbf{F}_{i_p}^{+}\mathbf{F}_{i_p}\mathbf{h} + \mathbf{F}_{i_p}^{+}\boldsymbol{v}_{i_p} \\ &= \underbrace{\frac{1}{N}\mathbf{F}_{i_p}^{H}\mathbf{F}_{i_p}}_{\mathbf{G}_{N\times N}} h + \frac{1}{N}\mathbf{F}_{i_p}{}^{H}\boldsymbol{v}_{i_p} \end{aligned} \tag{89}$$

where we used

$$\mathbf{F}_{i_p}{}^{+} = \mathbf{F}_{i_p}^{H}\underbrace{(\mathbf{F}_{i_p}\mathbf{F}_{i_p}^{H})^{-1}}_{\frac{1}{N}\mathbf{I}_{Np\times Np}} = \frac{1}{N}\mathbf{F}_{i_p}^{H}. \tag{90}$$

The non-zero coefficients of **h** are found through a set of iterations followed by adaptively decreasing thresholds:

$$\tilde{\mathbf{h}}_i = \lambda(\hat{\mathbf{h}}_0 - \mathbf{G}\cdot\hat{\mathbf{h}}_{i-1}) + \hat{\mathbf{h}}_{i-1}, \tag{91}$$

$$\hat{h}_i(k) = \begin{cases} \tilde{h}_i(k) & |\tilde{h}_i(k)| > \beta e^{-\alpha i} \\ 0 & \text{otherwise} \end{cases}, \tag{92}$$

where λ and i are the relaxation parameter and the iteration number, respectively, k is the index of channel impulse response and $\mathbf{G} = \frac{1}{N}\mathbf{F}_{i_p}^{H}\mathbf{F}_{i_p}$ is defined in (89). The block diagram of the proposed channel estimation method is shown in Figure 27.

8.1.4 Modified IMAT (MIMAT) for OFDM channel estimation [23]

In this method, the spectrum of the channel is initially estimated using a simple interpolation method such as linear interpolation between pilot sub-carriers. This initial estimate is further improved in a series of iterations between time (sparse) and frequency (information) domains to find the sparsest channel impulse response by using an adaptive thresholding scheme; in each iteration, after finding the locations of the taps (locations with previously estimated amplitudes higher than the threshold), their respective amplitudes are again found using the MMSE criterion. In each iteration, due to thresholding, some of the false taps that are noise samples with amplitudes above the threshold are discarded. Thus, the new iteration starts with a lower number of false taps. Moreover, because of the MMSE estimator, the valid taps approach their actual values in each new iteration. In the last iteration, the actual taps are detected and the MMSE estimator gives their respective values. This method is

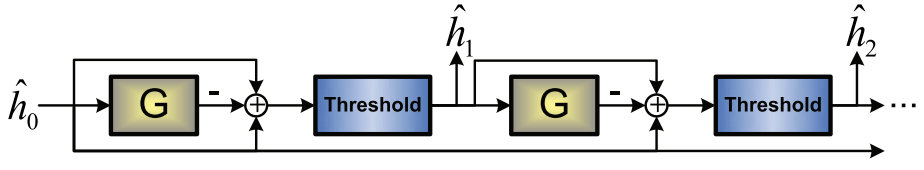

Figure 27 Block diagram of the IMAT method.

similar to RDE and IDE methods discussed in Sections 2.6 and 4.1.2. The main advantage of this method is its robustness against side-band zero-padding.[o]

Table 18 summarizes the steps in the MIMAT algorithm. In the threshold of the MIMAT algorithm, α and β are constants which depend on the number of taps and initial powers of noise and channel impulses. In the first iteration, the threshold is a small number, and with each iteration it is gradually increased. Intuitively, this gradual increase of the threshold with the iteration number, results in a gradual reduction of false taps (taps that are created due to noise). In each iteration, the tap values are obtained from

$$\hat{\mathbf{H}}_{LSi_p} = \mathbf{H}_{i_p} + \boldsymbol{\nu}_{i_p} = \tilde{\mathbf{F}} \cdot \mathbf{h}_t + \boldsymbol{\nu}_{i_p} \tag{93}$$

where t denotes the index of nonzero impulses obtained from the previous step and $\tilde{\mathbf{F}}$ is obtained from $\mathbf{F}_{i_p}$ by keeping the columns determined by t. The amplitudes of nonzero impulses can be obtained from simple iterations, pseudo-inverse, or the MMSE equation (94) of Table 18 that yields better results under additive noise environments.

The equation that has to be solved in (93) is usually over-determined which helps the suppression of the noise in each iteration step. Note that the solution presented in (94) represents a variant of the MMSE solution when the location of discrete impulses are known. If further statistical knowledge is available, this solution can be modified and a better estimation is obtained; however, this makes the approximation process more complex. This algorithm does not need many steps of iterations; the positions of the non-zero impulses are perfectly detected in three or four iterations for most types of channels.

Table 18 MIMAT algorithm for OFDM channel estimation

- **Initialization**:
  - Find an initial estimate of the time domain channel using linear interpolation: $\hat{h}^{(0)} = \hat{h}_{linear}$
- **Iterations**:
  1. Set Threshold$= \beta e^{\alpha i}$.
  2. Using the threshold from the previous step, find the locations of the taps t by thresholding the time domain channel from the previous iteration($\hat{h}^{(i-1)}$).
  3. Solve for the values of the non-zero impulses using MMSE:

$$\hat{h}_t = SNR \cdot \tilde{\mathbf{F}}^H (\tilde{\mathbf{F}} \cdot SNR \cdot \tilde{\mathbf{F}}^H + \mathbf{I})^{-1} \tag{94}$$

  4. Find the new estimate of the channel ($\hat{h}^{(i)}$) by substituting the taps in their detected positions.
  5. Stop if the estimated channel is close enough to the previous estimation or when a maximum number of iterations is reached.

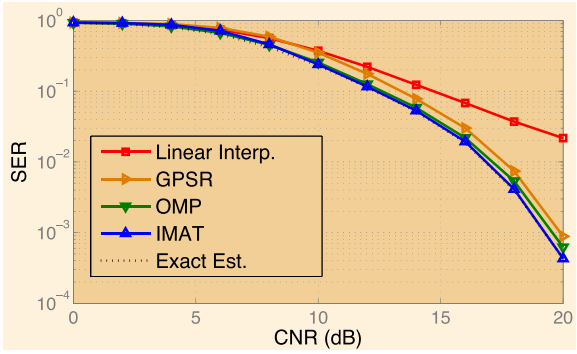

Figure 28 SER vs. CNR for the ideal channel, linear interpolation, GPSR, OMP, and the IMAT for the Brazil channel at $Fd = 0$ without zeropadding effect.

8.2 Simulation results and discussions

For OFDM simulations, the DVB-H standard was used with the 16-QAM constellation in the 2K mode (2^{11} FFT size). The channel profile was the Brazil channel D. Figures 28, 29, 30, and 31 show the symbol error rate (SER) versus the carrier-to-noise ratio (CNR) after equalizing using different sparse reconstruction methods such as orthogonal matching pursuit (OMP) [88], compressive sampling matching pursuit (CoSaMP) [41], gradient projection for sparse reconstruction (GPSR) [44], IMAT and MIMAT. Also the standard linear interpolation in the frequency domain using the noisy pilot samples is simulated. In these simulations, we have considered the effects of zero-padding and Doppler frequency in the SER of estimation. As can be seen in Figures 28, 29, 30, and 31, the SER obtained from the sparsity-based algorithms reveal almost perfect approximation of the hypothetical ideal channel (where the exact channel frequency response is used for equalization).

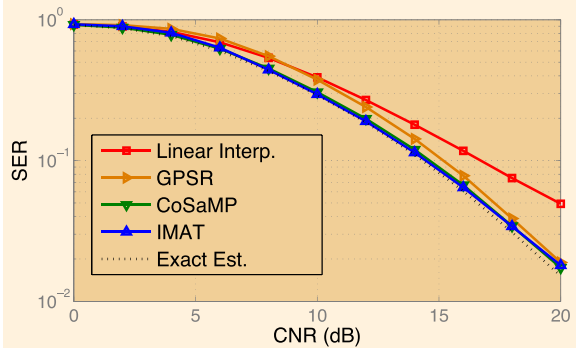

Figure 29 SER vs. CNR for the ideal channel, linear interpolation, GPSR, CoSaMP, and the IMAT for the Brazil channel at $Fd = 50$ Hz without zeropadding effect.

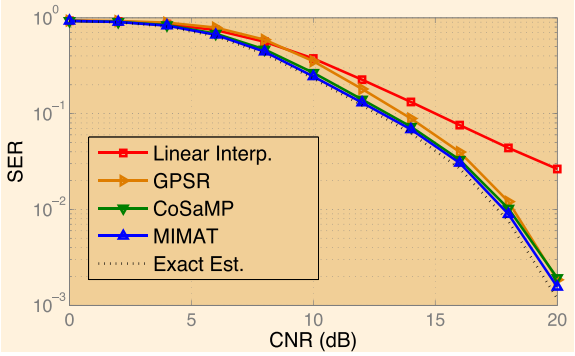

Figure 30 SER vs. CNR for the ideal channel, linear interpolation, GPSR, CoSaMP and the MIMAT for the Brazil channel at $Fd = 0$ including zeropadding effect.

9 Conclusion

A unified view of sparse signal processing has been presented in tutorial form. The sparsity in the key areas of sampling, coding, spectral estimation, array processing, component analysis, and channel estimation has been carefully exploited. Some form of uniform or random sampling has been shown to underpin the associated sparse processing methods used in each of these fields. The reconstruction methods used in each application domain have been introduced and the interconnections among them have been highlighted.

This development has revealed, for example, that the iterative methods developed for random sampling can be applied to real-field block and convolutional channel coding for impulsive noise (salt-and-pepper noise in the case of images) removal, SCA, and channel estimation for orthogonal frequency division multiplexing systems. These iterative reconstruction methods have been shown to be naturally extendable to spectral estimation and sparse array processing due to their similarity to channel coding in terms of mathematical models with significant improvements. Conversely, the minimum description length method developed for spectral estimation and array processing has potential for application in other areas. The error locator polynomial method developed for channel coding has, moreover, been shown to be a discrete version of the annihilating filter used in sampling with a finite rate of innovation and the Prony method in spectral estimation; the Pisarenko and MUSIC methods are further improvements of the Prony method when additive noise is also considered.

Linkages with emergent areas such as compressive sensing and channel estimation have also been considered. In addition, it has been suggested that the linear programming methods developed for compressive sensing and SCA can be applied to other applications with possible reduction of sampling rate. As such, this tutorial has provided a route for new applications of sparse signal processing to emerge, which can potentially reduce computational complexity and improve performance quality. Other potential applications of sparsity are in the areas of sensor networks and sparse array design.

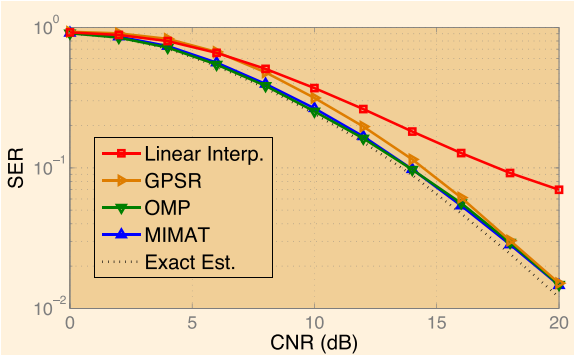

Figure 31 SER versus CNR for the ideal channel, linear interpolation, GPSR, OMP, and the MIMAT for the Brazil channel at $Fd = 50$ Hz including zeropadding effect.

Endnotes

[a]Sparse Signal Processing, Panel Session organized and chaired by F. Marvasti and lectured by Profs. E. Candes, R. G. Baraniuk, P. Marziliano, and Dr. A. Cichoki, ICASSP 2008, Las Vegas, May 2008.

[b]A list of acronyms is given in Table 2 at the end of this section.

[c]The sequence of vectors $\{\mathbf{v}_n\}$ is called a Riesz basis if there exist scalars $0 < A \leq B < \infty$ such that for every absolutely summable sequence of scalars $\{a_n\}$, we have the following inequalities [207]:

$$A\left(\sum_n |a_n|^2\right) \leq \left\|\sum_n a_n \mathbf{v}_n\right\|_{\ell_2}^2 \text{ leq} B\left(\sum_n |a_n|^2\right).$$

[d]Note that the Strang-Fix condition can also be used for an exponential polynomial assuming the delta functions are non-uniformly periodic; in that case τ_r in equation (35) is similar to$E$, the DFT of the impulses, as defined in Appendices 1 and 2.

[e]We call the set of indices of consecutive zeros syndrome positions and denote it by Θ; this set includes the complex conjugate part of the Fourier domain.

[f]The kernel of SDFT is $\exp\left(\frac{2\pi j}{n} i\, q\right)$, where q is relatively prime w.r.t. n; this is equivalent to a sorted version of DFT coefficients according to a *mod* rule, which is a kind of structured interleaving pattern.

[g]This has some resemblance to soft decision iteration for turbo codes [109].

[h]Similar to array processing to be discussed in the next

section, we can resolve any closely spaced sources conditioned on (1) limited snapshots and infinite SNR, or (2) limited SNR and infinite number of observations, while the spatial aperture of the array is kept finite.

[i]Statistical efficiency of an estimator means that it is asymptotically unbiased and its variance goes to zero.

[j]The array in ESPRIT is composed of sensor doublets with the same displacement. The parameters of the impinging signals can be estimated via a rotational invariant property of the signal subspace. The complexity and storage of ESPRIT is less than MUSIC; it is also less vulnerable to array imperfections. ESPRIT, unlike MUSIC results in an unbiased DOA estimate; nonetheless, MUSIC outperforms ESPRIT, in general.

[k]For a video introduction to these concepts, please refer to http://videolectures.net/icml08_grunwald_mdl.

[l]Spherical subspace implies the eigenvalues of the autocorrelation matrix are equal in that subspace.

[m]Similar to Pisarenko method for spectral estimation in Section 5.2.

[n]These acronyms are defined in Table 2 at the end of Section 1.

[o]In current OFDM standards, a number of subcarriers at both edges of the bandwith are set to zero to ease the process of analog bandpass filtering.

Appendix 1
ELP decoding for erasure channels [59]

For lost samples, the polynomial locator for the erasure samples is

$$H(z_i) = \prod_{m=1}^{k} \left(z_i - \mathrm{e}^{j\frac{2\pi \cdot i_m}{n}} \right) = \sum_{t=0}^{k} h_t \, z_i^{k-t} \tag{95}$$

$$H(z_{i_m}) = 0, \quad m = 1, 2, \ldots, k \tag{96}$$

where $z_i = \mathrm{e}^{j\frac{2\pi \cdot i}{n}}$. The polynomial coefficients $h_t, \quad t = 0, \ldots, k$ can be found from the product in (95); it is easier to find h_t by obtaining the inverse FFT of $H(z)$. Multiplying (96) by $e[\, i_m] \cdot \left(z_{i_m}\right)^r$ (where r is an integer) and summing over m, we get

$$\sum_{t=0}^{k} h_t \cdot \sum_{m=1}^{k} \left(e[\, i_m] \cdot \left(z_{i_m}\right)^{k+r-t} \right) = 0 \tag{97}$$

Since the inner summation is the DFT of the missing samples $e[\, i_m]$, we get

$$\sum_{t=0}^{k} h_t \cdot E[\, k + r - t] = 0 \tag{98}$$

where $E[\, .]$ is the DFT of $e[\, i]$. The received samples, $d[\, i]$, can be thought of as the original over-sampled signal, $x[\, i]$, minus the missing samples $e[\, i_m]$. The error signal, $e[\, i]$, is the difference between the corrupted and the original over-sampled signal and hence is equal to the values of the missing samples for $i = i_m$ and is equal to zero otherwise. In the frequency domain, we have

$$E[\, j] = X[\, j] - D[\, j], \quad j = 1, \ldots, n \tag{99}$$

Since $X[\, j] = 0$ for $j \in \Theta$ (see the footnote on page 40), then

$$E[\, j] = -D[\, j], \quad j \in \Theta \tag{100}$$

The remaining values of $E[\, j]$ can be found from (98), by the following recursion:

$$E[\, r] = \frac{-1}{h_k} \sum_{t=1}^{k} h_{k-t} E[\, r + t] \tag{101}$$

where $r \notin \Theta$ and the index additions are in $mod(n)$.

Appendix 2
ELP decoding for impulsive noise channels [31,104]

For all integer values of r such that $r \in \Theta$ and $r + k \in \Theta$, we obtain a system of k equations with $k+1$ unknowns (h_t coefficients). These equations yield a unique solution for the polynomial with the additional condition that the first nonzero h_t is equal to one. After finding the coefficients, we need to determine the roots of the polynomial in (95). Since the roots of $H(z)$ are of the form $\mathrm{e}^{j\frac{2\pi \cdot i_m}{n}}$, the inverse DFT (IDFT) of the $\{h_m\}_{m=0}^{k}$ can be used. Before performing IDFT, we have to pad $n - 1 - k$ zeros at the end of the $\{h_m\}_{m=0}^{k}$ sequence to obtain an n-point signal. We refer to the new signal (after IDFT) as $\{H_i\}_{i=0}^{n-1}$. Each zero in $\{H_i\}$ represents an error in $r[\, i]$ at the same location.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
We would like to sincerely thank our colleagues for their specific contributions in various sections in this article. Especially, Drs. S. Holm from University of Oslo who contributed a section in sparse array design, M. Nouri Moghadam, the director of the Newton Foundation, H. Saeedi from University of Massachusetts, and K. Mehrany from EE Dept. of Sharif University of Technology who contributed to various sections of the original paper before revision. We are also thankful to M. Valiollahzadeh who edited and contributed to the SCA section. We are especially indebted to Prof. B. Sankur from Bogazici University in Turkey for his careful review and comments. We are also thankful to the students of the Multimedia Lab and members of ACRI at Sharif University of Technology for their invaluable help and simulations. We are specifically indebted to A. Hosseini, A. Rashidinejad, R. Eghbali, A. Kazerouni, V. Montazerhodjat, S. Jafarzadeh, A. Salemi, M. Soltanalian, M. Sharif and H. Firouzi. The work of Akram Aldroubi was supported in part by grant NSF-DMS 0807464.

Author details
[1]Electrical Engineering Department, Advanced Communication Research Institute (ACRI), Sharif University of Technology, Tehran, Iran. [2]Department of Electical Engineering, Iran University of Science and Technology, Tehran, Iran. [3]Math Department, Vanderbilt University, Nashville, USA. [4]Department of Computing, University of Surrey, Surrey, UK. [5]Electrical and Electronic Department, Loughborough University, Loughborough, UK.

Received: 30 July 2011 Accepted: 23 November 2011
Published: 22 February 2012

References

1. F Marvasti, *Nonuniform Sampling: Theory and Practice*. (Springer, New York, 2001)
2. RG Baraniuk, A lecture on compressive sensing. IEEE Signal Process. Mag. **24**(4), 118–121 (2007)
3. M Vetterli, P Marziliano, Blu T, Sampling signals with finite rate of innovation. IEEE Trans. Signal Process. **50**(6), 1417–1428 (2002)
4. S Lin, DJ Costello, *Error Control Coding*. (Prentice-Hall, Englewood Cliffs, 1983)
5. T Richardson, R Urbanke, *Modern Coding Theory*. (Cambridge University Press, Cambridge, 2008)
6. F Marvasti, M Hung, MR Nakhai, in *Proc. IEEE Int. Conf. Acoust, Speech Signal Proc. ICASSP'99*, vol. 5. The application of walsh transform for forward error correction (Phoenix, AZ, USA, 1999), pp. 2459–2462
7. SL Marple, *Digital Spectral Analysis*. (Prentice-Hall, Englewood Cliffs, 1987)
8. SM Kay, SL Marple, *Spectrum analysis—a modern perspective. Proc. IEEE (Modern Spectrum Analysis II) 69(11)*, ((1981))
9. SM Kay, *Modern Spectral Estimation: Theory and Application*. (Prentice-Hall, Englewood Cliffs, 1988)
10. P Stoica, RL Moses, *Introduction to Spectral Analysis*. (Upper Saddle River, Prentice-Hall, 1997)
11. P Stoica, A Nehorai, maximumlikelihoodandCramer-Raobound Music. IEEE Trans. ASSP. **37**(5), 720–741 (1989)
12. P Stoica, A Nehorai, Performance study of conditional and unconditional direction-of-arrival estimation. IEEE Trans. ASSP. **38**(10), 1783–1795 (1990)
13. S Holm, A Austeng, K Iranpour, JF Hopperstad, in *Nonuniform Sampling: Theory and Practice*, ed. by F Marvasti. Sparse sampling in array processing (Springer, New York, 2001), pp. 787–833
14. S Aeron, M Zhao, V Saligrama, in *in Asilomar Conference on Signals, Systems and Computers, ACSSC'06*. Fundamental tradeoffs between sparsity,sensing diversity,sensing capacity (Oct–Nov, Pacific Grove,CA, 2006), pp. 295–299
15. P Bofill, M Zibulevsky, Underdetermined blind source separation using sparse representations. Signal Process Elsevier. **81**(11), 2353–2362 (2001)
16. MA Girolami, JG Taylor, in *Self-Organising Neural Networks:Independent Component Analysis and Blind Source Separation* (Springer, London, 1999)
17. P Georgiev, F Theis, A Cichocki, Sparse component analysis and blind source separation of underdetermined mixtures. IEEE Trans. Neural Netw. **16**(4), 992–996 (2005)
18. M Aharon, M Elad, AM Bruckstein, The k-svd: an algorithm for designing of overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. **54**(11), 4311–4322 (2006)
19. M Aharon, M Elad, AM Bruckstein, On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them. Linear Algebra Appl. **416**(1), 48–67 (2006)
20. R Gribonval, M Nielsen, Sparse representations in unions of bases. IEEE Trans. Inf. Theory. **49**(12), 3320–3325 (2003)
21. P Fertl, G Matz, in *Proc. Asil. Conf. Sig. Sys. and Computers*. Efficient OFDM channel estimation in mobile environments based on irregular sampling (Pacific Grove, US, May 2007), pp. 1777–1781
22. O Ureten, N Serinken, in *IEEE conf. on Vehicular Technol. Conf. (VTC)*. Decision directed iterative equalization of OFDM symbols using non-uniform interpolation (Ottawa, Canada, Sep. 2007), pp. 1–5
23. M Soltanolkotabi, A Amini, F Marvasti, in *Proc. EUSIPCO'09*. OFDM channel estimation based on adaptive thresholding for sparse signal detection (Glasgow, Scotland, Aug 2009), pp. 1685–1689
24. JL Brown, Sampling extentions for multiband signals. IEEE Trans. Acoust. Speech Signal Process. **33**, 312–315 (1985)
25. OG Guleryuz, Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising, parts I and II. IEEE Trans. Image Process. **15**(3), 539–571 (2006)
26. H Rauhut, On the impossibility of uniform sparse reconstruction using greedy methods. Sampl Theory Signal Image Process. **7**(2), 197–215 (2008)
27. T Blu, P Dragotti, M Vetterli, P Marziliano, P Coulot, Sparse sampling of signal innovations: Theory, algorithms, and performance bounds. IEEE Signal Process. Mag. **25**(2) (2008)
28. F Marvasti, Guest editor's comments on special issue on nonuniform sampling. Sampl Theory Signal Image Process. **7**(2), 109–112 (2008)
29. D Donoho, Compressed sensing. IEEE Trans. Inf. Theory. **52**(4), 1289–1306 (2006)
30. E Candès, in *Int. Congress of Mathematics*, vol. 3. Compressive sampling (Madrid, Spain, 200), pp. 1433–1452
31. S Zahedpour, S Feizi, A Amini, M Ferdosizadeh, F Marvasti, Impulsive noise cancellation based on soft decision and recursion. IEEE Trans. Instrum. Meas. **58**(8), 2780–2790 (2009)
32. S Feizi, S Zahedpour, M Soltanolkotabi, A Amini, F Marvasti, *Salt and pepper noise removal for images*. (St. Petersburg, Russia, June 2008), pp. 1–5
33. PD Grunwald, IJ Myung, MA Pitt, *Advances in Minimum Description Length: Theory and Applications*. (MIT Press, Cambridge, 2005)
34. A Kumar, P Ishwar, K Ramchandran, in *Acoustics, Speech, and Signal Processing*, vol. 3. On distributed sampling of bandlimited and non-bandlimited sensor fields (Berkeley, CA, USA, Aug 2004), pp. 925–928
35. P Fertl, G Matz, in *Proc. Acoustics, Speech and Sig. Proc., (ICASSP)*, vol. 3. Multi-user channel estimation in OFDMA uplink systems based on irregular sampling and reduced pilot overhead (Honolulu, US, April 2007), pp. 297–300
36. F Marvasti, Spectral analysis of random sampling and error free recovery by an iterative method. Trans. IECE Jpn. **69**(2), 79–82 (1986)
37. RA DeVore, Deterministic constructions of compressed sensing matrices. J. Complex. **23**(4–6), 918–925 (2007)
38. SG Mallat, Z Zhang, Matching pursuits with time-frequency dictionaries. IEEE Trans. Signal Process. **41**(12), 3397–3415 ((1993))
39. R Gribonval, P Vandergheynst, On the exponential convergence of matching pursuit in quasi-incoherent dictionaries. IEEE Trans Inf. Theory. **52**(1), 255–261 (2006)
40. JA Tropp, Greed is good: Algorithmic results for sparse approximation. IEEE Trans. Inf. Theory. **50**(10), 2231–2242 (2004)
41. D Needell, JA Tropp, Cosamp: iterative signal recovery from incomplete and inaccurate samples. App. Comp. Harmon. Anal. **26**, 301–321 (2009)
42. DL Donoho, M Elad, V Temlyakov, Stable recovery of sparse overcomplete representations in the presence of noise. IEEE Trans. Inf. Theory. **52**(1), 6–18 ((2006))
43. MS Bazaraa, JJ Jarvis, HD Sherali, *Linear programming and Network Flows*. (Wiley, New York, 1990)
44. M Figueiredo, R Nowak, S Wright, Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. IEEE J. Sel. Top. Signal Process. **1**(4), 586–597 (2007)
45. I Daubechies, MD Friese, CD Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun. Pure Appl Math. **57**, 1413–1457 (2004)
46. IF Gorodnitsky, BD Rao, Sparse signal reconstruction from limited data using FOCUSS, a re-weighted minimum norm algorithm. IEEE Trans. Signal Process. **45**(3), 600–616 (1997)
47. GH Mohimani, M Babaie-Zadeh, C Jutten, A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0-norm. IEEE Trans Signal Process. **57**(1), 289–301 (2009)
48. F Marvasti, AK Jain, Zero crossings,bandwidth compression and restoration of nonlinearly distorted band-limited signals. J. Opt Soc. Am. **3**(5), 651–654 (1986)
49. A Aldroubi, C Cabrelli, U Molter, Optimal non-linear models for sparsity and sampling. J Fourier Anal. Appl. (Special Issue on Compressed Sampling). **14**(6), 48–67 (2008)
50. A Amini, F Marvasti, Convergence analysis of an iterative method for the reconstruction of multi-band signals from their uniform and periodic nonuniform samples. STSIP. **7**(2), 109–112 (2008)
51. PJSG Ferreira, in *Nonuniform Sampling: Theory and Practice*, ed. by F Marvasti. Iterative and non-iterative recovery of missing samples for 1-D band-limited signals (Springer, Boston, 2001), pp. 235–281
52. PJSG Ferreira, The stability of a procedure for the recovery of lost samples in band-limited signals. IEEE Trans. Signal Process. **40**(3), 195–205 (1994)
53. F Marvasti, *A Unified Approach to Zero-crossings and Nonuniform Sampling of Single and Multi-dimensional Signals and Systems*. (Nonuniform Publication, Oak Park, 1987)
54. F Marvasti, *Nonuniform Sampling*. (RJ Marks, ed.) (Springer, New York, 1993)
55. AI Zayed, PL Butzer, in *Nonuniform Sampling: Theory and Practice*, ed. by F Marvasti. Lagrange interpolation and sampling theorems (Springer, formerly Kluwer Academic/Plenum Publishers, New York, 2001), pp. 123–168

56. F Marvasti, P Clarkson, M Dokic, U Goenchanart, C Liu, Reconstruction of speech signals with lost samples. IEEE Trans. Signal Process. **40**(12), 2897–2903 (1992)
57. M Unser, Sampling-50 years after Shannon. Proc. IEEE. **88**(4), 569–587 (April 2000)
58. F Marvasti, in *Nonuniform Sampling: Theory and Practice*, ed. by F Marvasti. Random topics in nonuniform sampling (Springer, formerly Kluwer Academic/Plenum Publishers, 2001), pp. 169–234
59. F Marvasti, M Hasan, M Eckhart, S Talebi, Efficient algorithms for burst error recovery using FFT and other transform kernels. IEEE Trans. Signal Process. **47**(4), 1065–1075 (1999)
60. PJSG Ferreira, Mathematics for multimedia signal processing II: Discrete finite frames and signal reconstruction. Signal Process Multimed IOS press. **174**, 35–54 (1999)
61. F Marvasti, M Analoui, M Gamshadzahi, Recovery of signals from nonuniform samples using iterative methods. IEEE Trans. ASSP. **39**(4), 872–878 (1991)
62. H Feichtinger, K Gr¨ôchenig, in *Wavelets- Mathematics and Applications*, ed. by JJ Benedetto, M Frazier. Theory and practice of irregular sampling (CRC Publications, Boca Raton, 1994), pp. 305–363
63. PJSG Ferreira, Noniterative and fast iterative methods for interpolation and extrapolation. IEEE Trans. Signal Process. **42**(11), 3278–3282 (1994)
64. A Aldroubi, K Grôchenig, Non-uniform sampling and reconstruction in shift-invariant spaces. SIAM Rev. **43**(4), 585–620 (2001)
65. A Aldroubi, Non-uniform weighted average sampling exact reconstruction in shift-invariant and wavelet spaces. Appl. Comput. Harmon. Anal. **13**(2), 151–161 (2002)
66. A Papoulis, C Chamzas, Detection of hidden periodicities by adaptive extrapolation. IEEE Trans. Acoust. Speech Signal Process. **27**(5), 492–500 (1979)
67. WYXu C Chamzas, An improved version of Papoulis-Gerchberg algorithm on band-limited extrapolation. IEEE Trans. Acoust. Speech Signal Process. **32**(2), 437–440 (1984)
68. PJSG Ferreira, Interpolation and the discrete Papoulis-Gerchberg algorithm. IEEE Trans. Signal Process. **42**(10), 2596–2606 (1994)
69. K Gröchenig, T Strohmer, in *Nonuniform Sampling: Theory and Practice* , ed. by F Marvasti. Numerical and theoretical aspects of no (Springer, formerly Kluwer Academic/Plenum Publishers, New York, 2001), pp. 283–324
70. DC Youla, Generalized image restoration by the method of alternating orthogonal projections. IEEE Trans. Circuits Syst. **25**(9), 694–702 (1978)
71. DC Youla, H Webb, Image restoration by the method of convex projections: Part 1-theory. IEEE Trans. Med. Imag. **1**(2), 81–94 (1982)
72. K Gröchenig, Acceleration of the frame algorithm. IEEE Trans. Signal Process. **41**(12), 3331–3340 (1993)
73. A Ali-Amini, M Babaie-Zadeh, C Jutten, A New Approach for Sparse Decomposition and Sparse Source Separation (EUSIPCO2006,Florence) (2006)
74. F Marvasti, in *Nonuniform Sampling: Theory and Practice*, ed. by F Marvasti. Applications to error correction codes (Springer, New York, 2001), pp. 689–738
75. E Candes, J Romberg, T Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory. **52**(2), 489–509 (2006)
76. E Candes, T Tao, Near-optimal signal recovery from random projections: universal encoding strategies. IEEE Trans. Inf. Theory. **52**(12), 5406–5425 (2006)
77. Y Tsaig, D Donoho, Extensions of compressed sensing. Signal Process. **86**(3), 549–571 (2006)
78. AJ Jerri, The Shannon sampling theorem-its various extension and applications: a tutorial review. Proc IEEE. **65**(11), 1565–1596 (1977)
79. E Candes, M Wakin, An introduction to compressive sampling. IEEE Signal Process. Mag. **25**(2), 21–30 (2008)
80. Y Eldar, Compressed sensing of analog signals in shift-invariant spaces. IEEE Trans. Signal Process. **57**(8), 2986–2997 (2009)
81. E Candes, J Romberg, Sparsity and incoherence in compressive sampling. Inverse Probl. **23**, 969–985 (2007)
82. D Donoho, X Hou, Uncertainty principle and ideal atomic decomposition. IEEE Trans.Inf. Theory. **47**(7), 2845–2862 (2001)
83. RG Baraniuk, M Davenport, R DeVore, M Wakin, A simple proof of the restricted isometry property for random matrices. Constr Approx. **28**(3), 253–263 (2008)
84. AC Gilbert, MJ Strauss, JA Tropp, R Vershynin, in *Proc. Allerton conf. Comm., Control and Comp*. Algorithmic linear dimension reduction in the ℓ_1 norm for sparse vectors (IL, USA, 2006)
85. DL Donoho, For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. **59**, 797–829 (2004)
86. JA Tropp, Recovery of short linear combinations via ℓ_1 minimization. IEEE Trans. Inf. Theory. **90**(4), 1568–1570 (2005)
87. E Candes, T Tao, Decoding by linear programming. IEEE Trans. Inf. Theory. **51**, 4203–4215 (2005)
88. J Tropp, A Gilbert, Signal recovery from partial information via orthogonal matching pursuit. IEEE Trans. Inf. Theory. **53**(12), 4655–4666 (2007)
89. D Donoho, J Tanner, Precise undersampling theorems. Proc IEEE. **98**(6), 913–924 (2010)
90. E Candes, J Romberg, Quantitative robust uncertainty principles and optimally sparse decompositions. Found. Comput. Math. **6**(2), 227–254 (2006)
91. E Candes, J Romberg, T Tao, Stable signal recovery from incomplete and inaccurate measurements. Commun Pure Appl. Math. **59**(8), 1207–1223 (2006)
92. V Saligrama, Deterministic designs with deterministic gaurantees:Toepliz compressed sensing matrices, sequence design and system identification. arXiv. **0806**, 4958 (2008)
93. Ev Berg, MP Friedlander, G Hennenfent, F Herrmann, R Saab, Ö Yılmaz, Sparco: a testing framework for sparse reconstruction, Dept. Comp. Sci. Univ. Br. Columbia, Vancouver. Tech. Rep TR-2007-20 (October 2007)
94. R Berinde, AC Gilbert, P Indyk, H Karloff, MJ Strauss, in *Proc. Allerton conf. Comm., Control and Comp*. Combining geometry and combinatorics:a unified approach to sparse signal recovery (IL, US, 2008). pp. 798–805
95. MF Duarte, MB Wakin, RG Baraniuk, in *Proc. SPARS05*, vol. 1. Fast reconstruction of piecewise smooth signals from random projections (Rennes, France, Nov 2005), pp. 1064–1070
96. AC Gilbert, Y Kotidis, S Muthukrishnan, MJ Strauss, One-pass wavelet decompositions of data streams. IEEE Trans. Knowl. Data Eng. **15**(3), 541–554 (2003)
97. AC Gilbert, MJ Strauss, JA Tropp, R Vershynin, in *ACM STOC*. One sketch for all: fast algorithms for compressed sensing (CA, San Diego, June 2007), pp. 237–246
98. S Sarvotham, D Baron, RG Baraniuk, in *Technical Report ECE-0601*. Compressed sensing reconstruction via belief propagation (ECE Dept., Rice University, July 2006). http://dsp.rice.edu/sites/dsp.rice.edu/files/cs/csbpTR07.142006.pdf
99. S Sarvotham, D Baron, RG Baraniuk, in *IEEE ISIT*. Sudocodes-fast measurement and reconstruction of sparse signals (Seattle, USA, 2006), pp. 2804–2808
100. W Xu, B Hassibi, in *IEEE Inf. Theory Workshop, ITW'07*. Efficient compressive sensing with deterministic guarantees using expander graphs (Tokyo, Japan, Sep. 2007), pp. 414–419
101. A Aldroubi, H Wang, K Zaringhalam, *Sequential compressed sampling via Huffman codes*, (2008)
102. PL Dragotti, M Vetterli, T Blu, Sampling moments and reconstructing signals of finite rate of innovation:Shannon meets strang-fix. IEEE Trans. Signal Process. **55**(5), 1741–1757 (2007)
103. I Maravic, M Vetterli, Sampling and reconstruction of signals with finite rate of innovation in the presence of noise. IEEE Trans. Signal Process. **53**(8), 2788–2805 (2005)
104. P Azmi, F Marvasti, Robust decoding of DFT-based error-control codes for impulsive and additive white gaussian noise channels. IEEE Proc. Commun. **152**(3), 265–271 (2005)
105. F Marvasti, M Nafie, Sampling theorem: a unified outlook on information theory, block and convolutional codes. Spec. Issue Info. Theory Appl., IEICE Trans. Fundam. Electron. Commun. Comput. Sci. Sec. E. **76**(9), 1383–1391 (1993)
106. J Wolf, Redundancy, the discrete Fourier transform, and impulse noise cancellation. IEEE Trans. Commun. **31**(3), 458–461 (1983)
107. RE Blahut, Transform techniques for error control codes. IBM J. Res. Dev. **23**(3), 299–315 (1979)

108. T Jr Marshall, Coding of real-number sequences for error correction: a digital signal processing problem. IEEE J. Sel. Areas Commun. **2**(2), 381–392 (2002)
109. C Berrou, A Glavieux, P Thitimajshima, in *Proc. Int. Conf. Comm. (ICC)*, vol. 2. Near Shannon limit error- correcting coding and decoding:Turbo codes (Geneva, Switzerland, 1993), pp. 1064–1070
110. CN Hadjicostis, GC Verghese, Coding approaches to fault tolerance in linear dynamic systems. IEEE Trans. Inf. Theory. **51**(1), 210–228 (2005)
111. CJ Anfinson, FT Luk, A linear algebraic model of algorithm-based fault tolerance. IEEE Trans. Comput. **37**(12), 1599–1604 (1988)
112. VSS Nair, JA Abraham, Real-number codes for fault-tolerant matrix operations on processor arrays. IEEE Trans. Comput. **39**(4), 426–435 (1990)
113. ALN Reddy, P Banerjee, Algorithm-based fault detection for signal processing applications. IEEE Trans. Comput. **39**(10), 1304–1308 (1990)
114. JMN Vieira, PJSG Ferreira, in *Proc. of ICASSP'97*, vol. 3. Interpolation, spectrum analysis, error-control coding, and fault tolerant computing (Munich, Germany, Apr 1997), pp. 1831–1834
115. M Nafie, F Marvasti, Implementation of recovery of speech with missing samples on a DSP chip. Electron Lett. **30**(1), 12–13 (1994)
116. A Momenai, S Talebi, Improving the stability of DFT error recovery codes by using sparse oversampling patterns. Elsevier. **87**(6), 1448–1461 (2007)
117. F Marvasti, Error concealment of speech, image and video signals. U.S Patents 6,601, 206 (July 2003)
118. C Wong, F Marvasti, W Chambers, Implementation of recovery of speech with impulsive noise on a DSP chip. Electron Lett. **31**(17), 1412–1413 (1995)
119. D Mandelbaum, On decoding of Reed-Solomon codes. IEEE Trans. Inf. Theory. **17**(6), 707–712 (1971)
120. Prony de B G R, Essai éxperimental et analytique: sur les lois de la dilatabilité de fluides élastique et sur celles de la force expansive de la vapeur de l'alkool, á différentes températures. J. l'École. Polytech. **1**, 24–76 (1795)
121. JJ Fuchs, Extension of the Pisarenko method to sparse linear arrays. IEEE Trans. Signal Process. **45**(10), 2413–2421 (1997)
122. R Schmidt, Multiple emitter location and signal parameter estimation. IEEE Trans. Antennas Propag. **34**(3), 276–280 (1986)
123. JJ Fuchs, On the application of the global matched filter to DOA estimation with uniform circular array. IEEE Trans. Signal Process. **49**(4), 702–709 (2001)
124. JJ Fuchs, in *Proc. IEEE Int. Conf. Acoustics Speech Signal Proc., ICASSP'96, vol. 6,*. Linear programming in spectral estimation. Application to array processing (Atlanta, GA, USA, 7–10, May 1996), pp. 3161–3164
125. H Krim, M Viberg, Two decades of array signal processing research: the parametric approach. IEEE Signal Process. Mag. **13**(4), 67–94 (1996)
126. BDV Veen, KM Buckley, Beamforming: a versatile approach to spatial filtering. IEEE ASSP Mag. **5**(2), 4–24 (1988)
127. S Valaee, P Kabal, An information theoretic approach to source enumeration in array signal processing. IEEE Trans. Signal Process. **52**(5), 1171–1178 (2004)
128. R Roy, T Kailath, Esprit-estimation of signal parameters via rotational invariance techniques. IEEE Trans. ASSP. **37**(7), 984–995 (1989)
129. M Wax, T Kailath, Detection of signals by information theoretic criteria. IEEE Trans. ASSP. **33**(2), 387–392 (1985)
130. I Ziskind, M Wax, Maximum likelihood localization of multiple sources by alternating projection. IEEE Trans. ASSP. **36**(10), 1553–1560 (1988)
131. M Viberg, B Ottersten, Sensor array processing based on subspace fitting. IEEE Trans. Signal Process. **39**(5), 1110–1121 (1991)
132. S Shahbazpanahi, S Valaee, AB Gershman, A covariance fitting approach to parametric localization of multiple incoherently distributed sources. IEEE Trans. Signal Process. **52**(3), 592–600 (2004)
133. J Rissanen, Modeling by shortest data description. Automatica. **14**, 465–471 (1978)
134. H Akaike, A new look on the statistical model identification. IEEE Trans. Autom. Control. **19**(6), 716–723 (1974)
135. M Kaveh, H Wang, H Hung, On the theoretical performance of a class of estimators of the number of narrow-band sources. IEEE Trans. ASSP. **35**(9), 1350–1352 (1987)
136. QT Zhang, KM Wong, PC Yip, JP Reilly, Statistical analysis of the performance of information theoretic criteria in the detection of the number of signals in array processing. IEEE Trans. ASSP. **37**(10), 1557–1567 (1989)
137. HV Poor, *An Introduction to Signal Detection Estimation*. (Springer, New York, 1994)
138. TM Cover, JA Thomas, *Elements of Information Theory*, 2nd edn. (Wiley, Hoboken, 2006)
139. J Rissanen, A universal prior for integers and estimation by minimum description length. Annals Stat. **11**(2), 416–431 (1983)
140. B Nadler, A Kontorvich, Model selection for sinusoids in noise: statistical analysis and a new penalty term. IEEE Trans. Signal Process. **59**(4), 1333–1345 (2011)
141. TW Anderson, T Wilbur, *An Introduction to Multivariate Statistical Analysis*, 2nd edn. (Wiley, New York, 1958)
142. F Haddadi, MRM Mohammadi, MM Nayebi, MR Aref, Statistical performance analysis of detection of signals by information theoretic criteria. IEEE Trans. Signal Process. **58**(1), 452–457 (2010)
143. M Gastpar, M Vetterli, in *Lecture Notes in Computer Science*. Source-channel communication in sensor networks (Springer, New York, 2003), pp. 162–177
144. AM Sayeed, in *Sensor Networks, IPSN'03, UW Tech. Rep. ECE-1-04*. A statistical signal modeling framework for wireless sensor networks, in Proc. 2nd Int. Workshop on Info. Proc. (Univ. Wisconsin, Madison, WI, Feb 2004), pp. 162–177
145. K Liu, AM Sayeed, in *Proc. 42nd Annual Allerton Conf. on Comm., Control and Comp.* Optimal distributed detection strategies for wireless sensor networks (Monticello, IL, Oct 2004), pp. 703–707
146. A D'Costa, V Ramachandran, A Sayeed, Distributed classification of gaussian space-time sources in wireless sensor networks. IEEE J. Sel. Areas Commun. **22**(6), 1026–1036 (2004)
147. WU Bajwa, J Haupt, AM Sayeed, R Nowak, in *Proc. Int. Symposium on Info. Proc. in Sensor Networks, IPSN'06*. Compressive wireless sensing (Nashville, TN, Apr 2006), pp. 134–142
148. R Rangarajan, R Raich, A Hero, *Sequential design of experiments for a rayleigh inverse scattering problem*. (Bordeaux, France, July 2005), pp. 625–630
149. Y Yang, RS Blum, in *Fourth IEEE Workshop Sens. Array Multichannel Proc. vol. 12*, vol. 14. Radar waveform design using minimum mean-square error and mutual information (Waltham, MA, USA, 2006), pp. 234–238
150. A Kumar, P Ishwar, K Ramchandran, in *Int. Simp. On Info. Proc. In Sensor Netwroks ISPN2004*. On distributed sampling of smooth non-bandlimited fields (Berkeley, CA, April 2004), pp. 89–98
151. E Meijering, A chronology of interpolation: from ancient astronomy to modern signal and image processing. Proc. IEE. **90**(3), 319–342 (March 2002)
152. D Ganesan, S Ratnasamy, H Wang, D Estrin, Coping with irregular spatio-temporal sampling in sensor networks. ACM SIGCOMM Comput. Commun Rev. **34**(1), 125–130 (2004)
153. R Wagner, R Baraniuk, S Du, D Johnson, A Cohen, in *Proc. of Int. Workshop Info. Proc. in Sensor Networks, IPSN'06*. An architecture for distributed wavelet analysis and processing in sensor networks (Nashville, TN, Apr 2006), pp. 243–250
154. R Wagner, H Choi, R Baraniuk, V Delouille, in *Proc. IEEE Stat. Signal Proc. Workshop (SSP)*. Distributed wavelet transform for irregular sensor network grids, (Bordeaux, July 2005), pp. 1196–1201
155. SS Pradhan, J Kusuma, K Ramchandran, Distributed compression in a dense microsensor network. IEEE Signal Process. Mag. **19**(2), 51–60 (2002)
156. M Gastpar, M Vetterli, Power, spatio-temporal bandwidth distortion in large sensor networks. IEEE J Sel. Areas Commun. **23**(4), 745–754 (2005)
157. WU Bajwa, AM Sayeed, R Nowak, *Matched source-channel communication for field estimation in wireless sensor networks*, vol. 44. (CA, Los Angeles, April 2005), pp. 332–339
158. R Mudumbai, J Hespanha, U Madhow, G Barriac, in *Proc. of the Int. Symposium on Info. Theory, ISIT'05*. Scalable feedback control for distributed beamforming in sensor networks (Adelaide, SA, Sept 2005), pp. 137–141
159. J Haupt, R Nowak, Signal reconstruction from noisy random projections. IEEE Trans. Inf. Theory. **52**(9), 4036–4068 (2006)
160. D Baron, MB Wakin, MF Duarte, S Sarvotham, RG Baraniuk, *Distributed compressed sensing, submitted for publication, pre-print*, (November 2005). http://www.ece.rice.edu/drorb/pdf/DCS112005.pdf

161. W Wang, M Garofalakis, K Ramchandran, in *Proc. IPSN'07*. Distributed sparse random projections for refinable approximation (Cambridge, MA, April 2007), pp. 331–339
162. MF Duarte, MB Wakin, D Baron, RG Baraniuk, *Universal distributed sensing via random projections*. (Nashville, TN, April 2006), pp. 177–185
163. J Haupt, WU Bajwa, M Rabbat, R Nowak, Compressed sensing for networked data. IEEE Signal Process. Mag. **2**, 92–101 (2008)
164. M Crovella, E Kolaczyk, in *INFOCOM 2003. Twenty-Second Annual Joint Conf. of the IEEE Computer and Communications Societies. IEEE*, vol. 3. Graph wavelets for spatial traffic analysis (Boston University, MA, March 2003), pp. 1848–1857
165. R Coifman, M Maggioni, Diffusion wavelets. Appl. Comput. Harmon Anal. **21**(1), 53–94 (2006)
166. WU Bajwa, J Haupt, AM Sayeed, R Nowak, Joint source-channel communication for distributed estimation in sensor networks. IEEE Trans. Inf. Theory. **53**(10), 3629–3653 (2007)
167. S Boyd, A Ghosh, B Prabhakar, D Shah, Randomized gossip algorithms. IEEE Trans. Inf. Theory IEEE/ACM Trans. Netw. **52**(6), 2508–2530 (2006)
168. M Rabbat, J Haupt, A Singh, R Nowak, in *Proc. IPSN'06*. Decentralized compression and predistribution via randomized gossiping (Nashville, TN, April 2006), pp. 51–59
169. SJ Kim, K Koh, M Lustig, S Boyd, D Gorinevsky, A method for large-scale ℓ_1-regularized least squares problems with applications in signal processing and statistics. IEEE J. Sel. Top. Signal Process. **1**(4), 606–617 (2007)
170. Y Rachlin, R Negi, P Khosla, in *Proc. of the Fourth Int. Symposium on Info. Proc. in Sensor Networks*. Sensing capacity for discrete sensor network applications (Pittsburgh, PA, April 2005), pp. 126–132
171. S Aeron, M Zhao, V Saligrama, in *IEEE Info. Theory Workshop, ITW'07*. Information theoretic bounds to sensing capacity of sensor networks under fixed SNR (Lake Tahoe, CA, 2007), pp. 84–89
172. S Aeron, M Zhao, V Saligrama, Information theoretic bounds for compressed sensing. IEEE Trans. Inf. Theory. **52**(10), 5111–5130 (2010)
173. S Sanei, J Chambers, *EEG Signal Processing*. (Wiley, Hoboken, 2007)
174. J Herault, C Jutten, in *Proc. of American Institute of Physics (AIP) Conf.: Neural Networks for Computing*. Space or time adaptive signal processing by neural network models (Snowbird,Utah, US, 1986), pp. 206–211
175. AM Aibinu, AR Najeeb, MJE Salami, AA Shafie, Optimal model order selection for transient error autoregressive moving average (TERA) MRI reconstruction method. Proc. World Acad. Sci. Eng. Technol. **32**, 191–195 (2008)
176. MS Pedersen, J Larsen, U Kjems, LC Parra, *A Survey of Convolutive Blind Source Separation Methods, Springer Handbook of Speech Processing*. (Springer, Berlin, 2007)
177. P Comon, Independent component analysis: a new concept. Signal Process. **36**, 287–314 (1994)
178. M Zibulevsky, BA Pearlmutter, Blind source separation by sparse decomposition in a signal dictionary. Neural Comp. **13**(4), 863–882 (2001)
179. Y Luo, JA Chambers, S Lambotharan, I Proudler, Exploitation of source non-stationarity in underdetermined blind source separation with advanced clustering techniques. IEEE Trans. Signal Process. **54**(6), 2198–2212 (2006)
180. Y Li, S Amari, A Cichocki, DWC Ho, S Xie, Underdetermined blind source separation based on sparse representation. IEEE Trans. Signal Process. **54**(2), 423–437 (2006)
181. A Jourjine, S Rickard, O Yilmaz, in *Proc. of IEEE Conf. on Acoustic, Speech, and Signal Proc. ICASSP'2000*, vol. 5. Blind separation of disjoint orthogonal signals: demixing n sources from 2 mixtures (Istanbul, Turkey, 2000), pp. 2985–2988
182. L Vielva, D Erdogmus, C Pantaleon, I Santamaria, J Pereda, JC Principe, in *Proc. of IEEE Conf. On Acoustic, Speech, and Signal Proc. ICASSP'02*, vol. 3. Underdetermined blind source separation in a time-varying environment (Orlando, FL, USA, 13), pp. 3049–3052
183. I Takigawa, M Kudo, A Nakamura, J Toyama, in *Proc. of 5th Int. Conf. on Independent Component Analysis*. On the minimum ℓ_1-norm signal recovery in underdetermined source separation (Granada, Spain, 2004), pp. 22–24
184. CC Took, S Sanei, J Chambers, in *in Proc IEEE ICASSP'06*. A filtering approach to underdetermined BSS with application to temporomandibular disorders (IEEE, France, May 2006), pp. 1124–1127
185. T Melia, S Rickard, Underdetermined blind source separation in echoic environment using DESPRIT, EURASIP. J. Adv. Signal Process. **2007**(Article No 86484), 19 (2007)
186. K Nazarpour, S Sanei, L Shoker, JA Chambers, in *Proc. of EUSIPCO 2006*. Parallel space-time-frequency decomposition of EEG signals for brain computer interfacing (Florence, Italy, p. 2006
187. R Gribonval, S Lesage, A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges. Proc. of ESANN, 323–330 (April 2006)
188. SS Chen, DL Donoho, MA Saunders, Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. **20**, 33–61 (1998)
189. E Candes, J Romberg, ℓ_1-magic: Recovery of sparse signals. http://www.acm.caltech.edu/l1magic/.Availableonline
190. JJ Fuchs, On sparse representations in arbitrary redundant bases. IEEE Trans. Inf. Theory. **50**(6), 1341–1344 (2004)
191. J Bobin, Y Moudden, JL Starck, M Elad, Morphological diversity and source separation. IEEE Signal Process. Lett. **13**(7), 409–502 (2006)
192. J Bobin, JL Starck, JM Fadili, Y Moudden, DL Donoho, Morphological component analysis: an adaptive thresholding strategy. IEEE Trans. Image Process. **16**(11), 2675–2685 (2007)
193. A Schmidt, JMF Moura, in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Proc., ICASSP'09*. Field inversion by consensus and compressed sensing IEEE,Taipei, Taiwan, 2009), pp. 2417–2420
194. M Mishali, YC Eldar, in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Proc., ICASSP'09 IEEE*. Sparce source separation from orthogonal mixtures (Taipei, Taiwan, 2009), pp. 3145–3148
195. H Schulze, C Lueders, *Theory and Applications of OFDM and CDMA:Wideband Wireless Communications*. (Wiley, NJ, 2005)
196. M Ozdemir, H Arslan, Channel estimation for wireless ofdm systems. IEEE Commun. Surv. Tutor. **9**(2), 18–48 (2007)
197. JJV de Beek, O Edfors, M Sandell, SK Wilson, P Borjesson, in *Proc. 45th IEEE Vehicular Technology Conf., Chicago*, vol. 2. On channel estimation in OFDM systems (Chicago, IL, July 1995), pp. 815–819
198. M Morelli, U Mengali, A comparison of pilot-aided channel estimation methods for ofdm systems. IEEE Trans. Signal Process. **49**(12), 3065–3073 (2001)
199. O Edfors, M Sandell, J-JV de Beek, SK Wilson, OFDM channel estimation by singular value decomposition. IEEE Trans. Commun. **46**(7), 931–939 (1998)
200. P Strobach, Low-rank adaptive filters. IEEE Trans. Signal Process. **44**(2), 2932–2947 (1996)
201. S Coleri, M Ergen, A Puri, A Bahai, Channel estimation techniques based on pilot arrangement in OFDM systems. IEEE Trans. Broadcast. **48**(3), 223–229 (2002)
202. SG Kang, YM Ha, EK Joo, A comparative investigation on channel estimation algorithms for OFDM in mobile communications. IEEE Trans. Broadcast. **49**(2), 142–149 (2003)
203. G Tauböck, F Hlawatsch, in *exploiting channel sparsity for reducing pilots, in Proc. ICASSP'08*. A compressed sensing technique for OFDM channel estimation in mobile environments (Las Vegas, March, 2008), pp. 2885–2888
204. MR Raghavendra, K Giridhar, Improving channel estimation in OFDM systems for sparse multipath channels. IEEE Signal Process. Lett. **12**(1), 52–55 (2005)
205. T Kang, RA Iltis, in *in IEEE Int. Conf. on Acoustics, Speech and Signal Proc., ICASSP'08*. Matching pursuits channel estimation for an underwater acoustic OFDM modem (Las Vegas, March, 2008), pp. 5296–5299
206. P Pakrooh, A Amini, F Marvasti, *Ofdm pilot allocation for sparse channel estimation*, (2011)
207. O Christensen, *An introduction to frames and Riesz basis, ser. Applied and Numerical Harmonic Analysis*. (Basel, Birkhauser, 2003)

doi:10.1186/1687-6180-2012-44
Cite this article as: Marvasti *et al.*: **A unified approach to sparse signal processing.** *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:44.