Mauro Mangia · Fabio Pareschi
Valerio Cambareri · Riccardo Rovatti
Gianluca Setti

Adapted Compressed Sensing for Effective Hardware Implementations

A Design Flow for Signal-Level Optimization of Compressed Sensing Stages

Springer

Adapted Compressed Sensing for Effective Hardware Implementations

Mauro Mangia • Fabio Pareschi • Valerio Cambareri
Riccardo Rovatti • Gianluca Setti

Adapted Compressed Sensing for Effective Hardware Implementations

A Design Flow for Signal-Level Optimization of Compressed Sensing Stages

Springer

Mauro Mangia
ARCES
Università di Bologna
Bologna, Italy

Valerio Cambareri
ICTEAM/ELEN
Université Catholique de Louvain
Louvain-la-Neuve, Belgium

Gianluca Setti
ENDIF
Università di Ferrara
Ferrara, Italy

Fabio Pareschi
ENDIF
Università di Ferrara
Ferrara, Italy

Riccardo Rovatti
DEI, ARCES
Università di Bologna
Bologna, Italy

ISBN 978-3-319-61372-7 ISBN 978-3-319-61373-4 (eBook)
DOI 10.1007/978-3-319-61373-4

Library of Congress Control Number: 2017943984

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To the one and only
woman in this book
and in my life.

Riccardo

Preface

The aim of this book is to give a concrete answer to the following question:

Can compressed sensing effectively yield optimized means for signal acquisition, encoding, and encryption, either in analog or digital circuits and systems, when implementation constraints are considered in its realization?

The reason why this question is important is that compressed-sensing (CS) has been intensely discussed in the engineering community for more than a decade as a hot research topic, gathering a great deal of effort from a large community that unites scientists in applied mathematics and information theory, as well as engineers of analog/digital circuits and optical systems. Yet, several investigations have been dominated by a few misconceptions that somehow hindered the application of this promising technique to real-world systems.

The first concept is that optimization and adaptivity are fundamentally pointless since CS is born as a universal technique that cannot be significantly improved.

The second is that even if one wants to optimize CS, the degrees of freedom to do it are not there, since it is a technique that spreads information so uniformly that no criteria are able to tell important parts to emphasize from less important parts to neglect.

Both concepts are grounded in fundamental mathematical results that are indeed the pillars of CS and are indispensable pieces of the formal construction on which the whole discipline relies. Regrettably, starting from formally true theorems, the folklore has sometimes derived misleading design guidelines.

The idea that adaptivity is useless, often indicated as *universality*, has its roots in the seminal papers originating the very concept of CS and in other later information-theoretic results. In the original setting, such an idea is extremely important to put CS in the right perspective and give it the full dignity of an acquisition method with general applicability. The mathematical derivations produce upper bounds on the ratio between the performance of an adaptive strategy with respect to that of non-adaptive CS. Such bounds being finite, we know that the performance of an agnostic CS is not too far away for the most specialized technique one may devise. Yet, in

practical cases, constants are so large that the theoretical bounds say, nothing but that adaptivity cannot outperform non-adaptivity by more than a factor, say 100. Clearly, no engineer would be prevented from trying the optimization of a system by the knowledge that improvements will be less than 10000%!

The other concept that is sometimes invoked to divert people from serious CS optimization is that of *democracy*. CS works by encoding high-dimensional signals into lower-dimensional collections of measurements, and *democracy* has been developed to decide how to deal with measurements that may have been corrupted during acquisition. Under suitably specified conditions, all measurements can be considered as equally important as they all contribute in the same way to the mathematical properties that guarantee that the original signal can be retrieved. This implies that simply discarding the corrupted information leads to a graceful degradation in performance.

The development is based on a worst-case analysis that is intrinsically invariant with respect to symmetries of the system since worst-case configurations can be replicated exploiting the same symmetry. It is not surprising that measurements computed with substantially the same procedure are equally important from such a pessimistic point of view. Nevertheless, this does not prevent some measurement from being more informative than others in non-worst-case conditions.

The truth is that, overall, mathematical *universality* and *democracy* have very little to do with the real performance of CS systems. Measurements can be selected and can be optimized in a variety of quite effective ways, even taking into account typical implementation constraints and the need to make the final embodiment less expensive with respect to common cost functions like area, power, time, etc.

The aim of this book is to show how this can be done and what benefits can be expected as far as acquisition performance and implementation costs are concerned.

Chapter 1 is dedicated to a brief review of the main ideas defining CS and guaranteeing that it is a viable option. Chapters 2 and 3 address rakeness-based design of CS describing how it derives from the highly *non-democratic* nature of non-worst-case CS, showing how it improves reconstruction performance over *universal* and agnostic CS, and finally discussing pros and cons of adapting sensing to the class of signals to acquire.

Chapter 4 addresses the computational complexity of CS from the point of view of hardware implementations. After identifying the key parameters on which the operating cost of CS-based acquisition depends, it adapts rakeness-based design to address the trade-off between such a cost and reconstruction performance.

Chapter 5 takes a brief detour to discuss how random processes can be generated so that they have only a very limited number of values while also reproducing some prescribed second-order statistical feature. This is a general problem with applications going beyond implementation-friendly rakeness-based CS.

Chapter 6 describes the main architectural options in implementing CS systems and shows their implications on signal-level functionality. It also tackles the problem of saturation with an approach aiming at extracting every little piece of information even from corrupted measurements with a truly "everything but the *oink*" philosophy.

Chapter 7 lists and discusses several CS implementations that see it embedded in the analog-to-digital part of the signal chain, giving rise to an analog-to-information stage. A final comparison chart shows how rakeness-based design of CS allows to obtain the most effective implementation.

Chapter 8 takes a different point of view and looks at CS as a purely digital lossy compression stage whose main feature is that of being extremely simple. CS lossy compression is paired with lossless compression, and overall performance is evaluated to show that, when rakeness-based CS is adopted, one obtains an extremely simple but effective bit squeezing mechanism. Such a mechanism is then put to work for the acquisition of biosignals and implementations with various levels of complexity being analyzed.

Finally, Chap. 9 focuses on an extremely useful side effect of CS, i.e., that it may be used simultaneously as an efficient acquisition scheme and as a low-complexity encryption stage. The fact that encryption comes almost for free implies that security is somehow limited, but the overall robustness to classical attacks is good enough to be considered when very low-cost systems are sought.

The book spans a quite wide range of concepts and, though it aims at being self-contained and easy to follow for those interested in application of CS, it requires some taste for mathematical issues, especially in the first few chapters. Though not overly detailed, also system- and circuit-level considerations may require some confidence in the design of mixed-signal circuits or digital architectures.

Bologna, Italy — Mauro Mangia
Ferrara, Italy — Fabio Pareschi
Louvain-la-Neuve, Belgium — Valerio Cambareri
Bologna, Italy — Riccardo Rovatti
Ferrara, Italy — Gianluca Setti

Contents

Chapter 1
Introduction to Compressed Sensing: Fundamentals and Guarantees

1.1 Signal Acquisition and Compressed Sensing

To interact with the physical world, information processing systems must be able to perform three fundamental activities: acquire information on the phenomenon with which they are supposed to interact, process this information to decide if and how to act back, and transform this decision into a physical effect.

This book concentrates on a technique involved in the first of these three activities, i.e., the acquisition of information. All modern engineering recognizes that information is carried by *signals*, i.e., physical quantities like voltages or currents, that change randomly in time and can be modeled as a stochastic process.

Natural stochastic processes are intrinsically continuous in time and magnitudes meaning that their instances are function $x(t) : \mathbb{R} \mapsto \mathbb{R}^s$ for some $s \geq 1$. Though vector processes are ubiquitous (group of sensors, images, etc.), we concentrate on the case $s = 1$ and note that, nowadays digital processing is discrete in both time and magnitude. Hence, acquisition always implies sampling in time and quantization so that the time evolution of the physical signal is translated into a stream of binary words.

The conventional approach is to decide a sampling rate r_x measured in samples per second, and take samples at multiples of $1/r_x$ to produce the sequence $x_k : \mathbb{Z} \mapsto \mathbb{R}$ such that $x_k = x(k/r_x)$. Each sample is then quantized into the integer $Q(x_k)$ equivalent to a binary word that finally enters subsequent digital processing. Figure 1.1 shows this conventional two-stages approach.

Here, we are not interested in the design of the sampling step, and assume that r_x is the *sufficient rate*, defined as the minimum rate such that the sequence x_k of non-quantized samples contains the information sought by the application. Quite often, the sufficient rate r_x coincides with the Nyquist rate, i.e., with twice the largest frequency in the spectrum of the waveform to acquire, since this choice establishes a mathematical bijection between that waveform and the sequence itself.

M. Mangia et al., *Adapted Compressed Sensing for Effective Hardware Implementations*, DOI 10.1007/978-3-319-61373-4_1

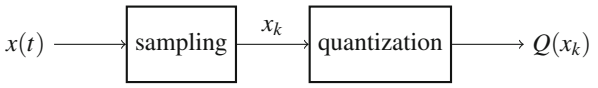


Fig. 1.1 The two-stages decomposition of a basic acquisition

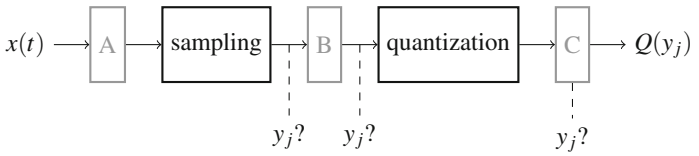


Fig. 1.2 The acquisition signal chain modified by Compressed Sensing. Depending on implementation choices, the subsufficient-rate sequence y_k may appear at different points

Yet, for both practical and theoretical reasons the sufficient rate can be different from the Nyquist one and all subsequent considerations are independent of it. What is important here is that Compressed Sensing (CS) aims at translating the signal waveform into a sequence of scalars y_j (that we will call *measurements*) whose rate $r_y < r_x$ is *subsufficient* and whose quantized version $Q(y_j)$ can be passed to digital processing since it contains the needed information. This is why, CS is often advocated as a method to achieve sub-Nyquist sampling.

To reach its goal, CS performs some early processing on the signal by inserting intermediate stages in the acquisition signal chain. Additional processing may be added at different points as reported in Fig. 1.2 where:

A is a continuous-time, analog preprocessing stage;
B is a discrete-time, analog processing stage;
C is a digital postprocessing stage.

Depending on the implementation strategies (that will be discussed later in the book), the subsufficient-rate sequence y_j may appear at different points in the signal chain. Notwithstanding this, the relationship between y_j and the sufficient-rate sequence of samples x_j is always block-wise linear.

There are two block sizes m and n, with $m < n$ such that subsequent blocks of n adjacent samples x_k are mapped linearly into subsequent blocks of m adjacent measurements y_j (Fig. 1.3). More formally, for any $l \in \mathbb{Z}$ there are an $m \times n$ matrix $\boldsymbol{A}^{(l)}$ a vector $\boldsymbol{x}^{(l)} \in \mathbb{R}^n$ with $\boldsymbol{x}_k^{(l)} = x_{ln+k}$, and a vector $\boldsymbol{y}^{(l)} \in \mathbb{R}^m$ with $\boldsymbol{y}_j^{(l)} = y_{lm+j}$ such that

$$\boldsymbol{y}^{(l)} = \boldsymbol{A}^{(l)}\boldsymbol{x}^{(l)} \tag{1.1}$$

Such a block encoding clearly performs a rate reduction $r_x/r_y = n/m$. Yet, since the rate of x_j is the sufficient rate, such a rate reduction is feasible only under specific conditions and assumptions.

Note first that the measurements y_k are not samples, i.e., though the sufficient rate r_x was defined as the lowest rate at which samples must be extracted from the signal

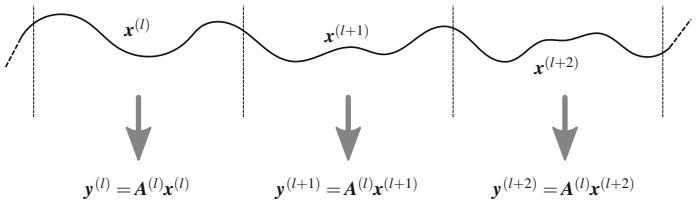


Fig. 1.3 A two-dimensional random vector that concentrates along a one-dimensional subspace

to preserve information, this does not prevent a smaller number of scalars that are not samples to contain the same information.

For this to be possible, one must accept that sampling may not be the most effective way of squeezing information out of a waveform. This is exactly what happens to the classes of signals for which CS expresses its potential: intrinsically low-dimensional signals conventionally indicated as *sparse*.

We will see in the following that, for these signals one may go back from measurements to samples despite the rate reduction, providing that the linear mappings $\boldsymbol{A}^{(l)}$ in (1.1) are properly designed and suitable reconstruction algorithms are employed.

Actually, though the linear operator $\boldsymbol{A}^{(l)}$ typically changes with l, CS acquisition operates independently on each block by segmenting the original signal $x(t)$ in non-overlapping windows (whose duration n/r_x corresponds to n samples at the sufficient rate r_x), and producing a vector of m measurements for each window. With this, analysis and design may concentrate on a single window/block, dropping the $\cdot^{(l)}$ superscript.

Finally, note that the physical realization of the acquisition system will surely superimpose noise to the signal, and that the quantization operated on the measurements before they are fed into processing stages can be thought as a further disturbance. Overall, the relationship between sufficient samples and measurements produced by CS is in general

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x} + \boldsymbol{\eta}^{\boldsymbol{x}}) + \boldsymbol{\eta}^{\boldsymbol{y}} \tag{1.2}$$

where $\boldsymbol{\eta}^{\boldsymbol{x}}$ and $\boldsymbol{\eta}^{\boldsymbol{y}}$ take into account all nonidealities affecting $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. As usual, we prefer a unique noise source at the end of the processing chain such that the encoding stage of CS is summarized by

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\eta} \tag{1.3}$$

with $\boldsymbol{\eta} = \boldsymbol{A}\boldsymbol{\eta}^{\boldsymbol{x}} + \boldsymbol{\eta}^{\boldsymbol{y}}$. The encoding must be conceived so that it is possible to go back from the m-dimensional vector $\boldsymbol{y}$ to the n-dimensional vector $\boldsymbol{x}$ despite the fact that we would like m to be as small as possible.

Three concepts interact in the design and operation of such a system: low-dimensional signal models, sensing operators, and reconstruction algorithms.

1.2 Low-Dimensional Signal Models

Since we are reasoning block-by-block and the information we need is contained in the samples x_j, we may identify signals with random vectors $\boldsymbol{x} \in \mathbb{R}^n$. Given any subset $X \subseteq \mathbb{R}^n$ one may ask whether X is a good representative of the distribution of $\boldsymbol{x}$ in the signal space by defining the distance

$$\Delta(X, \boldsymbol{x}) = \min_{\boldsymbol{\xi} \in X} \|\boldsymbol{x} - \boldsymbol{\xi}\|_2$$

that is the minimum error in which one incurs when tries to approximate $\boldsymbol{x}$ with the closest point in X.

The average error energy $E_\Delta = \mathbf{E}_{\boldsymbol{x}}\left[\Delta(X, \boldsymbol{x})^2\right]$ may be matched with the signal average energy $E_x = \mathbf{E}_{\boldsymbol{x}}\left[\|\boldsymbol{x}\|_2^2\right]$ and if E_Δ / E_x is very small, then one may say that $\boldsymbol{x}$ concentrates in X.

The subset X in which a signal concentrates may have different shapes. Here, we are interested in X's that are subspaces or union of subspaces. The geometric dimensions of these subspaces are good proxies of the true information content of the signals since to express a point in a κ-dimensional subspace of $\mathbb{R}^n$, only κ scalar quantities are needed.

We may illustrate the subspace/union of subspaces cases with two examples that allow us to formulate some general definition.

1.2.1 Concentrated and Localized Signals

As an example of the first case, consider a random vector $\boldsymbol{x} \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, i.e., distributed according to a multivariate Gaussian with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ whose probability density function (PDF) is

$$g(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \boldsymbol{\xi}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} e^{-\frac{1}{2}(\boldsymbol{\xi}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi}-\boldsymbol{\mu})} \tag{1.4}$$

In particular, set $n = 2$, $\boldsymbol{\mu} = (0, 0)^\top$, and $\boldsymbol{\Sigma} = \begin{pmatrix} 5 & 3 \\ 3 & 2 \end{pmatrix}$ and note that, since $\boldsymbol{x}$ is zero-mean, its correlation matrix $\mathscr{X} = \mathbf{E}_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{x}^\top]$ coincides with the covariance matrix $\boldsymbol{\Sigma}$ and the average energy of the vector is $E_{\boldsymbol{x}} = \mathrm{tr}(\mathscr{X}) = \mathrm{tr}(\boldsymbol{\Sigma}) = 7$ ($\mathrm{tr}(\cdot)$ indicates the trace of its matrix argument).

Figure 1.4 shows a large number of realizations of such a vector as points on the two-dimensional plane $\mathbb{R}^2$. Simple visual inspection reveals that the points tend to align with the red straight line that correspond to the subspace $\mathrm{span}(\boldsymbol{d}_0)$ of the multiples of the vector $\boldsymbol{d}_0 = (1 + \sqrt{5}, 2)^\top$.

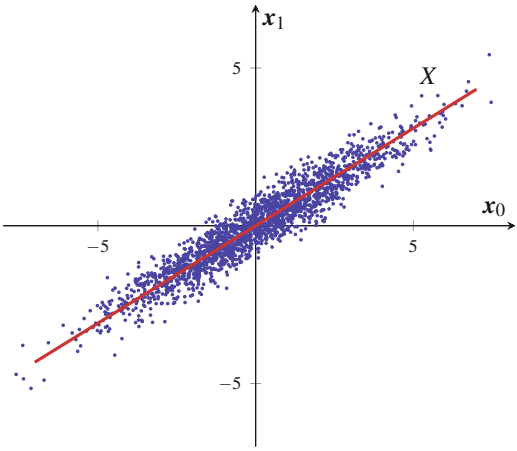


Fig. 1.4 A two-dimensional random vector that concentrates along a one-dimensional subspace

If the subset X is such a subspace, one may easily compute $\Delta(X, \boldsymbol{x})$ by orthogonal projection yielding $\Delta(X, \boldsymbol{x}) = \boldsymbol{d}_1^\top \boldsymbol{x} / \|\boldsymbol{d}_1\|_2$ where $\boldsymbol{d}_1 = (1 - \sqrt{5}, 2)^\top$ is orthogonal to $\boldsymbol{d}_0$. Hence $\Delta(X, \boldsymbol{x})$ is a zero-mean Gaussian random variable with variance $\boldsymbol{d}_1^\top \mathscr{X} \boldsymbol{d}_1 = \boldsymbol{d}_1^\top \boldsymbol{\Sigma} \boldsymbol{d}_1 = E_\Delta \simeq 0.15$ and the visual intuition is confirmed by the fact that the average error energy is only approximately 2% of the signal energy ($E_\Delta \simeq 0.15$ vs $E_{\boldsymbol{x}} = 7$).

In the language of matrices, $\boldsymbol{d}_0$ and $\boldsymbol{d}_1$ are eigenvectors of the correlation matrix $\mathscr{X}$ and E_Δ is the eigenvalue corresponding to $\boldsymbol{d}_1$, i.e., to the direction that is orthogonal to the subspace along which $\boldsymbol{x}$ concentrates.

Within this framework, it is easy to generalize our example to a generic dimensionality n. The correlation matrix $\mathscr{X}$ is symmetric and positive semidefinite, thus it features a set of orthogonal eigenvectors $\boldsymbol{d}_0, \ldots, \boldsymbol{d}_{n-1}$ and associated eigenvalues $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{n-1} \geq 0$. If the partial sum of eigenvalues $\sum_{j=\kappa}^{n-1} \lambda_j$ is small compared with the trace $\mathrm{tr}(\mathscr{X}) = \sum_{j=0}^{n-1} \lambda_j$, then $\boldsymbol{x}$ can be said to concentrate in the subspace $\mathrm{span}(\boldsymbol{d}_0, \ldots, \boldsymbol{d}_{\kappa-1})$. This is often named *principal component analysis* since the first κ eigenvectors can be seen as the *components* along which the signal has most of the energy.

The point of view of principal component analysis allows also to capture less sharp behaviors in which there is no true *concentration*, but the sequence of eigenvalues is non-constant, thus revealing that the energy of the signal is not uniformly distributed in the signal space. The degree of non-uniformity can be measured by defining a *localization* index.

Definition 1.1 Given a random vector $\boldsymbol{x} \in \mathbb{R}^n$ let $\mathscr{X}$ be its correlation matrix $\mathscr{X} = \mathbf{E}_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{x}^\top]$ and $\lambda_0 \geq \lambda_1 \geq \cdots \geq 0$ its eigenvalues. The degree of localization of $\boldsymbol{x}$ is

$$\mathscr{L}_{\boldsymbol{x}} = \sum_{j=0}^{n-1} \left(\frac{\lambda_j}{\mathrm{tr}(\mathscr{X})} - \frac{1}{n} \right)^2 = \frac{\mathrm{tr}(\mathscr{X}^2)}{\mathrm{tr}^2(\mathscr{X})} - \frac{1}{n} \tag{1.5}$$

Table 1.1 $\mathscr{L}_{\boldsymbol{x}}$ for real-world signals classes

Signal	Sufficient rate r_x	n	$\mathscr{L}_{\boldsymbol{x}}$
ECG	720 sample/s	360	0.187
Speech	20 Ksample/s	200	0.069
EMG	400 sample/s	200	0.021
B&W printed letters	24×24 pixels	576	0.016

$\mathscr{L}_{\boldsymbol{x}}$ is nothing but a normalized squared distance between the eigenvalues of $\mathscr{X}$ and a sequence of uniform eigenvalues that would characterize a white signal. Localization is such that $0 \leq \mathscr{L}_{\boldsymbol{x}} \leq 1-1/n$ where the upper bound is due to the fact that $\mathscr{X}$ is positive semidefinite. In fact, we must have $|\mathscr{X}_{j,k}| \leq \sqrt{\mathscr{X}_{j,j}\mathscr{X}_{k,k}}$ for $j,k = 0,\ldots,n-1$, and thus $\mathrm{tr}(\mathscr{X}^2) = \sum_{j=0}^{n-1}\sum_{k=0}^{n-1}\mathscr{X}_{j,k}\mathscr{X}_{k,j} \leq \sum_{j=0}^{n-1}\sum_{k=0}^{n-1}\mathscr{X}_{j,j}\mathscr{X}_{k,k} = \mathrm{tr}^2(\mathscr{X})$.

Clearly, $\mathscr{L}_{\boldsymbol{x}} = 0$ when all the eigenvalues are equal and there is no preferred direction along which energy is distributed. At the opposite side, $\mathscr{L}_{\boldsymbol{x}} = 1 - 1/n$ when only one eigenvalue is non-null and $\boldsymbol{x}$ is always aligned with the corresponding eigenvector.

Real-world signals typically feature $\mathscr{L}_{\boldsymbol{x}} > 0$. As an example, Table 1.1 reports the localization index for 4 real-world signals: Electro Cardio Grams (ECG) and Electro Myo Grams (EMG) taken from the Physionet database [8], black and white still images of isolated printed letters [15], and speech segments of 10 ms taken from the EMU database [11].

ECG, EMG, and speech segments are one-dimensional signals and for each of them, once the sufficient rate r_x is set, we collect windows $\boldsymbol{x}^{(l)}$ of n subsequent samples for $l = 0,\ldots,N-1$. The correlation matrix $\mathscr{X} = \mathbf{E}_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{x}^\top]$ is estimated as $\mathscr{X} \simeq \frac{1}{N}\sum_{l=0}^{N-1}\boldsymbol{x}^{(l)}\left(\boldsymbol{x}^{(l)}\right)^\top$ and its eigenvalues extracted to compute (1.5).

Still images are constant, two-dimensional signals whose acquisition implicitly yields a certain number of pixels values. In this case we consider 24×24 images whose pixel values are aligned in the 576-dimensional vectors $\boldsymbol{x}^{(l)}$ to estimate $\mathscr{X}$ as before.

1.2.2 Sparse and Compressible Signals

As a second example consider $n = 3$ and three independent vectors $\boldsymbol{d}_0, \boldsymbol{d}_1, \boldsymbol{d}_2 \in \mathbb{R}^3$. Assume that the vectors have unit length and define the PDF of $\boldsymbol{x}$ exploiting (1.4) as in

$$f_{\boldsymbol{x}}(\boldsymbol{\xi}) = \frac{1}{3}\sum_{j=0}^{2} g\left(\mathbf{0}, \boldsymbol{d}_j\boldsymbol{d}_j^\top + \frac{\sigma_\eta^2}{3}\boldsymbol{I}; \boldsymbol{\xi}\right)$$

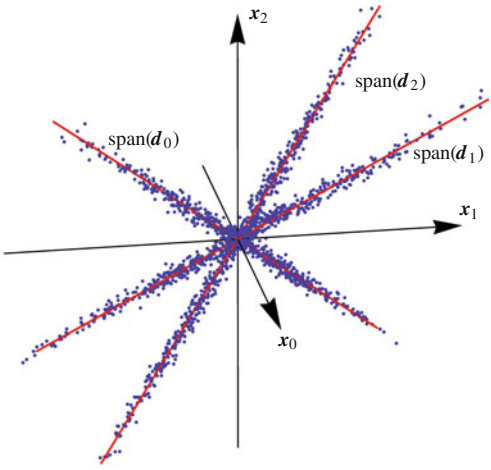


Fig. 1.5 A three-dimensional random vector that concentrates in a union of one-dimensional subspaces

where $\mathbf{0} = (0, 0, 0)^\top$ and $\boldsymbol{I}$ is the 3×3 identity matrix and $\sigma_\eta^2 \ll 1$. Since the vector is zero-mean we still have that correlation and covariance matrices coincide. From $\|\boldsymbol{d}_j\|_2 = 1$ we get $\mathrm{tr}(\boldsymbol{d}_j \boldsymbol{d}_j^\top + {}^{\sigma_\eta^2}\!/_3) = 1 + \sigma_\eta^2$ that is also the average energy of $\boldsymbol{x}$.

Figure 1.5 shows a large number of realizations of such a vector as points in the three-dimensional space $\mathbb{R}^3$ for $\boldsymbol{d}_0 = {}^1\!/\sqrt{3}(1, 1, 1)^\top$, $\boldsymbol{d}_1 = {}^1\!/\sqrt{3}(1, -1, 1)^\top$, and $\boldsymbol{d}_2 = {}^1\!/\sqrt{3}(-1, 1, 1)^\top$. Again, visual inspection is enough to reveal that the points tend to concentrate in one of the three subspaces $\mathrm{span}(\boldsymbol{d}_j)$ for $j = 0, 1, 2$ so that we may set $X = \mathrm{span}(\boldsymbol{d}_0) \cup \mathrm{span}(\boldsymbol{d}_1) \cup \mathrm{span}(\boldsymbol{d}_2)$.

Also in this case the error one commits in thinking that $\boldsymbol{x} \in X$ can be computed by orthogonal projection and, thanks to the definition of $f_{\boldsymbol{x}}$, immediately yields $E_\Delta = \sigma_\eta^2 \ll E_{\boldsymbol{x}}$. This kind of concentration is commonly indicated as *compressibility* and, in its extreme form when $E_\Delta = 0$, *sparsity*.

To clarify the reason, note that if we build the matrix $\boldsymbol{D} = \boldsymbol{d}_0 \boldsymbol{d}_0^\top + \boldsymbol{d}_1 \boldsymbol{d}_1^\top + \boldsymbol{d}_2 \boldsymbol{d}_2^\top$ then $\boldsymbol{D}$ is non-singular and $\boldsymbol{D}^{-1}$ brings $\boldsymbol{d}_j$ on the j-th coordinate axis for $j = 0, 1, 2$. Therefore, $\boldsymbol{\xi} = \boldsymbol{D}^{-1}\boldsymbol{x}$ concentrates along coordinate axes, i.e., has only one substantially non-null component. This allows to generalize the concept to higher dimensionality saying that $\boldsymbol{x}$ is κ-sparse (κ-compressible) when there is a basis $\boldsymbol{D}$ such that $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$ where $\boldsymbol{\xi}$ is a sparse vector with at most $\kappa < n$ (substantially) non-null entries.

As a further generalization, we may accept that $\boldsymbol{D}$ is a *dictionary*, that is a collection of $d > n$ column vectors that contains a subset of n independent vectors and thus is able to express every $\boldsymbol{x} \in \mathbb{R}^n$ though not in a unique way.

The rationale behind accepting dictionaries is that very low sparsity counts κ may be extremely beneficial in reducing the number of measurements m. In fact, we will later see that the minimum number of measurements needed for effectively reconstructing $\boldsymbol{x}$ from $\boldsymbol{y}$ is $m^* = \mathsf{O}(\kappa \log(d))$. Adding *specialized* vectors to the

set that can be used to express $\boldsymbol{x}$ may help achieving this goal. As an extreme case, for example, if $\boldsymbol{x}$ could be only one out of a finite number of waveforms, those waveforms would be collected as columns of $\boldsymbol{D}$ to yield $\kappa = 1$ and $m^* = \mathsf{O}(\log(d))$ as it is natural since to identify one out of d possible choices only $\log_2(d)$ bits are needed.

Definition 1.2 Given a random vector $\boldsymbol{x} \in \mathbb{R}^n$, we say that it is κ-sparse (κ-compressible) when there is a full rank $n \times d$ matrix ($d \geq n$) such that for every instance of $\boldsymbol{x}$ there is at least one $\xi \in \mathbb{R}^d$ such that $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$ and $\boldsymbol{\xi}$ has not more than $\kappa < n$ (substantially) non-null entries.

Since this does not produce any loss of generality, it is assumed that the columns $\boldsymbol{D}_{\cdot,0}, \ldots, \boldsymbol{D}_{\cdot,d-1}$ of $\boldsymbol{D}$ have unit length $\left\|\boldsymbol{D}_{\cdot,j}\right\|_2 = 1$.

Classical CS works on signals that are κ-sparse or κ-compressible with $\boldsymbol{D}$ being named the *sparsity basis* or *sparsity dictionary*. This is often stated saying that CS exploits the *a priori* knowledge that $\boldsymbol{x}$ is sparse or compressible or, in short, the *sparsity prior*.

The effectiveness of the whole machinery is usually subsumed by few significant parameters like

- The *compression ratio* $\mathtt{CR} = n/m$;
- The *sparsity ratio* n/κ;
- The *measurement overhead* m/κ;
- The *dictionary redundancy* d/n;

that identify the main goals and trade-offs in the design.

The adaptation method that we will describe specializes the conventional approach exploiting the fact that $\mathscr{L}_{\boldsymbol{x}} > 0$ for almost all real-world signals, i.e., the *localization prior*.

1.3 Sensing Operators

If $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$ for some $\boldsymbol{D}$ and $\boldsymbol{\xi}$, then, by temporarily assuming that noise effects are negligible, $\boldsymbol{\eta} = 0$, (1.3) can be rewritten as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{D}\boldsymbol{\xi} = \boldsymbol{B}\boldsymbol{\xi} \tag{1.6}$$

for an $m \times d$ matrix $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{D}$.

Since $m < n \leq d$, the main problem in going from $\boldsymbol{y}$ to $\boldsymbol{\xi}$ is the non-uniqueness of the solution since even if $\boldsymbol{B}$ is full rank, the sheer effect of dimensionality reduction implies $\ker(\boldsymbol{B}) \neq \{0\}$. Hence $\boldsymbol{B}$ is non-injective and each $\boldsymbol{y}$ has multiple possible counterimages among which $\boldsymbol{\xi}$ is to be found.

To exemplify how sparsity helps in this, let us analyze a toy case in which $n = 3$ but $\boldsymbol{x}$ is known to be 1-sparse. This is exactly the situation described in the second example of the previous section.

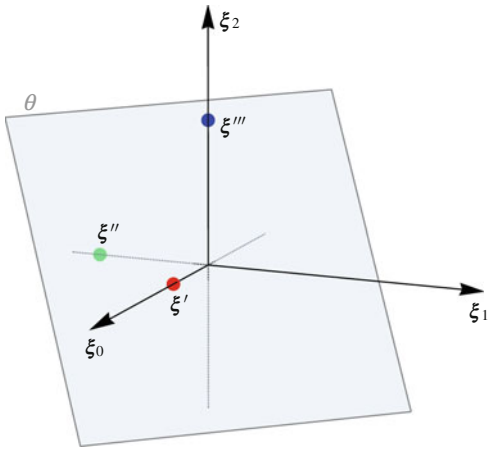


Fig. 1.6 Points representing a 1-sparse signal $\boldsymbol{x} \in \mathbb{R}^3$ and a plane θ onto which they can be projected to reduce their dimensionality

Sparsity means that the signal $\boldsymbol{x}$ is mapped into a vector $\boldsymbol{\xi}$ with only one non-null component. Hence, different instances of $\boldsymbol{x}$ are equivalent to points like $\boldsymbol{\xi}'$, $\boldsymbol{\xi}''$, and $\boldsymbol{\xi}'''$ in Fig. 1.6. Though $\boldsymbol{x} \in \mathbb{R}^3$, the sparsity prior suggests that each of those points can be identified by less than three scalars. Yet, even if the axis on which each of them lies were known, at least 1 scalar should be specified for each point to indicate its position along that axis. With this one may reasonably expect that the number of scalars needed to identify each point is 2 (more than a single coordinate but less than the full three-coordinate set needed for generic points).

To see that this is exactly what happens consider the plane θ defined by the equation $\xi_0 + \xi_1 + \xi_2 = 0$. The projections on θ can be obtained by a linear mapping corresponding to the matrix

$$\boldsymbol{B} = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & \sqrt{\frac{2}{3}} \end{pmatrix} \tag{1.7}$$

and produce the points $\boldsymbol{y}' = \boldsymbol{B}\boldsymbol{\xi}'$, $\boldsymbol{y}'' = \boldsymbol{B}\boldsymbol{\xi}''$, and $\boldsymbol{y}''' = \boldsymbol{B}\boldsymbol{\xi}'''$ on θ in Fig. 1.7a. Figure 1.7b shows what would be like to look only at the two-dimensional projections. Since the projections on θ of the three coordinate axes are well apart one from the other, the sparsity prior is enough to infer $\boldsymbol{\xi}'$ from $\boldsymbol{y}'$, $\boldsymbol{\xi}''$ from $\boldsymbol{y}''$ and $\boldsymbol{\xi}'''$ from $\boldsymbol{y}'''$.

1.4 Coherence

This qualitative behavior can be made more precise. The projection of the coordinate axes through $\boldsymbol{B}$ are nothing but the columns $\boldsymbol{B}_{\cdot,0}, \ldots, \boldsymbol{B}_{\cdot,d-1}$ of $\boldsymbol{B}$, and are important since the measurements vector $\boldsymbol{y}$ is the linear combination of only κ of them (in

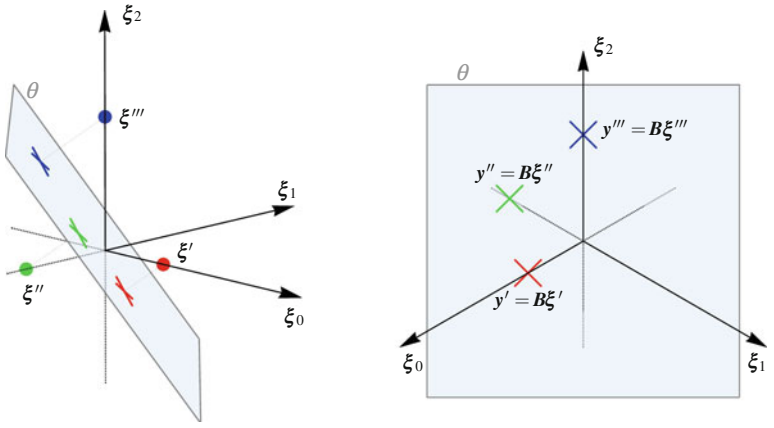


Fig. 1.7 Projection of points representing a 1-sparse signal $\boldsymbol{x} \in \mathbb{R}^3$ onto the plane θ and graphical reconstruction of the original points starting from projections for two differt wiev points

our case $\kappa = 1$ and thus $\boldsymbol{y}$ actually lies on one of them). Hence, the more they are distinguishable, the easier to go from $\boldsymbol{y}$ to the original $\boldsymbol{\xi}$. In our context, complete distinguishability would mean linear independence that is impossible since each of them is a m-dimensional vector and there are $d > m$ of them. Yet, it is possible to formalize the idea of a set of vectors that is *as linear independent as possible* given the dimensionality constraints by resorting to the concept of *frame*.

The theory of frames is huge and the applicability of frames goes well beyond the role that we here give to them: that of being an intuitive support to the requirement we will pose on $\boldsymbol{B}$. For a more focused but synthetic discussion of the concept one may refer to [13, 14]. What we need here is the definition of a special kind of frames.

Definition 1.3 A set $\{\boldsymbol{b}_0, \dots, \boldsymbol{b}_{d-1}\}$ of m-dimensional vectors is said to be a tight, normalized, and equiangular frame (TNEF) if a length ℓ and an angle α exist such that

$$\left|\boldsymbol{b}_j^\top \boldsymbol{b}_l\right| = \begin{cases} \ell^2 & \text{if } j = l \\ \ell^2 \cos(\alpha) & \text{if } j \neq l \end{cases} \tag{1.8}$$

and

$$\frac{m}{d}\ell^{-2} \sum_{j=0}^{d-1} \left(\boldsymbol{b}_j^\top \boldsymbol{v}\right) \boldsymbol{b}_j = \boldsymbol{v} \tag{1.9}$$

for every $\boldsymbol{v} \in \mathbb{R}^m$.

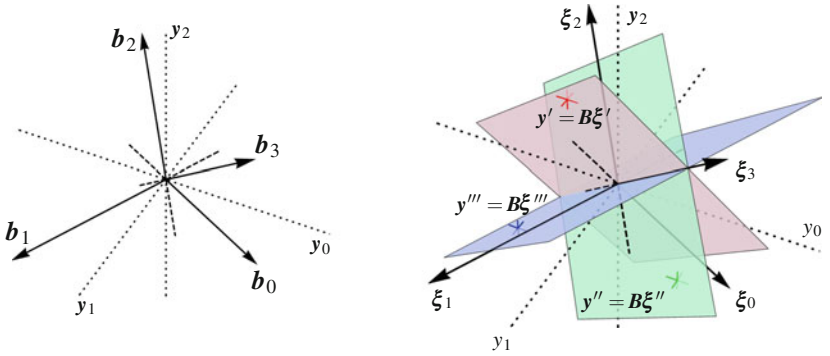


Fig. 1.8 A tight, equiangular normalized frame in $\mathbb{R}^3$ with 4 vectors (**a**) and its use in signal reconstruction (**b**)

Note that (1.8) sets normalization of lengths and equiangularity between the vectors, while (1.9) says that their collection behaves like an orthogonal basis up to a constant depending on the dimensionality reduction m/d. For $m < d$, TNEF are the collection of vectors that best approximate the behavior of an orthogonal basis in which vectors are maximally distinguished.

Actually, the celebrated Naimark's dilation theorem can be specialized to our finite dimensional, real domain to clarify [13] that every TNEF is actually made of the orthogonal projections onto $\mathbb{R}^m$ of the vectors of an orthonormal basis of $\mathbb{R}^d$. If such a higher dimensional basis is the one along which our signal is sparse, the corresponding TNEF is clearly a very good candidate to allow inversion from $\boldsymbol{y}$ to $\boldsymbol{\xi}$.

As an example of the power of such an embedding, Fig. 1.8a shows the arrangement of 4 vectors in $\mathbb{R}^3$ that form a TNEF with $\ell = 1$ and $\cos(\alpha) = 1/3$.

If the vectors of such a TNEF are used to build the 3×4 matrix $\boldsymbol{B}$, this can be used to map 2-sparse vectors $\boldsymbol{\xi}', \boldsymbol{\xi}'', \boldsymbol{\xi}''' \in \mathbb{R}^4$ into the measurement vectors $\boldsymbol{y}', \boldsymbol{y}'', \boldsymbol{y}''' \in \mathbb{R}^3$. In this case the mapping cannot be visualized but its results can be drawn in $\mathbb{R}^3$ as in Fig. 1.8b.

Since the projections of the coordinate axes of $\mathbb{R}^4$ are perfectly distinguishable in $\mathbb{R}^3$ so are the projection of the coordinate planes to which the $\boldsymbol{\xi}$ (that are 2-sparse) belong. Figure 1.8b reports 3 out of the 6 possible coordinate planes. It is clear that, with the exception of particular cases corresponding to 1-sparse signals, the measurements $\boldsymbol{y}'$, $\boldsymbol{y}''$, and $\boldsymbol{y}'''$ only belong to one of these projections and thus indicate which components are non-null in the original signals.

Given these extremely favorable behavior, it is a real pity that TNEF do not exist for every choice of m and d. Actually, their very construction is a thoroughly investigated but still unsolved problem.

What can be of help in this situation is another important property that hinges on the concept of *mutual coherence* [6].

Definition 1.4 Given an $m \times d$ matrix $\boldsymbol{B}$ with column vectors $\boldsymbol{B}_{\cdot,0}, \ldots, \boldsymbol{B}_{\cdot,d-1}$ the mutual coherence is

$$\mu(\boldsymbol{B}) = \mu(\boldsymbol{B}_{\cdot,0}, \ldots, \boldsymbol{B}_{\cdot,d-1}) = \max_{j \neq l} \frac{\left| \boldsymbol{B}_{\cdot,j}^{\top} \boldsymbol{B}_{\cdot,l} \right|}{\left\| \boldsymbol{B}_{\cdot,j} \right\|_2 \left\| \boldsymbol{B}_{\cdot,l} \right\|_2} \tag{1.10}$$

It is well known [18] that mutual coherence, that is nothing but the cosine of the largest angle between any two vectors $\boldsymbol{B}_j$, is bounded by

$$\sqrt{\frac{(d-m)^+}{(d-1)m}} \leq \mu(\boldsymbol{B}) \leq 1 \tag{1.11}$$

where $(\cdot)^+ = \max\{0, \cdot\}$.

For $d > m$, the lower bound in (1.11) is achieved by a normalized set of vectors only if it is a TNEF. In our cases, m and d are set by external constraints and may prevent the construction of a true TNEF. Yet, it is most natural to require that $\mu(\boldsymbol{B})$ is as small as possible to be as close as possible to a TNEF, i.e., to guarantee that the projections of coordinate axes on the measurement space are as distinguishable as possible to allow reconstruction of $\boldsymbol{\xi}$.

In the toy case of Fig. 1.6 the perfect distinguishability of the projections of the coordinate axes is due to the fact that the set

$$\boldsymbol{B}_{\cdot,0} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{6}} \end{pmatrix} \quad \boldsymbol{B}_{\cdot,1} = \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} \end{pmatrix} \quad \boldsymbol{B}_{\cdot,2} = \begin{pmatrix} 0 \\ \sqrt{\frac{2}{3}} \end{pmatrix}$$

is a TNEF with $\ell = \sqrt{2/3}$, and $\alpha = \pi/3$ and in fact $|\boldsymbol{B}_{\cdot,j}^{\top} \boldsymbol{B}_{\cdot,l}| = 1/3$ implies $\mu(\boldsymbol{B}) = 1/2$ that is precisely the lower bound in (1.11) for $d = 3$ and $m = 2$.

We will see that the coherence of the columns of $\boldsymbol{B}$ regulates the performance of some algorithms that are able to reconstruct $\boldsymbol{\xi}$ from $\boldsymbol{y}$. It is then most natural to say that a possible design criterion for $\boldsymbol{A}$ is to make the columns of $\boldsymbol{B} = \boldsymbol{AD}$ as incoherent as possible.

1.5 Restricted Isometries

The previous discussion leverages on sparsity only implicitly by inferring the need that the projections through $\boldsymbol{B}$ of the coordinate axes in the $\boldsymbol{\xi}$ space are as distinguishable as possible. Yet, the sparsity prior can be further exploited to highlight another property that is desirable for $\boldsymbol{B}$.

Define the support of a vector $\boldsymbol{v}$ as the set $\text{supp}(\boldsymbol{v}) = \{j | v_j \neq 0\}$ that contains the positions of its non-null entries, and for any finite set C, denote with $|C|$ its cardinality.

With this notation, if $|\text{supp}(\boldsymbol{\xi}')| = \kappa$ and $|\text{supp}(\boldsymbol{\xi}'')| = \kappa$ and $\Delta\boldsymbol{\xi} = \boldsymbol{\xi}' - \boldsymbol{\xi}''$, then $|\text{supp}(\Delta\boldsymbol{\xi})| \leq 2\kappa$ where the upper bound is hit when there is no cancellation in the componentwise difference $\boldsymbol{\xi}' - \boldsymbol{\xi}''$.

What we have to avoid in the design of $\boldsymbol{B}$ is that both $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}'$ and $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}''$ and thus $\Delta\boldsymbol{\xi} \in \ker(\boldsymbol{B})$. Hence, if one can provide a matrix $\boldsymbol{A}$ such that the kernel of $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{D}$ contains only vectors with more than 2κ non-null component, then is able to uniquely associate any possible $\boldsymbol{y}$ in (1.6) with its corresponding $\boldsymbol{\xi}$.

To formalize this concept further, consider a generic index subset $K \subseteq \{0, \ldots, n-1\}$ and, for any $\boldsymbol{v}$ or matrix $\boldsymbol{M}$, indicate with $\boldsymbol{v}_K$ and $\boldsymbol{M}_{\cdot,K}$, respectively, the same entity with its index constrained in K. Set now $K = \text{supp}(\Delta\boldsymbol{\xi})$ with $|K| = 2\kappa$. Whatever K and $\Delta\boldsymbol{\xi}_K$ we would like to have

$$0 \neq \boldsymbol{B}\Delta\boldsymbol{\xi} = \boldsymbol{B}_{\cdot,K}\Delta\boldsymbol{\xi}_K$$

Hence, every possible submatrix $\boldsymbol{B}_{\cdot,K}$ formed by selecting 2κ columns from $\boldsymbol{B}$ must be full rank.

Classical theoretical guarantees elaborate on this concept so that it can be applied not only to the noiseless case $\boldsymbol{\eta} = 0$. When $\boldsymbol{\eta} \neq 0$, (1.6) becomes $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi} + \boldsymbol{\eta}$ and reconstruction of $\boldsymbol{\xi}$ from $\boldsymbol{y}$, if possible, is always affected by some error.

What becomes important in this case is the relationship between the amount of disturbance, the features of $\boldsymbol{B}$, and the error suffered in retrieving $\boldsymbol{\xi}$. This calls for a translation of our previous considerations in terms of energies.

In general, when $\Delta\boldsymbol{\xi}$ is mapped by $\boldsymbol{B}$, it gives a vector $\Delta\boldsymbol{y} = \boldsymbol{B}\Delta\boldsymbol{\xi} = \boldsymbol{B}_{\cdot,K}\Delta\boldsymbol{\xi}_K$. Requiring $\Delta\boldsymbol{\xi}_K \notin \ker(\boldsymbol{B}_{\cdot,K})$ is equivalent to say $\|\Delta\boldsymbol{y}\|_2 > 0$ and thus that the *energy gain* $\|\Delta\boldsymbol{y}\|_2^2 / \|\Delta\boldsymbol{\xi}_K\|_2^2$ between $\Delta\boldsymbol{\xi}_K$ and $\Delta\boldsymbol{y}$ is non-null.

We may first assess the average behavior of such a gain by assuming that $\boldsymbol{B}_{\cdot,K}$ is full rank and that $\Delta\boldsymbol{\xi}_K$, a full vector, has a radial distribution, i.e., it can be written as the product $\Delta\boldsymbol{\xi} = \alpha\boldsymbol{v}$ where $\boldsymbol{v}$ is a vector uniformly distributed on the $(n-1)$-dimensional surface of a unit sphere and $\alpha \geq 0$ is a random scalar. With this, the average energy gain between $\Delta\boldsymbol{\xi}_K$ and $\Delta\boldsymbol{y}$ is

$$\gamma = \mathbf{E}_{\Delta\xi_K}\left[\frac{\|\boldsymbol{B}_{\cdot,K}\Delta\boldsymbol{\xi}_K\|_2^2}{\|\Delta\boldsymbol{\xi}_K\|_2^2}\right] = \mathbf{E}_{\boldsymbol{v}}\left[\frac{\|\boldsymbol{B}_{\cdot,K}\boldsymbol{v}\|_2^2}{\|\boldsymbol{v}\|_2^2}\right] = \mathbf{E}_{\boldsymbol{v}}\left[\boldsymbol{v}^\top\boldsymbol{B}_{\cdot,K}^\top\boldsymbol{B}_{\cdot,K}\boldsymbol{v}\right]$$

To evaluate the last expression note that $\boldsymbol{B}_{\cdot,K}^\top\boldsymbol{B}_{\cdot,K}$ is a $2\kappa \times 2\kappa$, non-singular symmetric matrix whose eigenvalues are the squares of the so-called singular values of $\boldsymbol{B}_{\cdot,K}$. In general we will indicate with $\sigma_j(\cdot)$ the j-th singular value of a matrix argument and with $\sigma_{\min}(\cdot)$ and $\sigma_{\max}(\cdot)$ the minimum and the maximum of the set of singular values. Associated with such eigenvalues/squared singular values, there are the orthonormal eigenvectors $\boldsymbol{b}_j$ such that $\boldsymbol{B}_{\cdot,K}^\top\boldsymbol{B}_{\cdot,K} = \sum_{j=0}^{2\kappa-1}\sigma_j^2(\boldsymbol{B}_{\cdot,K})\boldsymbol{b}_j\boldsymbol{b}_j^\top$. Hence

$$\mathbf{E}_{\boldsymbol{v}}\left[\boldsymbol{v}^\top\boldsymbol{B}_{\cdot,K}^\top\boldsymbol{B}_{\cdot,K}\boldsymbol{v}\right] = \sum_{j=0}^{2\kappa-1}\sigma_j^2(\boldsymbol{B}_{\cdot,K})\mathbf{E}\left[\boldsymbol{v}^\top\boldsymbol{b}_j\boldsymbol{b}_j^\top\boldsymbol{v}\right] = \sum_{j=0}^{2\kappa-1}\sigma_j^2(\boldsymbol{B}_{\cdot,K})\mathbf{E}\left[\left(\boldsymbol{b}_j^\top\boldsymbol{v}\right)^2\right]$$

Yet, since the $\boldsymbol{b}_j$ are orthonormal, we have $\sum_{j=0}^{2\kappa-1} \left(\boldsymbol{b}_j^\top \boldsymbol{v}\right)^2 = \|\boldsymbol{v}\|_2^2 = 1$ and, since $\boldsymbol{v}$ is uniformly distributed on the surface of the $(n-1)$-dimensional unit sphere, all the projections $\boldsymbol{b}_j^\top \boldsymbol{v}$ must be statistically indistinguishable. Overall $\mathbf{E}_{\boldsymbol{v}}\left[\left(\boldsymbol{b}_j^\top \boldsymbol{v}\right)^2\right] = 1/(2\kappa)$ for any $j = 0, \ldots, 2\kappa-1$ to yield

$$\gamma = \frac{1}{2\kappa}\sum_{j=0}^{2\kappa-1} \sigma_j^2(\boldsymbol{B}_{\cdot,K}) = \frac{1}{2\kappa}\mathrm{tr}(\boldsymbol{B}_{\cdot,K}^\top \boldsymbol{B}_{\cdot,K}) = \frac{1}{2\kappa}\sum_{j\in K} \left\|\boldsymbol{B}_{\cdot,j}\right\|_2^2 \tag{1.12}$$

where we have first exploited the fact that the singular values are the eigenvalues of $\boldsymbol{B}_{\cdot,K}^\top \boldsymbol{B}_{\cdot,K}$ and then that the diagonal of such a matrix contains the squared lengths of the columns of $\boldsymbol{B}_{\cdot,K}$, i.e., $\boldsymbol{B}_{\cdot,j}$ for $j \in K$. Note that, if the columns of $\boldsymbol{B}$ are normalized to a certain length ℓ then $\gamma = \ell^2$.

The value in (1.12) is such that, on average, $\|\Delta \boldsymbol{y}\|_2^2$ is γ times $\|\Delta \boldsymbol{\xi}_K\|_2^2$, though every instance of $\Delta \boldsymbol{\xi}_K$ may be *amplified* in a different way. To avoid pathological cases, one may think to require that the deviation of the actual energy gain from its average values is extremely limited.

Definition 1.5 A matrix $\boldsymbol{B}$ is said to enjoy the *Restricted Isometry Property* (RIP) for a certain sparsity level κ if there is a constant $\delta_{2\kappa} < 1$ such that

$$\gamma(1-\delta_{2\kappa}) \leq \frac{\|\boldsymbol{B}_{\cdot,K}\Delta\boldsymbol{\xi}_K\|_2^2}{\|\Delta\boldsymbol{\xi}_K\|_2^2} \leq \gamma(1+\delta_{2\kappa})$$

for every possible subset $K \subseteq \{0, \ldots, n-1\}$ of cardinality $|K| = 2\kappa$ and any $\Delta\boldsymbol{\xi}_K \in \mathbb{R}^{2\kappa}$. The constant $\delta_{2\kappa}$ is called the *Restricted Isometry Constant* (RIC).

Ideally, if $\delta_{2\kappa} = 0$, all 2κ-sparse vectors would experience the same energy gain and $\boldsymbol{B}_K$ would be a *restricted isometry* in the sense that the lengths of 2κ-sparse vectors are preserved up to a scaling factor $\sqrt{\gamma}$.

The non-average energy gain for the specific vector $\Delta\boldsymbol{\xi}_K$ can be written by defining the unit-length vector $\boldsymbol{v} = \Delta\boldsymbol{\xi}_K / \|\Delta\boldsymbol{\xi}\|_2$ to obtain

$$\|\boldsymbol{B}_{\cdot,K}\boldsymbol{v}\|_2^2 = \sum_{j=0}^{2\kappa-1} \sigma_j^2(\boldsymbol{B}_{\cdot,K}) \left(\boldsymbol{b}_j^\top \boldsymbol{v}\right)^2$$

that, if the singular values of $\boldsymbol{B}_K$ are $\sigma_0^2(\boldsymbol{B}_{\cdot,K}), \sigma_1^2(\boldsymbol{B}_{\cdot,K}), \ldots, \sigma_{2\kappa-1}^2(\boldsymbol{B}_{\cdot,K})$, can be bounded from above by $\sigma_{\max}^2(\boldsymbol{B}_{\cdot,K})$ and from below by $\sigma_{\min}^2(\boldsymbol{B}_{\cdot,K})$. With this and with (1.12), the requirement on $\boldsymbol{B}_{\cdot,K}$ can be translated in

$$1-\delta_{2\kappa} \leq \frac{\sigma_{\min}^2(\boldsymbol{B}_{\cdot,K})}{\frac{1}{2\kappa}\sum_{j=0}^{2\kappa-1}\sigma_j^2(\boldsymbol{B}_{\cdot,K})} \leq \frac{\sigma_{\max}^2(\boldsymbol{B}_{\cdot,K})}{\frac{1}{2\kappa}\sum_{j=0}^{2\kappa-1}\sigma_j^2(\boldsymbol{B}_{\cdot,K})} \leq 1+\delta_{2\kappa} \tag{1.13}$$

that allows to find the minimum possible value for $\delta_{2\kappa}$.

In our toy case the three matrices $\boldsymbol{B}_{\cdot,K}$ with $|K| = 2\kappa = 2$ are

$$\boldsymbol{B}_{\cdot,\{0,1\}} = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \end{pmatrix} \quad \boldsymbol{B}_{\cdot,\{0,2\}} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{6}} & \sqrt{\frac{2}{3}} \end{pmatrix} \quad \boldsymbol{B}_{\cdot,\{1,2\}} = \begin{pmatrix} -\frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{6}} & \sqrt{\frac{2}{3}} \end{pmatrix} \tag{1.14}$$

and none of them is singular. Moreover they all have the same singular values $\sigma_0^2(\boldsymbol{B}_{\cdot,K}) = 1$ and $\sigma_1^2(\boldsymbol{B}_{\cdot,K}) = 1/3$ so that (1.12) gives $\gamma = 2/3$ while (1.13) becomes $1 - \delta_{2\kappa} \leq 1/2 \leq 3/2 \leq 1 + \delta_{2\kappa}$ yielding $\delta_{2\kappa} = 1/2$. Hence, our projection preserves on the average 66% and never less than 33% of the energy of 2-sparse signals.

Clearly, when κ, m, and d have realistic values, the computation of the RIC becomes a task with a combinatorially increasing complexity linked to the number of matrices $\boldsymbol{B}_{\cdot,K}$ that is $\binom{d}{\kappa}$.

Be it easily computable or not, an objective in designing the sensing matrix $\boldsymbol{A}$ that tries to more carefully exploit the sparsity prior is to make the RIC of the resulting $\boldsymbol{B} = \boldsymbol{AD}$ as small as possible.

Note that, though originating from different perspective, coherence and RIP are all concerned with the features of the matrices $\boldsymbol{Z}_K = \boldsymbol{B}_{\cdot,K}^\top \boldsymbol{B}_{\cdot,K}$. In particular, if we assume that all the columns of $\boldsymbol{B}$ are normalized to the same unit length, then $\gamma = 1$ and the coherence is

$$\mu(\boldsymbol{B}) = \max_{j \neq l} \left| (\boldsymbol{Z}_{\{0,\ldots,2\kappa-1\}})_{j,l} \right|$$

while the RIC is

$$\delta_{2\kappa} = \max_{|K|=2\kappa} \max \left\{ \lambda_{\max}(\boldsymbol{Z}_K) - 1, 1 - \lambda_{\min}(\boldsymbol{Z}_K) \right\}$$

where $\lambda_j(\cdot)$ indicates the j-th eigenvalue of its matrix argument and $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ indicate the minimum and the maximum of the set of eigenvalues.

Since, independently of K, $|(\boldsymbol{Z}_K)_{j,k}| \leq \max_{j \neq l} \left| (\boldsymbol{Z}_{\{0,\ldots,2\kappa-1\}})_{j,l} \right|$ and since $\boldsymbol{Z}_K$ is a $2\kappa \times 2\kappa$ matrix, a straightforward application of the Gershgorin's circle theorem gives

$$\delta_{2\kappa} \leq 2\kappa \mu(\boldsymbol{B})$$

Though the bound is typically quite loose, it is often considered to be a hint to the fact that both coherence and RIC are able to push sensing matrices towards the structure needed to achieve good sensing performance.

1.5.1 Random Sensing Matrices

One cannot escape from observing that all the above design criteria for $\boldsymbol{A}$ are actually given on $\boldsymbol{B}$. Hence, for each class of signals, and thus of $\boldsymbol{D}$ on which sparsity is

verified, one should re-design $\boldsymbol{A}$ to meet the chosen criterion. What should ideally be pursued is the solution of one of the two optimization problems

$$\begin{aligned} &\arg\min_{\boldsymbol{A}\in\mathbb{R}^{m\times n}} \mu(\boldsymbol{B}) \\ &\text{s.t. } \boldsymbol{B}=\boldsymbol{A}\boldsymbol{D} \end{aligned} \tag{1.15}$$

or

$$\begin{aligned} &\arg\min_{\boldsymbol{A}\in\mathbb{R}^{m\times n}} \delta_{2\kappa}(\boldsymbol{B}) \\ &\text{s.t. } \boldsymbol{B}=\boldsymbol{A}\boldsymbol{D} \end{aligned} \tag{1.16}$$

where $\mu(\boldsymbol{B})$ and $\delta_{2\kappa}(\boldsymbol{B})$ indicate, respectively, the mutual coherence and the RIC of the columns of the matrix $\boldsymbol{B}$.

Both problems are too difficult to attack. The one in (1.15) is non-convex and can be tackled only by resorting to relaxations and approximations that makes the final sensing performance largely suboptimal. The one in (1.16) owes its hardness to the combinatorial nature of the definition of RIC and becomes unmanageable even for relatively small-scale instances.

This is why, the design flow of CS acquisition stages usually pursues a completely different path. The main idea is that, intuitively speaking, good sensing matrices are those that are *well spread* in the signal space. From this point of view what better strategy to achieve, at least on average, a good *spreading* than to use a random matrix?

We list here definitions that cope with this requirement.

Definition 1.6 We say that a $p \times q$ random matrix $\boldsymbol{M} \in \mathbb{R}^{pq}$ comes from a zero-mean, independent-row Random Gaussian Ensemble ($\boldsymbol{M} \sim \mathrm{RGE}(\mathscr{M})$) if there is a $q \times q$ symmetric, positive-definite matrix $\mathscr{M}$ such that every row of $\boldsymbol{M}$ is a jointly Gaussian random vector $\sim \mathrm{N}(0, \mathscr{M})$.

The probability density function of $\boldsymbol{M}$ is

$$f_{\mathrm{RGE}}(\boldsymbol{M}) = \frac{1}{\sqrt{(2\pi)^{pq}\det^{p}(\mathscr{M})}} e^{-\frac{1}{2}\mathrm{tr}(\mathscr{M}^{-1}\boldsymbol{M}^{\top}\boldsymbol{M})} \tag{1.17}$$

Clearly, (1.17) is a structured version of the probability density function of a pq-dimensional real jointly Gaussian random vector [10].

Note that, if $\boldsymbol{M}' \sim \mathrm{RGE}(\boldsymbol{I})$ where $\boldsymbol{I}$ is the $n \times n$ identity matrix, and if $\sqrt{\mathscr{M}}$ is the symmetric matrix such that $\sqrt{\mathscr{M}}\sqrt{\mathscr{M}} = \mathscr{M}$, then $\boldsymbol{M} = \boldsymbol{M}'\sqrt{\mathscr{M}} \sim \mathrm{RGE}(\mathscr{M})$. Hence, from the algorithmic point of view matrices $\boldsymbol{M} \sim \mathrm{RGE}(\mathscr{M})$ are very simple to generate.

Definition 1.7 We say that a $p \times q$ random matrix $\boldsymbol{M} \in \{-1, +1\}^{pq}$ comes from an independent-row, zero-mean Random Antipodal Ensemble ($\boldsymbol{M} \sim \mathrm{RAE}(\mathscr{M})$) if there is a $q \times q$ positive-definite matrix $\mathscr{M}$ such that every row of $\boldsymbol{M}$ is an antipodal, zero-mean random vector with correlation matrix $\mathscr{M}$.

Definition 1.8 We say that a $p \times q$ random matrix $\boldsymbol{M} \in \{-1, 0, +1\}^{pq}$ comes from an independent-row, zero-mean Random Ternary Ensemble ($\boldsymbol{M} \sim \mathrm{RTE}(\boldsymbol{\mathcal{M}})$) if there is a $q \times q$ positive-definite matrix $\boldsymbol{\mathcal{M}}$ such that every row of $\boldsymbol{M}$ is a ternary, zero-mean random vector with correlation matrix $\boldsymbol{\mathcal{M}}$.

Definition 1.9 We say that a $p \times q$ random matrix $\boldsymbol{M} \in \{0, 1\}^{pq}$ comes from an independent-row Random Binary Ensemble ($\boldsymbol{M} \sim \mathrm{RBE}(\boldsymbol{\mathcal{M}})$) if there is a $q \times q$ positive-definite matrix $\boldsymbol{\mathcal{M}}$ such that every row of $\boldsymbol{M}$ is a binary random vector with correlation matrix $\boldsymbol{\mathcal{M}}$.

Important special cases of the above definitions are the ensembles in which entries are independent and identically distributed (iid) that will be indicated by RGE (iid), RAE (iid), RTE (iid), and RBE (iid).

Note that since for zero-mean Gaussian and balanced antipodal and binary random variables incovariance is equivalent to independence, we have $\mathrm{RGE}(\mathrm{iid}) = \mathrm{RGE}(\boldsymbol{I})$, $\mathrm{RAE}(\mathrm{iid}) = \mathrm{RAE}(\boldsymbol{I})$, and $\mathrm{RBE}(\mathrm{iid}) = \mathrm{RBE}\left(\boldsymbol{I}/4 + \mathbf{1}\mathbf{1}^{\top}/4\right)$ where $\boldsymbol{I}$ is the $q \times q$ identity matrix and $\mathbf{1} = (1, \ldots, 1)^{\top}$. The iid versions of the above ensembles have noteworthy asymptotic properties with respect to coherence and RIP.

By adapting the results in [3, 12] we get

Theorem 1.1 *If the $p \times q$ matrix $\boldsymbol{M}$ is made of iid entries with $\mathbf{E}[\boldsymbol{M}_{j,l}] = 0$ and $\mathbf{E}[\boldsymbol{M}_{j,l}^2] = \sigma_{\boldsymbol{M}}^2$, and $p, q \to \infty$ with $\log q = \mathsf{O}(p)$, then*

$$\mu(\boldsymbol{M}) = \mathsf{O}\left(\frac{\log q}{p}\right) \tag{1.18}$$

In (1.18), the number p of the degrees of freedom of the space hosting the vectors plays a different role with respect to number q of the vector to spread in that space. Yet, if the number of vectors to stuff does not increase more than exponentially with their dimensionality, the coherence vanishes when the size of the matrix increases.

As far as RIP is concerned we may resort to the classical result of Marchenko e Pastur [16] that can be adapted to give

Theorem 1.2 *If the $p \times q$ matrix $\boldsymbol{M}$ is made of iid entries with $\mathbf{E}[\boldsymbol{M}_{j,l}] = 0$ and $\mathbf{E}[\boldsymbol{M}_{j,l}^2] = \sigma_{\boldsymbol{M}}^2$, and $p, q \to \infty$ with $q/p \to r$ and $0 < r < 1$, then the squares of the singular values of $\boldsymbol{M}/\sqrt{p}$ asymptotically distribute according to*

$$f_{\mathrm{MP}}(\xi) = \begin{cases} \frac{\sqrt{(\xi - r^-)(r^+ - \xi)}}{2\sigma_{\boldsymbol{M}}^2 \pi r \xi} & \text{if } r^- \le \xi \le r^+ \\ 0 & \text{otherwise} \end{cases}$$

with $r^{\pm} = \sigma_{\boldsymbol{M}}^2 \left(1 \pm \sqrt{r}\right)^2$.

Note that the $\sigma_{\boldsymbol{M}}^2$ correction is effective only in the RTE (iid) case for which we may have $\sigma_{\boldsymbol{M}}^2 < 1$ since in the other cases we have $\sigma_{\boldsymbol{M}}^2 = 1$.

Theorem 1.2 clearly bounds the minimum and maximum singular values of $\boldsymbol{M}/\sqrt{p}$ with $\sigma^2_{\min}(\boldsymbol{M}/\sqrt{p}) \geq r^-$ and $\sigma^2_{\max}(\boldsymbol{M}/\sqrt{p}) \leq r^+$. Moreover, we know that the average of the squared singular values of $\boldsymbol{M}/\sqrt{p}$ is

$$\mathbf{E}\left[\sigma_j^2(\boldsymbol{M}/\sqrt{p})\right] = \int_{r^-}^{r^+} \xi f_{\mathrm{MP}}(\xi)\mathrm{d}\xi = \sigma^2_{\boldsymbol{M}}$$

We may now think that $\boldsymbol{M}$ is one of the $m \times 2\kappa$ submatrices $\boldsymbol{B}_{\cdot,K}$ to set $p = m$, $q = 2\kappa$ and assume that, if $2\kappa/m \to r$ with $0 < r < 1$.

Since in general $\sigma_j^2(\boldsymbol{M}) = p\sigma_j^2(\boldsymbol{M}/\sqrt{p})$ we may asymptotically estimate the inner terms in (1.13) as

$$1 - \delta_{2\kappa} \leq \left(1 - \sqrt{r}\right)^2 < \left(1 + \sqrt{r}\right)^2 \leq 1 + \delta_{2\kappa}$$

that allows a RIC as small as

$$\delta_{2\kappa} = \max\left\{\left(1 + \sqrt{r}\right)^2 - 1, 1 - \left(1 - \sqrt{r}\right)^2\right\} \tag{1.19}$$

Hence, by keeping m sufficiently larger than 2κ one may ensure that, for matrices large enough, a small RIC is achieved.

Other approaches exist (see, e.g., [1, 7, chapter 5]) that overcome some of the limitations of Theorem 1.2 and allow the estimation of RICs in non-asymptotic conditions and without the need of iid matrices. Yet, as far as the magnitude of the RIC is concerned, none of these more sophisticated machineries offers a definite advantage over (1.19) that remains a state-of-the-art estimation of what can be guaranteed for random choices of $\boldsymbol{B}$.

From the intuitive point of view, such guarantees should at least approximately extend to $\boldsymbol{B} = \boldsymbol{AD}$ when $\boldsymbol{A}$ is drawn from one of the ensembles defined above since if $\boldsymbol{A}$ is random, then also $\boldsymbol{B}$ is random though with a difference coherence and a different RIC.

To pursue this direction a little further we need to define the concept of sub-Gaussian norm and sub-Gaussian random variables and vectors relaying on the definition and on the results in [7, chapter 5].

Definition 1.10 For any real random variable α the quantity

$$\|\alpha\|_{\mathrm{sG}} = \sup_{t \geq 1} \frac{1}{\sqrt{t}} \mathbf{E}^{1/t}\left[|\alpha|^t\right]$$

is called the *sub-Gaussian norm* of α. For any real random vector $\boldsymbol{\alpha}$, the sub-Gaussian norm is defined as

$$\|\boldsymbol{\alpha}\|_{\mathrm{sG}} = \sup_{\|\boldsymbol{\beta}\|_2 = 1} \left\|\boldsymbol{\beta}^\top \boldsymbol{\alpha}\right\|_{\mathrm{sG}}$$

where $\boldsymbol{\beta}$ is any deterministic, unit-length vector.

The random quantities α and $\boldsymbol{\alpha}$ are said to be *sub-Gaussian* when their sub-Gaussian norm is finite, i.e., when $\|\alpha\|_{\mathrm{sG}} < \infty$ and $\|\boldsymbol{\alpha}\|_{\mathrm{sG}} < \infty$.

Sub-Gaussian random variables and vectors are generalization of Gaussian random variables and vectors whose construction preserves the properties and the tail probability decay needed for the measure to concentrate when the dimensionality increases.

Sub-Gaussianity may be exploited to bracket singular values of random matrix without requiring that its entries are independent.

Theorem 1.3 *If the $p \times q$ matrix $\boldsymbol{M}$ is made of independent sub-Gaussian rows $\boldsymbol{M}_{j,\cdot}$ with the same correlation matrix $\mathbf{E}[\boldsymbol{M}_{j,\cdot}\boldsymbol{M}_{j,\cdot}^{\top}] = \mathscr{M}$, then there are two constants $C, c > 0$ depending only on $\max_{j=0,\ldots,q-1} \left\|\boldsymbol{M}_{\cdot,j}\right\|_{\mathrm{sG}}$ such that if*

$$\delta = C\sqrt{\frac{q}{p}} + \frac{\tau}{\sqrt{p}}$$

then for every $\tau > 0$

$$\sigma_{\max}\left(\frac{1}{p}\boldsymbol{M}^{\top}\boldsymbol{M} - \mathscr{M}\right) \leq \max\{\delta, \delta^2\}$$

holds with probability at least $1 - 2e^{-c\tau^2}$.

Here, we are not overly interested in the non-asymptotic nature of Theorem 1.3 and thus assume $p, q \to \infty$ and $q/p \to r$ with $0 < r < 1$ as before. This allows to take τ arbitrarily large and obtain

$$\sigma_{\max}\left(\frac{1}{p}\boldsymbol{M}^{\top}\boldsymbol{M} - \mathscr{M}\right) \leq \Delta \tag{1.20}$$

with probability 1 for $\Delta = \max\{C\sqrt{r}, C^2 r\}$.

For any two symmetric and positive-semidefinite matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ (in our case $\boldsymbol{P} = \boldsymbol{M}^{\top}\boldsymbol{M}/p$ and $\boldsymbol{Q} = \mathscr{M}$) the matrix $\boldsymbol{P} - \boldsymbol{Q}$ is symmetric (though not necessarily positive semidefinite) and its singular values coincide with the absolute values of the eigenvalues. Hence, (1.20) can be translated into

$$\left|\lambda_j\left(\boldsymbol{P} - \boldsymbol{Q}\right)\right| \leq \Delta \quad \text{for } j = 0, \ldots, q-1 \tag{1.21}$$

Moreover, any $q \times q$ symmetric matrix $\boldsymbol{R}$ can be written as $\boldsymbol{R} = \sum_{j=0}^{q-1} \lambda_j(\boldsymbol{R})\boldsymbol{e}_j\boldsymbol{e}_j^{\top}$ where $\boldsymbol{e}_j$ are its orthonormal eigenvectors. Hence, given any unit-length vector $\boldsymbol{s}$ we have $\boldsymbol{s}^{\top}\boldsymbol{R}\boldsymbol{s} = \sum_{j=0}^{q-1} \lambda_j(\boldsymbol{R}) \left(\boldsymbol{e}_j^{\top}\boldsymbol{s}\right)^2$ where the coefficients $\left(\boldsymbol{e}_j^{\top}\boldsymbol{s}\right)^2$ are positive and sum to $\|\boldsymbol{s}\|_2^2 = 1$ and thus produce a convex combination of the eigenvalues. Since bounding the absolute value of a set of quantities is equivalent to bounding all their convex combinations, (1.21) is equivalent to bounding $\left|\boldsymbol{s}^{\top}(\boldsymbol{P} - \boldsymbol{Q})\boldsymbol{s}\right|$ and thus to

$$\left|\boldsymbol{s}^{\top}\boldsymbol{P}\boldsymbol{s} - \boldsymbol{s}^{\top}\boldsymbol{Q}\boldsymbol{s}\right| \leq \Delta \quad \forall\, \|\boldsymbol{s}\|_2 = 1 \tag{1.22}$$

This implies $|\lambda_{\max}(\boldsymbol{P}) - \lambda_{\max}(\boldsymbol{Q})| \leq \Delta$. In fact we may proceed by contradiction and assume, by possibly exchanging $\boldsymbol{P}$ with $\boldsymbol{Q}$, that $\lambda_{\max}(\boldsymbol{Q}) < \lambda_{\max}(\boldsymbol{P}) - \Delta$, a strict upper bound that holds for all the eigenvalues of $\boldsymbol{Q}$ and thus for all their convex combinations $\boldsymbol{s}^\top \boldsymbol{Q}\boldsymbol{s}$. Yet, a unit-length vector $\boldsymbol{s}_{\max}$ exists such that $\lambda_{\max}^\top(\boldsymbol{P}) = \boldsymbol{s}_{\max}\boldsymbol{P}\boldsymbol{s}_{\max}$ so that (1.22) is violated for $\boldsymbol{s} = \boldsymbol{s}_{\max}$.

An analogous argument ensures that $|\lambda_{\min}(\boldsymbol{P}) - \lambda_{\min}(\boldsymbol{Q})| \leq \Delta$. In fact, if it was not true and $\lambda_{\min}(\boldsymbol{P}) > \lambda_{\min}(\boldsymbol{Q}) + \Delta$ the same strict lower bound applies to all the eigenvalues of $\boldsymbol{P}$ and thus to all their convex combinations $\boldsymbol{s}^\top \boldsymbol{P}\boldsymbol{s}$. Yet, a unit-length vector $\boldsymbol{s}_{\min}$ exists such that $\lambda_{\min}(\boldsymbol{Q}) = \boldsymbol{s}_{\min}^\top \boldsymbol{Q}\boldsymbol{s}_{\min}$ so that (1.22) is violated for $\boldsymbol{s} = \boldsymbol{s}_{\min}$.

We may now go back to $\boldsymbol{P} = \boldsymbol{M}^\top \boldsymbol{M}/p$ and $\boldsymbol{Q} = \mathscr{M}$, so that $\lambda_j(\boldsymbol{P}) = \sigma_j^2(\boldsymbol{M}/\sqrt{p})$ and $\lambda_j(\boldsymbol{Q}) = \lambda_j(\mathscr{M})$, to obtain

$$\lambda_{\min}(\mathscr{M}) - \Delta \leq \sigma_{\min}^2(\boldsymbol{M}/\sqrt{p}) \leq \sigma_{\max}^2(\boldsymbol{M}/\sqrt{p}) \leq \lambda_{\max}(\mathscr{M}) + \Delta$$

In addition to this consequence of Theorem 1.3 we also have information on the average of the singular values of $\boldsymbol{M}/\sqrt{p}$. In fact,

$$\frac{1}{q}\sum_{j=0}^{q-1} \sigma_j^2(\boldsymbol{M}/\sqrt{p}) = \frac{1}{q}\mathrm{tr}\left(\frac{1}{p}\boldsymbol{M}^\top \boldsymbol{M}\right) = \frac{1}{q}\sum_{j=0}^{q-1}\frac{1}{p}\sum_{l=0}^{p-1} \boldsymbol{M}_{l,j}^2 \to \frac{1}{q}\mathrm{tr}(\mathscr{M})$$

where we have exploited the fact that the rows are independent and $\mathbf{E}[\boldsymbol{M}_{l,j}^2] = \mathscr{M}_{j,j}$ to apply the law of large numbers.

We may now think that $\boldsymbol{M}$ is one of the $m \times 2\kappa$ submatrices $\boldsymbol{B}_{\cdot,K}$ whose rows have a $2\kappa \times 2\kappa$ second-order statistics $\mathscr{B}$ to set $p = m$, $q = 2\kappa$ and assume that, if $2\kappa/m \to r$ with $0 < r < 1$.

Since in general $\sigma_j^2(\boldsymbol{M}) = p\sigma_j^2(\boldsymbol{M}/\sqrt{p})$ we may asymptotically estimate the inner terms in (1.13) as

$$1 - \delta_{2\kappa} \leq \frac{\lambda_{\min}(\mathscr{B}) - \Delta}{\lambda_{\mathrm{ave}}(\mathscr{B})} \leq \frac{\lambda_{\max}(\mathscr{B}) + \Delta}{\lambda_{\mathrm{ave}}(\mathscr{B})} \leq 1 + \delta_{2\kappa}$$

where $\lambda_{\mathrm{ave}}(\mathscr{B}) = \mathrm{tr}(\mathscr{B})/q$ is the average of the eigenvalues of $\mathscr{B}$. Given the above bound, $\delta_{2\kappa}$ can be taken at least as small as

$$\delta_{2\kappa} = \max\left\{\frac{\lambda_{\min}(\mathscr{B}) - \Delta}{\lambda_{\mathrm{ave}}(\mathscr{B})} - 1, 1 - \frac{\lambda_{\max}(\mathscr{B}) + \Delta}{\lambda_{\mathrm{ave}}(\mathscr{B})}\right\} \tag{1.23}$$

Since $\Delta = \max\{C\sqrt{r}, C^2 r\}$, by keeping m sufficiently larger than 2κ and assuming that the sub-Gaussian norm of the rows of $\boldsymbol{B}$ is not too large (i.e., the associated PDF allows a sufficiently fast concentration of measure) the RIC is still under control.

The fact that the choice of a particular PDF is no longer fundamental but we may rely on results that hold for a class of PDFs also helps addressing the fact that

$\boldsymbol{B} = \boldsymbol{AD}$ and we are not designing $\boldsymbol{B}$ directly. Actually, when either $\boldsymbol{A} \sim \text{RGE}(\mathscr{A})$, or $\boldsymbol{A} \sim \text{RAE}(\mathscr{A})$, or $\boldsymbol{A} \sim \text{RTE}(\mathscr{A})$, or $\boldsymbol{A} \sim \text{RBE}(\mathscr{A})$ for a certain $\boldsymbol{A}$, it is easy to prove that, if $\boldsymbol{D}$ is an orthonormal basis, $\boldsymbol{B} = \boldsymbol{AD}$ is made of independent sub-Gaussian rows whose second-order statistics is $\mathscr{B} = \boldsymbol{D}^\top \mathscr{A} \boldsymbol{D}$.

This covers the random-$\boldsymbol{A}$ approach with some reasonable guarantees that the resulting measurements vector can be used to retrieve $\boldsymbol{\xi}$ and thus $\boldsymbol{x}$.

To illustrate this point with an example, assume that $\boldsymbol{A} \sim \text{RGE}(\mathscr{A})$ or $\boldsymbol{A} \sim \text{RAE}(\mathscr{A})$ with $\mathscr{A}_{j,l} = \omega^{|j-l|}$ that, for $0 \leq \omega < 1$ corresponds to a unit-power smoothly low-pass process with power spectrum

$$\Psi(f) = \frac{1-\omega^2}{1+\omega^2-2\omega\cos(2\pi f)}$$

with $\omega = 0$ implying a flat spectrum and thus the iid version of our ensembles.

Gaussian $\boldsymbol{A}$ are generated by generating a matrix $\boldsymbol{M} \in \text{RGE (iid)}$ and setting $\boldsymbol{A} = \sqrt{\mathscr{A}}\boldsymbol{M}$. Antipodal $\boldsymbol{A}$ are generated by setting $\boldsymbol{A}_{j,0} = \pm 1$ with $\Pr\{\boldsymbol{A}_{j,0} = +1\} = 1/2$ and then all the subsequent entries of each row iteratively with $\Pr\{\boldsymbol{A}_{j,l-1}\boldsymbol{A}_{j,l} > 0\} = (1+\omega)/2$ for $l = 1, \ldots, n-1$.

The sparsity basis $\boldsymbol{D}$ is taken as the DCT type II orthonormal basis $\boldsymbol{D}_{j,l} = \cos[\pi j(l+1/2)/d]$. From the orthonormality of $\boldsymbol{D}$ we get that $\boldsymbol{A} \sim \text{RGE}(\mathscr{A})$ implies $\boldsymbol{B} = \boldsymbol{AD} \sim \text{RGE}\left(\sqrt{\boldsymbol{D}}^\top \mathscr{A} \sqrt{\boldsymbol{D}}\right)$ is made of (sub-)Gaussian independent rows. In the $\boldsymbol{A} \sim \text{RAE}(\mathscr{A})$ case, each row of $\boldsymbol{B} = \boldsymbol{AD}$ is such that $\|\boldsymbol{B}_{j,\cdot}\|_2^2 = n$ and thus is a bounded sub-Gaussian vector.

We also set $m = 128$, $n = 256$, and $\kappa = 3$ to perform a Montecarlo estimation of the PDF of the singular value of $\boldsymbol{B}_{\cdot,K}$ normalized to their own average

$$\hat{\sigma}_j^2(\boldsymbol{B}_{\cdot,K}) = \frac{\sigma_j(\boldsymbol{B}_{\cdot,K})}{\frac{1}{2\kappa}\sum_{l=0}^{2\kappa-1}\sigma_l^2(\boldsymbol{B}_{\cdot,K})}$$

The results are reported in Fig. 1.9a and b for $\boldsymbol{A} \sim \text{RGE}(\mathscr{A})$ and $\boldsymbol{A} \sim \text{RAE}(\mathscr{A})$, respectively, and for $\omega = 0, 0.1, 0.2$. Note how the profiles are extremely similar regardless of the different ensembles from which $\boldsymbol{A}$ is taken, thus confirming that what really matter is the sub-Gaussian nature of the vectors. In both cases the PDF spreads out as ω increases, i.e., as the gap between $\lambda_{\min}(\mathscr{A})$ and $\lambda_{\max}(\mathscr{A})$ increases as shown in Fig. 1.10.

Starting from the same data, Fig. 1.11a and b shows the profiles of the probability that the RIC falls below a certain threshold. The spreading of the above PDF is reflected in the worsening of this statistic, though values of $\delta_{2\kappa}$ less than $1/2$ are obtained approximately 70% of the times even with $\omega = 0.2$.

Actually, the next section shows that coherence and RIC enter the bound on the reconstruction error made by algorithms that compute $\boldsymbol{\xi}$ from $\boldsymbol{y}$. Regrettably, such bounds are almost often too loose that there is no monotonic relationship between coherence and RIC and the true reconstruction performance.

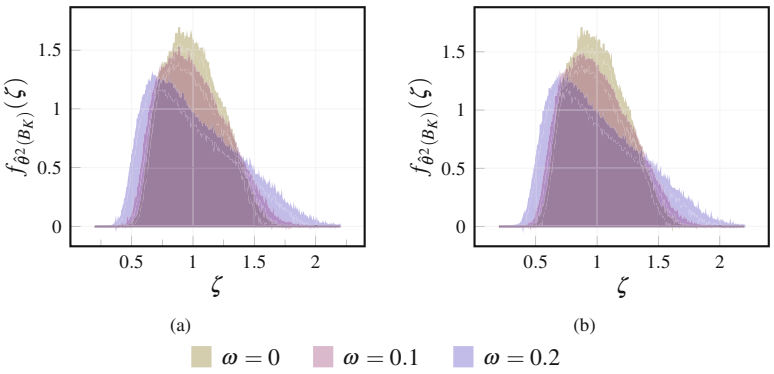


Fig. 1.9 The PDF of the singular values of $\boldsymbol{B}_{\cdot,K}$ for $m = 128$, $n = d = 256$, $\boldsymbol{D}$ the DCT orthonormal basis and (**a**) $\boldsymbol{A} \sim \mathrm{RGE}(\mathscr{A})$, (**b**) $\boldsymbol{A} \sim \mathrm{RAE}(\mathscr{A})$ with $\mathscr{A}_{j,l} = \omega^{|j-l|}$

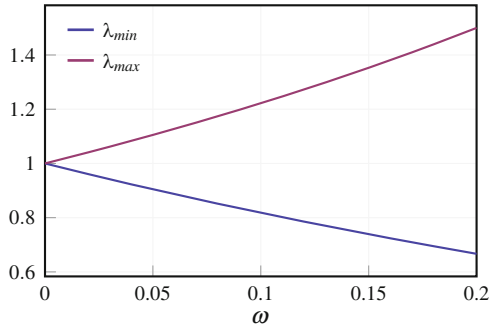


Fig. 1.10 The minimum and maximum eigenvalues of a 256×256 correlation matrix $\mathscr{A}_{j,l} = \omega^{|j-l|}$ as a function of ω

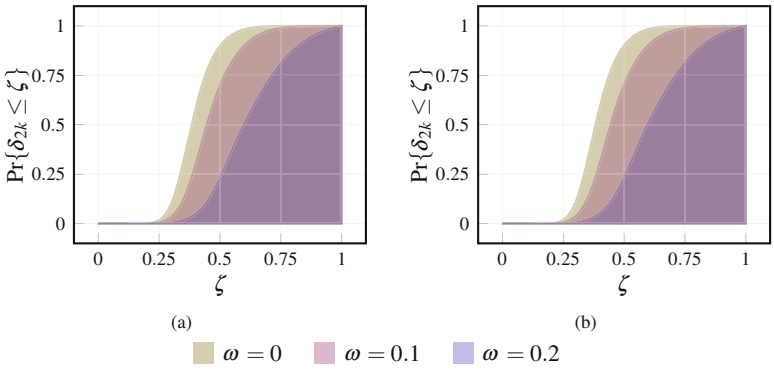


Fig. 1.11 The probability that a system with (**a**) $\boldsymbol{A} \sim \mathrm{RGE}(\mathscr{A})$, (**b**) $\boldsymbol{A} \sim \mathrm{RAE}(\mathscr{A})$ with $\mathscr{A}_{j,l} = \omega^{|j-l|}$, yields a $\boldsymbol{B} = \boldsymbol{AD}$ sensing matrix with a RIC not larger than a certain value

Equation (1.23), for example, and the evidence in Figs. 1.9 and 1.11 imply that adopting non-isotropic rows with $\mathscr{B} \neq \boldsymbol{I}$ increases the spread of the eigenvalues and worsens the RIC apparently hinting at a decay in performance.

Indeed this may be true only if nothing is known about the original signal but the fact that it is sparse and the worst-case analysis that underlies the concepts of coherence and RIP is a good proxy of what may happen in an actual acquisition.

Yet, the adaptation technique introduced in the next chapters designs possibly non-diagonal correlation matrices $\mathscr{A}$ and exploits one of the above ensembles to generate sensing matrices with independent rows whose statistics substantially optimizes acquisition performance.

Therefore, though they are important technical tools to establish guarantees, merit figures like coherence and RIC can hardly be considered a design criterion when things come to performance optimization, especially in the presence of further information on the original signal, leaving the general idea of making $\boldsymbol{A}$ random as the unique practical guideline.

1.6 Signal Reconstruction

Signal reconstruction must address the problem of solving (1.3), i.e.,

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\eta} = \boldsymbol{B}\boldsymbol{\xi} + \boldsymbol{\eta} \tag{1.24}$$

to find $\boldsymbol{\xi}$ while $\boldsymbol{\eta}$ is a disturbance that may remain unknown except from the fact that it is bounded by a certain maximum energy $\|\boldsymbol{\eta}\|_2^2 \leq \epsilon^2$ that is possibly small with respect to $\|\boldsymbol{y}\|_2^2$. If this assumption holds, and in absence of further information on $\boldsymbol{\xi}$, any point in the cylinder

$$\|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\xi}\|_2 \leq \epsilon \tag{1.25}$$

is a candidate solution. Actually, we know that $\boldsymbol{\xi}$ is κ-sparse and assume that $\boldsymbol{B}$ has been designed to take advantage of this prior. The $\boldsymbol{\xi}$ we are looking for should satisfy both (1.25) and $\|\boldsymbol{\xi}\|_0 \leq \kappa$ where the usual definition of $\|\boldsymbol{\xi}\|_p = \left(\sum_{j=0}^{d-1} \boldsymbol{\xi}_j^p\right)^{1/p}$ for $p > 0$ is extended to $\|\boldsymbol{\xi}\|_0 = |\text{supp}(\boldsymbol{\xi})|$ (the number of nonzero elements in $\boldsymbol{\xi}$) that is only a pseudo-norm since it does not scale when its argument is scaled, i.e., it is not a homogeneous function.

Actually, a proper design of $\boldsymbol{B}$ should guarantee that the κ-sparse element of (1.25) is unique. Though this is usually an implicit assumption, it can be made formal in the noiseless $\epsilon = 0$ case [5, 9].

Theorem 1.4 *Let* $\bar{\boldsymbol{\xi}}$ *be such that* $\boldsymbol{y} = \boldsymbol{B}\bar{\boldsymbol{\xi}}$ *and* $\left\|\bar{\boldsymbol{\xi}}\right\|_0 = \kappa$. *If*

$$\kappa < \frac{1}{2}\left(1 + \frac{1}{\mu(\boldsymbol{B})}\right)$$

then for any $\boldsymbol{\xi} \neq \bar{\boldsymbol{\xi}}$ *such that* $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$ *we have* $\|\boldsymbol{\xi}\|_0 > \kappa$.

Guarantees like the one above allow us to identify the solution of our reconstruction problem with the solution of the following minimization problem:

$$\begin{aligned} &\arg\min_{\boldsymbol{\xi}\in\mathbb{R}^d} \|\boldsymbol{\xi}\|_0 \\ &\quad \text{s.t. } \|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\xi}\|_2 \leq \epsilon \end{aligned} \tag{1.26}$$

Regrettably, (1.26) is a non-convex optimization problem due to the non-convex behavior of $\|\cdot\|_0$. In fact, if one plots, for example, the set of points $\boldsymbol{\xi} \in \mathbb{R}^3$ such that $\|\boldsymbol{\xi}\|_0 \leq 1$ obtains Fig. 1.12 that clearly shows a non-convex set.

Non-convex optimization problems are difficult to solve and, in particular, the discrete structure of $\|\cdot\|_0$ implies a combinatorial search that makes (1.26) an NP-hard problem [17], i.e., something that nobody wants to include in a signal processing chain.

A possible source of inspiration to cope with this comes from Fig. 1.13. If we define $S_d^p(r) = \{\boldsymbol{\xi} \,|\, \boldsymbol{\xi} \in \mathbb{R}^d \wedge \|\boldsymbol{\xi}\|_p \leq r\}$, Fig. 1.13 shows $S_3^p(1)$ for different values of $p > 0$. Visual inspection suggests that the lower the p the better $S_3^p(1)$ approximates $S_3^0(1)$ in Fig. 1.12. More formally, one can see that $S_d^p(1) \to S_d^0(1) \cap S_d^2(1)$ as $p \to 0$, where $S_d^0(1)$ is clipped by $S_d^2(1)$ to cope with the non-homogeneity of $\|\cdot\|_0$.

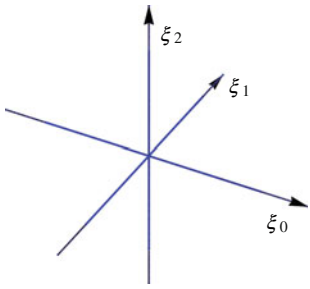


Fig. 1.12 The set of points $\boldsymbol{\xi} \in \mathbb{R}^3$ such that $\|\boldsymbol{\xi}\|_0 \leq 1$

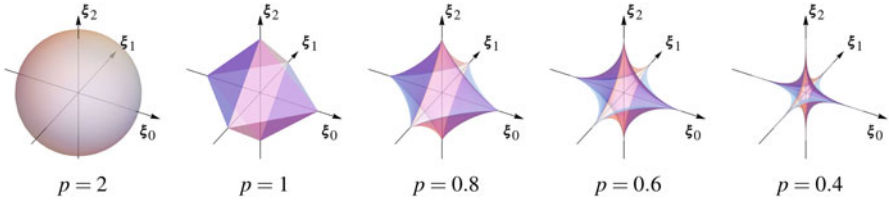


Fig. 1.13 Three-dimensional spheres with equation $\|\boldsymbol{\xi}\|_p \leq 1$

Starting from these considerations, we are tempted to approximate $\|\cdot\|_0$ with $\|\cdot\|_p$ with a p as small as possible or, formally speaking, the smallest p yielding a convex merit figure, that is $p = 1$. Hence, (1.26) becomes

$$\begin{aligned} &\arg\min_{\boldsymbol{\xi}\in\mathbb{R}^d} \|\boldsymbol{\xi}\|_1 \\ &\text{s.t. } \|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\xi}\|_2 \leq \epsilon \end{aligned} \tag{1.27}$$

that is usually indicated as Basis-Pursuit with DeNoising (BPDN), with a noiseless version featuring $\epsilon = 0$ named simply Basis Pursuit (BP).

To illustrate how BP and BPDN may succeed in reconstructing $\boldsymbol{\xi}$ from $\boldsymbol{y}$ we may go back to our original example of Figs. 1.6 and 1.7 and assume to be given the measurement vector $\boldsymbol{y}'''$ on the plane θ. If $\epsilon = 0$ in (1.27), then the constraint is $\boldsymbol{y}''' = \boldsymbol{B}\boldsymbol{\xi}$ requiring that $\boldsymbol{\xi}$ stays in the 1-dimensional subspace of points that have the same projection $\boldsymbol{y}'''$ on θ. Such a subspace is the thick blue line that extends from $\boldsymbol{y}'''$ perpendicularly to θ in Fig. 1.14a. Among all the points of that line, (1.27) chooses the one on the *sphere* $S_3^1(r)$ with the least possible r. Due to the *peaky* shape of $S_3^1(r)$ that tends to protrude along the coordinate axes, this yields the point $\hat{\boldsymbol{\xi}}$ that is the true original signal.

In the noisy case, $\boldsymbol{y}'''$ is not the projection of $\boldsymbol{\xi}'''$ on θ. Yet, the set of feasible solutions $\|\boldsymbol{y}''' - \boldsymbol{B}\boldsymbol{\xi}\|_2 \leq \epsilon$ is a cylinder with axis $\boldsymbol{y}''' = \boldsymbol{B}\boldsymbol{\xi}$ that includes the true signal $\boldsymbol{\xi}'''$ as shown in Fig. 1.14b. Again, the shape of $S_3^1(r)$ is such that the solution of (1.27) is a sparse vector $\hat{\boldsymbol{\xi}}$ with the same nonzero component as $\boldsymbol{\xi}'''$, though the presence of noise makes $\hat{\boldsymbol{\xi}} \neq \boldsymbol{\xi}'''$.

This intuitive mechanism works in higher dimensions as confirmed by some classical theorems on the issue.

The first theorem leverages on coherence and ensures the reconstruction of $\boldsymbol{\xi}$ that are exactly κ-sparse.

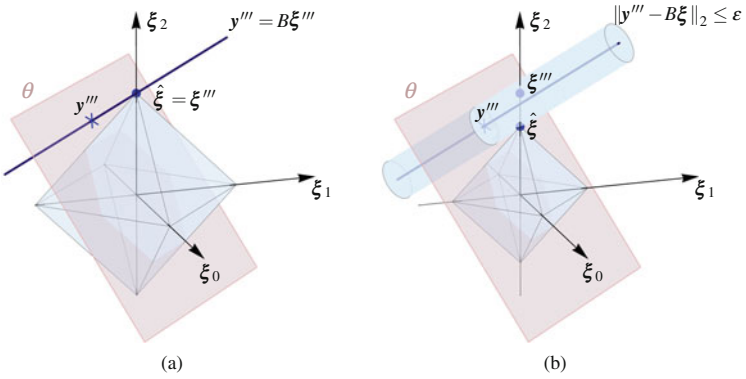


Fig. 1.14 Retrieval of $\boldsymbol{\xi}'''$ in the setting of Figs. 1.6 and 1.7 by means of BP (**a**) and BPDN (**b**)

Theorem 1.5 ([9]) *If $\mu(\boldsymbol{B}) < 1/(2\kappa - 1)$ and there is a $\boldsymbol{\xi}$ such that $\|\boldsymbol{\xi}\|_0 = \kappa$ and $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$, then the solution $\hat{\boldsymbol{\xi}}$ of (1.27) with $\epsilon = 0$ is such that $\hat{\boldsymbol{\xi}} = \boldsymbol{\xi}$.*

Theorem 1.5 can be combined with Theorem 1.1 to give a rough estimation of how many measurements are needed to achieve signal reconstruction. In fact, if we use m measurements of an n-dimension signal, Theorem 1.1 allows to estimate the coherence of the matrix used for sensing as $\mu = \mathsf{O}\left(\frac{\log n}{m}\right)$. Following Theorem 1.5, reconstruction can be guaranteed if such a coherence is $\mu < 1/(2\kappa - 1)$ where κ is the sparsity. Overall, we may estimate that to effectively reconstruct an n-dimensional signal that is κ-sparse one should deploy a number of measurements of the order

$$m^* = \mathsf{O}\left(\kappa \log(n)\right) \tag{1.28}$$

Actually, a different path of reasoning exploiting RIP-related considerations [2] that are out of the scope of this short rehearsal of the theoretical foundations of CS suggests the slightly more favorable trend

$$m^* = \mathsf{O}\left(\kappa \log\left(\frac{n}{\kappa}\right)\right) \tag{1.29}$$

that is often employed for a rough sizing of CS systems.

Going back to reconstruction performance, a different, more general result admits that $\boldsymbol{\xi}$ is not exactly κ-sparse and applies to the larger class of original signals that have a good κ-sparse approximation. This is formalized by defining a thresholded version $\boldsymbol{\xi}^{\kappa\uparrow}$ of $\boldsymbol{\xi}$ that retains only the κ largest components of $\boldsymbol{\xi}$.

Theorem 1.6 ([4]) *If the RIC of $\boldsymbol{B}$ is $\delta_{2\kappa} < \sqrt{2} - 1$ and there is a $\boldsymbol{\xi}$ such that $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$, then the solution $\hat{\boldsymbol{\xi}}$ of (1.27) with $\epsilon = 0$ is such that*

$$\left\|\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}\right\|_2 \leq 2\frac{1 + (\sqrt{2} - 1)\delta_{2\kappa}}{1 - (\sqrt{2} + 1)\delta_{2\kappa}} \frac{\left\|\boldsymbol{\xi} - \boldsymbol{\xi}^{\kappa\uparrow}\right\|_1}{\sqrt{\kappa}} \tag{1.30}$$

For $\boldsymbol{\xi}$ that are exactly κ-sparse we have $\left\|\boldsymbol{\xi} - \boldsymbol{\xi}^{\kappa\uparrow}\right\|_1 = 0$ and thus a perfect reconstruction of the original signal.

Some guarantees also hold for the noisy case.

Theorem 1.7 ([4]) *If the RIC of $\boldsymbol{B}$ is $\delta_{2\kappa} < \sqrt{2} - 1$ and there is a $\boldsymbol{\xi}$ such that $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi} + \eta$, with $\|\eta\|_2 \leq \epsilon$, then the solution $\hat{\boldsymbol{\xi}}$ of (1.27) is such that*

$$\left\|\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}\right\|_2 \leq 2\frac{1 + (\sqrt{2} - 1)\delta_{2\kappa}}{1 - (\sqrt{2} + 1)\delta_{2\kappa}} \frac{\left\|\boldsymbol{\xi} - \boldsymbol{\xi}^{\kappa\uparrow}\right\|_1}{\sqrt{\kappa}} + 4\frac{\sqrt{1 + \delta_{2\kappa}}}{1 - (\sqrt{2} + 1)\delta_{2\kappa}}\epsilon \tag{1.31}$$

Equation (1.31) bounds reconstruction error with the sum of two terms, the first of which is the same as in Theorem 1.6 as it refers to the possibility of reconstruction

non-perfectly κ-sparse signals, while the second related the uncertainty on the measurement $\boldsymbol{y}$ with the uncertainty on the reconstructed signal.

Regrettably, the fact that considerations based on mutual coherence and RIP are implicitly worst-case analyses makes the above bounds quite loose and conditions under which they hold often too restrictive. As an example, the matrix in (1.7) that underlies the example in Figs. 1.6, 1.7, and 1.14 has a mutual coherence equal to 1 and is such that, if $\kappa = 1$ then $\delta_{2\kappa} = 1/2$. With this, Theorem 1.5 does not hold for $\kappa = 1$ and Theorems 1.6 and 1.7 cannot be applied since $\delta_{2\kappa} > \sqrt{2} - 1$. Despite this, perfect reconstruction is clearly possible at least in the noiseless case.

What surely remains is that BP and BPDN can be reasonably expected to help retrieving the original signal and they are convex and thus more easily tractable than the original (1.26). Actually, BP is not only convex but even a linear optimization problem. In fact, if one introduces the additional variables $\alpha_j = |\boldsymbol{\xi}_j|$, (1.27) with $\epsilon = 0$ is equivalent to

$$\begin{aligned} \arg\min_{\alpha_j} \ & \textstyle\sum_{j=0}^{d-1} \alpha_j \\ \text{s.t.} \ & \boldsymbol{B}\boldsymbol{\xi} = \boldsymbol{y} \\ & \boldsymbol{\xi}_j \leq \alpha_j \quad j = 0, \ldots, d-1 \\ & -\boldsymbol{\xi}_j \leq \alpha_j \quad j = 0, \ldots, d-1 \\ & \alpha \geq 0 \quad j = 0, \ldots, d-1 \end{aligned} \tag{1.32}$$

where the last three constraint inequality guarantee that each α_j is not smaller than the absolute values of $\boldsymbol{\xi}_j$ and the fact that we minimize $\sum_{j=0}^{d-1} \alpha_j$ pushes each α_j to its lower bound.

Hence, though specialized solvers exist for CS-related optimization problems, BP can be also addressed by means of standard, large-scale linear optimizers.

References

1. Z.D. Bai, Y.Q. Yin, Limit of the smallest eigenvalue of a large dimensional sample covariance matrix. Ann. Probab. **21**(3), 1275–1294 (1993)
2. R. Baraniuk et al., A simple proof of the restricted isometry property for random matrices. Constr. Approx. **28**(3), 253–263 (2008)
3. T. Cai, T. Jiang, Limiting laws of coherence of random matrices with applications to testing covariance structure and construction of compressed sensing matrices. Ann. Stat. **39**(3), 1496–1525 (2011)
4. E.J. Candès, The restricted isometry property and its implications for compressed sensing. Comptes Rendus Mathematique **346**(9), 589–592 (2008)
5. D.L. Donoho, M. Elad, Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. Proc. Natl. Acad. Sci. **100**(5), 2197–2202 (2003)
6. D.L. Donoho, X. Huo, Uncertainty principles and ideal atomic decomposition. IEEE Trans. Inf. Theory **47**(7), 2845–2862 (2001)
7. Y.C. Eldar, G. Kutyniok, *Compressed Sensing: Theory and Applications* (Cambridge University Press, Cambridge, 2012)

8. A.L. Goldberger et al., Physiobank, Physiotoolkit, and Physionet: components of a new research resource for complex physiologic signals. Circulation **101**(23), 215–220 (2000)
9. R. Gribonval, M. Nielsen, Sparse representations in unions of bases. IEEE Trans. Inf. Theory **49**(12), 3320–3325 (2003)
10. A.K. Gupta, D.K. Nagar, *Matrix Variate Distributions* (Chapman & Hall/CRC Press, Boca Raton, 2000)
11. J. Harrington et al., *The EMU Speech Database System*. Available at http://emu.sourceforge.net/
12. T. Jiang, The asymptotic distributions of the largest entries of sample correlation matrices. Ann. Appl. Probab. **14**(2), 865–880 (2004)
13. J. Kovačević, A. Chebira, Life beyond bases: the advent of frames Part I. IEEE Signal Process. Mag. **24**(4), 86–104 (2007)
14. J. Kovačević, A. Chebira, Life beyond bases: the advent of frames - Part II. IEEE Signal Process. Mag. **24**(6), 115–125 (2007)
15. M. Mangia, R. Rovatti, G. Setti, Rakeness in the design of analog-to-information conversion of sparse and localized signals. IEEE Trans. Circuits Syst. I Regul. Pap. **59**(5), 1001–1014 (2012)
16. V.A. Marčhenko, L.A. Pastur, Distribution of eigenvalues for some sets of random matrices. Sbornik Mathematics **1**(4), 457–483 (1967)
17. B.K. Natarajan, Sparse approximate solutions to linear systems. SIAM J. Comput. **24**(2), 227–234 (1995)
18. L. Welch, Lower bounds on the maximum cross correlation of signals (Corresp.). IEEE Trans. Inf. Theory **20**(3), 397–399 (1974)

Chapter 2
How (Well) Compressed Sensing Works in Practice

2.1 Non-Worst-Case Assessment of CS Performance

One of the main problems with coherence and restricted isometries is that the corresponding parameters are explicitly calibrated on worst-case scenarios. This corresponds to the desire of providing guarantees, thanks to which the system is known to operate correctly. Yet, acquisition systems work on random inputs and it is perfectly sensible to characterize their performance by probabilistic means. This is particularly true when the input is not their unique random components since, for example, the matrix $\boldsymbol{A}$ is also a possibly time varying, uncertain ingredient of the processing.

An instructive example of this alternative route is given by a more geometric approach to the properties of the minimization problem (1.27) (Basis Pursuit—BP) and to its relationship with the minimization problem (1.26). Everything hinges on a special kind of polytopes.

Definition 2.1 These are the definitions we need to proceed in our analysis

- A p-dimensional convex polytope $P \subset \mathbb{R}^p$ is the convex hull of a set V of points in $\mathbb{R}^p$.
- If no point in V can be dropped without changing the resulting convex hull, then the points in V are the *vertices* of P.
- The intersection $P \cap h$ of a p-dimensional convex polytope P with a $p-1$-dimensional hyperplane that does not contain any point of the interior of P is called a *facet* of P. Facets can be 0-dimensional (vertices), 1-dimensional (edges), or in general q-dimensional with $q < p$.
- Given a convex polytope $P \subset \mathbb{R}^p$ with vertices $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots$ and a $q \times p$ matrix $\boldsymbol{B}$, the convex hull of the points $\boldsymbol{B}\boldsymbol{v}_0, \boldsymbol{B}\boldsymbol{v}_1, \ldots$ is a q-dimensional convex polytope $Q = \boldsymbol{B}P \subset \mathbb{R}^q$.

M. Mangia et al., *Adapted Compressed Sensing for Effective Hardware Implementations*, DOI 10.1007/978-3-319-61373-4_2

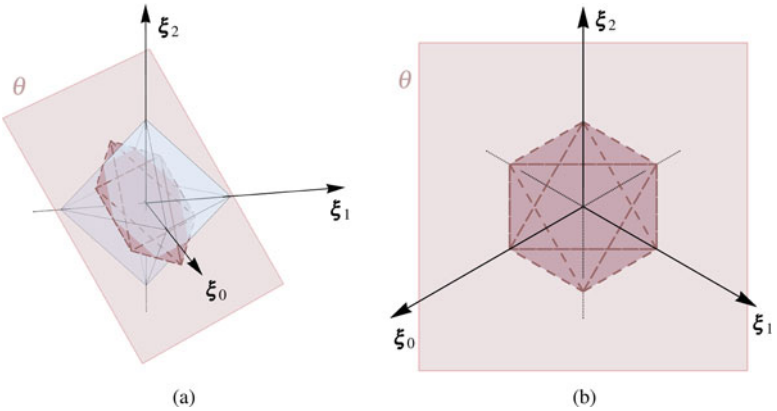


Fig. 2.1 Construction of $\boldsymbol{B}S_3^1(1)$ starting from $S_3^1(1)$ using $\boldsymbol{B}$ as in (1.7) (**a**) and its frontal view allowing face counting (**b**)

- A polytope is said to be *centrosymmetric* if $\boldsymbol{v} \in V$ implies $-\boldsymbol{v} \in V$.
- If $\boldsymbol{e}_j = (0, \ldots, 0, 1, 0, \ldots, 0)^\top$ where the unique 1 appears in the j-th position, then the *crosspolytope* in $\mathbb{R}^p$ is defined as the centrosymmetric convex hull of $V = \{\pm\boldsymbol{e}_0, \pm\boldsymbol{e}_1, \ldots, \pm\boldsymbol{e}_{p-1}\}$. In our previous notation the crosspolytope is nothing but $S_p^1(1)$.

Starting from these definitions that are standard elements in the theory of convex polytopes, one may develop specific concepts related to possibility of reconstructing the original signal from the measurements vector [7, 8, Theorem 7.5].

Definition 2.2 A p-dimensional centrosymmetric polytope is said to be *centrally q-neighborly* if every subset of V with q elements that does not include two antipodal vertices is the set of vertices of a $q-1$-dimensional facet of P.

Theorem 2.1 *If $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$ has a unique solution with not more than κ non-null components, then such a solution is the unique solution of BP if and only if $\boldsymbol{B}S_d^1(1)$ has $2d$ vertices and is centrally $(\kappa-1)$-neighborly.*

The point of interest in Theorem 2.1 is that it gives a necessary and sufficient condition for signal reconstruction: no worst-case bounding is involved. This has a substantial impact on the predictability of CS performance. As an example, we may go back to the matrix $\boldsymbol{B}$ in (1.7) and recall that its mutual coherence equal to 1 and its RIC equal to $1/2$ prevent the application of the results leveraging those concepts, i.e., of Theorems 1.5, 1.6, and 1.7.

Yet, Theorem 2.1 explains why reconstruction by means of BP is always effective in the noiseless case. Figure 2.1a reports the construction of $\boldsymbol{B}S_3^1(1)$ starting from $S_3^1(1)$. The same $\boldsymbol{B}S_3^1(1)$ is visualized in Fig. 2.1b. In this case $d = 3$ and $\kappa = 1$ and it is easy to verify that $\boldsymbol{B}S_3^1(1)$ has $2d = 6$ vertices each of them trivially being a face so that the polytope is centrally 0-neighborly.

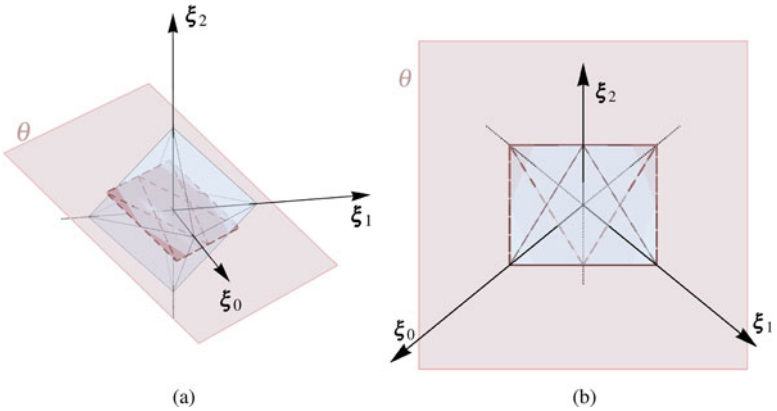


Fig. 2.2 Construction of $\boldsymbol{B}S_3^1(1)$ starting from $S_3^1(1)$ using $\boldsymbol{B}$ as in (2.1) (**a**) and its frontal view allowing face counting (**b**)

The same theorem indicates when the choice of $\boldsymbol{B}$ may prevent us from getting a reconstruction. As an example consider

$$\boldsymbol{B} = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{pmatrix} \tag{2.1}$$

and the resulting $\boldsymbol{B}S_3^1(1)$ as in Fig. 2.2. In this case, the number of vertices is only $4 < 2d = 6$ and Theorem 2.1 implies that reconstruction by means of (1.27) may be impossible since the sparsity prior may not be enough to select a unique solution of $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$. What happens is described in Fig. 2.2 in which $S_3^1(1)$ is projected with the new $\boldsymbol{B}$ on a new plane θ. Two out of 8 faces of $S_3^1(1)$ are orthogonal to θ so that one of the vertices of each of these two faces gets mapped into a point on the projection of the edge connecting the other two vertices, disappearing in the final polytope $\boldsymbol{B}S_3^1(1)$.

This is what prevents reconstruction. In fact, assume that you want to recover the same point $\boldsymbol{\xi}'''$ as in Fig. 1.14a. The straight line corresponding to $\boldsymbol{y}''' = \boldsymbol{B}\boldsymbol{\xi}$ that is orthogonal to θ is also parallel to 2 faces of $S_3^1(1)$ so that its intersection with $S_3^1(\|\boldsymbol{\xi}'''\|_1)$ is a whole segment, each point of which is a solution of BP. This is visualized in Fig. 2.3.

As a further application to our toy case, the very same Theorem 2.1 explains why, no matter how the plane θ is positioned, the solution of BP is not able to retrieve $\boldsymbol{\xi}$ from $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$ when it is known that $\boldsymbol{\xi}$ is 2-sparse instead of 1-sparse. In fact, to have $2d$ vertices, $\boldsymbol{B}S_3^1(1)$ must be a hexagon. Yet, only the pairs of consecutive vertices belong to a $(\kappa - 1)$-dimensional facet (that for $\kappa = 2$ is an edge) of a hexagon, while other pairs do not, preventing central neighborliness. Hence, no matter how θ is positioned, BP cannot be used for signal reconstruction.

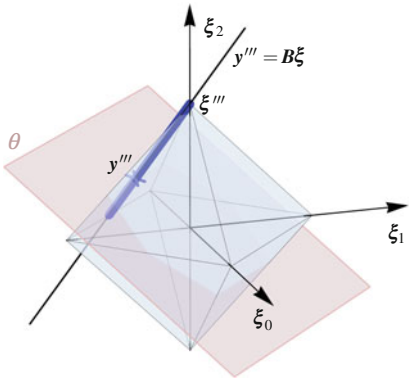


Fig. 2.3 Trying the reconstruction of $\boldsymbol{\xi}'''$ starting from $\boldsymbol{y}''' = \boldsymbol{B}\boldsymbol{\xi}'''$ by means of BP. All the points in the *blue thick segment* are equally good solutions

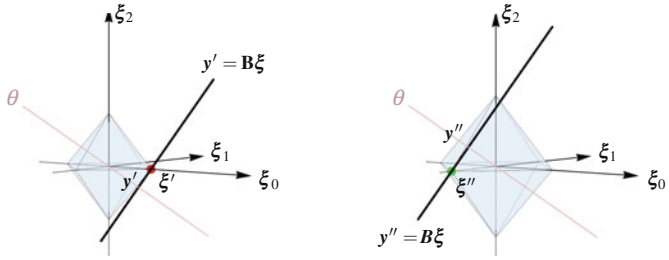


Fig. 2.4 Successful reconstruction of $\boldsymbol{\xi}'$ starting from $\boldsymbol{y}' = \boldsymbol{B}\boldsymbol{\xi}'$ (**a**) and of $\boldsymbol{\xi}''$ from $\boldsymbol{y}'' = \boldsymbol{B}\boldsymbol{\xi}''$ (**b**) by means of BP

Lastly, this powerful point of view can be extended to the case in which the signal to retrieve is random. In fact, Theorem 2.1 is a guarantee independent of $\boldsymbol{\xi}$, that ceases to hold if even a single $\boldsymbol{\xi}$ cannot be reconstructed. Yet, even when the guarantee does not hold, like for the matrix in (2.1), there are signals that can be reconstructed.

In particular, in passing from $S_3^1(1)$ to $\boldsymbol{B}S_3^1(1)$, 2 out of 6 vertices are lost and the signals $\boldsymbol{\xi}$ that cannot be retrieved are exactly those at the corresponding vertices of $S_3^1(\|\boldsymbol{\xi}\|_1)$, like $\boldsymbol{\xi}'''$ in Fig. 2.3. Yet, Fig. 2.2 shows that two other pairs of vertices appear in $\boldsymbol{B}S_3^1(1)$ and signals on the corresponding vertices of $S_3^1(\|\boldsymbol{\xi}\|_1)$ can still be reconstructed. This is shown in Figs. 2.4a and b where the same $\boldsymbol{\xi}'$ and $\boldsymbol{\xi}''$ as in Fig. 1.6 are uniquely identified by the intersection of the straight line $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$ and the minimum radius $\|\cdot\|_1$ ball.

Intuitively speaking, if the original 1-sparse signal $\boldsymbol{\xi}$ has the same probability of aligning with each of the axes, the probability that BP is effective in recovering it is equal to the ratio of the number of surviving vertices over the number of original vertices, i.e., $4/6 = 2/3$.

All this can be generalized to cope with a larger sparsity κ. To understand how, we may first define $\phi_k(\cdot)$ as the operator that counts the number of κ-dimensional facets of its polytope argument and state the following [7, Theorem 3].

Theorem 2.2 *Let $\boldsymbol{B}$ be an $m \times d$ matrix such that if $\boldsymbol{B}\boldsymbol{\alpha} = 0$ for a vector $\boldsymbol{\alpha}$ with less than m nonzeros then $\boldsymbol{\alpha} = 0$. Let also $\kappa < m/2$.*

Given a subset $K \subset \{0, \ldots, d-1\}$ of cardinality κ, we may have that if $\text{supp}(\boldsymbol{\xi}) = K$ then $\boldsymbol{\xi}$ can be reconstructed from $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$ by means of BP. Indicate with K_{BP} the number of such subsets, and with $K_{\text{tot}} = \binom{d}{\kappa}$ the total number of possible subsets of cardinality κ. Then

$$\frac{K_{\text{BP}}}{K_{\text{tot}}} \geq \frac{\phi_{\kappa-1}(\boldsymbol{B}S_d^1(1))}{\phi_{\kappa-1}(S_d^1(1))} \tag{2.2}$$

Assuming that the original signal has the same probability of featuring any of the K_{tot} supports of cardinality κ, the above result can be immediately recast into probabilistic terms to say that $p_{\text{BP}} = K_{\text{BP}}/K_{\text{tot}}$ is the probability of successful reconstruction by means of BP and is not less than the ratio of facets counts in (2.2). A dual result is available for the case in which $\boldsymbol{B}$ is random [8, Theorem 7.7].

Theorem 2.3 *Let $\boldsymbol{B}$ be a random $m \times d$ matrix whose probability distribution is invariant for any signed permutation of rows. Let $\boldsymbol{\xi} \in \mathbb{R}^d$ be a κ-sparse vector and $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$ the corresponding random measurement vector. The probability p_{BP} that BP retrieves $\boldsymbol{\xi}$ from $\boldsymbol{y}$ is bounded by*

$$p_{\text{BP}} \geq \frac{\mathbf{E}[\phi_{\kappa-1}(\boldsymbol{B}S_d^1(1))]}{\phi_{\kappa-1}(S_d^1(1))}$$

Note that, in general, facets counting is a combinatorial task so that the computation of $\phi_{\kappa-1}(\cdot)$ in high-dimensional settings can be expensive if not impossible. From this point of view, the introduction of random matrices $\boldsymbol{B}$ can be helpful if paired with the asymptotic conditions that are the mathematical equivalent of the high-dimensional setting in which CS is applied. Many sophisticated results are born in this area, whose simplest prototype is probably the one that we rephrase here in our terms [10].

Theorem 2.4 *Let $\boldsymbol{B} \sim \text{RGE}\,(\text{iid})$ with unit variance entries, $d = (\texttt{DR} \times \texttt{CR})m$, and $m = \texttt{OH}\,\kappa$. There is a function $\psi(\cdot)$ such that*

$$\lim_{d\to\infty} \frac{\phi_{\kappa-1}(\boldsymbol{B}S_d^1(1))}{\phi_{\kappa-1}(S_d^1(1))} = \begin{cases} 1 & \text{if } \texttt{DR} \times \texttt{CR} < \psi(\texttt{OH}) \\ 0 & \text{if } \texttt{DR} \times \texttt{CR} > \psi(\texttt{OH}) \end{cases}$$

Collecting the results in Theorems 2.2, 2.3, and 2.4 one gets that, as the dimensionality increases, there is a crisp *phase transition* in the possibility of

reconstructing $\boldsymbol{\xi}$ from its random projections. The excess of measurements with respect to the actual degrees of freedom in the signal (OH) controls the possibility of accommodating a certain dimensionality reduction DR $\times$ CR while maintaining the retrievability of the original signal.

Though Theorem 2.4 leverages RGE (iid), the existence and shape of the function ψ has been empirically found to be a general property [9] when the entries of $\boldsymbol{B}$ are iid or its rows are an iid random subset of certain orthonormal basis.

In the noiseless and exactly sparse case, this makes polytope-based analysis much closer to real performance than coherence or RIP-based considerations since neither the finite-dimension results, nor asymptotics of random $\boldsymbol{B}$ rely on worst-case bounding.

As a consequence, though neither the theory which heavily relies on symmetry considerations, nor the empirical evidence gathered so far in the Literature, say much on the possibility of straightforwardly applying this point of view to the design of a proper sensing matrix $\boldsymbol{A}$, we know that if n is large and we increase m enough, CS will eventually work very well (the *probability 1* implicit in Theorem 2.4).

From a more engineering point of view, this can be reversed to say that our aim is to find the minimum possible m for which CS works very well. Properly evolved and specialized, this is the key idea behind the discussion in the chapters to follow.

2.2 Beyond Basis Pursuit

Despite its theoretical appeal, BP is only an archetypal reconstruction method. In practical terms BP and its denoising variant BPDN have been implemented with a variety of methods, ranging from straightforward mapping to classical mathematical programming problems leveraging linear and quadratic optimization tools, to specialized procedures that look at them as a particular case of a convex optimization task.

The activity in this field revealed, for example, that it is sometimes convenient to address the BP or BPDN problems not in the *synthesis form* contained in (1.27) but in an alternative *analysis form*.

Note, in fact, that (1.27) depends only on $\boldsymbol{B}$ and thus considers and seeks to reconstruct the signal $\boldsymbol{\xi}$ in the sparsity domain. Once that $\boldsymbol{\xi}$ is known one may *synthesize* $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$.

Assume now that a linear operator $\boldsymbol{D}^*$ is available such that if $\boldsymbol{\xi} = \boldsymbol{D}^*\boldsymbol{x}$ then $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$. When $\boldsymbol{D}$ is a non-singular square matrix we simply have $\boldsymbol{D}^* = \boldsymbol{D}^{-1}$ and when $\boldsymbol{D}$ is a frame, $\boldsymbol{D}^*$ is the dual frame operator. With this, we may concentrate directly on the true signal $\boldsymbol{x}$ and try to solve the "equivalent"

$$\begin{aligned} &\arg\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{D}^*\boldsymbol{x}\|_1 \\ &\quad \text{s.t. } \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 \leq \epsilon \end{aligned} \tag{2.3}$$

Clearly, (1.27) and (2.3) are not equivalent when $\boldsymbol{D}$ is not an invertible matrix. In fact, $\boldsymbol{D}^*\boldsymbol{\xi}$ is only one of the many possible representations of $\boldsymbol{x}$ in the sparsity domain and is the only one considered while scanning the feasibility space of (2.3) while (1.27) considers all of them. What happens is that the choice made by $\boldsymbol{D}^*$ acts as a further prior and in this role, it is often useful to decreased dimensionality of the analysis form of BP and BPDN and help them finding good solutions.

Beyond this, accounting for all the methods and implementations described in the Literature and/or made available to practitioners is out of the scope of this book. Yet, it is useful to mention some of the most widespread tools distinguishing between those helping the implementation of BP, BPDN, and their other variants, those tackling the reconstruction problem from a theoretically different point of view, and those that are mainly based on heuristic considerations and yield lightweight iterative procedures that may be extremely useful when the resources dedicated to signal retrieval are limited.

Further to their implementation in commercial, large-scale solvers, BP and BPDN can be solved by quite a few implementations. Among them it is worthwhile mentioning those in Table 2.1 where we give the commonly used acronym, a pointer to some ready-to-use code and references to the relevant Literature.

Since BP and BPDN are convex optimization problems, they can be tackled by convex solvers with wider applicability. Those in Table 2.2 are particularly effective in modeling and solving the two standard reconstruction problems. Additionally, their greater generality can be used to add constraints that model priors further to sparsity that may available on the signal, thus increasing reconstruction performance.

Further to these methods, instead of depending on the $\|\cdot\|_1$ norm and its favorable geometry, signal reconstruction can be approached from completely different points of view, e.g., from the estimation, or machine learning, or regression point of view. Different approaches result in different algorithms some of which are listed in Table 2.3.

Table 2.1 Some dedicated BP/BPDN solvers

Solver	Url	Reference
SPGL1	www.math.ucdavis.edu/~mpf/spgl1/	[2]
NESTA	statweb.stanford.edu/~candes/nesta/	[1]

Table 2.2 Some solvers of convex optimization problems that can be used for signal retrieval

Solver	Url	Reference
CVX	cvxr.com/	[12, 13]
Unlocbox	lts2.epfl.ch/unlocbox/	[5]

Table 2.3 Some signal reconstruction methods based on various heuristic

Solver	Url	Reference
GAMP	gampmatlab.wikia.com	[17]
IRLS	http://stemblab.github.io/irls/	[6]
SBL	dsp.ucsd.edu/~zhilin/BSBL.html	[14]

Table 2.4 Some signal reconstruction methods based on various heuristic

Solver	Url	Reference
FOCUSS	dsp.ucsd.edu/~jfmurray/software.htm	[11]
OMP	http://www.mathworks.com/matlabcentral/fileexchange/32402-cosamp-and-omp-for-sparse-recovery	[16]
CoSaMP		[16]
Iterative hard thresholding	www.personal.soton.ac.uk/tb1m08/sparsify/sparsify.html	[3]
	http://sparselab.stanford.edu/	

Table 2.5 Code sketch for CoSaMP

Require: $\boldsymbol{y}$ vector of measurements
Require: κ sparsity level
Require: $\boldsymbol{B} = \boldsymbol{AD}$ sensing matrix
  $\hat{\boldsymbol{\xi}} \leftarrow 0$ ▷ signal guess
  $\Delta\boldsymbol{y} \leftarrow \boldsymbol{y} - \boldsymbol{B}\hat{\boldsymbol{\xi}} = \boldsymbol{y}$ ▷ error in reproducing measurements
  repeat
    $\Delta\boldsymbol{\xi} \leftarrow \boldsymbol{B}^\top \Delta\boldsymbol{y}$ ▷ error in signal guess
    $J = \text{supp}(\hat{\boldsymbol{\xi}}) \cup \text{supp}(\Delta\boldsymbol{\xi}^{2\kappa\uparrow})$ ▷ support to correct error in signal guess
    $\hat{\boldsymbol{\xi}} \leftarrow 0$
    $\hat{\boldsymbol{\xi}}_J = \left(\boldsymbol{B}_{\cdot,J}\right)^\dagger \boldsymbol{y}$
    $\hat{\boldsymbol{\xi}} \leftarrow \hat{\boldsymbol{\xi}}^{\kappa\uparrow}$ ▷ new signal guess
    $\Delta\boldsymbol{y} = \boldsymbol{y} - \boldsymbol{B}\hat{\boldsymbol{\xi}}$ ▷ new error in reproducing measurements
  until convergence

Finally, procedures exist that retrieve the original signal by considering that the main issue in the computation of $\boldsymbol{\xi}$ is not finding a generic solution to $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$ but to find the sparse one. Starting from this, it is possible to generate solutions iteratively adjusting their sparsity at each step. Different heuristics may be used to promote sparsity and give raise to different methods, some of which are listed in Table 2.4. The simple structure of these methods and their relatively good performance make them ideal for CS embodiments in which the resources devoted to signal reconstruction are limited.

As an example of how simple such algorithms can be, assume that $\boldsymbol{B}$ is well approximated by a random matrix with i.i.d., zero-average, entries and that the $\|\cdot\|_2$ norm of each column is approximately equal. Since the columns of $\boldsymbol{B}$ are independent, the matrix $\boldsymbol{B}^\top\boldsymbol{B}$ is well approximated by a diagonal matrix.

Assume now that an estimate $\hat{\boldsymbol{\xi}}$ is given of the true $\boldsymbol{\xi}$. The measurement vector corresponding to $\hat{\boldsymbol{\xi}}$ is $\hat{\boldsymbol{y}} = \boldsymbol{B}\hat{\boldsymbol{\xi}}$ whose difference with respect to the true measurement vector is $\Delta\boldsymbol{y} = \boldsymbol{B}(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi})$. Thanks to the previous considerations on $\boldsymbol{B}^\top\boldsymbol{B}$, we also have that $\boldsymbol{B}^\top\Delta\boldsymbol{y} = \boldsymbol{B}^\top\boldsymbol{B}(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}) \approx \|\boldsymbol{B}_{\cdot,0}\|_2\,(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi})$.

Hence, the largest nonzero components of $\boldsymbol{B}^\top\Delta\boldsymbol{y}$ indicate the components of the signal that have been mistaken most by taking $\hat{\boldsymbol{\xi}}$ instead of $\boldsymbol{\xi}$. This is the core step in the CoSaMP algorithm whose complete definition is given in Table 2.5 where: $\cdot^{p\uparrow}$ that takes a vector and gives its thresholded version in which all but the p

largest component are set to zero, $\cdot^{\dagger}$ indicates the Moore–Penrose pseudo-inverse of a matrix, and given an index set J, a vector $\boldsymbol{v}$, and a matrix $\boldsymbol{M}$, $\boldsymbol{v}_J$ is the subvector of $\boldsymbol{v}$ containing only the entries of $\boldsymbol{v}$ with indexes in J, while $\boldsymbol{M}_{\cdot J}$ is the submatrix of $\boldsymbol{M}$ made of the columns of $\boldsymbol{M}$ whose indexes stay in J.

Though the convergence criterion is not specified, it is clear that the procedure itself is much simpler than solving a convex optimization problem. This is the reason why methods like this and like the others in Table 2.4 are often used in limited-resources realizations of reconstruction stages (see, e.g., [4]).

2.3 A Framework for Performance Evaluation

In the light of the discussions in Chap. 1 and of the initial section of this chapter, it is easy to state that a precise assessment of the performance of a CS system is far from easy.

An obvious intuition is that performance must be related to the magnitude of the reconstruction error, i.e., to the difference between the true sparse representation ξ and the one estimated by the reconstruction algorithm $\hat{\boldsymbol{\xi}}$ or between the true signal $\boldsymbol{x}$ and $\hat{\boldsymbol{x}} = \boldsymbol{D}\hat{\boldsymbol{\xi}}$.

Yet, the classical theory of Chap. 1 follows a worst-case leitmotif and gives bounds on quantities like $\left\| \hat{\boldsymbol{\xi}} - \boldsymbol{\xi} \right\|_2$ that are either rarely applicable (for example, because they pose too strict requirements on measurement matrices $\boldsymbol{A}$) or quite loose and ultimately very far from actual behavior.

Even the non-worst-case approach described at the beginning of this chapter has problems since its face-counting argument, though allowing a much sharper distinction between what can be reconstruct and what cannot, scales poorly as dimension increase and cannot be applied in practice.

Last but not least, the construction of the matrices $\boldsymbol{A}$ is often done by random means. This, paired with the intrinsic random nature of the signal to acquire, implies that reconstruction error is a quite complicated random quantity.

The most straightforward way of addressing all these problems is to resort to extensive Montecarlo simulations. Such an approach is the most common both in the Literature and in practice and consists in generating a large number W of signal instances $\boldsymbol{x}^{(j)}$ and of measurement matrices $\boldsymbol{A}^{(j)}$ for $j = 0, \ldots, W-1$, use each of them to compute $\boldsymbol{y}^{(j)} = \boldsymbol{A}^{(j)}\boldsymbol{x}^{(j)}$ and then run one of the algorithms mentioned before to compute the estimation $\hat{\boldsymbol{x}}^{(j)}$ and consequently the reconstruction error in that case. The statistic of such an error is usually summarized in single numbers by means of one of the two approaches.

To begin with, it is most natural to define a Reconstruction Signal-to-Noise-Ratio

$$\texttt{RSNR[dB]} = 20 \log_{10} \left(\frac{\|\boldsymbol{x}\|_2}{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2} \right)$$

that acts as a merit figure, i.e., the larger the RSNR[dB], the better the reconstruction.

Then one may try to estimate the Average RSNR[dB] as

$$\texttt{ARSNR[dB]} = \mathbf{E}\left[20\log_{10}\left(\frac{\|\boldsymbol{x}\|_2}{\|\hat{\boldsymbol{x}}-\boldsymbol{x}\|_2}\right)\right] \approx \frac{1}{W}\sum_{j=0}^{W-1} 20\log_{10}\left(\frac{\left\|\boldsymbol{x}^{(j)}\right\|_2}{\left\|\hat{\boldsymbol{x}}^{(j)}-\boldsymbol{x}^{(j)}\right\|_2}\right) \tag{2.4}$$

Alternatively, one may assume that the reconstruction is correct when the corresponding RSNR[dB] exceeds a certain $\texttt{RSNR[dB]}_{\min}$ and define a Probability of Correct Reconstruction (PCR) as

$$\texttt{PCR} = \Pr\{\texttt{RSNR[dB]} \geq \texttt{RSNR[dB]}_{\min}\} \approx \frac{\left|\left\{\frac{\left\|\boldsymbol{x}^{(j)}\right\|_2}{\left\|\hat{\boldsymbol{x}}^{(j)}-\boldsymbol{x}^{(j)}\right\|_2} \geq 10^{\frac{\texttt{RSNR[dB]}_{\min}}{20}}\right\}\right|}{W}$$

Clearly, ARSNR[dB] and PCR are general-purpose merit figures and real-world applications may provide more significant indexes for establishing the acquisition performance. When applications are addressed at the end of this book, those merit figures will be possibly described and applied.

Yet, examples made to describe the adaptive method we address will use ARSNR[dB] and PCR and a uniform framework for the accumulation of Montecarlo trials.

In particular, we are interested in n-dimensional signals $\boldsymbol{x}$ that are both localized and κ-sparse with respect to a certain reference system $\boldsymbol{D}$ that we assume to be an orthonormal basis.

To generate samples of $\boldsymbol{x}$ we start from an instance of a zero-mean Gaussian random vector $\boldsymbol{x}'$ with covariance/correlation matrix $\mathscr{X}'$ and do the following steps:

$\boldsymbol{x}' \sim \mathrm{N}\left(0, \mathscr{X}'\right)$
$\boldsymbol{\xi}' \leftarrow \boldsymbol{D}^{-1}\boldsymbol{x}' = \boldsymbol{D}^{\top}\boldsymbol{x}'$
$\boldsymbol{\xi} \leftarrow (\boldsymbol{\xi}')^{\kappa\uparrow}$
$\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$

that formalize the intuitive idea of taking a possibly non-white vector ($\boldsymbol{x}'$) project it onto the basis along which we want our signal to be sparse, sparsify it and map it back into its original basis. Clearly, if $\kappa = n$ we have $\boldsymbol{x} = \boldsymbol{x}'$ since no clipping takes place.

As a first remark, note that from $\mathbf{E}[\boldsymbol{x}'] = 0$ we have $\mathbf{E}[\boldsymbol{\xi}'] = 0$, $\mathbf{E}[\boldsymbol{\xi}] = 0$ and $\mathbf{E}[\boldsymbol{x}] = 0$. Moreover, if we define the covariance/correlation matrices $\mathscr{X}' = \mathbf{E}[\boldsymbol{x}\boldsymbol{x}'^{\top}]$, $\boldsymbol{\Xi}' = \mathbf{E}[\boldsymbol{\xi}'\boldsymbol{\xi}'^{\top}]$, $\boldsymbol{\Xi} = \mathbf{E}[\boldsymbol{\xi}\boldsymbol{\xi}^{\top}]$, $\mathscr{X} = \mathbf{E}[\boldsymbol{x}\boldsymbol{x}^{\top}]$, we have $\boldsymbol{\Xi}' = \boldsymbol{D}^{\top}\mathscr{X}'\boldsymbol{D}$ and $\mathscr{X} = \boldsymbol{D}\boldsymbol{\Xi}\boldsymbol{D}^{\top}$.

Hence, if $\mathscr{X}'$ is a diagonal matrix both the components of $\boldsymbol{x}'$ and the components of $\boldsymbol{\xi}'$ are independent and the same happens for the nonzero components of $\boldsymbol{\xi} = (\boldsymbol{\xi}')^{\kappa\uparrow}$ causing $\boldsymbol{\Xi}$, and thus also $\mathscr{X}$ to be diagonal. In this case, $\boldsymbol{x}$ will only be κ-sparse but not localized. In fact, by recalling (1.5) we have

$$\mathscr{L}_{\boldsymbol{x}} = \frac{\mathrm{tr}(\mathscr{X}^2)}{\mathrm{tr}^2(\mathscr{X})} - \frac{1}{n} = 0$$

since for diagonal correlations we have $\mathrm{tr}(\mathscr{X}^2) = n\mathscr{X}_{0,0}^2$ while $\mathrm{tr}^2(\mathscr{X}) = n^2\mathscr{X}_{0,0}^2$.

Localization can be imposed by choosing a non-diagonal $\mathscr{X}'$ whose features will approximately be translated into those of $\mathscr{X}$. In fact, since the κ largest components of $\boldsymbol{\xi}'$ are carried over to $\boldsymbol{\xi}$, $\boldsymbol{\xi}$ is the best possible κ-sparse approximation of $\boldsymbol{\xi}'$ and the same relationship holds between $\boldsymbol{x}$ and $\boldsymbol{x}'$. Hence, the larger the κ, the more similar the behavior of $\boldsymbol{x}$ to that of $\boldsymbol{x}'$.

Though, the relationship between the localization of $\mathscr{X}'$ and that of $\mathscr{X}$ is difficult to model analytically we may provide some numerical evidence on the effectiveness of this method within the specific framework that we will use in our examples.

In particular we will consider $\mathscr{X}'$ such that $\mathscr{X}'_{j,k} = \omega^{|j-k|}$ for some $-1 < \omega < 1$. As noted in Chap. 1, this means that $\boldsymbol{x}'$ is a chunk of a stationary stochastic process with power spectrum

$$\Psi(f) = \frac{1-\omega^2}{1+\omega^2-2\omega\cos(2\pi f)}$$

that assumes a high-pass profile for $-1 < \omega < 0$, a flat/white profile for $\omega = 0$, and a low-pass profile for $0 < \omega < 1$. With some calculations one gets

$$\mathscr{L}_{\boldsymbol{x}'} = \frac{2}{n^2}\sum_{j=1}^{n-1} j\omega^{2(n-j)} = \frac{2\omega^2}{n}\frac{n\left(1-\omega^2\right)+\omega^{2n}-1}{n\left(1-\omega^2\right)^2} \tag{2.5}$$

Assume now $n = 128$, and $\boldsymbol{D}$ as the orthonormal Discrete Cosine Transform (DCT) basis. By generating a large amount of sample vectors $\boldsymbol{x}'$ and thus $\boldsymbol{x}$ we may estimate their localization and the power spectrum of the process from which they are taken. The result of such estimations is reported in Figs. 2.5 and 2.6 for different values of ω and sparsity κ (remember that $\kappa = n$ implies $\boldsymbol{x} = \boldsymbol{x}'$, i.e., $\boldsymbol{x}$ is a Gaussian random vector with an exponential correlation controlled by the decay ω).

In particular Fig. 2.5 shows how $\mathscr{L}_{\boldsymbol{x}}$ changes when ω changes. Note that increasing $|\omega|$ increases the localization of the generated signal $\boldsymbol{x}$. As far as *what kind* of localization is conferred to $\boldsymbol{x}$, Fig. 2.6 shows that when $\boldsymbol{x}'$ is low–low pass also $\boldsymbol{x}$ is low-pass, and vice versa.

Overall, our generation methods prove itself to be a practical way of ensuring sparsity while at least qualitatively controlling the localization of the signal and, as such, will be used in all the non-real-world examples of this volume.

To keep such examples not too far from realistic conditions, we refer to Table 1.1 and focus on processes with localizations compatible with those of real-world signals. This helps defining some prototype signals that are reported in Table 2.6.

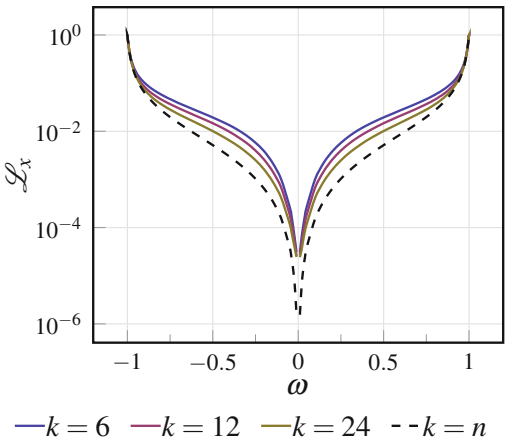


Fig. 2.5 The localization of $\boldsymbol{x}$ when ω changes and for different values of κ

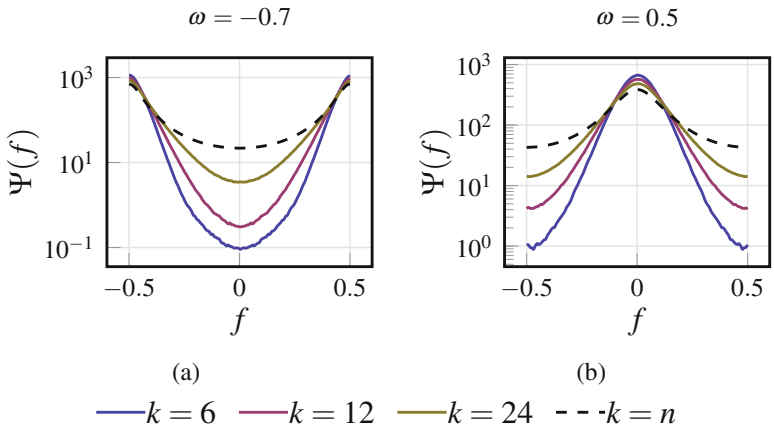


Fig. 2.6 The spectrum of $\boldsymbol{x}$ in a low-pass (**b**) and high-pass (**a**) case for different values of κ

Table 2.6 Definition of prototype signals used in the toy examples

Signal name	$\mathscr{L}_x$	κ	ω
ZL: $\mathscr{L}_x = 0$—white	0	6	0
		12	
		24	
LL: low $\mathscr{L}_x$	0.02	6	± 0.509
		12	± 0.584
		24	± 0.669
ML: medium $\mathscr{L}_x$	0.06	6	± 0.810
		12	± 0.853
		24	± 0.878
HL: high $\mathscr{L}_x$	0.2	6	± 0.959
		12	± 0.964
		24	± 0.966

Since most of our discussion hinges on the design of the matrix $\boldsymbol{A}$ producing the compressed measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, all the examples will address a specific design option or compare a number of them.

To do so we will rely on signals $\boldsymbol{x}$ generated as above and simulate the acquisition process by first perturbing them with a random vector $\boldsymbol{\eta}^{\boldsymbol{x}}$ made of independent, zero-mean Gaussian components whose variance is adjusted to match a prescribed Intrinsic Signal-to-Noise Ratio

$$\texttt{ISNR[dB]} = 20 \log_{10} \left(\frac{\|\boldsymbol{x}\|_2}{\|\boldsymbol{\eta}^{\boldsymbol{x}}\|_2} \right)$$

Such a perturbation is injected to simulate inaccuracies in the acquisition stages, including the possible quantization.

The perturbed signal is then used to produce measurements by using the matrix $\boldsymbol{A}$ under assessment and obtaining $\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x} + \boldsymbol{\eta}^{\boldsymbol{x}})$. When not explicitly declared otherwise, we will produce the reconstructed signal to be matched against the true signal by feeding $\boldsymbol{y}, \boldsymbol{A}$, and ISNR into the functions provided by the SPGL1 package mentioned in the previous section implementing either the BP or BPDN method, whose robustness and ease of use make it the ideal candidate for the concoction of examples.

2.4 Practical Performance

The first section of this chapter shows that, when not modeled from a worst-case point of view, CS is a promising technique that may allow to reconstruct an n-dimensional signal $\boldsymbol{x}$ from m scalar measurements in a vector $\boldsymbol{y}$ with $m \ll n$.

To give a quantitative appreciation of what can be achieved, assume that $\boldsymbol{x}$ is n-dimensional with $n = 128$, that is $\kappa = 6$ sparse with respect to a DCT orthonormal basis and that is generated as described before with $\omega = 0$ and $\texttt{ISNR[dB]} = 60\,\text{dB}$.

Take $\boldsymbol{A} \sim$ RGE (iid) and, for each value of m from 6 to 64 perform a Montecarlo simulation. For each trial compute the RSNR to accumulate a profile of ARSNR as a function of m. By fixing $\texttt{RSNR[dB]}_{\text{min}} = \texttt{ISNR[dB]} - 5\,\text{dB} = 55\,\text{dB}$, we may also estimate the PCR for each m. The result is reported in Fig. 2.7.

Since both ARSNR and PCR are the-larger-the-better merit figures, the sigmoidal trends in both plots are the practical implication of theorems like Theorem 2.4. In fact, coherently with what theory says, there is some critical value of m after which performance dramatically increases, giving rise to what is often called *phase transition*.

Beyond reflecting theoretical results, plots like those in Fig. 2.7 give a quantitative appreciation of achievable *compression*. For example, Fig. 2.7a shows that for $m = 64$ the ARSNR slightly exceeds the $\texttt{ISNR} = 60\,\text{dB}$ and thus indicates that the original signal, whose dimensionality is $n = 128$, can be acquired with a compression ratio $\texttt{CR} \simeq 2$ with no loss of accuracy (actually with a little amount

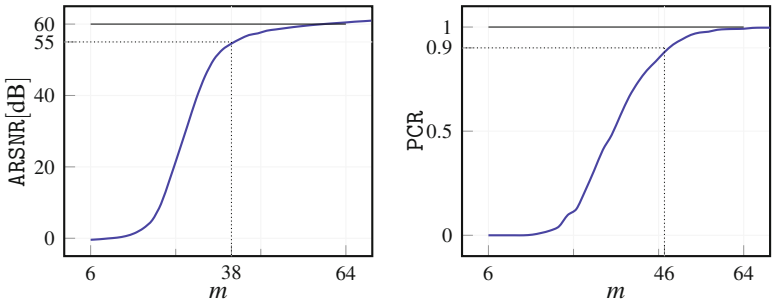


Fig. 2.7 Montecarlo assessment of performance for a classical CS system: when the number of measurements increases both the ARSNR (**a**) and PCR (**b**) increase

of denoising). Yet, one may decide that an $\texttt{ARSNR} = 55\,\text{dB}$ is enough for the application at hand and derive from the same plot that $m = 38$ measurements are enough to meet the specification, increasing the compression ratio to $\texttt{CR} \simeq 3.4$.

Clearly, this concerns average performance. A stricter point of view would be to require that $\texttt{RSNR} = 55\,\text{dB}$ is not achieved on average but at least 90% of the times. Since Fig. 2.7b estimates the probability that RSNR exceeds that threshold as a function of m, one gets that this more stringent specification can be met sizing the system with $m = 46$, that still gives $\texttt{CR} \simeq 2.8$.

This is a somehow impressive performance since it places well apart from the worst-case scenarios that were addressed in deriving the guarantees. As an example, Theorem 1.7, specialized to the case in which $\boldsymbol{x}$ is perfectly κ-sparse with respect to an orthonormal basis, says that the error between the original signal $\boldsymbol{x}$ and its reconstruction $\hat{\boldsymbol{x}}$ can be bounded as

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 = \left\|\boldsymbol{D}\hat{\boldsymbol{\xi}} - \boldsymbol{D}\boldsymbol{\xi}\right\|_2 = \left\|\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}\right\|_2 \leq 4\frac{\sqrt{1+\delta_{2k}}}{1-(\sqrt{2}+1)\delta_{2k}}\epsilon$$

where ϵ is such that $\|\eta\|_2 \leq \epsilon$, and δ_{2k} is the RIC of $\boldsymbol{A}$. For $\delta_{2k} \geq 0$, the coefficient of ϵ is monotonically increasing and thus, even in the best possible conditions, the guarantee of Theorem 1.7 on the RSNR is

$$\texttt{RSNR[dB]} = 20\log_{10}\left(\frac{\|\boldsymbol{x}\|_2}{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2}\right) \geq 20\log_{10}\left(\frac{\|\boldsymbol{x}\|_2}{4\,\|\boldsymbol{\eta}\|_2}\right) \geq \texttt{ISNR[dB]} - 12\,\text{dB}$$

that, due to its worst-case nature, gives little hint on the fact that, for example, a small average denoising effect can be obtained.

Figure 2.8 shows the trends of same merit figures when the signal to acquire is either $\kappa = 6$-sparse, or $\kappa = 12$-sparse, or $\kappa = 24$-sparse. Clearly, since κ is the minimum number of scalars that are needed to identify $\boldsymbol{x}$, a progressively larger

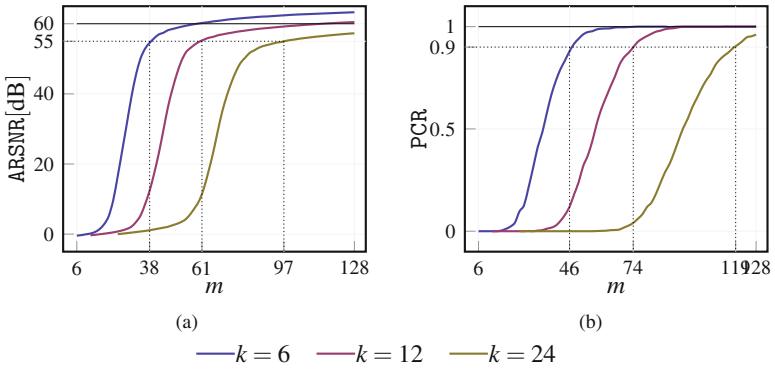


Fig. 2.8 Montecarlo assessment of performance for a classical CS system: when the number of measurements increases both the ARSNR (**a**) and PCR (**b**) increase, though with trends depending on the sparsity κ

Table 2.7 Numerical matching between the asymptotic trend in (1.29) and the empirical evidence of Fig. 2.8. The increase in the sparsity of the signal κ implies an increase in the minimum number m^* of measurements needed to achieve a certain performance that is compared with the $O(\kappa \log(n/\kappa))$ trend

	ARSNR $\geq$ 55 dB		PCR $\geq$ 0.9	
κ	m^*	$\frac{m^*}{\kappa \log_2(n/\kappa)}$	m^*	$\frac{m^*}{\kappa \log_2(n/\kappa)}$
6	38	1.43	46	1.74
12	61	1.49	74	1.81
24	97	1.67	119	2.05

number of measurement is needed to achieve a good signal reconstruction and the corresponding curves move to the right.

As an example, to obtain ARSNR $\geq$ 55 dB one needs at least $m^* = 38$ measurements when the signal is $\kappa = 6$-sparse, but $m^* = 61$ measurements if the signal to reconstruct is $\kappa = 12$-sparse, and $m^* = 97$ measurements for $\kappa = 24$-sparse signals. Though the trend of m^* against n and κ is identified only in asymptotic terms by (1.28) and (1.29), it may be used as a rough estimate of m^* even in finite cases.

In fact, by looking at Table 2.7 one is tempted to adopt as a first sizing criterion $m^* = c\kappa \log_2(n/\kappa)$ with a constant c in the range $2 \leq c \leq 3$.

Though all this may seem a success, from an engineering point of view it is only a starting point. In fact, performances like those in Fig. 2.7 are estimated for a system in which $\boldsymbol{A}$

- has entries that are infinite precision and unbounded;
- is maximally random within the variance constraint on its entries, and thus is completely agnostic both of its role and of its optimization possibilities.

Yet, any real-world implementation of the multiplication of $\boldsymbol{x}$ by $\boldsymbol{A}$ will imply a finite-range calculation with a limited precision, either because of noise if the implementation is analog, or because of quantization if the implementation is digital.

Moreover, instead of simply *accepting* measurements as they happen to be computed by a maximally random policy, one may try to look for measurements that best identify the signal itself so to squeeze as much information as possible in the $m < n$ scalars that will represent $\boldsymbol{x}$. The hope of this quest for *good* measurements is that a smaller number of substantial pieces of information can do the same job of a larger number of purely random looks at the signal.

Leaving the finite-range/finite-precision issue to a following chapter, note that this second aim seems to go against a quite commonly accepted idea, suggestively indicated as *democracy*, that each measurement carries roughly the same amount of information about the signal being acquired. The mathematical foundation of this idea is solid, it has to do with the RIC of the matrices $\boldsymbol{A}$ and with how such constant changes when few rows are dropped: it turns out that when the number of surviving rows is still larger than the minimum number needed to guarantee signal reconstruction, then which row was discarded has little effect on the RIC constant of the resulting matrix.

Yet, the practical effects of such a formal development are negligible for at least two reasons. The first is that we have seen how RIC-based performance bounds are to be taken only as guarantees since they are so loosely correlated with real-world performance that their use as a design criterion is ineffective. In this case, a small change in the RIC of $\boldsymbol{A}$ implies a small change in the performance guarantee but says nothing on the change of the actual performance.

The second is that when seeking for optimally designed CS stages, one never moves from the minimum m needed for a correct reconstruction far enough to be able to speculate about dropping some measurements and still working above that limit. The aim of system optimization is to push m as low as possible.

2.5 Countering the Myth of Democracy and Paving the Way for Practical Optimization

Beyond the considerations in the previous section, the fact that measurement *democracy* is a myth incorrectly inferred from a sound mathematical result can be demonstrated with an easy formal argument if we go back to the simplified setting that characterizes the first sections of this chapter: no noise (i.e., $\texttt{ISNR} = \infty$), and straightforward BP for reconstruction.

Within such a framework, we benefit from a powerful geometric insight on the reasons why the basic reconstruction strategy is successful and when it fails. In particular we may concentrate on failures and have a second look at a slightly rotated and simplified version of Fig. 2.3, that is Fig. 2.9.

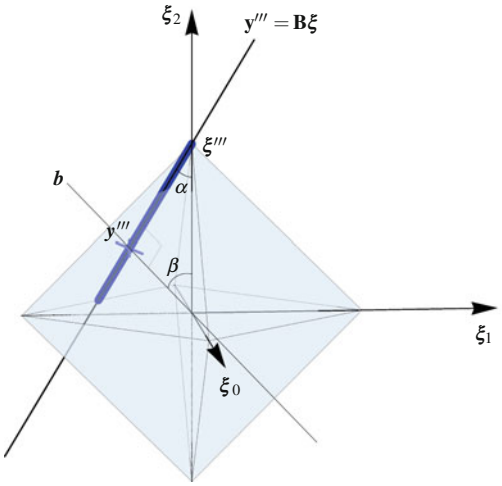


Fig. 2.9 A rotated and simplified version of Fig. 2.3 that highlights the relative positions of the signal $\boldsymbol{\xi}''$, the projection $\boldsymbol{y}'''$ and the face of $S_3^1(\|\boldsymbol{\xi}'''\|_1)$

In that figure, it is easy to verify that the solution to BP is not unique (all the points on the thick blue segment are possible reconstructions of the original signal) due to the fact that the projection plane contains a direction $\boldsymbol{b}$ that is orthogonal to one of the 2-dimensional facets of $S_3^1(\|\boldsymbol{\xi}'''\|_1)$ to which $\boldsymbol{\xi}'''$ itself belongs.

This may be reworded saying that one of the vectors onto which the signal is projected (one of the rows $\boldsymbol{b}$ of the matrix $\boldsymbol{B} = \boldsymbol{AD}$) forms with the signal an angle β such that its complement $\alpha = \pi/2 - \beta$ is equal to the angle between the signal ξ and a facet of $S_3^1(\|\boldsymbol{\xi}'''\|_1)$.

In this case $\alpha = \arccos\left(\sqrt{2/3}\right)$ and if we might ensure that the rows of $\boldsymbol{B}$ avoid forming with the signal an angle $\beta = \pi/2 - \arccos\left(\sqrt{2/3}\right)$, a case like the one depicted in Fig. 2.9 never occurs and signals like ξ''' are correctly reconstructed.

Clearly, other *bad* cases may happen. As an example, Fig. 2.10 shows that there is another choice of $\boldsymbol{b}$ that prevents BP from retrieving the original signal. Again, the reason is that the angle α between the signal and a facet of $S_3^1(\|\boldsymbol{\xi}'''\|_1)$ (in this case it is a 1-dimensional facet) is complementary to the one β between the signal and the direction $\boldsymbol{b}$ along which we are projecting. In this case $\alpha = \beta = \arccos\left(\sqrt{1/2}\right) = \pi/4$.

If we avoided choosing directions $\boldsymbol{b}$ whose angle the signal angles is one of the two computed above, both *bad* cases would be prevented.

Though the detailed proof is out of the scope of this volume, in the general n-dimensional case, when the signal is κ-sparse, angles between $\beta^{\min} = \pi/2 - \arccos(1/\sqrt{1+\kappa})$ and $\beta^{\max} = \pi/2 - \arccos\left(\sqrt{\frac{n-\kappa}{n-\kappa+1}}\right)$ must be avoided. In our case $n = 3$ and $\kappa = 1$ so that $\beta^{\min}$ and $\beta^{\max}$ boil down to $\pi/4$ and $\pi/2 - \arccos\left(\sqrt{2/3}\right)$ computed before. To be on the safe side and not to take too subtle decisions depending on n and κ, all angles in $[\pi/4, \pi/2]$ should be avoided.

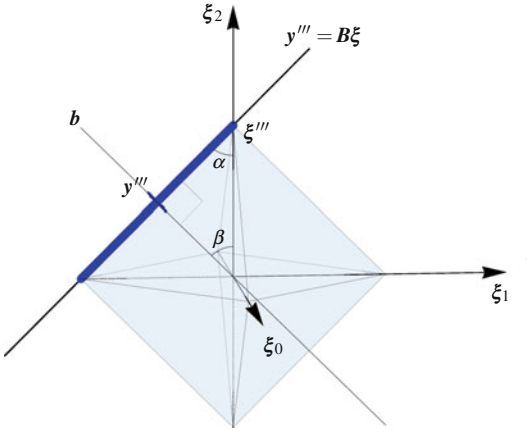


Fig. 2.10 Another *bad* choice of a direction $\boldsymbol{b}$ to use for projection. Also in this case BP cannot reconstruct the original signal as all the points in the *thick blue segment* are possible solutions

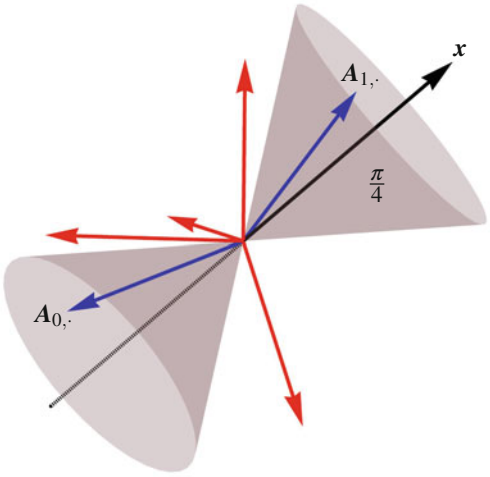


Fig. 2.11 Among many candidate vectors randomly pointing in space, only the two falling in the cone whose axis is $\boldsymbol{x}$ and whose aperture is $\pi/4$ become the rows $\boldsymbol{A}_{0,\cdot}$ and $\boldsymbol{A}_{1,\cdot}$ of the matrix $\boldsymbol{A}$ used for acquisitions

Hence, to ensure maximum reconstruction performance in a noiseless environment, it is advisable to select matrices $\boldsymbol{B}$ whose rows form with ξ an angle strictly smaller than $\pi/4$. When $\boldsymbol{D}$ is an orthonormal basis, this directly translates into a prescription for the angle between the rows of $\boldsymbol{A} = \boldsymbol{B}\boldsymbol{D}^\top$ and the signal $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$. Such a prescription translates into a simple geometric criterion: rows may be generated as n-dimensional vectors whose entries are independent random variables $\sim \mathrm{N}(0, 1)$ but are included in $\boldsymbol{A}$ only if their angle with $\boldsymbol{x}$ is less than $\pi/4$. Such a method will be indicated as *cone-constrained* CS as accepted rows are vectors falling in the cone whose axis is $\boldsymbol{x}$ and whose aperture is $\pi/4$ as exemplified in Fig. 2.11.

To have a practical appreciation of how much this criterion affects performance we may adopt the same simulation setting as above in the noiseless $\mathtt{ISNR} = \infty$ case and substituting BPDN with BP implemented as a purely linear optimization

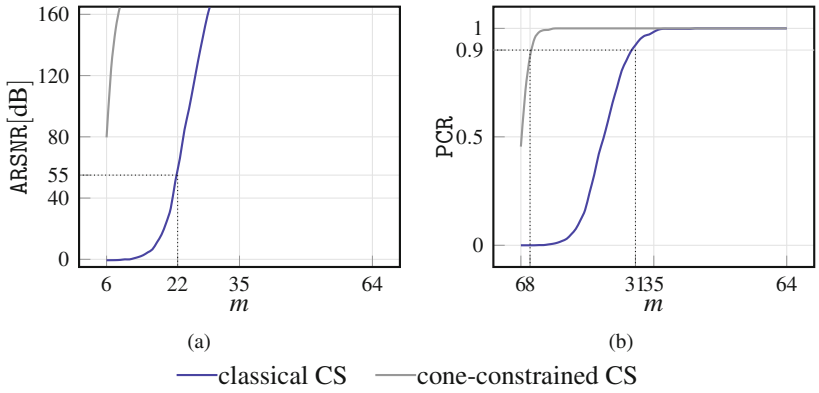


Fig. 2.12 Montecarlo assessment of performance for classical CS (*blue*) and cone-constrained CS (*gray*): the ideal cone-constrained CS clearly exhibits far better performance. (**a**) performance in terms of ARSNR while (**b**) is for PCR

problem (1.32). In these conditions we simulate the performance of classical CS, and cone-constrained CS. The results are shown in Fig. 2.12.

The absence of noise clearly improves reconstruction performance of classical CS. By comparing Fig. 2.7 with Fig. 2.12 we get that an average quality $\texttt{ARSNR} = 55\,\text{dB}$ can be reached with $m = 22$ measurements instead of $m = 38$ measurements, and an $\texttt{RSNR} = 55\,\text{dB}$ can be guaranteed 90% of the times with $m = 31$ measurements instead of $m = 48$ measurements.

Yet, cone-constrained CS has definitely better performance since the average reconstruction quality never falls below 80 dB and $\texttt{RSNR} = 55\,\text{dB}$ can be guaranteed 90% of the times with only $m = 8$ measurements.

Overall, the measurements we select are clearly carrying more information about the signal with respect to measurements picked randomly, and no democracy exists in the real-world. Although this may be taken as a discomforting truth from a social point of view, it is actually extremely good news from the point of view of engineering of CS. In fact, when not all the options are equally good, optimization may be called into play to look for the best design alternatives.

Regrettably, cone-constrained CS is only a theoretical tool since it has no concrete chance to be implemented. To understand why, we have to spend a few words on some very high-level implementation constraints that are at the base of any successful application of CS.

Though it is true that this book focuses on the design of CS stages according to the scheme of Fig. 1.2 we cannot avoid to place such an acquisition subsystem in a slightly more general perspective. This is what Fig. 2.13 does considering the same quantities as in Fig. 1.2. All the acquisition stages (sampling, quantization, compression) can be seen as a single block that *encodes* the analog waveform into the subsufficient-rate sequence of digital scalars $Q(y_k)$. Such sequence is passed to some other subsystem that is interested in knowing $x(t)$ and *decodes* the sequence $Q(y_k)$ into an approximation $\hat{x}(t)$.

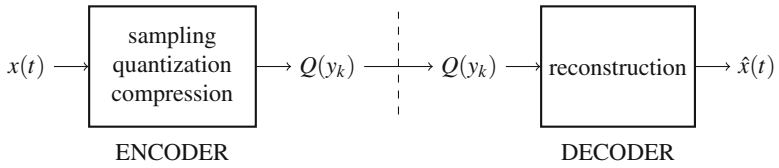


Fig. 2.13 A higher-level view of the role of signal acquisition

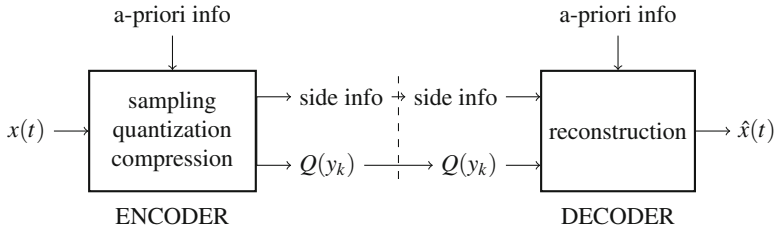


Fig. 2.14 The encoder-channel-decoder view with additional signal paths

The higher-level point of view reveals an encoder–decoder structure that highlights the fact that the only continuous communication between the encoder and the decoder is the subsufficient sequence, i.e., in principle, no other signal dependent information is passed from the acquisition subsystem to the subsystem that uses the acquired signal.

In terms of a CS acquisition mechanism, this means that, for example, the rows of $\boldsymbol{A}$ are not communicated to the decoder, that must be able to know them independently. This is why, a more realistic view at the scheme in Fig. 2.13 should comprise few other details.

First, the encoder and the decoder sides must share some a priori information. If, for example, $\boldsymbol{A}$ is fixed, then it must enter the design of both sides. Alternatively, if $\boldsymbol{A}$ is a time varying instance of a random matrix ensemble (as in our examples), the encoder and the decoder may share the design of a reproducible pseudorandom number generator and the initial state from which it works. In this case the operations of encoder and decoder must be synchronized thus implying a small amount of side information to be transferred from encoder to decoder further to the subsufficient sequence $Q(y_k)$. The resulting more realistic view of the acquisition system is given in Fig. 2.14. Clearly, for the compression scheme to be effective, the total transferred information (the subsufficient sequence plus the side information) must amount to less bits than what would be needed by the sheer transmission of a sufficient sequence of samples.

This is the main reason why cone-constrained CS cannot be effectively employed. In fact, what we may do to apply the method in practice is to deploy two identical copies of a pseudorandom number generator both at the encoder and

the decoder, synchronize them and let them run to produce candidate rows for the matrix $\boldsymbol{A}$. The encoder tests each of them and accepts only the first m of them whose angle with $\boldsymbol{x}$ is less than $\pi/4$ to build $\boldsymbol{A}$. Then, it computes $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ and communicates to the decoder both the vector $\boldsymbol{y}$ and the side information needed to identify the rows it used.

If we assume that to find m rows one must examine M candidates, the number of bits of side information is $\left\lceil \log_2 \binom{M}{m} \right\rceil$ since our task is to identify a specific subset of m elements among M possible candidates. Overall, the amount of information that must be transferred from the encoder to the decoder is $m\mathtt{b}_{\boldsymbol{y}} + \left\lceil \log_2 \binom{M}{m} \right\rceil$, where $\mathtt{b}_{\boldsymbol{y}}$ is the number of bits used for each sample of the subsufficient sequence $Q(y_k)$. This must be compared with the straightforward option of quantizing each samples with $\mathtt{b}_{\boldsymbol{x}}$ bits so that the bitwise compression ratio is

$$\mathrm{CR}^{\mathrm{bit}} = \frac{n\mathtt{b}_{\boldsymbol{x}}}{m\mathtt{b}_{\boldsymbol{y}} + \left\lceil \log_2 \binom{M}{m} \right\rceil} \tag{2.6}$$

Regrettably, the ratio between m and M suffers from a well-known effect of dimensionality on the shape of S_n^2 spheres. Assuming that the candidate rows span all the possible angles uniformly (this is what happens, for example, if their entries are independent normals), the probability that one of them falls within the proper cone is equal to the ratio between the measure of the surface of the two spherical caps $\Gamma' \cup \Gamma''$ illustrated in Fig. 2.15 and the measure of the surface ∂S_n^2 of the whole sphere S_n^2.

From [15] we get that such a ratio is

$$\frac{\mu(\Gamma' \cup \Gamma'')}{\mu(\partial S_n^2)} = B_{\sin^2(\pi/4)}\left(\frac{n-1}{2}, \frac{1}{2}\right) = B_{1/2}\left(\frac{n-1}{2}, \frac{1}{2}\right)$$

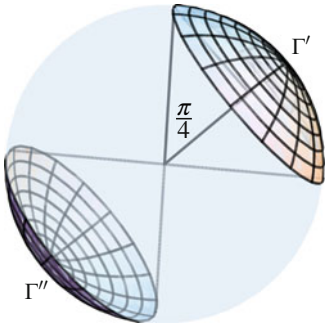


Fig. 2.15 The spherical caps whose surface is proportional to the probability of generating a random measurement falling into the $\pi/4$ cone

that uses the incomplete regularized beta function

$$B_\zeta(p,q) = \frac{\displaystyle\int_0^\zeta t^{p-1}(1-t)^{q-1}\mathrm{d}t}{\displaystyle\int_0^1 t^{p-1}(1-t)^{q-1}\mathrm{d}t}$$

from which we derive that $B_{1/2}\left(\frac{n-1}{2},\frac{1}{2}\right) \leq 2^{-\frac{n-1}{2}}$ for $n \geq 1$ is decreasing not less than exponentially with n. This means, for example, that the probability of a candidate 128-dimensional row of falling into the $\pi/4$ cone whose axis is any given signal $\boldsymbol{x}$ is less than 7.6×10^{-21}.

Assume now that we want to guarantee that $\texttt{RSNR} \geq 55\,\text{dB}$ at least 90% of the times. From Fig. 2.12 we get that $m = 8$ measurements are enough. Yet, the average number of independent candidate rows to evaluate before accumulating $m = 8$ measurements is $8/(7.6 \times 10^{-21}) = 1.1 \times 10^{21}$, and the side information that needs to be communicated amounts to 544 bit. If we assume $\mathtt{b}_y = 12$ (a sensible choice to achieve $\texttt{RSNR} = 55\,\text{dB}$), the total number of bits needed to encode the $n = 128$-dimensional window is $544 + 8 \times 12 = 640$ bit.

Without compression, we may roughly estimate the RSNR achieved by the straightforward quantization of each sample assuming that signal behaves almost sinusoidally so that $\texttt{RSNR} = 6.02\mathtt{b}_{\boldsymbol{x}} + 1.76\,\text{dB}$ where $\mathtt{b}_{\boldsymbol{x}}$ is the number of bits used for each sample. To have $\texttt{RSNR} = 55\,\text{dB}$ we may set $\mathtt{b}_{\boldsymbol{x}} = 9$ so that the total number of bits would be $128 \times 9 = 1152$ bit. Hence, the bitwise compression ratio (2.6) of cone-constrained CS is $\texttt{CR}^{\text{bit}} = 1152/640 \simeq 1.8$.

Note that, if we decided to use every row produced by the generator we would not need to send any side information beyond an initial synchronization, while Fig. 2.12 tells us that the same performance level as before would be guaranteed by $m = 21$ measurements, for a total of $m\mathtt{b}_y = 31 \times 12 = 372$ bit and a corresponding bitwise compression ratio $\texttt{CR}^{\text{bit}} = 1152/372 \simeq 3.1$.

All this said, there is no point in trying an implementation of cone-constrained CS since it does not give any real advantage with respect to purely random CS which also enjoys a much smaller computational burden (even if generating and testing a candidate row took a single nanosecond, accumulating 8 measurements in the $\pi/4$ cone would take more than 33000 years![1]).

However, what we are left with is an intuition that a possible criterion to increase the amount of information that a measurement y carries about the signal $\boldsymbol{x}$ is to obtain it as $y = \boldsymbol{a}^\top\boldsymbol{x}$ using a vector $\boldsymbol{a}$ lying on straight line whose angle with the straight line containing $\boldsymbol{x}$ is 'small', whatever this may signify. From here on what we do can only be intuitively justified but, as we will see in the more applicative

[1]To avoid this curse of dimensionality, the simulations leading to Fig. 2.12 had to generate rows of $\boldsymbol{A}$ by properly modulating the length of random rotations of $\boldsymbol{x}$ itself with angle smaller than $\pi/4$, the conceptual sieving procedure being totally unfeasible.

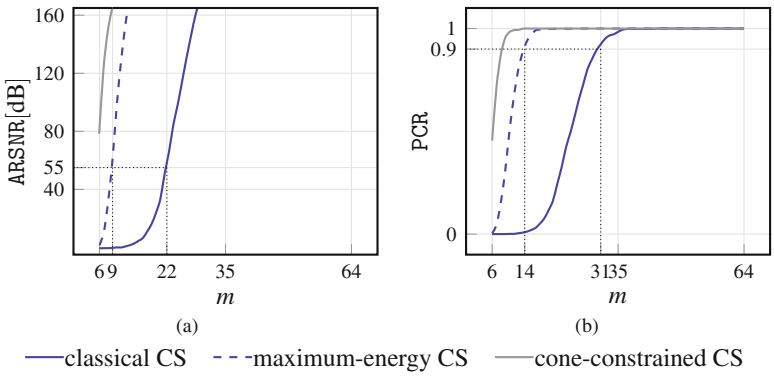


Fig. 2.16 Montecarlo assessment of performance for classical CS (*solid blue*), maximum-energy CS (*dashed blue*), and cone-constrained CS (*solid gray*): maximum-energy CS is not as performing as cone-constrained CS but it outperforms classical CS. (**a**) performance in terms of ARSNR while (**b**) is for PCR

chapters of this volume, gives raise to a powerful heuristic criterion supporting a well-defined and effective design flow for practical CS acquisition.

The first trivial remark is that, given two non-collinear vectors $\boldsymbol{v}'$ and $\boldsymbol{v}'$, forming an angle $\widehat{\boldsymbol{v}' \; \boldsymbol{v}''}$, the angle between the straight lines containing them is $\min\{\widehat{\boldsymbol{v}' \; \boldsymbol{v}''}, \pi - \widehat{\boldsymbol{v}' \; \boldsymbol{v}''}\}$. Hence, such an angle gets smaller when $\widehat{\boldsymbol{v}' \; \boldsymbol{v}''}$ either goes to 0 or π, i.e., when the absolute values $\cos^2\left(\widehat{\boldsymbol{v}' \; \boldsymbol{v}''}\right)$ increases. Since $y^2 = \left(\boldsymbol{a}^\top \boldsymbol{x}\right)^2 = \|\boldsymbol{a}\|_2^2 \, \|\boldsymbol{x}\|_2^2 \cos^2\left(\widehat{\boldsymbol{a} \; \boldsymbol{x}}\right)$ and $\boldsymbol{x}$ is assigned, if we may assume that all the rows of $\boldsymbol{A}$ have approximately the same length, smaller angles correspond to higher energies of the measurement y.

A new, heuristic method is naturally born from these considerations. In a system analogous to what has been sketched for cone-constrained CS, let the row generator produce M candidates. Then compose the matrix $\boldsymbol{A}$ with the rows $\boldsymbol{a}$ corresponding to the m largest values of $(\boldsymbol{a}^\top \boldsymbol{x})^2$. The measurements are computed as $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ and passed to the decoder. This strategy is named *maximum-energy* CS.

In this new configuration both M and m are degrees of freedom. This gives us some control on the amount of bits spent on side information $\left\lceil \log_2 \binom{M}{m} \right\rceil$, an amount that must be traded with the quality of the reconstruction. In this case we do not have a theoretical background allowing to anticipate reconstruction performance and we have to rely on simulations. If we do so, we may add a track to Fig. 2.12 and obtain Fig. 2.16.

Maximum-energy CS is simulated generating $M = 512$ candidates and taking the m largest energy measurements for $m = \kappa = 6$ to $m = n/2 = 64$. Since it is only a heuristic approximation of the cone-constrained policy, performance decreases but is still much higher than that of classical CS.

If we assume that our target reconstruction quality is $\texttt{ARSNR} \geq 55\,\text{dB}$, then classical CS achieves it with $m = 22$ while maximum-energy CS requires only $m = 9$. These numbers allow to compute the bitwise compression ratio (2.6) for the same case as above in which $\texttt{b}_x = 9$ and $\texttt{b}_y = 12$. Classical CS yields $\texttt{CR}^{\text{bit}} = 1152/(22 \times 12) = 1152/264 \simeq 4.4$. Maximum-energy CS yields $\texttt{CR}^{\text{bit}} = 1152/\left(9 \times 12 + \left\lceil \log_2 \binom{512}{9} \right\rceil\right) = 1152/171 \simeq 6.7$ and is therefore able to provide a gain further to the mere reduction of the number of scalar measurements.

If we assume that our target reconstruction quality is to guarantee that 90% of the times we have $\texttt{RSNR} \geq 55\,\text{dB}$, then classical CS achieves it with $m = 31$ while maximum-energy CS requires only $m = 14$. The corresponding bitwise compression ratios are $\texttt{CR}^{\text{bit}} \simeq 3.1$ for classical CS and $\texttt{CR}^{\text{bit}} \simeq 4.5$ for maximum-energy CS.

Overall, maximum-energy CS seems to be a good candidate to leverage the intuitive criterion we have developed for measurement quality while keeping adaptivity to a level that can be managed by adding a reasonable amount of side information.

This is true even in a noisy environment. In fact, we may go back to our original setting in which $\texttt{ISNR} = 60\,\text{dB}$, and keep the same configuration of maximum-energy CS to obtain curves like the ones in Fig. 2.17.

Since noise is back, performance deteriorates also for maximum-energy CS. Yet, the new method is still able to yield better bitwise compression ratios. In fact, looking at Fig. 2.17a we get that the reference quality level $\texttt{ARSNR} = 55\,\text{dB}$ is achieved with $m = 38$ measurements by classical CS, and with $m = 16$ measurements by maximum-energy CS. The usual computation of $\texttt{CR}^{\text{bit}}$ with $\texttt{b}_x = 9$ and $\texttt{b}_y = 12$ gives $\texttt{CR}^{\text{bit}} = 1152/456 \simeq 2.5$ for classical CS and $\texttt{CR}^{\text{bit}} = 1152/292 \simeq 3.9$ for maximum-energy CS.

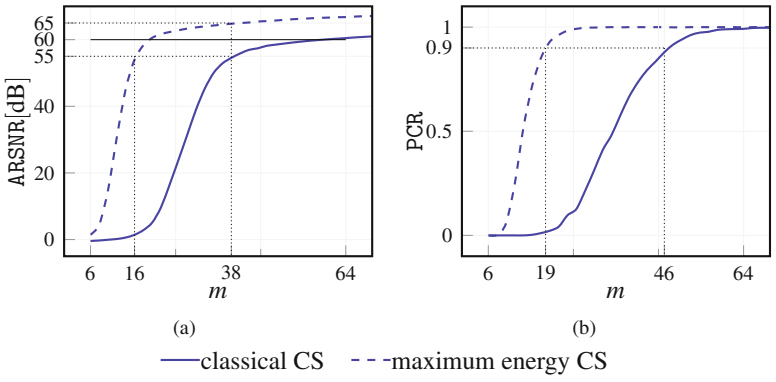


Fig. 2.17 Montecarlo comparison between performance of classical CS (*solid*) and maximum-energy CS (*dashed*): both the ARSNR (**a**) and PCR (**b**) curves are dramatically improved

Figure 2.17a also shows that the adapted method is able to provide noteworthy average denoising. For example, when classical CS achieves our reference performance level $\texttt{ARSNR} = 55\,\text{dB}$, maximum-energy CS is able to yield $\texttt{ARSNR} = 65\,\text{dB} > \texttt{ISNR}$. Clearly, this comes at some expense since maximum-energy CS accumulates and communicates $\left\lceil \log_2 \binom{512}{38} \right\rceil = 192\,\text{bit}$ in addition to the 456 bit used to encode the measurements. Yet, in this case reconstruction provides an accuracy that would not be attained by simply encoding the samples.

To confirm that what we have observed so far can be of use, we may explore different types of signals with different sparsities. As before, no formal analysis is available but an extensive Montecarlo assessment can be pursued using different values of κ and different values of ω in the exponential correlation signal mode we defined in Sect. 2.3.

A sample of the results of such an assessment is reported in Fig. 2.18 where we give the trends of both ARSNR and PCR (with target $\texttt{RSNR} = 55\,\text{dB}$) when $n = 128$ and $\kappa \in \{6, 12, 24\}$ while trying a high-pass signal with medium localization (ML HP), a white signal, and a low-pass signal with large localization (HL LP) as defined in Table 2.6. The white case with $\kappa = 6$ is the same as the one in Fig. 2.17.

In terms of the minimum number of measurements needed to match a certain reconstruction quality, the improvement of maximum-energy CS over classical CS is undoubtable. The fact that it may result in a better bitwise compression ratio must be checked case by case. From Fig. 2.18 it is clear that the performance of classical CS is independent of localization while that of maximum-energy CS is not. Hence, we may focus on the white case and summarize our quantifications in Table 2.8.

Overall, the maximum-energy criterion seems to behave rather well. Yet, its implementation has two drawbacks that may limit practical application. First, one has to compute many more measurements ($M = 512$ in our examples) than what is then communicated to the decoder. If the cost of computing a measurement is not negligible, this may have an impact on the encoder complexity. Due to the dimensionality effect that we already noted when analyzing cone-constrained CS, such an impact is expected to dramatically increase as n increases. Moreover, side information must be computed and communicated to the decoder further to measurements and this may also imply an increased encoder complexity.

Intuitively, this potentially increased complexity depends on the fact that maximum-energy CS automatically adapt itself to the specific instance $\boldsymbol{x}$ of the signal it is acquiring.

In the next chapter, we will see that a slightly less performing method can be devised which, leveraging the same measurement energy principle and adapting to the class of signals to acquire rather than to a specific instance, allows to increase performance while keeping encoder complexity to a minimum.

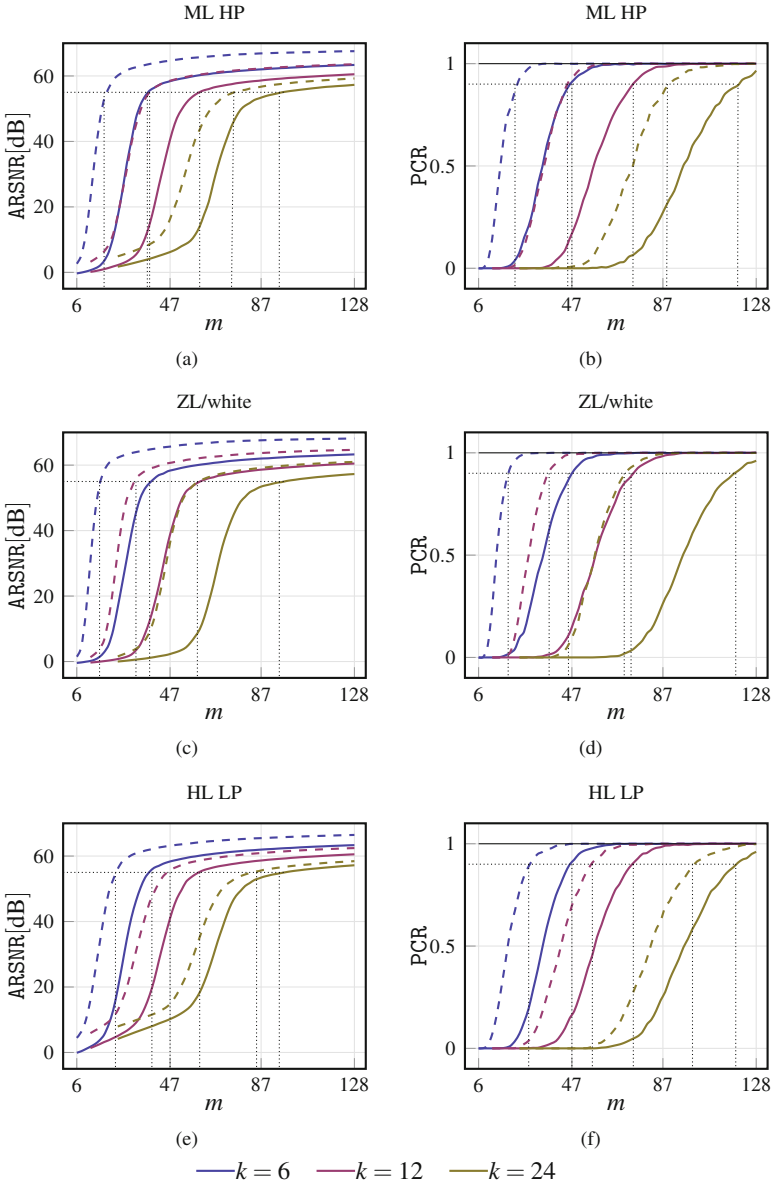


Fig. 2.18 Montecarlo comparison between performance of classical CS (*solid*) and maximum-energy CS (*dashed*): both the ARSNR (**a**) and PCR (**b**) curves are dramatically improved in all configurations

Table 2.8 Bitwise compression ratios of classical CS and maximum-energy CS when $\mathtt{b}_x = 8$ and $\mathtt{b}_y = 12$ and for two different reconstruction quality requirements. Straightforward encoding of $n = 128$ samples would require 1152 bit

ARSNR = 55 dB							
	Maximum-energy CS				Classical CS		
κ	m	M	$m\mathtt{b}_y + \lceil \log_2 \binom{M}{m} \rceil$	$\mathtt{CR}^{\mathrm{bit}}$	m	$m\mathtt{b}_y$	$\mathtt{CR}^{\mathrm{bit}}$
6	16	512	292	3.9	38	452	2.5
12	32	512	553	2.1	59	708	1.6
24	59	512	968	1.2	95	1140	1.0
PCR = 0.9							
	Maximum-energy CS				Classical CS		
κ	m	M	$m\mathtt{b}_y + \lceil \log_2 \binom{M}{m} \rceil$	$\mathtt{CR}^{\mathrm{bit}}$	m	$m\mathtt{b}_y$	$\mathtt{CR}^{\mathrm{bit}}$
6	19	512	342	3.9	47	564	2.5
12	37	512	632	1.8	73	876	1.3
24	70	512	1131	1.0	119	1428	0.81

References

1. S. Becker, J. Bobin, E.J. Candès, NESTA: a fast and accurate first-order method for sparse recovery. SIAM J. Imag. Sci. **4**(1), 1–39 (2011)
2. E. van den Berg, M.P Friedlander, Probing the Pareto frontier for basis pursuit solutions. SIAM J. Sci. Comput. **31**(2), 890–912 (2008)
3. T. Blumensath, M.E. Davies, Iterative thresholding for sparse approximations. J. Fourier Anal. Appl. **14**(5-6), 629–654 (2008)
4. D. Bortolotti et al., Energy-aware bio-signal compressed sensing reconstruction on the WBSN-gateway. IEEE Trans. Emerg. Top. Comput. **PP**(99), 1–1 (2016)
5. P.L. Combettes, J.-C. Pesquet, A proximal decomposition method for solving convex variational inverse problems. Inverse Prob. **24**(6), p. 065014 (2008)
6. I. Daubechies et al., Iteratively reweighted least squares minimization for sparse recovery. Commun. Pure Appl. Math. **63**(1), 1–38 (2010)
7. D.L. Donoho, Neighborly Polytopes and Sparse Solution of Underdetermined Linear Equations, Technical report, Department of Statistics, Stanford University, 2005
8. D.L. Donoho, J. Tanner, Counting faces of randomly projected polytopes when the projection radically lowers dimension. J. Am. Math. Soc. **22**(1), 1–53 (2009)
9. D.L. Donoho, J. Tanner, Observed universality of phase tran- sitions in high-dimensional geometry with implications for modern data analysis and signal processing. Philos. Trans. R. Soc. Lond. A Math. Phys. Eng. Sci. **367**(1906), 4273–4293 (2009)
10. D.L. Donoho, J. Tanner, Precise undersampling theorems. Proc. IEEE **98**(6), 913–924 (2010)
11. I.F. Gorodnitsky, B.D. Rao, Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. IEEE Trans. Signal Process. **45**(3), 600–616 (1997)
12. M. Grant, S. Boyd, CVX: Matlab Software for Disciplined Convex Programming version 2.1. http://cvxr.com/cvx, Mar 2015
13. M. Grant, S. Boyd, Graph implementations for nonsmooth convex programs, in *Recent Advances in Learning and Control*, ed. by V. Blondel, S. Boyd, H. Kimura. Lecture Notes in Control and Information Sciences (Springer, Heidelberg, 2008), pp. 95–110
14. S. Ji, Y. Xue, L. Carin, Bayesian compressive sensing. IEEE Trans. Signal Process. **56**(6), 2346–2356 (2008)

15. S. Li, Concise formulas for the area and volume of a hyperspherical cap. Asian J. Math. Stat. **4**(1), 66–70 (2011)
16. D. Needell, J.A. Tropp, CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Appl. Comput. Harmon. Anal. **26**(3), 301–321 (2009)
17. S. Rangan, Generalized approximate message passing for estimation with random linear mixing, in *2011 IEEE International Symposium on Information Theory Proceedings*, IEEE, July 2011, pp. 2168–2172

Chapter 3
From Universal to Adapted Acquisition: Rake That Signal!

3.1 Average Maximum Energy

Chapter 2 showed that, if one moves from a worst-case analysis of CS (the one classically used to provide mathematically sound guarantees) and get interested in what really makes an encoder–decoder pair successful, a criterion to improve performance is to choose measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ whose energy is large.

The straightforward application of this criterion to each distinct instance of the signal $\boldsymbol{x}$ requires some overhead on what is computed at the encoder (the candidate measurements that are not energetic enough to be chosen as the most representative) and on what is communicated from the encoder to the decoder (the bits needed to define which measurements are chosen among those that are computed). These overheads are due to the *adaptivity* of the max-energy approach, i.e., its ability to change the acquisition matrix $\boldsymbol{A}$ in response to the particular signal instance $\boldsymbol{x}$, something that must be done at run-time.

Either of these two overheads may be unacceptable in some implementations and in this chapter we develop what will be our core technique to leverage the maximum-energy criterion while not imposing any additional burden neither to the encoder nor to the communication.

The idea is to change from an *adaptive* approach to an *adapted* approach, i.e., to a mechanism that is tuned at design-time on the specific class of signals to acquire. To do so, we assume that rows $\boldsymbol{a}$ of $\boldsymbol{A}$ are generated independently and design their generator so that the energy $\left(\boldsymbol{a}^\top \boldsymbol{x}\right)^2$ is maximized. Since the generation mechanism is designed a priori, such a maximization cannot be done individually for each instance of the signal $\boldsymbol{x}$. Rather, it is sensible to look for maximization of the average energy.

Assuming that the statistic of the vector $\boldsymbol{a}$ is our design parameter, we may define

$$\rho(\boldsymbol{a}, \boldsymbol{x}) = \mathbf{E}_{\boldsymbol{a},\boldsymbol{x}} \left[\left(\boldsymbol{a}^\top \boldsymbol{x}\right)^2 \right] \tag{3.1}$$

M. Mangia et al., *Adapted Compressed Sensing for Effective Hardware Implementations*, DOI 10.1007/978-3-319-61373-4_3

and look for

$$\operatorname{argmax}_{f_{\boldsymbol{a}|\boldsymbol{x}}} \ \rho(\boldsymbol{a},\boldsymbol{x})$$

where $f_{\boldsymbol{a}|\boldsymbol{x}}$ is the conditioned probability density function of $\boldsymbol{a}$ given $\boldsymbol{x}_j$.

The quantity $\rho(\boldsymbol{a},\boldsymbol{x})$ to maximize is the average ability of the process generating the rows of $\boldsymbol{A}$ of collecting energy from the signal $\boldsymbol{x}$ and is called *rakeness*. Clearly $\rho(\alpha\boldsymbol{a},\boldsymbol{x}) = \alpha^2\rho(\boldsymbol{a},\boldsymbol{x})$ for any $\alpha \in \mathbb{R}$ so that the above maximization has no sense if we do not set a constraint that prevents scaling from generating solutions that are seen as different. Since we are dealing with energy, it is most natural to require that the average energy of $\boldsymbol{a}$ is fixed so that our design is synthesized by the following optimization problem:

$$\begin{aligned} &\operatorname*{argmax}_{f_{\boldsymbol{a}|\boldsymbol{x}}} \ \rho(\boldsymbol{a},\boldsymbol{x}) \\ &\quad \text{s.t. } \mathbf{E}_{\boldsymbol{a}}\left[\|\boldsymbol{a}\|_2^2\right] = 1 \end{aligned}$$

where we have decided that the energy of the rows of $\boldsymbol{A}$ is fixed to 1.

To proceed, note that $\boldsymbol{a}$ and $\boldsymbol{x}$ may be assumed independent so that $f_{\boldsymbol{a}|\boldsymbol{x}} = f_{\boldsymbol{a}}$. In fact, even if our goal is to make the two statistics related, the generation of $\boldsymbol{x}$ is due to the process to acquire while the generation of $\boldsymbol{a}$ is a task of the acquisition system that we assume to have no knowledge of the specific instance it is going to acquire. Exploiting this independence we may write

$$\begin{aligned} \rho(\boldsymbol{a},\boldsymbol{x}) = \mathbf{E}_{\boldsymbol{a},\boldsymbol{x}}\left[\left(\boldsymbol{a}^\top\boldsymbol{x}\right)^2\right] = \mathbf{E}_{\boldsymbol{a},\boldsymbol{x}}\left[\boldsymbol{a}^\top\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{a}\right] = \mathbf{E}_{\boldsymbol{a},\boldsymbol{x}}\left[\operatorname{tr}\left(\boldsymbol{a}\boldsymbol{a}^\top\boldsymbol{x}\boldsymbol{x}^\top\right)\right] = \\ = \operatorname{tr}\left(\mathbf{E}_{\boldsymbol{a},\boldsymbol{x}}\left[\boldsymbol{a}\boldsymbol{a}^\top\boldsymbol{x}\boldsymbol{x}^\top\right]\right) = \operatorname{tr}\left(\mathbf{E}_{\boldsymbol{a}}\left[\boldsymbol{a}\boldsymbol{a}^\top\right]\mathbf{E}_{\boldsymbol{x}}\left[\boldsymbol{x}\boldsymbol{x}^\top\right]\right) = \operatorname{tr}\left(\mathscr{A}\mathscr{X}\right) \end{aligned}$$

where we have introduced the correlation matrices $\mathscr{A} = \mathbf{E}_{\boldsymbol{a}}\left[\boldsymbol{a}\boldsymbol{a}^\top\right]$ and $\mathscr{X} = \mathbf{E}_{\boldsymbol{x}}\left[\boldsymbol{x}\boldsymbol{x}^\top\right]$ that are symmetric and positive semidefinite. Hence, independence allows us to simplify the design of $f_{\boldsymbol{a}}$ into the design of the second-order statistic of $\boldsymbol{a}$ depending on the second-order statistic of $\boldsymbol{x}$. By noting that $\mathbf{E}_{\boldsymbol{a}}\left[\|\boldsymbol{a}\|_2^2\right] = \sum_{j=0}^{n-1}\mathbf{E}_{\boldsymbol{a}}\left[a_j^2\right] = \operatorname{tr}\left(\mathscr{A}\right)$ and recalling that a correlation matrix of a real random vector must be positive semidefinite ($\mathscr{A} \succeq 0$) and symmetric ($\mathscr{A} = \mathscr{A}^\top$) we may reformulate our design problem as

$$\begin{aligned} &\operatorname*{argmax}_{\mathscr{A}\in\mathbb{R}^{n\times n}} \ \operatorname{tr}\left(\mathscr{A}\mathscr{X}\right) \\ &\qquad \operatorname{tr}\left(\mathscr{A}\right) = 1 \\ &\quad \text{s.t. } \mathscr{A} \succeq 0 \\ &\qquad \mathscr{A} = \mathscr{A}^\top \end{aligned} \tag{3.2}$$

This formulation immediately highlights what we loose in changing from an *adaptive* to an *adapted* mechanism, i.e., in optimizing based on ensemble features rather than on the features of each instance. In fact, if the process to sense is stationary and white, one has $\mathscr{X} = \sigma^2 \boldsymbol{I}$, where σ^2 is the power of the process and $\boldsymbol{I}$ the $n \times n$ identity matrix. With this $\mathrm{tr}\,(\mathscr{A}\mathscr{X}) = \sigma^2 \mathrm{tr}\,(\mathscr{A})$. Thanks to the power normalization constraint, the merit function of (3.2) is then fixed to $\mathrm{tr}\,(\mathscr{A}\mathscr{X}) = n\sigma^2$ and no optimization is possible. This means that the method we are developing will not work when the signal to sense is white or, using the terminology of Chap. 1, when its localization is null. Luckily enough, most real-world signals are not white and this is not a fatal weakness.

In general, from an optimization point of view, it is interesting to note that the trace of the product of two symmetric matrices is actually a scalar product between them, that induces the Frobenius norm since for any two $n \times n$ symmetric matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ one has

$$\mathrm{tr}\,(\boldsymbol{PQ}) = \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} \boldsymbol{P}_{j,k} \boldsymbol{Q}_{j,k} \qquad \text{and} \qquad \sqrt{\mathrm{tr}\,(\boldsymbol{P}^2)} = \sqrt{\sum_{j=0}^{n-1} \sum_{k=0}^{n-1} \boldsymbol{P}_{j,k}^2}$$

Hence, rakeness is linear in our degrees of freedom and its gradient is

$$\nabla_{\mathscr{A}} \rho(\boldsymbol{a}, \boldsymbol{x}) = \mathscr{X} \tag{3.3}$$

Moreover, the subspace of symmetric, positive-semidefinite matrices with a given trace is convex and thus (3.2) is a convex programming problem.

Actually, it is a very simple one, whose solution can be derived considering the eigenvector decomposition $\mathscr{X} = \boldsymbol{UMU}^\top$ where the diagonal matrix $\boldsymbol{M} = \mathrm{diag}\,(\mu_0, \ldots, \mu_{n-1})$ contains the eigenvalues μ_j of $\mathscr{X}$ and the matrix $\boldsymbol{U}$ aligns as columns the corresponding eigenvectors $\boldsymbol{u}_j$ scaled to be orthonormal. If the eigenvalues are sorted so that $\mu_0 \geq \mu_1 \geq \cdots \geq \mu_{n-1} \geq 0$, then the solution of (3.2) is $\mathscr{A} = n\boldsymbol{u}_0 \boldsymbol{u}_0^\top$. A process with such a correlation matrix is a degenerate one in which all the instances are equal to $\boldsymbol{u}_0$. Clearly, such a process cannot generate independent rows for $\boldsymbol{A}$.

Yet, all this suggests that an approach close to the well-known method of Principal Component Analysis (PCA) may be of interest. PCA is an average-energy-driven analysis technique that aims at finding which subspace of the whole signal space contains, on average, most of the energy of the signal. If one sets the dimensionality of the subspace to m, then it turns out that to contain the largest possible fraction of the signal energy, the subspace itself must be the span of $\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{m-1}$ (where, as before, we have assumed that the corresponding eigenvalues are sorted in non-increasing order). The m eigenvectors corresponding to the m largest eigenvalues are called the *principal components* of $\boldsymbol{x}$.

Within the framework of PCA, one may think to momentarily relax the assumption that $\boldsymbol{A}$ has independent rows and build it row by row picking properly normalized versions of the first m principal components. Such a strategy can be

easily put to test by using the same toy configurations introduced and used in Chap. 2. The results are reported in Fig. 3.1 where we compare the performance of classical CS with PCA-based CS for different values of κ and localization. Note that this time we avoid white signals since we already know that the method we are developing cannot yield improvements since it considers average energies.

To interpret the results one may consider that from the projections $y_0, \ldots, y_{m-1}$ on the first m principal components $\boldsymbol{x}$ could be estimated as $\hat{\boldsymbol{x}} = \sum_{j=0}^{m-1} y_j \boldsymbol{u}_j$ with an average error

$$\mathbf{E}\left[\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2^2\right] = \sum_{j=m}^{n-1} \mu_j \tag{3.4}$$

Though our reconstruction does not hinge on a least-square principle, when such an error becomes extremely small, one may reasonably expect that even the sparsity-based reconstruction becomes very good. This is what happens for very high-localization signals (the HL signals in the last row of Fig. 3.1) since when $\mathscr{L}_x$ is very high, the sequence of eigenvalues of $\mathscr{X}$, once sorted in descending order, exhibit a rapidly vanishing trend that makes (3.4) very small for relatively low m. On the contrary, for low-localization signals, PCA-based CS performs even worse than classical CS.

Moreover, overall, PCA-based CS suffers from the increase in sparsity κ, that modifies the very shape of the performance curves worsening it consistently (as seen in the $\kappa = 24$ curves in Fig. 3.1).

All this suggests that, to exploit the average energy maximization criterion in a more *robust* way one should avoid overspecialization of the matrix $\boldsymbol{A}$, i.e., allow that the projections span also less energetic direction in the light of two fundamental ideas: (i) since energy is considered only on average, excluding less (on average) energetic directions means leaving out subspaces that are indeed visited by the instances of the signal; (ii) the energy raked from the signal is only one of the points on which reconstruction is based, the other being sparsity, so that focusing only on the former may be suboptimal in a wide variety of cases.

3.2 Rakeness-Localization Trade-Off

The mathematical tool we may use to model this qualitative intuition is localization. The "process" that generates each row of the matrix $\boldsymbol{A}$ in PCA-based CS is a degenerate one yielding always the same instance and thus it is maximally localized. Since $\boldsymbol{A}$ represents the set of directions along which we are probing the signals, such a maximum localization implies maximum specialization of the probes. To prevent this from happening we may first revert to random and independent generation of the rows of $\boldsymbol{A}$ and then put a constraint on the localization of the process generating each of them.

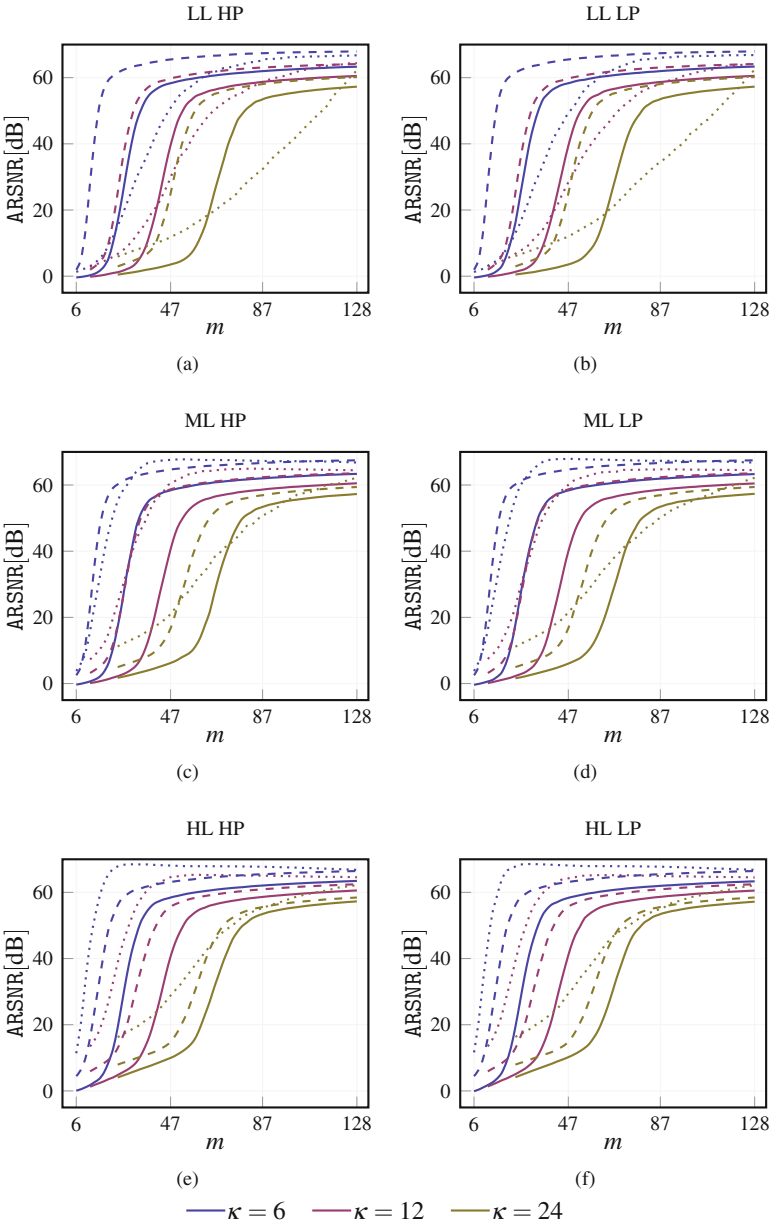


Fig. 3.1 Montecarlo comparison between performance of classical CS (*solid*), max-energy CS (*dashed*), and PCA-based CS (*dotted*). Only ARSNR is shown

A sensible adjustment of (3.2) is then

$$\begin{aligned} \underset{\mathscr{A}\in\mathbb{R}^{n\times n}}{\operatorname{argmax}} \ & \operatorname{tr}(\mathscr{A}\mathscr{X}) \\ \text{s.t.} \ & \operatorname{tr}(\mathscr{A}) = 1 \\ & \mathscr{A} \succeq 0 \\ & \mathscr{A} = \mathscr{A}^\top \\ & \mathscr{L}_{\boldsymbol{a}} \leq \mathscr{L}_{\boldsymbol{a}}^{\max} \end{aligned} \tag{3.5}$$

where we have introduced a bound on $\mathscr{L}_{\boldsymbol{a}}$. Clearly the new parameter $\mathscr{L}_{\boldsymbol{a}}^{\max} > 0$ administers the trade-off between maximizing rakeness and preserving a not-too-high localization of the probing process.

Assuming for $\mathscr{A}$ a spectral decomposition of the kind $\mathscr{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$ with $\boldsymbol{V}$ the matrix of orthonormal eigenvectors and $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_0, \ldots, \lambda_{n-1})$ the diagonal matrix of eigenvalues such that $\lambda_0 \geq \cdots \geq \lambda_{n-1}$, (3.5) can be recast into

$$\begin{aligned} \underset{\boldsymbol{V}\in\mathbb{R}^{n\times n},\lambda_0,\ldots,\lambda_{n-1}}{\operatorname{argmax}} \ & \operatorname{tr}\left(\boldsymbol{V}\operatorname{diag}(\lambda_0, \ldots, \lambda_{n-1})\,\boldsymbol{V}^\top\mathscr{X}\right) \\ \text{s.t.} \ & \boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I} \\ & \textstyle\sum_{j=0}^{n-1} \lambda_j = 1 \\ & \lambda_0 \geq \cdots \geq \lambda_{n-1} \geq 0 \\ & \textstyle\sum_{j=0}^{n-1} \left(\lambda_j - \frac{1}{n}\right)^2 \leq \mathscr{L}_{\boldsymbol{a}}^{\max} \end{aligned} \tag{3.6}$$

in which the constraints do not depend on the choice of the matrix $\boldsymbol{V}$. Hence, the two sets of available degrees of freedom (the set of eigenvectors in $\boldsymbol{V}$ and the set of eigenvalues) can be chosen independently.

Yet, whatever values are decided for the eigenvalues, the Wielandt–Hoffman inequality [5, Theorem 4.3.53] says that, since $\mu_0 \geq \cdots \geq \mu_{n-1} \geq 0$ and $\lambda_0 \geq \cdots \geq \lambda_{n-1} \geq 0$, then

$$\operatorname{tr}\left(\boldsymbol{V}\operatorname{diag}(\lambda_0, \ldots, \lambda_{n-1})\,\boldsymbol{V}^\top\boldsymbol{U}\operatorname{diag}(\mu_0, \ldots, \mu_{n-1})\,\boldsymbol{U}^\top\right) \leq \sum_{j=0}^{n-1} \lambda_j\mu_j$$

where equality is obtained for $\boldsymbol{V} = \boldsymbol{U}$. Hence $\boldsymbol{V} = \boldsymbol{U}$ is a condition for the optimum. With this (3.6) becomes

$$\begin{aligned} \underset{\lambda_0,\ldots,\lambda_{n-1}}{\operatorname{argmax}} \ & \textstyle\sum_{j=0}^{n-1} \lambda_j\mu_j \\ \text{s.t.} \ & \textstyle\sum_{j=0}^{n-1} \lambda_j = 1 \\ & \lambda_0 \geq \cdots \geq \lambda_{n-1} \geq 0 \\ & \textstyle\sum_{j=0}^{n-1} \left(\lambda_j - \frac{1}{n}\right)^2 \leq \mathscr{L}_{\boldsymbol{a}}^{\max} \end{aligned} \tag{3.7}$$

that highlights the structure of a linear merit function with linear and quadratic constraints.

The solution of (3.7) can be obtained in analytical terms [6], for example, by applying the Karush–Kuhn–Tucker conditions for optimality, and can be written depending on an integer $1 \leq J < n$ and on the two quantities

$$\Sigma_1(J) = \sum_{j=0}^{J-1} \mu_j \qquad \Sigma_2(J) = \sum_{j=0}^{J-1} \mu_j^2$$

that can be matched with the definition of localization in (1.5) to note that $\mathscr{L}_{\boldsymbol{x}} = {}^{\Sigma_2(n)}/_{\Sigma_1^2(n)} - {}^{1}/_{n}$. Based on these we may define

$$t(J) = \sqrt{\frac{\mathscr{L}_{\boldsymbol{a}}^{\max}}{\dfrac{\Sigma_2(J)}{\Sigma_1^2(J)} - \dfrac{1}{J}}}$$

and the sequence

$$\lambda_j(J) = \frac{\mu_j}{\Sigma_1(J)} t(J) + \frac{1}{J}\left[1 - t(J)\right]$$

that, for $j = 0, \ldots, J-1$ is an affine combination of the normalized sequence of the first J eigenvalues of $\mathscr{X}$ with the uniform sequence ${}^{1}/_{J}$, depending on the coefficient $t(J)$. If

$$J = \max\left\{J | \lambda_{J-1}(J) \geq 0\right\}$$

then the sequence of eigenvalues λ_j solving (3.5) is

$$\lambda_j = \begin{cases} \lambda_j(J) & \text{for } j = 0, \ldots, J-1 \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

In the particular case $J = n$ [2] , $t(n)$ becomes $\sqrt{{}^{\mathscr{L}_{\boldsymbol{a}}^{\max}}/_{\mathscr{L}_{\boldsymbol{x}}}}$ and one has the more readable

$$\lambda_j = \frac{\mu_j}{\operatorname{tr}(\mathscr{X})} \sqrt{\frac{\mathscr{L}_{\boldsymbol{a}}^{\max}}{\mathscr{L}_{\boldsymbol{x}}}} + \frac{1}{n}\left(1 - \sqrt{\frac{\mathscr{L}_{\boldsymbol{a}}^{\max}}{\mathscr{L}_{\boldsymbol{x}}}}\right) \tag{3.9}$$

that is surely valid whenever $\mathscr{L}_{\boldsymbol{a}}^{\max} \leq \mathscr{L}_{\boldsymbol{x}}$, i.e., when the process generating the rows of $\boldsymbol{A}$ is not more localized than the process to sense.

This last expression clearly shows how the need to increase localization to maximize the rakeness interacts with the localization constraint, which forces a white component (the uniform sequence ${}^{1}/_{n}$) into the blend that yields the optimal

eigenvalue sequence. When the localization constraint is strict, i.e., $\mathscr{L}_{\boldsymbol{a}}^{\max} \to 0$, the white component becomes the only ingredient in such a blend and classical CS is retrieved.

Actually, though (3.9) is not fully general, it is so simple that it is the main design tool for a rakeness-based design when we are allowed to neglect the additional constraints that an implementation may pose (and that will be addressed in the next chapter). In using it, the original degree of freedom $\mathscr{L}_{\boldsymbol{a}}^{\max}$ can be replaced by a generic real number $t > 0$ so that

$$\lambda_j = \frac{\mu_j}{\mathrm{tr}\,(\mathscr{X})} t + \frac{1}{n}(1-t) \quad \text{for } j = 0, \ldots, n-1 \tag{3.10}$$

or, equivalently,

$$\mathscr{A} = \frac{\mathscr{X}}{\mathrm{tr}\,(\mathscr{X})} t + \frac{1}{n} \boldsymbol{I}(1-t)$$

providing that $\lambda_{n-1} \geq 0$, i.e., for

$$0 \leq t \leq \frac{\mathrm{tr}\,(\mathscr{X})}{\mathrm{tr}\,(\mathscr{X}) - n\mu_{n-1}} \tag{3.11}$$

usually, $0 \leq t \leq 1$ is assumed.

Figure 3.2 tries a visual explanation of (3.5) and its solutions (3.8) and (3.10) for $n = 3$. In that case, each point of the three-dimensional space corresponds to a possible assignment of the eigenvalues $\lambda_0, \lambda_1, \lambda_2$. The trace constraint $\mathrm{tr}(\mathscr{A}) = 1$ translates into the requirements that the point representing the solution lies on the $\lambda_0 + \lambda_1 + \lambda_2 = 1$ plane.

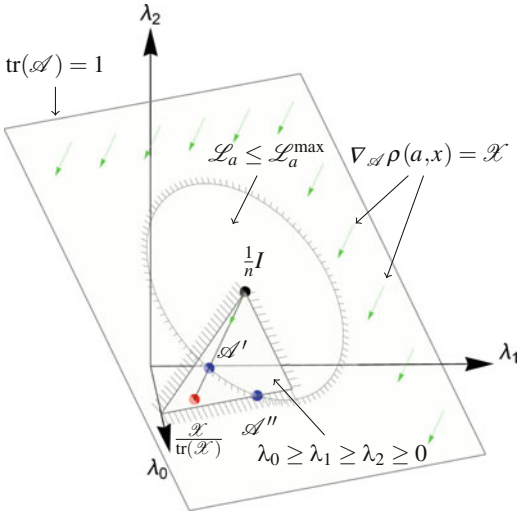


Fig. 3.2 A low-dimensional example of (3.5) and its solution (3.10)

From (3.3) we know that the gradient of the merit function of (3.5) is $\mathscr{X}$ itself, that, once projected on the constraining plane, yields the direction (green arrows) towards which one should go to increase rakeness. On that same plane, the localization constraint $\mathscr{L}_{\boldsymbol{a}} \leq \mathscr{L}_{\boldsymbol{a}}^{\max}$ is equivalent to require that the point representing the solution does not fall outside a circle centered in $(1/3, 1/3, 1/3)$ that represents $\frac{1}{n}\boldsymbol{I}$. Moreover, the constraint $\lambda_2 \geq \lambda_1 \geq \lambda_1 \geq 0$ identifies on the same plane a triangle that must contain the solution.

If the eigenvalues of $\mathscr{X}$ are properly sorted, the point representing $\mathscr{X}/\mathrm{tr}(\mathscr{X})$ lies within the same triangle but not necessarily within the circle corresponding to the localization constraint. In that case, since the gradient of the merit function is constantly pointing from $\frac{1}{n}\boldsymbol{I}$ to $\mathscr{X}/\mathrm{tr}(\mathscr{X})$, the solution of (3.5) is at the intersection of the segment connecting $\frac{1}{n}\boldsymbol{I}$ and $\mathscr{X}/\mathrm{tr}(\mathscr{X})$ and the boundary of the localization circle. General solutions (3.8) in which $J < n$ are represented by points like $\mathscr{A}''$ in Fig. 3.2 since they set to zero the last eigenvalues of the solution ($\lambda_2 = 0$ in this case).

In summary, all the above procedure is concretely summarized in the very simple four-steps design flow reported in Fig. 3.3.

As a final remark on the design flow, if the simplified solution (3.10) is employed, one may look back at (3.5) to discover that the corresponding value of rakeness is

$$\begin{aligned}\rho^*(\boldsymbol{a},\boldsymbol{x}) &= \sum_{j=0}^{n-1}\lambda_j\mu_j = \sum_{j=0}^{n-1} t\frac{\mu_j^2}{\mathrm{tr}\,(\mathscr{X})} + (1-t)\,\frac{1}{n}\sum_{j=0}^{n-1}\mu_j \\ &= t\frac{\mathrm{tr}\left(\mathscr{X}^2\right)}{\mathrm{tr}\,(\mathscr{X})} + (1-t)\,\frac{1}{n}\mathrm{tr}\,(\mathscr{X}) \\ &= \mathrm{tr}\,(\mathscr{X})\left\{t\mathscr{L}_{\boldsymbol{x}} + \frac{1}{n}\right\}\end{aligned} \tag{3.12}$$

From such an expression we learn a few things

- if t is fixed, $\rho^*(\boldsymbol{a},\boldsymbol{x})$ is increasing in $\mathscr{L}_{\boldsymbol{x}}$, i.e., the more localized the signal, the larger the raked energy;
- if $\mathscr{L}_x$ is fixed, $\rho^*(\boldsymbol{a},\boldsymbol{x})$ is increasing in t, i.e., the more we relax the constraint on the localization of the rows of the sensing matrix, the larger the energy we rake;
- for $t = 0$, $\boldsymbol{a}$ is white and the energy it rakes from $\boldsymbol{x}$ is $\mathrm{tr}(\mathscr{X})/n$, i.e., the average energy of a single component of $\boldsymbol{x}$.

It is evident that the design flow depends on the choice of t whose optimal value may, in principle, depend on the signal $\boldsymbol{x}$ and on the reconstruction quality one is aiming at. In practice, it can be empirically verified that the sensitivity of performance on t is quite small so that one may coarsely sample the design space in t and assess performance by simulation to choose the best value. This is the approach we used to produce the results shown in Figs. 3.4, 3.5, and 3.6, where the design flow of rakeness-based CS in Fig. 3.3 is followed for $t = 0.1, 0.3, 0.5, 0.7, 0.9$ and the best performing option is considered for each configuration.

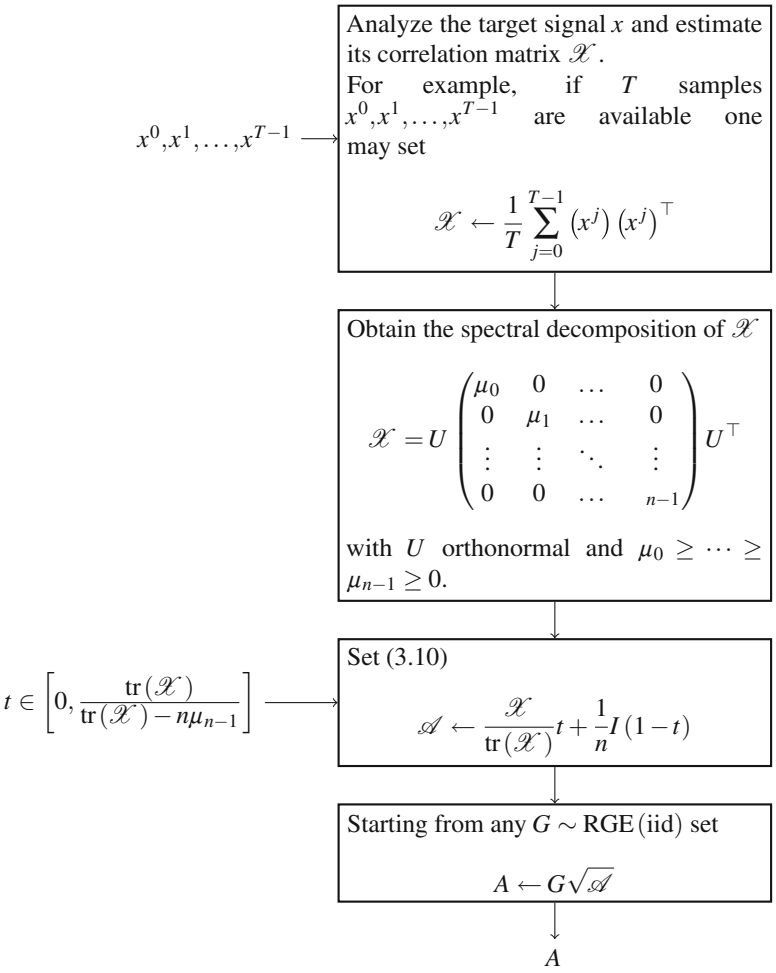


Fig. 3.3 Rakeness-based design flow that starts from samples of the signal to sense and from a choice of the parameter t

Figure 3.4 shows what performance can be expected for signals with a low localization. Rakeness-based CS is not able to perform as well as maximum-energy CS.

This is due to the fact that maximum-energy CS adapts to each single instance (at the price of an increased computational complexity and a communication overhead) while rakeness adapts sensing to the average behavior of the signal (and does not require any significant overhead).

In any case, all performance curves show a definite improvement with respect to classical CS both in the high-pass and low-pass cases that, despite the underlying signals $\boldsymbol{x}$ are completely different, result in almost identical trends.

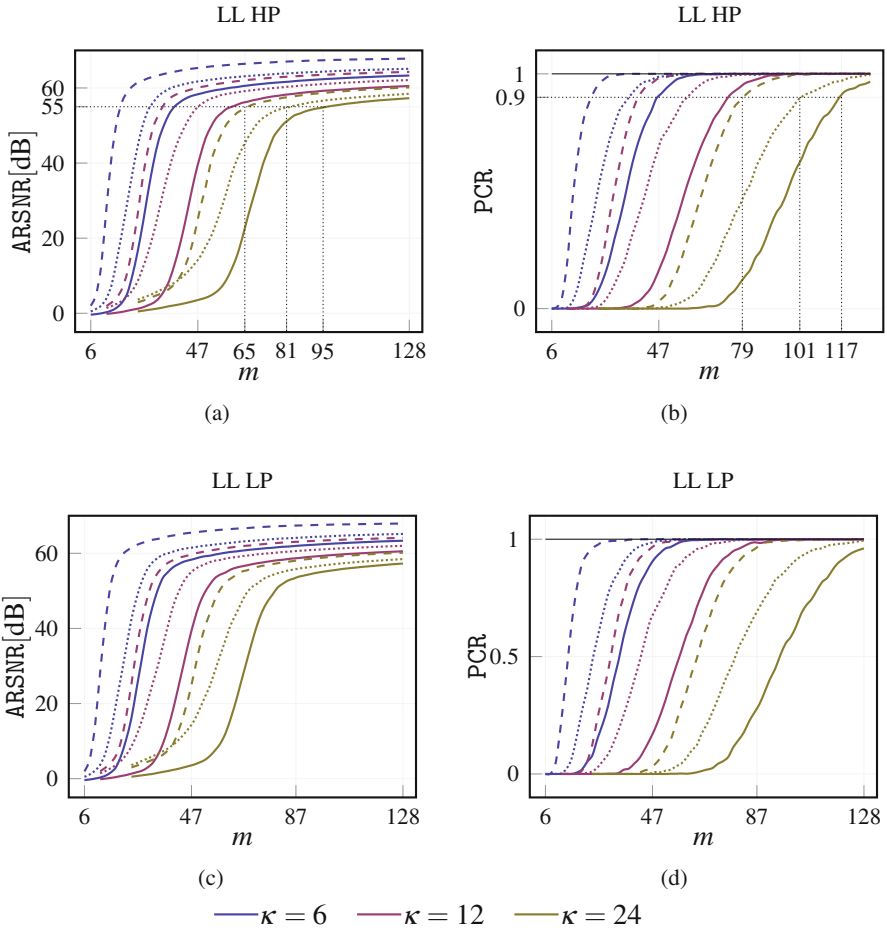


Fig. 3.4 Montecarlo comparison between performance of classical CS (*solid*), max-energy CS (*dashed*), and rakeness-based CS (*dotted*) for low-localization signals

As an example, even in the most unfavorable situation like the one in Fig. 3.4a and b that deals with a low-localization high-pass signal with sparsity $\kappa = 24$, to ensure $\texttt{ARSNR} = 55\,\text{dB}$, classical CS needs $m^* = 95$ measurements while rakeness-based CS requires $m^* = 81$ measurements and this brings from $\texttt{CR} \simeq 1.35$ to $\texttt{CR} \simeq 1.58$. If the specification is to have $\texttt{RSNR} \geq 55\,\text{dB}$ at least 90% of the times, then the adoption of rakeness-based CS brings from $\texttt{CR} \simeq 1.09$ to $\texttt{CR} \simeq 1.27$.

Figure 3.5 shows what performance can be expected for signals with a medium localization. In this case, despite the two techniques are different and with different computational costs, rakeness-based CS closely matches the performance of maximum-energy CS though it seems to suffer from an increase in sparsity (the $\kappa = 24$ curves are those for which the difference between rakeness-based CS and maximum-energy CS is relevant).

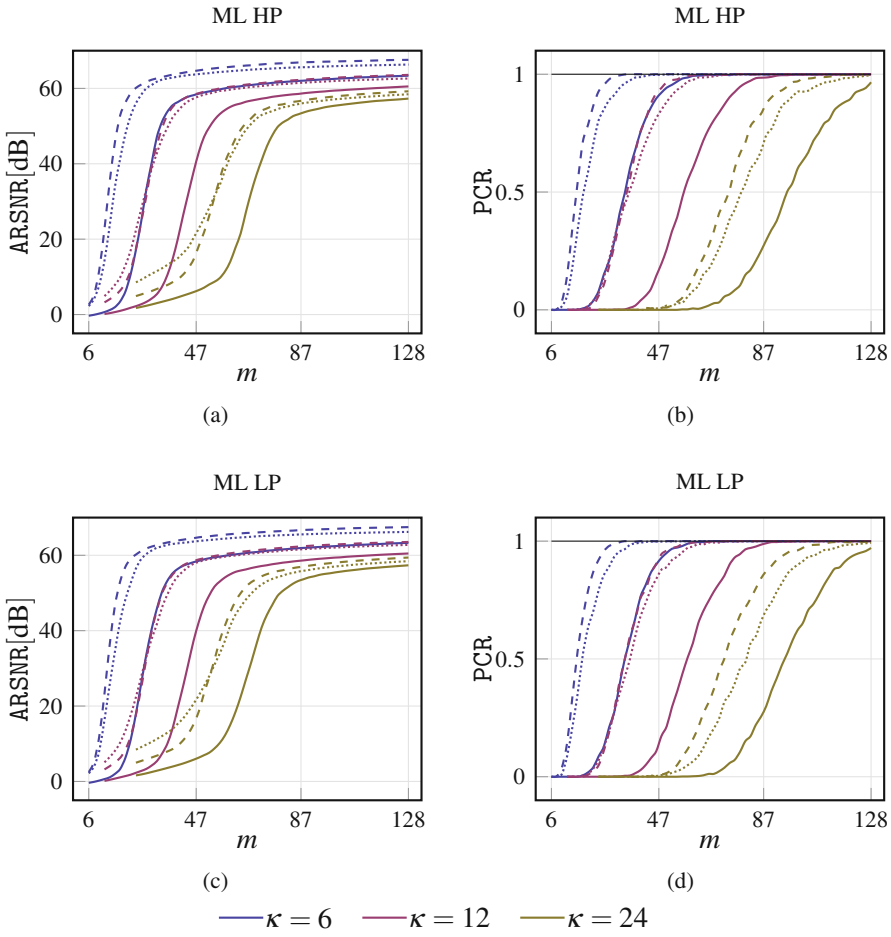


Fig. 3.5 Montecarlo comparison between performance of classical CS (*solid*), max-energy CS (*dashed*), and rakeness-based CS (*dotted*) for medium-localization signals

Figure 3.6 shows that the match between rakeness-based CS and maximum-energy CS increases as localization increases. As an example, in the very favorable case of Fig. 3.6c and d that deals with a high-localization low-pass signal with sparsity $\kappa = 6$, to ensure $\texttt{ARSNR} = 55\,\text{dB}$ classical CS requires $m^* = 38$ measurements while rakeness-based and maximum-energy CS require $m^* = 23$ and this brings from $\texttt{CR} \simeq 3.37$ to $\texttt{CR} \simeq 5.57$. If the specification is to have $\texttt{RSNR} \geq 55\,\text{dB}$ at least 90% of the times, then the adoption of rakeness-based CS brings from $\texttt{CR} \simeq 2.78$ to $\texttt{CR} \simeq 4.57$.

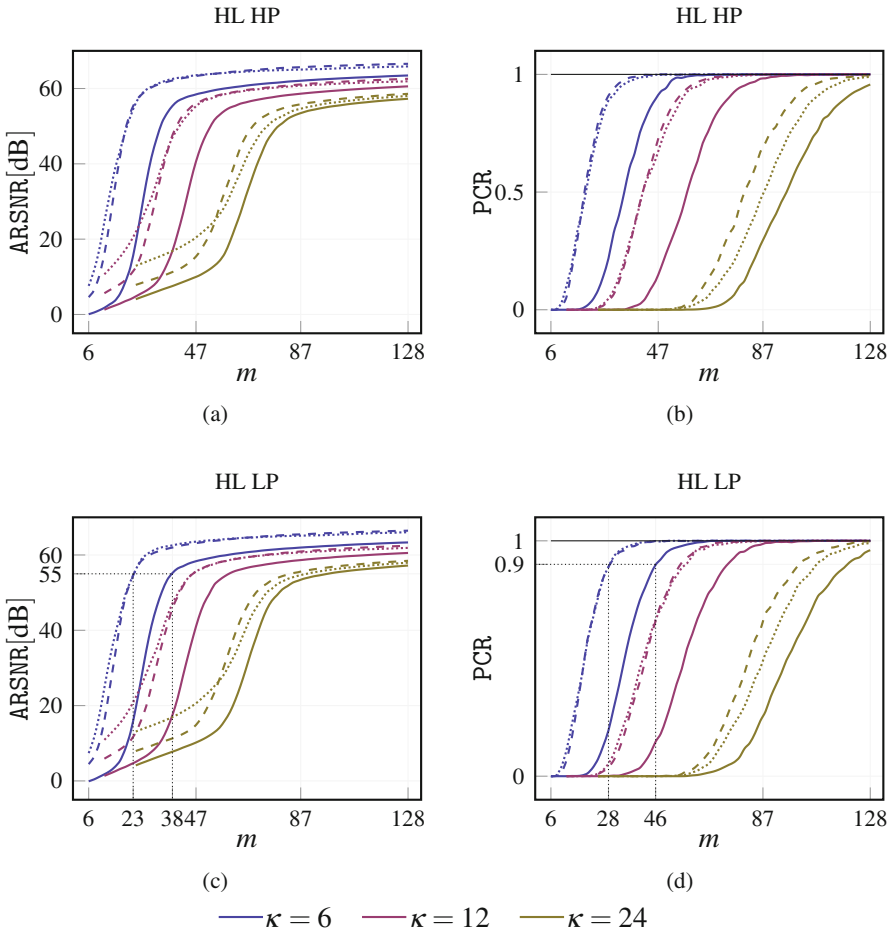


Fig. 3.6 Montecarlo comparison between performance of classical CS (*solid*), max-energy CS (*dashed*), and rakeness-based CS (*dotted*) for high-localization signals

To discuss the sensitivity of rakeness-based design with respect to the parameter t, we may compare the performance curves obtained by selecting the optimal t for each configuration with those resulting from $t = 1/2$. This is done in Fig. 3.7 focusing only on ARSNR. Though the solid curves are an upper bound on what can be obtained for $t = 1/2$, the two tracks are always very close and, what is most important, have almost identical phase transitions, i.e., the corresponding performance reach their maximum value for the same number of measurements. This phenomenon has been empirically verified in most cases and suggests to take $t = 1/2$ at least as a valid starting point for a first draft design.

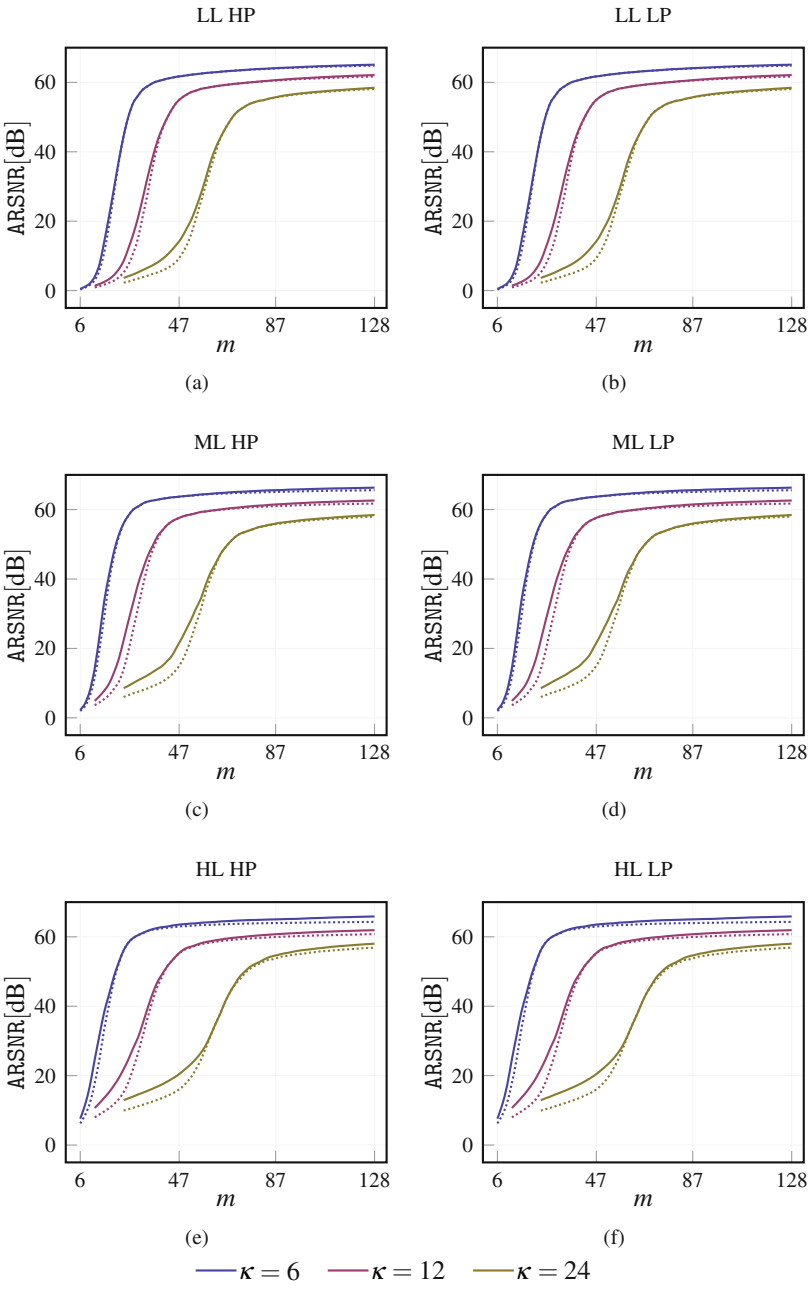


Fig. 3.7 Montecarlo comparison between performance of rakeness-based CS optimized in t (*solid*) and rakeness-based CS for $t = 1/2$ (*dotted*). Only ARSNR is reported

3.3 Rakeness and the Dark Side of Off-Line Adaptation

We have seen that rakeness-based CS is able to reproduce the performance improvement of maximum-energy CS for sufficiently localized signals, and that this is possible notwithstanding the much higher computational complexity of the latter which needs to compute $M \gg n \gg m$ measurements instead of only m.

All this does not come for free. In fact, rakeness-based CS relies on an off-line adaptation to the class of signals to sense, involving an optimization that is made at design-time. Clearly, this might be a problem whenever the characterization that can be given at design-time of the process to sense is only approximately true and there may be deviations at run-time.

As an example, one may want to acquire ECG tracks and to this aim analyzes a database of tracks acquired from healthy patients to extract the second-order information $\mathscr{X}$ needed to trigger rakeness-based design. Once put to work, the acquisition system may well be fed with non-healthy patients signals, for example, arrhythmic beats, whose frequency content is different from regular ones. As an alternative scenario, one may not have access a priori to samples of $\boldsymbol{x}$ and must base the design on reasonable but not exact assumptions on the signal to acquire, that result in approximated second-order features entering the design flow.

The question that naturally arises is whether rakeness-based CS is capable of acquiring these signals with a sufficient degree of accuracy.

Some specific versions of this problem will be addressed in future chapters that deal more specifically with implementations and corresponding applicative scenarios. By now, to keep the discussion as general as possible and give some quantitative background to the answer, we may formalize the situation as follows. If $\mathscr{X}$ is the true correlation matrix of $\boldsymbol{x}$, estimation errors or wrong a priori knowledge cause the rakeness-based design flow to consider a correlation matrix $\overline{\mathscr{X}} \neq \mathscr{X}$.

Unless the whole framework on which the design relies is heavily flawed, $\overline{\mathscr{X}}$ still contains some coarse-grain information on the signal though it may mistake some of its features. To model this, we will assume that the energy of each component of $\boldsymbol{x}$ (i.e., each of the diagonal elements of $\mathscr{X}$) is correctly identified while some uncertainty affects the cross-correlation terms in $\overline{\mathscr{X}}$.

More mathematically, note that if we start from the above spectral decomposition $\mathscr{X} = \boldsymbol{U}\mathrm{diag}\,(\mu_0, \ldots, \mu_{n-1})\,\boldsymbol{U}^\top$ and set $\boldsymbol{Q} = \boldsymbol{U}\mathrm{diag}\left(\sqrt{\mu_0}, \ldots, \sqrt{\mu_{n-1}}\right)\boldsymbol{U}^\top$ then $\boldsymbol{Q}$ is a positive-semidefinite and symmetric matrix such that $\boldsymbol{Q}\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^\top = \mathscr{X}$. From this we get that, if $\boldsymbol{q}_j$ is the j-th row of $\boldsymbol{Q}$, then $\boldsymbol{q}\boldsymbol{q}^\top = \|\boldsymbol{q}\|_2^2 = \mathscr{X}_{j,j} = \mathbf{E}[x_j^2]$.

Assume now that each row $\boldsymbol{q}_j$ is rotated by an angle ϵ in a random direction to yield the j-th row $\overline{\boldsymbol{q}}_j$ of a new matrix $\overline{\boldsymbol{Q}}$. Since rotations do not alter vector length, the matrix $\overline{\mathscr{X}} = \overline{\boldsymbol{Q}}\overline{\boldsymbol{Q}}^\top$ has the same diagonal entries of $\mathscr{X}$ and it is positive semidefinite and symmetric. It can be therefore assumed as a perturbation of $\mathscr{X}$ such that $\overline{\mathscr{X}} \to \mathscr{X}$ when $\epsilon \to 0$.

Looking at this from the opposite side, one may think that ϵ quantifies the error we make in estimating $\mathscr{X}$ that causes $\overline{\mathscr{X}}$ to enter the design instead of $\mathscr{X}$. To give an intuitive appreciation of how such an error modifies the information that are passed to the rakeness-based design flow we may adopt a geometric representation of the distribution of the energy in the signal space. For any given correlation matrix $\mathscr{C}$, the figure $\mathfrak{E}_{\mathscr{C}} = \left\{\xi \in \mathbb{R}^n | \xi^\top \mathscr{C}^{-1} \xi \leq 1\right\}$ is an ellipsoid whose axes align with the eigenvectors of $\mathscr{C}$ and $\mathscr{C}^{-1}$ and whose lengths are proportional to the corresponding eigenvalues of $\mathscr{C}$. Hence, the orientation and size of $\mathfrak{E}_{\mathscr{C}}$ geometrically represents the anisotropy of the energy distribution in the signal space: if the signal tend to align to a particular direction, $\mathfrak{E}_{\mathscr{C}}$ will be heavily elongated along that direction.

Applying this to our case, as $\mathscr{X}$ and $\overline{\mathscr{X}}$ have the same diagonal the sum of their eigenvalues is the same. From a geometric point of view the sum of the axes of $\mathfrak{E}_{\mathscr{X}}$ and of the axes of $\mathfrak{E}_{\overline{\mathscr{X}}}$ are the same. Within this constraint, the comparison between the shape of $\mathfrak{E}_{\mathscr{X}}$ and that of $\mathfrak{E}_{\overline{\mathscr{X}}}$ allows a visual appreciation of how much the distribution of signal energy assumed by rakeness-based design may be different from the true one.

Figure 3.8 reports such comparisons for $n = 3$ starting from $\mathscr{X}$ such that $\mathscr{X}_{i,j} = 2^{-|i-j|}$, and considering typical instances of $\overline{\mathscr{X}}$ for different values of ϵ. It is evident that for small values of ϵ (e.g., $\epsilon = \pi/50$) the energy distribution is almost identical while for larger values (e.g., $\epsilon = \pi/5$) what is considered by the design flow may be substantially different from what characterizes the signal to acquire.

The fact that $\overline{\mathscr{X}}$ is used in the design flow instead of $\mathscr{X}$ misleads the procedure and the resulting correlation matrix $\mathscr{A}$ is not the one solving (3.5). The rows of $\boldsymbol{A}$ generated according to such an $\mathscr{A}$ will have a suboptimal rakeness and can be expected to yield a lower performance. Performance plots are affected by ϵ as shown in Fig. 3.9 where we compare the performance of classical CS with that of rakeness-based CS based on a $\overline{\mathscr{X}}$ generated from $\mathscr{X}$ with $\epsilon = 0, \pi/100, \pi/50, \pi/20, \pi/10, \pi/5$ and assuming $t = 1/2$ in (3.10).

In those plots, arrows indicate how performance profile changes when ϵ increases, and their length gives a rough quantitative indication of the effect of the sensitivity to such an error. Though each case has its own peculiarities, it is clear that when the error is small the effect is almost negligible while larger differences cause a non-negligible degradation of performance. Yet, even in the case of large errors, the performance of rakeness-based CS usually does not become worse than that of classical CS.

A more quantitative appreciation of this can be obtained by considering the minimum number of measurements m^* needed to reach an ARSNR of 55 dB, i.e., 5 dB less than the ISNR with which the signal enters the acquisition system. Figure 3.10 reports how such an m^* changes when ϵ increases, and compares this with the minimum number of measurements needed to achieve the same

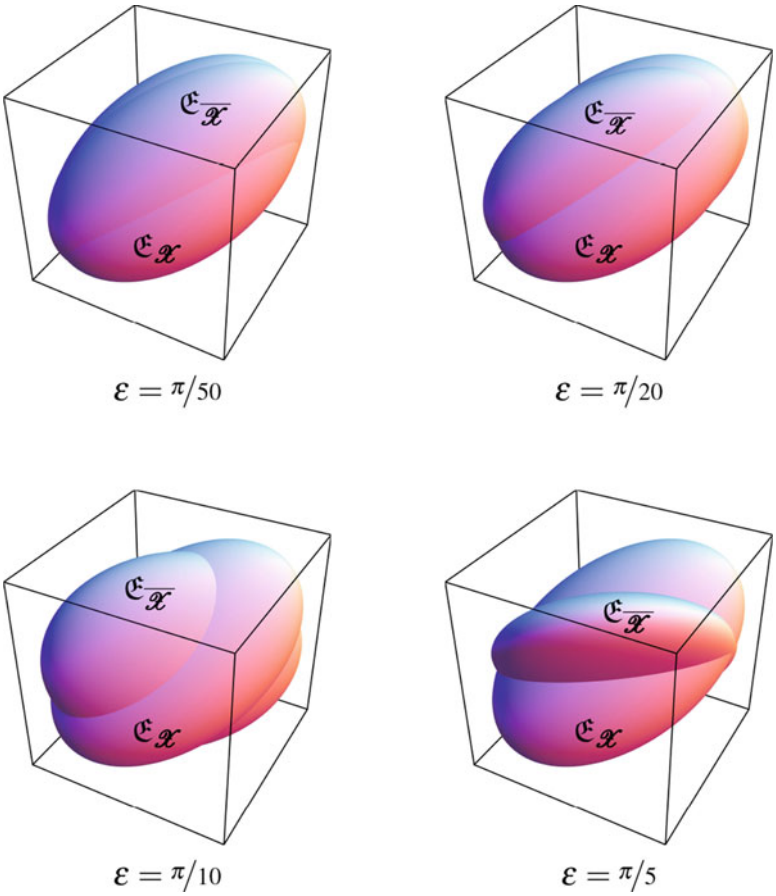


Fig. 3.8 A graphical intuition of how much the distribution of signal energy assumed by rakeness-based design (represented by $\mathfrak{E}_{\overline{\mathscr{X}}}$) may be different from the true one (represented by $\mathfrak{E}_{\mathscr{X}}$)

performance level by classical CS. Performance degradation reflects in the fact that the trends of m^* are increasing with m^* and tend to reduce the gap between classical CS and rakeness-based CS (i.e., what we gain in adopting rakeness-based design instead of classical CS). Yet, such a gap is not bridged completely but in the most challenging configuration, i.e., when the signal is not so sparse ($\kappa = 24$) and the error is sizeable ($\epsilon = \pi/5$).

Overall, when the second-order model of the signal to acquire is not badly mistaken, rakeness-based CS is able to yield improvement with respect to classical CS.

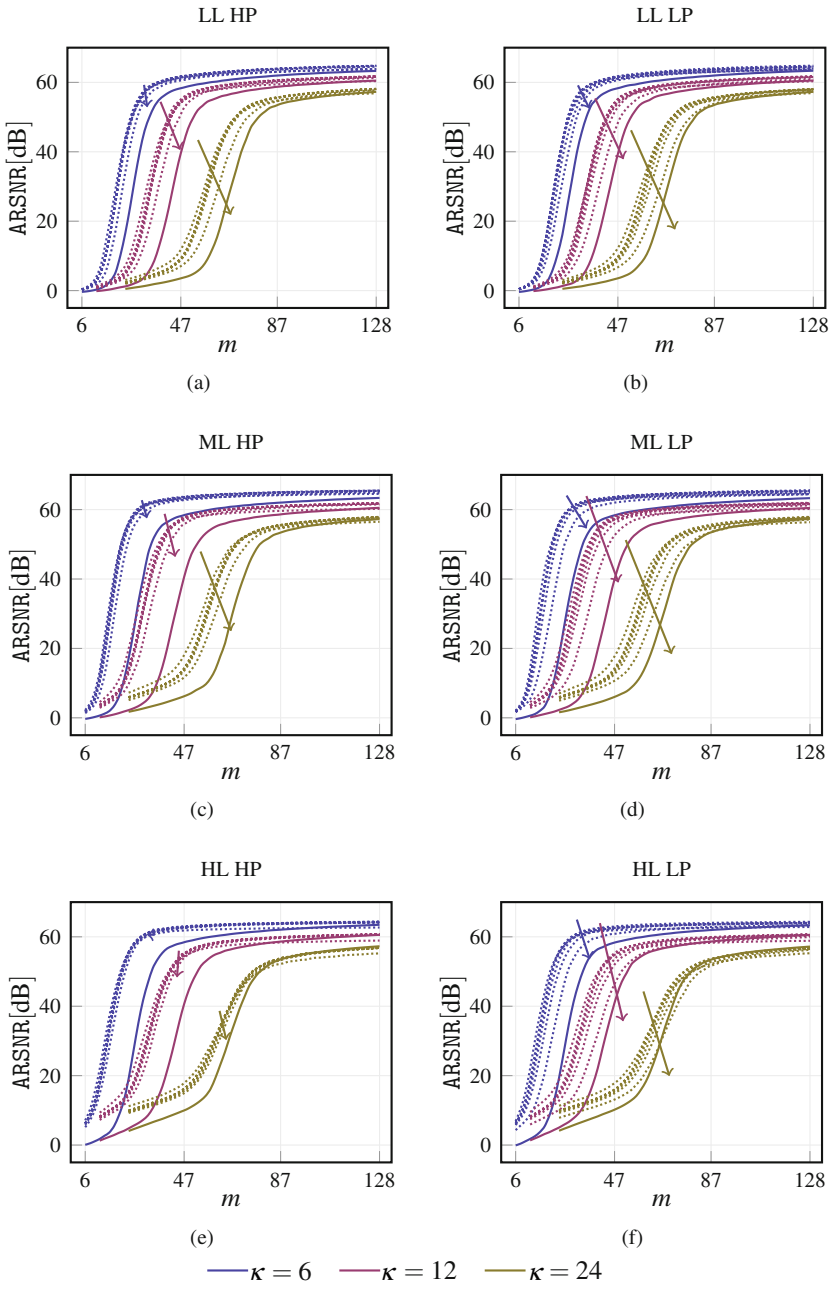


Fig. 3.9 Montecarlo comparison between performance of classical CS (*solid*) and rakeness-based CS (*dotted*) for different values of ϵ. Only ARSNR is shown. *Arrows* show how rakeness-based CS performance changes when ϵ increases, their length gives a quantitative idea of the effect

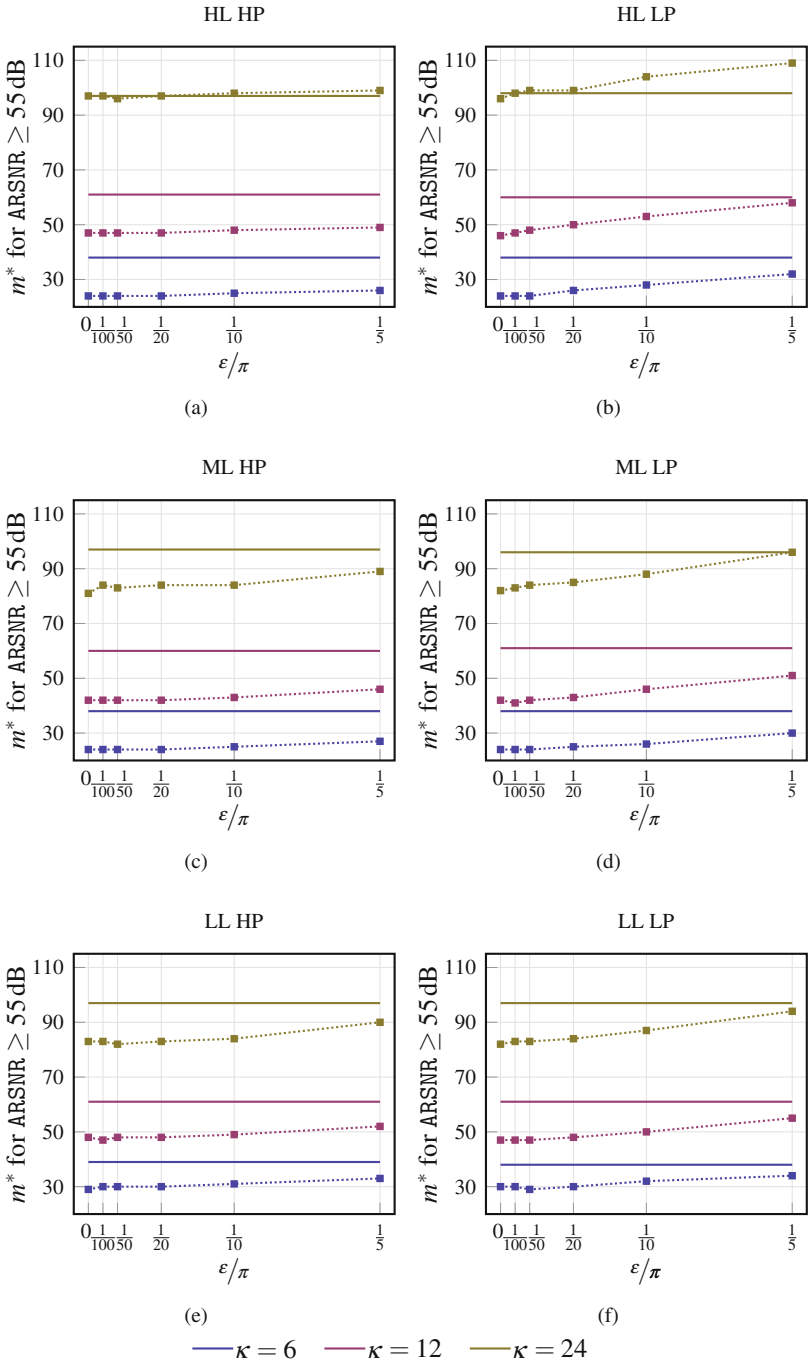


Fig. 3.10 The minimum number of measurements m^* needed by rakeness-based CS with $t = 1/2$ to achieve ARSNR $\geq$ 55 dB as a function of the perturbation angle ϵ (*dotted*) compared with that needed by classical CS (*solid*)

3.4 Rakeness and the Distribution of Measurements

As both the signal to acquire $\boldsymbol{x}$ and acquisition matrix $\boldsymbol{A}$ are random, the measurement vector $\boldsymbol{y}$ is a random vector and its components $\boldsymbol{y}_j$ are random variables.

Since rakeness-based design aims at increasing the average energy of each measurement, it surely alters the distribution of the $\boldsymbol{y}_j$, a distribution that is interesting for multiple reasons.

The first of these reasons is that measurements are stored or communicated in digital form and thus should be quantized either explicitly (if they are computed in analog terms) or implicitly (it they are computed digitally depending on previously converted data). The design of quantization schemes depends on the distribution of the scalar to quantize.

In the following chapters we will also see that the fact that the $\boldsymbol{y}_j$ exhibit a certain distribution is a key point in using CS not only for parsimonious acquisition but also as a mean to grant a limited but almost zero-cost form of privacy of the acquired data.

Results on the distributions of the y_j are better derived in asymptotic terms, i.e., when $n \to \infty$, and this requires some assumptions on the behavior of the sequence of random quantities making the rows of $\boldsymbol{A}$ and in the samples of $\boldsymbol{x}$.

To proceed formally we concentrate on a single measurement $y = \boldsymbol{a}^\top \boldsymbol{x}$, where $\boldsymbol{a}^\top$ is one of the rows of $\boldsymbol{A}$ and $\boldsymbol{x}^\top$ is the input signal. Since we are interested in $n \to \infty$ we should think to $\boldsymbol{a}$ and $\boldsymbol{x}$ as segments of two discrete-time random processes a_j and x_j that we will assume independent.

The formal development requires some assumptions to be put on these two processes, namely

- a_j and x_j are stationary;
- a_j and x_j are sufficiently mixing;
- $\mathbf{E}[a_j] = 0$
- $\mathbf{E}[a_j^{12}] < \infty$
- $\mathbf{E}[x_j^{12}] < \infty$

The last two assumptions are merely technical and are easily verified by any real-world implementation that, for example, restricts all its quantities in a finite range. Stationarity implies that the vectors $\boldsymbol{a}$ and $\boldsymbol{x}$ have the same statistical features independently of which part of the underlying process they copy and imposes that the corresponding correlation matrices $\mathscr{A}$ and $\mathscr{X}$ are Toeplitz.

Mixing implies that, though the processes may be made of dependent random variables, the dependency between these random variables decreases as they are pushed apart in time.

This is formalized by considering the generic process z_j associating a real random variable to each index j, two integers $p, q > 0$ and defining an event P on a set of q subsequent samples and an event Q on a set of p subsequent samples that are t times instant apart from those on which P is defined. The situation is the one described by Fig. 3.11 where we require that the two events tend to be independent as $t \to \infty$.

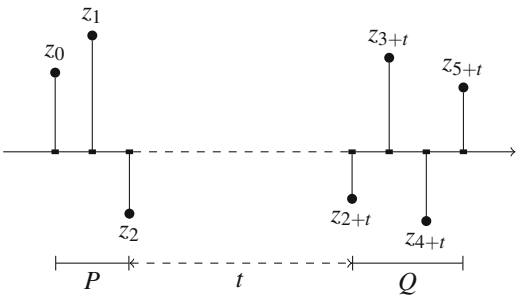


Fig. 3.11 An example of the two sets of samples on which the event P and Q are defined to introduce mixing. Independence arises when $t \to \infty$

In formulas, take any two measurable sets $P \subset \mathbb{R}^p$, $Q \subset \mathbb{R}^q$ and let

$$\pi_{P\times Q}(t) = \Pr\left\{\left(z_0, \ldots, z_{p-1}, z_{p-1+t}, \ldots, z_{p+q-2+t}\right) \in P \times Q\right\}$$

$$\pi_P = \Pr\left\{\left(z_0, \ldots, z_{p-1}\right) \in P\right\}$$

$$\pi_Q = \Pr\left\{\left(z_{p-1+t}, \ldots, z_{p+q-2+t}\right) \in Q\right\} = \Pr\left\{\left(z_0, \ldots, z_{q-1}\right) \in Q\right\}$$

In the following we will say that the process is *sufficiently mixing* if

$$|\pi_{P\times Q}(t) - \pi_P \pi_Q| = \mathsf{O}\left(t^{-5}\right)$$

Most common processes give rise to exponential decay of $|\pi_{P\times Q}(t) - \pi_P\pi_Q|$ and thus are mixing enough for our purposes. Moreover, if the processes underlying $\boldsymbol{a}$ and $\boldsymbol{x}$ are sufficiently mixing, also the process with samples $z_j = a_j x_j$ is sufficiently mixing and

$$y = \sum_{j=0}^{n-1} z_j$$

is therefore the sum of n subsequent samples of a mixing process. We also know that $\mathbf{E}[z_j] = \mathbf{E}[a_j x_j] = \mathbf{E}[a_j]\mathbf{E}[x_j] = 0$ and $\mathbf{E}[z_j^{12}] = \mathbf{E}[a_j^{12} x_j^{12}] = \mathbf{E}[a_j^{12}]\mathbf{E}[x_j^{12}] < \infty$.

Exploiting these assumptions we are able to define the asymptotic behavior of $y/\sqrt{n}$ as $n \to \infty$. In particular, we immediately get that $\mathbf{E}\left[y/\sqrt{n}\right] = 0$ and that, under all the above assumptions, we may exploit the most common version of central limit theorem that is able to cope with dependent random variables [1, Theorem 27.4] to say that, if

$$\lim_{n\to\infty} \frac{\mathbf{E}\left[y^2\right]}{n} = \sigma^2 > 0$$

then

$$\frac{y}{\sqrt{n}} \overset{n\to\infty}{\sim} \mathrm{N}\left(0, \sigma^2\right)$$

meaning that for large n, if the measurements are normalized to keep their energy finite, they tend to distribute as a Gaussian. The parameter σ^2 characterizing the limit distribution is clearly related to the rakeness. In fact, by definition $\mathbf{E}[y^2] = \rho(\boldsymbol{a}, \boldsymbol{x})$ and thus

$$\sigma^2 = \lim_{n\to\infty} \frac{1}{n}\rho(\boldsymbol{a}, \boldsymbol{x}) = \lim_{n\to\infty} \frac{1}{n} \sum_{j=0}^{n-1} \lambda_j \mu_j$$

If the $\boldsymbol{a}$ are drawn according to the simplified solution (3.10) of (3.5), then (3.12) can be used to anticipate that the variance of the measurements will increase as either the localization $\mathscr{L}_{\boldsymbol{x}}$ of the target system increases or as the constraint t on the localization of the rows of the sensing matrix is relaxed.

Figure 3.12 shows the practical effect of the latter property when n is finite. In particular, we concentrate on medium localization, low-pass signals that are $\kappa = 24$ sparse and perform a Montecarlo simulation to identify the empirical PDF f_y of a typical measurement for $t = 0.1, 0.5, 0.9$. The bell-shaped PDFs clearly increase their variance as t increases.

Figure 3.13 shows how the bell-shaped profiles of the empirical PDF of the measurement tend to be Gaussian. In fact, the solid lines are the Gaussian trends anticipated by the asymptotic theory when $t = 0.9$ and the histogram representing the empirical PDF clearly conforms to such a prediction as n increases from 64 to 512.

3.5 Rakeness Compared with Other Matrix Optimization Options

Rakeness-based design is not the only tool that appears in the Literature with the aim of building a matrix $\boldsymbol{A}$ to improve the performance of CS systems.

The other most significant attempts in such a field are inspired by the coherence concept as defined in Chap. 1 and its relationship with the idea of making the matrix $\boldsymbol{B} = \boldsymbol{AD}$ as close as possible to an equiangular frame (see Sect. 1.4).

In practice, the columns of the matrix $\boldsymbol{B}$ are seen as a collection of m-dimensional vectors whose coherence is the cosine of the largest angle between any two of them. Clearly, if such a collection has the properties of an equiangular frame, all the angles are equal and the maximum of such a cosine is minimized.

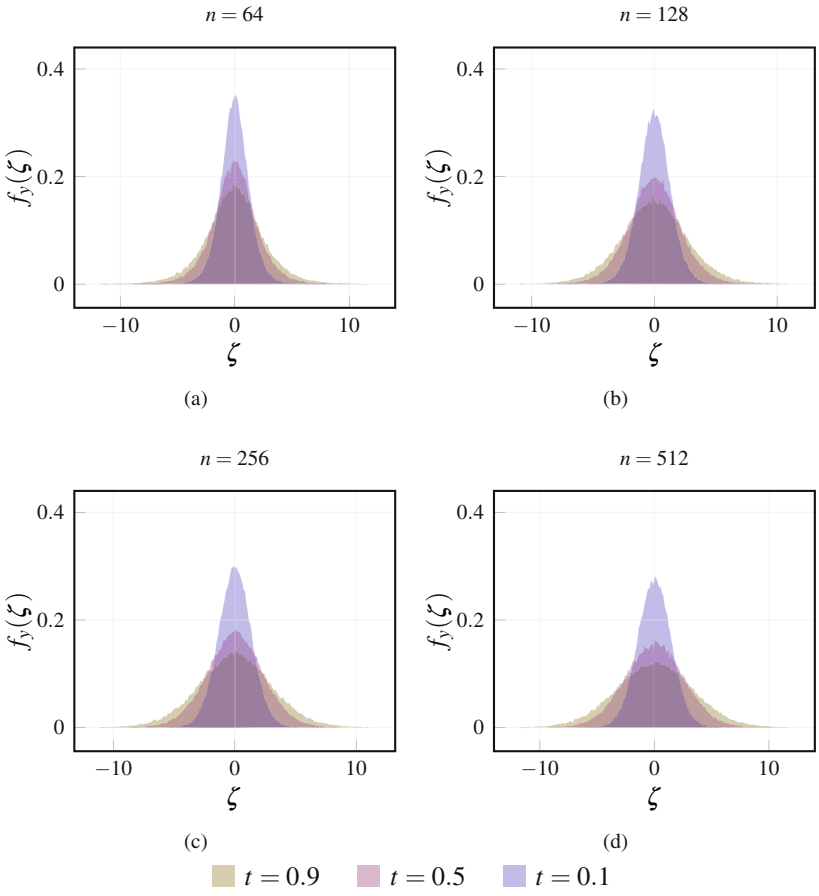


Fig. 3.12 The power of the measurements increases when t increases since the overall rakeness increases

This is beneficial since a number of theorems exist, analogous to the ones we report as Theorems 1.4 and 1.5, ensuring that the lower the coherence the easier for a BP or BPDN to retrieve the original signal.

As the construction of (possibly tight) equiangular frames is not an easy task, all the methods propose a heuristic to approximate their properties in a computationally feasible way. The result is a deterministic $\boldsymbol{A}$ that is linked to the features of the set of vectors $\boldsymbol{D}$ with respect to $\boldsymbol{x}$ which is sparse.

As an example, [4] and [7] pursue exactly this path and we will briefly present them within the framework proposed by [3] that also contain some limited improvement to those techniques.

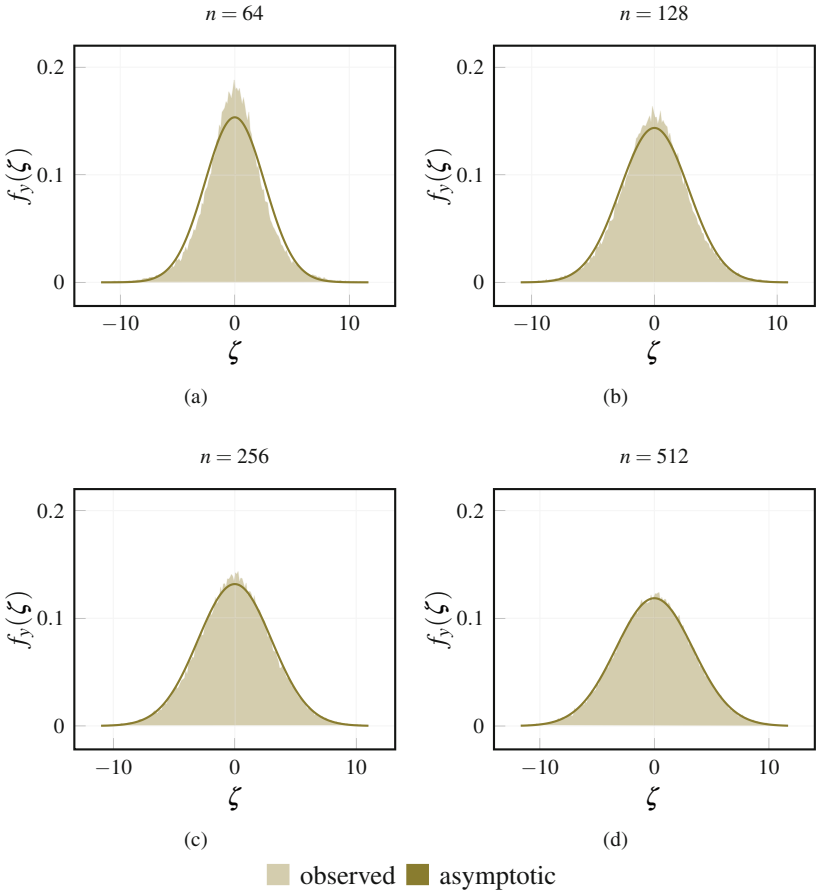


Fig. 3.13 As n increases the asymptotic Gaussian trends becomes an extremely accurate prediction of the true behavior of the measurements

From the definition in (1.10) we get that coherence has to do with the scalar products of columns of $\boldsymbol{B}$. By assuming that the columns are normalized to have unit norm, the cosines to be minimized are arranged as the entries of the $n \times n$ matrix $\boldsymbol{Z} = \boldsymbol{B}^\top \boldsymbol{B}$.

With this, optimizations like (1.15) become

$$\begin{aligned} \underset{\boldsymbol{Z} \in \mathbb{R}^{m \times n}}{\arg\min} \; & \|\boldsymbol{Z} - \boldsymbol{I}\|_\infty \\ \text{s.t.} \; & \boldsymbol{Z} \succeq 0 \\ & \boldsymbol{Z}^\top = \boldsymbol{Z} \\ & \boldsymbol{Z}_{j,j} = 1 \qquad 0 \leq j < n \\ & \operatorname{rank}(\boldsymbol{Z}) = m \end{aligned} \tag{3.13}$$

where $\boldsymbol{I}$ is the $n \times n$ identity matrix.

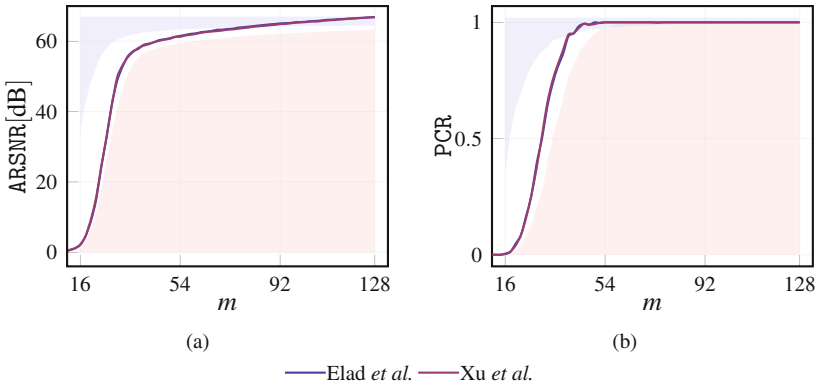


Fig. 3.14 Montecarlo assessment of performance due to coherence-based optimization. In the *light red region* performance is worse than conventional CS while in the *light blue*, performance is better than best possible rakeness-based CS

The objective function, jointly with the $\boldsymbol{Z}_{jj} = 1$ constraint, clearly aims at reducing the magnitude of the off-diagonal entries of $\boldsymbol{Z}$, i.e., the cosines to minimize. The rank and positive semidefiniteness constraints ensure that the matrix $\boldsymbol{Z}$ can be obtained as $\boldsymbol{Z} = \boldsymbol{B}^\top \boldsymbol{B}$ where $\boldsymbol{B}$ must be an $m \times n$ matrix.

From the solution of (3.13) one then infers $\boldsymbol{A} = \boldsymbol{Z}_{j,j}\boldsymbol{D}^\dagger$, where $\cdot^\dagger$ stands for the Moore–Penrose pseudo-inverse.

Intuitively speaking, the method by Elad et al. [4] and the one by Xu et al. [7] are two different heuristics addressing (3.13). Their performance is reported in Fig. 3.14 compared to classical CS and to rakeness-based design. The comparison highlights that, especially for what concerns the PCR guarantee, coherence-based design yields some improvement over classical CS. Most notably, when m is large, the two methods (whose difference is negligible, confirming the fact that they can be seen as two approaches to the same problem) perform very well. This is due to the fact that as m approaches n, the m degrees of freedom in each of the columns of $\boldsymbol{B}$ allow a very effective spreading of the corresponding vector in the n-dimensional space and $\boldsymbol{Z} \simeq \boldsymbol{I}$. Figure 3.14a shows that this phenomenon for $m \simeq n$ leads to a noteworthy denoising since the ISNR $= 60\,\text{dB}$ of the measured signal is significantly lower than the ARSNR of the reconstructed signal.

Yet, the plots also clarify that adaptivity to the sparsity dictionary $\boldsymbol{D}$ by coherence-based design is outperformed by rakeness-based design that considers not only $\boldsymbol{D}$ but also the second-order statistics of $\boldsymbol{x}$.

References

1. P. Billingsley, *Probability and Measure* (Wiley, New York, 2008)
2. V. Cambareri et al., A rakeness-based design flow for analog-to-information conversion by compressive sensing, in *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, IEEE, May 2013, pp. 1360–1363

3. N. Cleju, Optimized projections for compressed sensing via rank-constrained nearest correlation matrix. Appl. Comput. Harmon. Anal. **36**(3), 495–507 (2014)
4. M. Elad, Optimized projections for compressed sensing. IEEE Trans. Signal Process. **55**(12), 5695–5702 (2007)
5. R.A. Horn, C.R. Johnson, *Matrix Analysis* (Cambridge University Press, Cambridge, 2012)
6. M. Mangia, R. Rovatti, G. Setti, Rakeness in the design of analog-to-information conversion of sparse and localized signals. IEEE Trans. Circuits Syst. I Regul. Pap. **59**(5), 1001–1014 (2012)
7. J. Xu, Y. Pi, Z. Cao, Optimized projection matrix for compressive sensing. EURASIP J. Adv. Signal Process. **2010**(1), 560349 (2010)

Chapter 4
The Rakeness Problem with Implementation and Complexity Constraints

4.1 Complexity of CS

Though a mathematically grounded notion of complexity exists with the needed corollary of abstract results, what we here put under the umbrella of the generic term is similar to algorithmic time complexity and is whatever matters in quantifying the *operating costs* of an acquisition system based on CS.

The focus is on operating costs and, implicitly, on design costs since what we aim at is a design flow automatically administering the operating cost/performance trade-off. This is coherent with a framework in which the working-life of each of the acquisition systems we want to design dominates the implementation cost. In any case, implementation costs appear in our considerations as constraints to ensure that the overall design places itself in a neighborhood of what is *easy* to build.

To proceed we should detail the general architecture of the encoder side in Fig. 2.14 once CS enters the game as sketched in Fig. 1.2. In particular, we focus on the design of a *sensing node*, i.e., one of those small, possibly autonomous pieces of hardware among the key ingredients for the development of the future ubiquitous information processing systems that promise to be the implementation of grand concepts such as the *Internet of Things* and *cyber-physical systems*.

The *very big* picture appears quite often in nowadays technical presentations and papers and is exemplified in Fig. 4.1 with no claim of being either technically accurate or exhaustive. Sensing nodes can be deployed within different scenarios: Electro-Cardio-Gram, Electro-Encephalo-Gram, and sweat chemical composition can be used to monitor health, activity, and behavior of a human being; PH, temperature, and unusual sounds can be sensed in open field environments; traffic intensity, pollution, and wind may be of interest in urban settings; supply level, process consumption, and produced heat may be key quantities to observe in a production plant. Independently of the scenario, sensed data are transmitted through a network (most often but not necessarily wireless, possibly mesh-like and organized in more than one tier) up to an information hub in which conclusions are drawn and

M. Mangia et al., *Adapted Compressed Sensing for Effective Hardware Implementations*, DOI 10.1007/978-3-319-61373-4_4

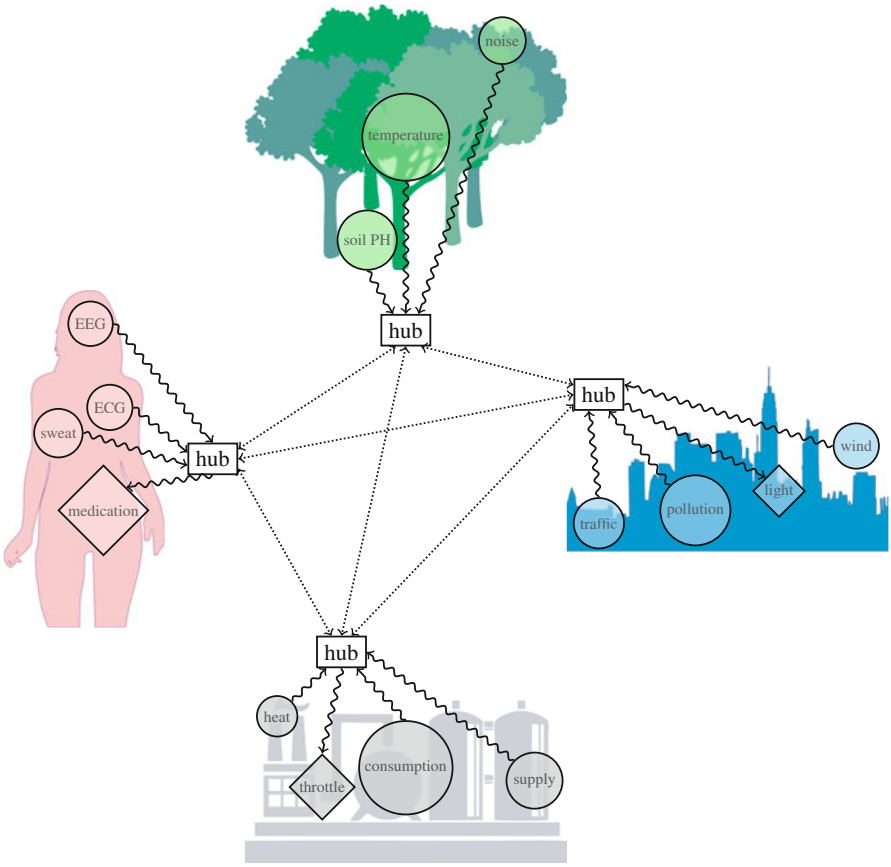


Fig. 4.1 The *very* big picture of the framework in which sensing nodes designed with CS may be employed

decision taken, depending on the scenario, on medication to apply, on timing of traffic lights, on pipe throttling in the production plant, etc. In principle, information hubs dedicated to a certain application may communicate with other information hubs to take decisions based on information coming from other networks of sensors.

The *circles* in Fig. 4.1 are the sensing nodes in which CS may have a role. Figure 4.2 shows a breakdown of each of the three stages (sampling, compression, dispatch) in one of the such nodes. The Analog Front End (AFE) inputs the external signal to a Sample-and-Hold (S/H) stage that makes the transition from continuous-time to discrete-time. The discrete-time nature of the samples is used in a series of Multiply-and-ACcumulate (MAC) loops in which the entries of $\boldsymbol{A}$ are the coefficient so that $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ is finally computed.

At least two options are available as far as measurement dispatch is concerned: transmit all the m scalars once they are computed or store them in a

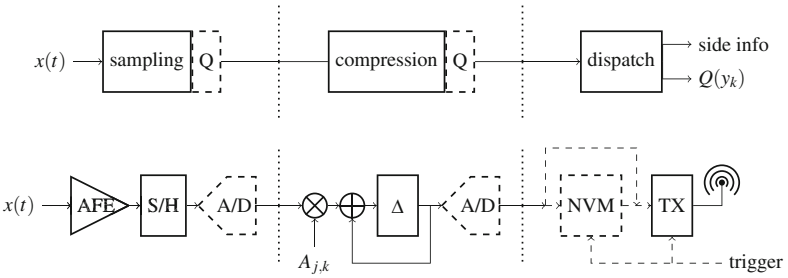


Fig. 4.2 The CS-based signal chain of a sensing node expanded into its main blocks

Non-Volatile-Memory (NVM) and wait until they can be transmitted to the hub. The second approach allows, for example, to postpone transmission until the encoder stage receives a proper trigger signal that may also transport the energy needed for data communication. Clearly, if complexity is associated with the amount of operations that are borne by the battery of the node, the two options imply a completely different weight of the dispatch stage.

Depending on where in Fig. 1.2 we put quantization, the Analog-to-Digital (A/D) converter may be after the S/H or after the application of CS immediately before dispatch.

With this breakdown, if we identify computation burden with consumed energy, the need of the three different stages depends on the features of the matrix $\boldsymbol{A}$ in $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$.

The AFE is active whenever a sample is to be provided to the subsequent stage. This is normally always true unless a whole column in $\boldsymbol{A}$ is made of zeros. If, on the contrary, we know that $\boldsymbol{A}_{\cdot,\bar{k}} = 0$ for a certain $\bar{k}$, then none of the sums

$$\boldsymbol{y}_j = \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k}\boldsymbol{x}_k = \sum_{\substack{k=0 \\ k\neq\bar{k}}}^{n-1} \boldsymbol{A}_{j,k}\boldsymbol{x}_k$$

uses the value of the $\bar{k}$-th sample that is always multiplied by zero. Assuming that the AFE can be switched off when not used, its computational burden can be made proportional to the number of nonzero columns in $\boldsymbol{A}$.

MAC operations must be performed only for $\boldsymbol{A}_{j,k} \neq 0$ but also in that case, the complexity of an individual MAC depends on the range of values that such a coefficient may assume. Two cases are particularly important: $\boldsymbol{A}_{j,k} \in \{-1, 0, +1\}$ (ternary-CS) and $\boldsymbol{A}_{j,k} \in \{0, 1\}$ (binary-CS). As schematized in Fig. 4.3a for a digital implementation, when $\boldsymbol{A}$ contains only ternary entries the MAC reduces to either sum, subtraction, or no update of the accumulator. The architecture can be further simplified as in Fig. 4.3b when $\boldsymbol{A}$ contains only binary entries. Analog implementations of MAC (those needed if the A/D stage is postponed immediately

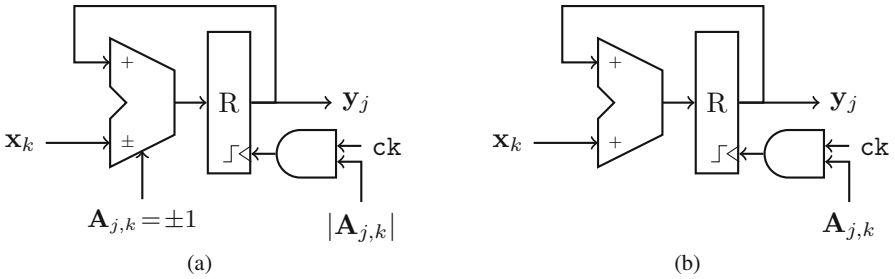


Fig. 4.3 The simplified MAC stages when $\boldsymbol{A}$ contains only ternary (**a**) or binary (**b**) entries

before storage) also clearly benefit from either ternary or binary constraint put on the elements of $\boldsymbol{A}$ and a thorough discussion of the corresponding architectures can be found in Chap. 6.

Independently of its analog or digital implementation, the computational burden of the stage that computes the measurements is proportional to the number of nonzero entries in $\boldsymbol{A}$. Yet, some special arrangements of these nonzero entries may be beneficial, for example, in implementations leveraging parallelism.

In fact, the matrix-by-vector product $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ can be unrolled either column-wise or row-wise depending on which of the two vectors $\boldsymbol{x}$ or $\boldsymbol{y}$ have its components simultaneously stored in the hardware. Once that the memory elements containing the measurements are reset to zero, the elementary updates

$$\boldsymbol{y}_j \leftarrow \boldsymbol{y}_j + \boldsymbol{A}_{j,k}\boldsymbol{x}_k$$

may be performed for each fixed k sweeping all $j = 0, \ldots, m-1$ (column-wise unrolling) or for each fixed j sweeping all $k = 0, \ldots, n-1$ (row-wise unrolling).

In analog implementations, it is quite typical to use column-wise unrolling and to perform all the updates due to the availability of a new sample in parallel. In this case, the number of nonzeros in each column of $\boldsymbol{A}$ is the number of updates that have to be performed in parallel.

The same sample-by-sample logic can underlay also digital implementations, either by means of custom-deployed logic or by means of a short software snippet that is triggered by the availability of a new sample. In this case the number of nonzeros in each column of $\boldsymbol{A}$ is proportional to the time needed by the execution of the code fragment.

When the system is designed to have the ADC in the leftmost position of Fig. 4.2, samples are available in digital form and can be buffered to have the whole vector $\boldsymbol{x}$ available for processing. This is commonly exploited to decouple acquisition and compression stages: two buffers are provided and while the acquisition fills a buffer with new samples, the compression stages operate on another buffer containing previously acquired samples. In this case the $\boldsymbol{A}\boldsymbol{x}$ product may be unrolled row-wise. What holds for column-wise unrolling can be repeated here in transposed form considering the number of nonzeros in each row of $\boldsymbol{A}$.

As far as the last stage is concerned, the storage or the transmission of the measurement vector has a cost surely proportional to the number m of the elements of $\boldsymbol{y}$.

Overall, the worst-case complexity of CS is $\mathsf{O}(n)$ for the sampling stage entailing the AFE, the S/H, and possibly the A/D. If is $\mathsf{O}(mn)$ for the stage computing the measurements and $\mathsf{O}(m)$ for storage and/or transmission. Starting from this worst-case scenario we may define a certain number of merit figures that measure our ability to reduce complexity.

The first and most obvious merit figure is the *compression ratio* (CR) defined as

$$\mathtt{CR} = \frac{n}{m}$$

so that the larger the compression ratio, the smaller the number of measurements that are necessary to reconstruct the signal. Other merit figures can be defined starting, for example, from the number $N \leq n$ of columns of $\boldsymbol{A}$ that contain at least a nonzero entry. With this

$$\mathtt{PR} = \frac{n}{N}$$

may be seen as a *puncturing ratio* since the original vector $\boldsymbol{x}$ may be *punctured* to drop unused samples and remain with $n\mathtt{PR}^{-1}$ useful components. The larger the puncturing ratio the smaller the number of samples that are actually involved in the computation of the measurements [1]. Figure 4.4 shows how CR and PR are related to the structure of the matrix $\boldsymbol{A}$.

The amount of computation needed to calculate all the measurements is clearly related to the total number W of the nonzero entries in $\boldsymbol{A}$ and can be quantified in relative terms defining the sparsity ratio of that matrix

$$\mathtt{SR} = \frac{nm}{W}$$

that is such that the larger the SR, the lower the computational burden.

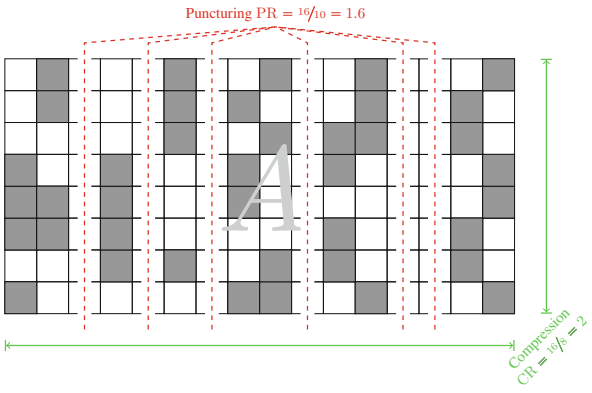


Fig. 4.4 Compression ratio and puncturing ratio from the structure of the matrix $\boldsymbol{A}$. *Gray boxes* correspond to nonzero entries

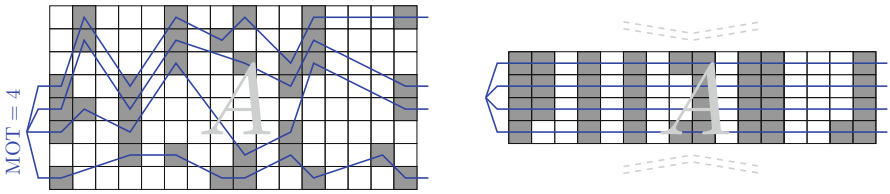


Fig. 4.5 The interpretation of output throttling in the column-wise unrolling of $\boldsymbol{Ax}$. The computational complexity is the same implied by a *vertically* throttled matrix. *Gray boxes* correspond to nonzero entries

Further to that, if M_j is the number of nonzero entries in the j-th column of $\boldsymbol{A}$, the quantity

$$\texttt{MOT} = \max_j \{M_j\}$$

can be seen as a an *maximum output throttling*. In fact, the complexity of computing $\boldsymbol{Ax}$ is equivalent to that of a product of $\boldsymbol{x}$ by a matrix that is throttled so that the height of its columns is MOT. Hence, at every sample step not more than MOT measurements need to be updated. Figure 4.5 gives an intuitive view of the output throttling.

Along the same path, if M_j is the number of nonzero entries in the j-th row of $\boldsymbol{A}$, the quantity

$$\texttt{MIT} = \max_j \{N_j\}$$

can be seen as a *maximum input throttling*. In fact, the complexity of computing $\boldsymbol{Ax}$ is equivalent to that of a product of $\boldsymbol{x}$ by a matrix that is horizontally throttled so that the width of its rows is at most MIT. Hence, every measurement needs at most the value of MIT out of the n available samples. Figure 4.6 gives an intuitive view of input throttling.

Overall, the merit figures we have defined allow us to give a general estimation of the complexity of each of the stages in Fig. 4.2 as follows:

1. the computational burden of the AFE, S/H, and possibly A/D stages is $\mathsf{O}\left(n\texttt{PR}^{-1}\right)$;
2. the computational burden of measurement computation is $\mathsf{O}\left(nm\texttt{SR}^{-1}\right)$;
3. the computational burden of storing and/or transmitting measurements is $\mathsf{O}\left(n\texttt{CR}^{-1}\right)$.

From the above estimations it is clear that the larger the merit figures, the lower the complexity of performing CS, though the impact of each stage on the overall complexity/cost depends on the details of the specific implementation. For example, if complexity is related to power consumption and transmission must be done in real-time and cannot be delayed to a moment in which it does not impact power budget, the last stage, and thus CR, will be the key player.

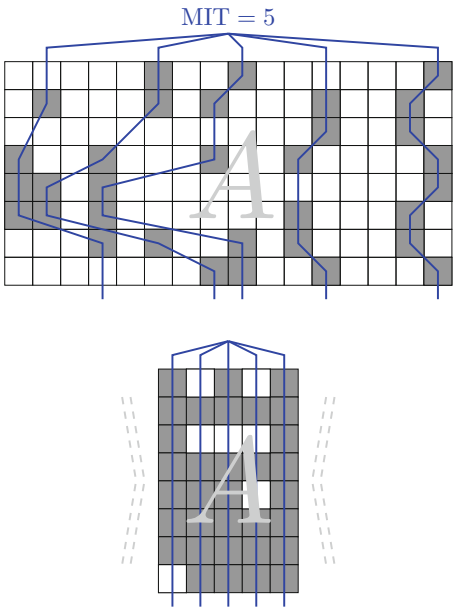


Fig. 4.6 The interpretation of input throttling in the column-wise unrolling of $\boldsymbol{Ax}$. The computational complexity is the same implied by a *horizontally* throttled matrix. *Gray boxes* correspond to nonzero entries

If, on the contrary, we are addressing a mostly analog implementation in which opamp-based processing is employed up to a final conversion that is used mainly to allow efficient storage of the results, the first stage will be the key part to be kept under control by means of PR.

As a third option, in a software implementation working on converted samples and competing for a time share of the micro with other simultaneous tasks, the matrix-by-vector product may be critical so that the key parameter is SR.

In this case, it may be useful to distinguish input and output throttling. In fact, the trivial double loop implementing $\boldsymbol{y} = \boldsymbol{Ax}$ can be unrolled in one of the two ways reported in Table 4.1. These correspond to the classical technique to store sparse matrices and operate with them. In particular, row-wise unrolling hinges on the possibility of knowing that, in the j-th row, there are not more than MIT nonzero entries whose positions can be stored as the indexes $k_0(j), k_1(j), \ldots, k_{\texttt{MIT}-1}(j)$. Conversely, column-wise unrolling hinges on the possibility of knowing that, in the k-th column, there are not more than MOT nonzero entries whose positions can be stored as the indexes $j_0(k), j_1(k), \ldots, j_{\texttt{MOT}-1}(k)$.

By looping on the stored indexes the computation burden of the inner loop is substantially reduced if $\texttt{MIT} \ll n$ or $\texttt{MOT} \ll m$, respectively.

The same considerations apply to possible all-digital implementation of CS. in this case, the most straightforward approach is to consider samples as they arrive from the analog-to-digital conversion chain as sketched in Fig. 4.7. If $\texttt{MOT} \ll m$, we may think of storing only the nonzero coefficients of each column and provide them to multipliers computing $\boldsymbol{A}_{j_l(k),k}\boldsymbol{x}_k$. A proper multiplexing logic is then in charge of accumulating the results of the multiplications into the registers corresponding to

Table 4.1 Two ways of unrolling the calculations in $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ to exploit the sparsity of $\boldsymbol{A}$

Row-wise unrolling	Column-wise unrolling
Assume $\boldsymbol{y}_j = 0$ for $j = 0, \ldots, m-1$ at initialization	

Require: $A_{j,k} = 0$ if $k \neq k_l(j)$ $\forall l$
for $j = 0, \ldots, m-1$ **do**
for $l = 0, \ldots,$ MIT $- 1$ **do**
$\boldsymbol{y}_j \leftarrow \boldsymbol{y}_j + \boldsymbol{A}_{j,k_l(j)}\boldsymbol{x}_k$
end for
end for
	Require: $\boldsymbol{A}_{j,k} = 0$ if $j \neq j_l(k)$ $\forall l$
	for $k = 0, \ldots, n-1$ **do**
	for $l = 0, \ldots,$ MOT $- 1$ **do**
	$\boldsymbol{y}_j \leftarrow \boldsymbol{y}_j + \boldsymbol{A}_{j_l(k),k}\boldsymbol{x}_k$
	end for
	end for

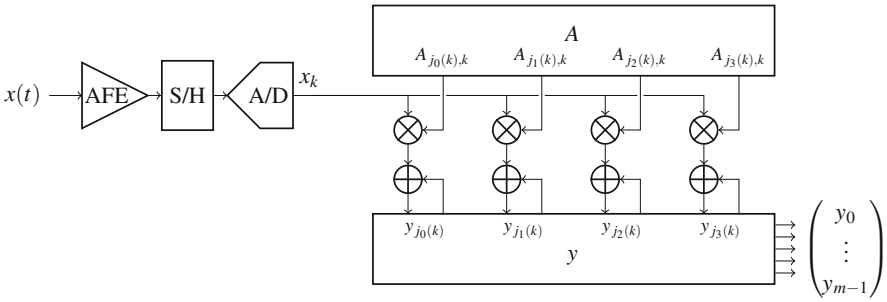


Fig. 4.7 A dedicated, all-digital implementation of CS with MOT $= 4$

$\boldsymbol{y}_{j_l(k)}$ for $l = 0, \ldots,$ MOT $- 1$ so that each measurement is updated only when the current sample affects it. Once all the samples have been processed, the registers contain the whole vector $\boldsymbol{y}$.

With this architecture, the number of multipliers and adders that must be deployed is only MOT instead of m.

As a dual option, one may think of storing the samples as they arrive from the conversion chain and make them available in parallel.

If the nonzero coefficients in every row of $\boldsymbol{A}$ are also stored, computation may proceed measure-by-measure. For each measure, the samples that affect and the nonzero coefficients in the corresponding row of $\boldsymbol{A}$ are retrieved to be presented at the inputs of the multipliers that may be summed to give the final value.

The number of multipliers reduces to MIT $\ll n$ while the final accumulation has an overall complexity surely smaller than that of MIT $- 1$ adders in a full MAC unit (Fig. 4.8).

The case of dedicated hardware implementations reveals that in some cases, once throttling is fixed, there is no point in using a matrix $\boldsymbol{A}$ with even less nonzero entries. In fact, for example, an implementation relying on row-wise unrolling to exploit the fact that MOT $\ll m$ deploys hardware resources depending on such a maximum number of nonzeros per column. The presence of even less nonzeros brings little benefit to resource saving while it further limits the number of times

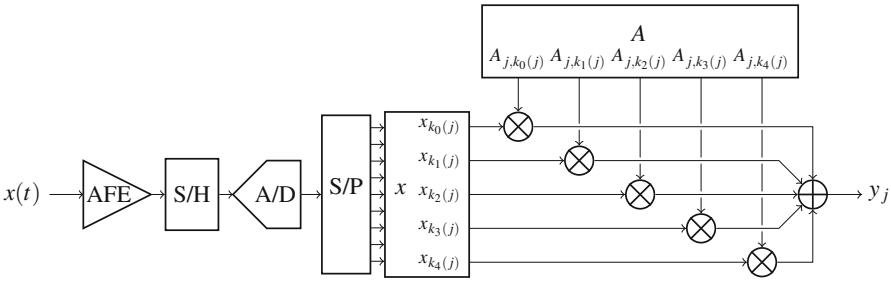


Fig. 4.8 A dedicated, all-digital implementation of CS with $\texttt{MIT} = 5$

in which the signal enters the computation of the measurements, thus potentially reducing the information content of the latters.

This is why it may be sensible to consider matrices $\boldsymbol{A}$ in which $M_0 = M_1 = \cdots = M_{n-1} = \texttt{OT}$, i.e., with an output throttling setting exactly the number of nonzeros in each column. With this, once that n and $\texttt{OT}$ are set, we know that $\boldsymbol{A}$ has $n\texttt{OT}$ nonzero entries.

If one considers column-wise unrolling to exploit the fact that $\texttt{MIT} \ll n$, it may be sensible to consider matrixes $\boldsymbol{A}$ in which $N_0 = N_1 = \cdots = N_{m-1} = \texttt{IT}$, i.e., with an input throttling setting exactly the number of nonzeros in each row. With this, once that n and $\texttt{IT}$ are set, we know that $\boldsymbol{A}$ has $m\texttt{IT}$ nonzero entries.

4.2 Rakeness and Zeroing

An ideal design flow would start from some application-related information on the relative weight of the different stages to identify the key parameter ($\texttt{CR}$, $\texttt{PR}$, $\texttt{SR}$) that should be increased to reduce operating costs. Then, it would choose a sensing matrix $\boldsymbol{A}$ that maximizes such a parameter while allowing a reconstruction of the original signal satisfying some minimum quality requirement.

There is an intuitive trade-off between any of the above parameters and the quality achievable in signal reconstruction.

The effect of $\texttt{CR}$ and $\texttt{PR}$ is clear (see Fig. 4.4): the higher the $\texttt{CR}$, the lower the number of scalars with which the same signal is encoded, while the higher the $\texttt{PR}$, the lower the number of samples that are used to generate the final measurements.

The effect of $\texttt{SR}$ is slightly more subtle. We may see it column-wise and observe that since there are $W = nm\texttt{SR}^{-1}$ nonzero entries in $\boldsymbol{A}$ and n columns, each column contains on the average $m\texttt{SR}^{-1}$ nonzero entries. This means that each sample in $\boldsymbol{x}$ affects on the average only $m\texttt{SR}^{-1}$ measurements. Dually, from $W = nm\texttt{SR}^{-1}$ and from the fact that there are m rows, we have that, on the average, every measurement in $\boldsymbol{y}$ depends only on $n\texttt{SR}^{-1}$ samples.

Whatever our point of view, the higher one of our parameters, the lower the chances that we have to look at the signal and extract information. It is then sensible to expect that an increase in any of such parameters corresponds to a decrease in reconstruction quality.

Regrettably, it is difficult to address such a trade-off by means of a straightforward optimization problem of the kind *"maximize saving constrained to maintain a minimum reconstruction quality."* The reason is twofold. First, we have no link but the intuitive rakeness criterion between the features of $\boldsymbol{A}$ and the reconstruction quality, and this prevents the formal definition of the "maintain a minimum reconstruction quality"constraint. Second, the rakeness-based design flow is itself a maximization problem addressing a trade-off, i.e., the one between the advantage of focusing on more energetic directions and the necessity of exploring the signal space to capture all the signal features.

To cope with this, one may reverse the point of view and retain the structure of the rakeness-localization trade-off problem in (3.5) and inject saving as further constraints.

The first implicit constraint is that either $\boldsymbol{A}_{j,k} \in \{-1, 0, +1\}$ or $\boldsymbol{A}_{j,k} \in \{0, 1\}$. This clearly impacts the kind of correlation matrices $\mathscr{A}$ that we may expect, i.e., on the feasibility space of our optimization problem.

To understand why, we may focus on the elementary case with $n = 2$ and $\boldsymbol{A}_{j,k} \in \{0, 1\}$ in which the only possible rows of $\boldsymbol{A}$ are the four elements of $\{0, 1\}^2$. If $\boldsymbol{a} \in \{0, 1\}^2$ and $p_{\boldsymbol{a}}$ is the probability that such a row appears in $\boldsymbol{A}$ then

$$\mathscr{A} = \mathbf{E}\left[\boldsymbol{a}\boldsymbol{a}^\top\right] = \sum_{\boldsymbol{a}\in\{0,1\}^2} p_{\boldsymbol{a}}\boldsymbol{a}\boldsymbol{a}^\top \tag{4.1}$$

Since $\sum_{\boldsymbol{a}\in\{0,1\}^2} p_{\boldsymbol{a}} = 1$ and $p_{\boldsymbol{a}} \geq 0$, $\mathscr{A}$ belongs to the convex hull of the matrices corresponding to $\boldsymbol{a}\boldsymbol{a}^\top$ for $\boldsymbol{a} \in \{0, 1\}^2$, i.e., of the four binary matrices

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

This point of view will be developed and exploited in Chap. 5. By now, we may use it to check that

$$\mathscr{A} = \begin{pmatrix} 1/2 & 1/2 \\ & \\ 1/2 & 1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

is a correlation matrix that can be obtained by a binary process that yields $\boldsymbol{a} = (1, 1)$ with probability $1/2$ and $\boldsymbol{a} = (0, 1)$ with probability $1/2$. On the contrary

$$\mathscr{A} = \begin{pmatrix} 1/2 & 1/4 \\ & \\ 1/4 & 1 \end{pmatrix} = \frac{1}{4}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \frac{1}{4}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \frac{3}{4}\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

is a correlation matrix (it is symmetric and positive definite since its eigenvalues are $1/4\left(3 \pm \sqrt{2}\right) > 0$) but cannot be produced by a binary process since the three coefficients are not probabilities due to the fact that $1/4 + 1/4 + 3/4 > 1$.

In this trivial case the conditions for (4.1) to hold can be derived by solving

$$\begin{cases} p_{(0,0)} \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} + p_{(1,0)} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + p_{(0,1)} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + p_{(1,1)} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \mathscr{A} \\ \\ p_{(0,0)} + p_{(1,0)} + p_{(0,1)} + p_{(1,1)} = 1 \end{cases} \tag{4.2}$$

whose solution

$$\begin{aligned} p_{(0,0)} &= 1 - \mathscr{A}_{0,0} - \mathscr{A}_{1,1} + \mathscr{A}_{0,1} \\ p_{(0,1)} &= \mathscr{A}_{1,1} - \mathscr{A}_{0,1} \\ p_{(1,0)} &= \mathscr{A}_{0,0} - \mathscr{A}_{0,1} \\ p_{(1,1)} &= \mathscr{A}_{0,1} \end{aligned}$$

must feature only nonnegative values. Hence, for $\mathscr{A}$ to be an 2×2 correlation matrix of a binary random vector, the entries of $\mathscr{A}$ must be such that

$$\max\{0, 1 - \mathscr{A}_{0,0} - \mathscr{A}_{1,1}\} \leq \mathscr{A}_{1,0} \leq \min\{\mathscr{A}_{0,0}, \mathscr{A}_{1,1}\} \tag{4.3}$$

Regrettably, for higher dimensionality, this straightforward path cannot be followed. In fact, the generic $\mathscr{A}_{0,0}$ is symmetric and has $n(n+1)/2$ degrees of freedom, so that (4.2) is made of $n(n+1)/2 + 1$ equations for the 2^n unknowns $p_{\boldsymbol{a}}$. Hence, its solution is not unique and one should ascertain whether at least one of the solutions is made of all nonnegative components. The complexity of such a task prevents the formulation of a constraint guaranteeing that $\mathscr{A}$ can be obtained by a binary process. To cope with this, we formulate only a relaxed constraint.

In particular we note that for $\mathscr{A}$ to be the $n \times n$ correlation matrix of a binary process it is necessary that every submatrix $\begin{pmatrix} \mathscr{A}_{j,j} & \mathscr{A}_{j,k} \\ \mathscr{A}_{k,j} & \mathscr{A}_{k,k} \end{pmatrix}$ for $0 \leq j < k < n$ is the correlation matrix of a 2-dimensional binary vector of the kind analyzed above. Hence we will require

$$\left(1 - \mathscr{A}_{j,j} - \mathscr{A}_{k,k}\right)^+ \leq \mathscr{A}_{j,k} \leq \min\left\{\mathscr{A}_{j,j}, \mathscr{A}_{k,k}\right\} \qquad 0 \leq j < k < n \tag{4.4}$$

where $(\cdot)^+ = \max\{0, \cdot\}$.

Clearly, these constraints are necessary but not sufficient to guarantee that the resulting $\mathscr{A}$ can be obtained from a binary process. This is tackled in the following chapter where we design generators of binary and ternary processes with a correlation *as close as possible* to a given one. We will see that such an implicit approximation step in the resulting design flow does not significantly impair the final performance gain.

The same path can be followed to give a necessary conditions for $\mathscr{A}$ to be a correlation matrix of a ternary process, i.e., when $\boldsymbol{A}_{j,k} \in \{-1, 0, +1\}^n$.

In this case the 2×2 case yields an expansion in 3^2 terms and the calculations are less straightforward. Yet, the results can be intuitively justified quite easily. In fact, if $\boldsymbol{a}$ is a row of $\boldsymbol{A}$ and $\mathscr{A} = \mathbf{E}\left[\boldsymbol{a}\boldsymbol{a}^\top\right]$, then $\mathscr{A}_{j,j} = \mathbf{E}[\boldsymbol{a}_j^2] = \Pr\left\{\boldsymbol{a}_j \neq 0\right\} \leq 1$.

Moreover, $\left|\mathscr{A}_{j,k}\right| = \left|\mathbf{E}\left[\boldsymbol{a}_j\boldsymbol{a}_k\right]\right| \leq \min\left\{\Pr\left\{\boldsymbol{a}_j \neq 0\right\}, \Pr\left\{\boldsymbol{a}_k \neq 0\right\}\right\}$. In fact, two ternary random variables are maximally positively (negatively) correlated when the chance that they have the same (opposite) sign is maximized, a chance that cannot exceed the probability that each of them is nonzero, i.e., the smallest of the probabilities of being nonzero. Hence, it must be

$$\left|\mathscr{A}_{j,k}\right| \leq \min\left\{\mathscr{A}_{j,j}, \mathscr{A}_{k,k}\right\} \qquad 0 \leq j < k < n \tag{4.5}$$

The binary and ternary cases have in common that $\mathscr{A}_{j,j} = \mathbf{E}\left[\boldsymbol{a}_j^2\right] = \Pr\left\{\boldsymbol{a}_j \neq 0\right\}$ and thus that setting $\mathscr{A}_{j,j} = \eta_j$ defines a parameter η_j that allows to control the average number of nonzeros at position j in $\boldsymbol{a}$. If one sets $\eta_j = 0$, then $\boldsymbol{a}_j = 0$. In the ternary case, setting $\eta_j = 1$ implies that $\boldsymbol{a}_j$ is no longer ternary but antipodal $\boldsymbol{a}_j \in \{-1, +1\}$.

This control on the zeros of the matrix $\boldsymbol{A}$ is the key to obtain savings on the running costs along the guidelines described above. To do so, the rakeness-based design flow described in Chap. 3 can be adjusted to include proper additional constraints.

In particular, in the ternary case we may solve

$$\begin{aligned}
\underset{\mathscr{A} \in \mathbb{R}^{n \times n}}{\operatorname{argmax}} \ & \operatorname{tr}(\mathscr{A}\mathscr{X}) \\
& \mathscr{A} \succeq 0 \\
& \mathscr{A} = \mathscr{A}^\top \\
\text{s.t.} \ & \mathscr{L}_{\boldsymbol{a}} \leq \tau^2 \mathscr{L}_{\boldsymbol{x}} \\
& \mathscr{A}_{j,j} = \eta_j \qquad\qquad 0 \leq j < n \\
& \left|\mathscr{A}_{j,k}\right| \leq \eta_j \qquad 0 \leq j \neq k < n
\end{aligned} \tag{4.6}$$

where τ^2 is used to parameterize the localization constraint with respect to the localization of the input signal, and the two last constraints interact with the symmetry of $\mathscr{A}$, i.e., with the fact that $\mathscr{A}_{j,k} = \mathscr{A}_{k,j}$, to ensure (4.5).

In the binary case, the same problem becomes

$$\begin{aligned}
\underset{\mathscr{A} \in \mathbb{R}^{n \times n}}{\operatorname{argmax}} \ & \operatorname{tr}(\mathscr{A}\mathscr{X}) \\
& \mathscr{A} \succeq 0 \\
& \mathscr{A} = \mathscr{A}^\top \\
& \mathscr{L}_{\boldsymbol{a}} \leq \tau^2 \mathscr{L}_{\boldsymbol{x}} \\
\text{s.t.} \ & \mathscr{A}_{j,j} = \eta_j \qquad\qquad 0 \leq j < n \\
& \mathscr{A}_{j,k} \leq \eta_j \qquad\qquad 0 \leq j \neq k < n \\
& \mathscr{A}_{j,k} \geq \left(\eta_j + \eta_k - 1\right)^+ \qquad 0 \leq j \neq k < n
\end{aligned} \tag{4.7}$$

4.3 Solving TRLT and BRLT by Projected Gradient and Alternating Projections

The two problems in (4.6) and (4.7) cannot be solved analytically due to the additional constraints we put to cut out some of the points that do not correspond to correlation matrices of ternary or binary vectors.

In fact, these constraints complicate the shape of the resulting feasibility space. As an example, consider a 3×3 correlation matrix $\mathscr{A}$ in a problems that sets $\mathscr{A}_{j,j} = \eta_j$, for $j = 0, 1, 2$. Due to symmetry the available degrees of freedom are only $\mathscr{A}_{0,1}$, $\mathscr{A}_{0,2}$, and $\mathscr{A}_{1,2}$.

The positive semidefinite constraint can be translated into inequalities involving the degrees of freedom by means of the Sylvester's criterion yielding

$$\begin{aligned} \mathscr{A}_{0,1}^2 &\leq \eta_0\eta_1 \\ \eta_2\mathscr{A}_{0,1}^2 + \eta_1\mathscr{A}_{0,2}^2 + \eta_0\mathscr{A}_{1,2}^2 - 2\mathscr{A}_{0,1}\mathscr{A}_{0,2}\mathscr{A}_{1,2} &\leq \eta_0\eta_1\eta_2 \end{aligned} \tag{4.8}$$

In the same space of degrees of freedom, the inequalities due to the ternary constraint are

$$\begin{aligned} |\mathscr{A}_{0,1}| &\leq \min\{\eta_0, \eta_1\} \\ |\mathscr{A}_{0,2}| &\leq \min\{\eta_0, \eta_2\} \\ |\mathscr{A}_{1,2}| &\leq \min\{\eta_1, \eta_2\} \end{aligned} \tag{4.9}$$

while the localization constraint becomes

$$\mathscr{A}_{0,1}^2+\mathscr{A}_{0,2}^2+\mathscr{A}_{1,2}^2 \leq \frac{1}{2}\left(\frac{1}{3} + \tau^2\mathscr{L}_x\right)(\eta_0 + \eta_1 + \eta_2)^2 - \frac{\eta_0^2 + \eta_1^2 + \eta_2^2}{2} \tag{4.10}$$

Figure 4.9c exemplifies the structure of the feasibility space for values of the parameters chosen to show all of its features, namely $\eta_0 = 7/10$, $\eta_1 = 3/5$, $\eta_2 = 3/10$, and $\tau^2\mathscr{L}_x = 21/100$. In these conditions, the inequalities in (4.8) are satisfied within the region shown in Fig. 4.9a, those in (4.9) are satisfied within the region shown in Fig. 4.9b, and the inequality in (4.10) is satisfied within the region shown in Fig. 4.9c. Overall, the feasibility space of (4.6) is the intersection of all the above and has the shape reported in Fig. 4.9d.

In the binary case (4.9) is substituted by the much stricter

$$\begin{aligned} (\eta_0 + \eta_1 - 1)^+ &\leq \mathscr{A}_{0,1} \leq \min\{\eta_0, \eta_1\} \\ (\eta_0 + \eta_2 - 1)^+ &\leq \mathscr{A}_{0,2} \leq \min\{\eta_0, \eta_2\} \\ (\eta_1 + \eta_2 - 1)^+ &\leq \mathscr{A}_{1,2} \leq \min\{\eta_1, \eta_2\} \end{aligned} \tag{4.11}$$

Adopting the same parameters as above, the feasibility space of (4.7) shrinks to the shape reported in Fig. 4.10.

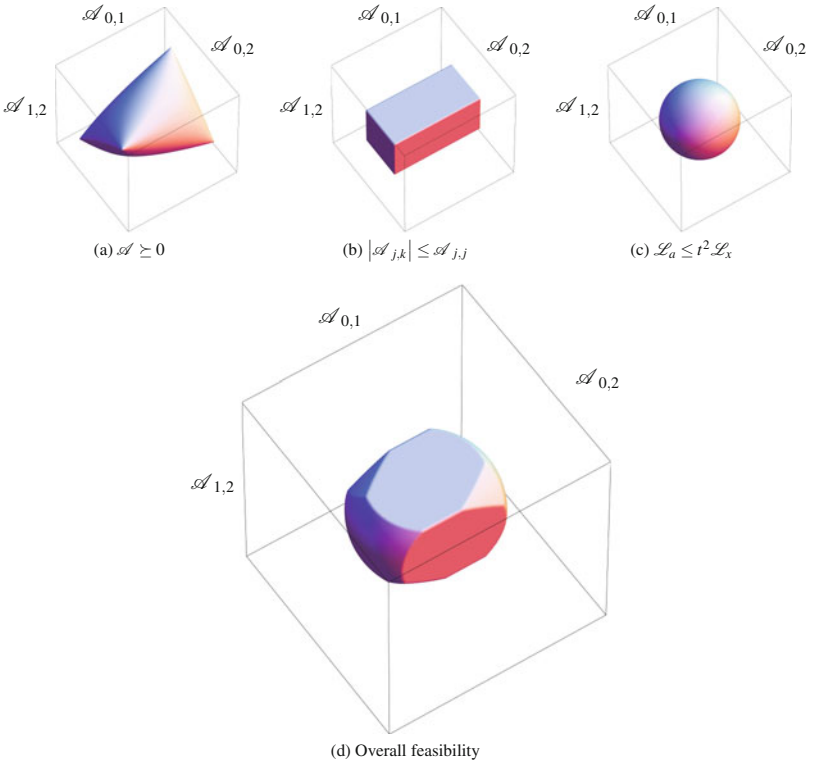


Fig. 4.9 The feasibility space of (4.6) as the intersection of the different constraints (the range along every axis is from $-7/10$ to $7/10$)

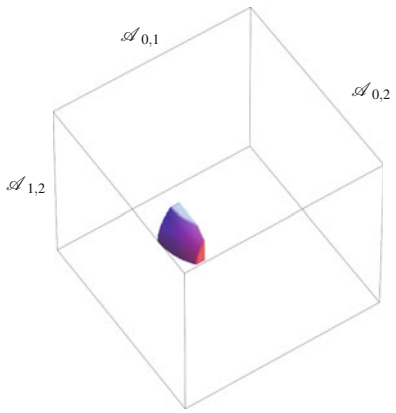


Fig. 4.10 The feasibility space of (4.7) (the range along every axis is from $-7/10$ to $7/10$)

Figures 4.9d and 4.10 reveal that the shape of the feasibility space can be very complicated. Yet, it is not difficult to accept that, independently of n, the set of admissible solutions remains a convex subset of $\mathbb{R}^{n(n-1)/2}$. Since the merit function $\mathrm{tr}\,(\mathscr{A}\mathscr{X})$ is linear in $\mathscr{A}$, both (4.6) and (4.7) are convex programming problems.

To tackle them in a general and quite scalable way allowing values of n in the hundreds, we may resort to the *projected gradient method* whose theoretical background is summarized by the following Theorem [3].

Theorem 4.1 *For a certain dimensionality p, let $c : \mathbb{R}^p \mapsto \mathbb{R}$ a convex cost function and $C \subset \mathbb{R}^p$ a convex feasibility set defining the optimization problem*

$$\min_{\boldsymbol{\xi}} c(\boldsymbol{\xi}) \quad \text{s.t.} \quad \boldsymbol{\xi} \in C$$

whose solution if $c\,(\xi^*)$ for some $\xi^* \in C$.

Let the projection-on-C operator π_C be defined as

$$\pi_C(\boldsymbol{\xi}) = arg\ min_{\boldsymbol{\zeta}} \|\boldsymbol{\zeta} - \boldsymbol{\xi}\|_2 \quad \text{s.t.} \quad \boldsymbol{\zeta} \in C \tag{4.12}$$

Starting from any $\boldsymbol{\xi}^{(0)} \in C$ define

$$\boldsymbol{\xi}^{(t+1)} = \pi_C\left(\boldsymbol{\xi}^{(t)} - \alpha^{(t)} \nabla_{\boldsymbol{\xi}} c\left(\boldsymbol{\xi}^{(t)}\right)\right) \tag{4.13}$$

for some coefficients $\alpha^{(t)} > 0$, $t = 0, 1, \ldots$ whose sequence is such that $\sum_{t=0}^{\infty} \alpha^{(t)} = \infty$ but $\sum_{t=0}^{\infty} \left(\alpha^{(t)}\right)^2 < \infty$.

If $\max_{0 \le t < T} \left\| \nabla_{\boldsymbol{\xi}} c\left(\boldsymbol{\xi}^{(t)}\right) \right\|_2 < \infty$, then

$$\lim_{T \to \infty} c\left(\boldsymbol{\xi}^*\right) - \min_{0 \le t < T} \left\{ c\left(\boldsymbol{\xi}^{(t)}\right) \right\} = 0$$

In less formal terms, (4.13) allows to move from one candidate solution to the next in two phases. First, one makes a step along the direction of the gradient of the cost function to decrease its value. Second, since this possibly yields a point out of the feasibility space, the projection operator is used to find its closest admissible approximation. Convergence is guaranteed if the lengths of the steps are not vanishing too fast and gradients are bounded, something that happens in our case since $\nabla_{\mathscr{A}} \mathrm{tr}\,(\mathscr{A}\mathscr{X}) = \mathscr{X}$ is constant.

Clearly, the critical point here is the computation of π_C whose complexity depends on the structure of C. In particular, C is typically given as the intersection of a certain number q of simpler convex subsets $C_0, C_1, \ldots, C_{q-1}$, i.e., $C = \bigcap_{j=0}^{q-1} C_j$. The C_j are simpler in the sense that the corresponding projection operators π_{C_j} are known and easy to compute.

A procedure leading from the π_{C_j} to π_C is better explained for $q = 2$. In that case, let $\boldsymbol{\zeta}^{(t,0)} = \boldsymbol{\xi}^{(t)} - \alpha^{(t)} \nabla_{\boldsymbol{\xi}} c\left(\boldsymbol{\xi}^{(t)}\right)$ be the point that is reached at the t-th step by following the gradient direction, and assume that $\boldsymbol{\zeta}^{(t,0)} \notin C$. Since $C = C_0 \cap C_1$ we may set

$$\begin{aligned}\boldsymbol{\zeta}^{(t,2s+1)} &= \pi_{C_0}\left(\boldsymbol{\zeta}^{(t,2s)}\right)\\ \boldsymbol{\zeta}^{(t,2s+2)} &= \pi_{C_1}\left(\boldsymbol{\zeta}^{(t,2s+1)}\right)\end{aligned}$$

for $s = 0, 1, \ldots$. If the sequence converges we may set $\boldsymbol{\xi}^{(t+1)} = \boldsymbol{\zeta}^{(t,\infty)} = \lim_{s\to\infty} \boldsymbol{\zeta}^{(t,s)}$ since it is sure that the limit belongs both to C_0 and C_1. This was first proposed for the case in which C_0 and C_1 are subspaces [5] since this implies not only that $\boldsymbol{\zeta}^{(t,\infty)} \in C_0 \cap C_1$ but also that $\boldsymbol{\zeta}^{(t,\infty)} = \pi_{C_0 \cap C_1}\left(\boldsymbol{\zeta}^{(t,0)}\right)$.

Regrettably, this is not necessarily true when we are dealing with more complicated convex sets C_j. Figure 4.11a shows this with an example. In that case C_0 is a disk and C_1 is a half-plane. Starting from $\boldsymbol{\zeta}^{(t,0)}$ the first two projections are enough to produce a point in $C_0 \cap C_1$ that is unchanged by further projections and thus is the limit of the sequence. Regrettably, such a limit is not the true projection.

To cope with the general case, we have to modify the algorithm as described in [2]. Formally speaking we may use two auxiliary offset sequences $\Delta\boldsymbol{\zeta}_0^{(s)}$ and $\Delta\boldsymbol{\zeta}_1^{(s)}$, initialized with $\Delta\boldsymbol{\zeta}_0^{(s)} = \Delta\boldsymbol{\zeta}_1^{(s)} = 0$, to write

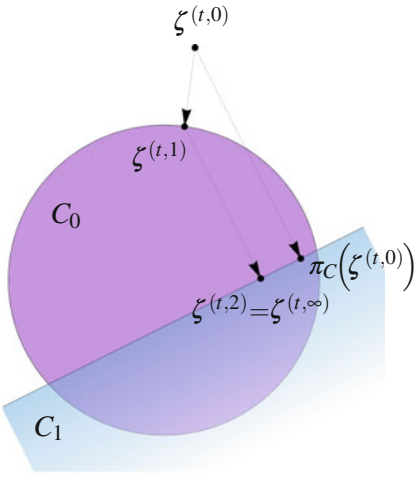
$$\begin{aligned}\boldsymbol{\zeta}^{(t,2s+1)} &= \pi_{C_0}\left(\boldsymbol{\zeta}^{(t,2s)} - \Delta\boldsymbol{\zeta}_0^{(s)}\right)\\ \Delta\boldsymbol{\zeta}_0^{(s+1)} &= \boldsymbol{\zeta}^{(t,2s+1)} - \boldsymbol{\zeta}^{(t,2s)} - \Delta\boldsymbol{\zeta}_0^{(s)}\\ \boldsymbol{\zeta}^{(t,2s+2)} &= \pi_{C_1}\left(\boldsymbol{\zeta}^{(t,2s+1)} - \Delta\boldsymbol{\zeta}_1^{(s)}\right)\\ \Delta\boldsymbol{\zeta}_1^{(s+1)} &= \boldsymbol{\zeta}^{(t,2s+2)} - \boldsymbol{\zeta}^{(t,2s+1)} - \Delta\boldsymbol{\zeta}_1^{(s)}\end{aligned}$$

In words, before applying π_{C_j}, we subtract to its argument the offset caused by the previous application of the same projection to "cancel" its effect that may have caused the sequence to stick to a point in $C_0 \cap C_1$ that is not the projection. The result of this "cancellation" is exemplified in Fig. 4.11b.

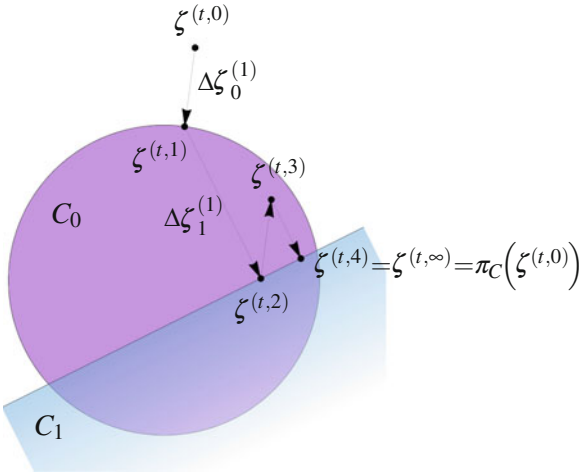
The above formulation can be easily generalized to $q > 2$ convex components C_j and ensures that $\lim_{s\to\infty} \boldsymbol{\zeta}^{(t,s)} = \pi_{\bigcap_{j=0}^{q-1} C_j}\left(\boldsymbol{\xi}^{(t)}\right)$.

The only missing ingredient in a recipe to solve (4.6) and (4.7) is the application to our specific case, i.e., the identification of the C_j and the explicit formulation of the corresponding π_{C_j}.

To do so, it is convenient to recall (see Chap. 3) that, for $n \times n$ symmetric matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$, the operator tr $(\boldsymbol{PQ})$ is a scalar product that induces the Frobenius norm,



(a)



(b)

Fig. 4.11 The alternating projection method (**a**) and its variant ensuring the convergence to the true projection (**b**)

that is the $\|\cdot\|_2$ norm of the collection of entries of the matrix, as well as the $\|\cdot\|_2$ norms of the collection of eigenvalues of the matrix, i.e.,

$$\mathrm{tr}\left(\boldsymbol{P}^2\right) = \|\boldsymbol{P}\|_2^2 = \sum_{j=0}^{n-1}\sum_{k=0}^{n-1}\boldsymbol{P}_{j,k}^2 = \sum_{j=0}^{n-1}\lambda_j^2$$

if λ_j is the j-th eigenvalue of $\boldsymbol{P}$. This allows to specialize the concept of projection (4.12) to our matrix space.

To begin with, say that C_0 is the set of points corresponding to symmetric and positive-semidefinite matrices, i.e., the one represented in Fig. 4.9a for the $n = 3$ case. Thanks to the spectral interpretation of the Frobenius norm, we may write that for any symmetric matrix $\boldsymbol{P}$ spectrally decomposed as $\boldsymbol{P} = \boldsymbol{U}\Lambda\boldsymbol{U}^\top$ with $\boldsymbol{U}$ orthogonal and $\Lambda = \text{diag}\,(\lambda_0, \ldots, \lambda_{n-1})$ we have

$$\pi_{C_0}(\boldsymbol{P}) = \boldsymbol{U} \max\{0, \Lambda\}\, \boldsymbol{U}^\top$$

Then we may assume that C_1' is the set of symmetric matrices that satisfy the ternary constraints (the one represented in Fig. 4.9b) while C_1'' is the set of symmetric matrices that satisfy the binary constraints. In the ternary case we may define the two matrices

$$\overline{\mathscr{A}}_{j,k} = \begin{cases} \eta_j & \text{if } j = k \\ \min\{\eta_j, \eta_k\} & \text{if } j \neq k \end{cases}$$

and

$$\underline{\mathscr{A}'}_{j,k} = \begin{cases} \eta_j & \text{if } j = k \\ -\min\{\eta_j, \eta_k\} & \text{if } j \neq k \end{cases}$$

to write

$$\pi_{C_1'}(\boldsymbol{P}) = \max\left\{\underline{\mathscr{A}'}, \min\left\{\overline{\mathscr{A}}, \boldsymbol{P}\right\}\right\}$$

In the binary case the lower bound must be redefined as

$$\underline{\mathscr{A}''}_{j,k} = \begin{cases} \eta_j & \text{if } j = k \\ \max\{0, \eta_j + \eta_k - 1\} & \text{if } j \neq k \end{cases}$$

to yield

$$\pi_{C_1''}(\boldsymbol{P}) = \max\left\{\underline{\mathscr{A}''}, \min\left\{\overline{\mathscr{A}}, \boldsymbol{P}\right\}\right\}$$

Finally, say that C_2 is the set of symmetric matrices obeying the localization constraint $\mathscr{L}_a \leq \tau^2 \mathscr{L}_x$. Starting from the definition of localization we may write

$$\text{tr}\left(\mathscr{A}^2\right) \leq \text{tr}^2\left(\mathscr{A}\right)\left(\frac{1}{n} + \tau^2 \mathscr{L}_x\right)$$

and thus

$$\|\mathscr{A}\|_2^2 \leq \left(\sum_{j=0}^{n-1} \eta_j\right)^2 \left(\frac{1}{n} + \tau^2 \mathscr{L}_x\right) = R^2$$

where R^2 remains implicitly defined. Hence, C_2 is a sphere of radius R and the corresponding projection operator is a simple scaling

$$\pi_{C_2}(\boldsymbol{P}) = \boldsymbol{P} \min\left\{1, \frac{R}{\|\boldsymbol{P}\|_2}\right\}$$

4.4 Unstructured and Structured Zeroing

When no implementation-related consideration puts constraints on the positions of the zeros in $\boldsymbol{A}$, the aim is simply to reduce the number of (signed)sums by imposing that the matrix is sparse with a sparsity controlled by $\mathtt{SR}$. This can be obtained by setting $\eta_j = \mathtt{SR}^{-1}$ for every j in (4.6) and (4.7).

Though this unstructured design does not allow to leverage the particular nonzero configurations that underlay puncturing and throttling it produces some resource saving. Clearly, such a saving comes at the expense of reconstruction quality. The resulting trade-off can be explored by simulation.

In the following we consider the framework defined in Chap. 2 focusing on signals that are $\kappa = 6$-sparse, with a medium localization (ML) and low-pass spectrum (LP). For the ternary case, we consider solutions $\mathscr{A}(\mathtt{SR})$ of (4.6) with $t = 1/2$ and for $\mathtt{SR} = 1, 2, 4, 8, 16, 32$, where $\mathtt{SR} = 1$ stands for a full matrix and $\mathtt{SR} = 32$ stands for a matrix that, on the average, has only 1 nonzero in 32 entries. We then use $\boldsymbol{A} \sim \mathrm{RTE}(\mathscr{A}(\mathtt{SR}))$ and simulate reconstruction performance. As reference cases, we also consider the reconstruction performance of $\boldsymbol{A} \sim \mathrm{RGE}(iid)$ (the most conventional CS choice), and $\boldsymbol{A} \sim \mathrm{RGE}(\mathscr{A})$ with $\mathscr{A}$ being the solution of (3.5).

Figure 4.12 shows the corresponding performance in terms of $\mathtt{ARSNR}$ and $\mathtt{PCR}$. By looking at the $\mathtt{SR} = 1$ curves, note that for rakeness-based design, passing from Gaussian entries to antipodal entries in $\boldsymbol{A}$ does not cause any performance degradation.

This is a well-known property that allows to avoid the generation and storage of many-bits samples of Gaussian random variables whenever a parsimonious implementation is sought [4].

Moreover, curves are practically invariant up to $\mathtt{SR} = 8$, i.e., when up to 87.5% of the entries in $\boldsymbol{A}$ are zeros. When $\mathtt{SR} = 16$ the performance of a ternary $\boldsymbol{A}$ with 93.75% of entries equal to zero is substantially equivalent to the performance of classical CS with $\boldsymbol{A} \sim \mathrm{RGE}(iid)$. Clearly, performance degradation eventually becomes an issue for even larger sparsity ratios as clarified by the $\mathtt{SR} = 32$ curves that correspond to 96.9% of zero entries in $\boldsymbol{A}$ but exhibit worse-than-conventional performance. In any case, since $\mathtt{SR}$ directly maps on resource saving, it is evident that rakeness-based design of antipodal sensing matrices has a huge potential in reducing CS working costs.

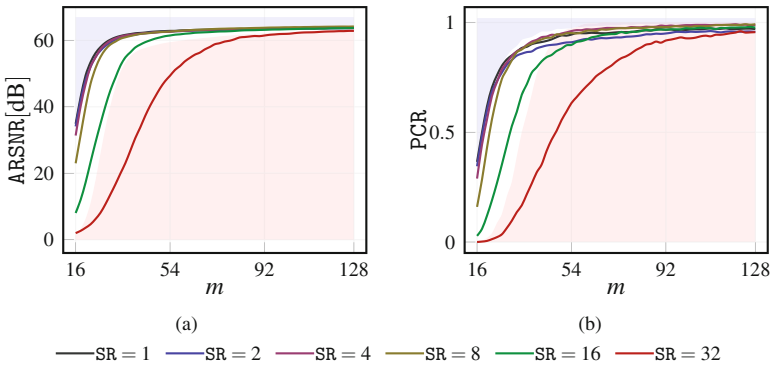


Fig. 4.12 Montecarlo comparison between performance of rakeness-based ternary CS with different sparsity ratios SR. In the *light red region* performance is worse than conventional CS while in the *light blue region*, performance is better than best possible rakeness-based CS

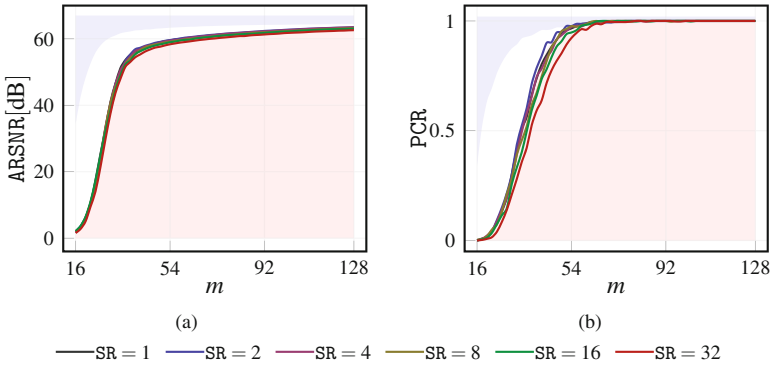


Fig. 4.13 Montecarlo comparison between performance of random ternary CS with different sparsity ratios SR. In the *light red region* performance is worse than conventional CS while in the *light blue region*, performance is better than best possible rakeness-based CS

To confirm that performance improvement over classical CS is due to rakeness-based CS, Fig. 4.13 reports the same cases as in Fig. 4.12 when the nonzero entries are taken as independent, zero mean ± 1.

Comparing the two figures, we get that the purely random choice of the value of the nonzeros is more robust to sparsification of $\boldsymbol{A}$ (curves are practically indistinguishable up to $\mathtt{SR} = 32$), though it is clearly unable to yield any of the advantages due to adaptation to the incoming signal.

This performance difference descends from a deep difference in the matrices $\boldsymbol{A}$ in the rakeness-based and conventional cases. Such a difference can be appreciated in Fig. 4.14. Despite the fact that the two matrices have the same number of nonzeros, the rakeness-based $\boldsymbol{A}$ adapts to the low-pass nature of the signal to align its nonzeros in low-pass (constant) runs that increase the amount of energy passed to the resulting measurement.

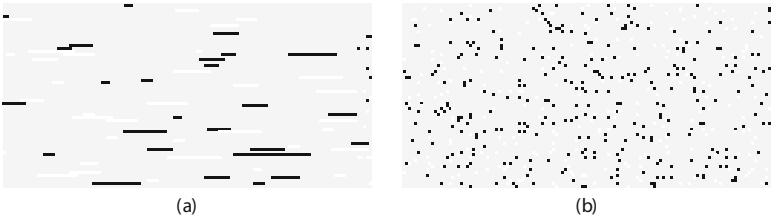


Fig. 4.14 Two typical unstructured projection matrices $\boldsymbol{A}$ with $\mathtt{SR} = 16$ for rakeness-based CS (**a**) and for conventional CS (**b**). *Gray zones* correspond to zeros while *black/white dots* mark $-1/+1$ entries

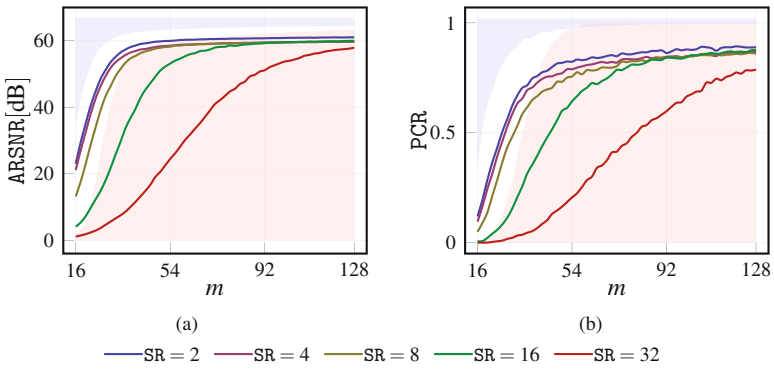


Fig. 4.15 Montecarlo comparison between performance of rakeness-based binary CS with different sparsity ratios SR. In the *light red region* performance is worse than conventional CS while in the *light blue*, performance is better than best possible rakeness-based CS

The binary, rakeness-based case is slightly different since the nonzeros are actually 1 and their positions completely identify the whole matrix. As a first consequence, there is not a $\mathtt{SR} = 1$ case that would yield a useless constant $\boldsymbol{A}$. Furthermore, results for those are reported in Fig. 4.15 where performance is assed in the same way as for ternary case.

It is evident that, despite the fact that binary matrices give the maximum possible simplification, their performance is significantly reduced with respect to what rakeness-based design can offer in general. For $\mathtt{SR} = 4$, ARSNR is analogous to what can be obtained by classical CS. Yet, when performance must be guaranteed in the form of a certain PCR, binary is always a poorer choice with respect to the classical option.

Such a poor performance is partially due to the fact that, if no check is made on the number of nonzeros in the rows of $\boldsymbol{A}$, the average number of nonzeros $\mathtt{SR}^{-1}$ may be met if some rows have a large majority of zeros (and thus collect information from a very small number of samples) while others are almost full (that, in the binary case, makes little distinction between samples as all the nonzero coefficients are equal to 1).

4.4.1 Puncturing

Puncturing is the simplest way of adding a useful structure to $\boldsymbol{A}$. Once that $\mathtt{PR}$ is set, we may randomly select the $m\mathtt{PR}^{-1}$ columns that are not forced to be null and keep their indexes in the index set $K \subseteq \{0, 1, \ldots, n-1\}$. Then, we restrict the signal correlation matrix to the corresponding time instants to obtain $\mathscr{X}_{|K}$ and aim at optimize the correlation matrix of the sensing rows of $\boldsymbol{A}$ focusing only on the entries that, in each row, are not forced to be zero, i.e., computing the best restricted $\mathscr{A}_{|K}$ by means of the restricted versions of (4.6) and (4.7).

In general, this is not the same as solving (4.6) and (4.7) for the full $\mathscr{A}$ and then dropping the entries that correspond to columns that are zeroed by puncturing. This is due to the constraint $\mathscr{L}_{\boldsymbol{a}} \leq t^2 \mathscr{L}_x$ that mixes the entries of $\mathscr{A}$ into those of $\mathscr{A}^2$ in the expression of $\mathscr{L}_a$. Yet, if $\mathscr{X}$ is Töplitz (i.e., the process generating the samples x_k is a stationary one), we set $\eta_j = \mathtt{SR}^{-1}$ independently of j, n is large, and the number of zeroed columns remains limited, then samples tend to be indistinguishable and one may think of solving the largest problem and subsampling the full solution $\mathscr{A}$ to obtain $\mathscr{A}_{|K}$.

To assess the impact of puncturing, we refer to the simulation setting described above and obtain performances when a variable number of column is zeroed in $\boldsymbol{A}$. Figure 4.16 shows the result.

For $\mathtt{PR} \simeq 1.5$, 1 in 3 columns is zeroed and thus one third of the samples is neglected and needs not to be acquired and converted. In this case, performance is practically indistinguishable from what is achieved by a full $\boldsymbol{A}$ with no constraint on the entries. Improvements over classical CS are present up to $\mathtt{PR} \simeq 2$, a case 50% of the columns are zeroed and thus 50% of the samples can be neglected, while with a little degradation with respect to classical CS one may adopt $\mathtt{PR} = 2.5$, i.e., discard 60% of the samples.

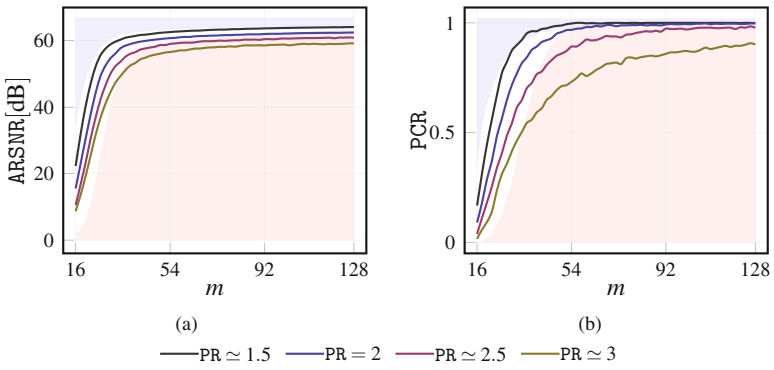


Fig. 4.16 Montecarlo comparison between performance of rakeness-based ternary CS with different puncturing ratios $\mathtt{PR}$. In the *light red region* performance is worse than conventional CS while in the *light blue region*, performance is better than best possible rakeness-based CS

In Chap. 6, we will see that such a noteworthy robustness to puncturing is due the fact that the support of the vectors in the sparsity basis (the orthonormal Discrete Cosine Transform basis as defined in Chap. 2) extends over the whole domain, so that skipping samples does not risk to miss any of them. It is here interesting to see that, when such an intrinsic robustness is present, rakeness-based design allows to effectively exploit it to save resources.

4.4.2 Input Throttling

When one considers throttling, things tend to complicate slightly and input throttling must be distinguished from output throttling.

If $\mathtt{IT}$ is set by implementation requirements, we know that $n - \mathtt{IT}$ entries will be zero in each row of $\boldsymbol{A}$. If only one row at a time is considered, this is equivalent to puncturing. Hence, if generation proceeds row by row, the above method is an option and one may randomly select the indexes of the nonzero components, collect them in the set K, and determine $\mathscr{A}_{|K}$ to generate that row. Each row has its own K and thus $\mathscr{A}_{|K}$.

As an alternative method, one may set $\eta_j = {}^{\mathtt{IT}}/_{n}$ so that, if the resulting $\mathscr{A}$ is used, only $\mathtt{IT}$ entries in the generated vector are nonzero on average. Multiple candidate rows can be generated until one contains exactly $\mathtt{IT}$ nonzeros and can be accepted. This sieving method has the advantage that the position of the zeros are implicitly effected by the statistic of the signal, something that does not happen with the first method that decides K a priori. As a drawback, sieving may be time consuming and can be an option only if $\boldsymbol{A}$ is generated off-line.

This is actually the method that we use to assess the performance that can be attained by input-throttled matrices produced by rakeness-based design. We perform Montecarlo simulations in the same conditions as above for different values of $\mathtt{IT}$ thus setting the number of nonzeros in each row of $\boldsymbol{A}$. Since $n = 128$, each value of $\mathtt{IT}$ implies a sparsity ratio $\mathtt{SR} = {}^{128}/_{\mathtt{IT}}$.

Figures 4.17 and 4.18 show the performances of ternary and binary rakeness-based CS when this kind of structuring is adopted for $\boldsymbol{A}$.

By comparing them with Figs. 4.12 and 4.15 one immediately realizes that input-throttling not only eases implementation but also improves performance. This is particularly true when $\mathtt{SR}$ is large and thus $\mathtt{IT}$ is small.

In that case, in fact, imposing only an average number of nonzeros by means of $\eta_j = \mathtt{SR}^{-1}$ would yield a non-negligible probability of generating rows with a insufficient number of entries (if $\mathtt{SR} = 32$ and thus $\eta = 0.03125$, it is quite common to have rows with 0 or 1 nonzeros) that produce almost useless measurements. When $\mathtt{IT}$ fixes the number of nonzeros in each row, rakeness-based design is able to administer them to capture the most significant feature of the signal. Such a phenomenon is more evident when $\mathtt{IT}$ is small.

As a result, ternary rakeness-based CS is able to yield the same performance given by unconstrained rakeness-based CS with full $\boldsymbol{A}$ while avoiding 96.9% of the

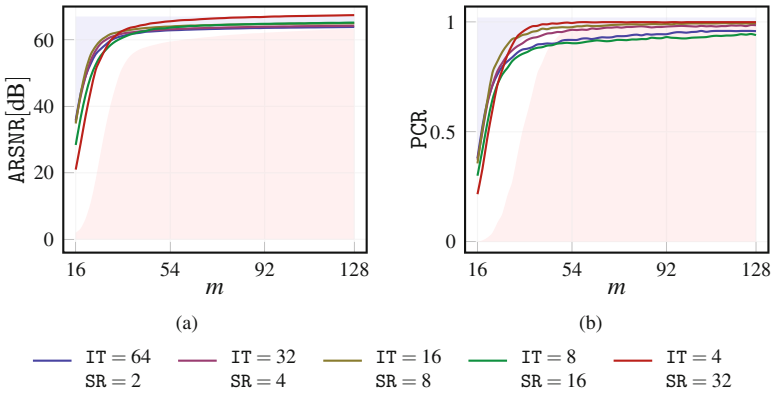


Fig. 4.17 Montecarlo comparison between performance of rakeness-based ternary CS with different input throttlings `IT` and subsequent sparsity ratios `SR`. In the *light red region* performance is worse than conventional CS while in the *light blue region*, performance is better than best possible rakeness-based CS

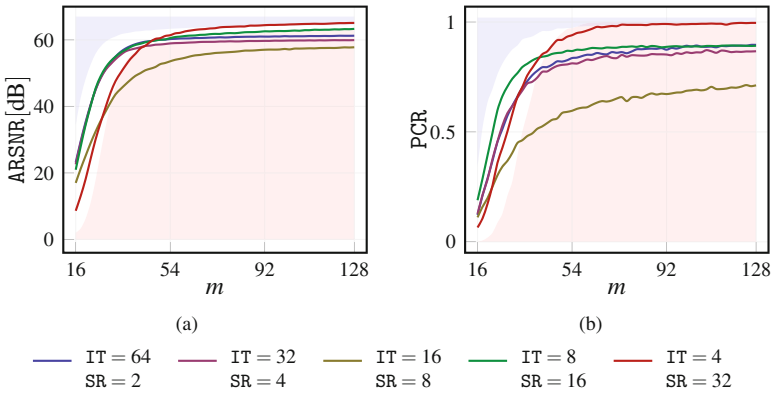


Fig. 4.18 Montecarlo comparison between performance of rakeness-based binary CS with different input throttlings `IT` and subsequent sparsity ratios `SR`. In the *light red region* performance is worse than conventional CS while in the *light blue region*, performance is better than best possible rakeness-based CS

MACs and performing, in our case, only $\texttt{IT} = 4$ signed sums per measure. Such an astounding result is somehow mimicked by binary rakeness-based CS that, with the same computational effort, reproduces the performance of classical unconstrained CS with full $\boldsymbol{A}$.

For such an extreme $\texttt{IT} = 4$, Fig. 4.19 shows what happens when we add puncturing to the design of ternary rakeness-based CS. Clearly, performance is reduced but the proper management of the nonzeros due to rakeness-based design allows to deliver the same performance as conventional unconstrained CS with full

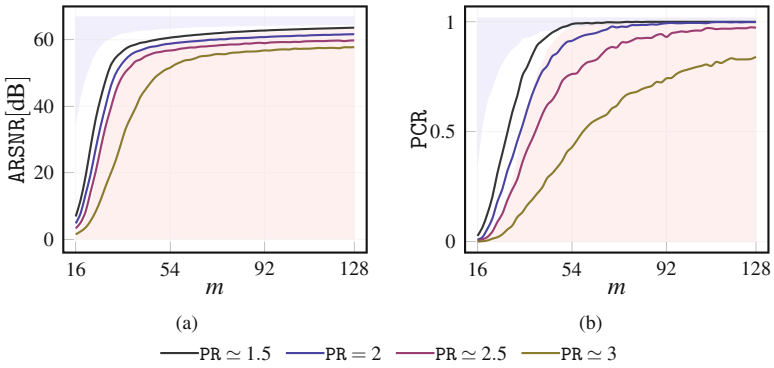


Fig. 4.19 Montecarlo comparison between performance of rakeness-based ternary CS with different puncturing ratios `PR` and input throttling `IT` $= 4$. In the *light red region* performance is worse than conventional CS while in the *light blue region*, performance is better than best possible rakeness-based CS

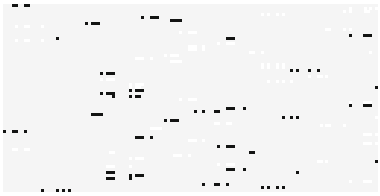


Fig. 4.20 A typical projection matrix with `SR` $= 2$ and `IT` $= 4$. *Gray zones* correspond to zeros while *black/white dots* mark $-1/+1$ entries

$\boldsymbol{A}$, not only computing only 4 signed sums per measurement, but also skipping 50% of the samples. The strong structure of the resulting projection matrix is exemplified in Fig. 4.20 in which the constant runs along rows are often broken by the zeroed columns.

4.4.3 Output Throttling

If `OT` is set by implementation requirements, the number of nonzero is known for each column. This does not pair seamlessly with the fact that rakeness-based design assumes independent rows in $\boldsymbol{A}$.

The most straightforward way of coping with this is to pre-define the nonzero pattern of the whole matrix and then slice it row-by-row. Hence, for each column we decide the positions of the `OT` nonzero elements and collect this information for the whole matrix. Then we proceed row by row by inferring K, computing $\mathscr{A}_{|K}$ and generating the proper sensing row $\boldsymbol{a}$.

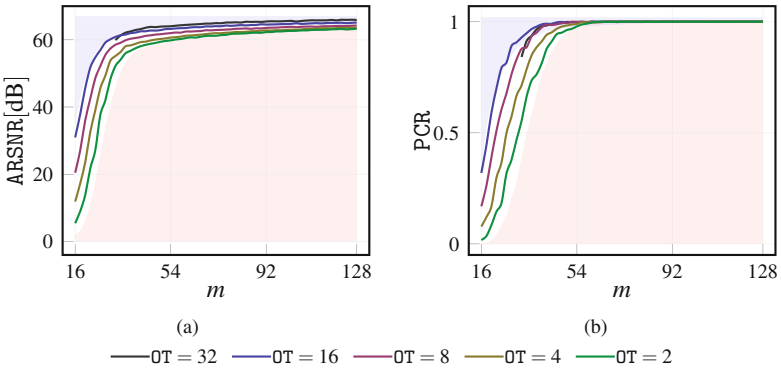


Fig. 4.21 Montecarlo comparison between performance of rakeness-based ternary CS with different output throttlings OT. In the *light red region* performance is worse than conventional CS while in the *light blue region*, performance is better than best possible rakeness-based CS

The resulting performance is reported in Fig. 4.21 for the ternary case. Performance degrades gracefully as IT decreases and remains not worse than classical CS even for IT $= 2$, i.e., when only two signed sums are performed at every new sample x_k.

References

1. D. Bellasi et al., A low-power architecture for punctured compressed sensing and estimation in wireless sensor-nodes. IEEE Trans. Circuits Syst. I Regul. Pap. **62**(5), 1296–1305 (2015)
2. J.P. Boyle, R.L. Dykstra, A method for finding projections onto the intersection of convex sets in Hilbert spaces, in *Advances in Order Restricted Statistical Inference*, ed. by R.L. Dykstra, T. Robertson, F.T. Wright. Proceedings of the Symposium on Order Restricted Statistical Inference Held in Iowa City Iowa, September 11–13, 1985 (Springer New York, New York, 1986), pp. 28–47
3. A.A. Goldstein, Convex programming in Hilbert space. Bull. Am. Math. Soc. **70**, 709–710 (1964)
4. J. Haboba et al., A pragmatic look at some compressive sensing architectures with saturation and quantization. IEEE J. Emerging Sel. Top. Circuits Syst. **2**(3), 443–459 (2012)
5. J. Von Neumann, On rings of operators. Reduction theory. Ann. Math. **50**(2), 401–485 (1949)

Chapter 5
Generating Raking Matrices: A Fascinating Second-Order Problem

As discussed in the previous chapters, sensing matrices characterizing the encoder stage are in general the row by row composition of random sequences with a proper second-order statistic characterization. A step towards a practical implementation imposes that each symbol of such matrices belongs to a finite set of values whose cardinality L can be limited to either 3 or 2. A trivial approach is to obtain these symbols by quantizing real quantities with a resulting perturbation on the imposed second-order statistics. Obviously, the introduced perturbation is strongly related to L and in particular the corner cases of either binary, antipodal, or ternary sequences feature the largest distortion. This limitation can drastically reduce the impact of the proposed sensing matrix design methods on the entire performance of the system. To overcome this impasse, we list here techniques aiming at generating sequences of binary, antipodal, or ternary symbols with an assigned second-order statistical characterization. The adoption of such techniques aims to reduce the impact of L on the entire system performance as much as possible.

5.1 Signal Modeling and Definitions

As a general guideline for this entire chapter, we refer to a stochastic process capable of generating random vectors $\boldsymbol{v} = (\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n-1})^\top \in \mathbb{R}^n$. These vectors are zero-mean, i.e.,

$$\mathbf{E}_{\boldsymbol{v}}[\boldsymbol{v}] = \mathbf{0}$$

while, in the more general case, the second-order statistic is given by means of the correlation matrix $\boldsymbol{\mathcal{V}} = \mathbf{E}_{\boldsymbol{v}}[\boldsymbol{v}\boldsymbol{v}^\top] \in \mathbb{R}^{n\times n}$.

We will first introduce a simple stationary stochastic process, and then a more general non-stationary case. We briefly recall that a stationary stochastic process

M. Mangia et al., *Adapted Compressed Sensing for Effective Hardware Implementations*, DOI 10.1007/978-3-319-61373-4_5

generates sequences whose statistical features are independent of the position in the sequence. As a consequence, the element $\mathcal{V}_{j,k}$ of the correlation matrix depends only on $|j-k|$.

In the stationary case, we will consider a reference case where $\boldsymbol{v}$ is generated by a stochastic process whose second-order statistic is described by the correlation matrix

$$\mathcal{V}_{j,k} = \omega^{|j-k|}$$

with $\omega \in]-1, 1[$. In this case, it is also possible to define a power spectrum $\Psi_{\boldsymbol{v}}$, that is given by

$$\Psi_{\boldsymbol{v}}(f) = \frac{1-\omega^2}{1+\omega^2-2\omega\cos(2\pi f)}$$

The shape of this process depends on the sign of ω. For positive values the process exhibits a low-pass profile, while negative values generate high-pass profiles. When $\omega = 0$, a simple flat/white profile (i.e., $\Psi_{\boldsymbol{v}}(f) = 1$) is achieved.

In the non-stationary case, we limit ourselves to consider the reference case where the correlation matrix $\mathcal{V}^{\text{ns}}$ is given by

$$\mathcal{V}^{\text{ns}}_{j,k} = \begin{cases} 1 & \text{if } j = k \\ \omega^{|j-k|+\frac{|j+k-n|}{16}} & \text{if } j \neq k \end{cases}$$

for some ω values in the range $]0, 1[$. Note that $\mathcal{V}^{\text{ns}}$ is not anymore a toeplitz matrix, but it is still symmetric positive semidefinite, i.e., its eigenvalues are non-negative.

5.2 Quantized Gaussian Sequences

We propose here a first trivial solution given by the direct quantization of real sensing matrices, i.e., we map each real value in a discrete one by a dictionary of L possible values. This solution can be effectively implemented in a physical device, and has been used in [2] as discussed in Sect. 7.4.

As an example, let us focus on the stationary case, and generate the sequences $\boldsymbol{v} \sim \text{RGE}(\mathcal{V})$ whose correlation matrix (evaluated as discussed both in Chaps. 3 and 4) is given by $\mathcal{V}$. Then, let us consider their quantized version $\boldsymbol{v}' = Q(\boldsymbol{v}, L)$ as shown in Fig. 5.1. Due to the quantization operation, we expect that the correlation matrix of $\boldsymbol{v}'$, given by $\mathcal{V}'_L = \mathbf{E}[\boldsymbol{v}'\boldsymbol{v}'^{\top}]$, is different from $\mathcal{V}$. Furthermore, we also expect that this difference is small for high value of L, while when L is low a non-negligible perturbation arises affecting the statistic of $\boldsymbol{v}'$ with respect to the ideal case represented by $\boldsymbol{v}$.

The approach can be mapped in two separate stages, as in Fig. 5.2.

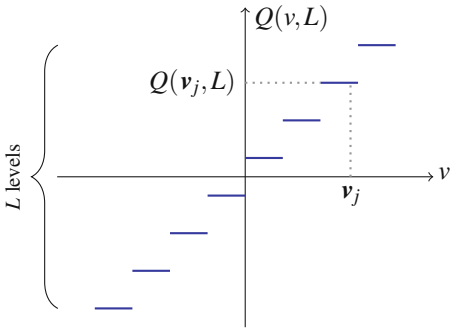


Fig. 5.1 Function performing quantization

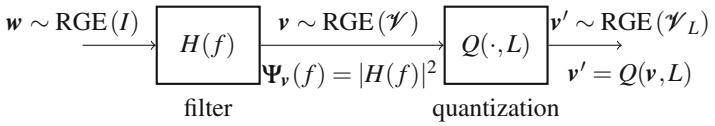


Fig. 5.2 A traditional path for the generation of quantized Gaussian sequences with a proper second-order characterization

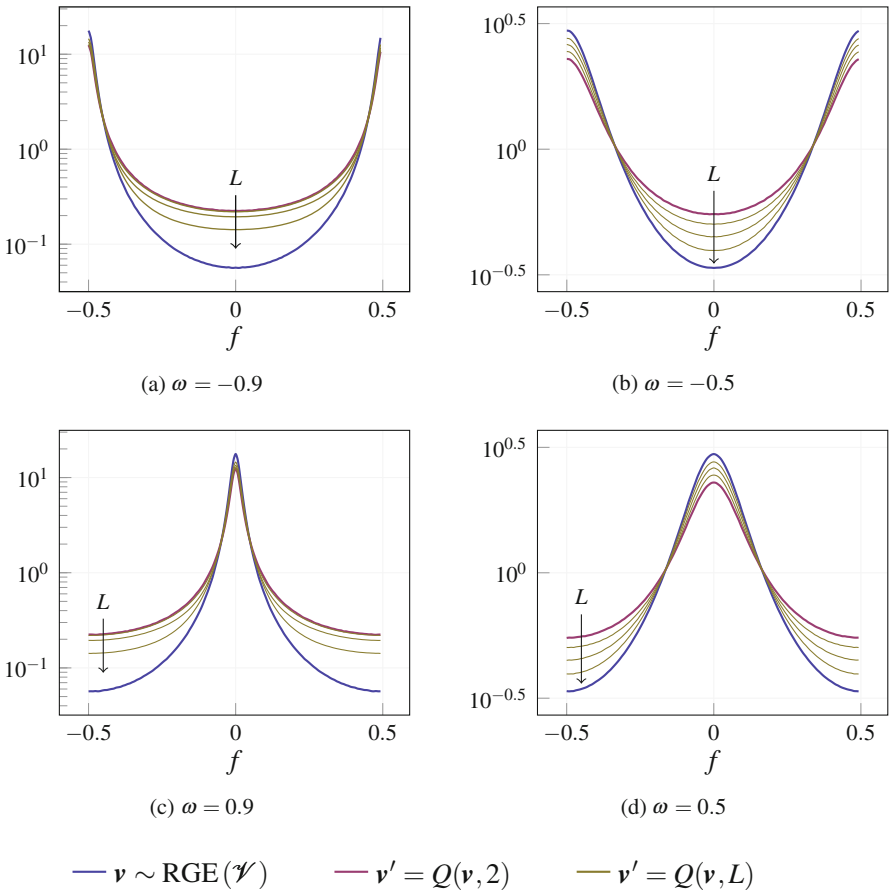
A first block is used to generate $\boldsymbol{v}$. From a practical point of view, one may synthesize a linear filter which reshapes realizations of a zero-mean, unit-variance, i.i.d. Gaussian process represented by $\boldsymbol{w}$ and with correlation/covariance matrix $\mathscr{W} = \mathbf{E}[\boldsymbol{w}\boldsymbol{w}^\top] = \boldsymbol{I}$ into vectors $\boldsymbol{v}$ with correlation $\mathscr{V}$. In terms of power spectrum, given the transfer function $H(f)$ of the filter, it is enough that $|H(f)|^2 = \Psi_{\boldsymbol{v}}(f)$ to obtain the desired process as output.

A second block follows, that is the quantization function where the degree of freedom is L, i.e., the number of possible output levels. The block quantizes the input sequences $\boldsymbol{v}$ to get $\boldsymbol{v}'$. In other words, the quantization stage produces digital words that assume only one of the L possible values and can be employed in the practical sensing of signals.

In order to evaluate the impact of the quantizing block on the overall generation process, the easiest way is to estimate the power spectrum of the quantized vectors $\boldsymbol{v}'$ with different L values, including the corner case where $L = 2$, i.e., $\boldsymbol{v}' \in \{-1, 1\}^n$.

Figure 5.3 shows results in terms of power spectrum of the quantized vectors $\boldsymbol{v}'$ for $n = 128$, $L = \{10, 6, 4, 2\}$, and for four different cases corresponding to different values of ω, including two low-pass profiles and two high-pass profiles. In the same plots the power spectrum of the non-quantized vectors $\boldsymbol{v}$ are also shown to highlight the impact of L in this process.

In particular, the magenta lines in Fig. 5.3 represent the power spectrum profiles for $L = 2$, which corresponds to the generation of antipodal sequences, and for which the introduced perturbation is maximum. This corner case has already been investigated in literature [3, 8], and the perturbation introduced in terms of correlation matrices can be estimated as follows.



(a) $\omega = -0.9$ (b) $\omega = -0.5$

(c) $\omega = 0.9$ (d) $\omega = 0.5$

Fig. 5.3 Power spectra of both $\boldsymbol{v} \sim \mathrm{RGE}(\mathcal{V})$ and its quantized versions $\boldsymbol{v}' = Q(\boldsymbol{v}, L)$ for different L levels including the corner case $L = 2$, i.e., where only two levels are allowed. Both (**a**) and (**b**) refer to a high-pass profile, while (**c**) and (**d**) have been computed in the high-pass case

Theorem 5.1 *Let* $Q(\cdot, 2) : \mathbb{R} \to \{+1, -1\}$ *be a clipping function such that*

$$Q(\zeta, 2) = \begin{cases} +1 & \text{if } \zeta \leq 0 \\ -1 & \text{if } \zeta < 0 \end{cases}$$

Let $\mathcal{V} \in \mathbb{R}^{n \times n}$ *be the correlation matrix of a stochastic process generating vector* $\boldsymbol{v} = (\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n-1})^\top$. *For a vector* $\boldsymbol{v}^c = (Q(\boldsymbol{v}_0, 2), Q(\boldsymbol{v}_1, 2), \ldots, Q(\boldsymbol{v}_{n-1}, 2))^\top$ *the corresponding correlation matrix* $\mathcal{V}^c$ *is represented by*

$$\mathcal{V}^c = \frac{2}{\pi} \frac{\mathrm{tr}(\mathcal{V})}{n} \sin^{-1}(\mathcal{V}) \tag{5.1}$$

where $\sin^{-1}(\cdot)$ *is applied componentwise.*

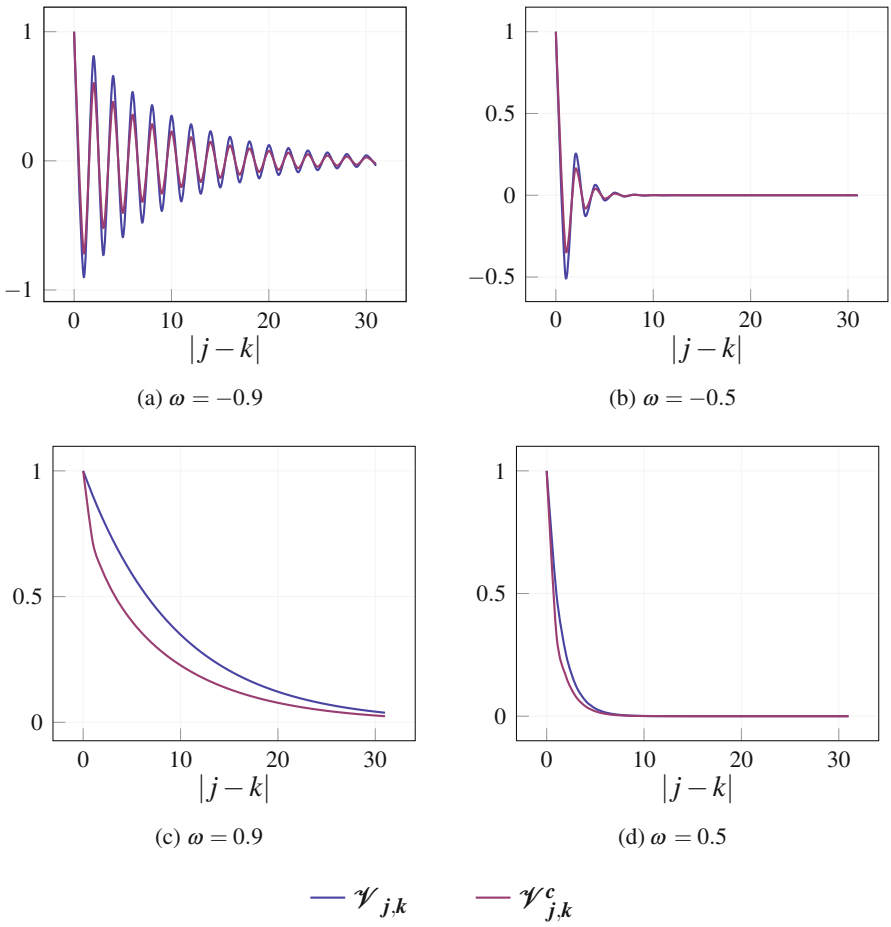


Fig. 5.4 Correlation profiles of first $n/4$ elements of both $\boldsymbol{v} \sim \text{RGE}(\mathcal{V})$ and its clipped versions. Both (**a**) and (**b**) refer to a high-pass profile, while (**c**) and (**d**) have been computed in the high-pass case

When considering our toy case, a visual representation of the two levels quantization impact in terms of difference between the generic $\mathcal{V}_{j,k}$ element of the correlation matrices is shown in Fig. 5.4. Two pairs of ω have been considered, one referring to the low-pass case and one to the high-pass profile.

5.3 Antipodal Sensing Sequences

The generation of quantized vectors, with particular reference to antipodal ones, with a prescribed second-order statistic requires approaches able to reduce, and possibly to eliminate, the intrinsic limitation described in the previous section. We describe here a few different strategies to obtain antipodal vectors generators

able to produce instances with a proper power spectrum or correlation matrix. We first discuss approaches generating antipodal symbols as instances of stationary processes, and then the non-stationary case will be taken into account.

5.3.1 Antipodal Generation in the Stationary Case

In order to begin a discussion about the generation of antipodal sensing sequences with a prescribed spectrum (or correlation profile), let us first consider for the sake of simplicity the stationary case.

A first method is related to Theorem 5.1, that implicitly suggests a way to counter the effect of clipping. One can invert equation (5.1) to have vectors $\boldsymbol{v} \in \{+1, -1\}^n$ with a proper correlation profile $\mathcal{V}$ by clipping zero-mean Gaussian vector $\boldsymbol{g}$ with a correlation matrix $\mathcal{G}$ computed as follows:

$$\mathcal{G} = \sin\left(\frac{\pi}{2}\frac{n}{\mathrm{tr}(\mathcal{V})}\mathcal{V}\right) \tag{5.2}$$

where as in (5.1), the function $\sin(\cdot)$ is computed componentwise. If the obtained $\mathcal{G}$ is a non-negative definite matrix, vectors $\boldsymbol{v}$ can be obtained by clipping the $\boldsymbol{g} \sim \mathrm{RGE}(\mathcal{G})$. We will refer to this approach as Clipped Gaussian (CG).

For the case $\omega = 0.9$, we propose in Fig. 5.5 the comparison between the imposed second-order characterizations (power spectrum and correlation) and the measured profiles evaluated over 10,000 antipodal sequences generated by CG. Results confirm the capability of the approach to generate antipodal vectors with a prescribed second-order statistics.

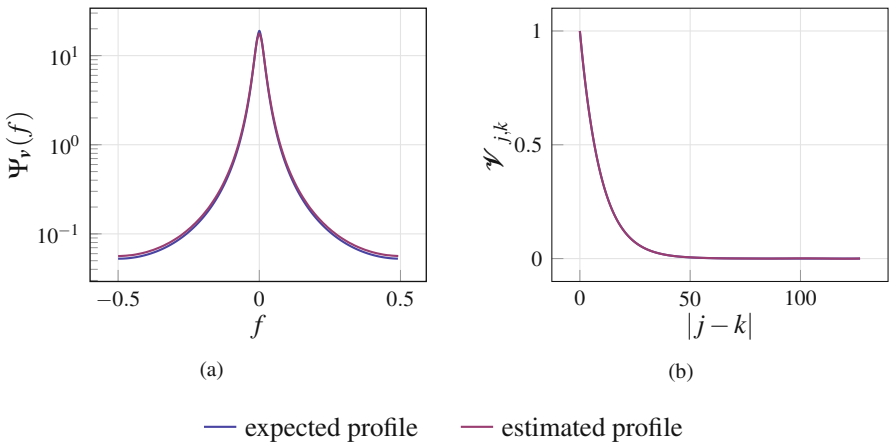


Fig. 5.5 Power spectrum (**a**) and correlation profile (**b**) for antipodal sequences generated by the Clipped Gaussian approach where $\omega = 0.9$

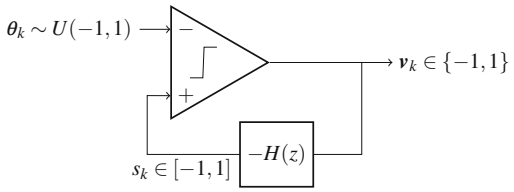


Fig. 5.6 Block scheme of the linear probability feedback process where θ_k is a random threshold and $H(z)$ is the transfer function of a causal time-invariant linear filter

An interesting and alternative way to generate antipodal symbols as instances of stationary process is the adoption of the so-called linear probability feedback (LPF) process [6, 7]. The main characteristic of this generator is the simplicity, as highlighted by the block scheme shown in Fig. 5.6 and adapted from [6]. The LPF mechanism relies on the design of a causal time-invariant linear filter with finite impulse response h_j with $j = \{1, 2, \ldots, Z\}$ and transfer function

$$H(z) = \sum_{j=1}^{Z} h_j z^{-j}$$

The process generated by the filter $-H(z)$ is then fed into a comparator and matched again θ_k, which is an instance of an independent random threshold uniformly distributed in $[-1, 1]$. The comparator yields antipodal values v_k that are the LPF output and that are continuously fed back into the filter. As discussed in [6, 7], the main assumption to ensure a correct symbols generation is that the filter output s_k is limited in the range $[-1, 1]$. Under the assumption that filter inputs are antipodal values, this constraint can be recast as follows.

$$\sum_{j=1}^{Z} |h_j| \leq 1 \tag{5.3}$$

In conclusion. the entire LPF mechanism can be summarized by the following set of statements.

$$\begin{aligned} \boldsymbol{v}_k &= \begin{cases} +1, \text{ if } s_k > \theta_k \\ -1, \text{ otherwise} \end{cases} \\ s_k &= \sum_{j=1}^{Z} h_j \boldsymbol{v}_{k-j} \qquad\qquad s_k \in [-1, 1] \\ \theta_k &\sim U(-1, 1) \end{aligned}$$

The main advantage of this generator is the possibility to express the exact power spectrum of the generated antipodal symbols analytically. If the filter transfer function $H(z)$ is known, then the power spectrum of the stationary process generating v_k is obtained by means of the following equation.

$$\Psi_v(f) = \frac{\left|1 + H(e^{2\pi i f})\right|^{-2}}{\int_{-1/2}^{1/2} |1 + H(e^{2\pi i f})|^{-2} \, \mathrm{d}f}$$

Despite the fact that this relation cannot be inverted to get $H(z)$ from $\Psi_v(f)$, its knowledge paves the way for many approximated approaches for the design of the filter that guarantees the generation of antipodal symbols with the assigned $\Psi_v(f)$.

Even if a complete description of the entire procedure is out of scope of this chapter, we quickly recap here the iterative synthesis procedure proposed in [6]. The approach is based on a simple gradient descent algorithm, modified to yield feasible solutions that satisfy (5.3), where filter taps h_j are iteratively evaluated in order to reduce an error function ϵ defined as

$$\begin{aligned}\epsilon &= \mathbf{E}\left[\left(\boldsymbol{v}_k + \sum_{j=1}^{Z} h_j \boldsymbol{v}_{k-j}\right)^2\right] \\ &= \mathcal{V}_{0,0} + 2\sum_{j=1}^{Z} \mathcal{V}_{j-1,0} h_j + \sum_{j=1}^{Z}\sum_{l=1}^{Z} \mathcal{V}_{j-1,l-1} h_j h_l\end{aligned}$$

where $\mathcal{V} \in \mathbb{R}^{Z \times Z}$ is a toeplitz correlation matrix depending on the desired power spectrum

$$\mathcal{V}_{j,k} = 2\int_0^{1/2} \Psi_v(f) \cos(2\pi(j-k)f) \mathrm{d}f \tag{5.4}$$

The procedure requires a feasible set of filter taps as starting point. Then, at the generic l-th step, taps are adjusted by means of the algorithm described in Table 5.1, where each iteration is basically composed by two steps. In the first step, a set of possible taps is evaluated following the direction in which the maximum error decrease is observed. Then, taps are rescaled to ensure that (5.3) is satisfied. The procedure ends when the residual error ϵ is smaller then a given tolerance.

In order to prove the effectiveness of the aforementioned approach, it has been used to generate sequences for the two spectral profiles considered in Fig. 5.3, namely for $\omega = \pm 0.9$ and $n = 128$. To this aim, we first evaluate $\mathcal{V}$ by (5.4), and then apply the above constrained gradient technique to get a set of Z taps.

Results are depicted in Fig. 5.7. The target power spectrum profiles (solid lines) are shown along with the spectral shapes estimated over 10,000 sequences obtained from the LPF generator. The order of the filter $-H(z)$, i.e., the number Z of taps considered is 2, 3, and 10, respectively. Note that in all considered cases, including also the simplest one where $Z = 2$, the observed profiles closely match the desired ones. These results are actually due in part to the smoothness of the required profile.

Table 5.1 Code sketch for filter taps evaluation in the LPF design

Require: $h_j^{(0)}$ with $j = \{1, \ldots, Z\}$ such that (5.3) is satisfied
Require: γ parameter controlling the rate of convergence
Require: $\mu > 0$ a small tolerance
  $l = 0$
  repeat
    for $j = 1$ **to** Z **do**
      $h'_j \leftarrow h_j^{(l)} - \gamma \frac{\partial}{\partial h_j} \epsilon^2$ ▷ direction of minimum error
    end for
    for $j = 1$ **to** Z **do**

$$h_j^{(l+1)} \leftarrow \begin{cases} h'_j & \text{if } \sum_{j=1}^{Z} |h'_j| \leq 1 \\ (1-\mu)\frac{h'_j}{\sum_{j=1}^{Z} |h'_j|} & \text{otherwise} \end{cases}$$

▷ taps rescaling to satisfy (5.3)

    end for
    $l \leftarrow l + 1$
  until $\epsilon \left(h_1^{(l+1)}, \ldots, h_Z^{(l+1)} \right)^2 \leq$ tolerance

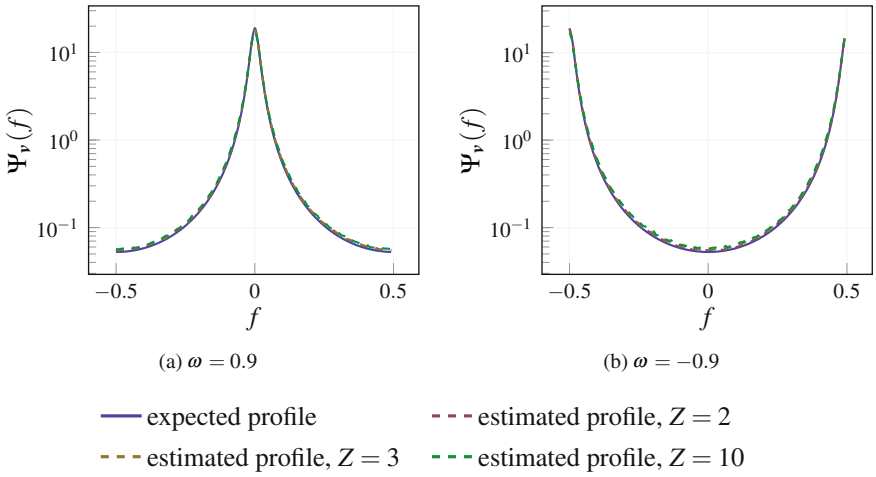


Fig. 5.7 Power spectrum for antipodal sequences generated by the LPF approach in the case (**a**) $\omega = 0.9$, (**b**) $\omega = -0.9$

In a real scenario the desired power spectrum can require a higher Z value depending on the spectrum shape to be matched. In general, we know that an increase in Z always corresponds to an improvement in the matching between the desired and achieved profiles, and in the limit case of $Z \to \infty$ it is possible to have a perfect match with any desired power spectrum profile.

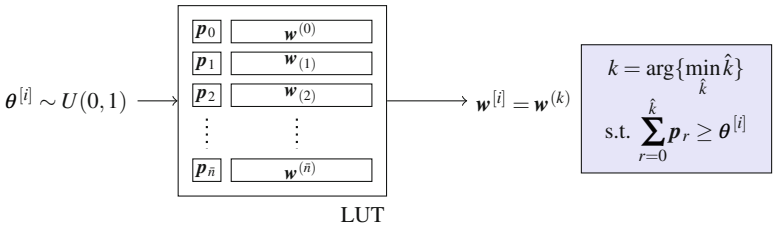


Fig. 5.8 Block scheme of the LUT-based antipodal vector generator, where $\{(\boldsymbol{p}_0, \boldsymbol{w}_0), \ldots, (\boldsymbol{p}_{\bar{n}}, \boldsymbol{w}_{\bar{n}})\}$ are couples of antipodal vectors with an associated probabilities to be generated, and $\theta^{(j)}$ is a random threshold uniformly distributed in $[0, 1]$ used to randomly select a vector $\boldsymbol{w}_k$ accordingly to the $\boldsymbol{p}_0, \ldots, \boldsymbol{p}_n$

5.3.2 Antipodal Generation in the Non-Stationary Case

In the non-stationary case, the CG approach can still be used for the generation of sequences with a given correlation profile. In fact, equation (5.2) does not pose any limitation on the stationarity of $\mathcal{V}$ and so of $\mathcal{G}$. However, the main limitation of the CG approach is that it is not a completely general method. As mentioned before, CG is based on the direct inversion of (5.1), but it is possible that the achieved $\mathcal{G}$ is not a positive-semidefinite matrix, and cannot be used as a correlation matrix.

A more general approach is proposed in [1], and it is based on a randomly addressed digital lookup table (LUT). The general structure of this generator is schematized in Fig. 5.8, and it is based on the assumption that a correlation profile corresponds to a probability assignment to all the 2^n possible sequences. A simple digital LUT is used to store sequences candidates for the generation. Each time an instance is required, the LUT is randomly addressed accordingly to such a probability assignment.

More formally, the aim of the approach is to generate n-dimensional antipodal random vectors $\boldsymbol{w} \in W^n$ with $W = \{-1, 1\}$, whose correlation matrix $\mathcal{W} = \mathbf{E}[\boldsymbol{w}\boldsymbol{w}^\top]$ is as close as possible to a given matrix $\mathcal{V}^{\text{ns}}$. Note that, since generated sequences are antipodal, the diagonal of $\mathcal{W}$ is made only of 1s, and this implies that either the diagonal of $\mathcal{V}^{\text{ns}}$ must presents the same profile, or that it should be possible to rescale $\mathcal{V}^{\text{ns}}$ in order to satisfy such constraint, i.e., $\mathcal{V}^{\text{ns}}_{j,j}$ is constant for $j = 0, \ldots, n$. In the rest of this section we refer to this class of correlation matrix where we also impose $\text{tr}(\mathcal{V}^{\text{ns}}) = n$.

At the generation of the i-th vector $\boldsymbol{w}^{[i]}$, the LUT is randomly addressed according to a proper probability assignment $p(\boldsymbol{w})$, in such a way that each vector $\boldsymbol{w}$ appears at the output with probability $p(\boldsymbol{w})$. From a theoretical point of view, this implies that the content of the LUT (i.e., all possible 2^n vectors $\boldsymbol{w}$ and a proper joint probability function $p : W^n \mapsto [0, 1]$) needs to be completely defined, with

$$\sum_{\boldsymbol{w} \in W^n} p(\boldsymbol{w}) = 1$$

Accordingly to a more practical point of view, the number of sequence whose probability is not null is in general much smaller than 2^n. If we indicate with P the support of the joint probability function p

$$P = \operatorname{supp} p = \{\boldsymbol{w} \in W^n \mid p(\boldsymbol{w}) > 0\}$$

it has been shown in [1] that its cardinality $\bar{n} = |P|$ is practically limited to a number of elements that is $\mathsf{O}(n^2)$.

This observation is very important from an implementation point of view, allowing us to estimate the memory allocation required to store the LUT. Each item is an n-bit string, so that the total number of bits required for the LUT is $\mathsf{O}(n^3)$ with an additional storage needed for the associated probability values, which remains compatible with a full hardware implementation.

As an example, an amount of 2 Mbit of memory is typically enough for n up to 128. Let us assume a number of sequences $\bar{n} = n^2 \approx 16 \cdot 10^3$. Let us also assume that the smaller probability is the one associated with $\boldsymbol{w}_0$, and it is in the order of magnitude of $\boldsymbol{p}_0 \approx 10^{-8} \approx 2^{-26}$. To express $\boldsymbol{p}_0$ with a simple but inefficient fixed point representation, 28 bits may be enough. Even with this simple approach, the memory allocation required to memorize $\boldsymbol{p}_0$ is effectively negligible with respect to that required to store $\boldsymbol{w}^{(0)}$. Using the same precision used for $\boldsymbol{p}_0$ for any entry of $\boldsymbol{p}$, a total amount of about 2.5 Mbit of memory is required.

Note also that this is, actually, a strong overestimation on the actual memory requirement. Referring to the previous example, vectors with a probability as low as $\boldsymbol{p}_0 \approx 10^{-8}$ have an expected impact on the actual correlation matrix $\mathscr{W}$ that is negligible. This paves the way to many possible optimization strategies based on the implementation point of view.

For example, one could fix a limited number of bit with which the unavoidably approximated value of $\boldsymbol{p}_j$ is represented. Dependently on the quantization and approximation strategy adopted, a minimum probability $\boldsymbol{p}_{\min}$ exists such that every $\boldsymbol{p}_j < \boldsymbol{p}_{\min}$ will be approximated with $\boldsymbol{p}_j = 0$. As a consequence, the number of sequences associated with a non-null probability is much smaller with respect to the ideal case. In other words, the table length $\bar{n}$ is decided by the number of bits used for representing the $\boldsymbol{p}_j$ and it is not necessarily $\mathsf{O}(n^2)$. The obtained $\mathscr{W}$ is just an approximation of the desired $\mathscr{V}^{\mathrm{ns}}$, but the memory requirements are strongly relaxed.

Thus said, let us discuss how it is possible to obtain the set of $\bar{n}$ sequence representing an assigned correlation profile with the associated probabilities values. Consider a set of antipodal sequences $(\boldsymbol{w}^{(0)}, \boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(\bar{n}-1)})$ with associated probabilities $\boldsymbol{p} = (\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{\bar{n}-1})^\top$, $\boldsymbol{p}_j \neq 0, \forall j$. The corresponding correlation matrix is given by

$$\mathscr{W} = \sum_{j=0}^{\bar{n}-1} \boldsymbol{p}_j \boldsymbol{w}^{(j)} \boldsymbol{w}^{(j)\top} \tag{5.5}$$

Our aim is to obtain the best possible match between $\mathscr{W}$ and the assigned correlation $\mathscr{V}^{\mathrm{ns}}$ independently of $\bar{n}$ that can be selected small as possible, and far from the intrinsic upper bound 2^n.

The solution of this problem is achieved by means of the minimization of the difference matrix $\boldsymbol{\Delta}$ defined as

$$\boldsymbol{\Delta} = \boldsymbol{\mathcal{V}}^{\text{ns}} - \boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{V}}^{\text{ns}} - \sum_{j=0}^{\bar{n}-1} \boldsymbol{p}_j \boldsymbol{w}^{(j)} \boldsymbol{w}^{(j)\top}$$

whose non-diagonal entries contain the deviations from the assigned correlation. This minimization is mapped into the solution on the following optimization problem:

$$\begin{aligned} &\min \|\boldsymbol{\Delta}\| \\ &\quad \boldsymbol{p}_j > 0 \qquad \text{for } j = \{0, 1, \ldots, \bar{n} - 1\} \\ &\text{s.t.} \sum_{j=0}^{\bar{n}-1} \boldsymbol{p}_j = 1 \end{aligned} \tag{5.6}$$

where $\|\boldsymbol{\Delta}\|$ can be simply computed as $\|\boldsymbol{\Delta}\| = \sum_{0 \le j<k<n} \left|\boldsymbol{\Delta}_{j,k}\right|$ by exploiting the fact that both correlation matrices are symmetric.

The problem represented by (5.6) has a non-linear objective function. This means that its solution is in general a hard task. Nevertheless, this particular formulation can be recast in a linear programming problem (LP) by the introduction of additional variables.

In more detail, $\boldsymbol{\Delta}$ is split into two $n \times n$ auxiliary matrices, one containing its positive part, $\boldsymbol{\Delta}^+$, and another one containing the negative part of $\boldsymbol{\Delta}$, namely $\boldsymbol{\Delta}^-$.

$$\begin{aligned} \boldsymbol{\Delta}^+_{j,k} &= \max\{\boldsymbol{\Delta}_{j,k}, 0\} \quad \text{for } j,k = 0, \ldots, n-1 \\ \boldsymbol{\Delta}^-_{j,k} &= \max\{-\boldsymbol{\Delta}_{j,k}, 0\} \text{ for } j,k = 0, \ldots, n-1 \end{aligned}$$

having as a consequences that $\boldsymbol{\Delta} = \boldsymbol{\Delta}^+ - \boldsymbol{\Delta}^-$ and $\|\boldsymbol{\Delta}\| = \sum_{0 \le j<k<n} \left(\boldsymbol{\Delta}^+_{j,k} + \boldsymbol{\Delta}^-_{j,k}\right)$.

The introduction of these additional variables allows us to recast the optimization problem (5.6) in the following LP:

$$\begin{aligned} &\min \textstyle\sum_{0 \le j<k<n} \boldsymbol{\Delta}^+_{j,k} + \boldsymbol{\Delta}^-_{j,k} \\ &\quad \boldsymbol{\Delta}^+ - \boldsymbol{\Delta}^- = \boldsymbol{\mathcal{V}}^{\text{ns}} - \sum_{j=0}^{\bar{n}-1} \boldsymbol{p}_j \boldsymbol{w}^{(j)} \boldsymbol{w}^{(j)\top} \\ &\text{s.t.} \sum_{j=0}^{\bar{n}-1} \boldsymbol{p}_j = 1 \\ &\quad \boldsymbol{\Delta}^+ \ge 0 \qquad \text{with } 0 \le j < k < n \\ &\quad \boldsymbol{\Delta}^- \ge 0 \qquad \text{with } 0 \le j < k < n \\ &\quad \boldsymbol{p}_j > 0 \qquad \text{for } j = 0, 1, \ldots, \bar{n} - 1 \end{aligned} \tag{5.7}$$

where matrix equalities and inequalities are meant to hold componentwise. To see that (5.7) is equivalent to (5.6), note that the minimization of $\sum_{0\le j<k<n}\left(\boldsymbol{\Delta}^{+}_{j,k}+\boldsymbol{\Delta}^{-}_{j,k}\right)$ ensures that at least one of $\boldsymbol{\Delta}^{+}_{j,k}$ and $\boldsymbol{\Delta}^{-}_{j,k}$ is zero for each couple of indexes j,k thus implying that $\boldsymbol{\Delta}^{+}$ and $\boldsymbol{\Delta}^{-}$ are, respectively, the positive and negative part of $\boldsymbol{\Delta}$ as defined before. Let us focus on a certain j,k pair. This implies $\boldsymbol{\Delta}^{+}_{j,k}+\boldsymbol{\Delta}^{-}_{j,k}=|\boldsymbol{\Delta}_{j,k}|$ so that $\sum_{0\le j<k<n}\boldsymbol{\Delta}^{+}_{j,k}+\boldsymbol{\Delta}^{-}_{j,k}=\sum_{0\le j<k<n}|\boldsymbol{\Delta}_{j,k}|$ as required.

Clearly, the target is achieved when $\boldsymbol{\Delta}^{+}=\boldsymbol{\Delta}^{-}=0$, which means that the imposed correlation is the assigned one, i.e., $\sum_{j=0}^{\bar{n}-1}\boldsymbol{p}_j\boldsymbol{w}^{(j)}\boldsymbol{w}^{(j)\top}=\boldsymbol{\mathcal{V}}^{\mathrm{ns}}$. As a further remark, we are considering here the generation of instances of non-stationary process nevertheless this method can be also used when $\boldsymbol{\mathcal{V}}^{\mathrm{ns}}$ is a correlation matrix characterizing a stationary process.

Moving a step forward to the solution of (5.7), we can observe that there are $V=\bar{n}+2N$ degrees of freedom with $N=\binom{N}{2}$ coming from:

- $\bar{n}$ degrees of freedom are the probabilities $\boldsymbol{p}$;
- for each $n\times n$ symmetric matrix $\boldsymbol{\Delta}^{+}$ and $\boldsymbol{\Delta}^{-}$ we have $N=\binom{n}{2}$ degrees of freedom counting the number of independent non-diagonal entries;

while the number of equality constraint is $N+1$:

- N equality constraints given by the matching between $\boldsymbol{\mathcal{V}}^{\mathrm{ns}}$ and $\boldsymbol{\mathcal{W}}$;
- one equality constraints to enforce probability normalization.

As previously anticipated, we are assuming that $\bar{n}<2^n$ and in particular that $\bar{n}=\mathsf{O}(n^2)$. The fact that (5.7) is an LP problem motivates this. As a first comment, the problem is surely feasible since, for any given $\hat{\boldsymbol{w}}\in W^n$ we may set $p(\hat{\boldsymbol{w}})=1$, $p(\boldsymbol{w})=0$ for $\boldsymbol{w}\in W^n-\{\hat{\boldsymbol{w}}\}$, and compute $\hat{\boldsymbol{\Delta}}=\boldsymbol{\mathcal{V}}^{\mathrm{ns}}-\hat{\boldsymbol{w}}\hat{\boldsymbol{w}}^{\top}$ which guarantees the feasibility space is certainly non-null. Moreover, since we are solving an LP we know that the minimum is surely achieved in a vertex of the polytope that is its feasibility space. To define a vertex we need as many equality constraints as the degrees of freedom. Hence at least one of the solution we seek is such that not only the $N+1$ equality constraint are satisfied but also $V-(N+1)$ inequality constraint out of the V available must be active.

If $V-(N+1)$ non-equality constraints are active, then $V-(N+1)$ degrees of freedom are set to zero. Since at most $2N$ of those degrees of freedom can be entries of $\boldsymbol{\Delta}^{+}$ and $\boldsymbol{\Delta}^{-}$, we get that at least $V-(N-1)-2N$ probabilities are null thus not more then $N+1=\mathsf{O}(n^2)$ of them can be nonzero.

To clarify the proposed method, we present a generic formulation of the LP problem coupled with a simple example. We write (5.7) in a standard form

$$\begin{aligned}&\min\ \boldsymbol{c}\boldsymbol{q}\\ &\text{s.t.}\ \ \boldsymbol{C}\boldsymbol{q}=\boldsymbol{b}\\ &\qquad\boldsymbol{q}\ge 0\end{aligned}\tag{5.8}$$

where $\boldsymbol{q}$ is the vector of variables to be determined, expressed as

$$\boldsymbol{q} = (\boldsymbol{\Delta}^{+}_{0,1}, \ldots, \boldsymbol{\Delta}^{+}_{n-2,n-1}, \boldsymbol{\Delta}^{-}_{0,1}, \ldots, \boldsymbol{\Delta}^{-}_{n-2,n-1}, \boldsymbol{p}_0, \ldots, \boldsymbol{p}_{\bar{n}-1})^{\top}$$

$\boldsymbol{c} \in \mathbb{R}^{(\bar{n}+2N)}$ is a vector defining the linear objective function that is composed as follows:

$$\boldsymbol{c}_j = \begin{cases} 1 & \text{for } j = 0, \ldots, 2N-1 \\ 0 & \text{for } j = 2N, \ldots, 2N+\bar{n}-1 \end{cases}$$

and finally, the $(N+1)\times(\bar{n}+2N)$ matrix $\boldsymbol{C}$ and the $\boldsymbol{b} \in \mathbb{R}^{(N+1)}$ vector characterizing equality constraints are

$$\boldsymbol{C} = \left(\begin{array}{c|c|ccc} 0, \ldots, 0 & 0, \ldots, 0 & 1, & \ldots, & 1 \\ \boldsymbol{I}_N & -\boldsymbol{I}_N & \left[\!\left[\boldsymbol{w}^{(0)} \boldsymbol{w}^{(0)\top} \right]\!\right], & \ldots, & \left[\!\left[\boldsymbol{w}^{(\bar{n}-1)} \boldsymbol{w}^{(\bar{n}-1)\top} \right]\!\right] \end{array} \right)$$

and

$$\boldsymbol{b} = (1, \boldsymbol{\mathcal{V}}^{ns}_{0,1}, \ldots, \boldsymbol{\mathcal{V}}^{ns}_{n-2,n-1})$$

where $[\![\cdot]\!]$ indicates any operator that takes a symmetric matrix and rearranges the entries in its higher-right part (excluding the diagonal) in a column vector. In this set of constraint the first row of $\boldsymbol{C}$ and the first element of $\boldsymbol{b}$ impose that the sum of all probability values is equal to one, while all remaining elements are used for the match between actual and desired correlation profile.

The solution of this optimization problem is a lookup table with $\bar{n} \leq N + 1 = \frac{n(n-1)}{2} + 1$ entries coupled with the associated probability vector. As an example, for $n = 3$ and for an assigned correlation matrix

$$\boldsymbol{\mathcal{V}}^{ns} = \begin{pmatrix} 1 & 0.9^2 & 0.9^{2.5} \\ 0.9^2 & 1 & 0.9^1 \\ 0.9^{2.5} & 0.9^1 & 1 \end{pmatrix}$$

and by setting $\bar{n} = N - 1 = 4$, we have

$$\boldsymbol{c} = (1, 1, 1, 1, 1, 1, 0, 0, 0, 0)$$

$$\boldsymbol{C} = \left(\begin{array}{ccc|ccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & -1 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 1 & -1 & 1 & -1 \end{array} \right)$$

$$\boldsymbol{b} = (1, 0.9^2, 0.9^{2.5}, 0.9^1)^{\top}$$

$$\begin{aligned}\boldsymbol{w}^{(0)} &= (1,1,1)\\ \boldsymbol{w}^{(1)} &= (1,1,-1)\\ \boldsymbol{w}^{(2)} &= (1,-1,-1)\\ \boldsymbol{w}^{(3)} &= (1,-1,1)\end{aligned}$$

The LP problem in (5.8) maps the task of designing the random vector generator into a precise mathematical procedure and highlights its main property (i.e., the fact that the cardinality of the lookup table is $\mathsf{O}(n^2)$) that ensures the viability of the proposed approach. Nevertheless, an important aspect is a stumbling block to the straightforward implementation of such a method in a real scenario.

Though, the solution has a number of non-null entries that is a $\mathsf{O}(n^2)$, the optimization problem entails a number of variables that is exponential in the size of the problem due to the fact that the positions of non-null $\boldsymbol{p}$ elements are unknown. Actually, though it goes out of the scope of this book, it can be proved that solving (5.8) including all 2^n possible antipodal vector is NP-hard. To tackle it for reasonable and useful values of n we should rely on slightly more advanced Operation Research methods. The fact that the number of columns of $\boldsymbol{C}$ is huge (it increases exponentially with n) encourages us to look into *column generation* methods especially because we know that a solution exists involving not more than $2N + 1$ columns: this is corresponding to the $N + 1$ probabilities and to the N potentially nonzero deviations.

Thanks to the linearity of (5.8) the proposed approach, column generation methods guarantee that optimality can be pursued iteratively by taking any candidate set of columns in the objective function and in the matrices $\boldsymbol{C}$ and possibly substituting its columns with new columns one at a time so that the objective function value decreases at each iteration.

The first step is to define a point in the feasibility space, as a trivial solution we select only one antipodal vector $\boldsymbol{w}^{(0)}$ with probability equal to one such that (5.8) can be recast in the following optimization problem.

$$\begin{aligned}&\min \hat{\boldsymbol{c}}\hat{\boldsymbol{q}}\\ &\text{s.t.} \quad \hat{\boldsymbol{C}}\hat{\boldsymbol{q}} = \hat{\boldsymbol{b}}\\ &\qquad \hat{\boldsymbol{q}} \geq 0\end{aligned} \tag{5.9}$$

with:

$$\hat{\boldsymbol{c}} = \left(1,\ldots,1\,\middle|\,1,\ldots,1\,\middle|\,0\right)$$

and

$$\hat{\boldsymbol{C}} = \left(\begin{array}{c|c|c} 0,\ldots,0 & 0,\ldots,0 & 1 \\ \boldsymbol{I}_N & -\boldsymbol{I}_N & \left[\!\left[\boldsymbol{w}^{(0)}\boldsymbol{w}^{(0)\top}\right]\!\right] \end{array}\right)$$

The second step is to identify a second column that can be added to the problem such that the corresponding solution has a lower objective function and then iteratively solve the new optimization problem and search for a new column until a proper stop criteria is satisfied. In this way the LP problem is never solved with a number of variables greater than $2N + \bar{n}$.

To deal with this aspect we need a function that counts the impact of a single antipodal vector that is not currently addressed in the objective function, such a function is the *reduced cost* that quantifies the increase in the objective function that one obtains by introducing a new antipodal vector $\boldsymbol{w}^{(1)}$ in (5.9). For the optimization problem under investigation, the reduced cost associated with $\boldsymbol{w}^{(1)}$,$f_C(\boldsymbol{w}^{(1)})$, can be evaluated as

$$f_C(\boldsymbol{w}_1) = -\hat{\boldsymbol{c}}(\hat{\boldsymbol{C}})^{-1} \begin{pmatrix} 1 \\ [\![\boldsymbol{w}^{(1)} \boldsymbol{w}^{(1)\top}]\!] \end{pmatrix}$$

The aim is to look for a new column with a minimum negative reduced cost so that the objective function value decreases either until it is zero or until there is not any new column with a negative reduced cost. The first stop criterion means that the problem solution perfectly matches the desired correlation matrix while no other columns with a negative reduced cost means that the minimum point in the objective function is reached although the corresponding LUT generates antipodal sequences with a correlation matrix that does not match the desired profile.

Therefore, the iterative mechanism requires a new column with minimum negative reduced cost to be added to the optimization problem. Let $\boldsymbol{w}^{(1)}$ be the obtained antipodal vector, introducing this vector in the optimization problem means a new column in the matrix $\hat{\boldsymbol{C}}$ and a new coefficient in the vector $\hat{\boldsymbol{c}}$ such that:

$$\hat{\boldsymbol{c}} = \left(1, \ldots, 1 \middle| 1, \ldots, 1 \middle| 0 \; 0\right)$$

and

$$\hat{\boldsymbol{C}} = \left(\begin{array}{c|c|cc} 0, \ldots, 0 & 0, \ldots, 0 & 1 & 1 \\ \boldsymbol{I}_N & -\boldsymbol{I}_N & [\![\boldsymbol{w}^{(0)} \boldsymbol{w}^{(0)\top}]\!] & [\![\boldsymbol{w}^{(1)} \boldsymbol{w}^{(1)\top}]\!] \end{array}\right)$$

The problem is now how to identify a column with a minimum negative reduced cost without spanning all possible antipodal vectors. This is a special case of a Binary Quadratic Problem (BQP) in n antipodal variables, i.e., the element of the vector $\boldsymbol{w}^{(i)}$ that is added to the problem in the i-th iteration. Since BQPs are NP-hard the authors in [1] recast the problem of finding $\boldsymbol{w}^{(i)}$ with the minimum reduced cost into a Binary Linear Programming (BLP) problem solved by heuristic approaches (among which they chose the evolutionary technique tailored to BQP problems as in [4]).

It is important to remark that the optimization problem (5.6) is solved off-line only once for each desired correlation matrix, i.e., only once for every class of signal to be acquired in according to methods described in Chaps. 3 and 4. The sequence

generation is instead running on-line, and simply requires a selection on the LUT entries with the probability profile that is the off-line solution of (5.6). As already anticipated, the evaluated profile $\boldsymbol{p}$ needs to be quantized within a finite set of value to be used in a real implementation, thus introducing unavoidable approximations.

In this last part, we want to focus on this aspect, highlighting that a twofold consequence follows:

- The solution of (5.6) is a set of antipodal sequences $(\boldsymbol{w}^{(0)}, \boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(\bar{n}-1)})$ with associated probabilities $\boldsymbol{p} = (\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{\bar{n}-1})^\top$, $\boldsymbol{p}_j \neq 0, \forall j$, both sets made of $\bar{n}$ elements. In the physically implemented lookup table, however, a quantized probabilities vector $\tilde{\boldsymbol{p}}$ is used, with $\mathtt{b}_p$ bit precision. Since it is only $\tilde{\boldsymbol{p}} \approx \boldsymbol{p}$, then

$$\tilde{\mathscr{W}} = \sum_{j=0}^{\bar{n}-1} \tilde{\boldsymbol{p}}_j \boldsymbol{w}^{(j)} \boldsymbol{w}^{(j)\top}$$

  that is just an approximation of the $\mathscr{W}$ computed in (5.5). The higher the $\mathtt{b}$, the better the approximation.
- Due to the approximation, some elements in $\tilde{\boldsymbol{p}}$ may be zero, even if the corresponding elements of $\boldsymbol{p}$ are nonzero. In other words, the used entries of the antipodal sequence set are $\tilde{n} < \bar{n}$.

We address the impact of such a limitation by directly showing the deviation in the correlation profile from the desired one in a single case, when $n = 32$, and the desired $\mathscr{V}^{\text{ns}}$ depicted in Fig. 5.9. In this case, solution of (5.6) gives $\bar{n} = 497$, and indicating with $\boldsymbol{w}^{(0)}$ the vector with the smallest probability, it is $\boldsymbol{p}_0 = 1.14 \cdot 10^{-7}$.

By considering a proper quantization function to encode the $\boldsymbol{p}$ with $\mathtt{b}_p \in \{4, 6, 8, 10\}$, the number of nonzero elements of the approximated probability vector $\tilde{\boldsymbol{p}}$ ranges from $\tilde{n} = 43$, for $\mathtt{b}_p = 4$, to $\tilde{n} = 447$, for $\mathtt{b}_p = 10$. The achieved $\tilde{\mathscr{W}}$ matrices are shown in Fig. 5.10. Note that using $\mathtt{b}_p = 8$ may be considered enough to get satisfactory results.

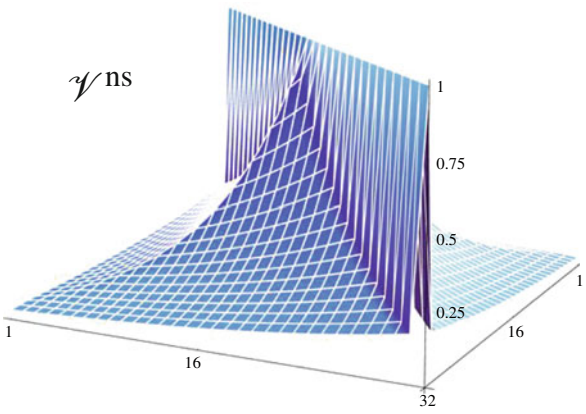


Fig. 5.9 Desired $\mathscr{V}^{\text{ns}}$ to be achieved in the example of the LUT-based antipodal vector generator

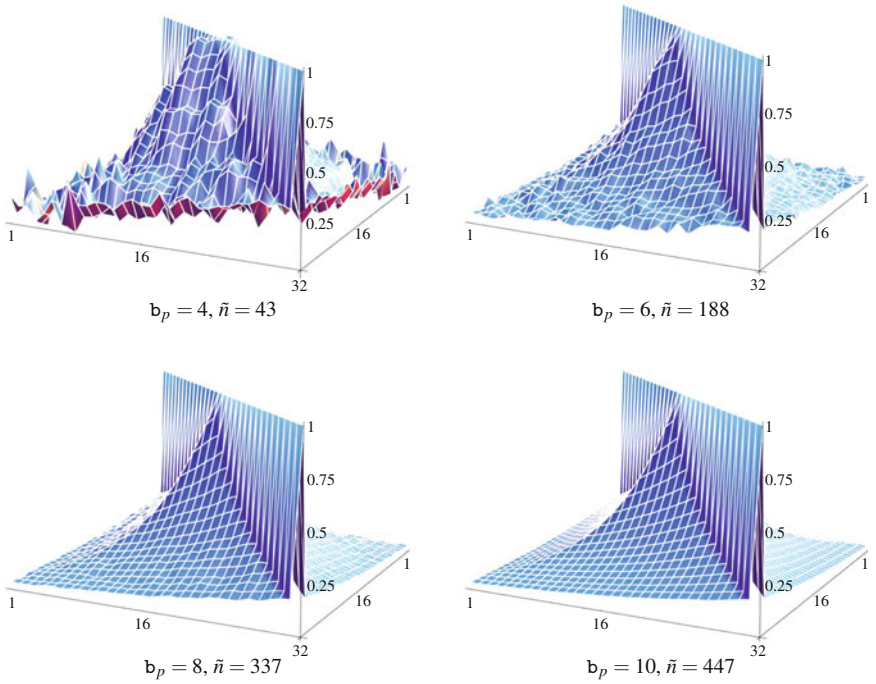


Fig. 5.10 Approximated correlation matrices $\tilde{\mathscr{W}}$ achieved by quantizing the probability vector $\boldsymbol{p}$ using a different number of bits $\mathtt{b}_p$

5.3.3 *Feasibility Space for Antipodal Sequences Generation*

As discussed before, generating antipodal sequences $\boldsymbol{v} = (\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{n-1})^\top$ with a prescribed $n \times n$ correlation matrix could be a hard task and the discussed generators must be taken into account in order to overcome such impasse. Nevertheless it is not enough to affirm that, with the constraint that sequences are composed by only ones and minus ones, we may impose any correlation profile.

Without the constraint $\boldsymbol{v} \in \{-1, 1\}^n$, the set of matrices $\mathscr{V}$ that are candidates to be a correlation matrix is that of $n \times n$ non-negative definite matrices. Taking a first step towards imposing antipodality we first observe that $\mathscr{V}$ must be an $n \times n$ non-negative definite matrix with unit diagonal entries, since $\boldsymbol{v} \in \{-1, 1\}^n$ we have $\mathscr{V}_{j,j} = \mathbf{E}[\boldsymbol{v}_j \boldsymbol{v}_j] = 1$. If the diagonal values are not ones but they are identical, one can simply rescale the matrix.

Let us indicate the set of non-negative definite $n \times n$ matrices with unit diagonal entries as S_n^{NND}. More formally, for an $n \times n$ matrix $\mathscr{V}$, we define $\mathscr{V}^{(l)}$ as its $l \times l$ upper-left submatrix such that, in addition to the requirements over the diagonals entries values, the Sylvester's criterion guarantees that $\mathscr{V} \in S_n^{\mathrm{NND}}$, if the determinants of all possible submatrices $\mathscr{V}^{(l)}$ are non-negative, with $l = 1, \ldots, n$.

$$\det \mathcal{V}^{(l)} \geq 0 \quad \text{for } l = 1, \ldots, n \tag{5.10}$$

Where these inequalities are defined in the space of the parameters $\mathcal{V}_{j,k}$ with $0 < j < k < n$ since correlation matrices are symmetric by definition, $\mathcal{V}_{j,k} = \mathbf{E}[\boldsymbol{v}_j \boldsymbol{v}_k] = \mathbf{E}[\boldsymbol{v}_k \boldsymbol{v}_j] = \mathcal{V}_{k,j}$. As an example, for $n = 2$ the equation (5.10) is $1 - \mathcal{V}_{0,1}^2 \geq 0$ which implies $|\mathcal{V}_{0,1}| < 1$.

When we impose $\boldsymbol{v} \in \{-1, 1\}^n$, due to the discrete nature of the vector $\boldsymbol{v}$, we can express the correlation matrix $\mathcal{V}$ in a different way.

$$\mathcal{V} = \mathbf{E}[\boldsymbol{v}\boldsymbol{v}^\top] = \sum_{\boldsymbol{v} \in \{-1,+1\}^n} p(\boldsymbol{v}) \boldsymbol{v}\boldsymbol{v}^\top = \sum_{\boldsymbol{v} \in \{-1,1\}^n} p(\boldsymbol{v}) \boldsymbol{v}^\times \tag{5.11}$$

where the $n \times n$ matrices $\boldsymbol{v}^\times = \boldsymbol{v}\boldsymbol{v}^\top$ remain implicitly defined. Note that, from $\boldsymbol{v} \in \{-1, 1\}^n$ we get $\boldsymbol{v}^\times = (-1\boldsymbol{v})^\times$. As a consequence the probabilities associated with $\boldsymbol{v}$ and $-1\boldsymbol{v}$ are the same, i.e., $p(\boldsymbol{v})$ implicitly defines $p(-1\boldsymbol{v})$.

Such a notation is useful to highlight how all possible matrices $\boldsymbol{v}^\times$ with the associated probabilities values $p(\boldsymbol{v})$ are the degree of freedom that one can use to impose that $\mathcal{V} \in S_n^{\text{NND}}$, i.e., the selected pairs $\boldsymbol{v}^\times$, $p(\boldsymbol{v})$ are such that (5.10) holds. Since all possible $p(\boldsymbol{v})$ are probabilities that an antipodal sequence occurs, we have $p(\boldsymbol{v}) \geq 0$ and $\sum_{\boldsymbol{v} \in \{-1,1\}^n} p(\boldsymbol{v}) = 1$, and considering also (5.11), we are defining a convex hull of the 2^n points $\boldsymbol{v}^\times$ with $\boldsymbol{v} \in \{-1, 1\}^n$, i.e., a polytope that we will indicate as S_n^{Ant}. Such a polytope represents the ensemble of correlation matrices $\mathcal{V}$ that satisfy the antipodality constrain.

For example, if $n = 2$ only four sequences exist,

$$\boldsymbol{v}_0 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad \boldsymbol{v}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \quad \boldsymbol{v}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad \boldsymbol{v}_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

and the associated $\boldsymbol{v}^\times$ matrices are

$$\boldsymbol{v}_0^\times = \boldsymbol{v}_3^\times = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad \boldsymbol{v}_1^\times = \boldsymbol{v}_2^\times = \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}$$

where $\boldsymbol{v}^\times = (-1\boldsymbol{v})^\times$ holds such that only two probabilities values are enough to obtain (5.11).

$$\mathcal{V} = \boldsymbol{p}_0 \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \boldsymbol{p}_1 \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & \boldsymbol{p}_0 - \boldsymbol{p}_1 \\ \boldsymbol{p}_0 - \boldsymbol{p}_1 & 1 \end{pmatrix} \tag{5.12}$$

with $\boldsymbol{p}_0, \boldsymbol{p}_1 \geq 0$ and $\boldsymbol{p}_0 + \boldsymbol{p}_1 = 1$ that yield values of $\mathcal{V}_{0,1} = \boldsymbol{p}_0 - \boldsymbol{p}_1$ in $[-1, 1]$. Such constraints, as expected, confirm (5.10) that says $1 - (\boldsymbol{p}_0 - \boldsymbol{p}_1)^2 > 0$. Note that (5.11) represents a convex combination of non-negative definite matrices which is by definition a non-negative defined matrix.

Referring to the CG approach discussed in Sect. 5.3.1, the generator produces antipodal sequences by clipping instances of a multivariate zero-mean, unit-variance

Gaussian vector with correlations matrix $\mathscr{G}$ evaluated by (5.2). Focusing on an assigned profile $\mathscr{V}$ with unit entries on the diagonal we have

$$\mathscr{G}_{j,k} = \sin\left(\frac{\pi}{2}\mathscr{V}_{j,k}\right) \tag{5.13}$$

Such an approach is successful only if the matrix $\mathscr{G}$ built from $\mathscr{V}$ is non-negative definite, i.e., it satisfies the Sylvester's criterion

$$\det \mathscr{G}^{(l)} \geq 0 \quad \text{for } l = 1, \ldots, n \tag{5.14}$$

Let us indicate with S_n^{Gau} the set of matrices $\mathscr{V}$ for which this happens, i.e., the set of correlation profile that can be obtained with the GC approach. This additional constraint implies that if one wants to use CG generation, the set of possible correlation profiles is restricted as follows:

$$S_n^{\text{Gau}} \subseteq S_n^{\text{Ant}} \subseteq S_n^{\text{NND}} \tag{5.15}$$

For the first inclusion, CG is a method to generate antipodal sequences and so the imposed correlation profile must be a point inside S_n^{Ant}. It is easy to anticipate that, for n large enough, the above inclusions are strict. In fact, having in mind the set of involved parameters $\mathscr{V}_{j,k}$ with $0 < j < k < n$:

- S_n^{NND} is defined by a set of n-th degree polynomial inequalities (5.10);
- S_n^{Ant} is an n-dimensional polytope (5.11) defined by linear inequalities;
- S_n^{Gau} is defined by a set of transcendental inequalities (5.14).

Without any surprise, $S_1^{\text{Gau}} = S_1^{\text{Ant}} = S_1^{\text{NND}}$. Also for $n = 2$ we have

$$\text{for } S_n^{\text{NND}} \text{ (5.10) requires } 1 - \mathscr{V}_{0,1}^2 \geq 0$$

$$\text{for } S_n^{\text{Ant}} \text{ (5.12) requires } |\mathscr{V}_{0,1}| \leq 1$$

$$\text{for } S_n^{\text{Gau}} \text{ (5.14) requires } 1 - \sin^2\left(\tfrac{\pi}{2}\mathscr{V}_{0,1}\right) \geq 0$$

where first two constraints hold for $\mathscr{V}_{0,1} \in [-1, 1]$ while last one is always true such that the only requirement for CG method is on $\mathscr{V}$, i.e., $1 - \mathscr{V}_{0,1}^2 \geq 0$.

For larger n, the difference between polynomial and linear inequalities comes into play. When $n = 3$ we have the following parameters:

$$\mathscr{V} = \begin{pmatrix} 1 & a & b \\ a & 1 & c \\ b & c & 1 \end{pmatrix}$$

where $a = \mathscr{V}_{0,1} = \mathscr{V}_{1,0}$, $b = \mathscr{V}_{0,2} = \mathscr{V}_{2,0}$, and $c = \mathscr{V}_{1,2} = \mathscr{V}_{2,1}$. All these parameter are mapped in the 3 set of matrices as follows. For $\mathscr{V} \in S_3^{\text{NND}}$ we have

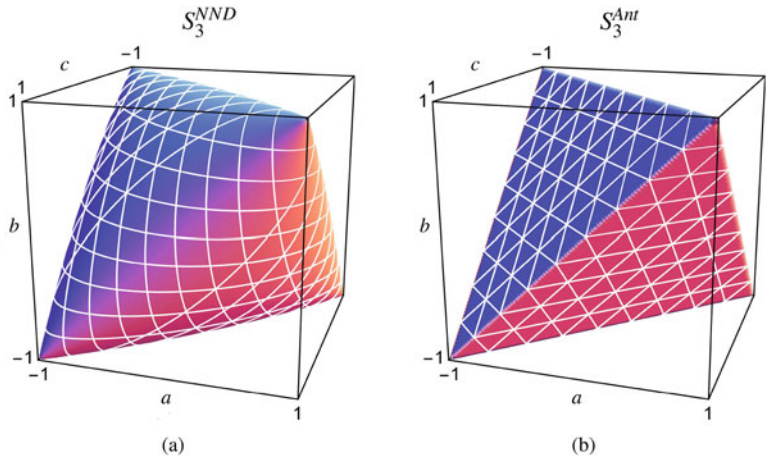


Fig. 5.11 Feasibility spaces in the three parameters composing a correlation matrix with $n = 4$. (**a**) is for unit diagonal non-negative definite matrices $\mathcal{V} \in S_3^{\text{NND}}$; (**b**) is for unit diagonal non-negative definite matrices that can be a correlation profile associated with antipodal sequences $\mathcal{V} \in S_3^{\text{Ant}}$

$$\begin{cases} 1 - a^2 \geq 0 \\ 1 + 2abc - a^2 - b^2 - c^2 \geq 0 \end{cases} \tag{5.16}$$

Such inequalities define the set S_3^{NND} shown in Fig. 5.11a while if we are imposing the $\boldsymbol{v} \in \{-1, 1\}^3$ the feasibility space is limited as in Fig. 5.11b. This last polytope is defined following the procedure used for (5.12). With $n = 3$ the correlation profile of a stochastic process generating antipodal sequences is defined by four matrices and by the associated probabilities,

$$\mathcal{V} = \begin{pmatrix} 1 & a & b \\ a & 1 & c \\ b & c & 1 \end{pmatrix} = \boldsymbol{p}_0 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}^{\times} + \boldsymbol{p}_1 \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}^{\times} + \boldsymbol{p}_2 \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}^{\times} + \boldsymbol{p}_3 \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}^{\times}$$

where the correlation matrix entries can be written as function of the probabilities values.

$$\begin{aligned} a &= -\boldsymbol{p}_0 - \boldsymbol{p}_1 + \boldsymbol{p}_2 + \boldsymbol{p}_3 \\ b &= -\boldsymbol{p}_0 + \boldsymbol{p}_1 - \boldsymbol{p}_2 + \boldsymbol{p}_3 \\ c &= \boldsymbol{p}_0 - \boldsymbol{p}_1 - \boldsymbol{p}_2 + \boldsymbol{p}_3 \end{aligned}$$

In order to guarantee that values $\boldsymbol{p}_0, \ldots, \boldsymbol{p}_3$ are probabilities we impose $\boldsymbol{p}_0\boldsymbol{p}_1\boldsymbol{p}_2\boldsymbol{p}_3 \geq 0$ and $\sum_{i=0}^{3} \boldsymbol{p}_i = 1$. Such inequalities identify the set of correlation matrices subject to the antipodality constraint shown in Fig. 5.11b.

$$\begin{cases} 0 \leq \frac{1}{4}(1-a-b+c) \leq 1 \\ 0 \leq \frac{1}{4}(1-a+b-c) \leq 1 \\ 0 \leq \frac{1}{4}(1+a-b-c) \leq 1 \\ 0 \leq \frac{1}{4}(1+a+b+c) \leq 1 \end{cases} \tag{5.17}$$

Also for $n = 3$ the feasibility space of the correlation profiles that can be obtained by CG is congruent with S_3^{Ant}. In this case the region shown in Fig. 5.11b corresponds also to the following inequalities obtained by (5.14) for $n = 3$.

$$\begin{cases} 1+\ 1-a^2 \geq 0 \\ 1+\ 2abc-a^{-}b^2-c^2 \geq 0 \\ 1+\ 2\sin\left(\frac{\pi}{2}a\right)\sin\left(\frac{\pi}{2}b\right)\sin\left(\frac{\pi}{2}c\right)+ \\ \quad -\sin\left(\frac{\pi}{2}a\right)^2-\sin\left(\frac{\pi}{2}b\right)^2-\sin\left(\frac{\pi}{2}c\right)^2 \geq 0 \end{cases} \tag{5.18}$$

Although for $n \leq 3$ the GC method correctly works for every $\mathcal{V} \in S_n^{\text{Ant}}$ we do not have yet evidence on how it works for higher values of n. For $n = 4$ we may consider

$$\mathcal{V} = \begin{pmatrix} 1 & 3/10 & 3/10 & 3/5 \\ 3/10 & 1 & 3/10 & 3/5 \\ 3/10 & 3/10 & 1 & 3/5 \\ 3/5 & 3/5 & 3/5 & 1 \end{pmatrix}$$

for which we have

$$\mathcal{V} = \frac{1}{80}\left[\begin{pmatrix}+1\\-1\\-1\\+1\end{pmatrix}^{\times}+\begin{pmatrix}+1\\-1\\+1\\-1\end{pmatrix}^{\times}+\begin{pmatrix}+1\\+1\\-1\\-1\end{pmatrix}^{\times}+\begin{pmatrix}+1\\+1\\+1\\-1\end{pmatrix}^{\times}\right]+$$
$$\frac{13}{80}\left[\begin{pmatrix}+1\\-1\\-1\\-1\end{pmatrix}^{\times}+\begin{pmatrix}+1\\+1\\-1\\+1\end{pmatrix}^{\times}+\begin{pmatrix}+1\\-1\\+1\\+1\end{pmatrix}^{\times}\right]+\frac{37}{80}\begin{pmatrix}+1\\+1\\+1\\+1\end{pmatrix}^{\times}$$

To apply the CG method we should use (5.13). Yet, the resulting matrix features a minimum eigenvalue of ≈ -0.019 and thus is not positive semidefinite although the considered correlation matrix is in S_4^{Ant}.

To further reinforce such evidence another example could be discussed, here we scan all the 4×4 correlation matrices whose out diagonal entries can be written as $\frac{w}{10}$ for $w = -10, \ldots, 10$ and find that 75480 of them are compatible with the antipodality constrain but cannot be obtained with the CG generator, i.e., the matrices that are in S_4^{Ant} when they are processed with (5.13) present at least one negative eigenvalue and so they do not belong to S_4^{Gau}. This shows that with $n = 4$ the inclusions (5.15) are strict as expected.

For higher values of n we proceed as follows. If $\mathcal{V}$ is the correlation matrix of $\boldsymbol{v} = (\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{n-1})^\top \in \mathbb{R}^n$, then $\mathcal{V}^{(n-1)}$ is the correlation of the subvector $(\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{n-2})^\top \in \mathbb{R}^{n-1}$. Hence, if the desired correlation profile is in one of the previously defined sets, i.e., if $\mathcal{V} \in S_n^*$ where $*$ is any of "NND," "Ant," or "Gau," then it must also be $\mathcal{V}^{(n-1)} \in S_{n-1}^*$. By reversing the implication we also get that if an $(n-1) \times (n-1)$ matrix exist not belonging to S_{n-1}^*, then there is a whole family of matrices not belonging to S_n^*.

This means that, if any of the inclusions $S_n^{\text{Gau}} \subseteq S_n^{\text{Ant}} \subseteq S_n^{\text{NND}}$ is strict for a certain $\hat{n}$ then it is strict also for any $n \geq \hat{n}$.

In the light of this and of the above examples, for all relevant dimensionalities ($n \geq 4$), the antipodality constraint limits the ability of synthesizing second-order statistical properties and the CG method is not completely general. When it fails the LUT method discussed in Sect. 5.3.2 is still able to correctly match the desired correlation profile so that for any possible correlation matrix associated with a process generating antipodal sequences at least a generation method able to reproduce such a profile exists.

In order to complete our analysis we discuss here cases where the process generating antipodal sequences is a stationary one. Its implies that we are looking for toeplitz non-negative definite matrices with the additional constraint to impose unit entries on the main diagonal. Having in mind this class of processes in the rest of this section we always refer to a matrix $\mathcal{V}$ defined as follows.

$$\mathcal{V} = \begin{pmatrix} 1 & a & b & c & \cdots & & & \\ a & 1 & a & b & c & \cdots & & \\ b & a & 1 & a & b & c & \cdots & \\ c & b & a & 1 & a & b & c & \cdots \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \cdots & c & b & a & 1 & a & b & c \\ & \cdots & c & b & a & 1 & a & b \\ & & \cdots & c & b & a & 1 & a \\ & & & \cdots & c & b & a & 1 \end{pmatrix}$$

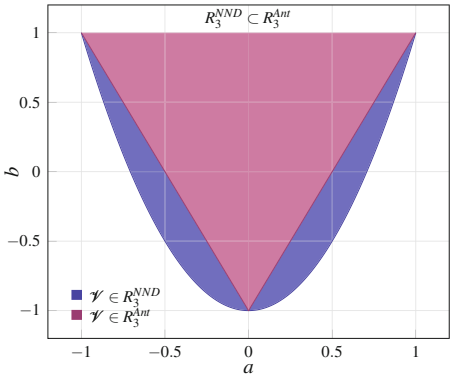


Fig. 5.12 Feasibility spaces of the two parameters in a correlation matrix of a stationary stochastic process with $n = 3$ where two different ensembles are considered: unit diagonal non-negative definite matrices $\mathscr{V} \in S_3^{\text{NND}}$, and unit diagonal non-negative definite matrices that can be a correlation profile associated with antipodal sequences $\mathscr{V} \in S_3^{\text{Ant}}$

With respect to the more general setting discussed before, here the degree of freedom in the design is limited to the free $n-1$ values over each diagonals named $\mathscr{V}_{(0)}, \ldots, \mathscr{V}_{(n-2)}$.

As before we first define the set of non-definite positive matrices with unit diagonal by (5.10) in order to obtain R_n^{NND} which represents the correlation matrices of a stationary process. After that we consider the impact of the antipodality constraint by (5.11) to obtain R_n^{Ant} and finally we look at R_n^{Gau} which is the set of the correlation profiles where GC method correctly works (5.14).

As before we have

- $R_1^{\text{NND}} = R_1^{\text{Ant}} = R_1^{\text{Gau}}$
- $R_2^{\text{NND}} = R_2^{\text{Ant}} = R_2^{\text{Gau}}$
- $R_3^{\text{NND}} \subset R_3^{\text{Ant}}$, and $R_3^{\text{Ant}} = R_3^{\text{Gau}}$

Where the last inclusion is shown in Fig. 5.12. With respect to the general setting, in the stationary cases, when $n = 4$ we have only 3 parameters and a complete analysis is reported in Fig. 5.13 showing that

$$R_4^{\text{NND}} \subset R_4^{\text{Ant}} \subset R_4^{\text{Gau}}$$

Furthermore, such a strict inclusions holds for every $n \geq 4$, i.e., for all relevant dimensionality. This is why CG method is not a general approach also for the stationary process cases while the LPF process discussed in Sect. 5.3.1 is able to generate any $\mathscr{V} \in R_n^{\text{Ant}}$.

As final remark, in terms of *complexity*, CG is a quite complex method. First, it requires the implementation of the $\sin(\cdot)$ function, that is not commonly available in the simplest architectures. Then, the solution provided by CG is defined only in terms of correlation matrix: an additional block capable of generating sequences with a prescribed correlation matrix is required. Note that this block has to be more complex than that depicted in Fig. 5.6 where a simple $H(f)$ filter and a quantization

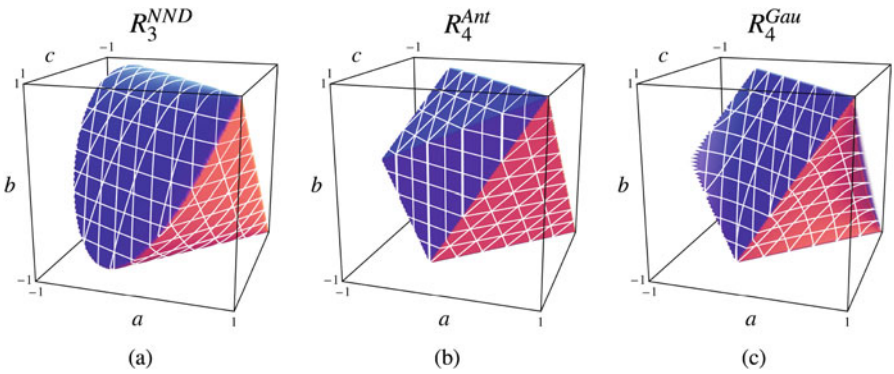


Fig. 5.13 Feasibility spaces in the three parameters composing a correlation matrix of a stationary process with $n = 4$. (**a**) is for unit diagonal non-negative definite toeplitz matrices $\mathscr{V} \in R_4^{\mathrm{NND}}$; (**b**) is for unit diagonal non-negative definite toeplitz matrices that can be a correlation profile associated with antipodal sequences $\mathscr{V} \in R_4^{\mathrm{Ant}}$; (**c**) is the feasibility space of correlation matrices associated with antipodal sequences generated by adopting the GC generator $\mathscr{V} \in R_4^{\mathrm{Gau}}$

block are present. Furthermore, the scheme of Fig. 5.6 can be considered only for the stationary case. For these reasons CG is a candidate for off-line generations. Conversely, LPF and LUT present very simple architectures, easily implementable both as software algorithms and with hardware primitives. In both approaches complexity can be further reduced depending on the desired performance so that they can be used for on-line generation.

5.4 Ternary and Binary Sensing Sequences

As already discussed in Chap. 4, the sensing matrix $\boldsymbol{A}$ plays an important role in the determination of the cost (either in terms of energy or in terms of number of basic operations) of the evaluation of the measurements vector $\boldsymbol{y}$. All solutions proposed in Chap. 4 for reducing this cost take into account the reduction of cardinality of $\boldsymbol{y}$, of the number of input samples used in the computation of $\boldsymbol{y}$, or of the number of arithmetic operations necessary to compute $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$. In particular, it has been shown how this optimization can be achieved by means of adopting sensing matrices whose rows are instances of ternary or binary processes with a prescribed second-order statistics.

In Sect. 5.3 we have discussed how to generate, given a prescribed second-order statistic, antipodal sensing vectors, i.e., $\boldsymbol{v} \in \{+1, -1\}^n$. We discuss here a generalization of the discussed CG method able to deal with the ternary, i.e., $\boldsymbol{v} \in \{+1, 0, -1\}^n$, and the binary case, i.e., $\boldsymbol{v} \in \{+1, 0\}^n$.

5.4.1 Ternary Sensing Sequences

The generation of ternary vectors $\boldsymbol{v} \in \{+1, 0, -1\}^n$ with a prescribed correlation matrix can be pursued by generalizing the methods proposed for antipodal vectors (see, e.g., [1, 3, 6, 7]).

In particular, we focus on the thresholding of Gaussian random vectors approach described in Sect. 5.3.1. As already observed, though not completely general, this is a very simple method and allows an almost equally simple generalization to the ternary case.

The generation of antipodal sequences $\boldsymbol{v} = (\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{n-1})^\top$ with a prescribed correlation matrix $\mathscr{V}$ is achieved by introducing an auxiliary random Gaussian vector $\boldsymbol{g} = (\boldsymbol{g}_0, \ldots, \boldsymbol{g}_{n-1})^\top$ generated with zero-mean, unit-variance, and a correlation matrix $\mathscr{G}$. By computing $\boldsymbol{v}$ componentwise from $\boldsymbol{g}$ according to

$$\boldsymbol{v}_j = \begin{cases} -1 & \text{if } \boldsymbol{g}_j \le 0 \\ +1 & \text{if } 0 < \boldsymbol{g}_j \end{cases} \tag{5.19}$$

we know that $\boldsymbol{v}$ has the desired correlation matrix if (5.2) holds, i.e.,

$$\mathscr{G} = \sin\left(\frac{\pi}{2}\frac{n}{\operatorname{tr}(\mathscr{V})}\mathscr{V}\right) \tag{5.20}$$

In order to generate $\boldsymbol{v}$ as a ternary vector, we follow the approach proposed in [5] and we introduce again, as in the antipodal case, the auxiliary random Gaussian vector $\boldsymbol{g}$. Then, (5.19) $\boldsymbol{v}$ has to be replaced by a three-level quantization function $\boldsymbol{v}_j = \tau^{\mathrm{t}}_{\theta_j}(\boldsymbol{g}_j)$, namely

$$\tau^{\mathrm{t}}_{\theta_j}(\boldsymbol{g}_j) = \begin{cases} -1 & \text{if } \boldsymbol{g}_j \le -\theta_j \\ 0 & \text{if } -\theta_j < \boldsymbol{g}_j \le \theta_j \\ +1 & \text{if } \theta_j < \boldsymbol{g}_j \end{cases} \tag{5.21}$$

where θ_j represents its symmetric threshold. Note that, for the maximum flexibility, it may be sensitive not to have a single threshold for the whole vector, but each entry $\boldsymbol{v}_j$ may be computed according to a different threshold θ_j.

Since we have changed the distortion function to get $\boldsymbol{v}$ from $\boldsymbol{g}$ with respect to the antipodal case, it is necessary to look also for a different distortion function to get $\mathscr{G}$ from $\mathscr{V}$. In other words, we need a replacement for equation (5.20). The calculations to switch from the desired $\mathscr{V}$ to $\mathscr{G}$ given the desired thresholds $\theta_0, \ldots \theta_{n-1}$ for the ternary case have been proposed in [5].

First of all, each $\boldsymbol{g}_j$ is a unit-variance Gaussian variable. The probability $\Pr\{|\boldsymbol{g}_j| \ge \theta_j\} = \operatorname{erfc}\left(\theta_j/\sqrt{2}\right)$, and from (5.21) we get $\Pr\{|\boldsymbol{g}_j| \ge \theta_j\} = \Pr\{\boldsymbol{v}_j^2 = 1\} = \mathscr{V}_{j,j}$. Hence we must set

$$\theta_j = \sqrt{2}\mathrm{erfc}^{-1}(\boldsymbol{\mathcal{V}}_{j,j}) \tag{5.22}$$

i.e., the thresholds $\theta_0, \ldots \theta_{n-1}$ are not a degree of freedom of the system, but are univocally determined by the main diagonal of the desired correlation matrix $\boldsymbol{\mathcal{V}}$.

Then, we recall that the element $\boldsymbol{\mathcal{V}}_{j,k}$ is the correlation between the elements $\boldsymbol{v}_j$ and $\boldsymbol{v}_k$, and it is defined as

$$\boldsymbol{\mathcal{V}}_{j,k} = \mathbf{E}\left[\boldsymbol{v}_j \boldsymbol{v}_k\right] = \mathbf{E}\left[\tau^{\mathrm{t}}_{\theta_j}(\boldsymbol{g}_j)\tau^{\mathrm{t}}_{\theta_k}(\boldsymbol{g}_k)\right] \tag{5.23}$$

where g_j and g_k, by definition, are zero-mean unit-variance jointly Gaussian random variables whose correlation is given by $\boldsymbol{\mathcal{G}}_{j,k}$ In the more general form, let us consider that if the correlation γ between two unit-variance jointly Gaussian random variable α and β is known, then their joint probability density is

$$f(\alpha, \beta, \gamma) = \frac{1}{2\pi\sqrt{1-\gamma^2}} e^{-\frac{\alpha^2+\beta^2-2\gamma\alpha\beta}{2(1-\gamma^2)}}$$

so that the correlation between $\tau^{\mathrm{T}}_{\theta'}(\alpha)$ and $\tau^{\mathrm{T}}_{\theta''}(\beta)$ is

$$\begin{aligned}
&\mathbf{E}\left[\tau^{\mathrm{t}}_{\theta'}(\alpha)\tau^{\mathrm{t}}_{\theta''}(\beta)\right] = \\
&= 2\int_{\theta'}^{\infty}\int_{\theta''}^{\infty} f(\alpha,\beta,\gamma)\mathrm{d}\alpha\mathrm{d}\beta - 2\int_{\theta'}^{\infty}\int_{-\infty}^{-\theta''} f(\alpha,\beta,\gamma)\mathrm{d}\alpha\mathrm{d}\beta = \\
&= \frac{1}{\sqrt{2\pi}}\int_{\theta''}^{\infty} e^{-\frac{\beta^2}{2}}\mathrm{erfc}\left(\frac{\theta'-\gamma\beta}{\sqrt{2(1-\gamma^2)}}\right)\mathrm{d}\beta + \\
&\qquad -\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{-\theta''} e^{-\frac{\beta^2}{2}}\mathrm{erfc}\left(\frac{\theta'-\gamma\beta}{\sqrt{2(1-\gamma^2)}}\right)\mathrm{d}\beta
\end{aligned} \tag{5.24}$$

where we have exploited the property $f(-\alpha, -\beta, \gamma) = f(\alpha, \beta, \gamma)$.

Pairing (5.24) and (5.23), and expressing thresholds accordingly to (5.22), we can obtain a function $T_{\eta',\eta''}$ such that

$$\boldsymbol{\mathcal{V}}_{j,k} = T_{\boldsymbol{\mathcal{V}}_{j,j},\boldsymbol{\mathcal{V}}_{k,k}}\left(\boldsymbol{\mathcal{G}}_{j,k}\right)$$

Note that this function depends not only on the correlation terms $\boldsymbol{\mathcal{G}}_{j,k}$ between the elements g_j and g_k, but also on the two thresholds θ_j and θ_k, that can be more conveniently expressed as a function of $\boldsymbol{\mathcal{V}}_{j,j}$ and $\boldsymbol{\mathcal{V}}_{k,k}$.

Regrettably, such a function cannot be given a fully analytical expression but has some recognizable properties. In particular, $T_{\eta',\eta''}(\gamma) = T_{\eta'',\eta'}(\gamma) = -T_{\eta',\eta''}(-\gamma)$ is continuous and monotonically increasing in γ, and can be extended by continuity in the domain $[-1, 1]$ with $T_{\eta',\eta''}(\pm 1) = \pm\min\{\eta', \eta''\}$. Moreover, coherently with [8], we have $T_{1,1}(\gamma) = \frac{2}{\pi}\sin^{-1}(\gamma)$.

The range of $T_{\eta',\eta''}$ is compatible with that of a correlation between two ternary variables. Hence, any desired matrix $\mathcal{V}$ can be transformed into a corresponding $\mathcal{G}$ by means of the $T^{-1}_{\mathcal{V}_{j,j},\mathcal{V}_{j,j}}$ defined as the inverse of $T_{\mathcal{V}_{j,j},\mathcal{V}_{k,k}}$. The replacement for (5.20) is given by

$$\mathcal{G}_{j,k} = T^{-1}_{\mathcal{V}_{j,j},\mathcal{V}_{k,k}}(\mathcal{V}_{j,k})$$

5.4.2 Binary Sensing Sequences

Binary random vectors $\boldsymbol{v} = (\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{n-1})^\top$ with a prescribed correlation matrix $\mathcal{V}$ can be generated following the very same approach used for the ternary case. Given an auxiliary random Gaussian vector $\boldsymbol{g} = (\boldsymbol{g}_0, \ldots, \boldsymbol{g}_{n-1})^\top$ generated with zero-mean, unit-variance, and a correlation matrix $\mathcal{G}$, we need now a two-level quantization function $\boldsymbol{v}_j = \tau^{\mathrm{b}}_{\theta_j}(\boldsymbol{g}_j)$ as

$$\tau^{\mathrm{b}}_{\theta_j}(\boldsymbol{g}_j) = \begin{cases} 0 & \text{if } \boldsymbol{g}_j < \theta_j \\ 1 & \text{if } \boldsymbol{g}_j \geq \theta_j \end{cases}$$

defined starting from a proper choice of the threshold levels $\theta_0, \ldots, \theta_{n-1}$.

As in the previous case, threshold θ_j is related to $\mathcal{V}_{j,j}$ by

$$\theta_j = \sqrt{2}\mathrm{erfc}^{-1}(2\mathcal{V}_{j,j})$$

while the element $\mathcal{V}_{j,k}$, defined as

$$\mathcal{V}_{j,k} = \mathbf{E}\left[\boldsymbol{v}_j \boldsymbol{v}_k\right] = \mathbf{E}\left[\tau^{\mathrm{t}}_{\theta_j}(\boldsymbol{g}_j)\tau^{\mathrm{t}}_{\theta_k}(\boldsymbol{g}_k)\right]$$

can be exploited by observing that if two jointly Gaussian zero-mean and unit-variance random variables α and β have correlation γ then

$$\begin{aligned} \mathbf{E}[\tau^{\mathrm{b}}_{\theta'}(\alpha)\tau^{\mathrm{b}}_{\theta''}(\beta)] &= \int_{\theta'}^{\infty}\int_{\theta''}^{\infty} f(\alpha,\beta,\gamma)d\alpha d\beta = \\ &= \frac{1}{2\sqrt{2\pi}}\int_{\theta''}^{\infty} e^{-\frac{\beta^2}{2}}\mathrm{erfc}\left(\frac{\theta' - \gamma\beta}{\sqrt{2(1-\gamma^2)}}\right)d\beta \end{aligned}$$

Following the same path that we used for ternary vectors, we obtain a function $B_{\eta',\eta''}$ that transforms the correlation of jointly Gaussian random variable in the correlation of the corresponding binarized random variable with assigned averages η' and η'' such that

$$\mathcal{V}_{j,k} = B_{\mathcal{V}_{j,j},\mathcal{V}_{k,k}}\left(\mathcal{G}_{j,k}\right)$$

This function has the same favorable properties as the function $T_{\eta',\eta''}$ of the ternary case. In particular, $B_{\eta',\eta''}(\gamma) = B_{\eta'',\eta'}(\gamma)$ is continuous and monotonically increasing in γ, and can be extended by continuity in the domain $[-1, 1]$ with $B_{\eta',\eta''}(-1) = \max\{0, \eta' + \eta'' - 1\}$ and $B_{\eta',\eta''}(1) = \min\{\eta', \eta''\}$.

Hence, any desired matrix $\mathscr{V}$ can be transformed into the corresponding $\mathscr{G}$ by defining $B^{-1}_{\eta',\eta''}(\cdot)$ is the inverse of $B_{\eta',\eta''}(\cdot)$, as

$$\mathscr{G}_{j,k} = B^{-1}_{\mathscr{V}_{j,j},\mathscr{V}_{k,k}}(\mathscr{V}_{j,k})$$

References

1. A. Caprara et al., Generation of antipodal random vectors with prescribed non-stationary 2-nd order statistics. IEEE Trans. Signal Process. **62**(6), 1603–1612 (2014)
2. D. Gangopadhyay et al., Compressed sensing analog front-end for bio-sensor applications. IEEE J. Solid State Circuits **49**(2), 426–438 (2014)
3. G. Jacovitti, A. Neri, G. Scarano, Texture synthesis-by-analysis with hard-limited Gaussian processes. IEEE Trans. Image Process. **7**(11), 1615–1621 (1998)
4. A. Lodi, K. Allemand, T.M. Liebling, An evolutionary heuristic for quadratic 0-1 programming. Eur. J. Oper. Res. **119**(3), 662–670 (1999)
5. M. Mangia et al., Rakeness-based design of low-complexity compressed sensing. IEEE Trans. Circuits Syst. I Regul. Pap. **64**(5) (2017)
6. R. Rovatti, G. Mazzini, G. Setti, Memory-m antipodal processes: spectral analysis and synthesis. IEEE Trans. Circuits Syst. I Regul. Pap. **56**(1), 156–167 (2009)
7. R. Rovatti et al., Linear probability feedback processes, in *2008 IEEE International Symposium on Circuits and Systems*, IEEE, May 2008, pp. 548–551
8. J.H. Van Vleck, D. Middleton, The spectrum of clipped noise. Proc. IEEE **54**(1), 2–19 (1966)

Chapter 6
Architectures for Compressed Sensing

The aim of this chapter is to move a first step into the hardware implementation of a compressed sensing (CS) based analog-to-information converter (AIC), providing a high-level overview of the different architectures and different solutions introduced so far in the scientific literature. For each architecture advantages and disadvantages will be analyzed, both from the architectures point of view and from the performance point of view. A more detailed analysis focusing on real circuits and including specific circuital solutions is presented in Chaps. 7 and 8.

6.1 Introduction and Definitions

In order to introduce this overview, it is necessary to briefly recall the introduction on signal acquisition systems of Chap. 1. In the conventional approach, depicted in Fig. 6.1a, samples are taken from an input signal $x(t)$ at a sufficiently high sampling rate $r_x = 1/T$ thus generating the sequence $x_k : \mathbb{Z} \mapsto \mathbb{R}$ such that $x_k = x(k\,T)$. Each sample is then quantized into the binary word $Q(x_k)$.

Two main differences can be identified in a CS based signal acquisition chain. The first can be observed in Fig. 6.1b. Differently from the standard sampling approach, in the CS some early additional processing is performed. The additional signal processing block may be added at different points in the chain and, accordingly, both the signal processing mathematical model and the position in which the subsufficient-rate sequence of measurements y_j appears may change.

The second difference is illustrated in Fig. 6.2. In the conventional approach, given a signal mathematically represented by the realization of a stochastic process in real domain $x(t) : \mathbb{R} \mapsto \mathbb{R}$ and a sampling rate r_x (or a sampling time step $T = 1/r_x$), it is possible to generate the sequence of real numbers $x_k : \mathbb{Z} \mapsto \mathbb{R}$ associated with $x(t)$ by defining $x_k = x(k\,T)$. Since $x(t)$ is defined for $\forall t \in \mathbb{R}$, it follows that x_k is defined $\forall k \in \mathbb{Z}$, i.e., the sequence of the x_k is *infinite-length*.

M. Mangia et al., *Adapted Compressed Sensing for Effective Hardware Implementations*, DOI 10.1007/978-3-319-61373-4_6

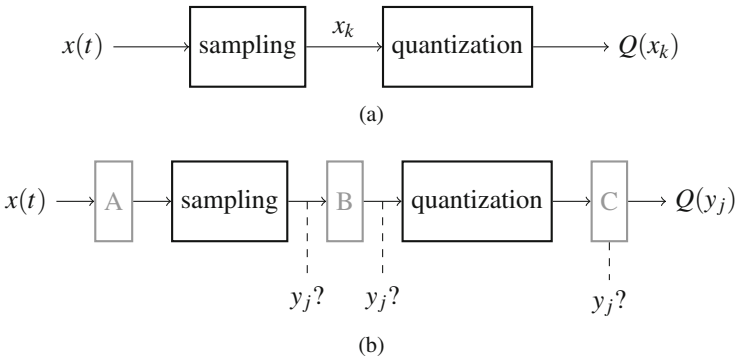


Fig. 6.1 (**a**) Basic two-stages decomposition of an acquisition process. (**b**) The acquisition signal chain modified accordingly to the CS paradigm

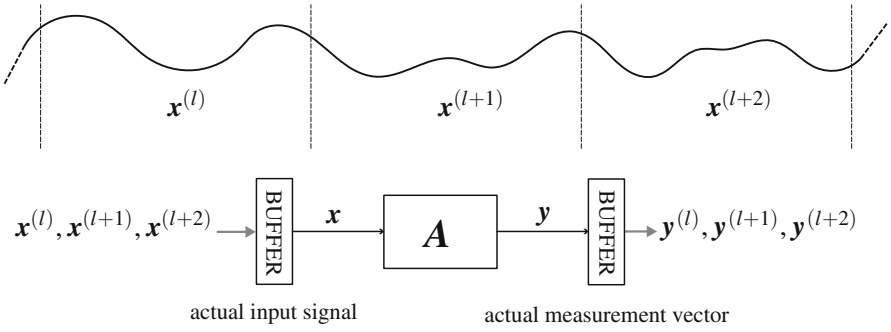


Fig. 6.2 Time window slicing approach used in the CS paradigm

In CS, in order to avoid dealing with infinite-dimension optimization problems, it is mandatory for signals to be defined in a finite-dimension space. In practical cases, the more common approach is to assume that $x(t)$ is defined only over a time windows of width T_w, i.e., only for $0 \leq t < T_w$. Given $n \in \mathbb{N}$ with $T_w = n\,T$, the Nyquist-rate sampling process maps $x(t)$ to the finite-length sequence $x_0, x_1, \ldots, x_{n-1}$ with $x_k = x(k\,T)$, or more conveniently to the sample vector $\boldsymbol{x} \in \mathbb{R}^n$, where in the following we will use the notation $\boldsymbol{x}_k$ to refer to k-th element of $\boldsymbol{x}$. Under this assumption all the developed CS theory can be applied to $\boldsymbol{x}$, i.e., m measurements are generated accordingly to an $m \times n$ sensing matrix $\boldsymbol{A}$ by means of the linear operation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, with $\boldsymbol{y} = (\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{m-1})^\top \in \mathbb{R}^m$ the vector composed by the m compressed measurements. Each measurement can also be individually expressed as $\boldsymbol{y}_j = \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k}\,\boldsymbol{x}_k$ (where $\boldsymbol{A}_{j,k}$ is the element of $\boldsymbol{A}$ positioned at the j-th row and k-th column) or $\boldsymbol{y}_j = \boldsymbol{A}_{j,\cdot}\,\boldsymbol{x}$ (where $\boldsymbol{A}_{j,\cdot}$ is the sensing vector given by the j-th row of $\boldsymbol{A}$).

Of course, real-world signals are unavoidably defined for $t \in \mathbb{R}$. To apply CS, it is necessary to slice any real signal $x(t)$ in adjacent windows of width T_w such that $x^{(l)}(t) = x(l\,T_w + t)$, with $0 \leq t < T_w$ and $l \in \mathbb{Z}$. The n samples obtained at rate $r_x = 1/T$ from each slice $x^{(l)}(t)$ are collected in a vector $\boldsymbol{x}^{(l)} \in \mathbb{R}^n$ and with them it is possible to generate, by means of the sensing matrix[1] $\boldsymbol{A}$, the measurement vector $\boldsymbol{y}^{(l)}$. This strategy is illustrated in Fig. 6.2.

In the following, in order to keep the mathematical notation as simple as possible, we will focus on a single slice of signal implicitly assuming that the signal slicing process is made *a priori*. In other words, in all this chapter the signal $x(t)$ is modeled as the realization of a continuous-time stochastic process defined only for $0 \leq t < T_w$, that can be sampled at a rate $r_x = 1/T$ with $T_w = n\,T$, giving rise to the sampling vector $\boldsymbol{x} = (x(0), x(T), \ldots, x(n\,T - T))^\top$ or, with a more compact notation, $\boldsymbol{x} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1})^\top$.

6.2 The CS Signal Acquisition Chain

From a formal point of view, the scheme of Fig. 6.1b identifies three classes of processing chain accordingly to the position where the additional linear computational block is added, as detailed in Fig. 6.3. These three classes, referred to as *case A*, *case B*, and *case C*, are detailed in the following.

In particular, the aim of this section is to give a mathematical background with which the three cases can be handled within the general framework considered up to now, where measurements are given by the matrix relation $\boldsymbol{y} = \boldsymbol{A}\,\boldsymbol{x}$, with $\boldsymbol{x} \in \mathbb{R}^n$ the vector of Nyquist-rate samples of the input signal, $\boldsymbol{y} \in \mathbb{R}^m$ the vector of the compressed measurements, and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ the sensing matrix. Since the m measurements are usually obtained by *replicating* an identical structure m times using different sensing vectors $\boldsymbol{A}_{j,\cdot}$, we will actually focus on the computation of a single measurement $\boldsymbol{y}_j = \boldsymbol{A}_{j,\cdot}\,\boldsymbol{x}$.

- ***Case A*** The traditional processing chain is modified by adding a linear computational block at its beginning, taking $x(t)$ as input. Since $x(t)$ is mathematically an analog continuous-time process, the additional computational block needs to be a *continuous-time analog* one. This block has to be modeled as a linear sensing functional operator $\mathscr{A}_j\{\cdot\}$, taking the continuous-time function $x(t)$ as input, and producing the continuous-time function $y_j(t)$ as output

$$y_j(t) = \mathscr{A}_j\{x\}\,(t)\,.$$

[1] In principle, it is possible to process each $\boldsymbol{x}^{(l)}$ vector with a different sensing matrix $\boldsymbol{A}^{(l)}$. This case is not considered here only for the sake of simplicity, and only one sensing matrix $\boldsymbol{A}$ is taken $\forall l \in \mathbb{Z}$.

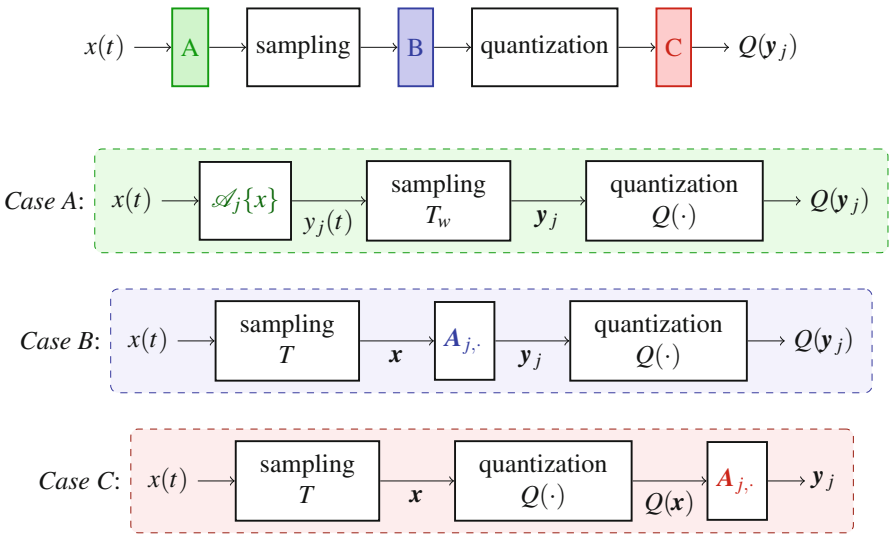


Fig. 6.3 Three classes of processing chain adopted by CS compared to the standard acquisition process. In *case A* a linear computational block is added before the sampling stage. In *case B* it is inserted between the sampling and the quantization stages. In *case C* it is added after the quantization stage

Then, the subsequent sampling stage generates the compressed measurement $\boldsymbol{y}_j$ by sampling the $y_j(t)$ function at the time T_w

$$\boldsymbol{y}_j = y_j(T_w) = \mathscr{A}_j\{x\}\,(T_w)\,.$$

Assuming that m is the number of operators applied in parallel, then m measurements $\boldsymbol{y}_j = \mathscr{A}_j\{x\}\,(T_w), j = 0, 1, \ldots, m-1$ are simultaneously collected at time T_w to generate the measurement vector $\boldsymbol{y}$. The measurement rate is actually $r_y = m/T_w$. Under the assumption that, when using a standard approach, the input signal could be sampled at Nyquist rate $r_x = 1/T$ and that $T_w = n\,T$, it is immediate to get the desired relation $r_x/r_y = n/m$.

Clearly, this mathematical approach is pretty general and allows a lot of degrees of freedom, but it is also quite difficult to handle it within the general CS framework considered up to now. In particular, two aspects suggest to introduce a more simple and practical modeling for this case. (*i*) The hardware realization of a generic $\mathscr{A}_j\{\cdot\}$ operator may be complex. In particular, and remembering that the $\mathscr{A}_j\{\cdot\}$ operators should be actually programmable ones as a necessary condition to get different $\boldsymbol{y}_j$ measurements, we should limit ourselves to consider operators that can be easily identified by a (possibly small) number of coefficients stored into a digital memory. (*ii*) It is mandatory to have a clear and simple relation between $\mathscr{A}_j\{\cdot\}$ and the sensing vector $\boldsymbol{A}_{j,\cdot}$ in order to apply all the theoretical background developed so far.

On the basis of the aforementioned observations, we can limit ourselves to consider $\mathscr{A}_j\{\cdot\}$ composed by a mixing stage (i.e., the analog multiplication) with a pulse-amplitude modulated (PAM) sensing signal $a_j(t)$ followed by an integrator. Mathematically we can express this either in the form of a convolution or multiply-and-integrate operation

$$y_j(t) = \frac{1}{T}\int_0^{T_w} a_j(t-\tau)x(\tau)\,\mathrm{d}\tau \tag{6.1}$$

$$y_j(t) = \frac{1}{T}\int_0^{t} a_j(\tau)x(\tau)\,\mathrm{d}\tau \tag{6.2}$$

where the constant $1/T$ is used for dimensionality purpose only ($\mathrm{d}\tau$ has the physical dimension of a time), and where in both cases the integration interval has been reduced to keep into account the fact that $x(t)$, accordingly to the aforementioned slicing approach, is defined only for $0 \le t < T_w$.

The sensing function $a_j(t)$ is a PAM signal with symbol period $T = T_w/n$ and where the amplitude of pulses are stored in the sensing vector $\boldsymbol{A}_{j,\cdot}$.

$$a_j(t) = \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k}\, g\left(\frac{t}{T} - k\right) \tag{6.3}$$

being $g(t)$ a normalized pulse. This modeling approach clearly solves the two issues identified above.

Among the convolution form expressed by (6.1) and the multiply-and-integrate one expressed by (6.2), the latter is the more commonly considered in CS theory. For this reason, and despite the fact that the two notations are almost identical and could be easily integrated in a single framework, we will limit ourselves to the second one. The actual *case A* considered can be represented by the diagram of Fig. 6.4.

Now, by combining (6.2) with (6.3), it is easy to get

$$\begin{aligned}\boldsymbol{y}_j = y_j(T_w) &= \frac{1}{T}\int_0^{T_w} \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k}\, g\left(\frac{\tau}{T} - k\right) x(\tau)\,\mathrm{d}\tau \\ &= \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k} \int_0^{T_w} \frac{1}{T}\, g\left(\frac{\tau}{T} - k\right) x(\tau)\,\mathrm{d}\tau. \end{aligned} \tag{6.4}$$

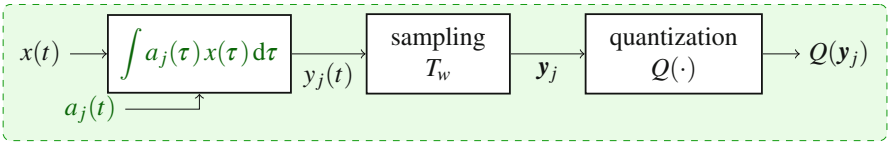


Fig. 6.4 Actual *case A* considered in this chapter

By defining

$$\tilde{\boldsymbol{x}}_k = \frac{1}{T}\int_0^{T_w} g\left(\frac{\tau}{T} - k\right) x(\tau)\,\mathrm{d}\tau \tag{6.5}$$

as the generalized Nyquist-rate samples[2] of the input signal then (6.4) can be rewritten in the desired form

$$\boldsymbol{y}_j = \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k}\,\tilde{\boldsymbol{x}}_k = \boldsymbol{A}_{j,\cdot}\,\tilde{\boldsymbol{x}}$$

where the generalized Nyquist-rate samples vector $\tilde{\boldsymbol{x}} = (\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_{n-1})^\top \in \mathbb{R}^n$ plays the role of the standard Nyquist-rate samples vector $\boldsymbol{x}$. Measurements are finally converted into digital words to be processed by the following (digital) processing stage.

In conclusion, this case can be included in the general CS framework by replacing the sampling vector $\boldsymbol{x}$ with the generalized sampling vector $\tilde{\boldsymbol{x}}$ given by (6.5).

Yet, two comments are mandatory.

First of all, even if it has been possible with some assumptions and with a specific mathematical background to include this case in the general CS framework, it is also possible to argue that by feeding $\boldsymbol{A}$ and $\boldsymbol{y}$ in any CS reconstruction algorithm, we would get the $\tilde{\boldsymbol{x}}$ instead of the $\boldsymbol{x}$. So, in this considered case, it is not possible to reconstruct the actual $x(t)$ signal.

However, as a second comment, we can argue that in many practical cases, the $\tilde{\boldsymbol{x}}$ is not so different from $\boldsymbol{x}$. Equation (6.5) represents the true Nyquist-rate sampling of $x(t)$, i.e., $\tilde{\boldsymbol{x}} = \boldsymbol{x}$, when $g(t) = T\,\delta(t)$ is given by the standard Dirac delta operator. In a more practical case, being $\delta(t)$ just a mathematical abstraction with poor practical implementation possibilities, it is common to consider a normalized pulse $g(t)$ equal to the ideal rectangular pulse $\chi(t) = 1$ when $0 \leq t < 1$ and 0 elsewhere. In this case it is easy to note from (6.5) that the generalized coefficient vector can be considered a good approximation of Nyquist-rate sample vector, i.e., $\tilde{\boldsymbol{x}} \approx \boldsymbol{x}$ if $x(t)$ is the realization of a *quasi-stationary* stochastic process.

From a hardware point of view, this case has raised some interest in the field of high-frequency applications, such as radio-frequency (RF) receivers [3, 4] or radar receivers [16, 17]. The main reason is that accurately sampling a signal at a very high rate is much more difficult with respect to accurately mixing it with

[2]We adopt this name since, as observed in the following, it is possible to generate actual Nyquist-rate samples $\boldsymbol{x}_k$ as a particular case of (6.5) when the $g(\cdot)$ is a Dirac delta operator.

another signal, even if at very high frequencies (the PAM signal $x(t)$ must have a update time T equal to the Nyquist frequency of $x(t)$) [17]. In Chap. 7 a few integrated circuits implementing this case will be described.

- ***Case B*** A linear computational block is inserted between the sampling and the quantization stages, taking as input the sampling vector $\boldsymbol{x} \in \mathbb{R}^n$ containing the samples of $x(t)$, $0 < t < nT$ at Nyquist rate $r_x = 1/T$, and generating the compressed measurement $\boldsymbol{y}_j$. This additional stage is a *discrete-time analog* processing block.

  Being linear, this block performs a weighted sum of all samples in $\boldsymbol{x}$. By assuming that the coefficients are the real quantities stored in the j-th sensing vector $\boldsymbol{A}_{j,\cdot}$, we can mathematically describe this operator as

$$\boldsymbol{y}_j = \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k}\, \boldsymbol{x}_k = \boldsymbol{A}_{j,\cdot}\, \boldsymbol{x}$$

  that is directly the desired relation.

  Note that, assuming again that m operators similar to the described one are applied in parallel, then m measurements are delivered each nT time units. The measurement rate can be expressed as $r_y = m/n/T$, with a compression ratio equal to $r_x/r_y = n/m$. As in the previous case, measurements are finally quantized to deal with the following allegedly digital world.

  Note also that this case, conversely from the previous one, represents the direct implementation of the standard CS framework developed so far without the need of any approximation. No additional hypotheses need to be assumed, nor specific mathematical framework developed in order to get $\boldsymbol{x}$ from any CS reconstruction algorithm given $\boldsymbol{A}$ and $\boldsymbol{y}$.

  Form a hardware point of view, this case finds application when $x(t)$ is low-frequency and it is particularly easy and unexpensive (form energetic point of view) to accurately sample it. Commonly, this case is implemented by exploiting intrinsic sampling capabilities of the switched-capacitor architecture [6, 12, 14]. A few integrated circuits implementing this case by means of switched-capacitor implementation will be detailed in Chap. 7.

- ***Case C*** The (linear) computational block is added at the end of the original processing stage, taking the quantized values $Q(\boldsymbol{x})$ of the input signal samples vector $\boldsymbol{x}$, where the quantization function $Q(\cdot)$ applied to $\boldsymbol{x}$ has to be considered element-wise.

  This case is very similar to the previously considered one. We can assume that coefficients used in the weighted sum are stored in the j-th sensing vector $\boldsymbol{A}_{j,\cdot}$, and this mathematically leads to

$$\boldsymbol{y}_j = \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k} Q(\boldsymbol{x}_k) = \boldsymbol{A}_{j,\cdot} Q(\boldsymbol{x}) \tag{6.6}$$

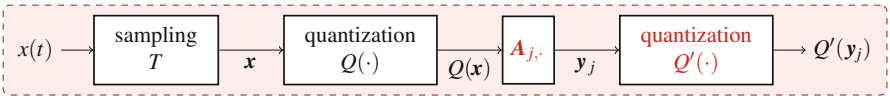


Fig. 6.5 Actual *case C* considered in this chapter

Furthermore, assuming m operators working in parallel, the measurement rate can be expressed as $r_y = m/n/T$, with $r_x/r_y = n/m$.

This case is of particular interest when also the $\boldsymbol{A}_{j,k}$ values belong to a quantized set, so that this additional computational block takes the form of a *digital postprocessing stage*, i.e., a digital algorithm. In other words, relation (6.6) does not require any particular hardware, and can be implemented on any finite-state machine equipped with a proper arithmetic and logic unit (ALU). In the following, we always implicitly make this assumption when considering this case.

Note also that in this case the $\boldsymbol{y}$ is already composed by digital quantities that can be delivered *as is* to the following digital reconstruction algorithm. However, it is a common practice to apply an additional re-quantization function (such as the $Q'(\cdot)$ in Fig. 6.5) for the purpose of reducing the bitrate required to transmit measurements. The *case C* of Fig. 6.3 is more adequately described by the processing chain of Fig. 6.5 [2].

The first two aforementioned cases will be referred to as *analog CS*, being $\boldsymbol{x}$ (or $\tilde{\boldsymbol{x}}$) made of analog quantities, and will be detailed in Chap. 7. In particular, *case A* identifies continuous-time analog CS systems, while *case B* discrete-time CS systems. Conversely, we refer to the third case as *digital CS*, that will be detailed in Chap. 8.

In the rest of this chapter we take into account some overall considerations that can be applied both to analog CS architectures and to the digital ones. The aim is twofold. On the one side, we clearly identify a few CS different hardware families accordingly to topological properties of the sensing matrix $\boldsymbol{A}$. The structure of $\boldsymbol{A}$, in fact, is directly mapped into the hardware complexity needed to compute $\boldsymbol{y}$.

As an example, in order to reduce hardware complexity both in the the analog and in the digital case, the multiplication with the $\boldsymbol{A}_{j,k}$ coefficients may be relaxed by representing the $\boldsymbol{A}_{j,k}$ as digital quantities using a very limited number of bits. This reduces the complexity of the digital-to-analog converter (DAC) used for the $\boldsymbol{A}_{j,k}$ generation in the analog case [6] as well as the complexity of the required ALU in the digital case. As an extreme case, it is also possible to require that $\boldsymbol{A}_{j,k}$ are 1-bit quantities, i.e., that $\boldsymbol{A}_{j,k} \in \{-1, 1\}$ or $\boldsymbol{A}_{j,k} \in \{0, 1\}$. Here, a full multiplier block (either an analog or a digital one) is not necessary anymore, and can be replaced, respectively, by a sign inversion block [3, 12] or by an enable/disable block [14].

On the other side, CS performance is strongly dependent on $\boldsymbol{A}$. Typically, matrices $\boldsymbol{A}$ presenting a structure that allows easy hardware implementation are mapped

into CS systems with poor performance either in terms of signal reconstruction quality or flexibility, defined as the property of a CS acquisition system to correctly work with many different input signal classes.

Accordingly to this, the choice of $\boldsymbol{A}$ represents a trade-off between hardware complexity and CS performance.

6.3 Architectures and Implementation Guidelines

Mainly, three architectures can be identified in the recent CS literature, namely the *Random Sampling* (RS), *Random Demodulator* (RD), and *Random Modulation Pre-Integration* (RMPI). In this section we not only provide an overview of these three approaches that require very different implementation efforts, but also achieve different performances.

In particular, we are interested in two aspects:

- how the differences in the described architectures are transposed in the sensing matrices $\boldsymbol{A}$ or,
- how CS performance changes accordingly to the considered architecture.

In order to estimate system performance, we use the same setup considered since Chap. 2. In detail, we run several Montecarlo simulations, where we assume that $\boldsymbol{x}$ is an instance of an n-dimensional stationary a stochastic process with $n = 128$, that is $\kappa = 6$ sparse with respect to an orthonormal basis $\boldsymbol{D}$, and (for the sake of simplicity) that has no localization, i.e., $\mathscr{L}_x = 0$.

With respect to $\boldsymbol{D}$, three cases are taken into account. In the first one, $\boldsymbol{D}$ is the orthonormal Discrete Cosine Transform (DCT) [1], commonly used in compression of digital images. In the second cases sparsity is considered on a wavelet basis, more in detail $\boldsymbol{D}$ is taken as the 4-th order Daubechies family [5] (Daub), that is a common choice in almost all biological signal processing systems. Finally the $n \times n$ identity basis $\boldsymbol{D} = \boldsymbol{I}_n$ is considered.

As observed in Chap. 1, many CS properties are based on the assumption of incoherence between $\boldsymbol{A}$ and $\boldsymbol{D}$. Due to the different $\boldsymbol{D}$, we expect different performance depending on the combination of the sparsity basis and of the architecture used.

At the end of the section we will also introduce a hybrid RD-RMPI architecture, that is commonly used in many practical cases to reduce hardware complexity with respect to the RMPI approach, with performance similar to the RMPI and much higher with respect to the RD.

6.3.1 Random Sampling

In standard acquisition systems, samples of the signal are taken regularly on the time axis at a given rate (usually not less than the Nyquist one). AICs relying on

RS avoid this regularity to produce a number m of randomly spaced measurements that, on the average, are less than those produced by Nyquist sampling, while still allowing the reconstruction of the whole signal, thanks to sparsity and other priors. The approach is actually the very same used since many years when looking for statistical properties of a very large population set.

In more general terms, m sampling instants τ_j, $j = 0, 1, \ldots, m-1$ are defined anywhere along the time axis, so that the j-th measurements are given by

$$\boldsymbol{y}_j = \int_0^{T_w} \delta(t - \tau_j) x(t) dt.$$

Yet, any straightforward implementation will choose the τ_j among regularly spaced time points, thus allowing to select them by digital quantities. In this case, a random sampling approach can be considered as consisting in taking only a random subset of size m among the n samples of the original signals.

From a hardware point of view, this is the most simple architecture one can use for CS, as is enough to properly modulate the clock of a standard analog-to-digital converter (ADC) to implement described operations, as illustrated in Fig. 6.6. The sensing matrix can be achieved by simply considering a sparse $\boldsymbol{A}$, where elements equal to 1 are present, one in each row, in all columns corresponding to a position in which sampling takes place. An example of $\boldsymbol{A}$ is depicted in Fig. 6.7. Such a matrix has the following properties:

$$\begin{cases} \boldsymbol{A}_{j,k} \in \{0, 1\}, & \forall j, \forall k \\ \|\boldsymbol{A}_{j,\cdot}\|_0 = 1, & \forall j \\ \|\boldsymbol{A}_{\cdot,k}\|_0 \leq 1, & \forall k \end{cases}$$

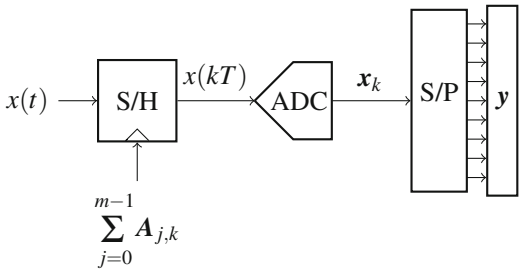


Fig. 6.6 Block scheme of the random sampling architecture

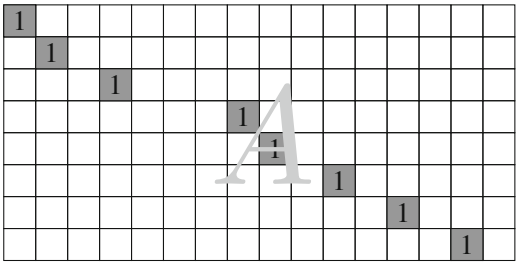


Fig. 6.7 Example of an 8×16 sensing matrix $\boldsymbol{A}$ corresponding to the implementation of a RS architecture. *White blocks* correspond to zero elements

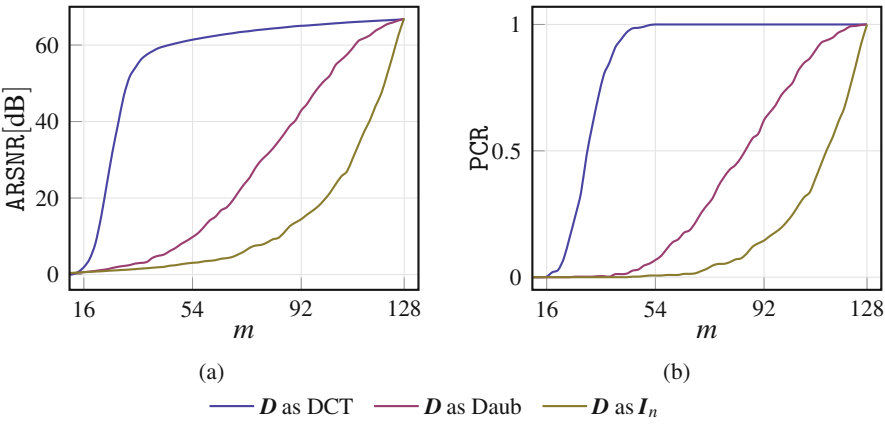


Fig. 6.8 Montecarlo comparison between performance of a RS system with $n = 128$, $\kappa = 6$ in terms of signal reconstruction quality—plot (**a**)—and probability of correct reconstruction—plot (**b**)—as a function of the number of measurement m. The considered sparsity bases are the Discrete Cosine Transform (DCT), the Daubechies-4 Wavelet (Daub), and the $n \times n$ identity matrix ($\boldsymbol{I}_n$)

i.e., there is one non-null element in each row, and at most one non-null element in each column. Sampling events take place at each time instant associated with a non-null column of $\boldsymbol{A}$, as indicated in Fig. 6.6.

Note that, according to this definition, random sampling belongs at the same time to all of the three cases defined in Sect. 6.2.

Such a hardware simplicity is, however, counterbalanced by low performance in terms of signal quality reconstruction. When looking for incoherence between $\boldsymbol{D}$ and $\boldsymbol{A}$, and given the structure of $\boldsymbol{A}$ as in Fig. 6.7, it is reasonable to assume that good performance is achieved only in the DCT case, since all DCT basis elements have a very large support. Conversely, in both the Wavelet case and the identity matrix case, the coherence between $\boldsymbol{A}$ and $\boldsymbol{D}$ is higher, and reconstruction performance is expected to be very poor. Results are shown in Fig. 6.8, and confirm this intuition.

In conclusion, RS is a very simple approach, but can ensure good quality only for particular signals, mainly those sparse on a Fourier (or similar) basis. In practical approaches, RS finds application only in the sampling of very high-frequency sinusoidal tones [15]. For this lack of generality, it will not be considered in the rest of this manuscript.

6.3.2 *Random Demodulator*

In the computation of $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, and assuming that $\boldsymbol{A}$ has not a particular structure, m parallel hardware blocks are necessary, each one computing the each measurements by means of an integrator or an adder depending on the case identified by Sect. 6.2.

The architecture known as RD avoids this replication. In other words, a single adder (or integrator) is used to compute all measurements by designing $\boldsymbol{A}$ such that

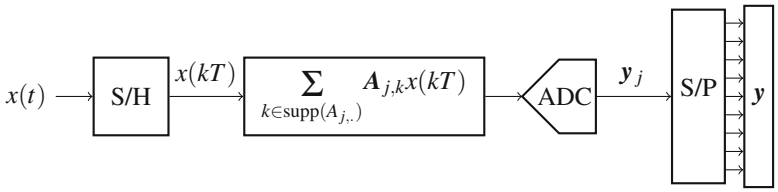


Fig. 6.9 Block scheme of the Random Demodulator architecture in the analog discrete-time case

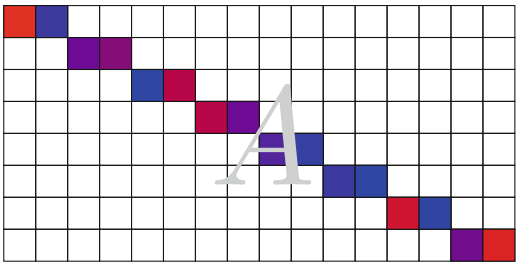


Fig. 6.10 Example of an 8×16 sensing matrix $\boldsymbol{A}$ corresponding to the implementation of a RD architecture. *White blocks* correspond to zero elements

the support of the j-th row has no intersections with the support of all other rows. This architecture, assuming an analog discrete-time implementation, is depicted in Fig. 6.9, while an example of a sensing matrix $\boldsymbol{A}$ allowing this simplified structure is depicted in Fig. 6.10.

Roughly speaking, and allowing $\boldsymbol{A}_{j,k} \in \mathbb{R}$ to consider the general case, each column has only one non-null element, while more than one non-null elements in a single row are allowed. The number of nonzero elements of the j-th row has been defined in Chap. 4 as N_j, so that also here we use $N_j = \|\boldsymbol{A}_{j,\cdot}\|_0$. Particularly interesting is the case where the system is *symmetric*, i.e., all rows have the same number of nonzero elements $N_j = N$, $\forall j$. In this case it is clearly $mN = n$. Otherwise, it is $\sum_{j=0}^{m-1} N_j = n$.

Note, however that, for the sake of exactness, we should consider that the $\boldsymbol{A}_{j,k}$ identified up to now as non-null entries of $\boldsymbol{A}$ are actually random variables drawn accordingly to a given reference distribution. If zero is an acceptable value for this distribution, then there is a probability that an expected non-null $\boldsymbol{A}_{j,k}$ may be actually equal to zero. To cope with this, we redefine the N_j constant and the $\|\cdot\|_0$ operator, considering that $\boldsymbol{A}_{j,k}$ belongs to the support of $\boldsymbol{A}$ (and so it is counted in the computation of the $\|\cdot\|_0$ norm) if, given all possible $\boldsymbol{A}$ instances, at least in some of them it is $\boldsymbol{A}_{j,k} \neq 0$. With this, we can formally write conditions for a sensing matrix $\boldsymbol{A}$ in order to be implemented on a RD system as

$$\begin{cases} \sum_{j=0}^{m-1} \|\boldsymbol{A}_{j,\cdot}\|_0 = n, & \forall j \\ \|\boldsymbol{A}_{\cdot,k}\|_0 = 1, & \forall k \end{cases}.$$

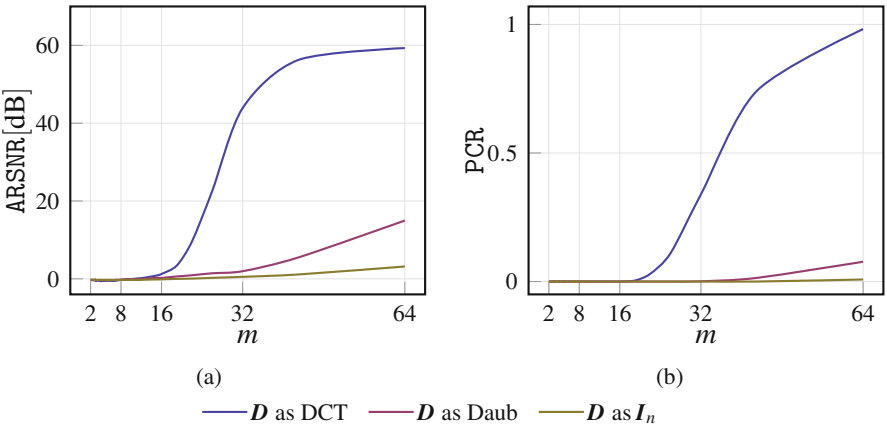


Fig. 6.11 Montecarlo comparison between performance of a RD system with $n = 128$, $\kappa = 6$ in terms of signal reconstruction quality—plot (**a**)—and probability of correct reconstruction—plot (**b**)—as a function of the number of measurement m. The considered sparsity bases are the Discrete Cosine Transform (DCT), the Daubechies-4 Wavelet (Daub), and the $n \times n$ identity matrix ($\boldsymbol{I}_n$)

Due to the lack of any parallel structure, however, performance is very similar to that of the RS. Simulation results for the same setting considered in the previous subsection (i.e., $n = 128$ and $\kappa = 6$) in terms of the number m of measurements are plotted in Fig. 6.11. The RD system is considered symmetric whenever possible, i.e., $N = n/m$ if n is an integer multiple of m. Otherwise, N_j for $j = 1, \ldots, m-2$ is the smallest integer larger than n/m, i.e., $N_j = \lceil n/m \rceil$, while N_{m-1} is smaller than the N_j and computed to satisfy the constraint $\sum_{j=0}^{m-1} N_j = n$.

Acceptable performance is achieved only in the DCT case, while in both the Daub and the identity matrix cases performance is very poor. Note also that, in order to have a sufficient number of measurements m, it is necessary that N is small. This is more evident when plotting the same performance curves as a function of N, as in Fig. 6.12. Only for very low values such as $N = 2$ or $N = 3$ the input signal can be effectively reconstructed. Note that in this plot, N has to be considered equal to $N = n/m$ if n is an integer multiple of m, and $N = \lceil n/m \rceil$ otherwise.

6.3.3 *Random Modulator Pre-Integration*

Let us assume that no constraints are posed on the $\boldsymbol{A}$, i.e.,

$$\begin{cases} \|\boldsymbol{A}_{j,\cdot}\|_0 = n, & \forall j \\ \|\boldsymbol{A}_{\cdot,k}\|_0 = m, & \forall k \end{cases}$$

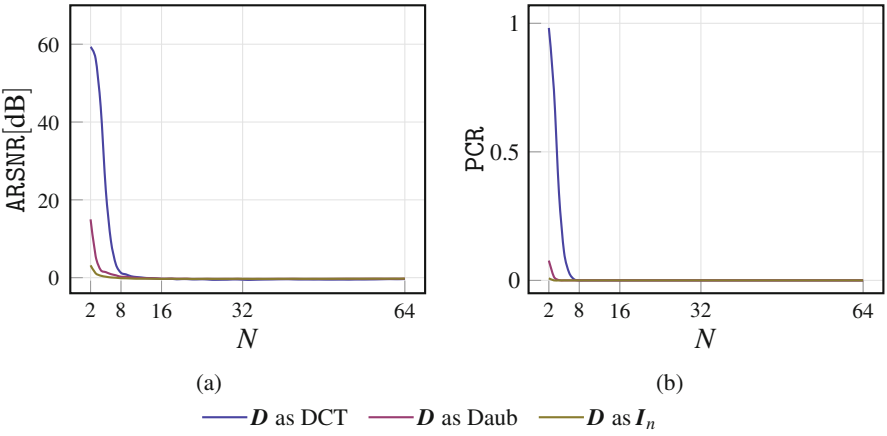


Fig. 6.12 Montecarlo comparison between performance of a RD system with $n = 128$, $\kappa = 6$ in terms of signal reconstruction quality—plot (**a**)—and probability of correct reconstruction—plot (**b**)—as a function of N. The considered sparsity bases are the Discrete Cosine Transform (DCT), the Daubechies-4 Wavelet (Daub), and the $n \times n$ identity matrix ($\boldsymbol{I}_n$)

where the $\|\cdot\|_0$ norm has to be intended in the extended way considered in the previous subsection. The hardware architecture capable of implementing this matrix is the RMPI, made of m parallel paths, each one with an integrator (or an adder) computing a single measurement $\boldsymbol{y}_j$. An RMPI architecture can be implemented in many different ways. An example for the analog discrete-time case is depicted in Fig. 6.13, highlighting the presence of m identical parallel paths. Of course, real implemented architectures could be slightly different. For example, blocks such as the sample/hold or the ADC can be multiplexed and shared [12].

Clearly, since this structure allows any possible sensing matrix $\boldsymbol{A}$ (an example of which is depicted in Fig. 6.14), it has optimal performance but at the cost of increasing hardware complexity (in terms, for example, of both area and power consumption) with m due to the parallel nature of the approach.

The plots of RMPI performance in terms of both signal reconstruction quality and probability of correct reconstruction can be found in Fig. 6.15. Results are clearly independent of the sparsity basis $\boldsymbol{D}$, and can be considered satisfactory starting from low values of m. Almost all of the AIC prototypes presented in the literature are based on an RMPI architecture [6, 12].

6.3.4 *Hybrid RD-RMPI Architecture*

The RMPI architecture has proven to be the most reliable one in terms of flexibility with respect to the input signal. Furthermore when considering biomedical signals, since it is known that they are sparse with respect to a Wavelet [5] or a Gabor [13] basis, this approach is the only one allowing a correct reconstruction.

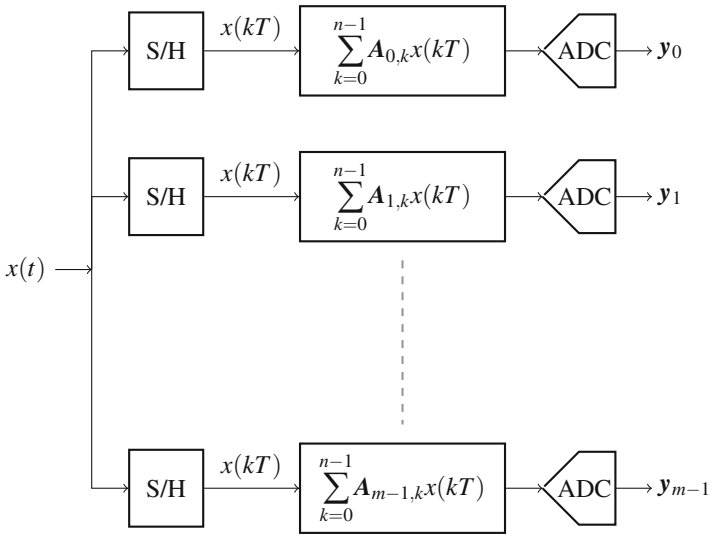


Fig. 6.13 Block scheme of the Random Modulator Pre-Integration architecture, highlighting the m parallel paths structure

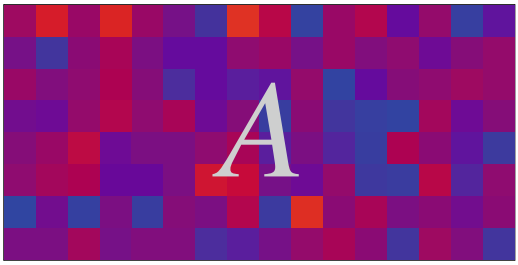


Fig. 6.14 Example of an 8×16 sensing matrix $\boldsymbol{A}$ corresponding to the implementation of a Random Modulator Pre-Integration architecture

However, when dealing with a hardware implemented system, embedding a large number of parallel channels may be an issue, for example, in terms of area or power consumption [14], or even due to clock distribution issues in high speed circuits [16]. In conclusion, and with particular reference to analog implementations, the number of parallel paths that can be implemented (and consequently, the number of measurements that can be computed at the same time) is limited.

The problem can be solved by using an approach that is actually hybrid between the RD and the RMPI ones. Let us assume that only a small number M of measurements can be computed at the same time, and that these measurements are computed by using only the first N samples of the actual time windows. If this is the case, by taking other N different samples (e.g., the ones immediately following those already considered) it would be possible to use the M paths used to compute the first set of measurements to generate M additional measurements. By repeating this q times, we end with an amount of $m = qM$ measurements computed by using $n = qN$ different input samples. This approach is sketched in Fig. 6.16.

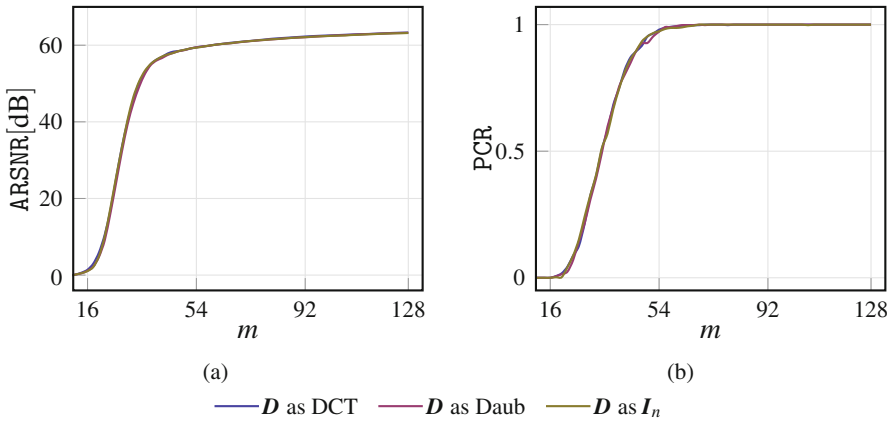


Fig. 6.15 Montecarlo comparison between performance of an RMPI system with $n = 128$, $\kappa = 6$ in terms of signal reconstruction quality—plot (**a**)—and probability of correct reconstruction—plot (**b**)—as a function of the number of measurement m. The considered sparsity bases are the Discrete Cosine Transform (DCT), the Daubechies-4 Wavelet (Daub), and the $n \times n$ identity matrix ($\boldsymbol{I}_n$)

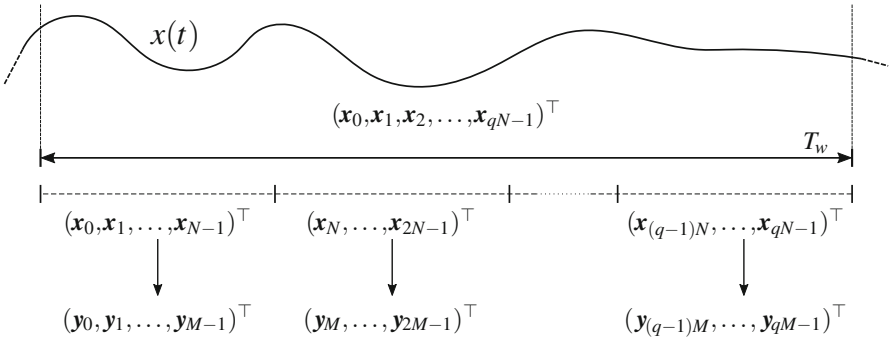


Fig. 6.16 Timing approach used in hybrid RD/RMPI architectures

This approach can be described by a highly structured $\boldsymbol{A}$, an example of that is depicted in Fig. 6.17. This sensing matrix is block diagonal, composed by q blocks. Each block is an $M \times N$, and describes how a block of N samples gives rise to a block of M measurements. Roughly speaking, each block represents an $M \times N$ RMPI system, that is applied to a different slice of the input signal in a way that is very similar to that used in the RD approach. By collecting the measurements from all blocks, it is possible to get a sufficient number of measurements to reconstruct the entire signal. According to another point of view, it is possible to see this approach as M RD systems working in a parallel fashion, as a full RMPI system. Note that each sensing block in $\boldsymbol{A}$ could be different from the each other, but they could also be identical to allow a simpler hardware implementation.

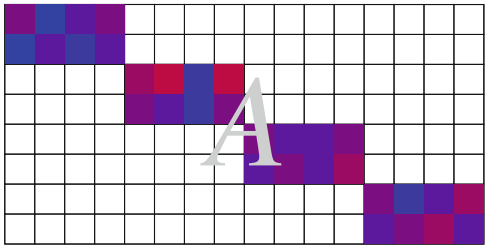


Fig. 6.17 Example of an 8×16 sensing matrix $\boldsymbol{A}$ corresponding to the implementation of a hybrid RD-RMPI architecture with $q = 4$. *White blocks* correspond to zero elements

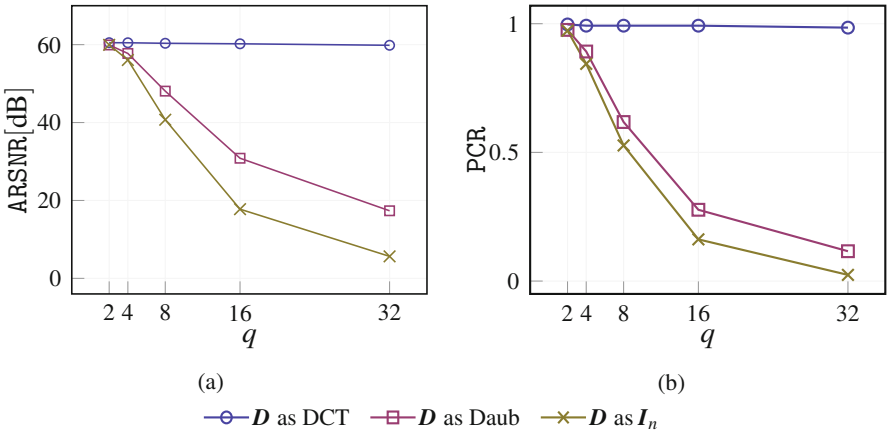


Fig. 6.18 Montecarlo comparison between performance of a hybrid RD/RMPI system with $n = 128$, $m = 64$, $\kappa = 6$ in terms of signal reconstruction quality—plot (**a**)—and probability of correct reconstruction—plot (**b**)—as a function of the number of blocks q. The considered sparsity bases are the Discrete Cosine Transform (DCT), the Daubechies-4 Wavelet (Daub), and the $n \times n$ identity matrix ($\boldsymbol{I}_n$)

Performance of a hybrid RD/RMPI system in terms of both signal reconstruction quality and probability of correct reconstruction can be found in Fig. 6.18. The system considered is the same as in the other cases, i.e., $n = 128$ and $\kappa = 6$. The number of measurements is fixed to $m = 64$, and results have been plotted as a function of the number of blocks q, with $N = n/q$ and $M = m/q$. The behavior is intermediate between the RD and the RMPI. For small values of q, a few large blocks can be found in the $\boldsymbol{A}$ (i.e., N and M are large) and performance are similar to that of the RMPI: independently of the sparsity basis $\boldsymbol{D}$, reconstruction is achieved with high quality and high probability to be correct. As q increases, blocks are smaller (M and N decrease), and the structure of $\boldsymbol{A}$ is similar to that of the RD approach. Here, exactly like in the RD case, good results are achieved only for the DCT sparsity basis, but the system is not capable anymore to correctly reconstruct signals sparse on the Daub or identity basis.

6.4 The Saturation Problem

Independently of the architecture adopted from those in Sect. 6.2, in general the computation of the j-th measurements can be written as

$$\boldsymbol{y}_j = \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k}\, \boldsymbol{x}_k \tag{6.7}$$

Then, the result $\boldsymbol{y}_j$ is quantized as $Q(y_j)$ before being either transmitted or stored into memory.

One or more of the $\boldsymbol{A}_{j,k}$, $k = 0, \ldots, n-1$, are different from zero and give a contribution to the sum. We indicate this quantity as $\|\boldsymbol{A}_{j,\cdot}\|_0 = N_j$.

As already seen in Chap. 3, the central limit theorem can be applied to (6.7). In detail, under the assumption that N_j is large enough (in practical cases, starting from values of N_j in the order of 10), then $\boldsymbol{y}_j$ can be assumed having a Gaussian distribution. This relies on the following assumptions:

i let us ignore in the sum (6.7) all terms for which $\boldsymbol{A}_{j,k} = 0$, i.e., let indicate with K the subset of indexes for which $\boldsymbol{A}_{j,k} \neq 0$ if and only if $k \in K$. In this way

$$\boldsymbol{y}_j = \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k}\, \boldsymbol{x}_k = \sum_{k \in K} \boldsymbol{A}_{j,k}\, \boldsymbol{x}_k.$$

Alternatively, we can also assume that $\boldsymbol{A}_{j,k} \neq 0$, $\forall k$, with $N_j = n$.

ii the N_j random variables given by the product $\boldsymbol{A}_{j,k}\, \boldsymbol{x}_k$ for $k \in K$ can be considered *independent* of each other. This happens when either the $\boldsymbol{A}_{j,k}$ are independent of each other, or the $\boldsymbol{x}_k$ are independent of each other.

iii the N_j random variables $\boldsymbol{A}_{j,k}\, \boldsymbol{x}_k$ with $k \in K$ can be considered *identically distributed*, with zero-mean $\mathbf{E}[\boldsymbol{A}_{j,k}\, \boldsymbol{x}_k] = 0$ and variance $\mathbf{E}[\boldsymbol{A}_{j,k}^2\, x_k^2]$ that is finite and independent of k.

Then, central limit theorem can be applied considering the random variables $\xi_{j,k} = \boldsymbol{A}_{j,k}\, \boldsymbol{x}_k / \sqrt{\mathbf{E}[\boldsymbol{A}_{j,k}^2\, \boldsymbol{x}_k^2]}$, that are independent of each other, with zero-mean and unity variance. The sum $\sum_{k \in K} \xi_{j,k} / \sqrt{N_j}$ can be approximated, for large N_j, with a standard normal random variable. Observing that

$$\boldsymbol{y}_j = \sqrt{N_j \mathbf{E}[\boldsymbol{A}_{j,k}^2\, \boldsymbol{x}_k^2]} \sum_{k \in K} \frac{1}{\sqrt{N_j}} \xi_{j,k}$$

then if $\sigma_y = \sqrt{N_j\, \mathbf{E}[\boldsymbol{A}_{j,k}^2\, \boldsymbol{x}_k^2]}$ has a finite value, also $\boldsymbol{y}_j$ can be seen as a zero-mean Gaussian variables with variance σ_y^2. The larger the N_j, the better this approximation.

The Gaussian limit implies a potentially serious design problem, assuming a pure Gaussian approximation, $\boldsymbol{y}_j$ can take value in the whole real set. In more practical cases, we can say that $\boldsymbol{y}_j$ may assume very large values, but the majority of the observed cases are located around the mean value $\mathbf{E}[\boldsymbol{y}_j] = 0$. This is an important issue when paired with a quantization function $Q(\cdot)$ that is uniform, i.e., all quantization steps have the same size, and with a limited conversion range, i.e., it presents two thresholds (an upper one and a lower one) identifying an interval inside which conversion is achieved, while outside saturation happens. All real-world quantizers (so, all real-word ADCs) are uniform with a finite conversion range.

Assuming for the sake of simplicity that σ_y is known, let us indicate with $\gamma_Q\sigma_y$ and with $-\gamma_Q\sigma_y$ the upper and the lower saturation thresholds of the $Q(\cdot)$, respectively. This implicitly define a quantization step $\Delta = 2\gamma_Q\sigma_y/l$, where l is the number of levels of the quantizer. However, due to the Gaussian approximation, instances of $\boldsymbol{y}_j$ may fall outside the quantization interval $[-\gamma_Q\sigma_y, \gamma_Q\sigma_y]$. In other words, there is a certain probability p_{sat} that $Q(\cdot)$ *saturates*, and due to the Gaussian approximation this is given by

$$p_{\text{sat}} = \Pr(|\boldsymbol{y}_j| > \gamma_Q\sigma_y) = \text{erfc}\left(\frac{\gamma_Q}{\sqrt{2}}\right)$$

and consequently a probability of *non-saturation*, given by

$$p_{\neg\text{sat}} = 1 - p_{\text{sat}} = 1 - \text{erfc}\left(\frac{\gamma_Q}{\sqrt{2}}\right)$$

where $\text{erfc}(\cdot)$ is the complementary error function. We refer to this as *static saturation*.

Yet, a second more subtle problem exists. Let us recast (6.7) as

$$\begin{aligned} y_j^{(i)} &= \sum_{k=0}^{i} \boldsymbol{A}_{j,k}\,\boldsymbol{x}_k \\ \boldsymbol{y}_j &= y_j^{(n-1)} \end{aligned} \tag{6.8}$$

In other words, $y_j^{(i)}$ is the intermediate value of $\boldsymbol{y}_j$ accumulated on the hardware computing (6.7) at step i. Also this block has an upper and a lower bound due, for example, to the limited number of bits in case (6.8) is computed on a digital hardware, or to saturation of the adder (or the integrator) used in an analog implementation. Let us indicate them with $\gamma_\Sigma\sigma_y$ and $-\gamma_\Sigma\sigma_y$, respectively. If at any time step i it happens that $|y_j^{(i)}| > \gamma_\Sigma\sigma_y$, we deal with a *dynamic saturation*.

Note that a static saturation is an event that can be easily detected, since measurements outside the conversion range are automatically converted either to the

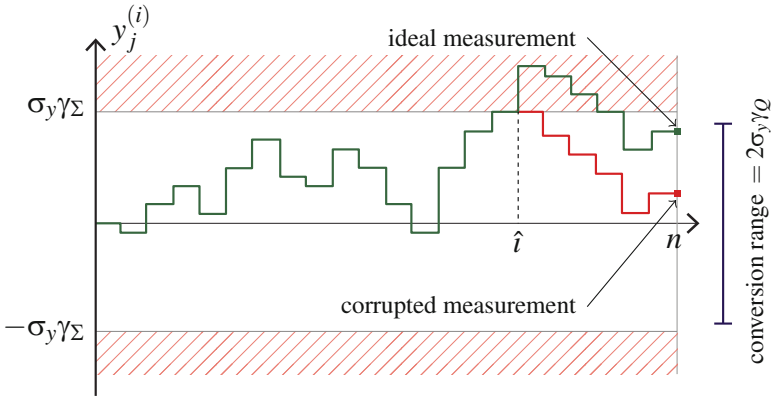


Fig. 6.19 Example of evolution of the computation of the j-th measurement by means of intermediate values $y_j^{(i)}$. If dynamic saturation happens at time step $\hat{\imath}$, evolution ends with a non-correct, corrupted value. Note that the final corrupted measurement value at time step n may still fall inside the $Q(\cdot)$ conversion range

maximum or minimum digital value by $Q(\cdot)$. Conversely, when a dynamic saturation happens at time step $\hat{\imath} < n-1$, the final value $\boldsymbol{y}_j$ is irreparably corrupted. Note that, depending on the values of $\boldsymbol{A}_{j,k}\,\boldsymbol{x}_k$ for $k = \hat{\imath}, \ldots, n-1$, the computed y_j may actually fall in the $Q(\cdot)$ conversion range as in the example of Fig. 6.19. Due to this, simply examining $\boldsymbol{y}_j$ is not useful for detecting these events.

The computation of the probability of a dynamic saturation is not an easy task, since the path followed by $y_k^{(i)}$ has to be modeled as a random walk. By indicating with $\tilde{p}_{\neg\text{sat}}$ the probability that, while computing (6.7), neither a static saturation, nor a dynamic saturation happens, it is, however, certainly $\tilde{p}_{\neg\text{sat}} \leq p_{\neg\text{sat}}$.

At this point, one may wonder how to deal with the above described static and dynamic saturation in real-world AICs. Some theoretical considerations on the effect of static saturation can be found in [10] while a discussion of both static and dynamic saturation in the more realistic model we adopt here has been first proposed in [11].

Two considerations are worth mentioning here. The first one regards the correct way in which γ_Q and γ_Σ should be selected. This is actually a trade-off and, referring in particular to γ_Q, we can say:

- when designing a system, it should be $\gamma_\Sigma \geq \gamma_Q$.
- when γ_Q has a low value, then the quantization step Δ is small, it is the quantization error in each measurement. Even if relating the measurement quantization error with the reconstruction error is not trivial (see Chap. 8 for details), it reasonable that the higher the measurement quantization error, the higher the reconstruction error. From this point of view, a low γ_Q is preferable. However, since $p_{\neg\text{sat}}$, and so $\tilde{p}_{\neg\text{sat}}$ is decreasing with γ_Q, we have to deal with many saturated or corrupted measurements;

- when γ_Q has a high value, $\tilde{p}_{\neg\text{sat}}$ may be low. However, Δ is large thus producing a non-negligible quantization error that either is inevitably reflected in poor reconstruction performance, or must be compensated by using a large number of quantization levels l.

The second consideration is about the way in which saturated measurements should be considered in the reconstruction algorithm. A first, straightforward approach to cope with static saturation is to exploit the allegedly "democracy" of the set of measurements [10], i.e., the fact that, under certain conditions, one may assume that the information content of each measurement is identical. If this were true, simply discarding saturated measurement would produce a graceful performance degradation since the acquisition system would behave as if it were designed to use a number of measurement equal to the number of non-saturated measurement. Moreover, non-degraded performance could be restored by simply taking further measurement until the original number is reached. Following [7], we will name this approach as SPD since it concretizes in Saturated Projection Dropping.

This aspect has been intensively discussed in Sect. 2.5. Perfect democracy only holds between measurement that are taken as perfect linear combinations of the samples. Saturation acts as a selector discarding those that have a larger value while keeping the smaller ones. From the point of view of the signal-to-noise ratio this is clearly not a democratic behavior and causes non-saturated measurements to be less useful than those that have to be dropped and cannot be perfectly replaced by simply trying more measurements. In fact, as discussed in Sect. 2.5, discarded measurements are the ones with the higher energy, i.e., those containing the largest quantity of information.

The problem of dealing with saturated measurements has been addressed in detail in [11], and has no simple solutions other than using a large value of γ_Q and an even larger γ_Σ thus unavoidably increasing measures quantization error. According to simulation results proposed in [11], exploiting at the decoder side the information that the j-th channel is saturated by replacing the corresponding equation $\sum_{k=0}^{n-1} = \boldsymbol{y}_j$ with the inequality $\sum_{k=0}^{n-1} > \gamma_Q \sigma_y$ or $\sum_{k=0}^{n-1} < -\gamma_Q \sigma_y$ (for a positive and negative static saturation, respectively) does not increase reconstruction performance with respect to simply dropping saturated measurements.

However, [11] proposes a possible workaround. By recalling that the philosophy underlying the entire CS framework is to recover a signal with the lowest possible amount of information, the number of measurement m is usually not far from its lower theoretical bound, and so discarding even a single measurement may lead to an insufficient quantity of information to correctly reconstruct the signal. In other words, in order to ensure reconstruction one should try to recover some amount of information even from corrupted measurements by replacing them with the last reliable data we have. As in the old common saying, exploiting "*everything but the Oink!*," one should be able to plug into the reconstruction algorithm whatever kind of information agrees with the measurement outcomes.

The workaround proposed in [11] is to introduce an almost negligible hardware overhead in order to be able to check at any time step i if saturation happens. This allows to exactly detect the time instant $\hat{\imath}$ when dynamic saturation occurs.

Given $\hat{\imath}$, we can reasonably assume that $y_j^{(\hat{\imath})} \approx \gamma_\Sigma \sigma_y$ if a *positive* dynamic saturation occurs, and that $y_j^{(\hat{\imath})} \approx -\gamma_\Sigma \sigma_y$ if a *negative* one happens. In other words,

$$\sum_{j=0}^{\hat{\imath}} \boldsymbol{A}_{j,k}\, x_k = \begin{cases} \gamma_\Sigma \sigma_y & \text{if positive saturation occurred} \\ -\gamma_\Sigma \sigma_y & \text{if negative saturation occurred} \end{cases}. \tag{6.9}$$

Replacing the equation associated with the j-th corrupted measurement with (6.9) in the decoding algorithm makes possible to effectively exploit all known information on the signal. This can be easily done by replacing the sensing matrix $\boldsymbol{A}$ with an adjusted one $\boldsymbol{A}'$ where the j-th row $\boldsymbol{A}_{j,\cdot}$ elements have been zeroed in correspondence to all time instants after the one in which saturation occurs. More formally, we may set

$$\boldsymbol{A}'_{j,k} = \begin{cases} \boldsymbol{A}_{j,k} & \text{for } k = 0, \ldots, \hat{\imath} - 1 \\ 0 & \text{for } k = \hat{\imath}, \ldots, n-1 \end{cases} \tag{6.10}$$

as well as

$$\boldsymbol{y}'_j = \begin{cases} \boldsymbol{y}_j & \text{if no saturation occurred} \\ \gamma_\Sigma \sigma_y & \text{if positive saturation occurred} \\ -\gamma_\Sigma \sigma_y & \text{if negative saturation occurred} \end{cases} \tag{6.11}$$

to reformulate the measurement equation (6.7) with

$$\boldsymbol{y}'_j = \sum_{k=0}^{n-1} \boldsymbol{A}'_{j,k}\, \boldsymbol{x}_k$$

that holds independently of any dynamic saturation event observed. Of course, if multiple saturation events are detected for the same j-th measurement, only the first one may to be used. Following [7], we indicate this approach as SPW, standing for Saturated Projection Windowing.

Note that this solution makes the matrix $\boldsymbol{A}'$ used for reconstruction a function of the signal samples x_k that caused saturation. Hence, the measurement vector passed to the decoder must contain also the information needed to switch from the signal-independent $\boldsymbol{A}$ to $\boldsymbol{A}'$. Note also that saturation may happen in more than one row at different time steps. This case is illustrated in the example of Fig. 6.20.

In [11] a comparison of the performance of the SPD and the SPW approaches is also proposed. We want to recall here just some results in terms of probability of

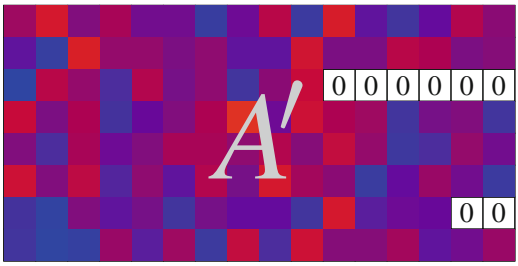


Fig. 6.20 Example of an RMPI sensing matrix $\boldsymbol{A}'$ corrected accordingly to (6.10) in order to cope with saturated measurement (SPW approach)

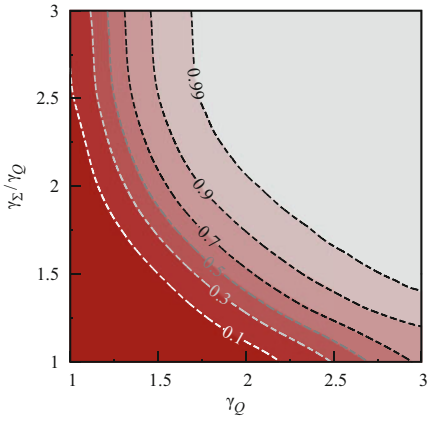


Fig. 6.21 PCR as a function of γ_Q and γ_Σ/γ_Q in a Montecarlo simulation adopting the SPD approach (adapted from [11])

correct reconstruction, that has been defined in [11] exactly as in Sect. 2.3, but with a different target $\texttt{RSNR}[\text{dB}]_{\text{min}}$ and different ISNR levels, in a system with $n = 256$, $m = 64$, $\kappa = 6$, $\boldsymbol{A} \in \{-1, +1\}^{m \times n}$ and, for the sake of simplicity, $\boldsymbol{D} = \boldsymbol{I}_n$.

We have plotted in Fig. 6.21 the contour plot of the relationship between PCR and the two parameters γ_Q and γ_Σ/γ_Q (darker colors correspond to lower PCR values) obtained from Montecarlo simulations of a system implementing the SPD approach, i.e., dropping all saturated measurements. Even for very high values of γ_Σ/γ_Q for which the probability of an undetected corruption at the summing stage vanishes, performance degradation is always substantial: a system aiming at 99% of PCR, while keeping the two saturation thresholds as close as possible, should reserve a $\gamma_Q > 3$ for the quantization stage and $\gamma\Sigma > 1.5 \times \gamma_Q = 4.5$ for the summing stage.

Figure 6.21 should be compared with Fig. 6.22, showing performance of the SPW under the same simulation setting. When corruption is properly handled, a 99% PCR can be easily reached for $\gamma_\Sigma/\gamma_Q \simeq 1$ and for very small values of γ_Q, thus allowing an extremely effective implementation. The approach suffers from a small drawback only for γ_Σ large and γ_Q small. The reason is clear: in this region, static saturation has a high probability, while dynamic saturation has a low one. In other words, we are dealing with a lot of saturated measurements that have to be discarded, not being possible to retrieve information from (6.9) due to lack of dynamic saturation. As in the SPD case, for all saturated measurements, using inequalities like $\gamma_Q \sigma_y <$

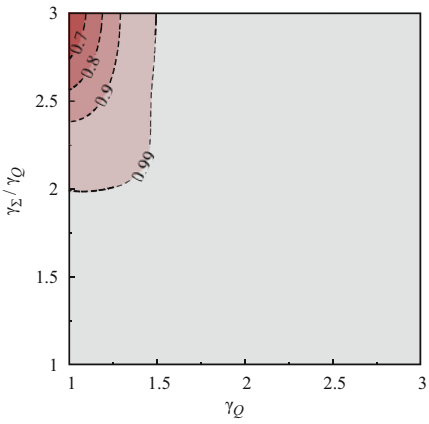


Fig. 6.22 PCR as a function of γ_Q and γ_Σ/γ_Q in a Montecarlo simulation adopting the SPW approach (adapted from [11])

$\boldsymbol{y}_j < \gamma_\Sigma \sigma_y$ or $-\gamma_\Sigma \sigma_y < \boldsymbol{y}_j < -\gamma_Q \sigma_y$ does not increase reconstruction performance. Accordingly to this observation, the optimal choice for SPW is to set $\gamma_\Sigma = \gamma_Q$.

The aforementioned SPW approach has been implemented in the RMPI prototype described in [12] with very good results. A short summary of this real-world case can be found in Sect. 7.6.

6.5 From Temporal Domain to Mixed Spatial–Temporal Domain

In the general model for CS systems defined in Sect. 6.2, the computation of the m-dimensional measurement vector $\boldsymbol{y}$ is achieved starting from an n-size input signal vector $\boldsymbol{x} \in \mathbb{R}^n$ by means of the sensing matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ where $\boldsymbol{x}$ is known to be sparse in a properly defined basis $\boldsymbol{D}$. Reconstruction is achieved with the usual minimization problem

$$\begin{aligned} &\arg\min_{\boldsymbol{\xi} \in \mathbb{R}^n} \|\boldsymbol{\xi}\|_1 \\ &\quad \text{s.t. } \|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\xi}\|_2 \leq \epsilon \end{aligned}$$

with $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$, $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{D}$ ($\boldsymbol{B} \in \mathbb{R}^{m \times n}$), and with ϵ a proper positive constant.

In all this chapter, the input signal has been modeled as a 1-dimension time-domain signal $x(t)$, that is sampled during a single time window $0 \leq t < T_w$ yielding an input signal vector $\boldsymbol{x} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1})^\top$ achieved, for example, by sampling $x(t)$ at Nyquist rate, i.e., $\boldsymbol{x}_k = x(kT)$.

Of course, many different signal models exist. In this section we want to deal with the case where the input signal is an n_S-dimension one, as in the case of a multi-lead electrocardiographic (ECG) or electroencephalographic (EEG) signal [14]. In this case, we have to model the input signal as an array of n_S time-domain functions

$\boldsymbol{x}(t) = (x^{(0)}(t), x^{(1)}(t), \ldots, x^{(n_S-1)}(t))$. Assuming that this signal is observed in a time windows $0 \leq t < T_w$, and that $T_w = dT$, than we deal with an *input matrix* $\boldsymbol{X} \in \mathbb{R}^{d \times n_S}$ defined as

$$\boldsymbol{X} = \underbrace{\left.\begin{pmatrix} x^{(0)}(0) & x^{(1)}(0) & \cdots & x^{(n_S-1)}(0) \\ x^{(0)}(T) & x^{(1)}(T) & \cdots & x^{(n_S-1)}(T) \\ \vdots & \vdots & \ddots & \vdots \\ x^{(0)}((d-1)T) & x^{(1)}((d-1)T) & \cdots & x^{(n_S-1)}((d-1)T) \end{pmatrix}\right\} d \text{ rows}}_{n_S \text{ colums}} \tag{6.12}$$

where $\boldsymbol{X}_{j,k} = x^{(k)}(jT)$. We can still define $n = dn_S$ to keep the same complexity (in terms of dimensionality) as the standard CS problem considered up to now.

For $n_S = 1$ this model is equivalent to the formulation already considered. The dual case, i.e., $d = 1$ takes into account a 1-dimension *spatial*-domain system. From a mathematical point of view, the only difference with respect to the standard approach is that we have an input row vector instead of an input column vector. Of course, it is not difficult to deal with this system under the framework developed so far.

More interesting is the case where $n_S > 1, d > 1$. From a *physical* point of view, we are dealing with a mixed time-spatial system, where we could exploit sparsity properties in both domains. To cite a few examples, it is known that in a stereo audio recording, right and left channels are strongly correlated. This information is always used in coding algorithms: the most used approach is to encode in high quality the common-mode information (i.e., the sum of the left and of the right channels) and in low quality the differential information [8]. A similar situation is present in a multi-lead EEG system, since signals coming from adjacent channels are strongly correlated [14].

Yet, in this section we are more interested in the mathematical aspect of this approach. So, we neglect the system physical aspects and assume that for some reason the input signal takes the form of a $d \times n_S$ matrix as in (6.12). In this case the relation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ used in the standard CS approach holds as $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{X}$ if we assume that $\boldsymbol{A}$ is a three-dimensional $m \times d \times n_S$ object, and that we define a proper product $\mathbb{R}^{m \times d \times n_S} \circ \mathbb{R}^{d \times n_S} \to \mathbb{R}^m$, so that $\boldsymbol{y}$ is still an m-size measurements vector. In the same way, the relation $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$, which has to be written now as $\boldsymbol{X} = \boldsymbol{D}\boldsymbol{\Xi}$, holds only if this product is not considered anymore a matrix/vector product, but a suitable defined product involving higher-dimension objects. Despite possible, this approach poses several issues, in particular in the definition of the $\boldsymbol{A}$ and of the $\boldsymbol{D}$.

A more pragmatic approach is to define a reshaping operator $\mathscr{P} : \mathbb{R}^{d \times n_S} \to \mathbb{R}^{d n_S}$ taking the sample matrix $\boldsymbol{X}$ as input and concatenating its rows or columns after each other, thus generating a sample vector $\boldsymbol{x} \in \mathbb{R}^{d n_S}$. Defining $n = dn_S$, this system

can be included in the general CS framework considering $\boldsymbol{x} = \mathscr{P}(\boldsymbol{X})$ as sampling vector. In other words, measurements are obtained by the usual matrix/vector multiplication

$$\boldsymbol{y} = \boldsymbol{A}\mathscr{P}(\boldsymbol{X})$$

where $A \in \mathbb{R}^{m \times d\, n_S}$ is the standard two-dimension sampling matrix.

This approach has been followed in [14]. In particular, authors used two different reshaping operators: one gathering elements of $\boldsymbol{X}$ accordingly to their sampling instant in order to simplify the computation of the measurements, and one gathering elements of $\boldsymbol{X}$ accordingly to the channel number in order to exploit the temporal sparsity properties of the EEG signal.

In more detail, measurements in [14] are taken accordingly to time-priority reshaping operator $\mathscr{P}(\cdot)$ defined as

$$\mathscr{P}(\boldsymbol{X}) = \left(x^{(0)}(0), x^{(1)}(0), \ldots, x^{(n_S-1)}(0), x^{(0)}((d-1)T), \ldots, x^{(n_S-1)}((d-1)T)\right)^\top$$

where $\boldsymbol{X}$ is defined as in (6.12). In other words, the first n_S elements of $\boldsymbol{x} = \mathscr{P}(\boldsymbol{X})$ are the sampling of the n_S EEG channels at the first sampling instant. Measurements are taken accordingly to $\boldsymbol{y} = \boldsymbol{A}\mathscr{P}(\boldsymbol{X})$, where

$$\boldsymbol{A} = \underbrace{\left.\begin{pmatrix} \boldsymbol{A}^{(0)} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}^{(1)} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{A}^{(d-1)} \end{pmatrix}\right\} d\, m_S = m \text{ rows}}_{d\, n_S = n \text{ colums}}$$

where the $\boldsymbol{A}^{(j)} \in \mathbb{R}^{m_S \times n_S}$ are sampling sub-matrices. In this way, the $\boldsymbol{A}^{(0)}$ generates m_S measurements from the n_S samples, one for each channel, at the time step 0, the $\boldsymbol{A}^{(1)}$ generates m_S measurements from samples at time step 1, and so on. The generation of measurements, from a hardware point of view, is simplified since at each time step n_S samples are generated and used to compute m_S measurements. After that, samples are discarded, and new ones are used to compute other m_S measurements. The total amount of measurement is m, with $m = d\, m_S$. Note that, accordingly to this definition, it is still $\boldsymbol{A} \in \mathbb{R}^{m \times n}$.

In order to avoid a complex definition in the whole spatial–temporal domain, sparsity in [14] is considered in the temporal domain only. In details, authors exploit the well-known property of EEG signals to be sparse in a Gabor domain [13]. To this aim, a second spatial-priority reshaping operator is introduced

$$\mathscr{S}(\boldsymbol{X}) = \left(x^{(0)}(0), x^{(0)}(T), \ldots, x^{(0)}((d-1)T), x^{(1)}(0), \ldots, x^{(n_S-1)}((d-1)T)\right)^\top$$

where the first d elements of $\boldsymbol{x} = \mathscr{S}(\boldsymbol{X})$ are the d Nyquist samples of the first EEG channel. By introducing $\boldsymbol{\Xi} \in \mathbb{R}^{d \times n_S}$ as the sparse coefficient matrix of $\boldsymbol{X} \in \mathbb{R}^{d \times n_S}$, mathematically

$$\mathscr{S}(\boldsymbol{X}) = \boldsymbol{D}\mathscr{S}(\boldsymbol{\Xi})$$

the sparsity matrix $\boldsymbol{D}$ is defined as

$$\boldsymbol{D} = \underbrace{\left.\begin{pmatrix} \boldsymbol{D}^{(0)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{D}^{(1)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{D}^{(n_S-1)} \end{pmatrix}\right\} d\, n_s = n \text{ rows}}_{dn_S = n \text{ colums}}$$

where the $\boldsymbol{D}^{(0)} = \boldsymbol{D}^{(1)} = \ldots = \boldsymbol{D}^{(n_S-1)} \in \mathbb{R}^{d \times d}$ are all equal to the each other, and given by the standard $d \times d$ sparsity matrix of a single channel EEG made by d samples.

In other words, due to the particular structure of $\boldsymbol{D}$, the sparsity properties of the k-th EEG channel given by the k-th column $\boldsymbol{X}_{\cdot,k-1}$ of $\boldsymbol{X}$, are determined only by the k-th column $\boldsymbol{\Xi}_{\cdot,k-1}$ of the sparse coefficient matrix $\boldsymbol{\Xi}$ as

$$\begin{pmatrix} x^{(k-1)}(0) \\ x^{(k-1)}(T) \\ \vdots \\ x^{(k-1)}((d-1)T) \end{pmatrix} = \boldsymbol{D}^{(k-1)} \begin{pmatrix} \boldsymbol{\Xi}_{0,k-1} \\ \boldsymbol{\Xi}_{1,k-1} \\ \vdots \\ \boldsymbol{\Xi}_{d-1,k-1} \end{pmatrix}$$

This approach easily allows to transfer all the know-how on the sparsity for a single channel to the multichannel approach. Reconstruction can be achieved by asking that the k-th EEG channel is sparse accordingly to its $\boldsymbol{D}^{(k-1)}$ sparsity matrix. Mathematically

$$\begin{aligned} &\arg\min_{\boldsymbol{\Xi} \in \mathbb{R}^{d \times n_S}} \|\mathscr{S}(\boldsymbol{\Xi})\|_1 \\ &\quad \text{s.t. } \|\boldsymbol{y} - \boldsymbol{B}\mathscr{S}(\boldsymbol{\Xi})\|_2 \leq \epsilon \end{aligned} \tag{6.13}$$

with $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{D}$ as usual.

However, even if this makes the reconstruction approach aligned with the general CS theory, it is easy to see that, due to the block-diagonal nature of $\boldsymbol{D}$, (6.13) induces coefficient-wise sparsity without considering the inter-channel correlation of the neural signals. The neural recovery model should employ the cross correlations of EEG signals to improve the reconstruction quality. To cope with this, an appropriate more complex model for multichannel neural signals is also introduced.

Authors of [14] modeled the dependency of neural signals using a mixed $\ell_{1,2}$ norm [9]. The mixed $\ell_{1,2}$ norm of $\boldsymbol{\Xi}$ is defined as

$$\|\boldsymbol{\Xi}\|_{1,2} = \sum_{j=0}^{d-1} \sqrt{\sum_{k=0}^{n_s-1} \boldsymbol{\Xi}_{j,k}{}^2}$$

i.e., the ℓ_2 norm of every row of $\boldsymbol{\Xi}$ is computed, and the ℓ_1 norm of all results gives the desired result. The solution to the multichannel neural recovery using the mixed $\ell_{1,2}$ norm is obtained by replacing the ℓ_1 norm by the mixed norm in the reconstruction problem as

$$\begin{aligned} &\underset{\boldsymbol{\Xi} \in \mathbb{R}^{d \times n_S}}{\arg\min} \ \|\mathscr{S}(\boldsymbol{\Xi})\|_{1,2} \\ &\quad \text{s.t.} \ \ \|\boldsymbol{y} - \boldsymbol{B}\mathscr{S}(\boldsymbol{\Xi})\|_2 \leq \epsilon \end{aligned} \tag{6.14}$$

According to [14], the $\ell_{1,2}$ norm respects the group structure of neural signals and imposes sparsity on the group of coefficients rather than each coefficient independently. Results show an improvement in the joint reconstruction of the multichannel EEG signal of about 5 dB when (6.14) is used with respect to the case when (6.13) is used.

References

1. N. Ahmed, T. Natarajan, K.R. Rao, Discrete cosine transform. IEEE Trans. Comput. **C-23**(1), 90–93 (1974)
2. V. Cambareri et al., A case study in low-complexity ECG signal encoding: how compressing is compressed sensing? IEEE Signal Process. Lett. **22**(10), 1743–1747 (2015)
3. X. Chen et al., A sub-Nyquist rate compressive sensing data acquisition front-end. IEEE J. Emerging Sel. Top. Circuits Syst. **2**(3), 542–551 (2012)
4. X. Chen et al., A sub-Nyquist rate sampling receiver exploiting compressive sensing. IEEE Trans. Circuits Syst. I Regul. Pap. **58**(3), 507–520 (2011)
5. I. Daubechies, Orthonormal bases of compactly supported wavelets. Commun. Pure Appl. Math. **41**(7), 909–996 (1988)
6. D. Gangopadhyay et al., Compressed sensing analog front-end for bio-sensor applications. IEEE J. Solid State Circuits **49**(2), 426–438 (2014)
7. J. Haboba et al., A pragmatic look at some compressive sensing architectures with saturation and quantization. IEEE J. Emerging Sel. Top. Circuits Syst. **2**(3), 443–459 (2012)
8. M. Hans, R.W. Schafer, Lossless compression of digital audio. IEEE Signal Process. Mag. **18**(4), 21–32 (2001)
9. M. Kowalski, B. Torrésani, Sparsity and persistence: mixed norms provide simple signal models with dependent coefficients. Signal Image Video Process. **3**(3), 251–264 (2009)
10. J.N. Laska et al., Democracy in action: Quantization, saturation, and compressive sensing. Appl. Comput. Harmon. Anal. **31**(3), 429–443 (2011)
11. M. Mangia et al., Coping with saturating projection stages in RMPI-based Compressive Sensing, in *2012 IEEE International Symposium on Circuits and Systems*, May 2012, pp. 2805–2808

12. F. Pareschi et al., Hardware-algorithms co-design and implementation of an analog-to-information converter for biosignals based on compressed sensing. IEEE Trans. Biomed. Circuits Syst. **10**(1), 149–162 (2016)
13. S. Qian, D. Chen, Discrete Gabor transform. IEEE Trans. Signal Process. **41**(7), 2429–2438 (1993)
14. M. Shoaran et al., Compact low-power cortical recording architecture for compressive multichannel data acquisition. IEEE Trans. Biomed. Circuits Syst. **8**(6), 857–870 (2014)
15. M. Wakin et al., A nonuniform sampler for wideband spectrally-sparse environments. IEEE J. Emerging Sel. Top. Circuits Syst. **2**(3), 516–529 (2012)
16. J. Yoo et al., A 100MHz-2GHz 12.5x sub-Nyquist rate receiver in 90 nm CMOS, in *2012 IEEE Radio Frequency Integrated Circuits Symposium*, June 2012, pp. 31–34
17. J. Yoo et al., Design and implementation of a fully integrated compressed-sensing signal acquisition system, in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2012, pp. 5325–5328

Chapter 7
Analog-to-Information Conversion

No monolithic implementations of Compressed Sensing (CS)-based acquisition systems have been proposed so far as commercial products. Yet, a number of prototypes have appeared in the scientific literature. In this chapter we present some hardware architectures recently proposed in the most important microelectronics conferences and journals, capable of working as CS-based analog-to-information converters (AICs).

All the works considered here share a common methodology. In fact, despite being very different from each other, the front-end implementation is always analog, i.e., all of them belong to the class of *analog* Compressed Sensing systems exploiting the random modulation pre-integration (RMPI) architecture. The order in which these works are considered is the same order in which they appeared in the literature.

Conversely, no architectures exploiting the random sampling (RS) architecture is considered here, even if some implementation of AIC based on this approach has been proposed so far [26]. The main reason is that a RS-based AIC is basically a standard analog-to-digital converter (ADC) where randomization is added into the control logic. However, the core architecture is typically the very same of a standard ADC with performance (bandwidth, precision, etc.) similar to that of the ADC embedded in the AIC, while RMPI solutions require a full-custom design. Furthermore, based on the observations of Chap. 6, RS architectures are not general purpose ones and are able to achieve good performance only on a limited class of input signals.

M. Mangia et al., *Adapted Compressed Sensing for Effective Hardware Implementations*, DOI 10.1007/978-3-319-61373-4_7

7.1 Introduction and Notation

To the best of authors' knowledge, the first RMPI prototype appeared in the literature has been presented by Yoo et al. at the *2012 IEEE Radio Frequency Integrated Circuits Symposium* [27], while some design aspects appeared a few months before in [28]. The circuit is a sub-Nyquist-rate receiver for radar pulse signal designed in 90 nm technology, with up to 2 GHz signals, and it is presented in Sect. 7.2.

In Sect. 7.3 we review the work of Chen et al. preliminarily appeared in [9] and then successively fully characterized at the end of 2012 in [8]. The circuit, fabricated in 90 nm CMOS process, implements a data acquisition front end for a radio frequency (RF) communication system when assuming a multi-tone input signal.

In Sect. 7.4 another analog-to-information converter for biomedical signal is presented. In more detail this circuit, presented in [12] by Gangopadhyay et al. and designed in 180 nm CMOS process, is an analog front end for electrocardiographic (ECG) signals. This architecture features a 6-bit multiplying digital-to-analog converter (DAC) embedded in the integrator circuit as multiplying block.

In Sect. 7.5 we propose a short review of [25] presented by Shoaran et al. in 2014. This work is a low-power sub-Nyquist sampler for the multichannel acquisition of cortical intracranial electroencephalographic (EEG) signals. The peculiarity of this architecture, which has been fabricated in 180 nm CMOS process, is to exploit sparsity in the spatial domain instead of in the temporal domain.

The last considered architecture has been presented by Pareschi et al. in [21]. This circuit, which has been designed in 180 nm CMOS process, is an analog-to-information converter for generic biomedical signals. Authors propose measurements both on ECG signals and electromyographic (EMG) signals. The peculiarity of this converter is to introduce a smart saturation checking mechanism, with which it is possible to reconstruct the acquired signal even if many measurements suffer saturation, and to exploit the *rakeness* approach [17] to minimize the number of measurement required to achieve signal reconstruction.

In this overview we will not focus on circuit performance (even if a detailed summary of measurements proposed by authors will be provided for every prototype), but on architectural solutions adopted to solve CS issues strictly related to the hardware implementation. When dealing with an actual implementation of a CS system, in fact, and in particular when referring to *analog* CS approaches, additional issues arise that are not encountered when the CS is studied from a signal processing point of view.

With the additional aim of providing a brief overview on CS theory and of introducing the notation used in this chapter, in the following we list the three main issues that any implementation of a CS-based AIC should face.

1. **Time continuity** The peculiarity of the CS paradigm is, in order to avoid infinite-dimension reconstruction problems, to deal with input signals $x(t)$ defined only for $0 \leq t < T_w$. Considering, for example, a discrete-time approach, the signal

$x(t)$, $0 \leq t < T_w$, can be sampled at Nyquist frequency $f_N = 1/T$, with $T_w = n\,T$, giving rise to n samples $\boldsymbol{x}_j = x(j\,T)$ with $j = 0, 1, \ldots, n-1$.

Given this, and referring to the *continuous-time* case, a set of m measurements $\boldsymbol{y} = (\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{m-1})^\top \in \mathbb{R}^m$ of the input signal are computed by integrating the product between $x(t)$ and m different sensing signals $a_j(t)$, $j = 0, 1, \ldots, m-1$. Each sensing function is typically given by pulse-amplitude modulated (PAM) signal, whose pulses length is set by the Nyquist rate, while amplitudes $\boldsymbol{A}_{j,k}$ are stored in the sensing matrix $\boldsymbol{A}$, i.e.,

$$a_j(t) = \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k}\, \chi\left(\frac{t}{T} - k\right) \tag{7.1}$$

where $\chi(\tau)$ is the normalized rectangular function $\chi(\tau) = 1$ when $0 \leq \tau < 1$ and 0 elsewhere. Mathematically, the j-th measurement is given by

$$y_j = \int_0^{T_w} a_j(\tau)\, x(\tau)\, \mathrm{d}\tau. \tag{7.2}$$

In the *discrete-time* approach, the AIC directly deals with the n Nyquist-rate input signal samples $\boldsymbol{x}_k = x(k\,T)$. In this case, by collecting all samples in the vector $\boldsymbol{x} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1})^\top \in \mathbb{R}^n$, the aim of the AIC is to compute m measurements by means of the matricial relation

$$\boldsymbol{y}_j = \boldsymbol{A}_{j,\cdot}\, \boldsymbol{x} \tag{7.3}$$

being $\boldsymbol{A}_{j,\cdot}$ the vector made by the j-th row of the sensing matrix $\boldsymbol{A}$.

Of course, real-world signals are defined for $t \in \mathbb{R}$, or $j \in \mathbb{Z}$. To cope with this, the slicing approach is typically adopted, as explained in Chap. 6 and illustrated in Fig. 7.1 along with a basic block diagram of a practical RMPI architecture both for continuous-time and discrete-time implementation.

In few words, and referring to the continuous-time case regulated by (7.2), the input signal $x(t)$ is sliced into subsequent adjacent blocks $x^{(l)}(t), x^{(l+1)}(t), \ldots$, each of them defined only over a time interval of length T_w, i.e., $x^{(l)}(t) = x(l\,T_w + t)$, $\forall l \in \mathbb{Z}$ and with $0 \leq t < T_w$. With this, the mathematical relation (7.2) can be applied and a set of measurements $\boldsymbol{y}$ can be computed, separately for each signal slice. From a circuital point of view, in order to get the $\boldsymbol{y}$ referred to $x^{(l)}(t)$, it is necessary first to shift in time the $a_j(t)$ functions in order to align them with the currently considered signal slice. Then, each $a_j(t)$ is mixed with $x(t)$, and the result is integrated over a time interval of length T_w, more precisely from the time instant $l\,T_w$ up to the time instant $(l+1)T_w$. After that, the output of the integrator (from a hardware point of view, typically a voltage level or a charge quantity) needs to be transferred to an ADC for the conversion into a digital word.

Yet, at the same time $(l+1)T_w$, the successive slice $x^{(l+1)}(t)$ starts, and all the aforementioned process needs to be repeated. It is reasonable to assume that

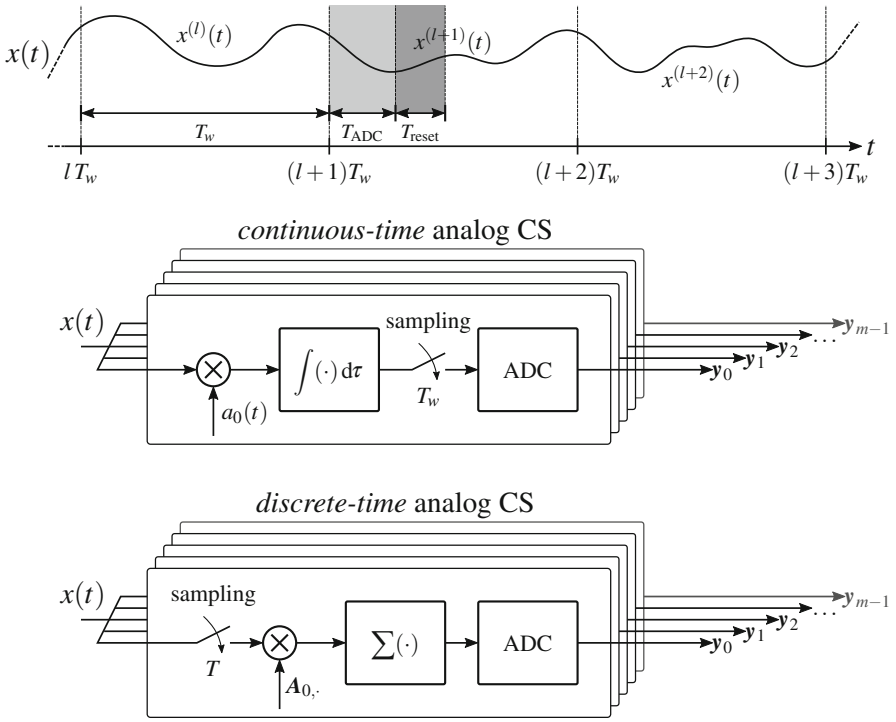


Fig. 7.1 Basic working principle of RMPI architecture, highlighting the slicing process applied to the input signal and the need for a design allowing time continuity in the elaboration of the signal slices, and the two different solutions realized by the *continuous-time* and the *discrete-time* approaches

the slice $x^{(l+1)}(t)$ is processed on the same hardware block used to process the slice $x^{(l)}(t)$; however, three conditions must be satisfied in order to do this. (*i*) The $\boldsymbol{y}_j$ value has to be completely transferred to the ADC. This operation requires a certain amount of time, which we indicate with T_{ADC}. (*ii*) The integrator has to be reset (e.g., the accumulated charge cleared) in order to make measurements of $x^{(l+1)}(t)$ independent of the previous ones computed for $x^{(l)}(t)$. Let us indicate with T_{reset} the amount of time required for this operation. (*iii*) The $a_j(t)$ functions have to be re-shifted to be aligned with $x^{(l+1)}(t)$.

While the third requirement is easily satisfied by periodically repeating the $a_j(t)$ with period T_w, the first two represent a serious problem. In fact both T_{ADC} and T_{reset} are typically non-negligible times, with the consequence that either some information of the input signal is lost, or additional resources need to be included in the design of the RMPI stage.

Of course, the same issue is present for discrete-time architectures, since the sum in (7.3) is typically computed with discrete-time integrators, which require a finite time T_{ADC} for transferring results to ADCs, and finite time T_{reset} to be cleared.

2. **Resource saving** The main goal of an AIC based on CS is, typically, energy saving. Yet, being AICs analog circuits, many classical solutions existing for reducing the power consumption of analog circuits can be applied, such as designing amplifiers working in sub-threshold conduction mode and exploiting technology scaling. Here we are interested in circuital solutions developed *ad-hoc* for reducing the power consumption of an RMPI integrators capable to limit the number of active elements, including trade-offs between complexity and performance of the designed hardware.

   A very simple example is the following one. In order to get m different measurements, both in the discrete-time and in the continuous-time approaches, m (identical) parallel structures are typically replicated and driven with different sensing functions $a_j(t)$ or sensing vectors $\boldsymbol{A}_{j,\cdot}$, with $j = 0, 1, \ldots, m-1$. According to this, a crucial requirement to limit the energy consumption of any AIC is to keep the number m of measurements as low as possible, since the overall energy requirements increases linearly with the parallelism and so with m. In other words, the higher the achievable compression ratio $\mathtt{CR} = n/m$, the lower the energy requirements of the sensing stage. However, as clearly explained in Chap. 1, there is a lower bound for m that is related to the sparsity level κ, typically expressed by means of the mathematical relation

$$m > \mathsf{O}\left(\kappa \log\left(\frac{n}{\kappa}\right)\right) \tag{7.4}$$

   i.e., the minimum number of measurement is almost linearly increasing with κ, and is logarithmically increasing with n.

   In many practical cases the lower bound for m given by (7.4) is too high for allowing a match between the AIC energy consumption and the usually very tight energetic budget available. In other cases, the problem is not given only by energetic constraints, but the lower bound for m is so high that size issues arise.

   To cite another example, both (7.2) and (7.3) require analog multiplications, which are very complex hardware operations. In order to reduce complexity, it is a common strategy to relax the representation of the coefficients $\boldsymbol{A}_{j,k}$ by using a very limited number of bits. As an extreme case, it is also possible to ask that $\boldsymbol{A}_{j,k}$ are 1-bit quantities, i.e., that $\boldsymbol{A}_{j,k} \in \{-1, 1\}$ or $\boldsymbol{A}_{j,k} \in \{0, 1\}$, so that a full multiplier block is not necessary.
3. **Saturation** Despite the fact that sometimes this aspect is not considered, when dealing with an AIC composed by multiple stages (as in the example of Fig. 7.1 where two basic stages, a continuous-time or a discrete-time integrator and an ADC, are present) saturation may occur in any of them.

   This issue has been exhaustively discussed in Sect. 6.4 and can be summarized as follows. The main problem is not that the ADC converter value may fall outside of its conversion range. This event in fact can be easily detected, as these measurements are automatically converted either to the maximum or minimum digital value, and can be ignored by the reconstruction algorithm. This, of course, may reduce the number of available measurement to a value smaller than the

minimum one given by (7.4), impacting correct signal reconstruction. However, reconstruction stage is aware of this situation, and could generate a proper warning message.

Conversely, more serious problems occur when the integrator reaches saturation for a time instant in the middle of the integration interval. In this case, the amplifier used in the integrator block enters a non-linear region, and the measurement $\boldsymbol{y}_k$ is irreparably corrupted. Furthermore, depending on the evolution of the system from the saturation event to the end of the integration interval, it may happen that the integrator output value fall in the ADC conversion range. In this case reconstruction stage is not aware of the erroneous measurement unless additional dedicated hardware is used.

7.2 AIC for Radar Pulse Signals by Yoo et al., 2012

The first circuit considered in this overview is a 100 MHz–2 GHz radar pulse receiver whose working principle and preliminary measurement results from a signal processing point of view have been presented at the *IEEE International Conference on Acoustics, Speech and Signal Processing* [28], held in Kyoto, Japan, 25–30 March, 2012, while detailed hardware description and measurements appeared a few months later at the *IEEE Radio Frequency Integrated Circuits Symposium* [27], Montreal, Canada, 17–19 June, 2012. In a joint work by the California Institute of Technology, CA, and Stanford University, CA, authors designed an RMPI analog preprocessor in CMOS 90 nm technology including 8 elaboration channels, resulting in an AIC with a dynamic range of 54 dB while digitizing measurement samples at a rate of 320 Ms/s, that is a factor 12.5 below the Nyquist rate $f_N = 4$ GHz. Total area occupation is 8.85 mm^2, while the power consumption is evaluated in 506.4 mW excluding the analog output buffers.

The microphotograph of the integrated circuit, taken from [27], is depicted in Fig. 7.2, while the simplified block diagram is depicted in Fig. 7.3. Basically, the circuit consists of 8 RMPI parallel channels with a common input node driven by a shared low-noise amplifier (LNA). Each channel includes a passive mixer capable of multiplying the input signal by a PAM sensing signal whose amplitudes $\boldsymbol{A}_{j,k} \in \{-1, +1\}$ are stored as 1-bit values in a local digital memory. The system is validated by using single tones and radar pulses as test signal.

7.2.1 Hardware Architecture

A simplified block diagram of the architecture of the AIC described in [27] is reported in Fig. 7.3. A LNA with 18 dB gain and 3 GHz bandwidth works as input stage. The output of the LNA drives $M = 8$ parallel RMPI channels, whose design

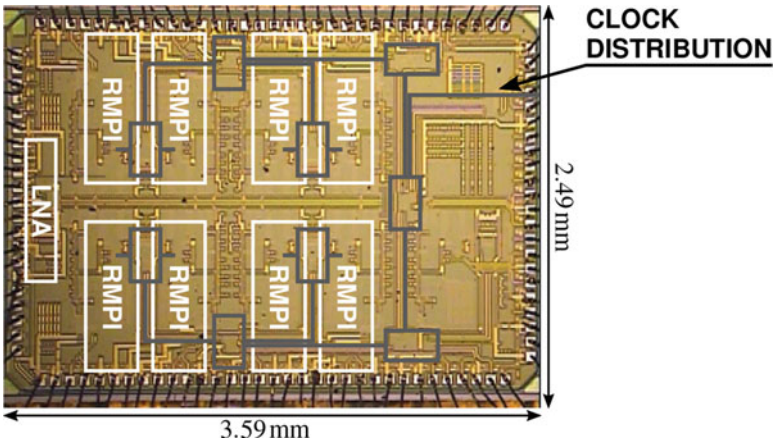


Fig. 7.2 Die microphotograph of the integrated circuit considered in Sect. 7.2 (adapted from [27])

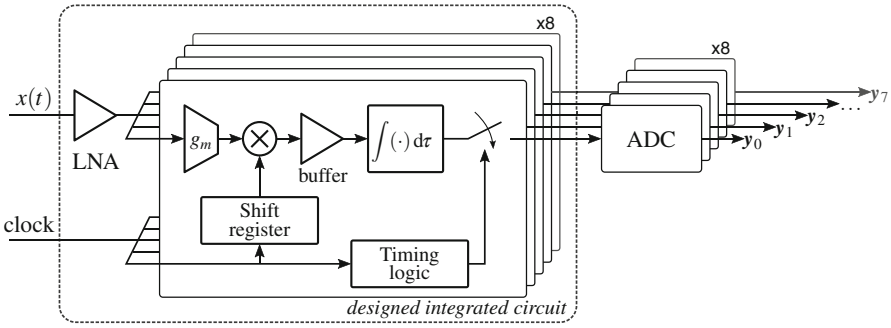


Fig. 7.3 Simplified block diagram of the architecture of the integrated circuit considered in Sect. 7.2

is based on a fully differential architecture, and it is almost identical to that of a standard RF receivers exploiting frequency down-conversion using a current-domain approach [3].

In detail, the large voltage-amplitude output of the LNA is converted into a large current-domain signal with a transconductor amplifier with gain g_m. While the LNA is shared, each RMPI channel has its own transconductor amplifier in order to reduce cross-talk among channels. The generated current signals are then mixed by standard analog passive mixers, where the port usually connected to the local oscillator (LO) in a classic down-conversion-based RF receiver is now connected to a programmable 128-bit shift registers playing the role of a serial memory. In other words, the LO ports of the k-th RMPI channel is driven by the PAM voltage signal that can be mathematically modeled as $a_j(t) = \sum_{k=0}^{N-1} \boldsymbol{A}_{j,k}\, \chi(t/T - k)$, where $\chi(t)$ is the normalized rectangular function $\chi(t) = 1$ when $0 \leq t < 1$ and 0 elsewhere,

where $T = 1/f_N$, and where the $\boldsymbol{A}_{j,k} \in \{-1, +1\}$ are the 1-bit digital values stored in the shift register. In such a way, the designed integrated circuit implements an *antipodal* RMPI architecture.

The use of a programmable shift register for storing the sensing vector has mainly two reasons. First of all, this solution has been chosen for testing purposes, easily allowing to load any bit sequence of any length up to 128 bit. The other reason is due to speed issues. The mixing with the sensing sequence needs to be at the Nyquist rate, that is in this case $f_N = 4\,\text{GHz}$. Using an internal serial memory is the only solution allowing versatility at this speed.

The current signal output by the mixer is first buffered for noise purposes, and then elaborated by the cascade of a class-A op-amp-based transconductance RC-integrator and a buffer that serves as the integrator stage in the RMPI stage and as the driver for the off-chip ADC, respectively.

Mathematically, this case belongs to continuous-time analog CS class (*case A* accordingly to the definition of Chap. 6). Indicating with $x(t)$ the differential voltage signal at the output of the LNA and assuming all buffers are unit gain, the mixer takes as input the current signal $g_m x(t)$ and the voltage signal $a_j(t) = \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k}\, \chi(t/T - k)$. The product is fed into the integrator input. Indicating with $T_i = N\,T$ the integration time, and assuming that the integrator gain is $1/C$ (from a dimensional point of view, must be the inverse of a capacitance), the j-th measurement is given by the output of the j-th integrator at the end of the integration time

$$\boldsymbol{y}_j = \int_0^{T_i} g_m\, x(\tau) \frac{1}{C} \sum_{k=0}^{N-1} \boldsymbol{A}_{j,k}\, \chi\left(\frac{\tau}{T} - k\right) \mathrm{d}\tau = \frac{g_m\, T}{C} \sum_{k=0}^{N-1} \boldsymbol{A}_{j,k} \tilde{\boldsymbol{x}}_k \tag{7.5}$$

where the $\tilde{\boldsymbol{x}}_k, k = 0, 1, \ldots, N-1$ implicitly defined in (7.5) represent the generalized Nyquist samples as detailed in Chap. 6, with $\tilde{\boldsymbol{x}}_j \approx x(j\,T)$ for a quasi-stationary signal. The dimensionless constant $G = g_m\, T/C$ represents the gain of the system; by introducing $\tilde{\boldsymbol{x}} = (\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_{N-1})^\top \in \mathbb{R}^N$ as the vector containing all the generalized Nyquist samples, we can write (7.5) in the more compact and usual notation $\boldsymbol{y}_j = G \boldsymbol{A}_{j,\cdot} \tilde{\boldsymbol{x}}$.

While both papers [28] and [27] detail many aspects of the designed integrated circuit, with particular emphasis on circuital solutions adopted to cope with problems arising in an AIC dealing with input signal with Nyquist frequency $f_N = 4\,\text{GHz}$, no time continuity mechanism is mentioned, nor saturation aspects are considered. Conversely, a hybrid RD-RMPI approach for signal reconstruction is adopted and detailed in both papers.

In each integrated circuit, only $M = 8$ RMPI channels have been placed. This number is actually too small to cope with the minimum number of measurements required for correct reconstruction given by (7.4). As a workaround, each time windows $T_w = n\,T$ is split into a number q of continuous-time intervals with length $T_i = N\,T$, namely *integration windows*, with $T_w = q\,T_i$ and consequently $n = qN$.

In each integration window a number M of measurements are taken from the M integration paths. In a time T_w, an amount of measurement equal to $m = qM$ is generated. This approach, known as hybrid RD/RMPI, has been detailed in Sect. 6.3, and allows performance similar to that of a full RMPI implementation, even with a limited number of integration channels.

This choice has also been pushed by a pure practical reason. Even with a number of RMPI channels limited to $M = 8$, and considering that $f_N = 4\,\text{GHz}$, timing differences can be significant, and differences in time and phase delays may hurt system performance. In order to minimize the timing differences, minimize jitter at the mixer, and minimize power consumption, the system clock is distributed in a binary symmetric tree topology that has been highlighted in Fig. 7.2. This has been detailed in [27].

In conclusion, we can summarize here the main aspects of the proposed architecture.

1. **Time continuity:** No time continuity mechanism is mentioned in [28] nor in [27].
2. **Resource saving:** Antipodal mode; hybrid RD-RMPI architecture.
3. **Saturation:** No saturation checking mechanism is mentioned in [28] nor in [27].

7.2.2 Experimental Results

Testing results for the designed prototype have been included both in [28] and in [27]. Here, we propose a selection of results taken from both papers.

In all measurements, the input signal has been generated by an arbitrary waveform generator, assuming a Nyquist frequency $f_N = 1/T = 4\,\text{GHz}$. The integration time has been set to $T_i = NT = 25\,\text{ns}$, i.e., measurements $\boldsymbol{y}_j$ are sampled at the rate $1/T_i = 40\,\text{MHz}$, so $N = 100$. The outputs of the RMPI channels are digitized off-chip by an external ADC, and then exported to a PC for reconstruction. The actual number of bits used in the digitization is not declared by authors. All measurements are achieved exploiting the full parallelism of a single integrated circuit, i.e., $M = 8$, leading to a measurement rate of $M/T_i = 320\,\text{MS/s}$, with a compression ratio, in terms of number of samples required for correct reconstruction, equal to $\texttt{CR} = 12.5$.

In all experiments, a hybrid RD-RMPI approach is used, collecting a number of measurements $m = qM$ in a time $T_w = qT_i = nT$ before each reconstruction. However, the actual value of the time window T_w used in the reconstruction algorithm (and therefore, of m and of n) is not specified. The digitized samples are used to reconstruct the input signal via a numerical optimization procedure and the input signal has been reconstructed using a variant of basis pursuit with reweighting [4].

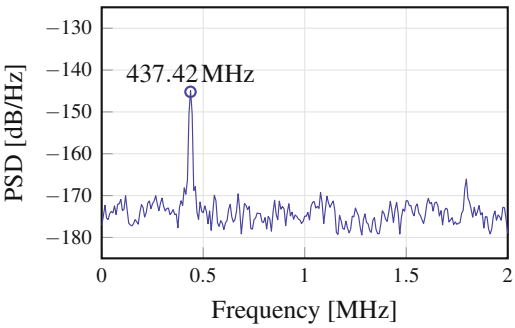


Fig. 7.4 Power spectrum density (PSD) of the reconstruction of a signal made of a 400 μV peak-to-peak amplitude single tone with frequency 437.5 MHz for the AIC considered in Sect. 7.2 (adapted from [27])

In the tests, two different stimuli have been considered as input signal.

The first one is given by a single tone low-amplitude sinusoidal signal used to verify the dynamic range achievable by the system. Figure 7.4 shows the power spectrum density (PSD) of the reconstruction of a signal made of a single tone with 400 μV peak-to-peak amplitude. The deviation of the measured frequency (437.42 MHz) with respect to the actual frequency (437.5 MHz) is negligible, proving the good behavior of the AIC in detecting very small amplitude sinusoidal tones. The dynamic range is evaluated in 54 dB.

In the second and more realistic test, pulses of multiple widths and frequencies are taken into account. In Fig. 7.5 two cases are considered, showing the envelope and the power spectrum (PSD) of reconstructions of signals made of 400 ns pulses compared with the original ones. Two cases are considered where carrier frequency is set to the two endpoints of its theoretical working frequency band, i.e., about 87 MHz and about 1947 MHz. The visually correct reconstruction shows the correct behavior of the CS system without any change in operating conditions (e.g., tuning of the sensing sequences). The frequency estimation error in all considered cases is negligible, since its average value is smaller than 69 kHz.

A more challenging test is represented in Fig. 7.6, where two overlapping pulses at different frequencies are considered, that accordingly to [27] is a signal that is difficult to handle even from standard Nyquist-rate receivers. The figure compares the reconstructed envelopes and the power spectra with the original ones in a case with two pulses of length 400 ns and frequencies 275 MHz and 401 MHz present an overlap (in time) equal to 200 ns. In the reconstructed signal, the carrier frequency of both pulses is estimated within an error of 234 kHz, while the mean-square error of the pulse-envelope reconstruction is less than 10%.

Finally, the limits of the proposed architecture are shown in Fig. 7.7 where short pulses are considered. The figure shows the reconstructed envelope of a 50 ns and of a 75 ns pulses. Despite the fact that the pulse-envelope reconstructions are of low quality, what is notable is that an accurate frequency estimation (evaluated in only 1.4 MHz) is possible in both cases starting from 16 and 24 compressed measurements, respectively.

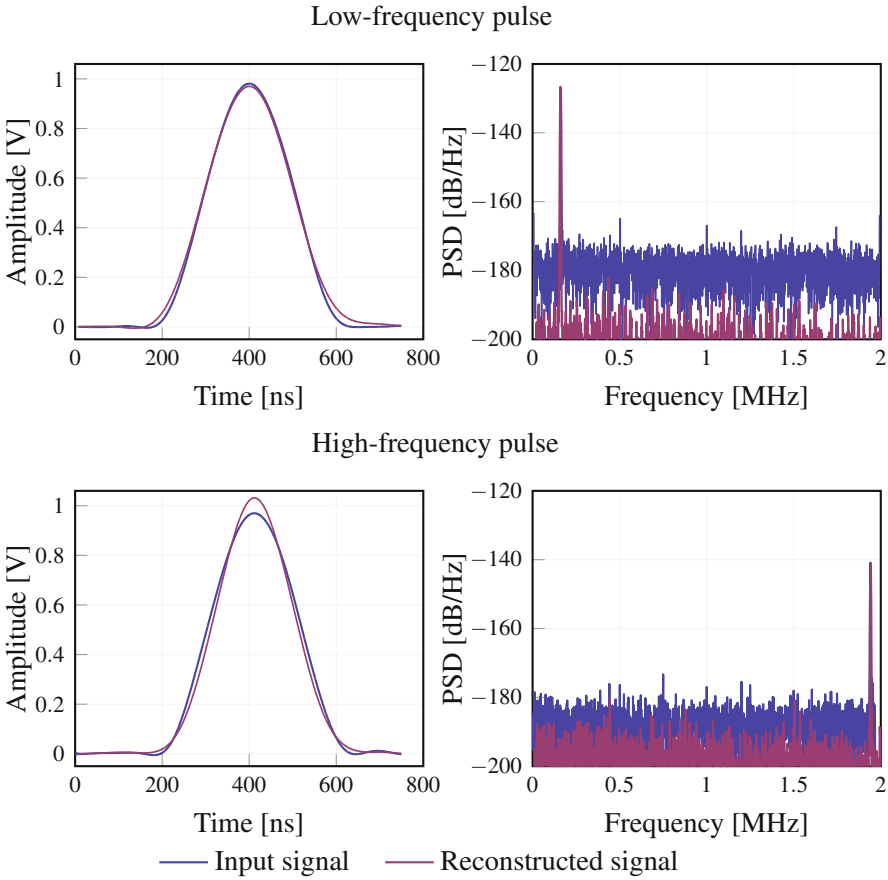


Fig. 7.5 Comparison between envelope amplitude and PSD of original and reconstructed signal made of 400 ns frequency pulses, in the case the carrier frequency is set to about 87 MHz (top plots) and to about 1947 MHz (bottom plots) for the AIC considered in Sect. 7.2 (adapted from [27])

7.3 AIC for Wideband Multi-tone BPSK Signals by Chen et al., 2012

In the September 2012 *Special Issue on Circuits, Systems and Algorithms for Compressive Sensing* of the *IEEE Journal of Emerging and Selected Topics in Circuits and Systems*, Chen et al. published the results of a joint work of the Department of Electrical Engineering, Texas A&M University, College Station, and of the Army Research Laboratory (ARL) in Adelphi, MD [8]. Preliminary simulation results were previously published in a 2011 issue of the *IEEE Transactions on Circuits and Systems–I: Regulars Papers* [9].

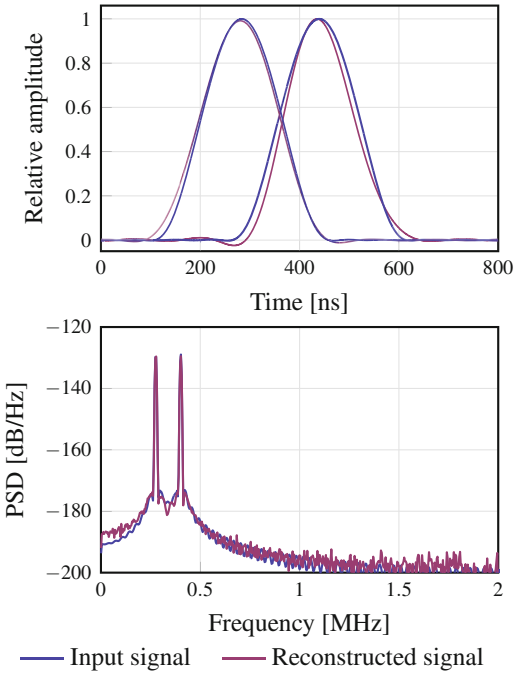


Fig. 7.6 Comparison between envelope amplitudes and PSD of original and reconstructed signal made of two 400 ns superimposing frequency pulses, with carrier frequency equal to 275 MHz and 401 MHz, respectively, for the AIC considered in Sect. 7.2. Envelope amplitudes have been normalized to the corresponding input signal ones (adapted from [27])

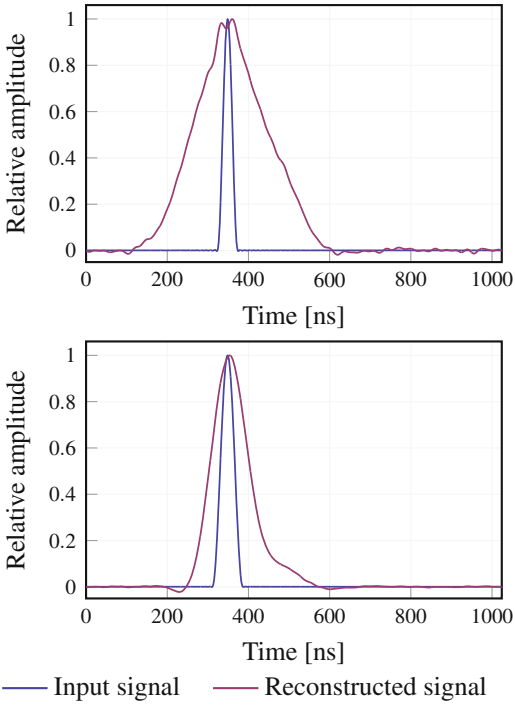


Fig. 7.7 Comparison between envelope amplitudes of original and reconstructed signal made of short frequency pulses (50 ns width on the left, 75 ns on the right) for the AIC considered in Sect. 7.2. Envelope amplitudes have been normalized to the corresponding input signal ones (adapted from [27])

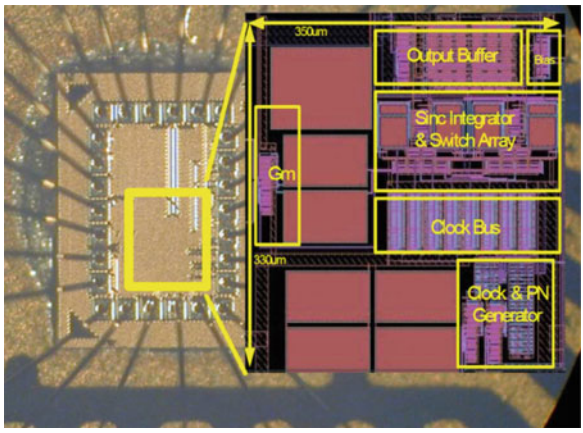


Fig. 7.8 Die photograph of the integrated circuit considered in Sect. 7.3 (adapted from [8])

The microphotograph of the integrated circuit, designed in 90 nm CMOS process, has been depicted in Fig. 7.8. The active area is $350 \times 330\,\mu$m and includes a single RMPI channel, whose simplified schematic is depicted in Fig. 7.9. The circuit has been designed to virtually target multi-tone binary phase-shift keying (BPSK) signal with a 1.5 GHz instantaneous signal bandwidth (equivalent Nyquist frequency 3 GS/s), but experimental characterization with only a signal bandwidth of 500 MHz has been reported by authors due to limitations in testing equipment. A complete CS system employing eight instances of the designed circuit in a single board has been tested. The testing Nyquist rate has been set to $f_N = 1.25$ GHz that is a value somewhat above the minimum required for a 500 MHz bandwidth signal.

Each integrated circuit includes the RMPI block composed by a mixer, an integrator, and a pseudorandom generator producing the $\boldsymbol{A}_{j,k}$ coefficients (an 11-bit linear feedback shift register). In this design $\boldsymbol{A}_{j,k} \in \{-1, +1\}$, i.e., an antipodal RMPI is realized. The ADC is not implemented on-chip; instead, a high-speed digital oscilloscope is used to digitize the $\boldsymbol{y}_j$ and to transfer them to a PC where signal reconstruction is achieved by using Matlab software.

A single observation time window is set to $T_w = 1/5$ MHz $= 200$ ns, delivering the $m = 72$ samples. This leads to a system throughput of 360 MS/s, equivalent to 28.8% of the Nyquist rate with an equivalent compression ratio of about $\mathtt{CR} = 3.5$. The overall power consumption of the CS system is 54 mW for the on-chip components, plus the power of the external ADC and of the system clock generator.

7.3.1 Hardware Architecture

The schematic of the core of the AIC architecture proposed in [8] has been depicted in Fig. 7.9. Basically, it is a differential circuit composed of a voltage/current converter (the g_m stage), followed by a digital mixer, which is actually made with

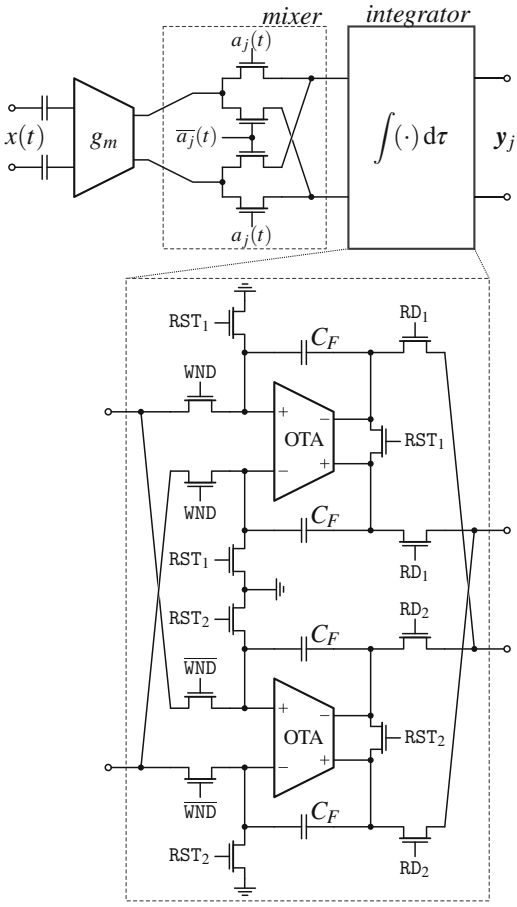


Fig. 7.9 Simplified schematic of the RMPI circuit embedded in the integrated circuit considered in Sect. 7.3

a few pass-transistors allowing either a direct path or an inverted path (i.e., the two differential lines can be inverted) in order to implement the multiplication by the $\boldsymbol{A}_{j,k} \in \{-1, +1\}$. The mixed signal is used as input signal for two OTA-based integrators working in interleaved mode.

The two integration paths are alternatively accumulating the input current into two couples of differential feedback capacitances C_F to solve the time continuity issue. Let us assume that, initially, the time window selection signal WND is high, while the two reset signals RST_1 and RST_2 are low and high, respectively. In this configuration, the second integrator is in clearing mode to remove all the charge accumulated on the C_F (thus forcing to zero the output voltage), while the first one is in integrating mode being connect to the mixer output. At the end of the integration time T_i, both WND and RST_2 go low and the signal from the mixer is disconnected from the first integrator, which starts to operate as a hold circuit retaining its voltage value. At the very same time the second integrator is switched

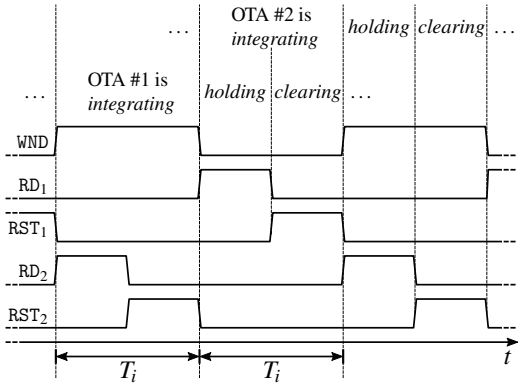


Fig. 7.10 Time diagram for the signals regulating the behavior of the two integration paths of Fig. 7.9 in the system considered in Sect. 7.3

from clearing to integrating mode by routing the signal from the mixer to its input, so that a new integration window can immediately start on the second path. This allows enough time to perform both the conversion of the charge accumulated into the first integrator (signal $\mathtt{RD}_1$ asserted), in the proposed circuit by means of transferring the accumulated voltage to an external ADC, and then a complete removal (signal $\mathtt{RST}_1$ asserted) of the charge accumulated on C_F. The only, non-stringent, requirement is that these two operations have to be completed in a time smaller than T_i to allow the first path to start a new integration period as soon as the second one enters the hold mode. A time diagram showing the signals controlling all these operations is depicted in Fig. 7.10.

With this approach it is easy to ensure integration time continuity at the cost of replicating only part of the active circuit (the integrator), while all other parts are shared among the two paths.

From a mathematical point of view this CS system, similarly to the previously considered case, is a continuous-time analog one, referred to as *case A* accordingly to the notation of Chap. 6. The input signal $x(t)$, assumed to be the differential voltage at the input port, is first transformed into a current signal $g_m\, x(t)$, then mixed with the $\boldsymbol{A}_{j,k}$ symbols by means of the pass-transistors of Fig. 7.9. This operation can be modeled as the multiplication between the $g_m\, x(t)$ and the PAM signal $a_j(t) = \sum_{k=0}^{N-1} \boldsymbol{A}_{j,k}\, \chi(t/T - k)$. The measurement $\boldsymbol{y}_j$ is given by the voltage level at the output of the integrator, sampled after a time $T_i = N\,T$, so that

$$\boldsymbol{y}_k = \int_0^{T_i} g_m\, x(\tau) \frac{1}{C_F} \sum_{k=0}^{N-1} \boldsymbol{A}_{j,k}\, \chi\left(\frac{\tau}{T} - k\right) \mathrm{d}\tau = \frac{g_m\, T}{C_F} \sum_{k=0}^{N-1} \boldsymbol{A}_{j,k} \tilde{\boldsymbol{x}}_k \tag{7.6}$$

where the $\tilde{\boldsymbol{x}}_k$, $k = 0, 1, \ldots, N-1$ represent the generalized Nyquist samples accordingly to the definition of Chap. 6. As in the previous case, by defining the vector $\tilde{\boldsymbol{x}} \in \mathbb{R}^N$ of the generalized Nyquist samples, and the dimensionless constant $G = g_m\, T/C_F$ as the gain of the system, we can write (7.6) in the more usual and compact notation $\boldsymbol{y}_k = G\boldsymbol{A}_{j,\cdot}\, \tilde{\boldsymbol{x}}$.

The system tested in [8], for the sake of energy reduction, embeds only a limited number $M = 8$ of analog RMPI cores, so it is able to produce 8 measurements in each integration window T_i. According to (7.4), this value is typically too small to ensure a correct signal reconstruction. In order to increase the available number m of measurements with respect to M, and similarly to previously considered system, authors propose a hybrid RD-RMPI approach.

In other words, and as detailed in Sect. 6.3, each time window $T_w = n\,T$ is split into a number q of continuous integration windows of length $T_i = N\,T$, with $T_w = q\,T_i$ and $n = qN$. Since in each T_i, a number M of measurements are collected, reconstruction is a time window T_w based on a total of $m = qM$ measurements. This is reflected in a highly structured, block diagonal overall sensing matrix $\boldsymbol{A}$, and allows performance similar to that of a full RMPI approach even with a small number of integration paths [29].

In conclusion, we can summarize here the main aspects of the proposed architecture.

1. **Time continuity:** Integrator is duplicated to allow time continuity.
2. **Resource saving:** Antipodal mode (no need for analog multiplier); hybrid RD-RMPI architecture.
3. **Saturation:** No saturation checking mechanism is mentioned neither in [8] nor in [9].

7.3.2 Experimental Results

The input considered in [8] is a wideband multi-tone BPSK signal with up to 100 carriers allocated from 5 to 500 MHz with 5 MHz frequency spacing between adjacent carriers. The sparsity is up to 4%, meaning that there are at most four active tones in a given sampling interval. Input signal is externally generated, along with the master clock for the pseudorandom generator at $f_N = 1.25\,\text{GHz}$. The periodic triggering signal to reset pseudorandom generator initial state is also externally provided.

In the system proposed for testing in [8] authors set $M = 8$ to limit the power consumption to 54 mW. Furthermore, aiming at $T_w = 1/5\,\text{MHz} = 200\,\text{ns}$, authors set $T_i = 1/45\,\text{MHz} \approx 22.2\,\text{ns}$ and $q = 9$. This leads to a sampling rate of the external DAC connected to each channel equal to $45\,\text{MS/s}$, and a total amount of measurement available in each signal slice equal to $m = qM = 72$. Since the Nyquist frequency is $f_N = 1.25\,\text{GHz}$, it is $n = f_N\,T_w = 250$.

Results for a sparsity level $\kappa = 1$, i.e., considering a single sinusoidal tone whose frequency is swept in the 50–450 MHz frequency range, have been depicted in Fig. 7.11. Results are presented in terms of signal-to-noise and distortion ratio (SNDR), defined as the overall signal power over the noise power of all other undesired frequency components, including the overall integrated noise power and harmonic distortion components. The reconstruction algorithm is the one described in [29] and based on OMP technique. The maximum achievable SNDR with the

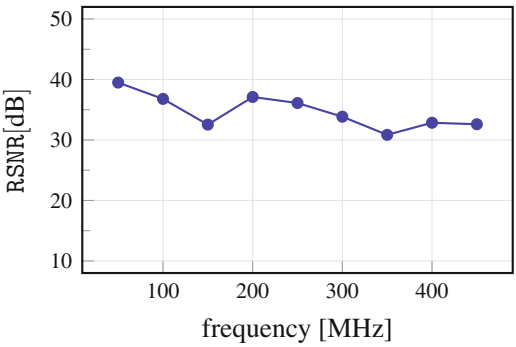


Fig. 7.11 Reconstructed single tone SNDR *vs* input tone frequency in experimental results proposed in [8]

Table 7.1 Reconstructed SNDR with multi-tone BPSK input signals in experimental results proposed in [8]

Case	Input frequencies (MHz)	RSNR (dB)
#1	50, 250, 490	29.3
#2	50, 250, −490	29.6
#3	20, 70, 250, 450	27.7
#4	50, 150, 250, 490	29.4
#5	−20, −70, 250, 450	29.5

single tone test is around 40 dB. The SNDR degrades gradually down to 34 dB as the frequency increase because of the front-end gain roll-off at high frequency.

An actual BPSK modulated signal, i.e., a multi-tone (with $\kappa = 3$ or $\kappa = 4$) wideband signal where tones have either a 0° phase or 180° phase accordingly to the encoded symbols, has also been used for system characterization. This signal has been generated by an arbitrary waveform generator, and an external magnitude equalizer has also been employed to allow the different carrier tones of the test signal to have unequal amplitudes. Some of the obtained results are given in Table 7.1, where a positive frequency has the meaning of a 0° phase shift, while a negative one that of a 180° phase shift. The input frequency components were successfully located by the CS algorithm, and unequal carrier amplitudes agree with the original input spectrum. In this case, however, the maximum achievable SNDR is about 10 dB lower than that of the single tone test.

The effective number of bits (ENOBs) of the CS data acquisition system has been estimated by authors in up to 6.4 bit from the 40 dB SNDR achieved by the single tone sweep. This value should be decreased by 1–2 bits when considering the multi-tone environment.

7.4 AIC for ECG Signals by Gangopadhyay et al., 2014

This circuit appeared in a 2014 issue of the *IEEE Journals of Solid-State Circuits* [12]. Gangopadhyay et al. presented description and achieved results of a fully integrated low-power CS analog front end for an ECG sensor. Switched-capacitor

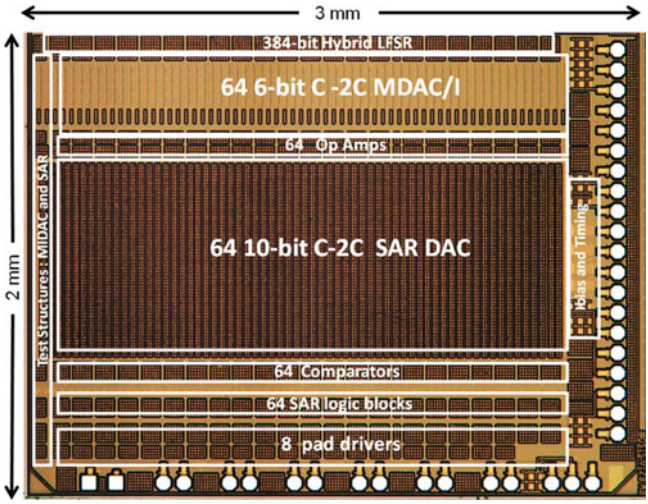


Fig. 7.12 Microphotograph of the integrated circuit described in Sect. 7.4 (adapted from [12])

circuits are used to achieve high accuracy and low power. The prototype has been implemented in 0.13 μm CMOS technology, embedding a 384-bit Fibonacci–Galois hybrid linear feedback shift register for generating sensing matrix elements $\boldsymbol{A}_{j,k}$ and 64 RMPI channels, each of them consisting of a switched-capacitor 6-bit C-2C multiplying DAC/integrator (MDAC/I) and of a 10-bit C-2C SAR ADC. The chip size is 2×3 mm (each channel has a height of $\simeq 36\,\mu$m) and, when clocked at 2 kHz, the total power dissipation, mainly of static power, is 28 nW and 1.8 μW for one and 64 active channels, respectively. The microphotograph of the integrated circuit is reported in Fig. 7.12. The core of the circuit is a differential 6-bit C-2C MDAC/I circuit whose simplified schematic is depicted in Fig. 7.13. The circuit is basically a switched-capacitor integrator where the sampling capacitor has been replaced by a C-2C network, thus being capable of performing at the same time both a multiplication by a 6-bit digital value and the integration. This approach allows the proposed circuit, differently from that considered up to now, to implement a sensing matrix with real values and not only with binary ones. The value of n is externally programmable among many values ($n \in \{128, 256, 512, 1024\}$).

7.4.1 Hardware Architecture

The analog core of the circuit proposed in [12] is the differential 6-bit C-2C MDAC/I circuit whose simplified schematic is reported in Fig. 7.13. Basically, the circuit is a switched-capacitor integrator, whose behavior is regulated by the two non-overlapping clock signals ϕ_1 and ϕ_2. When ϕ_1 is high (and ϕ_2 is low), the circuit is in a sampling mode. The op-amp inverting pin is forced to reference voltage

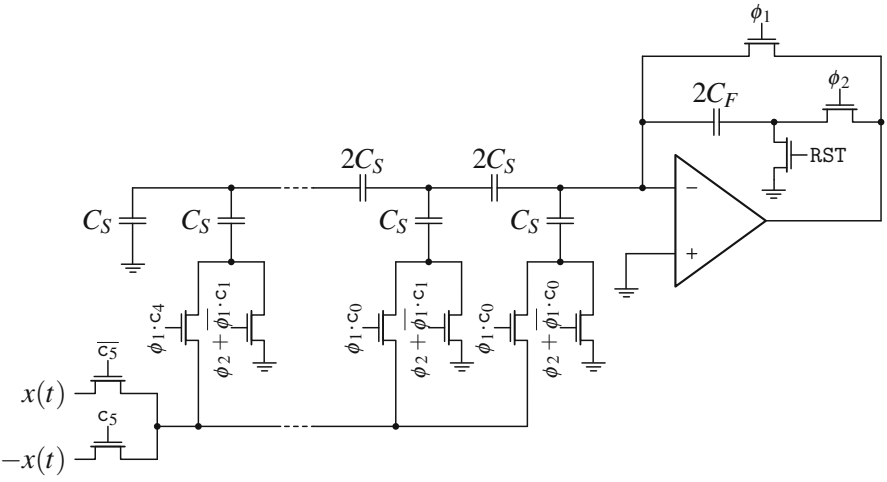


Fig. 7.13 Basic schematic (single-ended simplification) of the C-2C MDAC/I circuit embedded in each RMPI channel of the integrated circuit of Fig. 7.12 described in Sect. 7.4

(virtual short on op-amp input pins), and the input voltage is connected to the $C-2C$ ladder, which has a unit capacitance C_S and it is controlled by the 5-bit digital word $c_4\, c_3\, c_2\, c_1\, c_0$. An additional bit c_5 decides if the input voltage is connected directly or reversed (i.e., the two differential line are inverted) setting in this way the sign of the input signal. Note that the switched-capacitor approach makes this stage working as a sampling circuit. In fact, by indicating with $\hat{t}$ the time instant in which ϕ_1 has a high-to-low transition, the C_S ladder is actually loaded by the voltage level $\pm x(\hat{t})$. Due to this, and even if directly connected to $x(t)$, this circuit belongs to the *analog discrete-time* class, i.e., *case B* accordingly to the definition of Chap. 6. During the sampling mode, the feedback capacitor of value $2\,C_F$ is disconnected from the op-amp, and the charge previously accumulated in it is unaltered.

In the accumulation mode ϕ_2 is raised (and ϕ_1 lowered) and (part of) the charge stored into the $C-2C$ ladder is transferred to the feedback capacitance $2\,C_F$, re-connected to the op-amp. Due to the $C-2C$ architecture, only a fraction of the maximum charge is transferred to the feedback capacitance. More precisely, indicating with c the 6-bit number whose sign is given by c_5 and value determined by $c_4, \ldots, c_0$, and normalized such that $-1 < c < 1$, the charge transferred to the feedback capacitance is $2x(\hat{t})\, c\, C_S$.

When this process is repeated n times, indicating with $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}$ the value of the input signal at the n high-to-low transitions of ϕ_1, and assuming that at every cycle the value of c is changed according to the sensing matrix $\boldsymbol{A}$ (for the j-th accumulator at the k-th cycle it is $c = \boldsymbol{A}_{j,k}$) the voltage at the MDAC/I output is given by

$$\boldsymbol{y}_j = \frac{C_S}{C_F} \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k}\, \boldsymbol{x}_j = G \boldsymbol{A}_{j,\cdot}\, \boldsymbol{x}$$

where $\boldsymbol{x}$ is the vector of the input signal samples at the MDAC/I input, and where the implicitly defined dimensionless constant G is the gain of the integrator stage, which has been set to $\sim 1/3$ to prevent saturation at the output during integration.

In this way the MDAC/I is capable of approximating an RMPI channel controlled by a sensing matrix $\boldsymbol{A}$ made of real coefficients. The approximation of $\boldsymbol{A}$ entries with a 6-bit quantized values has been found to be enough for accurate reconstruction [1].

In the proposed circuit, in order to avoid the implementation of a dedicated large memory block for the $\boldsymbol{A}$ (assuming $n = 256$ and $m = 64$, it would require almost a 100-Kbit memory), coefficients ⊂ are generated with an on-chip hybrid linear feedback shift register (LFSR). Basically, 64 6-bit *Fibonacci* LFSRs have been integrated into the circuit, one for each of the $C - 2C$ MDAC/I circuits, and outputting the 6-bit coefficients programming the 6-bit MDAC. Then, these 64 LFSRs are further randomized by dithering their less significant bits (LSBs) in a *Galois* fashion, each LFSR using the MSBs of another stage. In this way, a Fibonacci–Galois 384-bit LFSR is designed. An external trigger signal enables a 384-bit seed load at the beginning of each integration frame.

With this generator, sensing matrices $\boldsymbol{A}$ whose random elements have two different statistical distributions can be generated.

1. 6-bit uniform distribution;
2. 1-bit Bernoulli (i.e., antipodal) distribution, achieved by setting only the most significant bit (MSB) of ⊂ by using the 6-bit LFSRs, and by forcing all other bits to 1.

Finally, a SAR ADC is connected to each RMPI channel to provide a digital representation of the measurement. At the end of each time frame, the accumulated charge is transferred to the DAC, and then the residual charge of each MDAC/I is cancelled. A $C - 2C$ SAR ADC [2] is used, thus minimizing the dynamic power dissipation and eliminating the input sampling buffer. To achieve an 8-bit ENOB, a 10-bit $C - 2C$ DAC is implemented. According to [12], time continuity between consecutive time signal slices is ensured by connecting the MDAC/I and the ADC in a pipeline way; however, no further details on this aspect are provided.

The aforementioned hardware is replicated $m = 64$ times (i.e., with $j = 1, 2, \ldots, m - 1$) in a single integrated circuit to allow generating up to $m = 64$ measurements every $T_w = n\,T$ time units.

The main features of the proposed architecture are as follows:

1. **Time continuity:** pipelining between the MDAC/I and the following SAR ADC.
2. **Resource saving:** embedding MDAC and integrator in a single op-amp circuit. On-chip LFSR for MDAC programming.
3. **Saturation:** No saturation checking mechanism is mentioned. Only reduced integrator gain.

7.4.2 *Experimental Results*

Testing results proposed in [12] involve both a synthetic input signal and realistic signals, and the use of different values of $n = 128, 256, 512$, and 1024, and different value of $m = 1 - 64$. The clock and input signal have been generated by arbitrary waveform generators, while digital words output from the 64 ADC stages are connected to a logic analyzer. Input signals are reconstructed using a standard basic pursuit algorithm on MATLAB environment.

Measurements on the SAR ADC show a SNDR of 40.6 dB, equivalent to an effective number of bits of 6.5-bit for a 200 Hz bandwidth signal. On-chip calibration was not implemented.

Measured results when using a two-tones sinusoidal signal are shown in Fig. 7.14. An input signal composed by the superimposition of two sinusoids (28 Hz and 50 Hz) is used. Signal has been reconstructed using $n = 256$ and $m = 64$, $m = 32$ and $m = 13$, i.e., a compression ratio of $\texttt{CR} = 4$, $\texttt{CR} = 8$, and $\texttt{CR} = 20$, respectively. The signal is reconstructed using a Fourier sparsity basis and a CVX ℓ_1-norm convex optimization [14]. According to the figure, the two-tone sinusoidal signal is well reconstructed, with an error between -80 LSB and 100 LSB. Performance results in terms of SNR are not provided.

Results when using a realistic ECG signal input (waveforms taken from the PhysioBank database [13]) are shown in Fig. 7.15. The ECG signal has been

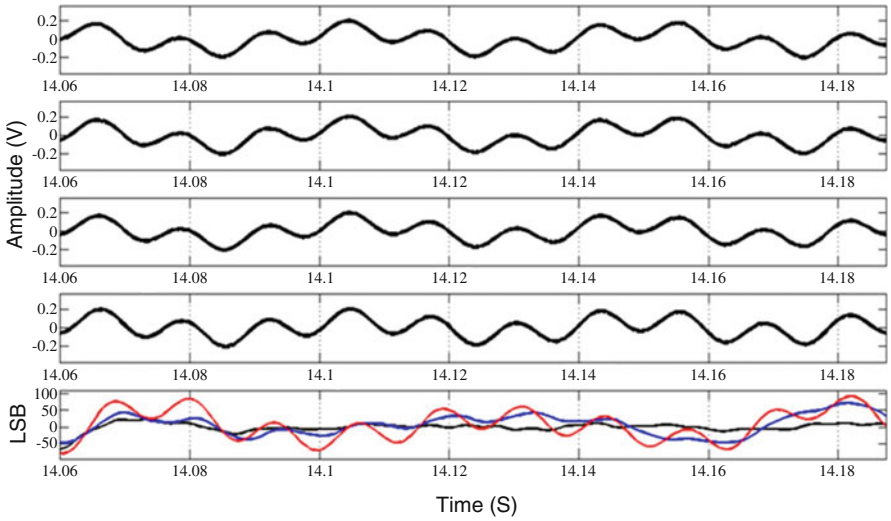


Fig. 7.14 Measured reconstructions for a two-tone signal (28 Hz and 50 Hz sinusoids) for the circuit proposed in [12]. From the top: raw signal; reconstructed waveforms for $n = 256$ and $m = 64$, $m = 32$, and $m = 13$, respectively, with compression factor equal to 4, 8, and 20, respectively (adapted from [12])

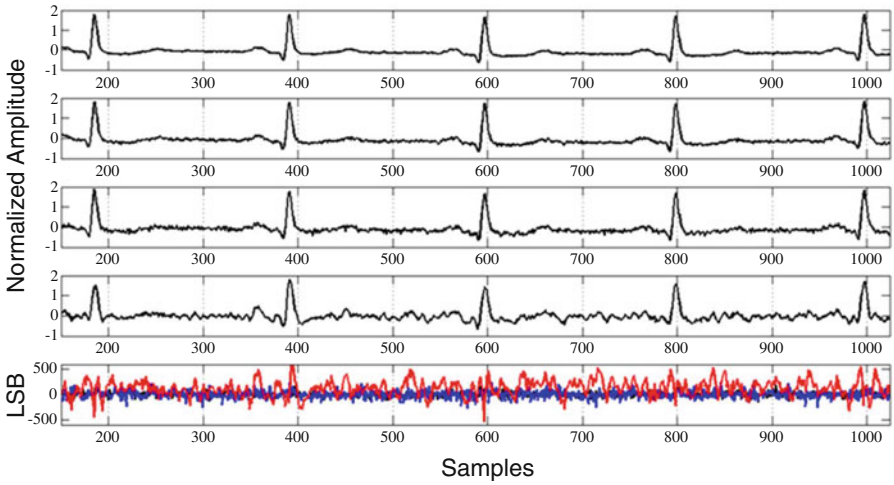


Fig. 7.15 Measured reconstructions of a synthesized ECG signal sparse in the Daubechies-4 wavelet domain using eight frames with $n = 128$ samples each. From the top: raw ECG; reconstructed waveforms with $m = 64$, $m = 32$, and $m = 10$, respectively, with compression factor equal to 2, 4, and 6, respectively (adapted from [12])

compressed by the proposed circuit and reconstructed using a wavelet basis as sparsity basis derived from the Daubechies db4 mother-wavelet [10] using the Tree Matching Pursuit algorithm [11]. Despite the fact that results in terms of SNR are not provided, also in this case there is a good visual match between input and reconstructed signal up to a compression rate equal to $\mathtt{CR} = 4$, while visible reconstruction artifacts are added at a compression rate equal to $\mathtt{CR} = 6$.

7.5 AIC for Intracranial EEG by Shoaran et al., 2014

The circuit considered in this section is an area- and power-efficient approach for compressed recording of cortical signals used in an implantable system (i.e., for intracranial EEG signals), and appeared in the December 2014 issue of the *IEEE Transactions on Biomedical circuits and systems* [25], authors Shoaran et al. from EPFL, Lausanne, Switzerland. The paper is the follow-up of a paper appeared at the *IEEE Biomedical circuits and systems conference* in October, 2014 [24].

The peculiarity of this circuit is to propose a new multichannel Compressed Sensing scheme which exploits time- and the spatial-sparsity of the signals recorded from the electrodes of the sensor array. The circuit has been designed and implemented in a 0.18 μm CMOS technology, and its microphotograph is shown in Fig. 7.16. The main target of the design consists of accommodating a large number of recording

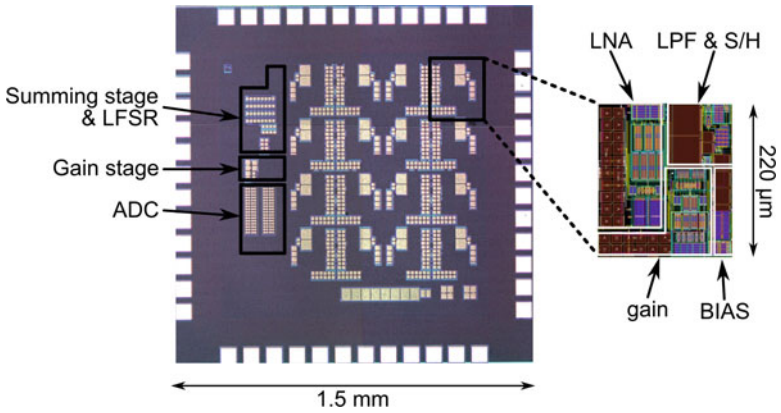


Fig. 7.16 Die microphotograph and single channel layout of the integrated circuit described in Sect. 7.5 (adapted from [24])

units into the available die area, while at the same time preserving sufficiently low-noise and low-power performance, since low power consumption and compact area are crucial aspects of any implantable recording system.

Each integrated circuit includes 16 recording channels consisting of a low-noise amplifier with a band-pass transfer function, an additional low-pass filter to limit the high cut-off frequency, a second gain stage, and a buffered sample-and-hold circuit, all presenting a differential architecture. The outputs of all channels are connected to a single summing and randomly accumulating stage based on a switched-capacitor architecture, performing the Compressed Sensing function. The result is then digitized through a single low-power ADC. The architecture is schematized in Fig. 7.17.

According to simulation and subsequent reconstruction results, the circuit is capable of achieving a fourfold compression on intracranial EEG signals with a signal-to-noise ratio (SNR) as high as 21.8 dB, with an overall power consumption of 10.5 µW within an effective area of 250 ×250 µm per channel.

7.5.1 Hardware Architecture

The hardware architecture of the circuit presented in [25] and schematized in Fig. 7.17 can be divided into a low-power small-size analog front end made of $N = 8$ channels for amplifying, filtering, and sampling the intracranial EEG signals, i.e., *signal conditioning* and the actual analog-to-information converter.

Despite the fact that the design of the signal conditioning block has some noteworthy features to deal with stringent requirements in terms of both area and power, a detailed overview is out of the scope of this chapter, whose aim is to compare different and efficient analog solution for the implementation of CS-based AICs.

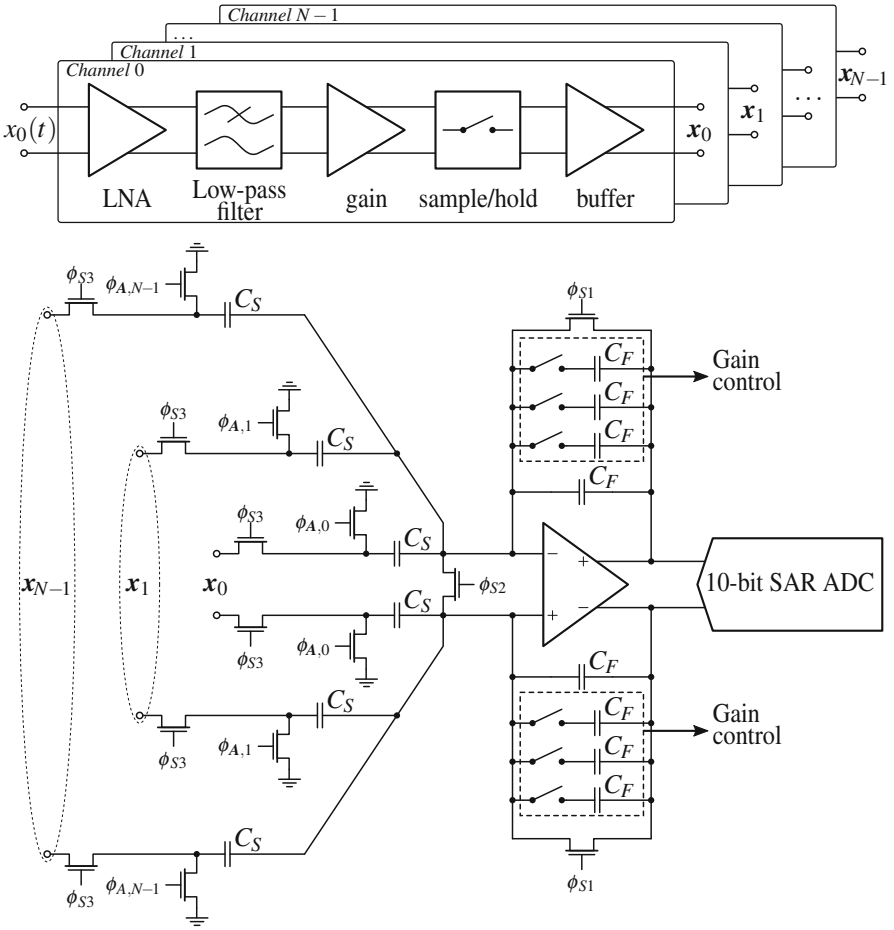


Fig. 7.17 Simplified architecture of the integrated circuit of Fig. 7.16 and proposed in [25], with circuital details of the AIC core

For this reason, we focus on the randomly controlled summing stage implementing an RMPI architecture, which nevertheless exhibits the interesting feature to exploit a mixed spatial- and time-domain signal sparsity. This circuit is based on a switched-capacitor integrator working in two phases (sampling and summing phases). Three different clock signals ϕ_{S1}, ϕ_{S2}, and ϕ_{S3} (along with additional signals $\phi_{\boldsymbol{A},0}$, $\phi_{\boldsymbol{A},1}$, …, $\phi_{\boldsymbol{A},N-1}$ directly controlled by the j-th row $\boldsymbol{A}_{j,\cdot}$ of the sensing matrix) are used to control the behavior providing a proper timing strategy (ϕ_{S1} comes before ϕ_{S2} and ϕ_{S2} before ϕ_{S3}) to reduce the effect of channel charge injection due to the switching activity both on the sampling capacitors C_S and on the feedback capacitors C_F.

In detail, the operation is as follows. Let us indicate with $x_0(t), x_1(t), \ldots, x_{N-1}(t)$ the N differential signals at the analog front-end input. The aim of this block is to amplify, filter, and sample the input signals at the Nyquist rate $f_N = 1/T$. For the sake of simplicity, we can model this circuit as a simple sample/hold, outputting the sampled version $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N-1}$ of the input signals to the AIC every T, or using a more compact notation, outputting the vector $\boldsymbol{x}$ made of the samples of the input signals. This causes the following AIC to belong to the discrete-time analog CS class (*case B* according to Chap. 6).

In sampling mode, ϕ_{S1}, ϕ_{S2}, and ϕ_{S3} are high, allowing the differential voltages across the k-th sampling capacitors C_S couple to be set to $\boldsymbol{x}_k$. At the same time, ϕ_{S1} shorts the feedback capacitors C_F, clearing any previously accumulated charge.

In summation mode (ϕ_{S1}, ϕ_{S2}, and ϕ_{S3} are low), the charge stored on all the capacitors C_S is summed and stored to the C_F only if the control signal $\phi_{\boldsymbol{A},k}$ of the corresponding k-th channel is high, with $k = 0, 1, \ldots, N-1$. By directly controlling the $\phi_{\boldsymbol{A},k}$ signal with the $\boldsymbol{A}_{j,k}$ coefficient, this strategy implements a binary multiplication by $\boldsymbol{A}_{j,k} \in \{0, 1\}$. Mathematically, the differential voltage $\boldsymbol{y}_j$ at the integrator output, at the and of the summation phase, is

$$\boldsymbol{y}_j = -\frac{C_S}{g\, C_F} \sum_{k=0}^{N-1} \boldsymbol{A}_{j,k} \boldsymbol{x}_k = G \boldsymbol{A}_{j,\cdot}\, \boldsymbol{x}$$

where g is an external parameter, with $g \in \{1, 2, 3, 4\}$, capable of controlling the gain of the summing stage by means of switches including additional capacitors in the integrator feedback path, i.e., capable of altering the actual value of C_F, and where the implicitly defined dimensionless coefficient G is the actual gain of the system. The possibility to run-time change the value of G (by means of changing g) is exploited to cope with saturation issues of the summing stage.

The $\boldsymbol{y}_j$ is then converted into a digital word with a Successive Approximation Register (SAR) ADC embedded into the integrated circuit to provide the measurement. A hybrid two-stage class A/AB topology [23] is used as the OTA, providing the desired rail-to-rail output swing. Based on the stringent area and power constraints of the implantable system, a 10-bit ADC capable of a sampling rate up to 20 kS/s is used. A popular SAR architecture, which enables low-power data conversion for medium resolution/speed applications, is used exploiting a binary-weighted capacitive array with attenuation capacitor as embedded DAC.

In a time period T, i.e., in the time period where $\boldsymbol{x}$ is constant, the aforementioned process is repeated M times with different rows $\boldsymbol{A}_{0,\cdot}, \boldsymbol{A}_{1,\cdot}, \ldots, \boldsymbol{A}_{M-1,\cdot}$ of the sensing matrix to get M measurements $\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{M-1}$. Elements $\boldsymbol{A}_{j,k}$ are internally generated with a simple linear feedback shift register due to hardware constraints. The timing diagram regulating this behavior is depicted in Fig. 7.18.

The process is repeated d times using different sample vectors $\boldsymbol{x}$. This allows to gather a total amount of $m = dM$ measurements as a linear combination of the samples of the N channels at different d sampling instants, i.e., an amount of $n =$

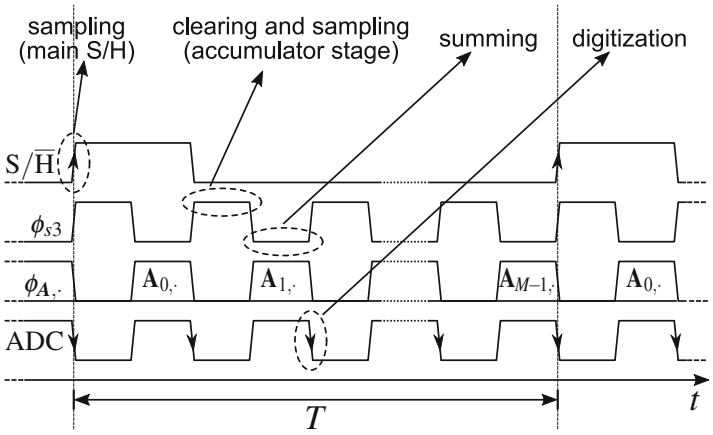


Fig. 7.18 Timing diagram for the signals regulating the behavior of the multichannel integrator of Fig. 7.17

dN sampling points. This mixed spatial–temporal approach allows to exploit the spatial–temporal sparsity properties of this particular signal, as detailed in Sect. 6.5.

The main aspects of the proposed architecture can be summarized as follows:

1. **Time continuity:** Each measurement is spawned over multiple channels in a single time step T, no need for synchronization mechanism.
2. **Resource saving:** Binary mode (no need for analog multiplier).
3. **Saturation:** A variable gain control is implemented in the integrator stage.

7.5.2 Experimental Results

Many measurement results are proposed in [25]. Here, we neglect those related to the signal conditioning block and we propose a summary for those referred to AIC performance. A long segment of multichannel intracranial EEG signal recorded from subdural strip and greed electrodes implanted on the left temporal lobe of a patient with medically refractory epilepsy have been used as input. Signals are recorded during an invasive pre-surgical evaluation phase to pinpoint the areas of the brain involved in seizure generation and to study the feasibility of a resection surgery. Data includes some minutes of pre-ictal, ictal, and post-ictal activities, sampled at 32 kS/s. The intracranial EEG signal has been recorded with a standard medical equipment, and the traces of 16 adjacent electrodes are applied to the proposed CS system as test signal.

The considered sparsity basis is the Gabor one [22] that, along with the wavelet basis, is the most commonly considered when applying CS to biomedical signals [7]. Given the multichannel architecture, neural recovery is performed using the standard reconstruction algorithm adopting the ℓ_1 norm and then adopting the mixed

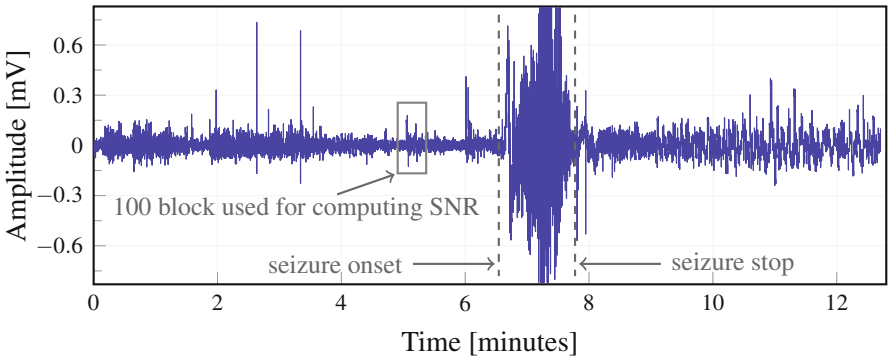


Fig. 7.19 Example (one channel trace only) of human intracranial EEG recording data used for testing the circuit considered in Sect. 7.5. Performance in terms of RSNR is calculated by averaging over the 100 blocks of signal (25.6 s total) in the low-voltage fast activity region indicated in the figure (adapted from [24])

$\ell_{1,2}$ norm. As detailed in Sect. 6.5, when using the ℓ_1 norm, only temporal sparsity is used to recover the input signal, while the mixed $\ell_{1,2}$ norm is capable to exploit the spatial–temporal sparsity properties of the signal. Results are compared in terms of reconstruction RSNR.

The performance of the circuit is validated for low-voltage fast activities which are shown to be associated with seizure onset. In detail, an example of a single channel of the intracranial EEG used in the test is plotted in Fig. 7.19. Performance is computed by averaging reconstruction quality over the 100 blocks of the signal highlighted in the figure, each one with length equal to $n = 1024$ samples, i.e., $T_w = 256$ ms at a $f_N = 4$ kHz sampling frequency, covering an overall observation time of 25.6 s.

A comparison showing the reconstruction of one block in a single channel when using the two reconstruction approaches is depicted in Fig. 7.20. As shown in the figures, applying the recovery considering the adjacent channels as in the $\ell_{1,2}$ case (bottom plot) results in an improved performance compared to the standard sparse recovery as in the ℓ_1 case (top plot).

When considering all the 100 blocks and all the 16 channels ARSNR are 16.6 dB and 21.8 dB for the ℓ_1 reconstruction and the $\ell_{1,2}$ reconstruction, respectively. Based on the statistical analysis reported in [15], a minimum SNR of 10.5 dB (corresponding to a percentage root-mean squared difference of 30%) is acceptable to maintain the diagnostically important data in the recovered signal, e.g., for successful seizure detection.

Results in terms of ARSNR when decreasing the number of measurements (i.e., increasing the compression ratio) are shown in Fig. 7.21. Even if, as expected, performance is decreasing when increasing CR, the target SNR of 10.5 dB indicating a potentially capability to correctly recover the low-voltage intracranial EEG signal over the entire recording period is achieved when using the mixed $\ell_{1,2}$ norm for a compression ratio as high as $\mathrm{CR} = 16$.

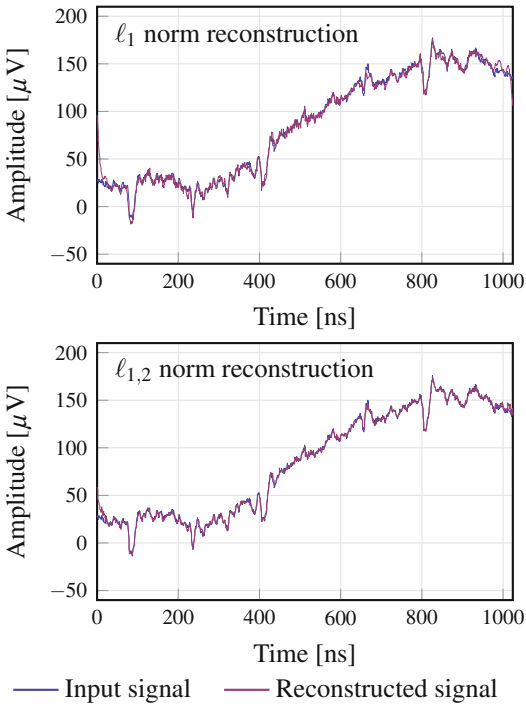


Fig. 7.20 Comparison of recovery performance using different reconstruction methods for a single block with length $n = 1024$ and compression ratio $\texttt{CR} = 4$ for the circuit considered in Sect. 7.5. Top plot: using standard ℓ_1 reconstruction, $\texttt{RSNR} = 21.3$ dB. Bottom plot: using mixed $\ell_{1,2}$ reconstruction, $\texttt{RSNR} = 28.0$ dB (adapted from [24])

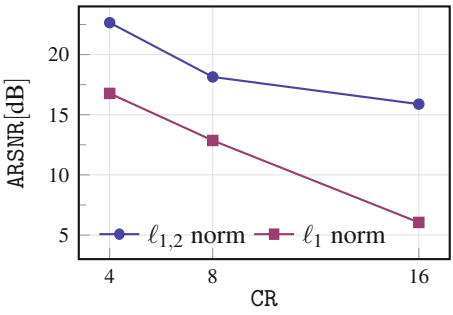


Fig. 7.21 Comparison of performance when using standard ℓ_1 recovery and mixed $\ell_{1,2}$ recovery for different compression ratios when testing the circuit considered in Sect. 7.5. In this case SNRs are averaged over 20 compression blocks

7.6 AIC for Biomedical Signals by Pareschi et al., 2016

In the February, 2016 issue of the *IEEE Transactions on Biomedical circuits and systems*, Pareschi et al. proposed an analog-to-information converter specifically designed for biomedical signals [21]. The circuit has been designed and fabricated in 180 nm 1.8 V CMOS technology, and its microphotograph is depicted in Fig. 7.22.

The circuit size is 2.3×3.7 mm, including 16 switched-capacitor integrators implementing 16 RMPI channels, and a shared 11-bit SAR ADC. The digital control

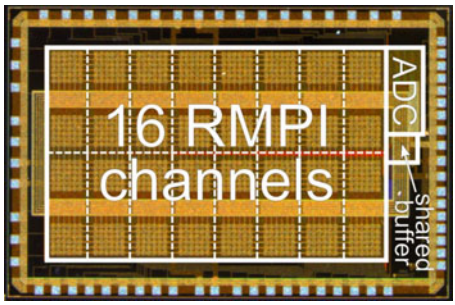


Fig. 7.22 Microphotograph of the integrated circuit considered in Sect. 7.6 (adapted from [21])

logic (excluding that of the SAR) has not been embedded in the circuit, and testing results provided in [21] have been obtained by controlling the designed circuit with an external FPGA.

The core of the circuit is shown in Fig. 7.23. It is a low-power fully differential switched-capacitor integrator capable of implementing an antipodal modulation, where multiplication with the sensing matrix elements $\boldsymbol{A}_{j,k}$ is achieved by means of simple switches that invert the differential input signal pair. The peculiarity of this circuit is the proposed joint hardware-algorithms optimization, that allows to increase performance without any cost in terms of hardware complexity or computational power spent by decoding. The prototype, in fact, is capable to exploit rakeness-based CS capable of increasing performances by exploiting the fact that biosignals are not only sparse, but also localized. Additionally, each channel also includes a smart saturation checking capability [18] by means of the two comparators of Fig. 7.22 that, with a minimal hardware cost, makes it possible to retrieve information from RMPI channels even in presence of saturation.

7.6.1 Hardware Architecture

The architecture of a single RMPI channel of the circuit described in [21] is the standard fully differential switched-capacitor integrator of Fig. 7.23. Due to the switched-capacitor architecture, this circuit has intrinsic sampling capabilities and, even if directly connected to $x(t)$, it actually processes its samples $x_j, x_{j+1}, x_{j+2}, \ldots$. According to the definition of Chap. 6, the AIC belongs the class of discrete-time analog CS (*case B*) systems.

This circuit behavior is regulated by two non-overlapping clock signals ϕ_1 and ϕ_2 of period T. In sampling mode (ϕ_1 high, ϕ_2 low) the differential input signal is connected to the differential pair of sampling capacitors C_S. Two additional switches at the input stage are used to select whether the signal has to be connected directly or by reversing the two differential line, acting as a modulator capable of performing a multiplication with $\boldsymbol{A}_{k,j} \in \{-1, +1\}$. This works as an analog mixer, taking $x(t)$ and $\boldsymbol{A}_{k,j}$ as input signals. At the time instant in which there is a high-to-low transition of ϕ_1 there is the sampling of the $x(t)$ input signal.

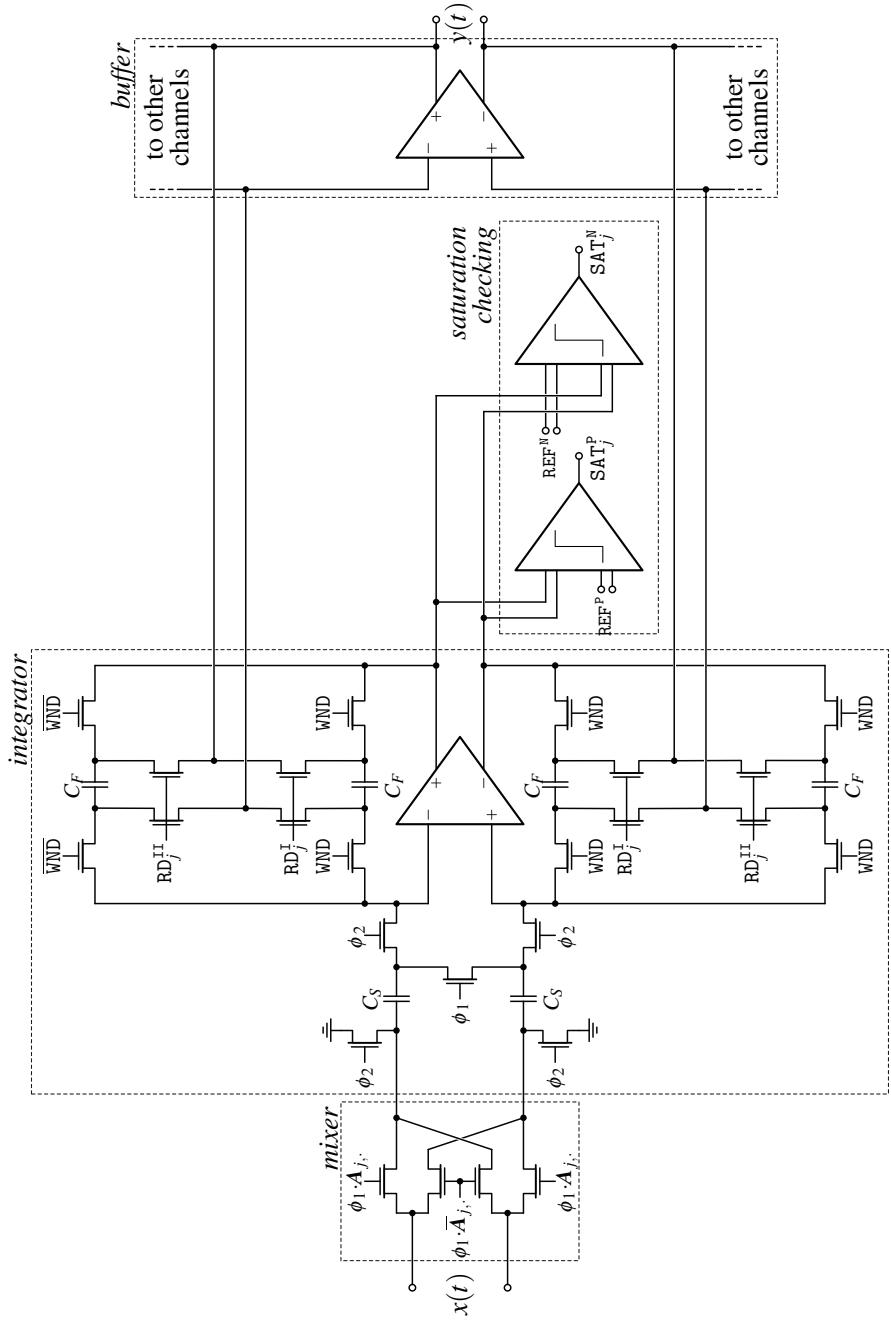


Fig. 7.23 Architecture of the switched-capacitor circuit implementing a single RMPI channel for the integrated circuit of Fig. 7.22 considered in Sect. 7.6

In the following summation mode (ϕ_1 low, ϕ_2 high) the charge stored in the differential couple C_S is removed and transferred either to one of the two differential couples C_F accordingly to the signal WND. This approach is used to solve the problem of ensuring continuity between successive windows of the input signal. When integrating even signal slices (WND is high, so $\overline{\text{WND}}$ is low) one differential couple of feedback capacitors C_F is used for integrating, while the other one is disconnected from the circuit, retaining the charge previously accumulated. During odd signal slices (WND is low and $\overline{\text{WND}}$ is high) the roles of the two couples are reversed, and the one previously used for integrating is disconnected for the circuit, allowing its accumulated charge to be converted into a digital word by a properly designed ADC, and to be cleared (RST signal asserted, not shown in Fig. 7.23 for the sake of clarity) to be able to start again a new integration process.

Mathematically, assuming $T_w = nT$, and indicating with $\boldsymbol{x}_k$ the differential voltage samples of the input signal available at the modulator differential input at the sampling instant of the k-th time step, the integrator voltage output after n time steps is given by

$$\boldsymbol{y}_j = -\frac{C_S}{C_F}\sum_{k=0}^{n-1}\boldsymbol{A}_{j,k}\,\boldsymbol{x}_x = G\boldsymbol{A}_{j,\cdot}\boldsymbol{x} \tag{7.7}$$

where the dimensionless constant G represents, as in previous cases, the gain of the integrator stage.

After T_w, the differential C_F couples retaining the measurements of all the 16 embedded RMPI channels are connected, one at a time, to a shared output buffer and to the SAR ADC.

The timing diagram regulating this behavior is depicted in Fig. 7.24.

The values of the sampling capacitor C_S and the feedback capacitors C_F values have been selected accordingly to leakage constraints. Aiming at an integration

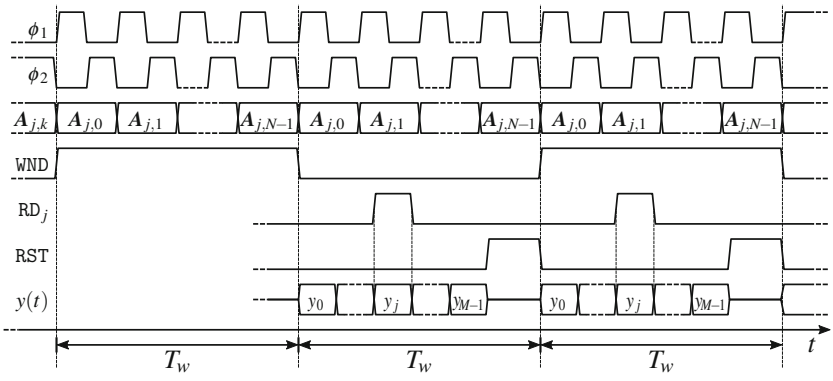


Fig. 7.24 Timing diagram for the signals regulating the behavior of the switched-capacitor circuit depicted in Fig. 7.23

time of $T_w = nT \approx 1\,\text{s}$ (that is enough for all biomedical signals of interest), and carefully designing the op-amp and the switches, the voltage dropout at the integrator output is comparable to the ADC LSB for a $T_w = 1\,\text{s}$ when $C_F \approx 40\,\text{pF}$ without applying any digital compensation technique [19]. With this value the gain of the integrator is set to $G = -1/8$ by designing $C_S = 5\,\text{pF}$, thus limiting saturation effects on the integrator.

Furthermore, two additional features are considered in the design of the AIC. The first one is the possibility to adopt rakeness-based sensing matrices as detailed in Chap. 3.

As a second, additional, feature, two comparators for each RMPI channel (clearly visible in Fig. 7.23) have been embedded. This allows to check if either the final or the intermediate integration voltage goes above or below the two threshold levels, enabling the smart saturation checking capability suggested in Chap. 6 [18]. In more detail, the Saturated Projection Windowing (SPW) algorithm is implemented, and whenever a dynamic saturation is detected, a flag signal (either indicating a positive or a negative saturation event) is generated. This information is useful for the reconstruction algorithm to recover information. Details can be found in Sect. 6.4

In conclusion, the main aspects of the proposed architecture can be summarized as follows:

1. **Time continuity:** Additional couple of feedback capacitors to allow time continuity without the need for any additional active circuit.
2. **Resource saving:** Antipodal mode (no need for analog multiplier). Rakeness-based sensing matrix to reduce the number of measurements.
3. **Saturation:** Reduced integration gain, additional comparators for smart saturation checking.

7.6.2 Experimental Results

Intensive experimental measurements are provided in [21]. Tests are divided into two parts. The first one is dedicated to measuring the circuit performance with some suitable artificial test signals. Then, results are provided for the behavior of the AIC by using real ECGs and EMGs taken from the PhysioNet database [13]. In all tests, signals have been generated with an external DAC driven by the same FPGA controlling the designed circuit, and reconstructed by using the iterative convex solvers SPGL-1 [5].

The performance of the integrated ADC is summarized as follows. The integral non-linearity (INL) is within 3.4 LSB at 11 bit resolution, with a spurious free dynamic range is measured in 64.2 dB with ENOB evaluated in approximately 9 bits. The power consumption is measured in $10\,\mu\text{W}$, yielding a figure of merit (defined as energy required per conversion per effective number of levels) of $198\,\text{fJ/conversion-level}$ [20].

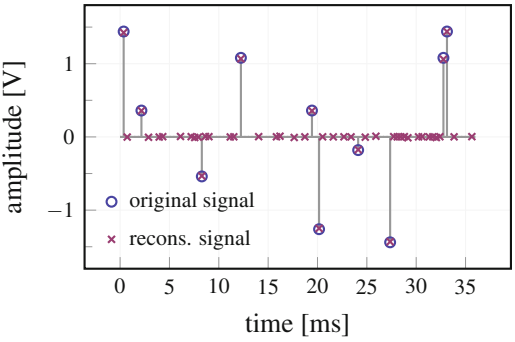


Fig. 7.25 Example of reconstruction of a signal sparse on the canonical basis with $\kappa = 2$, $n = 20$, and $m = 8$ for the circuit considered in Sect. 7.6. 5 signal slices are plotted, each with $T_w = 7.2\,\text{ms}$. All steps with non-vanishing amplitude are highlighted with a marker (adapted from [21])

The first measurement provided involves an input signal whose sparsity basis $\boldsymbol{D}$ is made with normalized unit pulses, i.e., accordingly to the notation of Chap. 1:

$$x(t) = \sum_{k=0}^{n-1} \xi_k \, \mathrm{u}\left(\frac{t}{T} - k\right) \tag{7.8}$$

where $\mathrm{u}(\tau) = 1,\ 0 < \tau < 1$ and 0 elsewhere. In this test, $n = 20$ and the sparsity level is set to $\kappa = 2$, i.e., only 10% of the $\boldsymbol{\xi}_j$ is nonzero. By manipulating (7.4), the authors of [21] were able to assert that the minimum number of measurements required for an accurate signal reconstruction in this setting is $m \geq 8$. Accordingly, only 8 RMPI channels of the designed circuit are used. In this test, the designed prototype with $T = 360\,\mu\text{s}$, i.e., with a switched-capacitor frequency equal to $f_N = 1/T = 2.78\,\text{kHz}$, was capable to achieve an ARSNR equal to 37.7 dB. An example showing both the input signal and the reconstructed signal over 5 consecutive time windows is depicted in Fig. 7.25.

Next, a more complex situation dealing with a synthetic signal where $\boldsymbol{D}$ is the Fourier basis is presented, i.e.,

$$x(t) = \sum_{k=0}^{n/2} \xi_j \cos(k\,t) + \sum_{k=n/2+1}^{n-1} \xi_j \sin((n-k)t) \tag{7.9}$$

In this setup, authors set $n = 64$ and $\kappa = 3$; to ensure accurate reconstruction according to (7.4), $m \geq 16$; as such all channels of the RMPI prototype are used. By setting $T = 360\,\mu\text{s}$ (i.e., $f_N = 2.78\,\text{kHz}$) measurements indicate an RSNR of 30.0 dB. The input and the reconstructed signal for this example in a single time window are been depicted in Fig. 7.26.

The Fourier-based setting has also been used for two additional and extremely interesting tests.

1. The first one regards the behavior of the circuit at different clock speed. The upper limit is given by the op-amp used in the integrator circuit, and has been

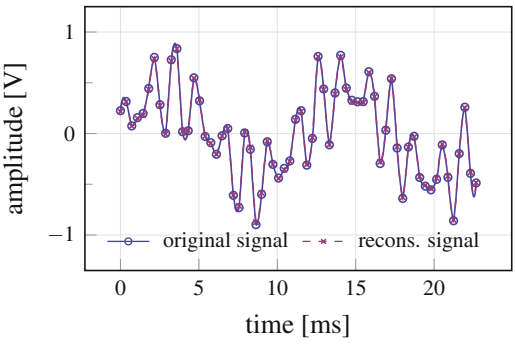


Fig. 7.26 Example of reconstruction of a signal sparse on the Fourier basis with $\kappa = 3$, $n = 64$, and $m = 16$ for the circuit considered in Sect. 7.6. A single signal slice with $T_w = 23$ ms is plotted. Actual sampling points are highlighted with a marker (adapted from [21])

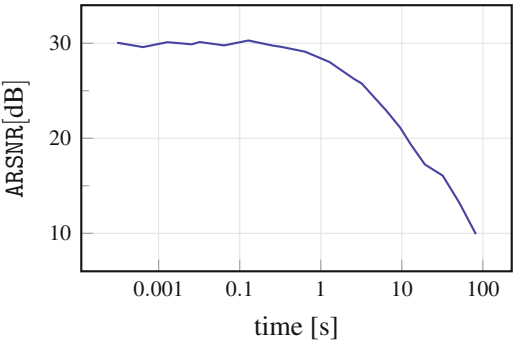


Fig. 7.27 RSNR for the Fourier sparse signal with $\kappa = 3$, $n = 64$, and $m = 16$, for different time window lengths T_w for the circuit considered in Sect. 7.6 (adapted from [21])

evaluated in approximately $f_N = 125$ kHz. For any clock speed below this limit, the reconstruction SNR is approximately constant around 30 dB. The lower bound is set by constraints imposed by the leakage currents.

Performance in terms of RSNR for different f_N as functions of the integration window length $T_w = n/f_N$ (that is the actual parameter determining the voltage drop due to leakage) is shown in Fig. 7.27. Performance is constant up to a value of T_w in the order of magnitude of the second. By defining the maximum integration time the one causing a 3 dB reconstruction SNR loss, this limit is $T_w < 1.6$ s. This ensures that the designed circuit can be correctly employed for the acquisition of low-bandwidth biomedical signals.

2. The second experiment is used to test the behavior of the AIC in presence of an input signal whose amplitude is large enough to cause a saturation into the system. The same input signal, sparse in the Fourier domain, has been scaled by a factor $0 < s \leq 2$. For $s \leq 1$ the voltage level associated with the measurements $\boldsymbol{y}$ is such that none of the 16 used RMPI channels reaches final or intermediate saturation, while some saturation is observed for $s > 1$. Results are shown in Fig. 7.28.

   For very low values of s low performance is observed, mainly due to the low energy of the measurements, that scales with s. As s increases, the reconstruction SNR increases. Intriguingly, when $s > 1$ a few saturation events occur. In this

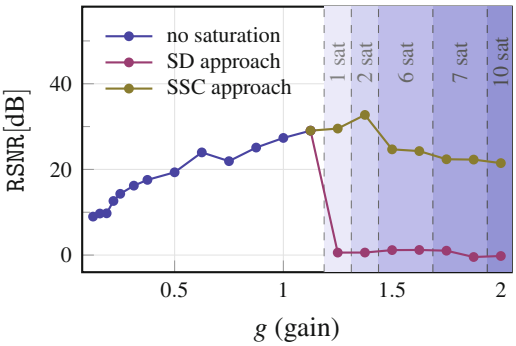


Fig. 7.28 Performance of the Fourier basis example at different signal gain s for the circuit considered in Sect. 7.6. The plot shows reconstruction SNR when adopting the proposed smart saturation check (SSC) and when saturated measurements are simply dropped (SD). The number of saturated measurements per time window is also indicated (adapted from [21])

cases, two reconstruction approaches are proposed by authors. (*i*) With *saturation drop* (SD in the plot) measurements where a saturation event is detected are discarded, and reconstruction is performed using only "good measurements." (*ii*) When *smart saturation checking* (SSC in the plot) is enabled. Interestingly, since mM is taken around its minimum value, dropping saturated measurements reduces the data available to the CS decoder to an insufficient level, and, as expected, one is not able to correctly reconstruct the input signal anymore. As in the figure, the reconstruction SNR has an abrupt fall at $s \approx 1$. Conversely, when SSC is employed, some amount of information is still recovered even from saturated measurements, and performance is still increasing with s when only a limited number of saturation events are detected. This can be intuitively explained as due to two effects. First, only a few of the 16 RMPI channels saturate, and most probably the saturation events are observed at the end of the integration windows, and this is assumed to bring a large quantity of information since the signal has been observed for a long time. Second, as s increases, also the power of the non-saturated measurements increases, with a better conversion accuracy. However, when s (and so the number of saturated measurements) further increases, performance drops even if it is still possible to reconstruct the input signal with an acceptable SNR. Note that for $s = 2$ reconstruction is still possible even if most of the measurements (i.e., 10 out of 16) reach saturation.

The prototype is then tested with real biomedical signals, more precisely, by using ECG and EMG input signals (including regular, irregular, and pathological ones) recorded from undisclosed healthy/unhealthy patients and made publicly available by the PhysioNet database [13]. In this test, both rakeness (using only two rakeness-based sensing matrices, one for the ECG, one for the EMG, estimated by using a training set not including the considered input signal to avoid biasing) and the SSC approaches have been used. In the first example an ECG signal with a

heartbeat of approximately 60 bpm from a healthy patient is considered. The signal is sampled at $f_N = 256\,\text{Hz}$, with $T_w = n/f_N = 0.5\,\text{s}$, so $n = 128$. The number of measurement for each time windows is $m = 16$. The sparsity basis used for signal reconstruction is the Symmlet-6 family of the orthogonal Wavelet functions [16]. Results are shown in Fig. 7.29 with three different scale factors, and compared with results obtained when using the standard (i.e., not rakeness-based, binaries antipodal random $\boldsymbol{A}$) approach.

By using a standard CS approach the signal reconstruction quality is visually very poor. Instead, when exploiting the rakeness approach the performance is visibly much higher. Furthermore, also this (realistic) system is capable to tolerate a limited amount of saturation events. For all considered scaling factors the signal has been reconstructed without any noticeable performance loss, considering that with $s = 1$ no saturation events are observed, with $s = 1.5$ an average number of 0.4 saturation events per time window is observed, while when $s = 2$ an amount of 1.5 saturation events are detected per time windows. For all cases, the obtained compression factor is equal to $\texttt{CR} = 8$.

The second biomedical signal example is similar to the first one, but an EMG signal of an healthy patient is considered. In this setting $f_N = 20\,\text{kHz}$ (the EMG signal is usually sampled at a higher frequency with respect to the ECG [6, 7]), $n = 256$, and $T_w = 12.8\,\text{ms}$. The considered sparsity basis is, as in the previous example, the Symmlet-6 Wavelet function family, and $m = 24$ is obtained by simultaneously using two prototypes. Results are shown in Fig. 7.30.

Again, there is a clear failure in the reconstruction effort in the standard CS approach, while in the rakeness approach reconstruct of the input signal is successfully achieved for $s = 1$ (no saturation events detected), for $s = 1.5$ (an average of 1.2 saturation events per time window is present), and also for $s = 2$ (corresponding to 2.5 saturation events per time windows). The compression factor in this example is equal to $\texttt{CR} \approx 10$.

Finally, a few tests on both irregular and pathological ECG and EMG signals are presented and shown in Fig. 7.31. In the figure small chunks of these uncommon signal instances taken from the PhysioNet database superimposed with the corresponding reconstructed signals are considered. The system setting is the same as in the healthy cases considered above (i.e., $m = 16$ for ECG signals, and $m = 24$ for EEG signals). The input signal is always correctly reconstructed, even if a few saturation events are registered in all cases.

7.7 Prototype Comparison

Table 7.2 presents a comparison between the most important features of the AICs presented in this chapter.

Note that the aim of the table is not to compare performance of the different solutions that, in our opinion, would not be truly meaningful. All considered integrated circuits share the same RMPI architecture, but with many differences.

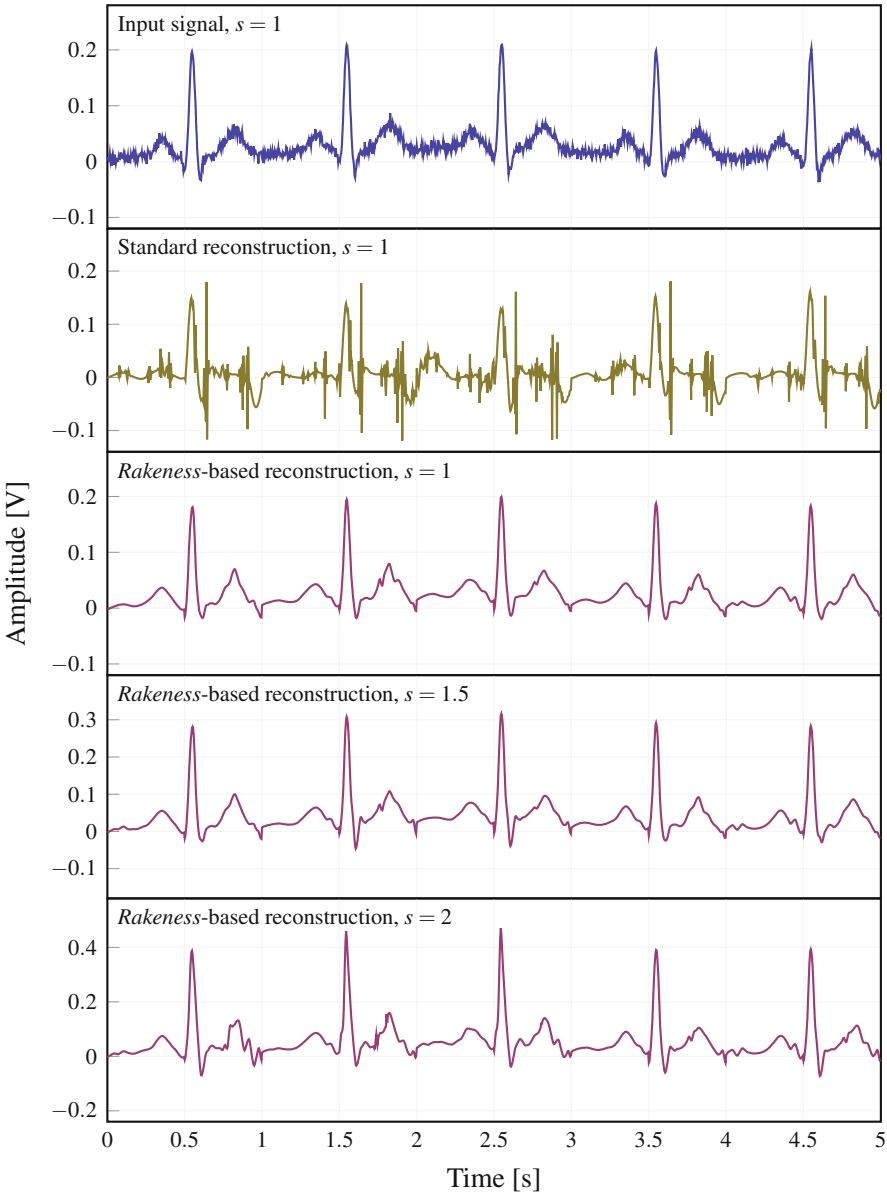


Fig. 7.29 Example of reconstruction of a real ECG signal with $f_N = 256$ Hz, $n = 128$, $m = 16$ for the circuit considered in Sect. 7.6 (10 consecutive time windows are plotted). From top to bottom: input signal, reconstructed signal with the standard CS approach, i.e., by using independent $a_{k,j}$ symbols (no rakeness), and signal reconstructed using the rakeness CS with the three scaling factor $s = 1$, $s = 1.5$, and $s = 2$ (adapted from [21])

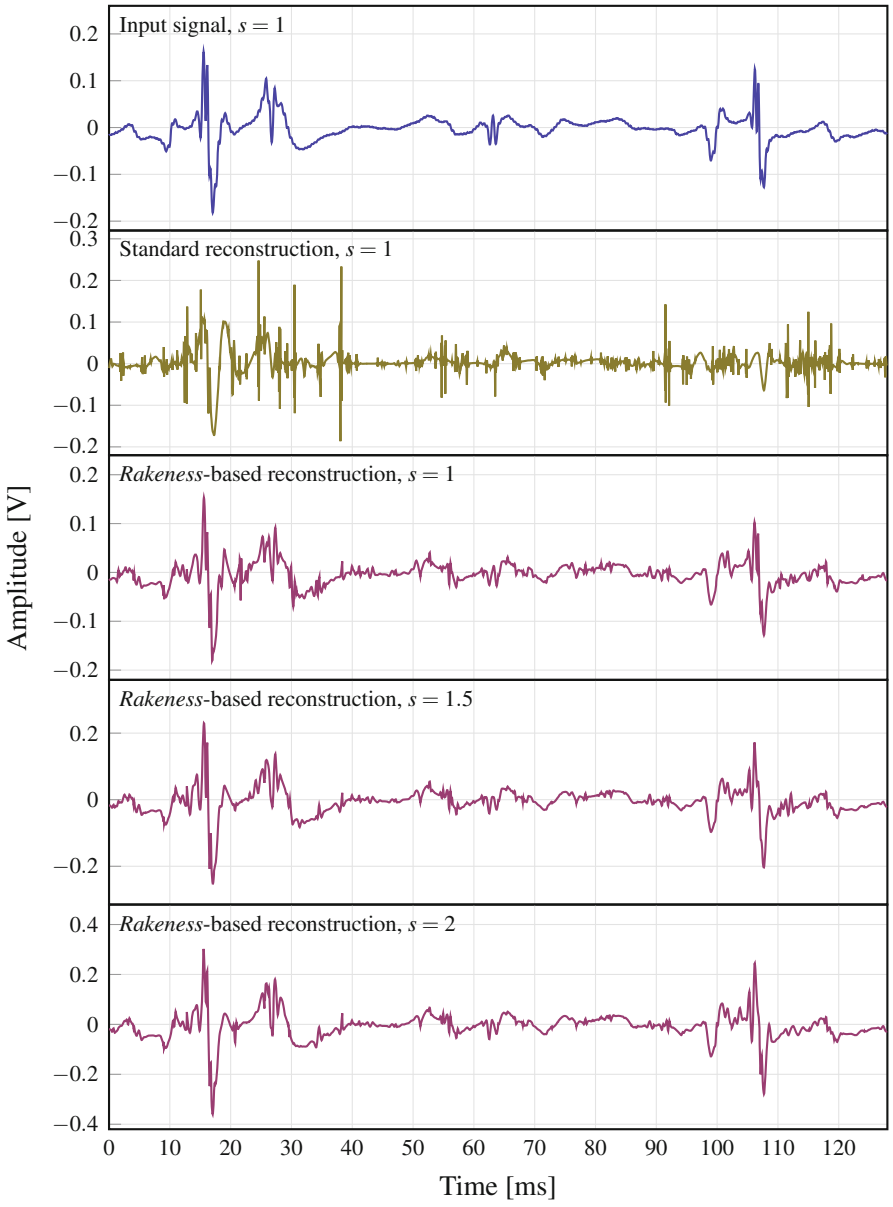


Fig. 7.30 Example of reconstruction of a real EMG signal with $f_N = 20\,\text{kHz}$, $n = 256$, $m = 24$ for the circuit considered in Sect. 7.6 (10 consecutive time windows are plotted). From top to bottom: input signal, reconstructed signal with the standard CS approach, i.e., by using independent $a_{k,j}$ symbols (no rakeness), and signal reconstructed using the rakeness CS with the three scaling factor $s = 1$, $s = 1.5$, and $s = 2$ (adapted from [21])

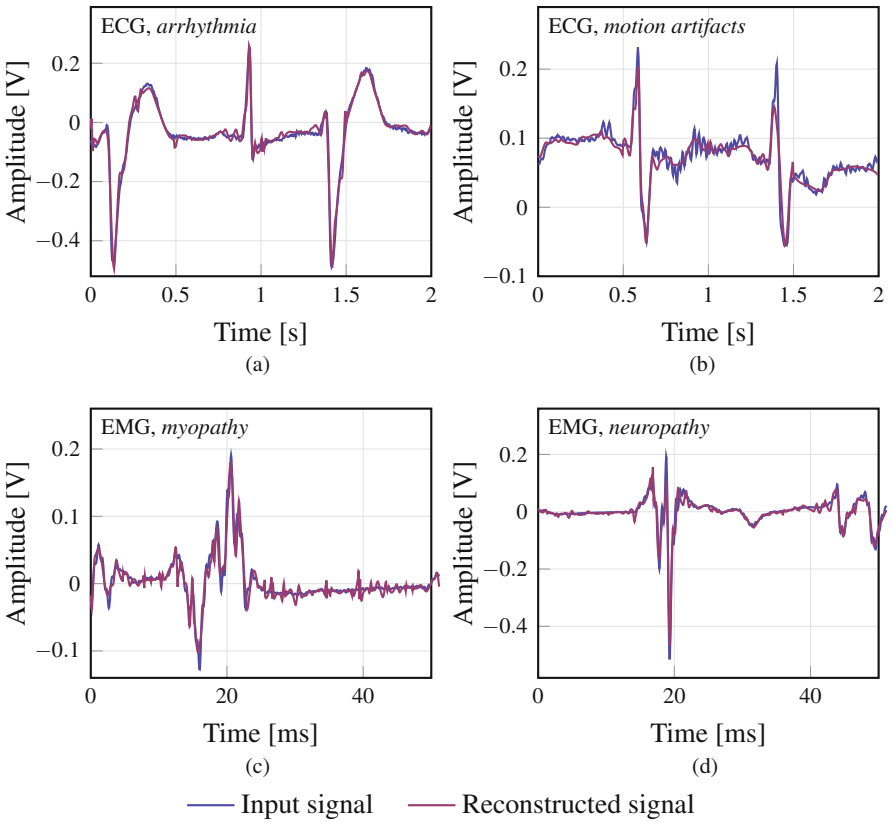


Fig. 7.31 Short chunks of real pathological/irregular ECGs and EMGs (4 consecutive time windows are shown for each signal) compared with the corresponding reconstructed signals for the circuit considered in Sect. 7.6 adopting the settings used in previous examples. (**a**) ECG signal track from a patient with arrhythmia, 2.5 saturation events per time windows on average. (**b**) ECG signal corrupted by motion artifacts, 1.75 saturation events per time windows. (**c**) EMG signal from a patient with myopathy, 1.5 saturation events occur per time windows. (**d**) EMG signal from a patient with chronic low back pain and neuropathy, 2 saturation events per time window (adapted from [21])

Some of them (see Shoaran et al. [25], Gangopadhyay et al. [12], and Pareschi et al. [21]) are AIC specifically designed for biomedical signals, with a switched-capacitor implementation that leads to a *discrete-time* approach. Furthermore, some solutions propose a very specific design, that allow them to work only with a peculiar class of signal [12, 25] while others are more general-purpose [21].

Other integrated circuits (see Yoo et al. [27] and Chen et al. [8]) are high frequency *continuous-time* AICs, tested only with sinusoidal signals and that do not embed the final ADC. Due to this, a comparison in what is typically the most interesting feature of AICs, i.e., the power consumption, is not possible even considering any normalization factor.

Table 7.2 Collection of data from recent CS-based AIC relying on RMPI architecture

	RMPI			ADC			Design				Testing		
	# ch.	$A_{j,k}$	A gen.	f_s [KS/s]	ENOB	Bit	Band [KHz]	Tech [nm]	V_{dd} [V]	Power [μW]	Signals	# sig.	Features
Yoo et al. [27]	8	$\{\pm 1\}$	Off-chip	External ADC			$2 \cdot 10^6$	90	1.5	$51 \cdot 10^4$	Radar (Sin)	1	RF, no embedded ADC
Chen et al. [8]	1	$\{\pm 1\}$	On-chip	External ADC			$5 \cdot 10^5$	90	–	$55 \cdot 10^3$	BPSK (Sin)	1	RF, no embedded ADC
Gangopadhyay et al. [12]	64	Uniform, 6 bit/1 bit	On-chip	0.2	6.5	10	1	130	1.2	1.8	ECG Sin	1	Low power Specifically designed for ECGs
Shoaran et al. [25]	16	$\{0, 1\}$	On-chip	20	9.2	10	1.9	180	1.2	168	iEEG	16	AFE design included; multi-lead signals needed
Pareschi et al. [21]	16	$\{\pm 1\}$	Off-chip	200	9	11	65	180	1.8	495	ECG EMG Sin Pulses	1	Smart saturation checking Works with many biosignals Rakeness approach

Yet, we think that a comparison between the different features, different solution adopted, and different application scenarios could be interesting to the reader. More interestingly, the peculiarity of Table 7.2 is to highlight the versatility of CS-based solution for the design of AIC. Even if still in embryonic state, CS technology has successfully proven to be effective in building AICs that can be used in an extremely wide range of applications.

References

1. E.G. Allstot et al., Compressive sampling of ECG bio-signals: Quantization noise and sparsity considerations, in *2010 Biomedical Circuits and Systems Conference (BioCAS)*, Nov. 2010, pp. 41–44
2. E. Alpman et al., A 1.1V 50mW 2.5GS/s 7b time-interleaved C-2C SAR ADC in 45nm LP digital CMOS, in *2009 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*, Feb. 2009, pp. 76–77,77a
3. R. Bagheri et al., An 800MHz to 5GHz software-defined radio receiver in 90nm CMOS, in *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*, Feb. 2006, pp. 1932–1941
4. S. Becker, Practical Compressed Sensing: modern data acquisition and signal processing. PhD thesis. California Institute of Technology, 2011
5. E. van den Berg, M.P. Friedlander, Sparse optimization with least-squares constraints. SIAM J. Optim. **21**(4), 1201–1229 (2011)
6. F. Chen, A.P. Chandrakasan, V. Stojanović, A signal-agnostic compressed sensing acquisition system for wireless and implantable sensors, in *Custom Integrated Circuits Conference (CICC), 2010 IEEE*, Sept. 2010, pp. 1–4
7. F. Chen, A.P. Chandrakasan, V.M. Stojanović, Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors. IEEE J. Solid State Circuits **47**(3), 744–756 (2012)
8. X. Chen et al., A sub-Nyquist rate compressive sensing data acquisition front-end. IEEE J. Emerging Sel. Top. Circuits Syst. **2**(3), 542–551 (2012)
9. X. Chen et al., A sub-Nyquist rate sampling receiver exploiting compressive sensing. IEEE Trans. Circuits Syst. I Regul. Pap. **58**(3), 507–520 (2011)
10. I. Daubechies, *Ten Lectures on Wavelets* (SIAM, Philadelphia, 1992)
11. M.F. Duarte, M.B. Wakin, R.G. Baraniuk, Fast re-construction of piecewise smooth signals from random projections, in *Online Proceedings of the Workshop on Signal Processing with Adaptative Sparse Structured Representations (SPARS)*, Rennes, France, Nov. 2005
12. D. Gangopadhyay et al., Compressed sensing analog front-end for bio-sensor applications. IEEE J. Solid State Circuits **49**(2), 426–438 (2014)
13. A.L. Goldberger et al., Physiobank, Physiotoolkit, and Physionet: components of a new research resource for complex physiologic signals. Circulation **101**(23), 215–220 (2000)
14. M. Grant, S. Boyd, *CVX: Matlab Software for Disciplined Convex Programming, version 2.1*. http://cvxr.com/cvx, Mar 2015
15. G. Higgins et al., EEG compression using JPEG2000: how much loss is too much? in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, Aug. 2010, pp. 614–617
16. S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*. Access Online via Elsevier, 2008
17. M. Mangia, R. Rovatti, G. Setti, Rakeness in the design of analog-to-information conversion of sparse and localized signals. IEEE Trans. Circuits Syst. I Regul. Pap. **59**(5), 1001–1014 (2012)

18. M. Mangia et al., Coping with saturating projection stages in RMPI-based Compressive Sensing, in *2012 IEEE International Symposium on Circuits and Systems*, May 2012, pp. 2805–2808
19. M. Mangia et al., Leakage compensation in analog random modulation pre-integration architectures for biosignal acquisition, in *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, IEEE, Oct. 2014
20. B. Murmann, A/D converter trends: power dissipation, scaling and digitally assisted architectures, in *2008 IEEE Custom Integrated Circuits Conference*, Sept. 2008, pp. 105–112
21. F. Pareschi et al., Hardware-algorithms co-design and implementation of an analog-to-information converter for biosignals based on Compressed Sensing. IEEE Trans. Biomed. Circuits Syst. **10**(1), 149–162 (2016)
22. S. Qian, D. Chen, Discrete Gabor transform. IEEE Trans. Signal Process. **41**(7), 2429–2438 (1993)
23. S. Rabii, B.A. Wooley, A 1.8-V digital-audio sigma-delta modulator in 0.8-μm CMOS. IEEE J. Solid State Circuits **32**(6), 783–796 (1997)
24. M. Shoaran, H. Afshari, A. Schmid, A novel compressive sensing architecture for high-density biological signal recording, in *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*, Oct. 2014, pp. 13–16
25. M. Shoaran et al., Compact low-power cortical recording architecture for compressive multichannel data acquisition. IEEE Trans. Biomed. Circuits Syst. **8**(6), 857–870 (2014)
26. M. Wakin et al., A nonuniform sampler for wideband spectrally-sparse environments. IEEE J. Emerging Sel. Top. Circuits Syst. **2**(3), 516–529 (2012)
27. J. Yoo et al., A 100MHz-2GHz 12.5x sub-Nyquist rate receiver in 90nm CMOS, in *2012 IEEE Radio Frequency Integrated Circuits Symposium*, June 2012, pp. 31–34
28. J. Yoo et al., Design and implementation of a fully integrated compressed-sensing signal acquisition system, in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2012, pp. 5325–5328
29. Z. Yu, S. Hoyos, B.M. Sadler, Mixed-signal parallel compressed sensing and reception for cognitive radio, in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2008, pp. 3861–3864

Chapter 8
Low-Complexity Biosignal Compression Using Compressed Sensing

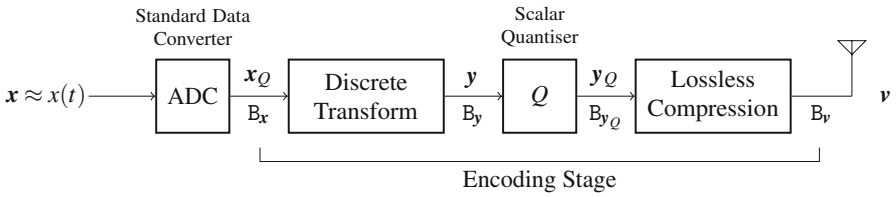
In this chapter we discuss the use of Compressed Sensing (CS) as a means to provide lossy digital signal compression with minimal hardware requirements, focusing our analysis on the specific case of biosignal compression for different types of time-series data. The driving intuition is that in future sensor network scenarios the nodes devoted to acquiring such signals will be heavily and increasingly resource-constrained, the most limiting factor therefore being power consumption spent to acquire, encode, and transmit the sensed data.

Thus, after discussing in the previous chapters how beneficial CS is to reducing the resources spent in signal acquisition, we now "zoom in" on the encoding stage, and present digital-to-digital embodiments of CS as a building block for digital signal compression. Indeed, as hinted in the previous chapters, CS with Random Antipodal Ensemble (RAE) sensing matrices (either i.i.d. or synthesized using the rakeness-based design flow introduced in Chaps. 3 and 4) only requires signed sums of samples; hence, it can be implemented with fixed-point digital hardware architectures that use multiplierless schemes and can be swiftly coded, e.g., on a field programmable gate array. With this common concept of CS being a low-complexity building block to provide lossy compression, we expand these considerations in the following sections.

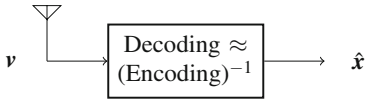
8.1 Low-Complexity Biosignal Encoding by CS

To understand which resources are taken by the encoding stage in the budget of a typical sensor node, we recall the scheme of Fig. 2.13 and put it in the context of digital signal compression as in Fig. 8.1. In a few words, a sensor digitizes an analog signal by means of, e.g., Nyquist-rate analog-to-digital converters (ADC) or similar means of signal acquisition. The acquired samples are then compressed on-board by a lossy or lossless encoder that typically operates by (*i*) applying a suitable

M. Mangia et al., *Adapted Compressed Sensing for Effective Hardware Implementations*, DOI 10.1007/978-3-319-61373-4_8



(a) Sensor node; the bitstream lengths are denoted by B.



(b) Processing node; the decoding stage contains the inverse operations of the encoding stage (or approximation thereof, if not fully invertible).

Fig. 8.1 A standard sensor node–processing node pair, highlighting the role of digital signal compression prior to transmission. Channel coding is regarded as part of the transmitter/receiver

transform (e.g., a discrete cosine transform (DCT) or wavelet transform (DWT)), (*ii*) quantizing the transformed coefficients, and (*iii*) a further lossless (or lossy) encoding, such as an entropy coding stage [39, Chapter 4], is applied to eliminate residual redundancy in the encoded bitstream and yield a compressed bitstream ready for channel coding, i.e., $\boldsymbol{v}$. This result can then be transmitted to a remote location or stored on a suitable local memory. The received (or stored) data will then be processed by an off-board decoding unit that retrieves the information content up to the data fidelity allowed by noise (both due to quantization and to intrinsic sources in the measurement process) and lossy encoding, by inverting exactly or approximately the operations performed by the encoding stage.

Since resource consumption in a sensor node is typically dominated by the power cost of data transmission, minimizing its rate by suitable lossy or lossless encoding stages is critical in reducing the nodes' resources. Hence, even if an encoder had non-negligible computational complexity, to minimize the amount of bits transmitted on a channel the system-level designer would tend to spend more resources on the encoding stage. However, its complexity also consumes resources by a non-negligible amount; let us then assume that signal acquisition and data compression are performed by low-power and low-complexity sensor nodes, and that these nodes connect to one or more processing nodes providing much larger computational power. In the event of such an asymmetry of available resources, we must reconsider the use of compression schemes designed for more common multimedia access, i.e., on the *opposite* assumption that the encoding is performed only once, and is therefore as computationally demanding as required, while decoding is performed only as multiple users access the information content, and must be as lightweight as possible to meet the resources of, e.g., mobile access

terminals. Hence, compression schemes that exploit only a few and very elementary computations are appealing to cope with such resource partitioning that is evidently unbalanced on the decoder side.

In this view, CS could act as a lossy compression stage in an encoder, which operates by projecting the signal onto a RAE sensing matrix (as given in Definition 1.7). Thus, the expense of computational or digital hardware complexity is expected to be minimal. On the other hand, the decoding stage will require the computational effort of solving a sparse (or structured) signal recovery problem, with the aid of efficient algorithms as they were recalled in Sect. 2.2.

Existing information-theoretic investigations that analyze CS as a digital-to-digital lossy compression, such as [17], did show that its rate-distortion performances [11] are asymptotically suboptimal w.r.t. simple transform-coding techniques that implement the scheme of Fig. 8.1 without resorting to CS. Although formally correct, these works do not account for a down-to-bits analysis of the digital hardware requirements of such transform-coding schemes, that often require floating-point multiplications to be accurately implemented and yield the desired performances. On the other hand, CS with RAE sensing matrices is extremely lightweight and multiplierless w.r.t. standard compression schemes. To improve its rate, we will also pair CS with a further entropy coding stage (i.e., Huffman coding [20]) to attain an even more compressed output bitstream $\boldsymbol{v}$.

Thus, we will illustrate how the task of encoding a signal by CS is well-suited to the tight resource requirements of sensor nodes, whereas signal recovery is more appropriately targeted to a central node that receives all the encoded streams. As a practical case for this application, we compare the performances of CS with some reference compression schemes for single-lead electrocardiographic (ECG) signal compression. In addition, we show that a direct application of the adaptation principles developed in Chap. 3 and the related techniques in Chaps. 4 and 5 allows for a further, significant code rate reduction for the proposed compression scheme w.r.t. non-adapted CS.

8.1.1 Lossy Compression Schemes for Biosignals

We here consider the specific case of ECG signals as a relevant example for the development of wireless health monitoring sensors; the appeal of such signals is due to the fact that they exhibit a quasi-stationary behavior over time, as they convey information on an essentially periodic phenomenon. Thus, n-samples windows $\boldsymbol{x}$ (i.e., when they are considered as random vectors) of this signal class are not only typically compressible w.r.t. a suitable DWT (i.e., they are accurately represented by only $\kappa \ll n$ nonzero wavelet coefficients), but are also endowed with additional structure given by the higher-order moments (i.e., the correlation properties) of this signal ensemble that can be leveraged by a suitable adaptation.

The standard approach to acquiring such signals is depicted in Fig. 8.1: the analog ECG is first acquired by ADC that discretizes it into n Nyquist-rate samples

collected in $\boldsymbol{x}$. Moreover, the ADC intrinsically requires quantization of the signal range to yield standard, pulse-code modulated (PCM) samples $\boldsymbol{x}_Q = Q_{\mathtt{b}_x}(\boldsymbol{x})$, with $Q_{\mathtt{b}_x}$ denoting uniform[1] scalar quantization at $\mathtt{b}_x$ bits per sample (hereafter bps) whose bins cover the full analog input range. The task of encoding $\boldsymbol{x}_Q$ prior to transmission can be divided into two stages (the bitstream lengths are denoted by B.):

1. a lossy encoding stage that allows for a reduced-size bitstream $\boldsymbol{y}_Q$ by accepting some information loss w.r.t. $\boldsymbol{x}_Q$. This is divided into a discrete transform that maps $\boldsymbol{x}_Q$ in a domain where a compressible behavior is observed followed by an additional quantization step, where information loss is allowed with the purpose of reducing the code rate;
2. a lossless encoding stage that eliminates the remaining redundancy in $\boldsymbol{y}_Q$ by operating on its symbols, returning a compressed binary string (i.e., a bitstream) $\boldsymbol{v}$ at the output. Typical examples of such a stage are entropy coding schemes as described in [39, Chapter 4].

The two stages achieve for an n-samples window a code rate of $r = \frac{\mathtt{B}_v}{n}$ bps with a total of $\mathtt{B}_v$ bits in the encoded bitstream. In particular, we here evaluate the possibility of using CS as digital signal compression scheme which applies linear dimensionality reduction on $\boldsymbol{x}_Q$, that is suitably used as a discrete transform in the scheme of Fig. 8.1. We now proceed as follows: firstly, we introduce two common compression techniques (one lossless, and one lossy w.r.t. $\boldsymbol{x}_Q$) that may be considered as terms of comparison for this task. Then we discuss a lossy compression scheme based on CS and tune it to attain optimal performances. Finally, we compare the three techniques as tuned as possible to see what are the optimal rates they achieve on ECG signals. We proceed by summarizing the first two schemes.

Huffman Coding

A low-complexity lossless compression scheme considered for this comparison amounts to processing the PCM samples in $\boldsymbol{x}_Q$ with standard Huffman coding (HC) [39], a simple and widely used entropy coding technique. HC takes a binary string as an input, and encodes it by a prefix-free variable-length code. This code entails the construction of an optimal *codebook* based on the probability distribution of the input, i.e., the most probable symbol in the input string is encoded by the shortest codeword, and so on in the construction of a binary tree that uniquely encodes all nonzero probability symbols.

The codebook is here assumed to be known a priori and is practically trained on the empirical distribution of a very large set of PCM samples (in particular, of a

[1]The integration of non-uniform, minimum-distortion quantizers at the ADC is a technologically complex task; for this reason, we limit this study to uniform scalar quantizers.

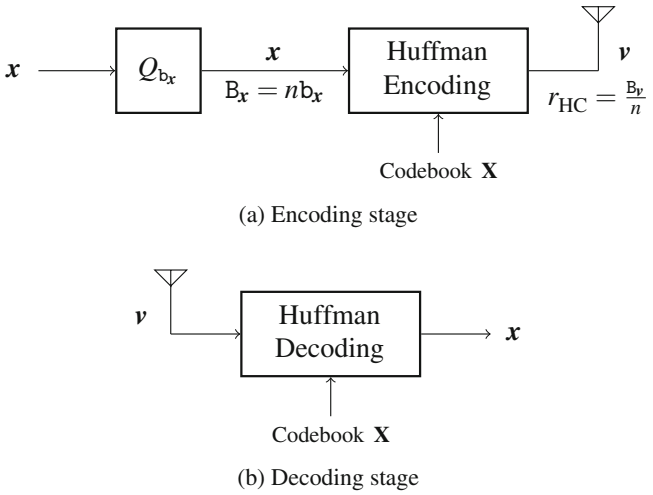


Fig. 8.2 A system-level view of Huffman coding

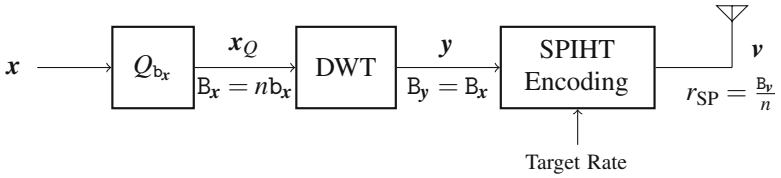
large dataset of ECG samples). Since this training set might not contain all possible words an *escape* codeword is added to the codebook, followed by $\lceil \log_2 q \rceil$ bits to represent all of the q symbols not appearing in the above set. Thus, the "quality loss" here is only due to the inevitable quantization of $\boldsymbol{x}$ into $\boldsymbol{x}_Q$ caused by ADC.

This compression scheme requires a minimum amount of computational resources: after the signal is quantized, we straightforwardly encode $\boldsymbol{x}_Q$ by using a lookup table that maps its fixed-length words to variable-length codewords in the encoded bitstream $\boldsymbol{v}$. Thus, provided there is enough storage available at the sensor node to allocate the optimal codebook, HC achieves a code rate r_{HC} with no fixed-point signal processing operation involved, and in an absolutely inexpensive fashion as it amounts to a suitably initialized lookup table. This scheme is depicted in Fig. 8.2.

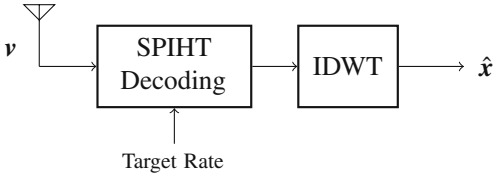
Set Partition Coding of Wavelet Coefficients

To the other end of our complexity comparison, we consider the application of Set Partition coding in Hierarchical Trees (SPIHT, [27]) that serves as a basic building block following the application of a wavelet transform in digital signal compression schemes (see Fig. 8.3). The SPIHT encoder operates on the DWT coefficients of $\boldsymbol{x}_Q$ (in particular, the authors of [27] suggest the optimality of 9/7 bi-orthogonal wavelets [28] for ECG signals) by constructing a map of their significance w.r.t. their magnitudes and dependencies in a tree representation of the wavelet coefficients.

The critical arithmetic complexity in this lossy encoding is in implementing the chosen DWT that, as efficient (e.g., as [24]) and specific (e.g., as [6]) as it can be made, requires at best fixed-point multiplications with carefully quantized filter



(a) Encoding stage



(b) Decoding stage

Fig. 8.3 A system-level view of set partition coding in hierarchical trees

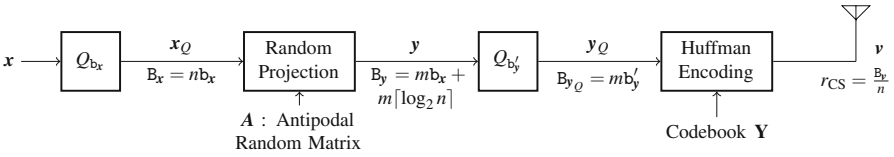
coefficients. Such a complexity is considered high for straightforward integration into low-resources digital processing stages for sensor nodes; we will report its attained code rates r_{SP} as a reference case that is generally expected to outperform the other schemes discussed in this chapter.

8.1.2 Lossy Compression by CS

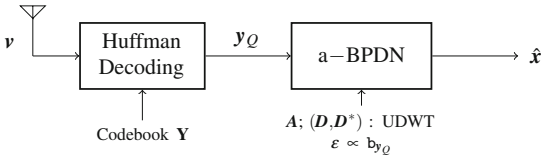
8.1.2.1 The Encoding Stage

As mentioned in Sect. 8.1.2 a dimensionality reduction is simply obtained as $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$; in this chapter, we will refer to $\boldsymbol{A}$ as the *encoding matrix* to emphasize that it is implemented in a digital-to-digital fashion; in particular, we let $\boldsymbol{A} \in \{-1, +1\}^{m \times n}, m < n$ since we want to implement it in very low-complexity digital hardware.

The proposed encoding stage is reported in Fig. 8.4a and summarized as follows. As dimensionality reduction is here performed in the digital domain, we will operate on quantized $\boldsymbol{x}_Q$; thus, the encoding operation is actually $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_Q$ represented by m digital words. Their wordlength will be $\mathtt{b}_y = \mathtt{b}_x + \lceil \log_2 n \rceil$ bits since each $\boldsymbol{y}_j$ is obtained by an inner product of the PCM samples in $\boldsymbol{x}_Q$ with a vector of sign changes, i.e., $\boldsymbol{y}_j = \pm \boldsymbol{x}_{Q_0} \pm \boldsymbol{x}_{Q_1} \pm \ldots \pm \boldsymbol{x}_{Q_{n-1}}$. This operation can be conveniently



(a) Encoding stage



(b) Decoding stage

Fig. 8.4 A system-level view of digital signal compression by CS

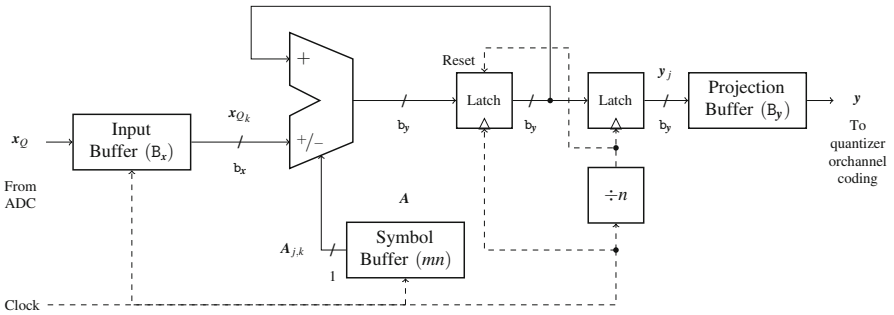


Fig. 8.5 A digital, multiplierless hardware implementation of the CS encoding stage RAE encoding matrices, using a single accumulator and fixed-point arithmetic. The buffers are local registers of size denoted by $(\cdot)$ bit; the *dashed lines* denote synchronization signals

mapped on mn cycles of a single accumulator, i.e., by an extremely simple and multiplierless scheme as that of Fig. 8.5.

To reduce the rate of the encoded bitstream, we quantize $\boldsymbol{y}$ by a second uniform scalar quantizer as $\boldsymbol{y}_Q = Q_{b'_y}(\boldsymbol{A}\boldsymbol{x}_Q)$ with $b_y{}' \leq b_y$. $Q_{b'_y}$ is scaled to operate in the range of $\boldsymbol{y}$ but keeps only $b_y{}'$ MSBs from each $\boldsymbol{y}_j$. We also note that the alternative of a non-uniform, minimum-distortion scalar quantizer (i.e., a Lloyd-Max quantizer [32]) could indeed be pursued here as only requiring the implementation of a suitable pre-distortion prior to uniform quantization, whereas vector quantization commonly requires more computational effort on the encoder [18]. In a low-complexity perspective we assume that a uniform quantizer is the simplest choice for this task, although other alternatives are indeed worth exploring (and has already been addressed in some works [22, 44]).

To further compress the encoded bitstream we evaluate the option of applying lossless HC with an optimal codebook trained on the empirical PMFs of each element of $\boldsymbol{y}_Q$, that are approximately Gaussian-distributed due to the mixing effect of $\boldsymbol{A}$. Thus, the encoded bitstream $\boldsymbol{v}$ attains a code rate r_{CS} that depends on $(m, \mathtt{b}_x, \mathtt{b}_y)$, the choice of $\boldsymbol{A}$ and the presence or absence of HC in the encoder/decoder.

8.1.2.2 Rakeness-Based Encoder Design

We now proceed to discuss a further degree of freedom in the choice of $\boldsymbol{A}$ as drawn from a suitably chosen, rakeness-based design of $\boldsymbol{A}$ rather than the non-adaptive choice of i.i.d. random symbols. Although assuming $\boldsymbol{A} \sim \mathrm{RAE}(\boldsymbol{I})$ fits equally well any kind of signal [7], we have shown how rakeness and signal localization can be leveraged to design $\boldsymbol{A} \sim \mathrm{RAE}(\mathscr{A})$ that maximizes the average energy of $\boldsymbol{y}$ and allows for lowering the requirements on the minimum m to attain successful signal recovery. We therefore use it as an encoder-side option to reduce the code rate, at the price of some side information as follows.

To carry out the design of $\boldsymbol{A}$ (i.e., the choice of $\mathscr{A}$), we recall that the quasi-stationary behavior of ECGs allows for a meaningful estimation of the signal's correlation matrix $\mathscr{X} = \boldsymbol{U}\boldsymbol{M}\boldsymbol{U}^*$, where $\boldsymbol{U}$ is equivalent to that of optimal transform-coding by the Karhunen–Loève Transform (KLT) [16], that is essentially identical to PCA. In many applications, the stationarity of $\mathscr{X}$ over time could be insufficient, and the update and transmission of its estimate $\widehat{\mathscr{X}}$ would make the KLT disadvantageous w.r.t. computing other transforms. However, for this particular type of biosignal the estimated $\mathscr{X}$ is not only stable, but typically also attains high values of $\mathscr{L}_{\boldsymbol{x}}$ in (1.5).

To leverage this structure, given the quasi-stationary behavior of ECGs, we here apply the synthesis methods and the design flow of Sect. 3.2, using an estimate of the correlation matrix $\widehat{\mathscr{X}}$ (depicted in Fig. 8.6a) given by the sample correlation of a large training set of 10^4 instances of $n = 256$ samples ECG window $\boldsymbol{x}$. Thus, the

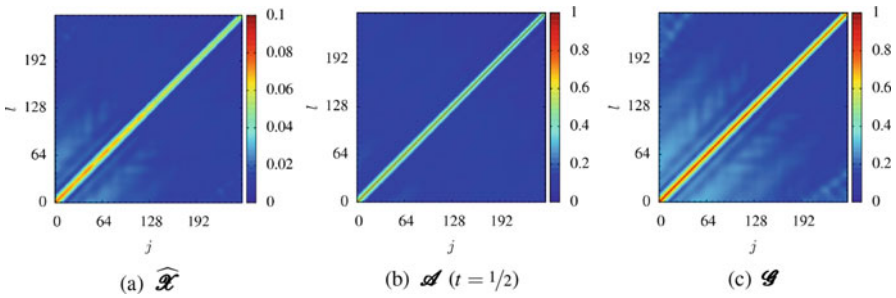


Fig. 8.6 Correlation matrices depicting the phases of a rakeness-based design flow for the encoding of ECG signals: from (**a**) to (**c**), the estimated correlation $\widehat{Xc}$ is fed into the design flow to yield a target correlation $\mathscr{A}$, which is synthesized as $\mathrm{RAE}(\mathscr{A})$ by clipping the realizations of a Gaussian ensemble $\mathrm{RGE}(\mathscr{G})$

synthesis problem is solved with the purpose of defining a RAE($\mathscr{A}$) from which $\boldsymbol{A}$ can then be generated. To do so, we follow precisely the scheme of Fig. 3.3, setting the parameter $t = 1/2$; to document this process, we report $\mathscr{A}$ in Fig. 8.6b. Given this target correlation for the RAE, we use the antipodal generation method in the stationary case described in Sect. 5.3.1. By plugging the obtained $\mathscr{A}$ as $\mathscr{V}$ in (5.2), we obtain $\mathscr{G}$ as depicted in Fig. 8.6c, which can then be used as the correlation of a Random Gaussian Ensemble (RGE) with zero-mean and the designed covariance matrix $\mathscr{G}$, RGE($\mathscr{G}$). $\boldsymbol{A} \sim$ RAE($\mathscr{A}$) can be obtained by clipping provided $\mathscr{G} \succeq 0$; this feasibility is widely discussed in Chaps. 4 and 5, and turns out to be practically true for all considered cases of $\mathscr{A}$ at parameter $t = 1/2$ resulting from ECG correlation matrices $\mathscr{X}$. In a sense, this whole synthesis strategy can be considered similar to a KLT with antipodal-valued random projection vectors, yet more robust due to how the rakeness-localization trade-off is tackled.

Thus, the resulting $\boldsymbol{y}$ will have by design (i.e., by the very definition of our adaptation criterion, that is rakeness) a larger variance than that produced by the classic RAE($\boldsymbol{I}$) case, so the following quantizer and Huffman code in Fig. 8.4a will require an adaptation to the new distribution of $\boldsymbol{y}$ to yield an appropriately quantized $\boldsymbol{y}_Q$.

8.1.2.3 The Decoding Stage

Since $\boldsymbol{A}$ is a dimensionality reduction and $\boldsymbol{y}$ undergoes a second quantization, this scheme is by definition lossy. However, we have previously recalled some theoretical guarantees that relate the sparsity of $\boldsymbol{x}$ w.r.t. $\boldsymbol{D}$ and the minimum number of measurements $\overline{m} = O(\kappa \log(p/\kappa))$ ensuring that $\boldsymbol{x}$ may be stably recovered from $\boldsymbol{y}_Q$ even in the presence of quantization noise. This guarantee allows us to consider the possibility that, when $\boldsymbol{x}$ is sufficiently sparse w.r.t. $\boldsymbol{D}$, some denoising may indeed be possible by a suitable choice of dictionary and recovery algorithm.

The decoding stage discussed in this section is reported in Fig. 8.4b. As a recovery algorithm we indeed considered *analysis-basis pursuit with denoising* (a-BPDN), i.e., as defined in (2.3) with $\varepsilon \geq 0$ set proportionally to the energy of the quantization noise introduced in the processing chain by both Q_{b_x} and $Q_{\mathrm{b}'_y}$.

As for $(\boldsymbol{D}, \boldsymbol{D}^*)$ we assume they are the synthesis and analysis operators of an *undecimated DWT* (also known as "translation invariant" or "redundant" DWT), i.e., an overcomplete transform whose operators form a *tight frame* (see Sect. 1.4), that is obtained by modifying the filter-bank and removing the decimation/up-sampling blocks to obtain an oversampled DWT instead of the usual critically sampled DWT (which results in $\boldsymbol{D}$ being orthonormal). This arrangement of a signal recovery algorithm and an analysis-sparsity prior was shown to be robust w.r.t. additive noise in several contributions [8, 12, 42]. We precisely aim at leveraging this robustness to mitigate the impact of quantization on the quality of $\hat{\boldsymbol{x}}$.

8.1.3 Performance Evaluation

In this section we evaluate the performances after decoding of the schemes in Fig. 8.4, with an emphasis on CS and its variants. We adopt the average reconstruction signal-to-noise ratio of the decoded signal (ARSNR, (2.4)) between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ as a performance index, where $\hat{\boldsymbol{x}}$ is taken at the decoded output of each of the considered techniques. We now proceed to specify some details of how this evaluation is carried out.

Dataset and Samples' Quantization

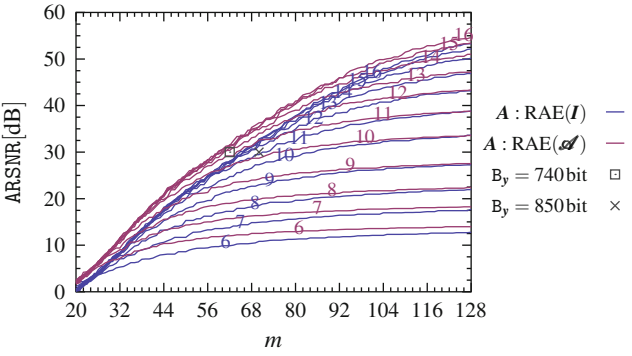
We here use a synthetic ECG generator [33] to produce 10^4 training instances of $\boldsymbol{x}$ with $n = 256$, corresponding to 1 s windows sampled at 256 Hz. The parameters of the generator are randomly drawn to obtain a training set oscillating at various heart rates and not corrupted by intrinsic or quantization noise. Each window is then quantized to its PCM samples $\boldsymbol{x}_Q$ at $\mathtt{b}_{\boldsymbol{x}}$ bps. Since the ECG PCM samples generally have a high crest factor $\mathrm{CF} = 20\log_{10}\frac{\sqrt{n}\|\boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_2} \approx 11$ dB they are non-uniformly distributed in the quantizer range. Thus, the SNR w.r.t. uniform white quantization noise is estimated as $\mathtt{SNR}_{Q_{\mathtt{b}_{\boldsymbol{x}}}}[\mathrm{dB}] = 10\log_{10}\frac{\hat{\mathbf{E}}[\|\boldsymbol{x}\|_2^2]}{\hat{\mathbf{E}}[\|\boldsymbol{x}_Q-\boldsymbol{x}\|_2^2]} \approx 6.02\,\mathtt{b}_{\boldsymbol{x}} - 11$ dB (as will be reported in Fig. 8.8a,b) where the second term is indeed due to the ECG signals' high crest factor.

Decoding Stage Details

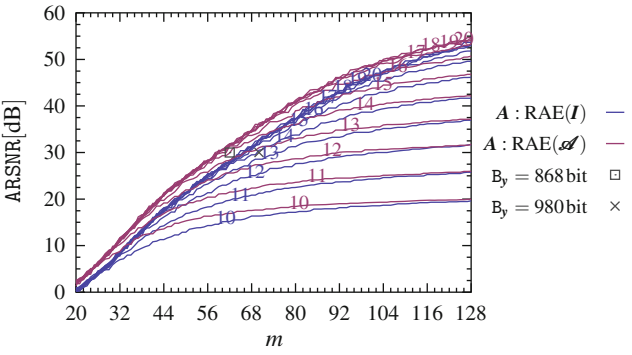
The choice of a suitable wavelet family for the UDWT and of a decoding algorithm for solving (2.3) is crucial for a fair evaluation of CS. We here assume that $(\boldsymbol{D}, \boldsymbol{D}^*)$ are those of the Symmlet-6 UDWT with $J = 4$ sub-bands (i.e., $p = (J+1)n$) [28, Chapter 5.2], and adopt this transform for signal recovery. For what concerns a-BPDN, we solve (2.3) by the UnLocBox implementation (i.e., [40]) of Douglas–Rachford splitting [10] with the data fidelity constraint of (2.3) tuned to the noise norm $\varepsilon = \|\boldsymbol{y}_Q - \boldsymbol{A}\boldsymbol{x}\|_2$ and ensuring that the algorithm converges up to a relative variation of 10^{-7} in the objective function.

Measurements' Quantization Effects

The main noise sources in the evaluated coding schemes are the uniform PCM quantizers $Q_{\mathtt{b}_{\boldsymbol{x}}}, Q_{\mathtt{b}'_{\boldsymbol{y}}}$. While the former is common to all evaluated schemes, the latter is only used in the CS encoding to reduce each element of $\boldsymbol{y}$ to $\mathtt{b}'_{\boldsymbol{y}} < \mathtt{b}_{\boldsymbol{y}}$ bits. Since these measurements are approximately Gaussian-distributed (as partly discussed in Sect. 3.4) $\mathtt{b}'_{\boldsymbol{y}} = \mathtt{b}_{\boldsymbol{y}} = \mathtt{b}_{\boldsymbol{x}} + \lceil\log_2 n\rceil$ would largely exceed the precision actually



(a) $\boldsymbol{y}_Q$ is quantized with $\mathtt{b}_y = \mathtt{b}_x$ bits (values of $\mathtt{b}_y$ on curves)



(b) $\boldsymbol{y}_Q$ is quantized with $\mathtt{b}_y = \mathtt{b}_x + \lceil \frac{1}{2} \log_2 n \rceil$ bits (values of $\mathtt{b}_y$ on curves)

Fig. 8.7 ARSNR for the RAE($\boldsymbol{I}$) (*dashed*) and rakeness-based RAE($\mathscr{A}$) (*solid*) CS with different quantization policies. For both figures $\mathtt{b}_x = 6, \ldots, 16$. For $\mathtt{b}_x = 10$, the points corresponding to bit budgets that allow an ARSNR ≈ 30 dB highlight the i.i.d. RAE($\boldsymbol{I}$) case (*cross*) and RAE($\mathscr{A}$) case (*square*)

required to represent $\boldsymbol{y}$ with negligible losses. Thus, to explore the effect of $\mathtt{b}_y$ we (*i*) encode by CS the ECG training set and train $Q_{\mathtt{b}'_y}$ with either $\mathtt{b}'_y = \mathtt{b}_x$ or $\mathtt{b}'_y = \mathtt{b}_x + \lceil \frac{1}{2} \log_2 n \rceil$ (*ii*) apply the same operation on 64 new test instances, solve (2.3) and compute ARSNR while varying $m = 20, \ldots, 128$ (up to $m = n/2$), $\mathtt{b}_x = 6, \ldots, 16$. Moreover, we run the very same procedure for rakeness-based CS trained as discussed in Sect. 8.1.3, with a suitably scaled range for $Q_{\mathtt{b}'_y}$ that must compensate for the fact that the measurements have a larger average energy as a result of our adapted CS methodology.

The outcome of this procedure is reported in Fig. 8.7, where it is observed that (*i*) rakeness-based CS with maximum energy RAE($\mathscr{A}$) outperforms standard CS with the RAE($\boldsymbol{I}$) in all the examined cases, as it relies on some a priori information on the signal being acquired; (*ii*) the quality gain obtained by using more bits for both $(\mathtt{b}_x, \mathtt{b}'_y)$ progressively saturates at an ARSNR limit imposed by the sparsity

level of ECG signals; (*iii*) for a fixed value of $\mathtt{b}_{\boldsymbol{x}}$, the total bit budget $\mathtt{B}_{\boldsymbol{y}} = m\mathtt{b}'_{\boldsymbol{y}}$ required to reach an ARSNR target hints at how redundant the chosen quantization policy is. This quantity is highlighted in both Fig. 8.7a,b, and shows how the quality improvement of choosing a more accurate quantizer $Q_{\mathtt{b}'_{\boldsymbol{y}}}$ for $\boldsymbol{y}_Q$ must be matched with a smaller m, and in particular that $\mathtt{b}'_{\boldsymbol{y}} = \mathtt{b}_{\boldsymbol{x}}$ is a better choice for achieving lower code rates with CS.

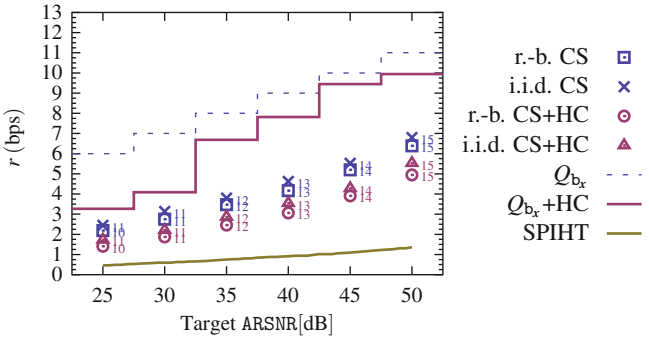
8.1.3.1 Rate Performances

Given the observed quantization effects, to understand which uniform scalar quantizer $Q_{\mathtt{b}'_{\boldsymbol{y}}}$ enables the lowest code rate, $\boldsymbol{y}_Q$ must be post-processed by optimally trained HC. In addition, we here assess how this attained rate, r_{CS}, compares with the rate performances achieved by the other schemes (Fig. 8.4) at some fixed target decoding performances, i.e., $\mathtt{ARSNR}[\mathrm{dB}] = \{25, 30, 35, 40, 45, 50\}$.

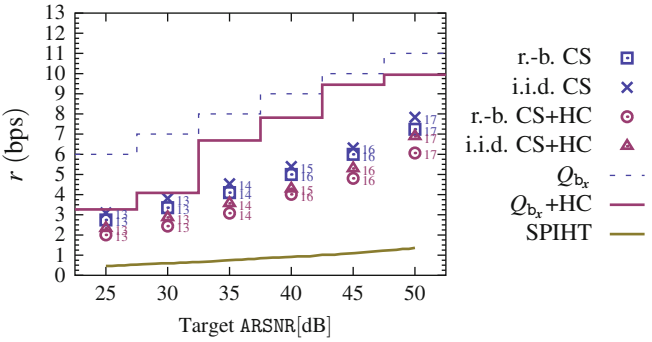
For a fair comparison, SPIHT for ECGs [27] is run from the authors' code by fitting instances of $\boldsymbol{x}_Q$ into full frames of 1024 PCM samples quantized at different $\mathtt{b}_{\boldsymbol{x}}$. The SPIHT encoder takes r_{SP} as an input, which we vary in $[1/n, 2]$; the minimum r_{SP} that guarantees the target ARSNR after decoding is then reported in Fig. 8.8a,b. As a further reference, we report the rates of uniform PCM quantization and its optimal HC, achieving a rate r_{HC}; since it is lossless, achieving an ARSNR target depends on $\mathtt{b}_{\boldsymbol{x}}$. While the average codeword length (and r_{HC}) could be estimated as the entropy of PCM samples, to account for the presence of escape symbols we run this encoding to find the actual r_{HC} of the test set.

These two reference methods are compared with various embodiments of CS (i.e., with or without HC; with different quantization policies; with or without a rakeness-based, maximum energy RAE($\mathscr{A}$) encoding matrix design) in Fig. 8.8a,b. It is observed that the rates attained in Fig. 8.8a are generally lower than those in Fig. 8.8b, thus confirming the small quality loss compared to the rate gain when assuming $\mathtt{b}'_{\boldsymbol{y}} = \mathtt{b}_{\boldsymbol{x}}$. In addition, the use of HC on the measurements reduces significantly the code rate of CS, as also does the use of rakeness-based encoding matrices. Moreover, by considering r_{CS} of rakeness-based CS with HC, Fig. 8.8a shows that an $\mathtt{ARSNR}[\mathrm{dB}] \approx 25\,\mathrm{dB}$ is achieved at $\mathtt{b}'_{\boldsymbol{y}} = \mathtt{b}_{\boldsymbol{x}} = 10\,\mathrm{bit}$ by $r_{\mathrm{CS}} \approx 1.41\,\mathrm{bps}$, while $r_{\mathrm{HC}} = 3.27\,\mathrm{bps}$. At higher ARSNR targets, CS is increasingly advantageous, placing itself at less than 50% of the code rate of PCM with optimal HC.

We conclude that, as a lossy compression, CS can achieve relatively low code rates, at the same time maintaining a globally low computational complexity on the encoder side. Given these low requirements, it lends itself as an agile lossy scheme for resource-constrained signal compression applications. This said, many degrees of freedom are still to be explored to improve upon these results; as mentioned, since the second scalar quantizer is fully digital and can be arbitrarily tuned, Lloyd-Max quantization could be used to reduce the measurements' distortion for a given code rate, exploiting the fact that their statistics are approximately Gaussian-distributed. In addition, we expect that recent developments in the modeling of quantization



(a) $\boldsymbol{y}_Q$ are quantized with $\mathtt{b}'_y = \mathtt{b}_x$



(b) $\boldsymbol{y}_Q$ are quantized with $\mathtt{b}'_y = \mathtt{b}_x + \lceil \frac{1}{2} \log_2 n \rceil$

Fig. 8.8 Achieved code rates of the evaluated compression schemes and their variants for the chosen ARSNR target specifications; the value of $\mathtt{b}'_y$ that allows a given rate is reported to the right of each marker . (**a**) $\boldsymbol{y}_Q$ are quantized with $\mathtt{b}'_y = \mathtt{b}_x$. (**b**) $\boldsymbol{y}_Q$ are quantized with $\mathtt{b}'_y = \mathtt{b}_x + \lceil \frac{1}{2} \log_2 n \rceil$

noise in the signal recovery problem, as done in [21, 36] (at the cost of an additional uniform dithering prior to quantization, as specified therein), could enable even higher recovery quality for this class of signals, and consequently lower code rates for the chosen distortion levels.

8.2 Dual Mode ECG Monitor by Bortolotti et al., 2015

The application discussed in [5] presents an interesting application that uses CS as a basic building block for ECGs signal compression properly designed for both Healthcare (HC) and Wellness (WN) applications. The presented system envisions a dual-mode wearable ECG monitor based on a multi-core DSP for multi-lead ECG compression. Furthermore impact of different technologies for either transmission

or local storage of the evaluated measurements is also analyzed showing the effectiveness of the rakeness-based approach.

8.2.1 System Architecture and Mathematical Model

The processing chain for a wearable biomedical monitoring system is typically split into three phases. First, input biosignals acquisition, followed by a stage aiming to process/compress the acquired digital data and finally management of the processed/compressed input. In the first stage the analog input biosignal is sampled and made available to a digital signal processor (DSP) block for processing. Subsequently the compressed output can be transmitted (to a personal server such as a smartwatch or smartphone) or locally stored for later off-line medical analysis. Motivated by the inherent parallel nature of medical-grade biomedical monitoring, where multi-channel signal analysis is suitable for parallel processing, the authors of this contribution propose a multi-core DSP to process in parallel data acquired from more than one lead or from more than one biosignal. From a technological point of view, emerging non-volatile memories (NVM) allow to have on-chip low-power storage, suitable for keeping a record of medical-grade compressed data on-board, i.e., directly on the device. As a matter of fact, it is quite important from a medical point of view to distinguish the triggering event that marks the distinction between normal, healthy heart activity to a suspicious behavior that requires medical attention.

Hereafter, we describe a unified architecture capable of offering signal qualities suitable for both medical-grade and lifestyle applications leveraging the quality offered by the rakeness-based CS. The envisioned dual-mode ECG monitor is able to handle in highly energy efficient way different application scenarios, such as healthcare and wellness, according to an external command. The main points are the following:

- It presents a dual-mode ECG monitor, suitable for *healthcare* applications, in which the target is medical-grade signal quality, and for *wellness* applications, e.g., heart rate detection, for which a lower signal quality level is acceptable.
- Identify different operating points for healthcare and wellness scenarios with their associated compression ratios. This work highlights that rakeness-based CS, as presented in Chaps. 3 and 4, leads to significant improvements w.r.t. standard CS in terms of data compression and energy efficiency for both application scenarios.
- Provide an up-to-date analysis on the energy gains including transmission and storage impact, considering several possible use-cases.

The main idea is to exploit the trade-off between data compression (i.e., to reduce as much as possible the number of entries in the measurement vector) and the goodness of the reconstruction for two different reconstruction quality standards used as targets for, respectively, HC and WN applications:

- High quality (HQ): the achieved data compression is such that ECG instances are correctly reconstructed for healthcare quality (i.e., medical-grade, with an accurate waveform representation of the heart cycle).
- Low quality (LQ): the achieved data compression is such that the reconstruction quality targets wellness applications (i.e., only tailored to provide accurate heart rate detection).

For a fixed class of signals, and as widely discussed in the prior sections, the reconstruction quality achieved by CS mainly depends on the cardinality of $\boldsymbol{y}$ such that a subset of the measurements used for HQ reconstruction can also be used for the LQ standard.

Before discussing the proposed architecture and related performance we briefly recap the mathematical model presented in first three chapters with specialization to the ECG class of signals. In agreement with the rest of this book, the acquired samples of a given time window are the entries of an $\mathbb{R}^n$ vector $\boldsymbol{x}$ while the encoder stage performs compression by projecting $\boldsymbol{x}$ on a set of m sensing sequences arranged as row of the sensing matrix $\boldsymbol{A}$ such that

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\eta}$$

where $\boldsymbol{\eta}$ models, in an additive fashion and for the sake of simplicity, sources of noise such quantization and other nonidealities in the measurement process. At the decoder stage the original samples are decoded by solving (1.27) adopting the SPGL1 optimization toolbox [4].

In more detail, we here adopt as input signals synthetic ECGs generated by [33][2]; it is then sampled at 256 Hz with 1 s epochs leading to $n = 256$; for each window, additive white Gaussian noise is considered as a model for all nonidealities, with noise power so that the intrinsic signal-to-noise ratio $\texttt{ISNR[dB]} = 45$ dB. Moreover, we adopt as sparsity basis $\boldsymbol{D}$ the orthonormal Symmlet-6; as $\boldsymbol{A} \sim \text{RAE}(\mathscr{A})$ (we will specify later which design is chosen for $\mathscr{A}$), all sensing sequences $\boldsymbol{A}_{j,.}$ are comprised of antipodal values in order to limit the computational burden of the DSP stage. System performance are in terms of ARSNR for both the rakeness-based CS and standard CS are obtained by performing Montecarlo simulations over 600 trials.

For the standard CS approach we always refer to RAE sensing matrices $\boldsymbol{A} \sim \text{RAE}(\mathscr{A})$ with $\mathscr{A} = \boldsymbol{I}$ (in particular, so that their antipodal symbols are all i.i.d.), whereas rakeness-based CS imposes that the rows of the sensing matrix $\boldsymbol{A}$ are designed with a correlation profile $\mathscr{A} \neq \boldsymbol{I}$ evaluated by (3.10), where $\mathscr{X}$ is estimated over 1000 randomly generated synthetic ECG instances. Furthermore, Linear Probability Feedback Processes as discussed in Sect. 5.3.1 are considered for the generation of the sensing matrix rows in this rakeness-based design case.

[2]The setup for synthetic ECG generation is widely described in [29]: the heart rate of each instance is randomly fixed in the range $40 \div 120$ bpm.

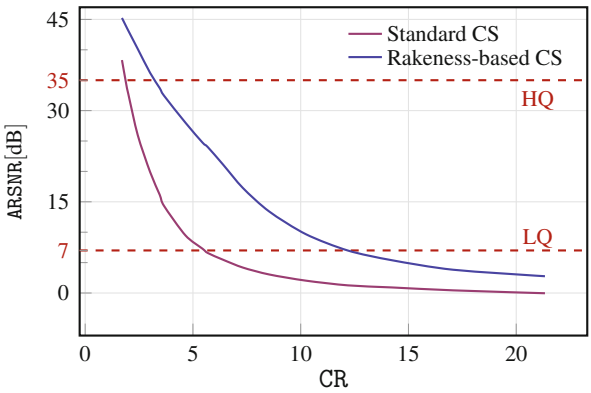


Fig. 8.9 ARSNR performances as a function of CR for synthetic ECGs with both CS approaches. The values of ARSNR that identify two operating points for HQ and LQ ECG reconstructions are reported on the graph

The resulting ARSNR as a function of the CR is shown in Fig. 8.9 where the ARSNR values for both signal quality levels are also reported. Starting from an ISNR $= 45$ dB the HQ standard was marked for ARSNR[dB] ≥ 35 dB (considering 10 dB loss as the price paid for data compression). The associated minimum CRs are 1.91 for standard CS and 3.46 for rakeness-based CS. Remarkably, an increase of CR is directly translated into a reduction of the amounts of bits to be transmitted or stored so that, in a scenario involving the acquisition of more than one channel, the reduced m implies a reduction of the computational load and the "memory footprint" required for each channel to compute its own measurement vector $\boldsymbol{y}$.

For the LQ operating point the reconstruction quality is fixed by a correct heart rate estimation, i.e., the amount of extracted signal information is enough to detect ECG peaks. To determine the minimum number of measurements required to achieve the desired quality, the reconstructed signals were processed by an automatic tool able to count intervals between two successive peaks interval.[3] We consider compatible with the LQ target the minimum number of measurements ensuring that, over 600 s of reconstructed signal, heart rate detection is correct over 98% of the detected peaks. The results show that this rate is always reached with ARSNR ≥ 7 dB that correspond to CR $= 5.45$ for standard CS and CR $= 12.19$ for rakeness-based CS.

To corroborate this analysis, the operating points that were determined considering synthetic ECGs are tested with real ECGs taken from PhysioNet [15], confirming the goodness of the proposed approach. Three seconds of real ECG signal with the corresponding reconstructed tracks for HQ an LQ and for both CS approaches are shown in Fig. 8.10. The same plot reports also the CR value associated with each configuration.

[3]The heart rate estimation was done by ecgBag, available at http://www.robots.ox.ac.uk/~gari/CODE/ECGtools/.

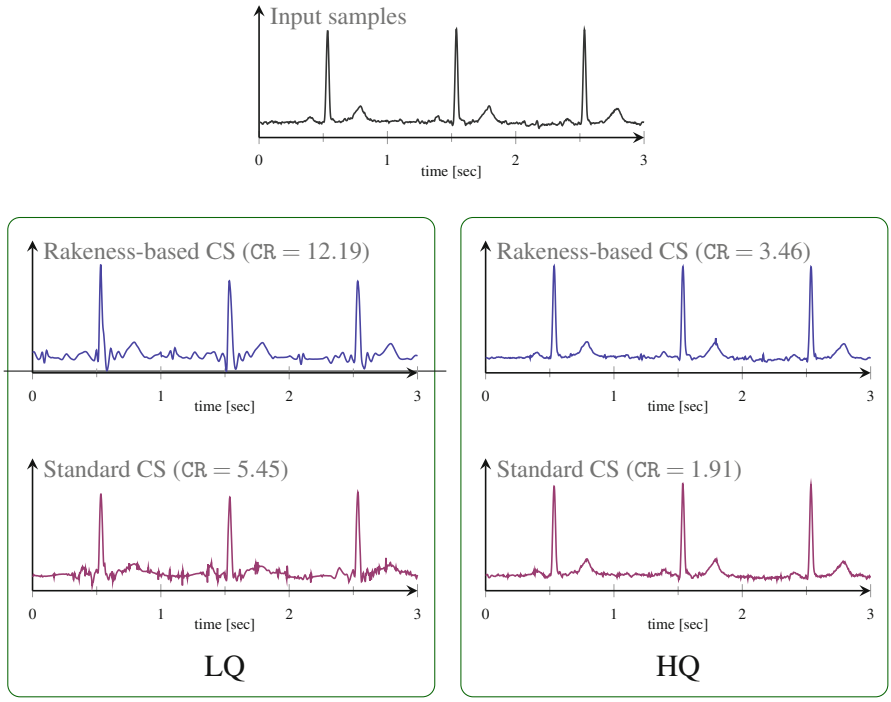
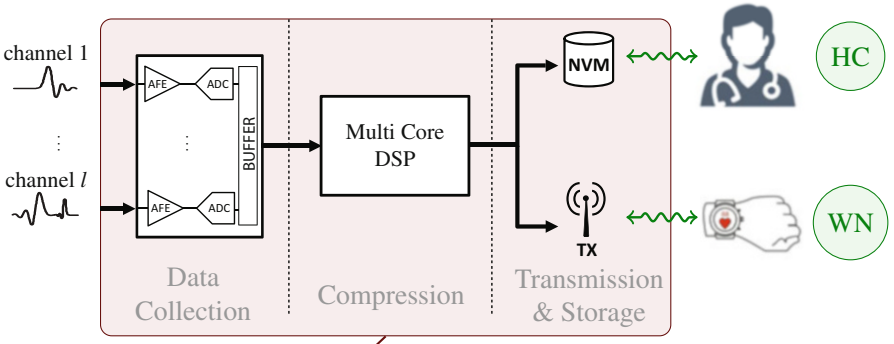


Fig. 8.10 Visual representation of the reconstruction quality for real ECG samples for both CS approaches operating in LQ and HQ. Each plot reports CR value that identifies the couple CS approach and operating point

8.2.2 Hardware Implementation and Energy Performance

A representation of the dual-mode ECG monitor is presented in Fig. 8.11. The reported block scheme is comprised of three separate blocks: the Analog Front-End (AFE), the multi-core DSP (MC-DSP), and the back-end for transmission (TX) or storage in a non-volatile memory (NVM).

In the considered architecture 8-lead biosignals are acquired and sampled by the AFE during the *Data Collection* phase, with a sampling frequency set according to the properties of the biosignal to analyze and the accuracy needed in the considered scenario. Once a set of new samples is ready in the AFE buffer, data are moved to the MC-DSP memory to perform data *Compression*. Data compression is achieved by matrix multiplications where sensing matrices are composed by antipodal values in order to limit the computational burden of the DSP stage. As a consequence during this phase, for most of the time, the whole system is idle thus it is possible to consider a deep low-power state (almost zero power) for both the MC-DSP and the TX & NVM back-end. As a matter of fact, this avoids unnecessary power consumption. The last stage, that is *Transmission & Storage*, manages compressed



		NVM		DSP	WN to HC
WN	SC1	-	LQ	LQ	N
	SC2	-	LQ	HQ	Y
	SC3	-	LQ	HQ	Y
HC	SC1	HQ	LQ	HQ	-
	SC2	HQ	LQ	HQ	-
	SC3	-	HQ	HQ	-

Fig. 8.11 Dual-mode ECG monitor block scheme (*top*) and the operating points for the three different use case scenarios (CS1, CS2, and CS3) for both wellness (WN) and healthcare (HC) applications

data for the current time window depending on the target usage, i.e., the data are either transmitted or stored in the device for future off-line medical analysis.

As described in the previous Sect. 8.2, the systems' operating points are designed for two different reconstruction quality levels, namely HQ and LQ. Such targets define two corresponding families of applications, i.e., the systems operating in HC mode require that the compressed signal achieves the HQ reconstruction quality target, while the WN mode is tuned to meet an LQ reconstruction quality target. These definitions pave the way to different realistic applications for the proposed device, which can be put in context as three use-cases that differ in the provided signal quality and potential medical usage of the device:

- SC1 $\begin{cases} \text{WN} = \text{CS}_{\text{LQ}} + \text{TX}_{\text{LQ}} \\ \text{HC} = \text{CS}_{\text{HQ}} + \text{TX}_{\text{LQ}} + \text{NVM}_{\text{HQ}} \end{cases}$
- SC2 $\begin{cases} \text{WN} = \text{CS}_{\text{HQ}} + \text{TX}_{\text{LQ}} + \text{BUF}_{\text{HQ}}^{\text{1min}} \\ \text{HC} = \text{CS}_{\text{HQ}} + \text{TX}_{\text{LQ}} + \text{NVM}_{\text{HQ}} \end{cases}$
- SC3 $\begin{cases} \text{WN} = \text{CS}_{\text{HQ}} + \text{TX}_{\text{LQ}} + \text{BUF}_{\text{HQ}}^{\text{1min}} \\ \text{HC} = \text{CS}_{\text{HQ}} + \text{TX}_{\text{HQ}} \end{cases}$

In these definitions $\cdot_{LQ}$ denotes the cases where the minimum number of measurements m for the LQ standard is computed (CS_{LQ}), transmitted (TX_{LQ}), or stored (NVM_{LQ}). In a similar fashion, $\cdot_{HQ}$ denotes that m, and thus CR, are increased as to meet the HQ target. The main difference between those definitions is based on the fact that the measurements for the LQ target are a subset of those for the HQ one. As an example, in a HC application for SC2, the MC-DSP first computes all the measurements required to meet the HQ target, and then only a subset is actually transmitted to the user for heart rate monitoring purposes, while the remaining measurements are stored locally for future medical-grade analysis.

The proposed device can switch from WN to HC operation by an external input that can be activated by the patient with arrhythmia symptoms (it is an optional feature in SC2 and SC3). To cope with this requirements BUF_{HQ}^{1min} refers to a circular buffer, located inside the MC-DSP memory, capable of storing the last observed minute in high quality to record the transition event from WN to HC operation. Figure 8.11 (bottom) summarizes behaviors in the three considered scenarios.

To understand the benefits introduced by rakeness-based CS, the energy required for compression and transmission/storage of a 1 s window may be analyzed as a measure of system-wise energy efficiency. The MC-DSP architecture has been modeled and integrated in a SystemC-based cycle-accurate virtual platform, with back-annotated power numbers for the architectural elements extracted from an equivalent register transfer logic architecture (RTL-equivalent)[13]. The design corner for the power numbers are (RVT,25 C,0.6 V) @ 10 MHz in a 28 nm FDSOI technology. A reverse body bias voltage $V_{RBB} = 1$ V is also considered to reduce the leakage contribution during the idle phases. Moreover, considering that the compression task is memory-bound by nature, the requirements in terms of core-memory bandwidth imply higher supply voltage for the memory (0.8 V) in order to sustain the throughput. Remarkably, the execution time discrepancy between the SystemC and the RTL platforms is less than 7%. Due to the multi-lead nature of the system a 1 KB instruction memory and a stack portion of 512 B per core is considered. In terms of data storage, the memory requirements to allocate the input samples amount to 256 samples/s and 8 channels at a 12-bit ADC resolution (per sample). Static data allocation is performed by means of cross-compiler attributes and linker script sections. The remaining memory footprint contributions (output and sensing data structures) depend on the proper CS approach and on the quality of service. Detailed numbers are shown in Table 8.1, along with the execution cycles required to perform CS approaches for both HQ/LQ operating points. As already mentioned, the sensing sequences associated with the LQ quality standard consist of a subset of the HQ sensing matrix rows; this limits the overhead of the dual-mode operation in terms of sequence storage and computation.

For the scenarios SC2 and SC3, the impact of the buffer to track the transition event (BUF_{HQ}^{1min}) corresponds to 69.4 KB. The data reported in Table 8.1 also highlight the benefits introduced by the rakeness-based approach. In particular:

- Rakeness-based CS reduces the amount of measurements on thus it reduces also the memory requirements;

Table 8.1 Execution cycles and memory footprint requirements for both CS approaches in LQ/HQ operating points

	CS approach	DSP time (cycles)	Mem.×Output (B)	Mem.×Sensing (KB)
LQ	Standard CS	109642	752	11.75
	Rakeness-based CS	49048	336	5.25
HQ	Standard CS	312246	2144	33.5
	Rakeness-based CS	172428	1184	18.5

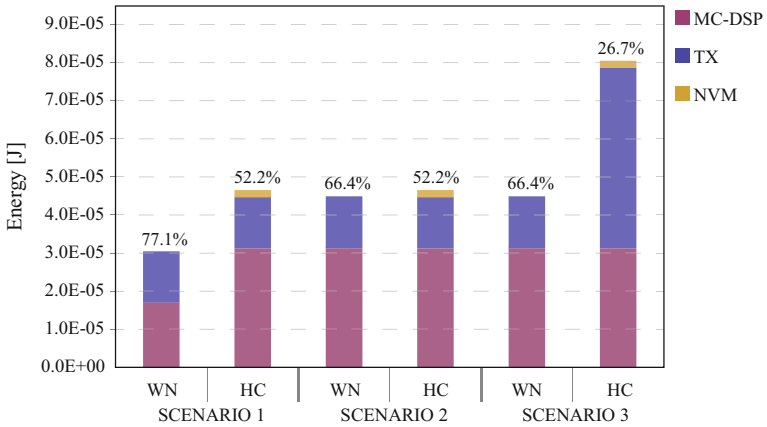


Fig. 8.12 Energy/1 sec-window for the different scenarios (SC1,SC2,SC3) and operation modes (HC,WN)

- Computing less measurements w.r.t. standard CS implies a reduction of the execution time (approaching $\approx 55\%$ for the LQ target) which positively affects the computation costs.

In addition, let us consider the energy requirements of the last stage in our biomedical monitoring system, i.e., the resources spent for storage or transmission. For the transmission subsystem, a Low-Energy (LE) Bluetooth transceiver was considered. The figure of merit to compute these resource requirements is the energy per transmitted bit, which for Bluetooth LE is $E_{TX} = 5\,\text{nJ/bit}$ [26], while for the storage technology the cost per bit is set at $E_{NVM} = 0.1\,\text{nJ/bit}$ [43]. Note that this analysis does not account for the AFE contribution, as it does not depend on either the used CS approach or the considered scenario and is therefore not taken into account.

The results of this evaluation in terms of energy for computation, storage, and transmission are shown in Fig. 8.12, which reports the energy consumed in the different use case scenarios (SC1,SC2,SC3) operating in the HC and WN modalities during a 1 sec compression window and when rakeness-based CS is employed. The bars report the stacked contribution of the MC-DSP and the NVM & TX back-end.

On top of each bar we report the energy gains, taking as the baseline a biomedical system performing standard CS and transmitting in HQ the compressed ECG data.

8.3 Zeroing for HW-Efficient CS in WSNs by Mangia et al., 2016

We here present a work that focuses further on the design of a sensor node for wireless body sensor networks based on the MC-DSP platform. The architecture presented by Mangia et al. in [31] is the same used in [5] whose block scheme is reported in Fig. 8.11 (top). Its focus is on a multi-channel biosignal sensor that takes care of the operations of acquisition, data compression and output transmission, or storage by leveraging CS.

One of the main results in [5] is that rakeness-based CS can reduce the computational burden of algorithms for data compression as well as the energy budget to transmit or locally store the compressed signals. Furthermore it is clarified that, although CS is a good candidate to reduce the computational cost for data compression, the energy spent on this task is not negligible w.r.t. the energy cost of transmission/storage; since sensor nodes are usually battery-powered, minimizing this energy is a fundamental issue, which is best tackled by means of cross-stage system-level optimization. By reducing the number of measurements, the rakeness-based CS paradigm achieves this near-optimality, particularly when combined with an additional design optimization named *zeroing* which we present below with the goal of reducing the global energy requirements. Interestingly, the analysis shows results for different strategies in transmission and storage including emerging technologies, i.e., the trade-off between energy for data compression and measurements' dispatch is explored as driven by the technology chosen to implement the last stage.

Chapter 4 discusses this topic offering approaches to limit the operational cost of the compression stage with very limited performance degradation in terms of data compression for a proper quality of service. In particular the computational burden is limited by imposing that a non-negligible amount of entries in $\boldsymbol{A}$ are zero. To achieve this goal, different strategies in the positioning of the zero entries of $\boldsymbol{A}$ are proposed depending on the properties of the DSP running this CS-based encoding stage. Furthermore, Sect. 4.2 presents parsimonius encoding stategies which include adaptivity in the sensing sequences to the signal class being acquired.

As aforementioned, we will dub this approach "zeroing" as we refer to the fact that it only sets the null value to a subset of entries in $\boldsymbol{A}$, with a clear reduction of the energy spent by the encoder to process the acquired samples since $\boldsymbol{A}_{j,k} = 0$ means that no operation is needed. The main focus will therefore be to find an optimal positioning for the zeros by means of rakeness-based CS, i.e., as to preserve high ARSNR recovery quality; this essentially involves an application of the design

flow in Sect. 4.2. Our results shall provide additional design guidelines for what concerns energy consumption and monitoring time, up to consider approaches to further squeeze the energy spent by the whole sensor node.

In more detail, the design of the sensing matrix shall follow two steps. Firstly, rakeness-based CS is considered to obtain antipodal random matrices with an $n \times n$ covariance matrix $\mathscr{A}$ designed starting from an estimation of the correlation profile characterizing the signal class being acquired, $\mathscr{X}$. After that, the antipodal matrix $\boldsymbol{A}$ is altered by randomly zeroing some entries and thus allowing the generation of $\boldsymbol{A} \in \{+1, 0, -1\}^{m \times n}$.

Resuming the discussion in Sect. 4.1 we recall the design parameter OT, i.e., *output throttling*, which counts the nonzero entries in each column of $\boldsymbol{A}$ such that $0 < \mathtt{OT} \leq m$. For each k-th column $\boldsymbol{A}_{\cdot,k}$, $k = 0 \ldots n-1$, $m - \mathtt{OT}$ randomly chosen entries are set to zero, leaving only OT nonzeros per column. At the end of the process, the computational complexity of the measurement vector $\boldsymbol{y}$ evaluation is reduced from $m \cdot n$ signed sums to $\mathtt{OT} \cdot n$ signed sums.

The clear drawback is that zeroing alters the statistical characterization of $\boldsymbol{A}$ initially imposed by the rakeness approach; clearly, the smaller the OT value, the higher the introduced distortion will be. As a consequence, a reduction of the benefits introduced by rakeness-based CS is expected, with the ARSNR performance eventually degrading back to that of standard CS for low OT values. In a case study where the signals of interest are real-world single-lead ECGs, we will see that:

- the performance (in terms of ARSNR) of the zeroing approach is higher than the standard approach for almost all considered value of OT;
- the ARSNR for a fixed m decreases with OT, while energy requirements are increasing with OT.

The performance in terms of ARSNR is here evaluated over real ECG signals available from the MIT-BIH arrhythmia on-line database [15], in particular, the results for the first 71.1 s of record number 101 are presented as a reference. This signal is sampled at 360 Hz and quantized with a 11-bit ADC, so that it is possible to estimate its ISNR $\approx$ 38.5 dB. This input signal is then split into 50 time windows with $n = 512$ samples each, leading to ≈ 1.42 s per window. For each value of m, we generate a unique sensing matrix $\boldsymbol{A}$ as selected by means of preliminary tests on synthetic ECG tracks [33]. The generation of $\boldsymbol{A}$ follows one of the options below:

- for standard CS, $\boldsymbol{A} \sim \mathrm{RAE}(\boldsymbol{I})$ is the antipodal random matrix ensuring the best ARSNR performances among the trials;
- for rakeness-based CS, synthetic ECGs are used to estimate the correlation profile $\mathscr{X}$ required by (3.10) in order to obtain $\mathscr{A}$. After that, we draw $\boldsymbol{A} \sim \mathrm{RAE}(\mathscr{A})$ from a pool of sensing matrices distributed according to this ensemble, choosing the one which maximizes the ARSNR over the trials. For a fair comparison, $\mathscr{X}$ was estimated over synthetic ECGs rather than the real ECG trials used in this comparison;
- for the zeroing CS, the matrix used to test rakeness-based CS performance was zeroed by randomly set to zero $m - \mathtt{OT}$ entries in every column.

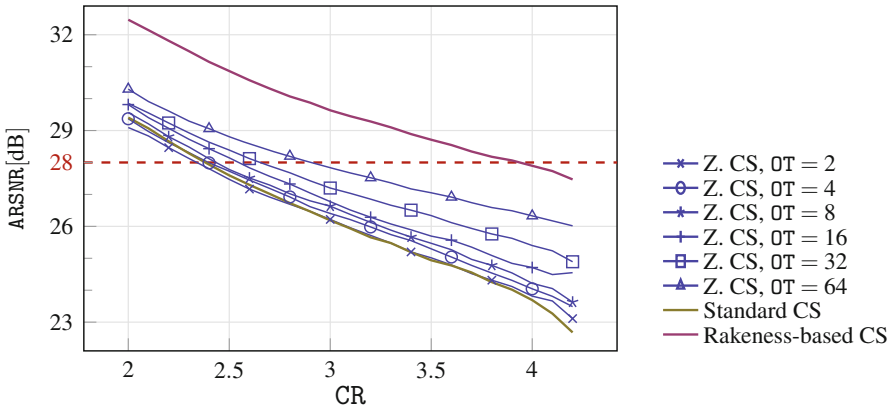


Fig. 8.13 Performance in terms of ARSNR as a function of CR for standard CS, rakeness-based CS, and zeroing CS with $\mathtt{OT} = \{2, 4, 8, 16, 32, 64\}$. Picture highlights $\mathtt{ARSNR} = 28$ targeting the proposed operating point

As a decoding procedure (2.3) was considered, i.e., with the reconstructed signals being obtained by promoting a sparse signal model on a UDWT in the family of Symmlet-6 wavelets with 4 sub-bands [28, Chapter 5.2]. As in Sect. 8.1.2.3, the decoder implementation uses Douglas-Rachford splitting as provided by UNLocBoX [8, 10, 40].

The results in terms of ARSNR are shown in Fig. 8.13 for the three considered approaches and with different values of OT. As expected, rakeness-based CS outperforms all other approaches in terms of ARSNR for all considered CR values. Furthermore, performance for the zeroing CS is increasing with OT and drops to standard CS only for $\mathtt{OT} = 2$. This results confirms that CS suffers from the introduced perturbation in the statistics of the rows of $\boldsymbol{A}$, corroborating the idea that a carefully optimized zeroing could actually prevent such performance degradation.

Another important issue involved in the zeroing approach is its algorithmic implementation, which can be made quite efficient for a fixed OT and for both standard and rakeness-based CS. Since zeroing yields sparse sensing matrices, the trivial double loop implementing the measurements' evaluation (i.e., $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$) can be column-wise unrolled in a way similar to such reported in Table 4.1. The main difference in the software implementation of both approaches is in how the sensing matrix is locally stored. In the double loop implementation its entries are simply allocated in the data memory; on the other hand, the column-wise unrolled implementation stores only the information related to the position and sign of the nonzero elements of $\boldsymbol{A}$.

To compare all mentioned CS approaches, a proper "quality of service" is assigned and then comparison can be done in terms of compression ratio, memory footprint end energy cost per processing, and for transmission/storage. Starting from an $\mathtt{ISNR} \approx 38.5$ dB the target is set to $\mathtt{ARSNR[dB]} = 28$ dB, with the corresponding operating points reported in Table 8.2. The same table also reports, for all the

Table 8.2 MC-DSP data memory footprint requirements for standard CS, rakeness-based CS, and zeroing CS (with OT = 2, 8, 16, 32, 64), targeting ARSNR = 28 dB

CS approach	DSP implementation	m	CR	$\boldsymbol{y}$ memory footprint (B)	$\boldsymbol{A}$ memory footprint (KB)
Rakeness-based CS	Double loop	130	3.94	2080	65
Zeroing CS OT = 64	Column-wise unrolled	175	2.93	2800	64
Zeroing CS OT = 32	Column-wise unrolled	192	2.67	3072	32
Zeroing CS OT = 16	Column-wise unrolled	200	2.56	3200	16
Zeroing CS OT = 8	Column-wise unrolled	209	2.45	3344	8
Zeroing CS OT = 2	Column-wise unrolled	218	2.35	3488	2
Standard CS	Double loop	256	2.00	4096	128

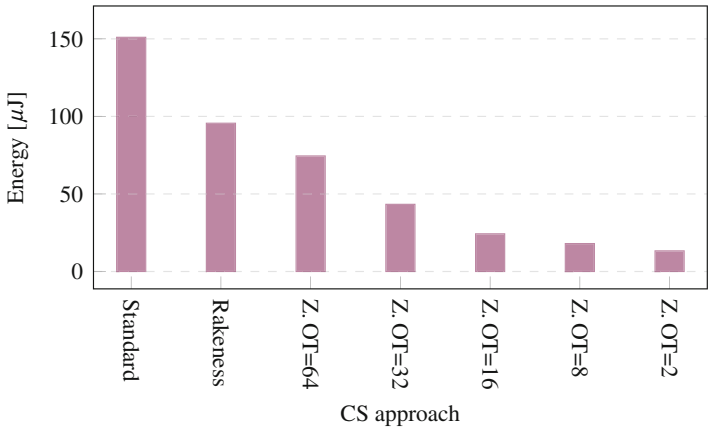


Fig. 8.14 Energy spent in the compression stage for standard CS, rakeness-based CS, and zeroing CS for different OT values

considered approaches, the memory footprint in the DSP data storage for both the output measurements and the sensing matrix allocation.

As for the application described in the previous sections, in order to quantify the energy spent in the DSP block, RTL simulations were run to profile the power consumption of the architectural elements to be later back-annotated inside the power models of a SystemC simulator where the design corner is the same as before. Figure 8.14 shows, for each CS use-case, the energy cost for multi-channel input data compression. The results show that both rakeness-based CS and zeroing CS improve the energy efficiency when compared to standard CS. Indeed, w.r.t. standard CS, $\approx 37\%$ less energy was consumed by rakeness-based CS while zeroing CS with OT $= 2$ leads to an energy gain approaching $\approx 91\%$. This is achieved, thanks to the few nonzero entries in the sensing matrix, coupled with an efficient implementation of this strategy.

8.3.1 Energy Analysis of Transmission/Storage Phase

To complete the previous analysis, we now estimate the energy consumption of the last stage for both cases: on one hand, wireless transmission of the outputs, and on the other hand, locally stored outputs with either widespread or forthcoming high-efficiency storage technologies.

For the case of wireless transmission of the compressed signals, the considered protocols ranged between power-hungry Near-Field Communication (NFC) to the more efficient Narrow-Band (NB) solution. In the storage scenario a non-volatile memory (NVM) is used to locally storage the compressed data; to this end, different promising technologies were considered ranging from the Resistive RAM (ReRAM) to the Conductive Bridging RAM (CBRAM). Table 8.3 reports the energy per bit for all these technologies.

As a first scenario, consider a last stage aiming at wireless transmission of the compressed data. The first set of results relies on the total energy consumed by the bio-sensing node to compress and transmit a window of samples with a compression ratio guaranteeing $\texttt{ARSNR[dB]} = 28$ dB. For this setting, the results are in Fig. 8.15, which refers to the transmission technologies listed in Table 8.3 and reports the total energy cost. The performances evaluated here account for all considered CS approaches, and for the case where no compression algorithm was run and the sensor node simply transmits the raw data. The same plots highlight also how the energy cost is split into the two main contributions, i.e., DSP (bottom part of each bar) and transmission (top part of each bar). It is then evident that the lowest transmission energy is always attained by rakeness-based CS; indeed, zeroing CS can approach this gain by lowering the DSP energy. When comparing to the "no compression" case (i.e., to transmitting the raw data), interesting results can be observed.

- For both NB and HBC (low energy per bit), the DSP power dominates the transmission, making data compression not convenient. Obviously, this is by all means not true in general, but it is found to hold for this peculiar general-purpose architecture. However, zeroing CS shows its effectiveness especially for $\texttt{OT} = 2$.

Table 8.3 Energy per transmitted/stored bit assuming different transmission (TX) and storage (NVM) technologies

Type	Technology	Energy [nJ/bit]	Reference
TX	Bluetooth Low Energy (BLE)	1	[26]
	Narrow Band (NB)	0.1	[38]
	Human Body Channel Communication (HBC)	0.24	[2]
	Near Field Communication (NFC)	10	[25]
NVM	Resistive RAM (ReRAM)	2	[9]
	Spin-torque-transfer magneto-resistive RAM (STT-MRAM)	0.1	[19]
	Flash memory (FLASH)	0.01	[43]
	Conductive Bridging RAM (CBRAM)	0.001	[14]

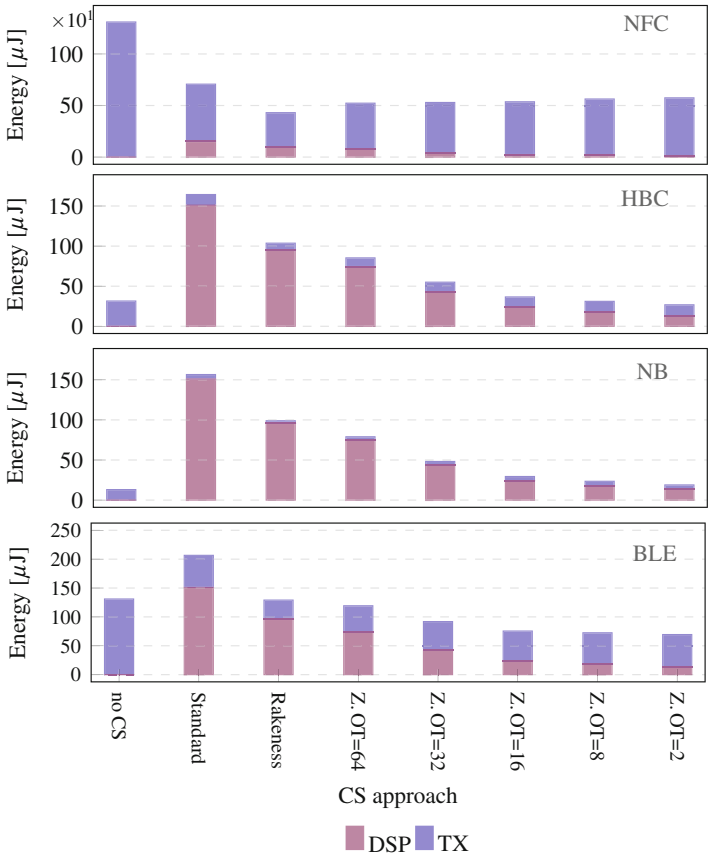


Fig. 8.15 Energy per window, considering different TX technologies and different CS approaches which include the case where any compression algorithm is used. *From top to bottom*: Near Field Communication, Human Body Channel, Narrow Band, and Bluetooth Low Energy

- Over the considered transmission technologies, BLE is an intermediate case. Here the introduced random zeroing shows the best energy-efficiency performance.
- For high transmission cost, as for NFC, energy in transmission dominates the DSP contribution. As a consequence, the rakeness-based CS, with the highest compression ratio, achieves the best energy efficiency. Note that the zeroing approach with $\mathtt{OT} = 64$ is relatively close to the minimum reached by rakeness-based CS.

Clearly, the adoption of more sophisticated zeroing procedures (amply discussed in Chap. 4) is a promising approach in the reduction of the energy cost also for cases where DSP cost is far from negligible.

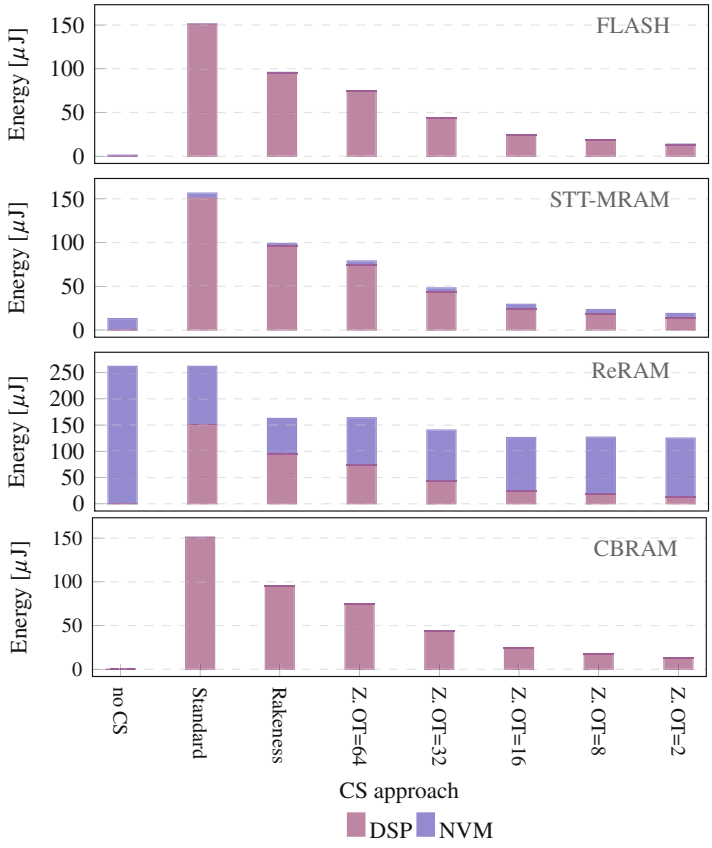


Fig. 8.16 Energy per window for different NVM technologies and different CS approaches which include the case where any compression algorithm is used. *From top to bottom*: Flash technology, Conductive Bridge RAM, Spin-transfer torque magnetic RAM, and Resistive RAM

The second analyzed scenario assumes that the bio-sensing node locally stores the compressed data. As before, a single plot for each considered technology reports results for both DSP and storage energy consumptions. Such results are in Fig. 8.16. Firstly, STT-MRAM, CBRAM, and FLASH configurations are characterized by a negligible storage energy cost w.r.t. the DSP, which makes the case of storing uncompressed data (no CS in plots) more energy efficient. However, for the case of ReRAM technology conclusions changes so that both rakeness-based CS and zeroing CS guarantee a non-negligible improvement.

For this scenario another figure of merit counts the impact of all analyzed CS approach, it is the memory footprint. Due to the fact that data are locally stored directly in the bio-sensing node, also the storage footprint is important due to area limitations in the sensor node. Results referring to this figure of merit, maximum

Table 8.4 Monitoring time as a function of different NVM storage capacities (technology independent)

CS approach		32KB (*s*)	1MB (*m*)	128MB (*h*)	1GB (*d*)
Rakeness-based CS		11.04	5.89	12.56	4.19
Zeroing CS	OT = 64	8.26	4.41	9.40	3.13
	OT = 32	7.61	4.06	8.66	2.89
	OT = 16	7.23	3.86	8.23	2.74
	OT = 8	6.79	3.62	7.72	2.57
	OT = 2	6.60	3.52	7.51	2.50
Standard CS		6.66	3.55	7.58	2.53
No CS		2.82	1.51	3.21	1.07

monitoring time for different embedded storage size, are shown in Table 8.4 for different memory sizes.

As an example, for 128 MB embedded NVM, rakeness-based CS allows to store more than 12 h of ECG track, 5 h more than standard CS. Without any data compression only 3 h can be stored. As expected, the zeroing approach is in between monitoring times of standard and rakeness-based CS.

8.4 Design of Low-Complexity CS by Mangia et al., 2017

The design flow described by Mangia et al. in [30] merges sensing matrix design based on rakeness with the zeroing approach presented in the previous section. The discussed approach is also discussed in Chap. 4 where, as in [30], such a merge was translated in the solution of two different optimization problems, (4.6) and (4.7), that give us the correlation matrices to be imposed to either ternary ($\boldsymbol{A}_{j,k} \in \{-1, 0, 1\}$) or binary ($\boldsymbol{A}_{j,k} \in \{0, 1\}$) sensing matrix rows. Both Chap. 4 and [30] name these optimization problems, respectively, TRLT and BRLT and discussions on their solutions are amply reported. The entire design flow can also explored by a set of freely available[4] MATLAB© functions along with some demo examples.

To recall the notation and parameters defined in Chap. 4, the most obvious design parameter is the compression ratio ($\mathtt{CR} = n/m$); the sparsity ratio SR of $\boldsymbol{A}$ and the puncturing ratio PR are also fundamental as they count the ratio in the amount of elementary operations required to compute $\boldsymbol{y}$ and the ratio in the amount on input samples involved in the same computation, respectively. These are estimated as

$$\mathtt{SR} = \frac{n\,m}{W} \qquad \mathtt{PR} = \frac{n}{N}$$

[4] http://cs.signalprocessing.it/download.html.

where W counts the total amount of needed operations and N counts the actual number of input samples used to compress the data.

Two other parameters drive this section, they are the input throttling IT and the output throttling OT which measure the computational burden of each iteration when the matrix multiplication $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ is either horizontally or vertically throttled. All these parameters allow us to explore different options as far as the computation of projections is concerned. We present these results in a real-world scenario, along with a comparison with other approaches proposed in the literature so far.

8.4.1 Low-Complexity Sensor Node for ECGs Acquisition

The analysis discussed hereafter refers to the use of a synthetic generator [33] that provides noiseless ECG signals, to which additive white Gaussian noise is added to achieve an ISNR $= 40$ dB. The parameters of the generator are randomly drawn in the same ranges used in [29] to obtain a profile population from which the input signal statistics $\mathscr{X}$ can be estimated. Tracks are sampled at 256 Hz so that one-second windows correspond to $n = 256$. The resulting vectors $\boldsymbol{x}$ are analyzed by an overcomplete dictionary, that is a Symmlet-6 UDWT with 4 sub-bands in order to provide a decoding algorithm similar to one used in [31] (as it was also presented in Sect. 8.1.2.3).

Figure 8.17 reports the reconstruction performance for ternary-valued random matrices $\boldsymbol{A}$. ARSNR is plotted against CR for different configurations as far as throttling and puncturing are concerned for standard CS (i.i.d. ternary-valued random sequences) and for the rakeness-based CS (ternary-valued random sequences with a given correlation matrix $\mathscr{A}$ imposed by the solution of TRLT). In the same plot, as references, black solid track identifies performance attainable with antipodal, rakeness-based, projections and a black dashed track corresponds to purely random antipodal projections.

Beyond that, the relative position of solid (for rakeness-based projections) and dashed (for purely random projections) lines show the effect of statistical adaptation when different saving policies are adopted. Specifically, we report the results for IT $= 4$ (i.e., each row of $\boldsymbol{A}$ has 4 nonzero entries so that SR $= 64$) and for PR $= 1.1$ (it means that 233 input samples over 256 are involved in the computation, or equivalently, $\boldsymbol{A}$ has 23 null columns).

A different point of view is given by the results shown in Table 8.5, which reports the savings for ECGs reconstructions with a fixed ARSNR. Here, such a quality threshold is set to 34 dB and, the savings are arranged so that each row corresponds to the parameter controlling a stage in a signal chain: PR accounts for how many samples are acquired, IT for the complexity in data compression, and CR for the measurement dispatch.

The first two columns report the savings for the two reference cases with full sensing matrices, i.e., the entries are never zeroed, for both rakeness-based projections and purely random projections. Then, column groups correspond to

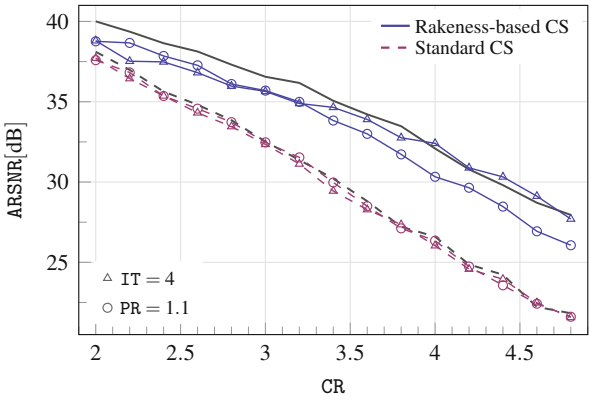


Fig. 8.17 ARSNR of synthetic ECG signals as a function of CR for different choices of projections and saving policies

Table 8.5 The savings achieved in the sampling, projection, and transfer stages when different strategies are employed to achieve a target ARSNR = 34 dB for synthetic ECGs

	Antipodal		Ternary			
	Stand. CS	Rak.-base CS	Rak.-base CS and Punc.		Rak.-base CS and Throt.	
PR	1	1	1.35	1.56	1	1
IT	256	256	190	164	4	8
CR	2.75	3.56	3.34	2.11	3.62	3.7

applying puncturing or throttling. Overall, empirical evidence shows that, even for this more realistic class of signals, target reconstruction quality can be achieved with noteworthy compression and saving in terms of projection computation.

As a further step towards more realistic scenarios it was considered real ECG tracks taken from the *MIT-BIH Arrhythmia Database* and from the *MIT-BIH noise stress database* [15, 34, 35]. The reason to target highly non-typical waveform is to assess the robustness of adaptation that is implicit in rakeness-based methods.

The statistical characterization we use at design-time is the one derived from the synthetic, noiseless, and artifact-free profiles used in the previous examples avoiding any bias. The acquisition is sized according to Table 8.5 with the aim of maximizing CR, i.e., as in the last column of that table. Hence, one-second windows of $n = 256$ samples are projected using a $^{n}/_{\mathrm{CR}} \times n = 69 \times 256$ ternary matrix $\boldsymbol{A}$. Such a matrix is drawn at random according to a second-order statistic resulting from the solution of a TRLT with $\mathrm{SR} = {}^{1}/_{8}$ and by selecting throttled rows that have only $\mathrm{IT} = 8$ nonzero entries. In Figs. 8.18 and 8.19 we report 10 seconds of an ECG signal with arrhythmia (Fig. 8.18) and 10 second of an ECG signal with arrhythmia and affected by motion artifacts (Fig. 8.19). In the entire figure the true waveform are rendered with dashed lines and coincide almost perfectly with the reconstructed waveform rendered as solid lines. From the quantitative point of view, notwithstanding a compression ratio of 3.7, the RSNR are 26 dB for the first case and 44 dB for the second case, that are achieved with a computational effort equivalent to only 8 sums/subtractions per measurement. Such results confirm in a real scenario the noteworthy performance of this approach.

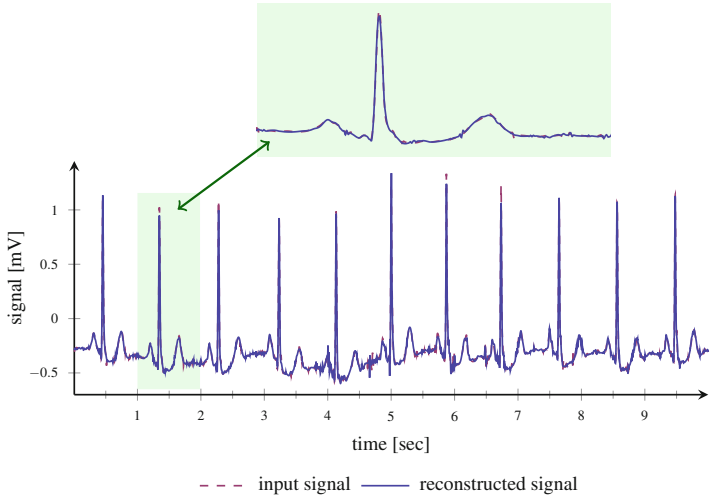


Fig. 8.18 Reconstruction of real-world ECG tracks by means of a ternary sparse matrix generated by rakeness-based design. The reported track is an ECG exhibiting arrhythmia

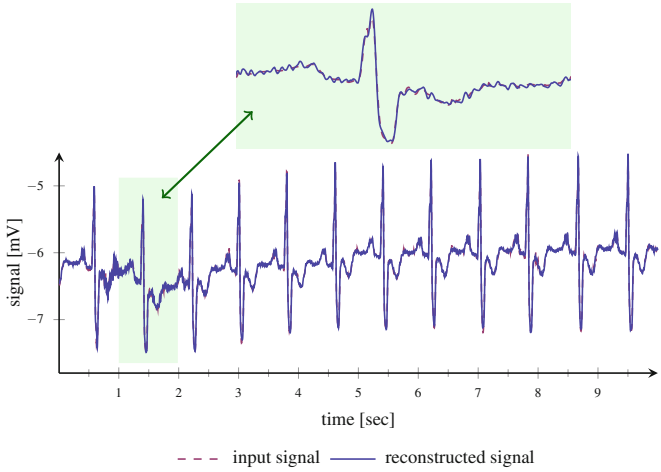


Fig. 8.19 Reconstruction of real-world ECG tracks by means of a ternary sparse matrix generated by rakeness-based design. The reported track is an ECG exhibiting arrhythmia whose acquisition is affected by motion artifacts

8.4.2 Comparison with Other Methods

In [47] the authors propose both a method to construct binary matrices $\boldsymbol{A}$ with optimized features and a decoding mechanism properly tuned for ECG reconstruction. The matrix $\boldsymbol{A}$ is built by placing a certain number d of nonzeros in each column so that the mutual coherence of the columns is kept as low as possible resulting in Minimal Mutual Coherence (MMC). Referring to the aforementioned design parameters, this means that the sensing mechanism is followed by an output throttling such that $\texttt{OT} = d$. To guide the reader, let us recall the definition of *mutual coherence* as already presented in (1.10), i.e.,

$$\mu(\boldsymbol{B}) = \max_{j \neq l} \frac{\left| \boldsymbol{B}_{\cdot,j}^{\top} \boldsymbol{B}_{\cdot,l} \right|}{\left\| \boldsymbol{B}_{\cdot,j} \right\|_2 \left\| \boldsymbol{B}_{\cdot,l} \right\|_2},$$

where $\boldsymbol{B} = \boldsymbol{A}\,\boldsymbol{D}$ and $\boldsymbol{D}$ is the basis (or dictionary) in which the input signal is sparse. In particular the main contribution regarding the encoder side is an algorithm that constructs binary matrices $\boldsymbol{A}$ with a proper OT that minimizes the mutual coherence for an assigned matrix $\boldsymbol{D}$.

At the decoder side, the reconstruction algorithm improves the general approach based on the sparsity criterion by exploiting a priori knowledge on the decay of the coefficients representing typical ECG signals when expressed on a Daubechies-6 DWT, so that standard ℓ_1 minimization is adapted to the ECG signal ensemble. The authors dubbed the resulting algorithm as Weighted ℓ_1 Minimization (WLM).

There is a main difference between this approach and the rakeness-based design flow; Zhang et al. propose to optimize a more general property of $\boldsymbol{A}$ than rakeness, as MMC does not account for the second-order statistics of the input signal; this approach of Zhang is also paired with an adapted decoder. On the contrary, rakeness-based CS is only limited to adapting the sensing procedure to the acquired signals, without any hint on how the signal should be decoded; in this way, the decoder does not necessarily require prior information on $\mathscr{X}$, hence avoiding the potential, non-negligible communication overhead of sharing $\mathscr{X}$.

Nevertheless, a performance comparison is possible and referring to the setting of [47, Section III] real ECG tracks were considered. In particular, the simulated setting uses the *MIT-BIH arrhythmia database* [34] where for the approach proposed by Zhang et al. input signals were acquired by means of binary matrices with $n = 512$ and different $(m, \texttt{OT})$ pairs, from $m = 96, OT = 4$ to $m = 256, \texttt{OT} = 12$. For each configuration,[5] the MMC matrices were constructed by means of [47, Algorithm 1]; on the other hand, binary rakeness-based matrices were built with the same compression ratios and with SR values that guaranteed the same amount on nonzero

[5] Although not explicitly reported in [47], to be effective a shuffling column of the highly structured generated matrix is needed. Note that this procedure keeps the low value of mutual coherence.

Table 8.6 Performance comparison between the rakeness-based approach and [47]

m	OT	CR	Average PRD		Rakeness-based is better
			[47]	[30]	
96	4	5.33	13.3%	10.7%	85%
128	5	4	8.85%	7.42%	80%
160	6	3.2	6.37%	5.81%	75%
256	8	2	3.40%	3.01%	83%
256	12	2	3.44%	3.09%	81%

entries. As additional requirements for the rakeness-based approach, the nonzero entries are equally divided into each row to ensure a constant IT.

The following comparisons are made encoding randomly selected windows from the first tracks of each pair in the database with a MMC matrix and with a rakeness-based matrix, and decoding the two resulting measurement vectors with WLM. As in [47], performance is measured by percentage root-mean-square difference (PRD) defined as follows:

$$\mathrm{PRD} = 100 \times \|\boldsymbol{x} - \hat{\boldsymbol{x}}\| / \|\boldsymbol{x}\|$$

where $\hat{\boldsymbol{x}}$ is the reconstructed signal which means that its values must be reduced as possible. Table 8.6 shows the result of such a comparison when 1000 trials are made for each configuration. The average PRD is reported for both encoding strategies along with the percentage of cases in which rakeness-based sensing outperforms MMC sensing. It is evident that, though performance tends to saturate to the same level, rakeness-based design is always convenient w.r.t. minimum coherence design.

Authors in [30] propose a second comparison with an interesting application discussed in [48]. In this work, the authors do not advocate any specialized encoding procedure, but mainly focus on a reconstruction algorithm dubbed Block-Sparse Bayesian Learning (BSBL), which is applied to the difficult problem of acquiring a signal coming from ECG sensor in which a mother's signal and that of her fetus superimpose, so that the latter can be retrieved by Independent Component Analysis (ICA).

The comparison proposed in [30] addresses the setting in [48, Section III.B], that is the most challenging one covered in [48]. Although the authors of [48] do not propose a new encoding strategy, their implementation is based on binary sensing matrices, which makes for an interesting comparison with the rakeness-based approach discussed in this section.

In the mentioned setting, the ground truth is the fetal ECG extracted by ICA directly from the non-compressed track. Also here, the reference case is based on matrices $\boldsymbol{A}$ with OT nonzeros per column. w.r.t. [47], the positioning of the nonzero entries is drawn at random. Rakeness-based matrices are still horizontally unrolled with a constant number of nonzero entries per row and a total number of nonzeros equal to the reference case. Signal windows of length $n = 512$ are then encoded by means of either $m = 256$ with $\mathrm{OT} = 12$, or $m = 205$ with $\mathrm{OT} = 10$ (higher

Table 8.7 Performance comparison between the rakeness-based approach and [48]

m	OT	CR	Average PCC [48]	Average PCC [30]	Rakeness-based is better
256	12	2	0.876	0.936	96%
205	10	2.5	0.793	0.858	97%

compression ratios do not allow a high quality recovery of the fetal track). The resulting measurement vectors are decoded by means of BSBL, and ICA is applied on the resulting tracks to find the fetal ECG.

In agreement with [48], the approaches' performance is quantified by the Pearson's Correlation Coefficient (PCC) between the ground truth and the extracted fetal ECG. For 100 trials, Table 8.7 shows the result of such a comparison. The average PCC is reported for the two encoding strategies, along with the percentage of cases in which rakeness-based sensing outperforms random sensing.

8.5 Implantable Neural Recording System by Zhang et al., 2014

The application considered in this section focuses on an area and power efficient multi-electrode arrays (MEA) to record neural signals. In particular, Zhang et al. proposed in [46] a signal dependent CS approach outperforming previously presented works in terms of compression rate and reconstruction quality. Additionally, the authors discussed a hardware implementation occupying an area of about $200 \times 300\,\mu$m per recording channel, with a power consumption of $0.27\,\mu$W at the operating frequency of 20 KHz. The considered signals are neural action potentials, commonly referred to as spikes, with a bandwidth up to 10 KHz and amplitude ranging from $50\,\mu$V to $500\,\mu$V. The main characteristic of the systems developed to acquire this class of signals is the large number of channels that simultaneously record the neuronal activity in a certain brain area by a MEA containing up to hundreds of electrodes. So many electrodes generate a large amount of data that requires, in a wireless transmission scenario, a power in the order of tens of mW. This motivates the effort of proposing CS as a compression algorithm able to reduce the transmitted data without a drastic increase in area and energy cost for data compression, in a way similar to other approaches focusing on wavelet transform-based compression methods [23, 37].

8.5.1 Signal Dependent CS

Requirements of this scenario are a high compression ratio and a compression algorithm as simple as possible, in order to not increase area and power consumption.

Despite the fact that CS seems a very good candidate for achieving this, to the best of our knowledge this is the only work in the literature proposing, for this class of signal, a CS approach capable of achieving a compression ratio in the same order of magnitude w.r.t. methods based on wavelet transform. This is obtained by authors of [46] with a two-step signal dependent approach. First, a training procedure identifies a set of atom functions that, rearranged in a signal dictionary, are able to guarantee a good sparse representation of the signal. Then, the signal is reconstructed by using both the computed signal dependent dictionary and a standard wavelet transform basis.

To recall the contents of Sect. 1.2.2, we say that a signal is sparse if a dictionary $\boldsymbol{D} \in \mathbb{R}^{n \times d}$ exists, with $d > n$, such that $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$, with $\boldsymbol{\xi}$ a vector with only a few non-null entries. One can say that $\boldsymbol{x}$ is κ-sparse over $\boldsymbol{D}$ when $\boldsymbol{\xi}$ has κ non-null entries with $\kappa \ll n$. As known, the achieved compression ratio is strongly related to κ value. This is the reason why authors investigate a *Dictionary Learning* technique trained over the same signal to be acquired in order to reduce as much as possible the amount of columns in $\boldsymbol{D}$ required to represent a generic window of a spike signal. The procedure starts by collecting an initial set of raw data split into n-length windows and then the K-SVD algorithm described in [1] is executed to determine the $\boldsymbol{D}$ columns that minimize the ℓ_2 norm between raw data and sparse representation for a fixed value of κ, i.e., the K-SVD method in [1] is used to solve the optimization problem

$$\begin{aligned} &\underset{\boldsymbol{D},\{\boldsymbol{\xi}^{(l)}\}_{l=0}^{L-1}}{\arg\min} \ \sum_{l=0}^{L-1} \left\| \boldsymbol{x}^{(l)} - \boldsymbol{D}\boldsymbol{\xi}^{(l)} \right\|_2^2 \\ &\text{s.t.} \ \left\| \boldsymbol{\xi}^{(l)} \right\|_0 \leq \kappa \ \text{with} \ 0 \leq l \leq L-1 \end{aligned}$$

where the $\{\boldsymbol{x}^{(0)}, \ldots, \boldsymbol{x}^{(L-1)}\}$ are the raw data used for training and $\{\boldsymbol{\xi}^{(0)}, \ldots, \boldsymbol{\xi}^{(L-1)}\}$ the associated κ-sparse representation.

The signal recovery framework is based on the assumption that each instance $\boldsymbol{x}$ is composed by two main contributions, one containing the mean shape of the spike signal, namely $\boldsymbol{x}_c$, and another vector $\boldsymbol{x}_f$ representing details characterizing the waveform, i.e.,

$$\boldsymbol{x} = \boldsymbol{x}_c + \boldsymbol{x}_f.$$

A visual representation of both contributions can be found in Fig. 8.20, showing a collection of recorded waveforms along with their average waveform. Accordingly to the signal representation proposed by Zhang et al., $\boldsymbol{x}_c$ can be represented as a 1-sparse signal $\boldsymbol{\xi}_c$ over the trained dictionary $\boldsymbol{D}$, while the residual part $\boldsymbol{x}_f$ has a wavelike nature and can be represented as a vector $\boldsymbol{\xi}_f$ sparse in a wavelet basis

$$\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}_c + \boldsymbol{W}^{-1}\boldsymbol{\xi}_f$$

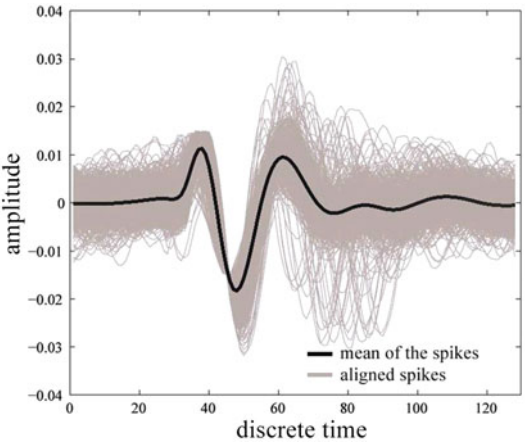


Fig. 8.20 Spikes segments (*gray lines*) from one neuron and corresponding mean (*black line*) of these spike frames (adapted from [46])

where $\boldsymbol{W}^{-1}$ is the basis corresponding to an inverse DWT. Due to the linearity of the sensing operator $\boldsymbol{A}$, this notation can be used also to identify the contributions of these two vectors to the measurement vector

$$\boldsymbol{y} = \boldsymbol{y}_c + \boldsymbol{y}_f = \boldsymbol{AD}\boldsymbol{\xi}_c + \boldsymbol{AW}^{-1}\boldsymbol{\xi}_f$$

Indeed, the authors of [46] suggested to use a sensing matrix $\boldsymbol{A} \sim \text{RAE}(\boldsymbol{I})$. As a consequence of this assumption, the recovering of $\boldsymbol{x}$ is achieved in two steps. Firstly, $\boldsymbol{\xi}_c$ is estimated by looking among signals that are 1-sparse in the trained dictionary. Referring to this signal with $\hat{\boldsymbol{\xi}}_c$, its contribution $AD\hat{\boldsymbol{\xi}}_c$ to the measurement vector is computed. After that, the measurement residual vector $\boldsymbol{y}_{\text{res}} = \boldsymbol{y} - AD\hat{\boldsymbol{\xi}}_c$ is computed, and used to estimate a vector $\hat{\boldsymbol{\xi}}_f$ sparse on the Wavelet basis and containing signal details. Authors name this recovery approach as Signal Dependent Neural Compressed Sensing (SDNCS). Its flow is summarized in Table 8.8.

As already anticipated, there is a twofold difference w.r.t. a standard CS recovery procedure. Signal recovery is split into the identification of the two vectors $\hat{\boldsymbol{\xi}}_c$ and $\hat{\boldsymbol{\xi}}_f$, and proper optimization problems are addressed for both recovery stages. In the recovery of $\boldsymbol{\xi}_c$ authors impose a fixed sparsity level, while the recovery of $\boldsymbol{\xi}_f$ is a standard ℓ_1 minimization with a regularization parameter λ that balances the trade-off between data fidelity and sparsity.

To further improve reconstruction performance, the authors propose to exploit the implementation of a simple on-chip spike detection circuit implementing the algorithm discussed in [3]. For a spike occurring at temporal location, instead of searching within the entire dictionary $\boldsymbol{D}$, the first recovery stage focuses through a much smaller set of atoms belonging to a sub-dictionary. This additional step increases the probability to converge on the global minimal solution instead of a local one, with a consequent increase in noise rejection. We refer to this approach

Table 8.8 Code sketch for the Signal Dependent Neural Compressed Sensing proposed in [46]

Require: $\boldsymbol{y}$ vector of measurements
Require: κ sparsity level for $\boldsymbol{x}_v$
Require: $\boldsymbol{A}$ sensing matrix
Require: $\boldsymbol{D}$ trained dictionary
Require: $\boldsymbol{W}^{-1}$ inverse wavelet transform
Compute estimator $\hat{\boldsymbol{\xi}}_c$ by solving: ▷ Trained dictionary related component
$\hat{\boldsymbol{\xi}}_c \leftarrow \arg\min_{\boldsymbol{\xi}_c} \|\boldsymbol{y} - \boldsymbol{AD}\boldsymbol{\xi}_c\|_2$ s.t. $\|\boldsymbol{\xi}_c\|_0 = \kappa$
Compute $\boldsymbol{y}_{\text{res}}$ by: $\boldsymbol{y}_{\text{res}} \leftarrow \boldsymbol{y} - \boldsymbol{AD}\hat{\boldsymbol{\xi}}_c$
Compute estimator $\hat{\boldsymbol{\xi}}_f$ by solving: ▷ Signal details related components
$\hat{\boldsymbol{\xi}}_f \leftarrow \arg\min_{\boldsymbol{\xi}_f} \left\|\boldsymbol{y}_{\text{res}} - \boldsymbol{W}^{-1}\boldsymbol{\xi}_f\right\|_2^2$ s.t. $\left\|\boldsymbol{\xi}_f\right\|_1 \leq \lambda$
return $\hat{\boldsymbol{x}} = \boldsymbol{D}\hat{\boldsymbol{\xi}}_c + \boldsymbol{W}^{-1}\hat{\boldsymbol{\xi}}_f$ ▷ Final estimation

as Signal Dependent Neural Compressed Sensing approach with Prior recovery information (SDNCS-P).

To highlight performance of both SDNCS and SDNCS-P w.r.t. other approaches, sample signals from University of Leicester neural signal database are used for simulations.[6] In particular, three different pieces of neural signals referred to as Easy1, Easy2, and Hard1 have been considered. The naming of the signals, which is consistent with that used in the dataset, refers to the effort required by spike classification. Easy1 and Easy2 contain spike shapes having large temporal variance, while spike shapes in the Hard1 set are very close in the temporal domain. All these data contain spikes from 3 different neurons sampled at 24 KHz, each segment containing only one spike. From the 2046 signal frames extracted for each task, 20% of them are used to train the signal dependent dictionaries (one for each task), while the remaining 80% are used for testing.

The quality of the recovered frames by SDNCS and SDNCS-P are compared with other compression methods representing the state of the art is spike detection. In particular, three approaches are considered: the spike detection windowing, where m samples are kept around a threshold crossing location while other samples of the signal are discarded; the wavelet transformed and thresholding (DWT), where signals are transformed into wavelet domain and only the m most significant coefficients are retained; and Compressed Sensing based on a single recovery step using wavelet basis as sparsity matrix (CS-DWT). In addition to the commonly used `ARSNR`, Classification Successful Rate is used as metric for measuring the quality of reconstructed signal. This is defined as the percentage of total number of spikes correctly classified using labels contained in the database as ground truth. In particular, authors refer to two different types of spike classifiers. The first one

[6]The dataset is on-line available: http://www2.le.ac.uk/departments/engineering/research/bioengineering/neuroengineering-lab.

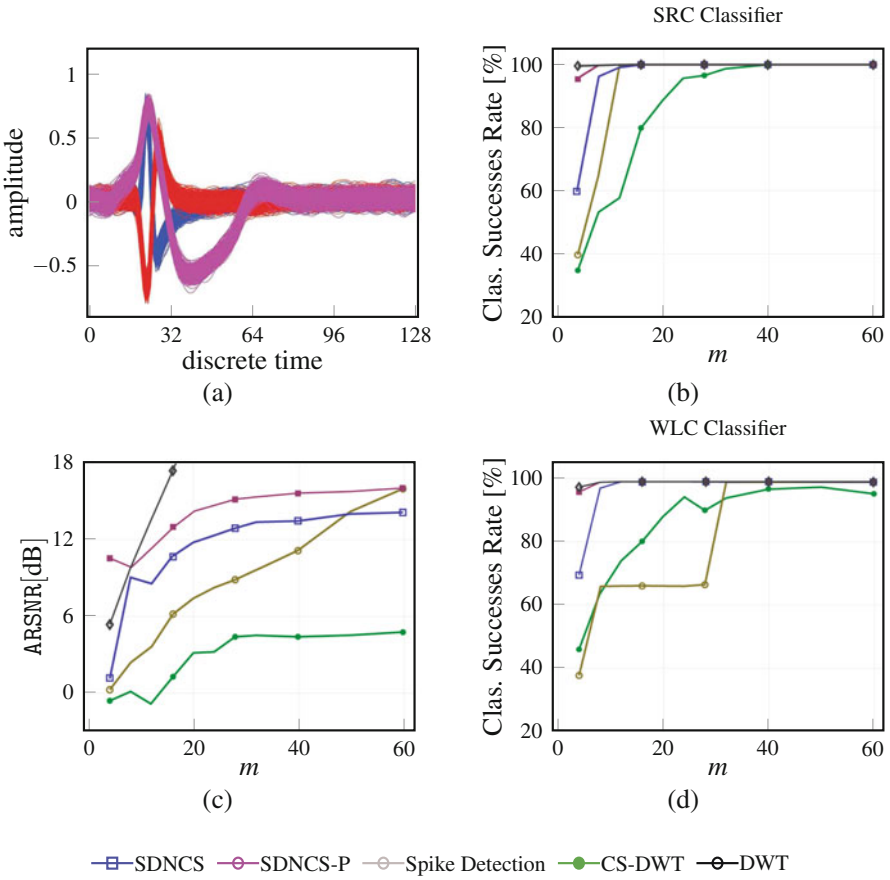


Fig. 8.21 Comparison of spike detection, DWT, CS-DWT, SDNC, and SDNC-P for the dataset named Easy1. Temporal view of spikes (**a**); the ARSNR as a function of m (**c**); Classifications Successes Rate with SRC classifier (**b**), and WLC classifier (**d**)

is the wavelet-based classifier (WLC) discussed in [41] and the second the Sparse Representation Classifier (SRC) proposed in [45].

The temporal shapes of the spikes from the three databases, along with results for the aforementioned figures of merit are shown in Figs. 8.21, 8.22, and 8.23, respectively. The SDNCS and SDNCS-P algorithms outperform both CS-DWT and Spike Detection across all merit figures while the results of both SDNCS and SDNCS-P are comparable with that obtained by DWT in terms of Classification Successes rate. However, as expected, DWT outperforms the proposed methods in terms of ARSNR. Although spikes signals are moderately sparse in the Wavelet domain, the DWT method considers the higher m wavelet coefficients only. Using them for the reconstruction of the neural signal, there is a good match with raw data for $m \geq 30$. Although the authors propose a comparison between DWT and CS-

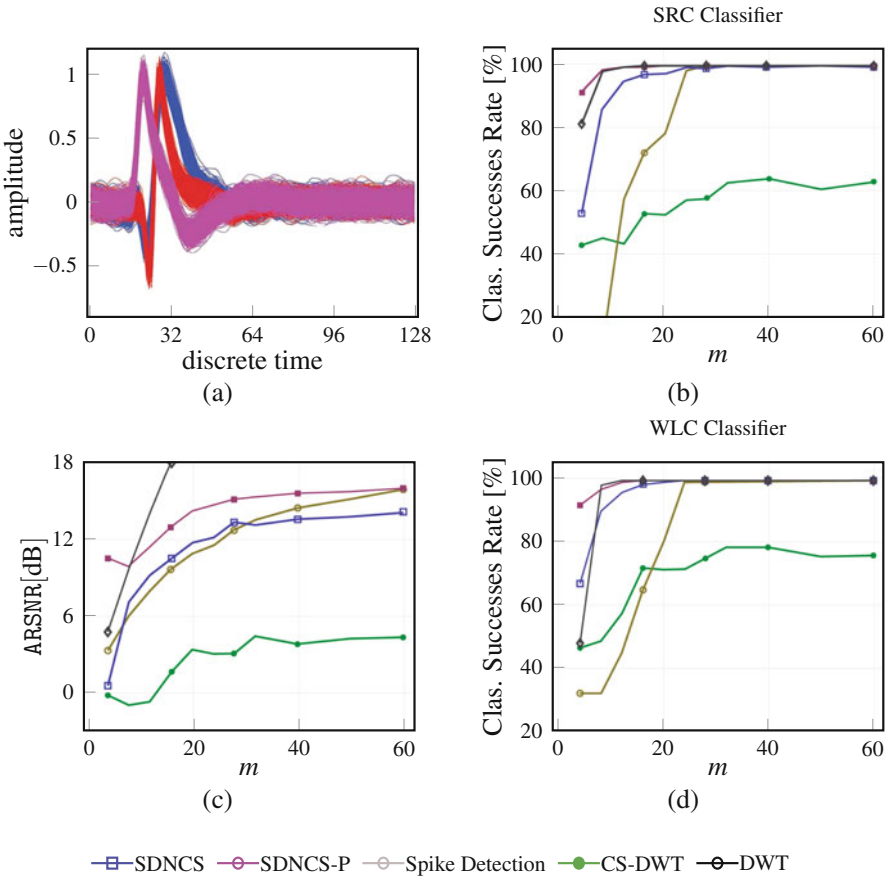


Fig. 8.22 Comparison of spike detection, DWT, CS-DWT, SDNC, and SDNC-P for the dataset named Easy2. Temporal view of spikes (**a**); the ARSNR as a function of m (**c**); Classifications Successes Rate with SRC classifier (**b**), and WLC classifier (**d**)

like methods for a fixed m, it is our opinion such a comparison is not fair. In fact, we have to take into account that DWT must transmit both the m higher coefficient of the wavelet atoms and their indexes. In other words, the amount of information required by this approach is much higher than only m coefficients as in the other approaches.

These considerations reinforce the conclusion that, for this particular class of signals, the proposed decoding approaches outperform standard CS and achieve similar performance of DWT method, but it is more suitable for low-area and low-power implementation.

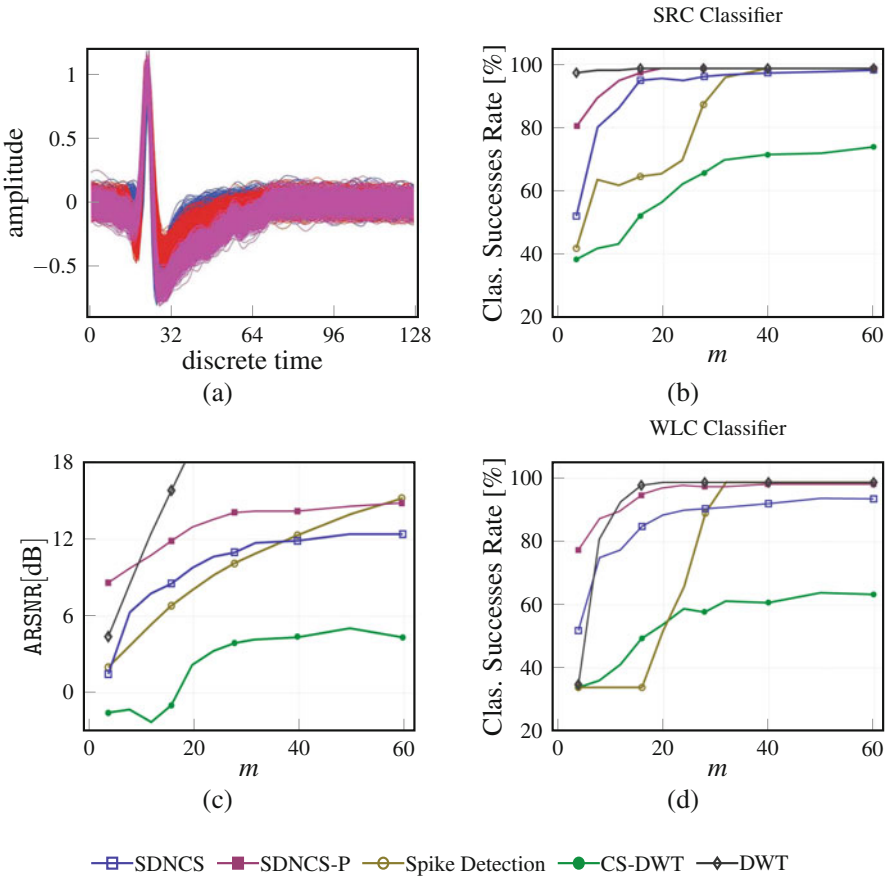


Fig. 8.23 Comparison of spike detection, DWT, CS-DWT, SDNC, and SDNC-P for the dataset named Hard1. Temporal view of spikes (**a**); the ARSNR as a function of m (**c**); Classifications Successes Rate with SRC classifier (**b**), and WLC classifier (**d**)

8.5.2 Hardware Implementation

In [46] authors also proposed a proof of concept device implementing the discussed CS approaches. A high-level block diagram of the system is shown in Fig. 8.24 along with the off-chip system performing first the dictionary training phase, and then the spike signals decoding and classification.

The signal processing chain includes first a conditioning block made of on-chip amplifiers and band pass filters in order to express captured phenomena with the appropriate bandwidth and amplitude. Then, an ADC digitizes the signal at its Nyquist rate generating digital words available either for a direct transmission (*training phase*) or for compression and transmission to the off-chip system, the compression phase. Collected raw data can also be used for optimal threshold

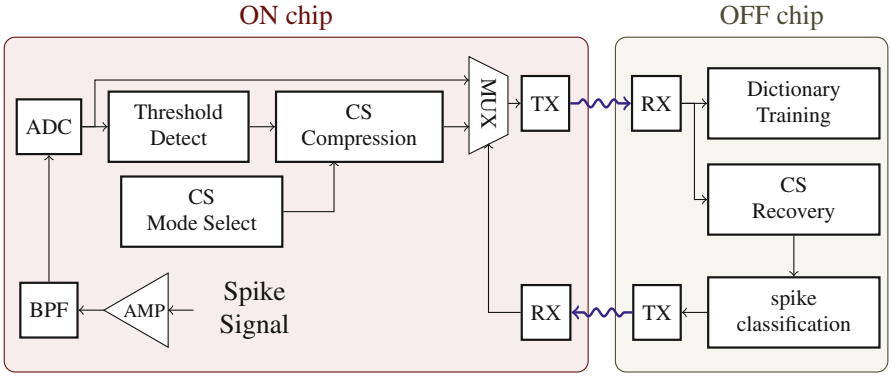


Fig. 8.24 Block diagram for implementation system proposed in [46]

configuration for the SDNCS-P. When the training phase is complete, the CS stage is activated so that the ADC output is mixed with random sensing matrix implemented using digital accumulators, one for each CS channel. This stage generates the compressed measurements that are ready for wireless transmission to the off-chip system. Note that, even if antipodal sensing matrices (i.e., with $\boldsymbol{A}_{i,k} \in \{-1, +1\}$) are used in the proposed simulation of the system, in the proof of concept design binary matrices (with $\boldsymbol{A}_{i,k} \in \{0, 1\}$) are implemented. This implementation is proposed to further reduce active power of the digital circuit. However, it is also highlighted in [46] that for high compression ratio values, antipodal sensing matrices outperform the adoption of binary sensing matrices. This paves the way for a trade-off that is strongly related to the transmission power cost. For the implementation of the sensing matrix generator the authors do not propose a specific implementation, but refer to both a random number generator, implemented by a Linear Feedback Shift Register, and a locally stored pre-generated sensing matrix.

Authors also detail power consumption measurements of the Compressed Sensing sub-system. This contribution is dominated by the active power of the accumulator circuits where such results refer to an implemented test structures of the CS Channel on the TSMC 180 nm with layout and microphotograph shown in Fig. 8.25. This device is designed with 100 CS channels, i.e., with m up to 100. The designed ADC is a 10 bit SAR ADC operating at 20 KHz. An additional cost must be taken into account for the SDNCS-P approach, it requires extra implementation of a 10 bit digital comparator before the CS mixing circuits to detect spikes occurrences. A buffer is also necessary to save data from a subwindows (in the proposed implementation the buffer size is such that 15 samples of the spike are saved prior to threshold crossing event). For a setting where 25 measurement are enough to reach a desired target quality, the total power consumption is 0.27 W at 20 KHz sampling frequency when the VDD is at 0.6 V. Note that this number is for the case where the Threshold Detect is not active. Regarding the area of the device, the 25 CS channels used as target occupy an area of $200 \times 300\,\mu$m, black blocks in Fig. 8.25.

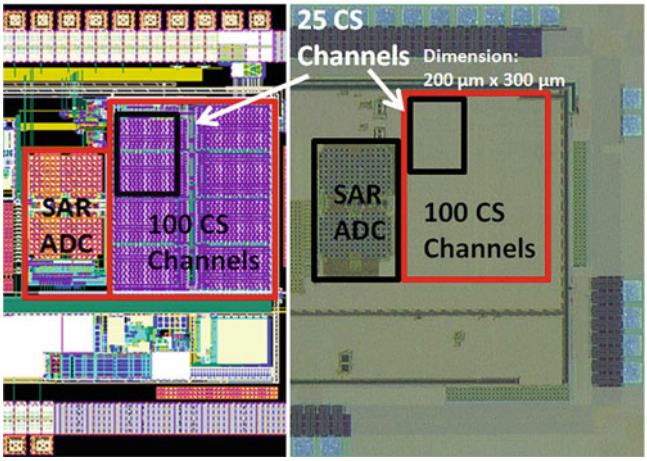


Fig. 8.25 Micro-graph and layout of the chip TSMC 0.18 µm process implementing CS compression. Picture is from [46]

References

1. M. Aharon, M. Elad, A. Bruckstein, rmK-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. **54**(11), 4311–4322 (2006)
2. J. Bae et al., A 0.24-nJ/b wireless body-area-network transceiver with scalable double-FSK modulation. IEEE J. Solid State Circuits **47**(1), 310–322 (2012)
3. R.G. Baraniuk et al., Model-based compressive sensing. IEEE Trans. Inf. Theory **56**(4), 1982–2001 (2010)
4. E. van den Berg, M.P. Friedlander, *SPGL1: A Solver for Large-Scale Sparse Reconstruction*. http://www.cs.ubc.ca/labs/scl/spgl1. June 2007
5. D. Bortolotti et al., An ultra-low power dual-mode ECG monitor for healthcare and wellness, in *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, Mar. 2015, pp. 1611–1616
6. R. Calderbank et al., Wavelet transforms that map integers to integers. Appl. Comput. Harmon. Anal. **5**(3), 332–369 (1998)
7. E.J. Candes, T. Tao, Near-optimal signal recovery from random projections: Universal encoding strategies? IEEE Trans. Inf. Theory **52**(12), 5406–5425 (2006)
8. E.J. Candes et al., Compressed sensing with coherent and redundant dictionaries. Appl. Comput. Harmon. Anal. **31**(1), 59–73 (2011)
9. M.-F. Chang et al., A 0.5V 4Mb logic-process compatible embedded resistive RAM (ReRAM) in 65nm CMOS using low-voltage current-mode sensing scheme with 45ns random read time, in *2012 IEEE International Solid-State Circuits Conference*, Feb. 2012, pp. 434–436
10. P.L. Combettes, J.-C. Pesquet, A Douglas–Rachford splitting approach to nonsmooth convex variational signal recovery. IEEE J. Sel. Top. Signal Process. **1**(4), 564–574 (2007)
11. T.M. Cover, J.A. Thomas, *Elements of Information Theory* (Wiley, New York, 2012)
12. J.E. Fowler, The redundant discrete wavelet transform and additive noise. IEEE Signal Process. Lett. **12**(9), 629–632 (2005)
13. M. Gautschi, D. Rossi, L. Benini, Customizing an open source processor to fit in an ultra-low power cluster with a shared L1 memory, in *Proceedings of the 24th Edition of the Great Lakes Symposium on VLSI*, GLSVLSI'14. ACM, Houston, Texas, USA, 2014, pp. 87–88

14. N. Gilbert et al., A 0.6V 8 pJ/write non-volatile CBRAM macro embedded in a body sensor node for ultra low energy applications, in *2013 Symposium on VLSI Circuits*, June 2013, pp. C204–C205
15. A.L. Goldberger et al., Physiobank, Physiotoolkit, and Physionet: components of a new research resource for complex physiologic signals. Circulation **101**(23), 215–220 (2000)
16. V.K. Goyal, Theoretical foundations of transform coding. IEEE Signal Process. Mag. **18**(5), 9–21 (2001)
17. V.K. Goyal, A.K. Fletcher, S. Rangan, Compressive sampling and lossy compression. IEEE Signal Process. Mag. **25**(2), 48–56 (2008)
18. R.M. Gray, Vector quantization. IEEE ASSP Mag. **1**(2), 4–29 (1984)
19. D. Halupka et al., Negative-resistance read and write schemes for STTMRAM in 0.13μm CMOS, in *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, IEEE, Feb. 2010, pp. 256–257
20. D.A. Huffman, A method for the construction of minimum-redundancy codes. Proc. IRE **40**(9), 1098–1101 (1952)
21. L. Jacques, D.K. Hammond, J.M. Fadili, Dequantizing compressed sensing: When oversampling and non-gaussian constraints combine. IEEE Trans. Inf. Theory **57**(1), 559–571 (2011)
22. L. Jacques, D.K. Hammond, J.M. Fadili, Stabilizing nonuniformly quantized compressed sensing with scalar companders. IEEE Trans. Inf. Theory **59**(12), 7969–7984 (2013)
23. A.M. Kamboh, A. Mason, K.G. Oweiss, Analysis of lifting and B-spline DWT implementations for implantable neuroprosthetics. J. Signal Process. Syst. **52**(3), 249–261 (2008)
24. K.A. Kotteri et al., A comparison of hardware implementations of the biorthogonal 9/7 DWT: convolution versus lifting. IEEE Trans. Circuits Syst. II Express Briefs **52**(5), 256–260 (2005)
25. W.L. Lien et al., A self-calibrating NFC SoC with a triple-mode reconfigurable PLL and a single-path PICC-PCD receiver in 0.11μm CMOS, in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, IEEE, Feb. 2014, pp. 158–159
26. Y.-H. Liu et al., A 1.9nJ/b 2.4GHz multistandard (Bluetooth Low Energy Zigbee, IEEE 802.15.6) transceiver for personal/body-area networks, in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb. 2013, pp. 446–447
27. Z. Lu, D. Youn Kim, W.A. Pearlman, Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm. IEEE Trans. Biomed. Eng. **47**(7), 849–856 (2000)
28. S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*. Access Online via Elsevier, 2008
29. M. Mangia, R. Rovatti, G. Setti, Rakeness in the design of analog-to-information conversion of sparse and localized signals. IEEE Trans. Circuits Syst. I Regul. Pap. **59**(5), 1001–1014 (2012)
30. M. Mangia et al., Rakeness-based design of low-complexity compressed sensing. IEEE Trans. Circuits Syst. I Regul. Pap. **64**(5), 1201–1213 (2017)
31. M. Mangia et al., Zeroing for HW-efficient compressed sensing architectures targeting data compression in wireless sensor networks. Microprocess. Microsyst. **48**, 69–79 (2017). Extended papers from the 2015 Nordic Circuits and Systems Conference, pp. 69–79
32. J. Max, Quantizing for minimum distortion. IRE Trans. Inf. Theory **6**(1), 7–12 (1960)
33. P.E. McSharry et al., A dynamical model for generating synthetic electrocardiogram signals. IEEE Trans. Biomed. Eng. **50**(3), 289–294 (2003)
34. G.B. Moody, R.G. Mark, The impact of the MIT-BIH arrhythmia database. IEEE Eng. Med. Biol. Mag. **20**(3), 45–50 (2001)
35. G.B. Moody, W.K. Muldrow, R.G. Mark, A noise stress test for arrhythmia detectors. In: *1984 Computers in Cardiology*, vol. 11, Sept. 1984, pp. 381–384
36. A. Moshtaghpour et al., Consistent basis pursuit for signal and matrix estimates in quantized compressed sensing. IEEE Signal Process. Lett. **23**(1), 25–29 (2016)
37. K.G. Oweiss et al., A scalable wavelet transform VLSI architecture for real-time signal processing in high-density intra-cortical implants. IEEE Trans. Circuits Syst. I Regul. Pap. **54**(6), 1266–1278 (2007)
38. G. Papotto et al., A 90nm CMOS 5Mb/s crystal-less RF transceiver for RF-powered WSN nodes, in *2012 IEEE International Solid-State Circuits Conference*, IEEE, Feb. 2012, pp. 452–454

39. W.A. Pearlman, A. Said, *Digital Signal Compression: Principles and Practice* (Cambridge University Press, Cambridge, 2011)
40. N. Perraudin et al., UNLocBoX: A matlab convex optimization toolbox using proximal splitting methods. *arXiv preprint arXiv:1402.0779* (2014)
41. R. Quian Quiroga, Z. Nadasdy, Y. Ben-Shaul, Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural Comput. **16**(8), 1661–1687 (2004)
42. I.W. Selesnick, M.A.T. Figueiredo, Signal restoration with overcomplete wavelet transforms: comparison of analysis and synthesis priors, in *SPIE Optical Engineering+ Applications*. International Society for Optics and Photonics, Aug. 2009, pp. 74460D–74460D
43. D. Shum et al., Highly reliable flash memory with self-aligned split-gate cell embedded into high performance 65nm CMOS for automotive amp; smartcard applications, in *2012 4th IEEE International Memory Workshop*, May 2012, pp. 1–4
44. J.Z. Sun, V.K. Goyal, Optimal quantization of random measurements in compressed sensing, in *2009 IEEE International Symposium on Information Theory*, IEEE, June 2009, pp. 6–10
45. J. Wright et al., Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 210–227 (2009)
46. J. Zhang et al., An efficient and compact compressed sensing microsystem for implantable neural recordings. IEEE Trans. Biomed. Circuits Syst. **8**(4), 485–496 (2014)
47. J. Zhang et al., Energy-efficient ECG compression on wireless biosensors via minimal coherence sensing and weighted ℓ_1 minimization reconstruction. IEEE J. Biomed. Health Informatics **19**(2), 520–528 (2015)
48. Z. Zhang et al., Compressed sensing for energy-efficient wireless telemonitoring of noninvasive fetal ECG via block sparse Bayesian learning. IEEE Trans. Biomed. Eng. **60**(2), 300–309 (2013)

Chapter 9
Security at the Analog-to-Information Interface Using Compressed Sensing

The rise of paradigms such as the Internet of Things, which envisions that the next generation of communication technologies will have to provide network access to billions of sensor nodes with minimal communication overhead, brings the matter of defending the privacy of data gathered and distributed by networked devices to the system designer's attention. In particular, since it is reasonable that large-scale networks will be comprised of a massive number of low-complexity nodes, even the resources spent for security purposes must be carefully tailored to the actual requirements of each application. Moreover, security becomes of even greater concern when the sensor nodes acquire sensitive biometric information or biomedical signals, e.g., for remote health monitoring or authentication purposes.

At the current state of the art, security is granted by dedicated encryption stages with varying levels of complexity. These stages protect the storage or transmission of information only after analog-to-digital conversion of the signal of interest, and generally correspond to a considerable expense of resources, especially in terms of power consumption and implementation costs. Thus, methods to balance this expense to the actual amount of security that is needed in each case are desirable.

To this end, we here investigate the possibility of using CS with i.i.d. RAE sensing matrices (i.e., comprised of sequences with i.i.d. antipodal symbols), whose properties have been largely illustrated in the previous chapters, as a method to introduce security directly into the acquisition process at the analog-to-information interface (e.g., using the schemes in Chaps. 6 and 7), or even jointly with digital signal compression (as it was discussed in Chap. 8).

As will be shown below, due to its linearity CS cannot be regarded in general as a means to provide *perfect secrecy* in the Shannon sense, since some general information about the signal will leak into the compressive measurements that are transmitted to the receiver. With this fact at hand, our treatment of security by CS is developed to show what privacy properties can still be gained by using on one hand an encoding algorithm as simple as a linear random projection, and on the other

M. Mangia et al., *Adapted Compressed Sensing for Effective Hardware Implementations*, DOI 10.1007/978-3-319-61373-4_9

hand a non-linear decoding algorithm on which the presence of noise sources and missing information have critical impact.

In Sect. 9.1 we will explore some basics and elementary definitions of secrecy. In Sect. 9.2, we discuss the types of conditions that have been recently proposed to address theoretically the security of CS, and evaluate some approaches to attack the compressive measurements by statistical cryptanalysis. Then, we proceed to discuss computational attacks to the simplest form of encoding by CS with RAE matrices in Sect. 9.3. Reassured of the security shown by CS against both types of attacks, we propose an encryption scheme based on CS in Sect. 9.4, which also enables multiple quality of access to the encrypted information by leveraging matrix perturbations. To further test the security of this multiclass encryption scheme, we also present computational attacks, as well as attacks based on signal recovery under matrix uncertainty.

Our overview provides some insight on the simplest and lowest-complexity form of cryptosystems that can be devised by means of CS, i.e., by projecting a signal using pseudorandomly generated universal encoding matrices in the sense of [14]. Indeed, this chapter focuses on obtaining fundamental results regarding the security provided solely by the latter matrices; many recent contributions should also be acknowledged in having gone further in the study of security by CS with more complex arrangements, including the use of further cryptographic layers [21, 59], security over finite fields [8], and CS with circulant matrices [6].

9.1 A Security Perspective on CS

9.1.1 CS as a Cryptosystem

In Chaps. 1 and 2 we have seen how some classes of random sensing matrices (hereafter dubbed *encoding matrices*) are *universal* for what concerns CS, i.e., that they are near-optimal in providing a non-adaptive linear dimensionality reduction method for any signal having a sparse representation on some basis $\boldsymbol{D}$. Due to this striking fact, the thought that such randomness could be used to provide, at least to some extent, a form of encryption has been anticipated since the foundations of CS [13, 14].

The first work to formally address a security perspective on CS is the one of Rachlin and Baron [49]. There, the authors look at CS with fundamental notions of classical information-theoretic secrecy [52]. Thus, they regard the source of information, Alice, as a transmitter who provides a *plaintext* $\boldsymbol{x}$ (the signal of interest) to an intended receiver. This receiver, Bob, is thus provided with the *ciphertext* $\boldsymbol{y}$ (the measurement vector) that is a secret, suitably transformed representation of the plaintext. He is therefore able to successfully recover $\boldsymbol{x}$ from $\boldsymbol{y}$ if he is also granted $\boldsymbol{A}$, or equivalently the *encryption key* or *shared secret* required to generate the encoding matrix at the receiver. This exchange is mediated by an *encryption algorithm* that, in our case, amounts to (*i*) generating in a secure fashion a stream

of pseudorandom bits containing the symbols of some encoding matrix $\boldsymbol{A}$, and (*ii*) applying $\boldsymbol{A}$ as $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, i.e., a linear transformation with a random encoding matrix, and transmitting the ciphertext via a standard communication channel. A *cryptosystem* is therefore the ensemble of operations required to *encrypt* the plaintext sent by Alice, and to *decrypt* it from the ciphertext that Bob received from the channel. In this perspective CS is properly regarded as a *private* or *symmetric key cryptosystem*, that is any means to provide security in which the encryption key is identical for both parties in a secure communication.

On the other hand, a malicious user, Eve, could intercept $\boldsymbol{y}$ on the channel and be interested in retrieving either $\boldsymbol{x}$, or even the encoding matrix $\boldsymbol{A}$ that contains symbols obtained from the encryption key; thus, an *attack* or *cryptanalysis* is any procedure capable of providing either the plaintext or the encryption key.

9.1.2 Preliminary Considerations

At first sight CS is similar to a linear block code, in a fashion analog to the McEliece and Niederreiter ciphers [41, 44]; however, two crucial differences can be highlighted. Firstly, the comparison of standard CS to other digital-to-digital means of encryption is not straightforward: while the basic theory of CS considers both the plaintext and ciphertext over the reals, even when quantized CS [27, 28] is considered, $\boldsymbol{x}$ is always regarded as a vector over the reals (i.e., $\mathbb{R}^n$) rather than integers (i.e., $\mathbb{Z}^n$), while the ciphertext $\boldsymbol{y}$ is instead comprised of suitably quantized, 1- to B-bit measurements. Secondly, the encoding matrix is not invertible: due to this, the successful decryption of a plaintext is actually depending on the output of a non-linear operation, i.e., the decoding algorithm. This is in turn influenced by the sparsity of the plaintext w.r.t. a suitable basis $\boldsymbol{D}$, and the perfect knowledge of $\boldsymbol{A}$ that is necessary in the recovery of $\boldsymbol{x}$ given $\boldsymbol{y}$. Due to these two basic facts, a successful means of decryption is any decoding algorithm that provides a sufficiently accurate recovery of $\boldsymbol{x}$ (in terms of `RSNR`) given the measurements $\boldsymbol{y}$ and $\boldsymbol{A}, \boldsymbol{D}$.

Since private-key cryptosystems operate by agreeing on a finite number of bits that form the above encryption key, the mn symbols that comprise $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ will be extracted from a pseudorandom sequence of bits that is obtained from the initial seed. In the case of CS, we can consider this seed as the encryption key itself, and the expanded pseudorandom binary sequence will eventually repeat depending on the number of bits spent into the encryption key. In the following we will assume that the period of the pseudorandom sequences generated by algorithmic expansion of the secret (e.g., by a *pseudorandom number generator*, PRNG) is sufficiently long as to guarantee that in a reasonable observation time the same $\boldsymbol{A}$ will never occur twice. This hypothesis is fundamental to ensure that $\boldsymbol{A}$ cannot be recovered from the knowledge of a sufficient number of plaintexts and ciphertexts, a simple observation first made by Drori [18], and is very close to the concept of *one-time pad* [57, Section 2.9]. Most of the literature on security by CS agrees on this fundamental requirement [4, 10, 11, 49] which, if violated, potentially allows the collection of

several plaintext–ciphertext pairs corresponding to the same encoding matrix, and with it the possibility of a successful cryptanalysis.

Another relevant effect is the dimension of the plaintext in our analysis. Since we will consider an input signal $\boldsymbol{x}$, it is sensible to consider two regimes or models w.r.t. its sample size:

- M_1: for finite n, we let $\boldsymbol{x} \in \mathbb{R}^n$ be a real-valued random vector. Its realizations are *finite-length* plaintexts denoted by the same letter $\boldsymbol{x}$, and are assumed to have finite energy $E_{\boldsymbol{x}} = \|\boldsymbol{x}\|_2^2$. We will let each $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$ with $\boldsymbol{D}$ an orthonormal basis and $\boldsymbol{\xi}$ being κ-sparse to comply with sparse signal recovery guarantees. $\boldsymbol{x}$ is then mapped as $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ to the measurements' random vector $\boldsymbol{y}$ $\mathbb{R}^m$ whose realizations are finite-length ciphertexts;
- M_2: for $n \to \infty$, we let $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}_k\}_{k=0}^{+\infty}$ be a real-valued random process. Its realizations or *infinite-length* plaintexts $\boldsymbol{x}$ are assumed to have finite power $W_{\boldsymbol{x}} = \lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} \boldsymbol{x}_k^2$. We may denote them as sequences $\boldsymbol{x} = \{\boldsymbol{x}^{(n)}\}_{n=0}^{+\infty}$ of finite-length plaintexts $\boldsymbol{x}^{(n)} = \left(\boldsymbol{x}_0 \cdots \boldsymbol{x}_{n-1}\right)^\top$. $\boldsymbol{\mathcal{X}}$ is mapped to either a random vector $\boldsymbol{y}$ of finite-length ciphertexts for finite m, or a random process $\boldsymbol{\mathcal{Y}} = \{\boldsymbol{y}_j\}_{j=0}^{+\infty}$ of infinite-length ciphertexts for $m, n \to \infty$, $\frac{m}{n} \to q$. Both cases are comprised of random variables $\boldsymbol{y}_j = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k} \boldsymbol{x}_k$.

We remark that the $\frac{1}{\sqrt{n}}$ scaling in the second model is meant to normalize the ciphertext to the length of the plaintext; this is not only theoretically needed for normalization purposes, but also practically required in the design of quantizer ranges.

Moreover, since both RGE and RAE random matrices are suitable to draw the encoding matrix (at least as long as they are chosen with i.i.d. symbols), we assume that the $\boldsymbol{A} \sim \mathrm{RAE}(\boldsymbol{I})$. This choice is motivated by the fact that the number of bits required to produce a single encoding matrix symbol is maximized, as the bits output by the PRNG when the key is expanded into a pseudorandom sequence are a precious resource. With this hypothesis, we will let any instance of the RAE encoding matrix be a generic, unique element in a long-period repeatable sequence.

Finally, the exploration of the security properties of CS will therefore require a discussion of two main forms of cryptanalysis: (*i*) statistical approaches that attempt to extract information about the plaintext from the ciphertext, and (*ii*) computational approaches that attempt to retrieve from the ciphertext and some other prior information (at worst amounting to the full plaintext), which encryption key was used to generate the encoding matrix by means of an exhaustive solution search.

9.1.3 Fundamental Security Limits

The golden standard in assessing whether a cryptosystem is endowed with security properties dates back to the seminal work of Claude Shannon [52]. In fact, the notion of secrecy introduced by Shannon requires that the distribution of the ciphertexts is identical independently of the plaintext being encrypted. This strict

condition ensures that any statistical analysis operated by collecting an arbitrarily large amount of ciphertexts shall not be able to produce at its output any information regarding the plaintext. Mathematically, this corresponds to the following statement.

Definition 9.1 (Perfect Secrecy) We say that a cryptosystem has *perfect secrecy* in the Shannon sense [52] if for all $\boldsymbol{x} \in \mathbb{R}^n, f(\boldsymbol{y}|\boldsymbol{x}) = f(\boldsymbol{y})$, or equivalently $f(\boldsymbol{x}|\boldsymbol{y}) = f(\boldsymbol{x})$.

This notion has also been expressed in terms of the *mutual information* [16, Section 8.5] between $\boldsymbol{x}$ and $\boldsymbol{y}$ (since we will not use this notion below, we leave its study to the reader and maintain our focus on the conditional PDFs mentioned in Definition 9.1).

The encoding performed by CS is a linear mapping, and as such it cannot completely hide the information contained in a plaintext $\boldsymbol{x}$. This has two main consequences. Firstly, linearity propagates scaling; hence, if we were provided with a plaintext $\boldsymbol{x}'$ and another one that is $\boldsymbol{x}'' = \alpha\boldsymbol{x}'$ for some $\alpha \in \mathbb{R}$, it would be simple to extract at least the scaling factor α. For the particular choice of $\alpha = 0$ this leads to a known argument of Rachlin and Baron [49] against the fundamental requirement for perfect secrecy.

Theorem 9.1 (Non-Perfect Secrecy of CS (from [49, Lemma 1])) *Let $\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \in \mathbb{R}^m$ be random vectors representing a plaintext and a ciphertext, and $\boldsymbol{A} \in \mathbb{R}^{m\times n}, m < n$ be any encoding matrix generated from an encryption key. This cryptosystem does not have perfect secrecy, i.e., the PDF of the ciphertext conditioned on the plaintext, $f(\boldsymbol{y}|\boldsymbol{x}) \neq f(\boldsymbol{y})$.*

To provide insight on the main security limit of CS, we also report the proof given in [49] with a slightly different notation.

Proof (Theorem 9.1) Assume there exists at least one plaintext $\boldsymbol{x} \notin \ker \boldsymbol{A}$ such that $f(\boldsymbol{x}) > 0$ (i.e., the plaintexts' PDF has nonzero density outside the null space of $\boldsymbol{A}$). Consider the ciphertext $\boldsymbol{y} = \boldsymbol{0}_m$. Then, we have

$$f(\boldsymbol{y})|_{\boldsymbol{y}=\boldsymbol{0}_m} \equiv \int_{\ker\, \boldsymbol{A}\subset\mathbb{R}^n} f(\boldsymbol{y}|\boldsymbol{x})|_{\boldsymbol{y}=\boldsymbol{0}_m} f(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \int_{\ker\, \boldsymbol{A}\subset\mathbb{R}^n} f(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} < 1,$$

Thus, since any one plaintext $\boldsymbol{x} \in \ker \boldsymbol{A}$ is such that $f(\boldsymbol{y}|\boldsymbol{x})|_{\boldsymbol{y}=\boldsymbol{0}_m} = 1$, this suffices to conclude that $f(\boldsymbol{y}|\boldsymbol{x})|_{\boldsymbol{y}=\boldsymbol{0}_m} \neq f(\boldsymbol{y})|_{\boldsymbol{y}=\boldsymbol{0}_m}$ and so that in all generality $f(\boldsymbol{y}|\boldsymbol{x}) \neq f(\boldsymbol{y})$. Note how this proof simply relies on the existence of $\ker \boldsymbol{A}$.

Secondly, linearity implies continuity. Hence, whenever $\boldsymbol{x}'$ and $\boldsymbol{x}''$ are close to each other for some fixed $\boldsymbol{A}$, the corresponding $\boldsymbol{y}'$ and $\boldsymbol{y}''$ will also be close to each other. The fact that $m < n$ would slightly complicate this setting since the counterimages of $\boldsymbol{y}''$ through $\boldsymbol{A}$ belong to a subspace in which points arbitrarily far from $\boldsymbol{x}'$ exist in principle (i.e., $\ker \boldsymbol{A} \neq \emptyset$). Yet, encoding matrices $\boldsymbol{A}$ are chosen by design as distance-preserving embeddings by the definition of RIP matrices. This fact is substantially opposite to the notion of *diffusion* (or *avalanche effect*) for digital-to-digital ciphers [52], i.e., the requirement that a change of one symbol in the plaintext should cause the change of all symbols in the ciphertext. Hence, close

ciphertexts strongly hint at close plaintexts (in the Euclidean sense) for some fixed $\boldsymbol{A}$. As an objection to this seemingly unavoidable issue, we resort to the assumption that a single draw of $\boldsymbol{A}$ may only be used once during the very large period of the pseudorandom encoding matrix generator. Thus, two neighboring plaintexts $\boldsymbol{x}', \boldsymbol{x}''$ will be mapped by different encoding matrices $\boldsymbol{A}', \boldsymbol{A}''$ to non-neighboring ciphertexts $\boldsymbol{y}', \boldsymbol{y}''$; if $\boldsymbol{A}', \boldsymbol{A}'' \sim \text{RAE}(\boldsymbol{I})$, then $\frac{mn}{2}$ of their symbols will differ on average, ensuring a diffusion-like property on the linear encoding performed by CS.

It is therefore well understood that two main security limits are observed in standard CS: firstly, perfect secrecy cannot be granted in general due to the existence of $\ker \boldsymbol{A}$; secondly, the design of encoding matrices with the RIP implies the preservation of distances between plaintexts and ciphertext, so the only way to design a cipher by CS is to ensure that any plaintext–ciphertext pair within the observation time of a malicious user is related by a different $\boldsymbol{A}$.

9.2 Statistical Cryptanalysis

We now focus on which security properties can still be achieved, and what information leaks into the distribution of the ciphertext by showing with a slightly different argument that the information leaking into the ciphertext by means of a linear encoding is the energy of the plaintext, as was confirmed in several independent works [5, 9, 49]. Moreover, we will prove that with any i.i.d. sub-Gaussian random encoding matrix a scaling factor α is actually *all* that can be inferred from the statistical analysis of CS-encoded ciphertexts.

The achievable security properties are shown in asymptotic and non-asymptotic configurations of CS, i.e., for $n \to \infty$ and finite n in full analogy with the models in Sect. 9.4.3.1. No guarantee of perfect secrecy can be given in full generality. We also remark that the presented evidence formally corresponds to statistical *ciphertext-only attacks* [57].

9.2.1 Asymptotic Secrecy

While perfect secrecy is unachievable, we now introduce the notion of *asymptotic spherical secrecy*. This is a weak form of secrecy, similar in principle to that of Wyner [58], yet posing an emphasis on same-power plaintexts. We show that CS with i.i.d. sub-Gaussian random encoding matrices has this property, i.e., no information can be inferred on a plaintext $\boldsymbol{x}$ in model (M_2) from the statistical properties of all its possible ciphertexts but its power. The implication of this property is the basic guarantee that a malicious *eavesdropper* intercepting the measurement vector $\boldsymbol{y}$ will not be able to extract any information on the plaintext except for its power.

Definition 9.2 (Asymptotic Spherical Secrecy) Let $\mathcal{X}$ be a random process whose plaintexts have finite power $0 < W_{\boldsymbol{x}} < \infty$, $\mathcal{Y}$ be the random process of the corresponding ciphertexts. A cryptosystem has asymptotic spherical secrecy if for any of its plaintexts $\boldsymbol{x} = \{\boldsymbol{x}^{(n)}\}_{n=0}^{+\infty}$ and ciphertexts $\boldsymbol{y} = \{\boldsymbol{y}^{(m)}\}_{m=0}^{+\infty}$ we have

$$f_{\mathcal{Y}|\mathcal{X}}(\boldsymbol{y}|\boldsymbol{x}) \underset{\text{dist.}}{\longrightarrow} f_{\mathcal{Y}|W_{\boldsymbol{x}}}(\boldsymbol{y}) \tag{9.1}$$

where the subscripts of f indicate the joint and conditional PDFs of the respective random processes, $f_{\mathcal{Y}|W_{\boldsymbol{x}}}$ denotes conditioning over plaintexts $\boldsymbol{x}$ with identical power $W_{\boldsymbol{x}}$, and $\underset{\text{dist.}}{\longrightarrow}$ denotes convergence in distribution as $m, n \to \infty$.
From an eavesdropper's point of view, asymptotic spherical secrecy means that given any ciphertext $\boldsymbol{y}$ we have

$$f_{\mathcal{X}|\mathcal{Y}}(\boldsymbol{x}|\boldsymbol{y}) \simeq \frac{f_{\mathcal{Y}|W_{\boldsymbol{x}}}(\boldsymbol{y})}{f_{\mathcal{Y}}(\boldsymbol{y})} f_{\mathcal{X}}(\boldsymbol{x}) \tag{9.2}$$

implying that any two different plaintexts with an identical, prior, and equal power $W_{\boldsymbol{x}}$ will remain indistinguishable from their ciphertexts in this asymptotic setting; thus, the following proposition holds.

Theorem 9.2 (Asymptotic Spherical Secrecy of i.i.d. Sub-Gaussian Random Encoding Matrices) *Let $\mathcal{X}$ be a random process with bounded-value plaintexts of finite power $W_{\boldsymbol{x}}$, $\boldsymbol{y}_j$ any random variable in the random process $\mathcal{Y}$ as in* (M_2). *For $n \to \infty$ we have*

$$f_{\boldsymbol{y}_j|\mathcal{X}}(\boldsymbol{y}_j) \underset{\text{dist.}}{\longrightarrow} \mathrm{N}(0, W_{\boldsymbol{x}}) \tag{9.3}$$

Thus, i.i.d. sub-Gaussian random encoding matrices provide independent, asymptotically secret measurements as in (9.1).
Since the rows of $\boldsymbol{A}$ are independent, the measurements $\boldsymbol{y}_j|W_{\boldsymbol{x}}$ conditioned over the power of the plaintext are also independent and Theorem 9.2 asserts that, although not secure in the Shannon sense, CS with suitable encoding matrices is able to conceal the plaintext up to the point of guaranteeing its security for $n \to \infty$. The proof of this statement follows.

Proof (Theorem 9.2) The proof is given by simple verification of the Lindeberg-Feller central limit theorem (see [7, Theorem 27.4]) for $\boldsymbol{y}_j$ in $\mathcal{Y}$ conditioned on a plaintext $\boldsymbol{x}$ of $\mathcal{X}$ in (M_2). By the hypotheses, the plaintext $\boldsymbol{x} = \{\boldsymbol{x}_k\}_{k=0}^{n-1}$ has power $0 < W_{\boldsymbol{x}} < \infty$ and $\forall k \in \{0, n-1\}, \boldsymbol{x}_k^2 \leq M_{\boldsymbol{x}}$ for some finite $M_{\boldsymbol{x}} > 0$. Any $\boldsymbol{y}_j|\mathcal{X} = \lim_{n\to\infty} \sum_{k=0}^{n-1} \boldsymbol{z}_{j,k}$ where we let $\boldsymbol{z}_{j,k} = \boldsymbol{A}_{j,k} \frac{\boldsymbol{x}_k}{\sqrt{n}}$ be a sequence of independent, non-identically distributed random variables of moments $\mu_{\boldsymbol{z}_{j,k}} = 0, \sigma^2_{\boldsymbol{z}_{j,k}} = \frac{\boldsymbol{x}_k^2}{n}$. By letting the partial sum $S_j^{(n)} = \sum_{k=0}^{n-1} \boldsymbol{z}_{j,k}$, its mean $\mu_{S_j^{(n)}} = 0$ and $\sigma^2_{S_j^{(n)}} = \frac{1}{n} \sum_{k=0}^{n-1} \boldsymbol{x}_k^2$. Thus, we verify the necessary and sufficient condition [7, (27.19)]

$$\lim_{n\to\infty} \max_{k=0,\ldots,n-1} \frac{\sigma^2_{z_{j,k}}}{\sigma^2_{S_j^{(n)}}} = 0,$$

by straightforwardly observing

$$\lim_{n\to\infty} \max_{k=0,\ldots,n-1} \frac{\frac{x_k^2}{n}}{\frac{1}{n}\sum_{k=0}^{n-1} x_k^2} \leq \frac{M_{\boldsymbol{x}}}{W_{\boldsymbol{x}}} \lim_{n\to\infty} \frac{1}{n} = 0.$$

The verification of this condition guarantees that $\boldsymbol{y}_j|\boldsymbol{\mathcal{X}} = \lim_{n\to\infty} S_j^{(n)}$ is normally distributed with $\mu_{\boldsymbol{y}_j} = 0$ and variance $\sigma^2_{\boldsymbol{y}_j|\boldsymbol{\mathcal{X}}} = \lim_{n\to\infty} \mathbf{E}[(S_j^{(n)})^2] = W_{\boldsymbol{x}}$, yielding (9.3).

The consequence of this definition of secrecy is that perfect secrecy for finite n can be achieved by a suitable normalization of the measurements, at least in the Gaussian random encoding matrix case. This has been formally stated by Bianchi et al. [5]. While this definitely holds in the mathematical sense, such a normalization clearly comes at a loss of relevant information about the power of the signal (or energy, for finite n); hence, the recovery of the original plaintext would be possible only up to a scaling (i.e., an information loss, albeit acceptable as similar to what happens in 1-bit CS [28]). If, moreover, the energy were to be transmitted separately on a secure channel, the perfect secrecy requirement would be delegated to the latter side-channel. Nevertheless, in the special case of a CS scheme designed to provide an exact normalization of the measurements perfect secrecy can be achieved; in general, this will be affected by the presence of quantization, leaving the interplay between the measurements' security and their resolution open to an interesting analysis.

Summarizing what we discussed, the asymptotic regime allowed the derivation of a weak notion of secrecy that shows how the information leakage from the plaintext into the ciphertext is only limited to $W_{\boldsymbol{x}}$ when the encoding matrices are drawn with i.i.d. sub-Gaussian entries.

9.2.2 Non-Asymptotic Secrecy

Since prospective applications of CS as a cryptosystem, and in general CS with i.i.d. sub-Gaussian random encoding matrices will entail finite-size configurations with n on the order of a few hundreds to over a million variables, it is of primary concern to show how the security properties introduced in the previous section scale in a non-asymptotic setting. The achievable security properties are tested below by two empirical methods and one theoretical result, that guarantees an extremely sharp rate of convergence to the distribution in (9.3) for finite n.

9.2.2.1 Statistical Cryptanalysis by Hypothesis Testing

As a first empirical illustration of the consequences of asymptotic spherical secrecy for finite n, we consider a statistical ciphertext-only attack aiming at distinguishing two unknown, orthogonal plaintexts $\boldsymbol{x}', \boldsymbol{x}'' : \langle \boldsymbol{x}', \boldsymbol{x}'' \rangle = 0$ from their ciphertexts $\boldsymbol{y}', \boldsymbol{y}''$. We assume that both plaintexts have finite energy (as in (M_1)). Then we let the attacker have access to a large number of ciphertexts collected in a set Y' obtained by applying different, randomly generated RAE($\boldsymbol{I}$) encoding matrices to a certain $\boldsymbol{x}'$, and to another set Y'' of ciphertexts, all of them corresponding either to $\boldsymbol{x}'$ or to $\boldsymbol{x}''$, and attempts to distinguish which is the true plaintext between the two.

This reduces the attack to an application of statistical hypothesis testing [16, Section 11.7], the null assumption being that the distribution underlying the statistical samples in Y'' is the same as that underlying the statistical samples in Y'. For maximum reliability we adopt a two-level testing approach: we repeat the above experiment for many random instances of the orthogonal plaintexts $\boldsymbol{x}'$ and $\boldsymbol{x}''$, performing a two-way Kolmogorov–Smirnov (KS) test to compare the empirical distributions obtained from Y' and Y'' produced by such orthogonal plaintexts. Each of the above KS tests yields a p-value quantifying the probability that two datasets coming from the same distribution exhibit larger differences w.r.t. those at hand. Given their meaning, individual p-values could be compared against a desired significance level to give a first assessment whether the null hypothesis (i.e., equality in distribution) can be rejected.

Yet, since it is known that p-values of independent tests on distributions for which the null assumption is true must be uniformly distributed in $[0, 1]$ we collect P of them and feed this second-level set of samples into a one-way KS test to assess uniformity at the standard significance level 5%.

This testing procedure is done for $n = 256$ in the cases $E_{\boldsymbol{x}'} = E_{\boldsymbol{x}''} = 1$ (same energy plaintexts) and $E_{\boldsymbol{x}'} = 1, E_{\boldsymbol{x}''} = 1.01$, i.e., with a 1% difference in energy between the two plaintexts. The resulting p-values for $P = 5000$ are computed by matching pairs of sets containing $5 \cdot 10^5$ ciphertexts, yielding the p-value histograms depicted in Fig. 9.1. We report these empirical PDFs of the p-values in the two cases along with the p-value of the second-level assessment, i.e., the probability that samples from a uniform distribution exhibit a deviation from a flat histogram larger than the observed one. When the two plaintexts have the same energy, all evidence concurs to say that the ciphertext distributions are statistically indistinguishable. In the second case, even a small difference in energy causes statistically detectable deviations and leads to a correct inference of the true plaintext between the two.

9.2.2.2 Statistical Cryptanalysis by the Kullback–Leibler Divergence

To reinforce even further the fact that any two plaintexts $\boldsymbol{x}', \boldsymbol{x}'' \in \mathbb{R}^n$ under different i.i.d. sub-Gaussian random encoding matrices cannot be inferred by a statistical

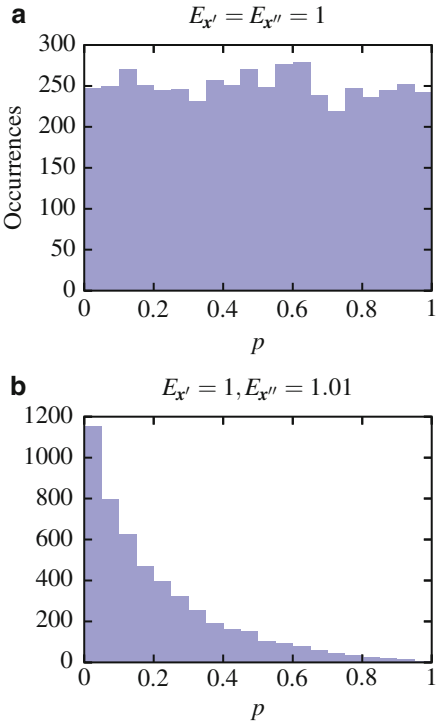


Fig. 9.1 Outcome of second-level KS statistical tests to distinguish between two orthogonal plaintexts $\boldsymbol{x}', \boldsymbol{x}''$. When $E_{\boldsymbol{x}'} = E_{\boldsymbol{x}''}$ (**a**), spherical secrecy applies and the uniform distribution of p-values shows that the corresponding ciphertexts are statistically indistinguishable. When $E_{\boldsymbol{x}'} \neq E_{\boldsymbol{x}''}$ (**b**), spherical secrecy does not apply and the distribution of p-values shows that the corresponding ciphertexts are distinguishable

analysis of their ciphertexts $\boldsymbol{y}', \boldsymbol{y}' \in \mathbb{R}^m$ even for finite n, we here attempt to do so by recalling the *Kullback–Leibler divergence* [16, (8.46)] of any two random variables a, b, i.e.,

$$\mathrm{D}\left(a \| b\right) = \int_{-\infty}^{+\infty} f(a) \log\left(\frac{f(a)}{f(b)}\right) \mathrm{d}a \, \mathrm{d}b \tag{9.4}$$

that is a simple measure of similarity between the PDF of a and b. We now evaluate $\mathrm{D}(\boldsymbol{y}'_j|\boldsymbol{x}' \| \boldsymbol{y}''_j|\boldsymbol{x}'')$ in (9.4) by considering two plaintexts $\boldsymbol{x}', \boldsymbol{x}'' \in \mathbb{R}^{2500}$ and extracting sequences of $n = \{50, 100, 150, \ldots, 2500\}$ samples from each of them. For every n the two sample collections are normalized to $E_{\boldsymbol{x}'} = E_{\boldsymbol{x}''} = 1$ and projected along 10^8 i.i.d. random vectors drawn as rows of matrices drawn from RAE($\boldsymbol{I}$), forming a large set of instances of $\boldsymbol{y}'_j|\boldsymbol{x}', \boldsymbol{y}''_j|\boldsymbol{x}''$. These samples are used to form the empirical[1]

[1]In order to enhance this evaluation, an optimal non-uniform binning is applied in the estimation of the histograms, since the PDFs are expected to be distributed as N(0, 1). This binning amounts to taking the inverse CDF of the standard normal distribution to obtain 256 uniform-probability bins, thus maximizing their entropy.

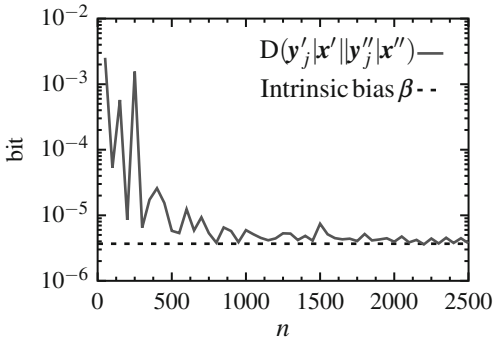


Fig. 9.2 Estimated Kullback–Leibler divergence between the probability distributions of two ciphertext elements corresponding to different plaintexts $\boldsymbol{x}', \boldsymbol{x}''$

PDF $\hat{f}(\boldsymbol{y}'_j|\boldsymbol{x}'), \hat{f}(\boldsymbol{y}''_j|\boldsymbol{x}'')$ and thus estimate the Kullback–Leibler divergence that is plotted in Fig. 9.2 against the value of n.

As a reference, we also report the theoretical expected value of the divergence estimated using two sets of n samples drawn from $\mathrm{N}(0, 1)$, i.e., due to the bias of the histogram estimator $\beta \simeq 3.67 \times 10^{-6}$ bit. It is clear that the distributions of the ciphertexts become statistically indistinguishable for n above a few hundreds, since the number of bits of information that can be apparently inferred from their differences (about 10^{-5} bit for $n > 500$) is mainly due to the bias β and thus cannot support a statistical cryptanalysis.

9.2.2.3 Non-Asymptotic Rate of Convergence

By now, we have observed with two methods how asymptotic spherical secrecy has finite-n effects; from a more formal point of view, we now evaluate the convergence rate of (9.3) for finite n to conclude with a guarantee that an eavesdropper intercepting the ciphertext will observe samples of an approximately Gaussian random vector bearing very little information in addition to the energy of the plaintext. We hereby consider $\boldsymbol{x}$ a random vector as in (M_1), for which a plaintext $\boldsymbol{x}$ of energy $E_{\boldsymbol{x}}$ lies on the sphere $E_{\boldsymbol{x}} S^2_{n-1} = \left\{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_2^2 \leq E_{\boldsymbol{x}}^2\right\}$. The procedure to verify the rate of convergence of (9.3) in this specific case substantially requires a study of the distribution of a linear combination of random variables, $\boldsymbol{y}_j = \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k} \boldsymbol{x}_k$ conditioned on $\boldsymbol{x} = \left(\boldsymbol{x}_0 \cdots \boldsymbol{x}_{n-1}\right)^\top \in E_{\boldsymbol{x}} S^2_{n-1}$.

The most general convergence rate for sums of i.i.d. random variables is given by the well-known Berry-Esseen Theorem [3] as $O(n^{-\frac{1}{2}})$. In our case we apply a recent, remarkable result of [31] that improves and extends this convergence rate, i.e., that addresses the case of inner products of i.i.d. random vectors (i.e., any row of $\boldsymbol{A}$) and vectors (i.e., the plaintexts $\boldsymbol{x}$) uniformly distributed on $\Sigma^{n-1}_{E_{\boldsymbol{x}}}$.

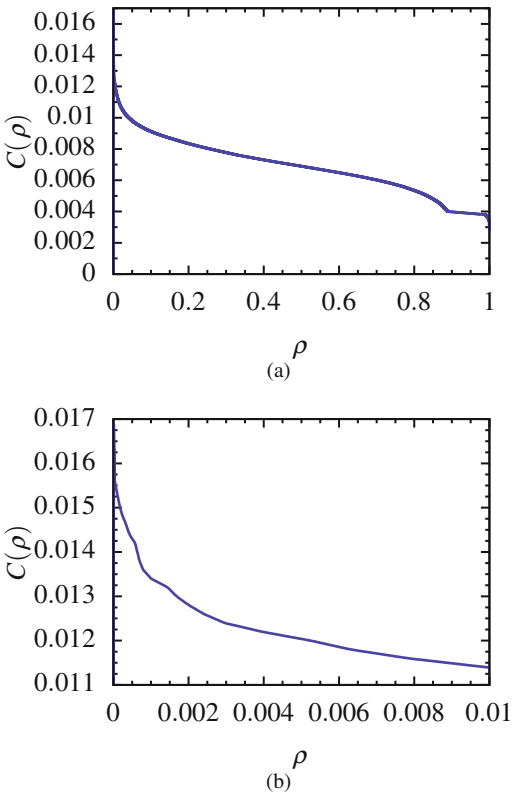


Fig. 9.3 Empirical evaluation of $C(\rho)$ in the convergence rate (9.5) based on a large number of plaintexts $\boldsymbol{x}$ on the sphere S^2_{n-1} and $n = 2^4, 2^5, \ldots, 2^{10}$. In the range $\rho \in (0, 1)$ (**a**) and zoomed in the range $\rho \in (0, 0.01)$ (**b**)

Theorem 9.3 (Rate of Convergence with i.i.d. Sub-Gaussian Random Encoding Matrices) *Let $\boldsymbol{x}, \boldsymbol{y}$ be random vectors as in* (M_1) *with $\boldsymbol{A}$ drawn from an i.i.d. sub-Gaussian random matrix ensemble with entries having zero-mean, unit-variance, and finite fourth-moment entries. For any $\rho \in (0, 1)$, there exists a subset $B \subseteq E_{\boldsymbol{x}} S^2_{n-1}$ with a probability measure $\sigma^{n-1}(B) \geq 1 - \rho$ such that all entries $\boldsymbol{y}_j$ in $\boldsymbol{y}$ verify*

$$\sup_{\alpha<\beta} \left| \int_\alpha^\beta f(\boldsymbol{y}_j | \boldsymbol{x} \in B) \, \mathrm{d}\boldsymbol{y}_j - \frac{1}{\sqrt{2\pi}} \int_\alpha^\beta e^{-\frac{t^2}{2E_{\boldsymbol{x}}}} \, \mathrm{d}t \right| \leq \frac{C(\rho)}{n} \tag{9.5}$$

for $C(\rho)$ a non-increasing function of ρ.

Theorem 9.3 with ρ sufficiently small means that it is most likely (actually, with probability exceeding $1 - \rho$) to observe an $O(n^{-1})$ convergence between $f(\boldsymbol{y}_j | \boldsymbol{x})$ and the limiting distribution $\mathrm{N}(0, E_{\boldsymbol{x}})$. The function $C(\rho)$ is loosely bounded in [31], so to complete this analysis we performed a thorough Montecarlo evaluation of its possible values. In particular, we have taken 10^4 instances of a random vector $\boldsymbol{x}$ uniformly distributed on S^2_{n-1} for each $n = 2^4, 2^5, \ldots, 2^{10}$. The PDF $f(\boldsymbol{y}_j | \boldsymbol{x})$ is estimated with the following procedure: we generate $5 \cdot 10^7$ rows drawn from the

RAE($\boldsymbol{I}$) and perform the usual linear encoding, thus yielding the same number of instances of $\boldsymbol{y}_j$ for each $\boldsymbol{x}$ and n. On this large sample set we are able to accurately estimate the previous PDF on 4096 equally probable intervals, and compare it to the same binning of the normal distribution as in the l.h.s. of (9.5) for each (x, n). This method yields sample values for (9.5), allowing an empirical evaluation of the quantity $C(\rho)$, as reported in Fig. 9.3. In this example, when $\rho \geq 10^{-3}$ Theorem 9.3 holds with $C(\rho) = 1.34 \cdot 10^{-2}$.

Proof (Theorem 9.3) We start by considering $\boldsymbol{y}_j$ in $\boldsymbol{y}$ of model (M_1) conditioned on a given $\boldsymbol{x}$ with finite energy $E_{\boldsymbol{x}}$. Each of such variables is a linear combination of n i.i.d. random variables $\boldsymbol{A}_{j,k}$ with zero-mean, unit-variance, and finite fourth-moments. The coefficients of this linear combination are the plaintext $\boldsymbol{x}$, which by now we assume to have $E_{\boldsymbol{x}} = 1$, i.e., to lie on the unit sphere S_{n-1}^2 of $\mathbb{R}^n$. Define $\gamma = \left(\frac{1}{n}\sum_{k=0}^{n-1} \mathbf{E}[\boldsymbol{A}_{j,k}^4]\right)^{\frac{1}{4}} < \infty$, which for RAE($\boldsymbol{I}$) encoding matrices is $\gamma = 1$, whereas for Gaussian random matrices $\gamma = 3^{\frac{1}{4}}$. This setting verifies [31, Theorem 1.1]: for any $\rho \in (0, 1)$ there exists a subset $B \subseteq S_{n-1}^2$ with a probability measure $\mu(B)$ such that $\sigma^{n-1}(B) = \frac{\mu(B)}{\mu(S_{n-1}^2)} \geq 1 - \rho$ and if $\boldsymbol{x} \in B$, then

$$\sup_{\substack{(\alpha,\beta)\in\mathbb{R}^2 \\ \alpha<\beta}} \left| \mathbf{P}\left[\alpha \leq \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k}\boldsymbol{x}_k \leq \beta\right] - \frac{1}{\sqrt{2\pi}} \int_\alpha^\beta e^{-\frac{t^2}{2}}\, \mathrm{d}t \right| \leq \frac{C(\rho)\gamma^4}{n} \tag{9.6}$$

with $C(\rho)$ a positive, non-increasing function. An application of this result to $\boldsymbol{x}$ with energy $E_{\boldsymbol{x}}$, i.e., on the sphere of radius $\sqrt{E_{\boldsymbol{x}}}$, $\gamma = 1$ ($\boldsymbol{A}$ Gaussian random) can be done by straightforwardly scaling the standard normal PDF in (9.6) to $\mathrm{N}(0, E_{\boldsymbol{x}})$, thus yielding the statement of Theorem 9.3.

9.3 Computational Cryptanalysis

In this section, we focus on quantifying the resistance of the lowest-complexity form of a CS cryptosystem, i.e., standard CS with RAE($\boldsymbol{I}$) encoding matrices, against known-plaintext attacks. These represent the most threatening form of computational cryptanalysis such a scheme will suffer. The properties and results of these attacks are fully explored here by theoretical means, as they can be mapped to a combinatorial optimization problem that models the most informed attack a malicious user may attempt.

We are going to show how the average number of candidate encoding matrix rows that match a plaintext–ciphertext pair is huge, thus making the search for the true encoding matrix inconclusive. Such a conclusion was anticipated by [46, 49], where the presented evidence essentially addressed brute-force enumeration; the main difference with our approach is that this quantification is theoretical, and yet

matches with surprising precision the odds of empirical attacks. Thus, the findings support a notion of computational security for CS-based encryption schemes.

9.3.1 Preliminary Considerations

We here focus on encoding matrices $\boldsymbol{A} \in \{-1, +1\}^{m \times n}$ drawn from a RAE($\boldsymbol{I}$), as they are remarkably simple and therefore suitable to be generated, implemented, and stored in digital devices. Due to their simplicity, these matrices are more easily subject to cryptanalysis. On the contrary, if many symbols were used in each element of $\boldsymbol{A}$ this would cause a rapid consumption of the bits generated by expansion of the secret. Thus, the RAE case serves as a basic reference for other random matrix ensembles and more complex configurations of cryptosystems based on CS.

To understand the relevance of the security issues addressed in this section, let us consider a first sequence of matrices $\{(\boldsymbol{A})_t\}_{t \in \mathbb{Z}}$ obtained by pseudorandom expansion of a seed Key $(\boldsymbol{A})$. Clearly, the strong assumption that any encoding matrix is never reused in the encoding is incompatible with the use of such sequences, as they will eventually repeat due to their pseudorandom nature. Nevertheless, we may assume that the sequences' period is sufficiently long to avoid repetition in the attacker's observation time. But even with this assumption standing, if an attacker was able to recover even a few elements in the above matrix sequences, this would potentially enable, e.g., PRNG cryptanalysis strategies (e.g., [39] for LFSRs) to break the cipher by retrieving the seeds in Fig. 9.12. Hence, to avoid such an event we focus on showing that a single, generic instance of $\boldsymbol{A}$ in $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ cannot be recovered even with the highest level of information, i.e., given $\boldsymbol{x}$ and $\boldsymbol{y}$.

Thus, we are considering a threatening situation in which an attacker has gained access to a known plaintext $\boldsymbol{x}$ corresponding to a known ciphertext $\boldsymbol{y}$. Based on these priors, the attacker aims at computing the true encoding $\boldsymbol{A}$, i.e., this malicious user attempts a *known-plaintext attack* (KPA). In the following we will consider this attack by assuming that *only one* $(\boldsymbol{x}, \boldsymbol{y})$ pair is known for a certain $\boldsymbol{A}$, consistently with the hypothesis that the same $\boldsymbol{A}$ only reappears after a long period.[2]

Starting from a single pair $(\boldsymbol{x}, \boldsymbol{y})$, depending on the level of information available to the attacker we obtain a KPA with increasing levels of threat (see Fig. 9.4); in fact, we could consider either a pure eavesdropper, Eve, and address the problem of retrieving $\boldsymbol{A}$ given $(\boldsymbol{x}, \boldsymbol{y})$, or allow even partial information about $\boldsymbol{A}$, leading to more sophisticated attacks; these will be expanded in Sect. 9.4.6.3. Since the KPA we discuss relies on deterministic knowledge of $\boldsymbol{x}$ and $\boldsymbol{y}$, we assume throughout this section that both plaintexts and ciphertexts are represented by digital words. This quantization is *unavoidable* as $\boldsymbol{x}$ and $\boldsymbol{y}$ will be stored, processed, and applied by a digital architecture from which the attacks are carried out. For simplicity, we let $\boldsymbol{x}$ be

[2]Note that if n independent $(\boldsymbol{x}, \boldsymbol{y})$ pairs were known for the same $\boldsymbol{A}$, one could resort to elementary linear algebra and infer the true encoding matrix by solving a simple linear system.

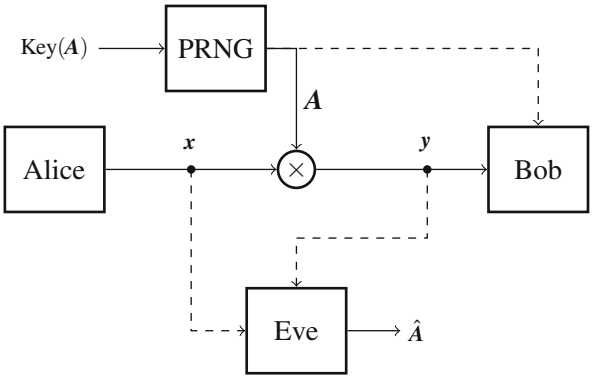


Fig. 9.4 A basic scheme of a known-plaintext attack as carried out by an eavesdropper Eve

such that its entries $\boldsymbol{x}_k \in \{-L, \ldots, -1, 0, 1, \ldots, L\}$ for some $L \in \mathbb{Z}_+$. Note that the number of bits representing the plaintext in this fashion is at least $\mathtt{b}_{\boldsymbol{x}} = \lceil \log_2(2L + 1) \rceil$, so we may assume $\mathtt{b}_{\boldsymbol{x}}$ is less than a few tens in typical embodiments (actually, $\mathtt{b}_{\boldsymbol{x}} \leq 16$ bit if the plaintext was previously generated by a common analog-to-digital converter). Consequently, the ciphertext will be represented by $\boldsymbol{y}$, so that each of its entries $\boldsymbol{y}_j$ is quantized with as many bits as needed to avoid any information loss. In this necessarily digital-to-digital perspective, we will see how the solutions in $\boldsymbol{A}$ are also a function of the number of bits representing the plaintext (and consequently the ciphertext).

Our KPA analysis applies on a single row[3] of $\boldsymbol{A}$. Furthermore, we note that the analysis is carried out in full compliance with Kerckhoffs's principle [30], i.e., the only information that the attackers are missing is their respective part of the encryption key, while any other detail on the sparsity basis is here regarded as known. The actual breaking of the encryption protocol would entail iterating the following attack for all m rows of many of the matrices in the sequence, thus requiring an even larger effort than the one described below. Nevertheless, even knowing one row without uncertainty could lead to a decryption of the pseudorandom sequence generating $\boldsymbol{A}$, hence the relevance of this simplified case.

9.3.2 Eavesdropper's Known-Plaintext Attack

Given a plaintext $\boldsymbol{x}$ and the corresponding ciphertext $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ we now assume the perspective of an eavesdropper, Eve, and attempt to recover $\boldsymbol{A}_{j,\cdot}$ with a set of symbols $(\hat{\boldsymbol{A}})_{j,\cdot} \in \{-1, +1\}^n$ such that the j-th symbol in the ciphertext,

$$\boldsymbol{y}_j = \sum_{k=0}^{n-1} \boldsymbol{A}_{j,k}\boldsymbol{x}_k = \sum_{k=0}^{n-1} \hat{\boldsymbol{A}}_{j,k}\boldsymbol{x}_k. \tag{9.7}$$

[3] $\boldsymbol{A}_{j,\cdot}$ here denotes the j-th row of a matrix $\boldsymbol{A}$.

Moreover, to favor the attacker[4] we assume all $\boldsymbol{x}_k \neq 0$. We now introduce a combinatorial optimization problem at the heart of the analyzed KPAs.

Problem 9.1 (Subset-Sum Problem) Let $\{\boldsymbol{u}_k\}_{k=0}^{n-1}, \boldsymbol{u}_k \in \{1, \ldots, L\}$ and $\upsilon \in \mathbb{Z}_+$. We define subset-sum problem (SSP, [38, Chap. 4]) the optimization problem of assigning n binary variables $\boldsymbol{b}_k \in \{0, 1\}$, $l = \{0, \ldots, n-1\}$ so that

$$\upsilon = \sum_{k=0}^{n-1} \boldsymbol{b}_k \boldsymbol{u}_k \tag{9.8}$$

We define *solution* any $\{\boldsymbol{b}_k\}_{k=0}^{n-1}$ verifying (9.8). In this configuration, the *density* of this combinatorial problem is defined as [32]

$$\delta(n, L) = \frac{n}{\log_2 L} \tag{9.9}$$

Although in general a SSP is NP-complete, not all of its instances are equally hard. In fact, it is known that *high-density* instances (i.e., with $\delta(n, L) > 1$) have plenty of solutions found or approximated by, e.g., dynamic programming, whereas *low-density* instances are harder, although for special cases polynomial-time algorithms have also been found [32]. On a historical note, such low-density hard SSP instances have been used in cryptography to develop the family of *public-key knapsack cryptosystems* [15, 42] although most have been broken with polynomial-time algorithms [45]. Problem 9.1 finds a direct application to model Eve's KPA as follows.

Theorem 9.4 (Eve's Known-Plaintext Attack) *The KPA to* $\boldsymbol{A}_{j,\cdot}$ *given* $(\boldsymbol{x}, \boldsymbol{y})$ *is equivalent to a SSP where each* $\boldsymbol{u}_k = |\boldsymbol{x}_k|$*, the variables*

$$\boldsymbol{b}_k = \frac{1}{2}\left(\operatorname{sign}(\boldsymbol{x}_k)\,\hat{\boldsymbol{A}}_{j,k} + 1\right)$$

and the sum

$$\upsilon = \frac{1}{2}\left(\boldsymbol{y}_j + \sum_{k=0}^{n-1} |\boldsymbol{x}_k|\right)$$

This SSP has a true solution $\{\bar{\boldsymbol{b}}_k\}_{k=0}^{n-1}$ *that is mapped to the row* $\boldsymbol{A}_{j,\cdot}$*, and other* candidate solutions *that verify* (9.8) *but correspond to matrix rows* $\hat{\boldsymbol{A}}_{j,\cdot} \neq \boldsymbol{A}_{j,\cdot}$.
We also define $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{A}_{j,\cdot})$ a *problem instance*. This mapping is obtained as follows.

[4] If any $\boldsymbol{x}_k = 0$ each corresponding term would give no contribution to the sum (9.7), thus making $\hat{\boldsymbol{A}}_{j,k}$ an undetermined variable in the attack. Hence, the sparsity of $\boldsymbol{x}$ would actually be an issue for the attacker, which is why the sparsity basis $\boldsymbol{D}$ never appears in the present evaluation.

Proof (Theorem 9.4) Define the binary variables $\boldsymbol{b}_k \in \{0,1\}$ so that $\text{sign}(\boldsymbol{x}_k)\hat{\boldsymbol{A}}_{j,k} = 2\boldsymbol{b}_k - 1$ and the positive coefficients $\boldsymbol{u}_k = |\boldsymbol{x}_k|$. With this choice (9.7) is equivalent to $\boldsymbol{y}_j = \sum_{k=0}^{n-1}(2\boldsymbol{b}_k - 1)\boldsymbol{u}_k$ which leads to a SSP with $\upsilon = \frac{1}{2}\left(\boldsymbol{y}_j + \sum_{k=0}^{n-1}|\boldsymbol{x}_k|\right)$. Since we know that each ciphertext entry $\boldsymbol{y}_j$ must correspond to the inner product between $\boldsymbol{x}$ and the row $\boldsymbol{A}_{j,\cdot}$, the latter's entries are straightforwardly mapped to the *true* solution of this SSP, i.e., $\{\bar{\boldsymbol{b}}_k\}_{k=0}^{n-1}$.

In our case we see that the density (9.9) is high since n is large and $\log_2 L$ is fixed by the digital representation of $\boldsymbol{x}$ (e.g., so that $\mathtt{b}_{\boldsymbol{x}} \leq 64$). We are therefore operating in a high-density region of problem (9.8). In fact, the resistance of the analyzed embodiment of CS to KPAs is not due to the hardness of the corresponding SSP but, as we show below, to the huge number of candidate solutions as n increases, among which an attacker should find the true solution to guess a single row of $\boldsymbol{A}$. Since no a priori criterion exists to select them, we consider them *indistinguishable*.

9.3.3 Expected Number of Solutions to an Eavesdropper's Known-Plaintext Attack

The next theorem calculates the expected number of candidate solutions to Eve's KPA by applying the theory developed in [51].

Theorem 9.5 (Expected Number of Solutions to Eve's Known-Plaintext Attack) *For large n, the expected number of candidate solutions of the KPA in Theorem 9.4, in which (i) all the coefficients $\{\boldsymbol{u}_k\}_{k=0}^{n-1}$ are i.i.d. uniformly drawn from $\{1,\ldots,L\}$, and (ii) the true solution $\{\bar{\boldsymbol{b}}_k\}_{k=0}^{n-1}$ is drawn with equiprobable and independent binary values, is*

$$\mathcal{S}_{\text{Eve}}(n,L) \simeq \frac{2^n}{L}\sqrt{\frac{3}{\pi n}} \tag{9.10}$$

The proof of Theorem 9.5 is given in the next section. This result (as well as the whole statistical mechanics framework from which it is derived) gives no hint on how much (9.10) is representative of finite-n behaviors. To compensate for that, we enumerated the solutions of several randomly generated small-n problem instances by using CPLEX as a binary programming solver [26] and forcing the computation of the full solution pool; this allowed a verification of the asymptotic expression of (9.10) by comparing its expected number of solutions with those effectively yielded by a computational implementation of Eve's KPA.

Such numerical evidence is reported in Fig. 9.5, where the empirical average number of solutions $\hat{\mathcal{S}}_{\text{Eve}}(n,L)$ to 50 problem instances with $L = 10^4$ and $n = \{16,\ldots,32\}$ is plotted and compared with (9.10). The remarkable matching observed there allows us to estimate, for example, that a KPA to the encoding of a gray-scale image of $n = 64 \times 64$ pixel quantized with $\mathtt{b}_{\boldsymbol{x}} = 8$ bit (unsigned) would

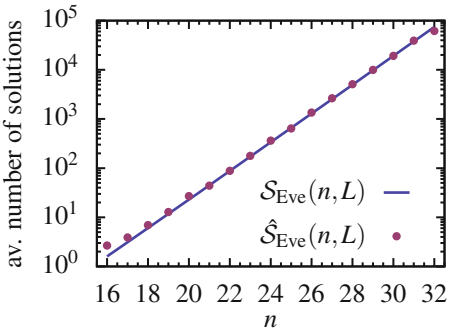


Fig. 9.5 Empirical average number of solutions for Eve's KPA compared to the theoretical approximation of (9.10) for $L = 10^4$

have to discriminate on the average between $1.25 \cdot 10^{1229}$ equally good candidate solutions for each of the rows of the encoding matrix. This number is not far from the total possible rows, $2^{4096} = 1.04 \cdot 10^{1233}$. Hence, any attacker using this strategy is faced with a deluge of candidate solutions, from which it would choose one presumed to be a piece of the encoding matrix to attempt a guess on $\boldsymbol{A}$.

Before proceeding to the proof of Theorem 9.5, let us introduce a technical definition required in the following developments.

Definition 9.3 We define the functions

$$F_p(a,b) = \int_0^1 \frac{\xi^p}{1+e^{a\xi-b}}\,\mathrm{d}\xi, \tag{9.11}$$

$$G_p(a,b) = \int_0^1 \frac{\xi^p}{\left(1+e^{a\xi-b}\right)\left(1+e^{b-a\xi}\right)}\,\mathrm{d}\xi. \tag{9.12}$$

We now proceed to proving the main statement by means of an interface with the theory developed by Sasamoto et al. [51] on the number of solutions of the SSP.

Proof (Theorem 9.5) Let us first note that, for large n, υ in Theorem 9.4 is an integer in the range $[0, \frac{nL}{2}]$, with the values outside this interval being asymptotically unachievable as $n \to \infty$ (see [51, Section 4]). We let $\tau = \frac{\upsilon}{nL}$, $\tau \in [0, \frac{1}{2}]$, and $a(\tau)$ be the solution in a of the equation $\tau = F_1(a,0)$ (i.e., [51, (4.2)]) that is unique since $F_p(a,0)$ in (9.11) is monotonically decreasing in a.

From [51, (4.1)] the number of solutions of a SSP with integer coefficients $\{\boldsymbol{u}_k\}_{k=0}^{n-1}$ uniformly distributed in $[1, L]$ is

$$\mathcal{S}_{\text{Eve}}(\tau,n,L) \simeq \frac{e^{n\left[a(\tau)\tau+\int_0^1 \log\left(1+e^{-a(\tau)\xi}\right)\mathrm{d}\xi\right]}}{\sqrt{2\pi nL^2 G_2(a(\tau),0)}}$$

that we anticipate to have an approximately Gaussian profile (see Fig. 9.6).

We now compute the average of $\mathcal{S}_{\text{Eve}}(\tau,n,L)$ in τ, that clearly depends on the probability of selecting any value of $\upsilon \in [0, \frac{nL}{2}]$, i.e., of $\tau \in [0, \frac{1}{2}]$. Since υ is the result of a linear combination, the probability that a specific value appears in a

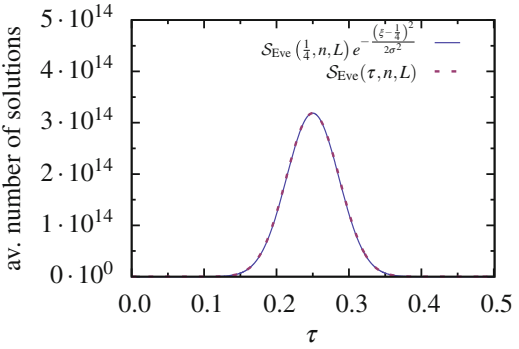


Fig. 9.6 Gaussian approximation of $\mathcal{S}_{\mathrm{Eve}}(\tau,n,L)$ for $n=64, L=10^4$ by letting $\sigma^2 \approx \frac{1}{12n}$

random instance of the SSP is proportional to the number of solutions associated with it. In normalized terms, the PDF of τ must be proportional to $\mathcal{S}_{\mathrm{Eve}}(\tau,n,L)$, i.e., τ is distributed as

$$f_\tau(t) = \frac{1}{\int_0^{\frac{1}{2}} \mathcal{S}_{\mathrm{Eve}}(\xi,n,L)\mathrm{d}\xi} \begin{cases} \mathcal{S}_{\mathrm{Eve}}(t,n,L), & 0 \le t \le \frac{1}{2} \\ 0, & \text{otherwise} \end{cases}$$

With $f_\tau(t)$ we can compute the expected number of solutions:

$$\mathbf{E}_\tau[\mathcal{S}_{\mathrm{Eve}}(\tau,n,L)] = \frac{\int_0^{\frac{1}{2}} \mathcal{S}^2_{\mathrm{Eve}}(\xi,n,L)\mathrm{d}\xi}{\int_0^{\frac{1}{2}} \mathcal{S}_{\mathrm{Eve}}(\xi,n,L)\mathrm{d}\xi} \tag{9.13}$$

Although we could resort to numerical integration, (9.13) can be simplified by exploiting what noted above, i.e., that $\mathcal{S}_{\mathrm{Eve}}(\tau,n,L)$ has an approximately Gaussian profile in τ (Fig. 9.6) with a maximum in $\tau = \frac{1}{4}$. Hence, the expectation in τ becomes

$$\begin{aligned} \mathbf{E}_\tau[\mathcal{S}_{\mathrm{Eve}}(\tau,n,L)] &\simeq \mathcal{S}_{\mathrm{Eve}}\left(\frac{1}{4},n,L\right) \frac{\displaystyle\int_{-\infty}^{\infty} \left(e^{-\frac{\left(\xi-\frac{1}{4}\right)^2}{2\sigma^2}}\right)^2 \mathrm{d}\xi}{\displaystyle\int_{-\infty}^{\infty} e^{-\frac{\left(\xi-\frac{1}{4}\right)^2}{2\sigma^2}} \mathrm{d}\xi} \\ &= \mathcal{S}_{\mathrm{Eve}}\left(\frac{1}{4},n,L\right) \frac{1}{\sqrt{2}} = \frac{2^n}{L}\sqrt{\frac{3}{\pi n}} \end{aligned}$$

that is actually independent of the σ^2 used in the Gaussian approximation, and in which we have exploited $a(\frac{1}{4}) = 0$ to obtain the statement of the theorem.

9.3.4 Expected Distance of Solutions to an Eavesdropper's Known-Plaintext Attack

A legitimate concern when Eve is presented with a large set of solutions output from a complete KPA to a row of $\boldsymbol{A}$ is that most of them could be good approximations of the true encoding matrix row. To see whether this is the case, we quantify the difference between $\boldsymbol{A}_{j,\cdot}$ and the corresponding candidate $\hat{\boldsymbol{A}}_{j,\cdot}$ resulting from a KPA in terms of their Hamming distance, i.e., as the number of entries in which they differ.

Theorem 9.6 (Expected Number of Solutions to Eve's Known-Plaintext Attack at a Given Hamming Distance from the True One) *The expected number of candidate solutions at Hamming distance h from the true solution of the KPA in Theorem 9.4, in which (i) all the coefficients* $\{\boldsymbol{u}_k\}_{k=0}^{n-1}$ *are i.i.d. uniformly drawn from* $\{1,\ldots,L\}$*, (ii) the true solution* $\{\bar{\boldsymbol{b}}_k\}_{k=0}^{n-1}$ *is drawn with equiprobable and independent binary values, is*

$$\mathcal{S}_{\text{Eve}}^{(h)}(n,L) = \binom{n}{h}\frac{P_h(L)}{2^h L^h} \tag{9.14}$$

where $P_h(L)$ *is a polynomial in L whose coefficients are reported in Table 9.1 for* $h = \{2,\ldots,15\}$.

The proof of this theorem and the derivation of Table 9.1 are reported below. We first want to propose some empirical evidence that the expression in (9.14) correctly anticipates the expected number of solutions at a given Hamming distance. The procedure simply entails processing the enumerated solutions in Sect. 9.3.2. Thus, Fig. 9.7 reports for $n = \{21, 23, \ldots, 31\}$ the empirical average, over 50 problem instances, of the number of solutions to Eve's KPA whose Hamming distance from the true one is a given value $h = \{2,\ldots,15\}$, as compared against the value predicted by (9.14) with the polynomial coefficients in Table 9.1. The remarkable matching we observe allows us to estimate that, in the case of a gray-scale image ($n = 4096$, $L = 128$), only $1.95 \cdot 10^{41}$ candidate solutions out of the average $1.25{\cdot}10^{1229}$ are expected to have a Hamming distance $h \leq 16$, while $6.33{\cdot}10^{76}$ attain a Hamming distance $h \leq 32$. Since these results apply to each row of the matrix being inferred, this indicates how the chance that a randomly chosen candidate solution is (or is close to) the true one is negligible.

We now proceed to present a proof of the result in (9.14) based on a counting argument.

Proof (Theorem 9.6) We here concentrate on counting the number of candidate solutions $\boldsymbol{b}$ to Eve's KPA that differ from the true one, $\bar{\boldsymbol{b}}$, by exactly h components (at Hamming distance h). We assume that $K \subseteq \{0,\ldots,n-1\}$ is the set of indices for which there is a disagreement, i.e., for all entries with index $k \in K$ we have $\boldsymbol{b}_k = 1 - \bar{\boldsymbol{b}}_k$; this set has cardinality h, and is one among $\binom{n}{h}$ possible sets. Since

Table 9.1 Table of coefficients of the polynomials $P_h(L) = \sum_{j=1}^{h-1} p_j^h L^j$ describing the expected Hamming distance of the solutions to Eve's KPA in (9.14) for $h = \{2, \ldots, 15\}$

h	p_1^h	p_2^h	p_3^h	p_4^h	p_5^h	p_6^h	p_7^h	p_8^h	p_9^h	p_{10}^h	p_{11}^h	p_{12}^h	p_{13}^h	p_{14}^h
2	2													
3	-3	3												
4	$\frac{14}{3}$	-4	$\frac{16}{3}$											
5	$-\frac{15}{2}$	$\frac{65}{12}$	$-\frac{15}{2}$	$\frac{115}{12}$										
6	$\frac{62}{5}$	$-\frac{15}{2}$	11	$-\frac{27}{2}$	$\frac{88}{5}$									
7	-21	$\frac{959}{90}$	$-\frac{203}{12}$	$\frac{707}{36}$	$-\frac{301}{12}$	$\frac{5887}{180}$								
8	$\frac{254}{7}$	$-\frac{140}{9}$	$\frac{1226}{45}$	$-\frac{266}{9}$	$\frac{334}{9}$	$-\frac{422}{9}$	$\frac{19328}{315}$							
9	$-\frac{255}{4}$	$\frac{2613}{112}$	$-\frac{731}{16}$	$\frac{14701}{320}$	$-\frac{457}{8}$	$\frac{2233}{32}$	$-\frac{1415}{16}$	$\frac{259723}{2240}$						
10	$\frac{1022}{9}$	$-\frac{2585}{72}$	$\frac{359105}{4536}$	$-\frac{7055}{96}$	$\frac{9869}{108}$	$-\frac{1725}{16}$	$\frac{28625}{216}$	$-\frac{48325}{288}$	$\frac{124952}{567}$					
11	$-\frac{1023}{5}$	$\frac{16973}{300}$	$-\frac{60775}{432}$	$\frac{5463953}{45360}$	$-\frac{435941}{2880}$	$\frac{7449761}{43200}$	$-\frac{19811}{96}$	$\frac{1091629}{4320}$	$-\frac{2764663}{8640}$	$\frac{381773117}{907200}$				
12	$\frac{4094}{11}$	$-\frac{2277}{25}$	$\frac{687791}{2700}$	$-\frac{72523}{360}$	$\frac{3907067}{15120}$	$-\frac{341143}{1200}$	$\frac{599327}{1800}$	$-\frac{7909}{20}$	$\frac{1045349}{2160}$	$-\frac{2205833}{3600}$	$\frac{41931328}{51975}$			
13	$-\frac{1365}{2}$	$\frac{591721}{3960}$	$-\frac{2020421}{4320}$	$\frac{44385419}{129600}$	$-\frac{7815847}{17280}$	$\frac{116257063}{241920}$	$-\frac{3192163}{5760}$	$\frac{110721221}{172800}$	$-\frac{13148473}{17280}$	$\frac{19285357}{20736}$	$-\frac{20345507}{17280}$	$\frac{20646903199}{13305600}$		
14	$\frac{16382}{13}$	$-\frac{44863}{180}$	$\frac{34353347}{39600}$	$-\frac{38237381}{64800}$	$\frac{1292711}{1600}$	$-\frac{42972293}{51840}$	$\frac{122732801}{129600}$	$-\frac{92420419}{86400}$	$\frac{53508931}{43200}$	$-\frac{76095383}{51840}$	$\frac{77441609}{43200}$	$-\frac{588168119}{259200}$	$\frac{866732192}{289575}$	
15	$-\frac{16383}{7}$	$\frac{1074679}{2548}$	$-\frac{583763}{360}$	$\frac{113982839}{110880}$	$-\frac{12673507}{8640}$	$\frac{58584511}{40320}$	$-\frac{400088153}{241920}$	$\frac{1033251187}{564480}$	$-\frac{23927713}{11520}$	$\frac{193398181}{80640}$	$-\frac{98109773}{34560}$	$\frac{279340567}{80640}$	$-\frac{1060693411}{241920}$	$\frac{467168310097}{80720640}$

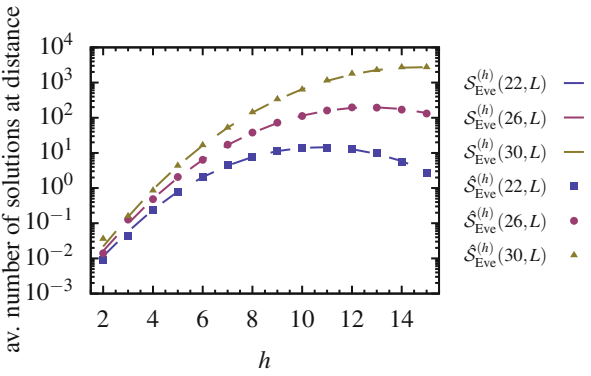


Fig. 9.7 Empirical average number of solutions for Eve's KPA at Hamming distance h from the true one, compared to the theoretical approximation of (9.14) for $L = 10^4$ and $n = 22, 26, 30$

both $\boldsymbol{b}$ and $\bar{\boldsymbol{b}}$ are solutions to the same SSP, and since the entries $\boldsymbol{b}_k = \bar{\boldsymbol{b}}_k$ are identical for $k \notin K$, $\sum_{k\in K} \left(1 - \bar{\boldsymbol{b}}_k\right) \boldsymbol{u}_k = \sum_{k\in K} \bar{\boldsymbol{b}}_k \boldsymbol{u}_k$ must hold, implying the equality

$$\sum_{\substack{k\in K \\ \bar{\boldsymbol{b}}_k=0}} \boldsymbol{u}_k - \sum_{\substack{k\in K \\ \bar{\boldsymbol{b}}_k=1}} \boldsymbol{u}_k = 0 \tag{9.15}$$

Although (9.15) recalls the well-known partition problem, in our case K is chosen by each problem instance that sets all $\boldsymbol{u}_k$ and $\bar{\boldsymbol{b}}_k$. Thus, (9.15) holds in a number of cases that depends on how many of the $2^h L^h$ possible assignments of all $\boldsymbol{u}_k$ and $\bar{\boldsymbol{b}}_k$ satisfy it. The only feasible cases are for $h > 1$, and to analyze them we assume $K = \{0, \ldots, h-1\}$ (the disagreements occur in the first h ordered indices) without loss of generality.

Moreover, when (9.15) holds for some $\{\bar{\boldsymbol{b}}_k\}_{k=0}^{n-1}$ it also holds for $\{1 - \bar{\boldsymbol{b}}_k\}_{k=0}^{n-1}$. Hence, we may count the configurations that verify (9.15) with $\bar{\boldsymbol{b}}_0 = 0$, knowing that their number will be only *half* of the total. With this, the configurations with $\bar{\boldsymbol{b}}_0 = 0$ must have $\bar{\boldsymbol{b}}_k = 1$ for *at least* one $l > 0$ in order to satisfy (9.15), giving $2^{h-1} - 1$ total cases to check.

The following paragraphs illustrate that, for $h < L$, the number of configurations that verify (9.15) can be written as a polynomial of order $h - 1$. With this in mind we can start with the explicit computation for $h = \{2, 3\}$.

1. for $h = 2$, there is only one feasible assignment for the entries $\bar{\boldsymbol{b}}_k$ of $\bar{\boldsymbol{b}}$, so $\boldsymbol{u}_0 = \boldsymbol{u}_1$ in (9.15), which makes $2L$ cases out of $2^2 L^2$;
2. for $h = 3$, one has 3 feasible assignments for the entries $\bar{\boldsymbol{b}}_k$ of $\bar{\boldsymbol{b}}$. Due to the symmetry of (9.15) all the configurations have the same behavior and we may focus on, e.g., $\bar{\boldsymbol{b}}_0 = \bar{\boldsymbol{b}}_1 = 0$ and $\bar{\boldsymbol{b}}_2 = 1 \Rightarrow \boldsymbol{u}_0 + \boldsymbol{u}_1 = \boldsymbol{u}_2$; this can be satisfied only when $\boldsymbol{u}_0 + \boldsymbol{u}_1 \leq L$, i.e., for $\frac{L(L-1)}{2}$ configurations. This makes a total of $2 \cdot 3 \cdot \frac{L(L-1)}{2} = 3L(L-1)$ over the $2^3 L^3$ possible configurations;
3. for $h > 3$, this procedure is much less intuitive; nevertheless, we can at least prove that the function $P_h(L)$ counting the configurations for which (9.15) holds is a polynomial in L of degree $h-1$. To show this, let us proceed in three steps.

a. Indicate with $\pi_{\bar{\boldsymbol{b}}}$ the $(h-1)$-dimensional subspace of $\mathbb{R}^h$ defined by $\sum_{\substack{k\in K\\ \boldsymbol{b}_k=0}} \boldsymbol{\xi}_k - \sum_{\substack{k\in K\\ \boldsymbol{b}_k=1}} \boldsymbol{\xi}_k = 0, \boldsymbol{\xi} \in \mathbb{R}^h$. The intersection $\alpha_{\bar{\boldsymbol{b}}}(L) = [1, L]^h \cap \pi_{\bar{\boldsymbol{b}}}$ is such that each assignment of $\{\boldsymbol{u}_k\}_{k=0}^{h-1} \in [1, L]^h$ satisfying (9.15) is an integer point in $\alpha_{\bar{\boldsymbol{b}}}$. To count those points define $\beta_{\bar{\boldsymbol{b}}}(L) = [0, L+1] \cap \pi_{\bar{\boldsymbol{b}}}$ and note that the number of integer points in $\alpha_{\bar{\boldsymbol{b}}}$ is equal to the number of integer points in the interior of $\beta_{\bar{\boldsymbol{b}}}$ (the points on the frontier of $\beta_{\bar{\boldsymbol{b}}}$ have at least one coordinate that is either 0 or $L+1$).

   Note how $[0, L+1]^h$ scales linearly with $L+1$ while $\pi_{\bar{\boldsymbol{b}}}$ is a subspace and therefore scale-invariant. Hence, their intersection $\beta_{\bar{\boldsymbol{b}}}(L)$ is a $h-1$-dimensional polytope that scales proportionally to the integer $L+1$, as required by Ehrhart's theorem [20]. The number $N_{\bar{\boldsymbol{b}}}(L)$ of integer points in $\beta_{\bar{\boldsymbol{b}}}(L)$ is then a polynomial in $L+1$ (and so L) of degree equal to the dimensionality of $\beta_{\bar{\boldsymbol{b}}}(L)$, i.e., $h-1$. From Ehrhart–Macdonald's reciprocity theorem [36] we know that the number of integer points in the interior of $\beta_{\bar{\boldsymbol{b}}}$ and thus in $\alpha_{\bar{\boldsymbol{b}}}$ is $(-1)^{h-1} N_{\bar{\boldsymbol{b}}}(-L)$, that is also a polynomial in L of degree $h-1$.

b. If two different assignments $\bar{\boldsymbol{b}}'$ and $\bar{\boldsymbol{b}}''$ are considered, then $\alpha_{\bar{\boldsymbol{b}}'}(L) \cap \alpha_{\bar{\boldsymbol{b}}''}(L) = [1, L]^h \cap \pi_{\bar{\boldsymbol{b}}'} \cap \pi_{\bar{\boldsymbol{b}}''}$. The same argument we used above tells us that the number of integer points in such an intersection is a polynomial in L of degree $h-2$ and, in general that the number of integer points in the intersection of any number of polytopes $\alpha_{\bar{\boldsymbol{b}}}(L)$ is a polynomial of degree not larger than $h-1$.

c. The number of configurations of $\{\boldsymbol{u}_k\}_{k=0}^{h-1}$ and $\bar{\boldsymbol{b}}$ that satisfy (9.15) w.r.t. the above K is the number of integer points in the union of all possible polytopes $\alpha_{\bar{\boldsymbol{b}}}$, i.e., $\bigcup_{\{\bar{\boldsymbol{b}}_k\}_{k=0}^{h-1}} \alpha_{\bar{\boldsymbol{b}}}(L)$. Such a number can be computed by the inclusion–exclusion principle that amounts to properly summing and subtracting the number of integer points in those polytopes and their various intersections. Since sum and subtraction of polynomials yield polynomials of non-increasing degree, we know that number is the evaluation of a polynomial $P_h(L)$ with degree not greater than $h-1$.

Let us finally write $P_h(L) = \sum_{j=0}^{h-1} p_j^h L^j$. In order to compute its coefficients p_j^h we may fix a binary configuration $\{\bar{b}_k\}_{k=0}^{h-1}$, count the points $\{\boldsymbol{u}_k\}_{k=0}^{h-1} \in \mathbb{Z}_+^h$ for which (9.15) is verified by means of integer partition functions (that also have a polynomial expansion), and subtract the points in which $\{\boldsymbol{u}_k\}_{k=0}^{h-1} \notin [1, L]^h$. By summation over all binary configurations, one can extract the coefficients associated with L^j for each h. Table 9.1 reports the result of this procedure as carried out by symbolic computation for $h \leq 15$.

9.4 Multiclass Encryption by Compressed Sensing

In this section we introduce a methodology for differentiating in a secure fashion the recovery quality attained by sets of receivers that are granted non-equal classes of access to the information content encoded by CS. We attain this by introducing

controlled matrix perturbations; hence, a study of their effect on signal recovery anticipates the proposed multiclass encryption protocol.

9.4.1 Security and Matrix Uncertainty by Random Perturbations

The sensitivity of recovery algorithms w.r.t. a perfect knowledge of the encoding matrix is a general issue for many applications in which CS acquires natural signals complying with a sparse signal model.

Quantifying this sensitivity in order to predict the result of signal recovery is therefore valuable when no a priori information can be exploited, e.g., when the encoding matrix is randomly perturbed without any exploitable structure. In this chapter we focus on this aspect by means of a simplified least-squares model for the signal recovery problem, which enables the derivation of its average performance estimate that depends only on the interaction between the encoding and perturbation matrices.

The effectiveness and stability of the resulting heuristic in CS configurations where this evaluation is meaningful is demonstrated by numerical exploration of signal recovery under three simple random perturbation matrix models in a variety of cases; the aim of this treatment is to develop a sense of the fact that this observation can be leveraged to introduce a CS-based cryptosystem that exploits such a sensitivity to missing information. Thus, understanding the effects of a perturbation in the encoding matrix is a valuable information in most applications of CS.

We here assume that the encoding matrix can be decomposed as[5] $\boldsymbol{A}^{(1)} = \boldsymbol{A}^{(0)} + \Delta\boldsymbol{A}$, where $\boldsymbol{A}^{(0)} \in \mathbb{R}^{m\times n}$ is known to the decoder while $\Delta\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a perturbation matrix. In this case any clue on $\Delta\boldsymbol{A}$ is generally unavailable, and the corresponding term of $\boldsymbol{y} = \boldsymbol{A}^{(1)}\boldsymbol{x} = \boldsymbol{A}^{(0)}\boldsymbol{x} + \Delta\boldsymbol{A}\boldsymbol{x}$ is signal dependent noise. We here let $\boldsymbol{D}$ be a known orthonormal basis that is available to the decoder, leaving the uncertainty to a perturbation matrix $\Delta\boldsymbol{A}$ and the actual sparse vector $\hat{\boldsymbol{\xi}}$ so that $\hat{\boldsymbol{x}} = \boldsymbol{D}\hat{\boldsymbol{\xi}}$.

9.4.1.1 Signal Recovery Algorithms with Matrix Perturbations

In the following, we will essentially discuss the sensitivity of signal recovery algorithms depending on the given prior information; this is indicated in the following as parameters in a "call" to these algorithms. Let us now start with the basic information that some structured noise is present on the measurements; with this information, a decoder may either:

[5]The reason for the superscripts $^{(1)}$ and $^{(0)}$ substantially distinguishes two levels of information, where the former denotes perfect knowledge of the truth and the latter a partial version of it.

1. choose to be *naive* and estimate $\hat{\boldsymbol{\xi}} = \mathrm{BP}(\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D})$ using Basis Pursuit (see Sect. 1.6), but feeding it the erroneous assumption that the measurements are not affected by noise, forcing $\boldsymbol{y} = \boldsymbol{A}^{(0)}\hat{\boldsymbol{x}}$;
2. in a more informed fashion, attempt to guess a noise threshold ε so that $\hat{\boldsymbol{\xi}} = \mathrm{BPDN}(\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D}, \varepsilon)$ using Basis Pursuit with Denoising (BPDN) which at least attempts denoising with the prior information that the solution is sparse. The noise threshold must be set so that the norm $\|\Delta\boldsymbol{A}\boldsymbol{x}\|_2 \leq \varepsilon$. In a particularly optimistic case, the actual norm $\varepsilon^\star = \|\Delta\boldsymbol{A}\boldsymbol{x}\|_2$ is here assumed to be known, in the so-called genie-tuning fashion;
3. in an ideal setting, be provided with the actual support of $\boldsymbol{\xi}$ in $\boldsymbol{D}$, T, so that it may estimate the solution via Oracle Least-Squares (OLS), i.e., $\hat{\boldsymbol{\xi}} = \mathrm{OLS}(\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D}, T) : \hat{\boldsymbol{\xi}}_T = (\boldsymbol{A}^{(0)}\boldsymbol{D}_{\cdot,T})^+\boldsymbol{y}$, $\hat{\boldsymbol{\xi}}_{T^c} = \boldsymbol{0}$. Note that this *non-perfectly informed* oracle solution is missing any prior on $\Delta\boldsymbol{A}$, therefore yielding the solution $\hat{\boldsymbol{\xi}}$ that minimizes the amount of error w.r.t. the components in T.

A variety of algorithms and problem formulations tackle the general case of signal recovery under perturbations [47, 60], where significant improvements are therein shown to be possible when some structure in $\Delta\boldsymbol{A}$ can be leveraged. However, we explicitly focus on the case in which $\Delta\boldsymbol{A}$ is drawn from a random matrix ensemble with i.i.d. entries that changes at each instance of $\boldsymbol{x}$; as noted in [47], signal recovery performances in this case are substantially limited by those of the aforementioned non-perfectly informed OLS estimate.

9.4.1.2 Recovery Guarantees with Matrix Perturbations

In terms of evaluating the effect of such matrix perturbations a first fundamental result was given by Herman and Strohmer [24], extending the established theoretical signal recovery guarantees for convex optimization [12] to such perturbed cases; the following definition is required for a summary of this result.

Definition 9.4 (Perturbation Constants (from [24])) Let

$$\sigma^{(\kappa)}_{\min\,/\,\max}(\boldsymbol{A}) = \min_{T\subseteq\{0,\dots,n-1\},|T|=\kappa} / \max \; \sigma_{\min\,/\,\max}(\boldsymbol{A}_{\cdot,T})$$

denote the extreme singular values among all κ-column sub-matrices of a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$. We define the perturbation constants

$$\begin{aligned} \epsilon^{(\kappa)}_{\boldsymbol{A}^{(1)}} &\geq \frac{\sigma^{(\kappa)}_{\max}(\Delta\boldsymbol{A}\,\boldsymbol{D})}{\sigma^{(\kappa)}_{\max}(\boldsymbol{A}^{(1)}\,\boldsymbol{D})} \\ \epsilon_{\boldsymbol{A}^{(1)}} \geq \frac{\sigma_{\max}(\Delta\boldsymbol{A}\,\boldsymbol{D})}{\sigma_{\max}(\boldsymbol{A}^{(1)}\,\boldsymbol{D})} &\geq \epsilon^{(\kappa)}_{\boldsymbol{A}^{(1)}} \end{aligned} \tag{9.16}$$

The modification of the celebrated stability theorem of CS [12] is here rephrased for the recovery of κ-sparse vectors in absence of other noise sources.

Theorem 9.7 (Stable Recovery by BPDN in the Presence of Perturbations (from [24, Theorem 2])) *Let* $\boldsymbol{y} = (\boldsymbol{A}^{(0)} + \Delta\boldsymbol{A})\boldsymbol{x} \in \mathbb{R}^m$ *be noisy measurements with additive perturbation noise* $\Delta\boldsymbol{A}\boldsymbol{x} \in \mathbb{R}^m$; $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$ *with* $\boldsymbol{D}$ *an orthonormal basis and* $\boldsymbol{\xi} \in \mathbb{R}^n : \|\boldsymbol{\xi}\|_0 = \kappa$; $\boldsymbol{A}^{(1)} = \boldsymbol{A}^{(0)} + \Delta\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *verify the RIP with constant* $\delta_{2\kappa} < \sqrt{2}\left(1 + \epsilon^{(2\kappa)}_{\boldsymbol{A}^{(1)}}\right)^{-2} - 1$ *and* $\epsilon^{(2\kappa)}_{\boldsymbol{A}^{(1)}} < 2^{\frac{1}{4}} - 1$. *Then* $\hat{\boldsymbol{\xi}} = \mathrm{BPDN}(\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D}, \gamma)$ *with noise threshold*

$$\gamma = \epsilon^{(\kappa)}_{\boldsymbol{A}^{(1)}} \sqrt{\frac{1+\delta_\kappa}{1-\delta_\kappa}} \|\boldsymbol{y}\|_2$$

is so that

$$\|\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}\|_2 \leq c_1' \gamma \tag{9.17}$$

where

$$c_1' = \frac{4\sqrt{1+\delta_{2\kappa}}\left(1 + \epsilon^{(2\kappa)}_{\boldsymbol{A}^{(1)}}\right)}{1 - (\sqrt{2}+1)\left[(1+\delta_{2\kappa})\left(1 + \epsilon^{(2\kappa)}_{\boldsymbol{A}^{(1)}}\right)^2 - 1\right]} \tag{9.18}$$

While formally correct, as in most other analyses based on the RIP the typical performances are significantly higher than the error norm bound (9.17). While prior works exist exploring the lower bound to the error at the output of sparse signal recovery [1, 2], the particular case of matrix perturbations is covered in some contributions [34, 35]. We then seek a design criterion following the principle that, given the non-linear behavior of sparse signal recovery algorithms (e.g., of convex optimization methods for non-smooth objective functions, such as the ℓ_1-norm), an approach based on sensible mathematical intuition and sufficiently motivated by simulation delivers applicable results.

9.4.1.3 Average Performances with Matrix Perturbations

The relative sophistication of BP and BPDN as non-smooth convex optimization problems prevents an average analysis of the sensitivity w.r.t. the perturbation matrix in typical recovery problems. Let us now assume in a simplified model that (*i*) $(\boldsymbol{A}^{(0)}, \Delta\boldsymbol{A})$ are drawn from two random matrix ensembles with known and i.i.d. distributions of entries, and (*ii*) an approximation of $\hat{\boldsymbol{x}} = \boldsymbol{D}\hat{\boldsymbol{\xi}}$ is obtained by solving $\mathrm{BP}(\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D})$ to satisfy $\boldsymbol{y} = \boldsymbol{A}^{(0)}\hat{\boldsymbol{x}}$. Pairing this with the original $\boldsymbol{y} = \boldsymbol{A}^{(1)}\boldsymbol{x}$ and with $\Delta\boldsymbol{A} = \boldsymbol{A}^{(1)} - \boldsymbol{A}^{(0)}$ we obtain

$$\boldsymbol{A}^{(0)}\Delta\boldsymbol{x} = \Delta\boldsymbol{A}\boldsymbol{x}, \ \Delta\boldsymbol{x} = \hat{\boldsymbol{x}} - \boldsymbol{x} \tag{9.19}$$

Starting from this, we further assume that $\Delta\boldsymbol{A}$ is indeed a *perturbation*, i.e., that its entity is small w.r.t. $\boldsymbol{A}^{(0)}$. In this way, the least-squares approximation error $\Delta\boldsymbol{x}$ is supposed to be small, so we could assume that $\hat{\boldsymbol{x}}$ lies in a ball centered on $\boldsymbol{x}$, and minimize its radius under the constraint (9.19), yielding the least-squares solution

$$\Delta\boldsymbol{x} = \operatorname{argmin} \boldsymbol{\Delta\zeta} \in \mathbb{R}^n \| \boldsymbol{\Delta\zeta} \|_2^2 \text{ s.t. } \boldsymbol{A}^{(0)} \boldsymbol{\Delta\zeta} = \Delta\boldsymbol{A}\boldsymbol{x} \tag{9.20}$$

that is $\Delta\boldsymbol{x} = (\boldsymbol{A}^{(0)})^+ \Delta\boldsymbol{A}\boldsymbol{x}$ (recalling that $\cdot^+$ denotes the Moore–Penrose pseudoinverse). To investigate the expectation of $\Delta\boldsymbol{x}$ when considered as a random vector, i.e., the mean square error of such a solution, we may then compute

$$\begin{aligned}
\mathbf{E}\left[\|\Delta\boldsymbol{x}\|_2^2\right] &= \operatorname{tr} \mathscr{C}_{\Delta\boldsymbol{x}} \\
&= \operatorname{tr} \mathbf{E}_{\boldsymbol{A}^{(0)},\Delta\boldsymbol{A},\boldsymbol{x}} \left[(\boldsymbol{A}^{(0)})^+ \Delta\boldsymbol{A}\boldsymbol{x}\boldsymbol{x}^\top \Delta\boldsymbol{A}^\top \left[(\boldsymbol{A}^{(0)})^+ \right]^\top \right], \\
&= \operatorname{tr} \mathbf{E}_{\boldsymbol{A}^{(0)},\Delta\boldsymbol{A}} \left[(\boldsymbol{A}^{(0)})^+ \Delta\boldsymbol{A} \mathscr{X} \Delta\boldsymbol{A}^\top \left[(\boldsymbol{A}^{(0)})^+ \right]^\top \right],
\end{aligned}$$

in the assumption that $\boldsymbol{A}^{(0)}$ and $\Delta\boldsymbol{A}$ are drawn from random matrix ensembles that are independent of $\boldsymbol{x}$, so the ratio

$$\frac{\mathbf{E}\left[\|\Delta\boldsymbol{x}\|_2^2\right]}{\mathbf{E}\left[\|\boldsymbol{x}\|_2^2\right]} = \operatorname{tr} \mathbf{E}_{\boldsymbol{A}^{(0)},\Delta\boldsymbol{A}} \left[(\boldsymbol{A}^{(0)})^+ \Delta\boldsymbol{A} \frac{\mathscr{X}}{E_{\boldsymbol{x}}} \Delta\boldsymbol{A}^\top \left[(\boldsymbol{A}^{(0)})^+ \right]^\top \right] \tag{9.21}$$

where the energy-normalized correlation matrix $\frac{\mathscr{X}}{E_{\boldsymbol{x}}}$ takes into account the second-order moments of the signal to acquire. Since $\boldsymbol{D}$ is assumed an orthonormal basis we may adopt a sparse signal model where each of $\binom{n}{\kappa}$ supports of $\boldsymbol{\xi}$ has the same probability, and its κ nonzero components are i.i.d. zero-mean random variables. With this, the correlation matrix $\frac{\mathscr{C}_{\boldsymbol{\xi}}}{E_{\boldsymbol{\xi}}} = \frac{1}{n}\boldsymbol{I}_n$ and $\mathscr{X} = \boldsymbol{D}\mathscr{C}_{\boldsymbol{\xi}}\boldsymbol{D}^\top = \frac{E_{\boldsymbol{\xi}}}{n}\boldsymbol{I}_n$. In this particular case, a simplified evaluation of the $\mathtt{ARSNR} = \frac{\mathbf{E}\left[\|\boldsymbol{x}\|_2^2\right]}{\mathbf{E}\left[\|\Delta\boldsymbol{x}\|_2^2\right]}$ due to a perturbation of the encoding matrix is possible, yielding

$$\mathtt{ARSNR} = n \left[\operatorname{tr} \mathbf{E}_{\boldsymbol{A}^{(0)},\Delta\boldsymbol{A}} \left[(\boldsymbol{A}^{(0)})^+ \Delta\boldsymbol{A}\Delta\boldsymbol{A}^\top \left[(\boldsymbol{A}^{(0)})^+ \right]^\top \right] \right]^{-1}. \tag{9.22}$$

The expectation on $\boldsymbol{A}^{(0)}$ and $\Delta\boldsymbol{A}$ depends on the CS configuration we are considering and may be computed in an empirical fashion by Montecarlo simulations for any random matrix ensemble of interest. On the other hand, the more suggestive and equivalent

$$\mathtt{ARSNR} = \left(\mathbf{E}_{\boldsymbol{A}^{(0)},\Delta\boldsymbol{A}} \left[\frac{1}{n} \sum_{j=0}^{n-1} \left[\sigma_j \left((\boldsymbol{A}^{(0)})^+ \Delta\boldsymbol{A} \right) \right]^2 \right] \right)^{-1} \tag{9.23}$$

links the expected performance to the average of the singular values of $\left(\boldsymbol{A}^{(0)}\right)^{+} \Delta\boldsymbol{A}$, yet it is much less attractive in terms of computational requirements for a numerical exploration. Note that such an estimate has a number of clear limitations:

1. since it focuses on non-denoising recovery (i.e., the solution of $\mathrm{BP}(\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D})$) it underestimates the attained recovery quality when the disturbance due to the perturbation can be compensated by the relative abundance of information on the problem due to (*i*) the availability of a large number of measurements in excess of the minimum required for recovery (therefore allowing efficient denoising) and (*ii*) knowing each instance's error norm $\varepsilon^{\star} = \|\Delta\boldsymbol{A}\boldsymbol{x}\|_2$ with which $\mathrm{BPDN}(\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D}, \varepsilon^{\star})$ may be solved;
2. the estimate will lose its validity for small values of m that do not allow an effective recovery, i.e., when even the *perfectly informed* $\mathrm{BP}(\boldsymbol{y}, \boldsymbol{A}^{(1)}\boldsymbol{D})$ fails. In this case it is not sensible to assume that either BP or BPDN identifies a good approximation of the true signal; the intrinsic reason is that $\|\Delta\boldsymbol{x}\|_2$ is not small (as the least-squares hypothesis in the neighborhood of $\boldsymbol{x}$ will not hold[6]) and the estimate will not yield a relevant prediction of the recovery quality.

Thus (9.22) and the more general (9.21) are expected to be most effective when m is sufficiently large, so that the phase transition of $\mathrm{BP}(\boldsymbol{y}, \boldsymbol{A}^{(1)}\boldsymbol{D})$ to the almost-sure recovery region has occurred, but not much larger than the minimum m required to achieve it. Actually, this is how efficient CS configurations will be designed and why (9.22) will match the examples presented below.

9.4.1.4 Practical Performances with Random Matrix Perturbations

In this section we choose different random matrix ensembles from which $\Delta\boldsymbol{A}$ is drawn, and introduce the *projection-to-perturbation ratio*,

$$\mathrm{PPR}_{\boldsymbol{A}^{(0)}, \Delta\boldsymbol{A}} = \frac{\mathbf{E}[\|\boldsymbol{A}^{(0)}\|_F^2]}{\mathbf{E}[\|\Delta\boldsymbol{A}\|_F^2]}$$

indicating the relative average energy of $\boldsymbol{A}^{(0)}$ w.r.t. $\Delta\boldsymbol{A}$ to control its impact.

Perturbation Models

We here focus on three perturbation models:

1. Dense Gaussian Additive (DGA): $\Delta\boldsymbol{A}$ is drawn from the Gaussian random matrix ensemble with i.i.d. entries of variance $\sigma_{\Delta\boldsymbol{A}}^2 = \frac{1}{\mathrm{PPR}_{\boldsymbol{A}^{(0)}, \Delta\boldsymbol{A}}}$;

[6] A more formal, yet similar intuition drives some of the considerations in [2], albeit addressing a slightly different estimation problem.

2. Dense Uniform Multiplicative (DUM): $\Delta\boldsymbol{A} = \boldsymbol{U} \circ \boldsymbol{A}^{(0)}$, where $\circ$ denotes the entry-wise product $\Delta\boldsymbol{A}_{j,k} = \boldsymbol{A}_{j,k} \cdot \boldsymbol{U}_{j,k}$ and the matrix $\boldsymbol{U}$ is drawn from a random matrix ensemble that is independent of $\boldsymbol{A}^{(0)}$ and has i.i.d. entries distributed as $\boldsymbol{U}_{j,k} \sim \mathrm{U}\left(-\frac{\beta}{2}, \frac{\beta}{2}\right)$ and $\beta = 2\sqrt{\frac{3}{\mathrm{PPR}_{\boldsymbol{A}^{(0)},\Delta\boldsymbol{A}}}}$;
3. Sparse Sign-Flipping (SSF): a random set of index pairs C is independently generated so that each entry

$$\Delta\boldsymbol{A}_{j,k} = \begin{cases} -2\boldsymbol{A}^{(0)}_{j,k}, & (j,k) \in C \\ 0, & (j,k) \notin C \end{cases} \tag{9.24}$$

corresponds to a sign-flipping of an element of $\boldsymbol{A}^{(0)}$, where each pair of $\{0,\ldots,m-1\} \times \{0,\ldots,n-1\}$ has a probability η of being chosen. The resulting sparse random matrix ensemble has a density $\eta = \frac{1}{4\,\mathrm{PPR}_{\boldsymbol{A}^{(0)},\Delta\boldsymbol{A}}}$ which controls $\sigma^2_{\Delta\boldsymbol{A}} = 4\eta$.

Experiments and Estimates

In this numerical experiment we consider a simple setting of dimensionality $n = 256$ and assume $\boldsymbol{D}$ is the DCT; we generate $\boldsymbol{\xi}$ as a white random vector by assuming equal probability of each of its $\binom{n}{\kappa}$ possible supports, letting its κ nonzero components be i.i.d. random variable distributed as $\mathrm{N}(0, \frac{1}{\kappa})$. We consider $\kappa = 8, 16, 32$ as prototypes of high- to low-sparsity signals.

The matrix $\boldsymbol{A}^{(0)} \in \mathbb{R}^{m \times n}$ is here drawn from the Gaussian random matrix ensemble with i.i.d. and unit-variance entries. As noted in the previous section, we expect the estimate (9.22) to apply after a perfectly informed BP yields a solution with sufficiently large m.

For a quantitative evaluation of this aspect, we generate 200 instances of $\boldsymbol{\xi}$, encode them with no perturbation, and then apply $\mathrm{BP}(\boldsymbol{y}, \boldsymbol{A}^{(1)}\boldsymbol{D})$ to measure the ARSNR with different values of m by means of SPGL1. Given that the precision setting of the solver allows a maximum $\mathrm{RSNR} \approx 120\,\mathrm{dB}$, by looking at the evidence in Fig. 9.8 we derive that a target ARSNR level of 110 dB is reached when $m = 103$ for $\kappa = 8$, $m = 138$ for $\kappa = 16$, and $m = 184$ for $\kappa = 32$, at which it is safe to assume that the decoder is operating after the phase transition.

At these (m, κ) pairs we explore the effect of perturbations and how closely it is predicted by (9.22); we choose the distribution parameters of the three models in section "Perturbation Models" to obtain a given $\mathrm{PPR}_{\boldsymbol{A}^{(0)},\Delta\boldsymbol{A}} \in \{0, 5, \ldots, 80\}\,\mathrm{dB}$. For the chosen (m, κ), we generate 200 instances of $(\boldsymbol{\xi}, \boldsymbol{A}^{(0)}, \Delta\boldsymbol{A})$, encode $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$ with $\boldsymbol{A}^{(1)} = \boldsymbol{A}^{(0)} + \Delta\boldsymbol{A}$ and attempt to recover $\hat{\boldsymbol{\xi}}$ by $\mathrm{BP}(\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D})$, $\mathrm{BPDN}(\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D}, \varepsilon^\star)$ and the non-perfectly informed $\mathrm{OLS}(\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D}, T)$. These three results are compared with the outcome of a Montecarlo simulation of our estimate in (9.22) averaged over 200 instances of $(\boldsymbol{A}^{(0)}, \Delta\boldsymbol{A})$.

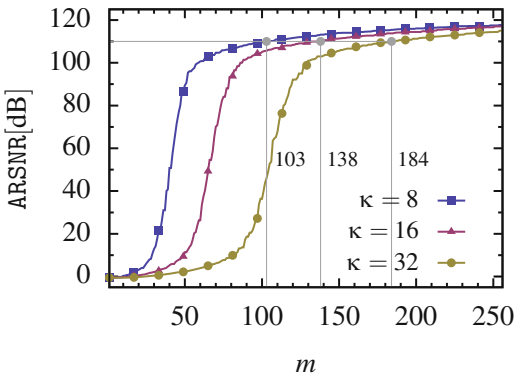


Fig. 9.8 ARSNR curves used to set m beyond the phase transition of BP$(\boldsymbol{y}, \boldsymbol{A}^{(1)}\boldsymbol{D})$

The results are depicted in Figs. 9.9, 9.10, and 9.11 in the case of $\kappa = 16$ and for the three different perturbation models. The ARSNR of each decoder can be compared with the estimate as the $\mathrm{PPR}_{\boldsymbol{A}^{(0)},\Delta\boldsymbol{A}}$ increases (i.e., the perturbation is made progressively smaller).

Moreover, since the estimate has negligible variations w.r.t. the perturbation model, we fix the latter to DGA and explore the effect of different sparsity levels at values for which the phase transition has occurred; the results are reported in Fig. 9.9. Note that, although it is only an estimate, (9.22) appears to be quite effective in anticipating the average performances right between BP$(\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D})$ and BPDN$(\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D}, \varepsilon^{\star})$. This is coherent with its derivation that starts from a non-denoising, naive BP but assumes that the recovery has the ability of coming as close as possible to the true solution in the least-squares sense.

As a result of this performance estimate, we can conclude that the estimate in (9.22) (or its extension to non-white signals in (9.21)) is indeed sufficiently accurate to predict the average recovery performances of signal recovery algorithms under matrix perturbations right between BP$(\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D})$ and BPDN$(\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D}, \varepsilon^{\star})$, and in particular when the configuration of CS being used is operating in the appropriate region of the phase transition curve, i.e., when the set of (m, n, κ) is so that recovery by BP is always feasible.

In the next section, we will focus on SSF as a method to introduce a controlled, conveniently generated perturbation in an encoding matrix to deliver data protection embedded in the sensing or encoding process; hence the interest in the devised estimate, that serves as a design formula for a lightweight encryption protocol.

9.4.2 *Elements of Multiclass Encryption*

We here discuss the introduction of a lightweight scheme for data protection, namely *multiclass encryption* by CS. Let us consider a scenario where multiple users receive the same ciphertext $\boldsymbol{y} = \boldsymbol{A}^{(1)}\boldsymbol{x}$, know the orthonormal basis $\boldsymbol{D}$ in which the plaintext

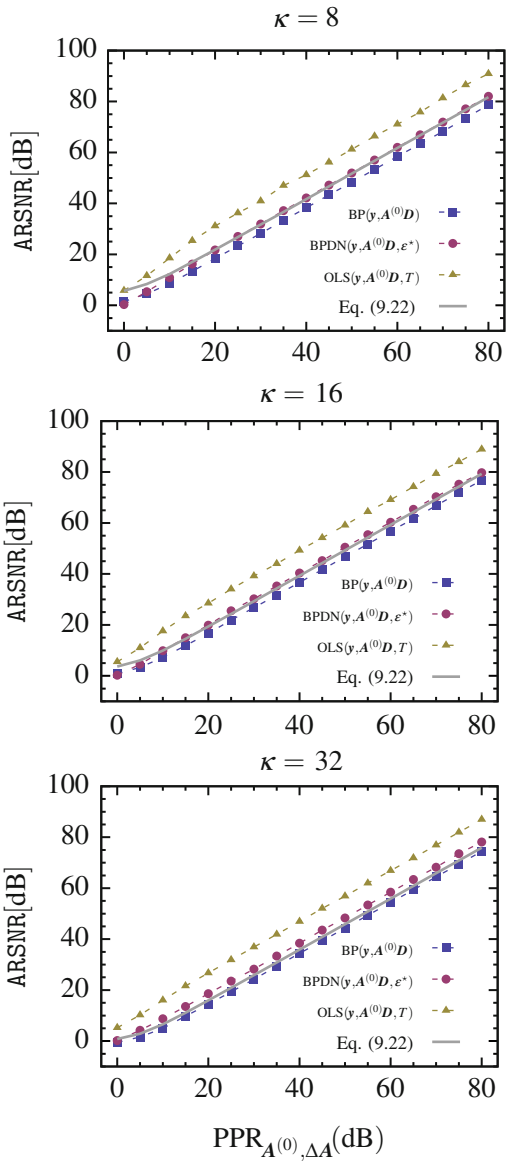


Fig. 9.9 Comparison of the ARSNR *versus* $\mathrm{PPR}_{\boldsymbol{A}^{(0)},\Delta\boldsymbol{A}}$ with the average performance estimate in (9.22) (*dashed*) against $\mathrm{BP}(\boldsymbol{y},\boldsymbol{A}^{(0)}\boldsymbol{D})$ (*empty circles*), $\mathrm{BPDN}(\boldsymbol{y},\boldsymbol{A}^{(0)}\boldsymbol{D},\varepsilon^\star)$ (*filled circles*), $\mathrm{OLS}(\boldsymbol{y},\boldsymbol{A}^{(0)}\boldsymbol{D},T)$ (*solid line*) for the DGA perturbation

$\boldsymbol{x}$ is κ-sparse, but are made different by the fact that some of them know the true encoding matrix $\boldsymbol{A}^{(1)}$, whereas the other users are only provided an approximate version of it, i.e., $\boldsymbol{A}^{(0)}$. The resulting mismatch between $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(0)}$—which will be used in the decoding process by the latter set of receivers—will then limit the quality of their signal recovery as detailed below.

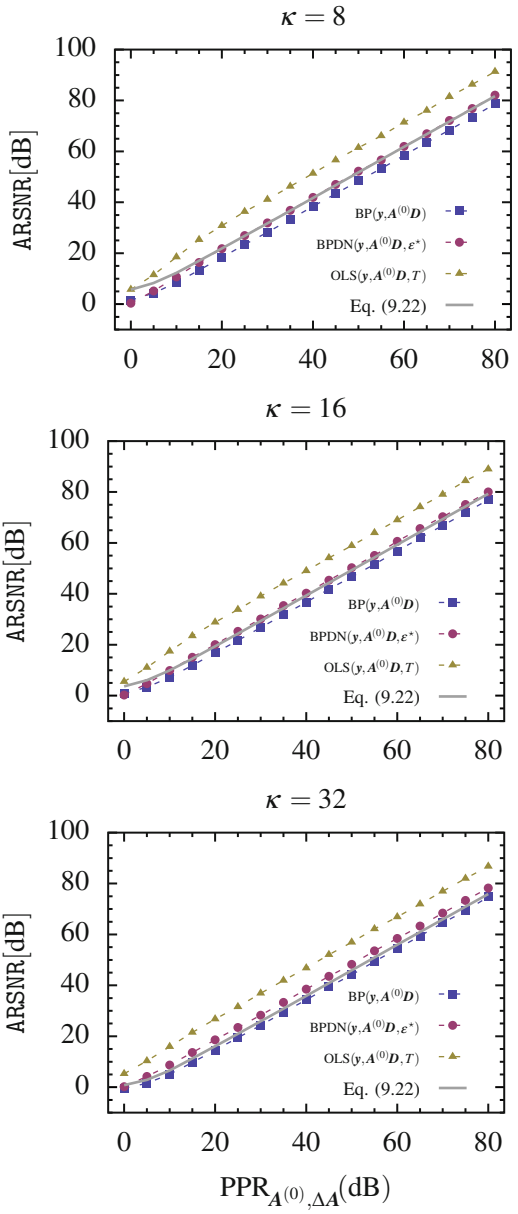


Fig. 9.10 Comparison of the ARSNR *versus* $\mathrm{PPR}_{\boldsymbol{A}^{(0)},\Delta\boldsymbol{A}}$ with the average performance estimate in (9.22) (*dashed*) against $\mathrm{BP}(\boldsymbol{y},\boldsymbol{A}^{(0)}\boldsymbol{D})$ (*empty circles*), $\mathrm{BPDN}(\boldsymbol{y},\boldsymbol{A}^{(0)}\boldsymbol{D},\varepsilon^{\star})$ (*filled circles*), $\mathrm{OLS}(\boldsymbol{y},\boldsymbol{A}^{(0)}\boldsymbol{D},T)$ (*solid line*) for the DUM perturbation

The Two-Class Case

With this principle in mind, a straightforward and undetectable method to introduce controlled perturbations is flipping the sign of a subset of the entries of the encoding matrix in a random pattern. More formally, let $\boldsymbol{A}^{(0)} \in \{-1,+1\}^{m\times n}$ denote the *initial*

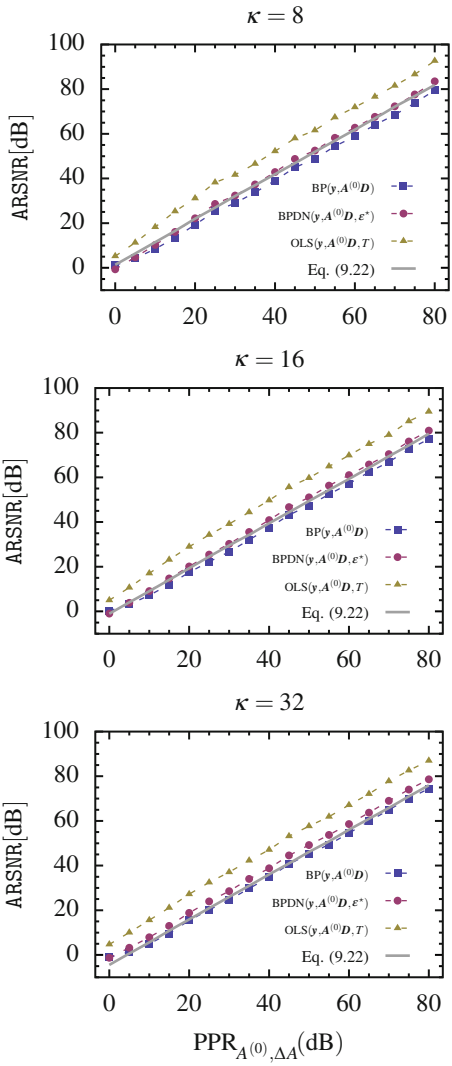


Fig. 9.11 Comparison of the ARSNR *versus* $\mathrm{PPR}_{\boldsymbol{A}^{(0)},\Delta\boldsymbol{A}}$ with the average performance estimate in (9.22) (*dashed*) against $\mathrm{BP}(\boldsymbol{y},\boldsymbol{A}^{(0)}\boldsymbol{D})$ (*empty circles*), $\mathrm{BPDN}(\boldsymbol{y},\boldsymbol{A}^{(0)}\boldsymbol{D},\varepsilon^{\star})$ (*filled circles*), $\mathrm{OLS}(\boldsymbol{y},\boldsymbol{A}^{(0)}\boldsymbol{D},T)$ (*solid line*) for the SSF perturbation

encoding matrix and $C^{(0)}$ be a subset of $c < mn$ index pairs chosen at random for each $\boldsymbol{A}^{(0)}$. We therefore construct the *true encoding matrix* $\boldsymbol{A}^{(1)}$ by taking

$$\forall (j,k) \in \{0,\ldots,m-1\} \times \{0,\ldots,n-1\},$$

$$\boldsymbol{A}^{(1)}_{j,k} = \begin{cases} \boldsymbol{A}^{(0)}_{j,k}, & (j,k) \notin C^{(0)} \\ -\boldsymbol{A}^{(0)}_{j,k}, & (j,k) \in C^{(0)} \end{cases} \tag{9.25}$$

and use it to encode $\boldsymbol{x}$ by $\boldsymbol{y} = \boldsymbol{A}^{(1)}\boldsymbol{x}$. Although this alteration simply involves inverting c randomly chosen sign bits in a buffer of mn pseudorandom symbols, we will use its linear perturbation model

$$\boldsymbol{A}^{(1)} = \boldsymbol{A}^{(0)} + \Delta\boldsymbol{A} \tag{9.26}$$

as in Sect. 9.4.1, where $\Delta\boldsymbol{A}$ is a c-sparse random matrix[7]

$$\forall (j,k) \in \{0,\ldots,m-1\} \times \{0,\ldots,n-1\},$$

$$\Delta\boldsymbol{A}_{j,k} = \begin{cases} 0, & (j,k) \notin C^{(0)} \\ -2\boldsymbol{A}^{(0)}_{j,k}, & (j,k) \in C^{(0)} \end{cases} \tag{9.27}$$

or equivalently

$$\forall (j,k) \in \{0,\ldots,m-1\} \times \{0,\ldots,n-1\},$$

$$\Delta\boldsymbol{A}_{j,k} = \begin{cases} 0, & (j,k) \notin C^{(0)} \\ 2\boldsymbol{A}^{(1)}_{j,k}, & (j,k) \in C^{(0)} \end{cases} \tag{9.28}$$

with sparse sign-flipping density $\eta = \frac{c}{mn}$. By doing so, any receiver is still provided an encoding matrix differing from the true one by an instance of $\Delta\boldsymbol{A}$. This perturbation is *undetectable*, i.e., $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(0)}$ are statistically indistinguishable since they are equal-probability realizations of the same RAE($\boldsymbol{I}$) ensemble, with all points in $\{-1,+1\}^{m\times n}$ having the same probability.

A *first-class* user receiving $\boldsymbol{y} = \boldsymbol{A}^{(1)}\boldsymbol{x} = (\boldsymbol{A}^{(0)} + \Delta\boldsymbol{A})\boldsymbol{x}$ and knowing $\boldsymbol{A}^{(1)}$ is therefore able to recover, in absence of other noise sources and with m sufficiently larger than the sparsity $\kappa : \boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}, \|\boldsymbol{\xi}\|_0 = \kappa$, the exact sparse solution $\hat{\boldsymbol{\xi}} = \boldsymbol{\xi}$ by solving BP($\boldsymbol{y}, \boldsymbol{A}^{(1)}\boldsymbol{D}$). A *second-class* user only knowing $\boldsymbol{y}$ and $\boldsymbol{A}^{(0)}$ is instead subject to an equivalent signal- and perturbation-dependent, non-white noise term $\boldsymbol{\epsilon}$ due to missing pieces of information on $\boldsymbol{A}^{(1)}$, that is

$$\boldsymbol{y} = \boldsymbol{A}^{(1)}\boldsymbol{x} = \boldsymbol{A}^{(0)}\boldsymbol{x} + \boldsymbol{\epsilon} \tag{9.29}$$

where $\boldsymbol{\epsilon} = \Delta\boldsymbol{A}\boldsymbol{x}$ is a pure disturbance since both $\Delta\boldsymbol{A}$ and $\boldsymbol{x}$ are unknown to the second-class receiver. Its approximation $\hat{\boldsymbol{x}}$ is obtained as the solution of, e.g., BP($\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D}$) or BPDN($\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D}, \varepsilon^\star$) with $\varepsilon^\star = \|\boldsymbol{\epsilon}\|_2$, where the considerations made in Sect. 9.4.1 seamlessly apply; performing signal recovery in the erroneous assumption that $\boldsymbol{y} = \boldsymbol{A}^{(0)}\hat{\boldsymbol{x}}$, i.e., with a corrupted encoding matrix, will lead to a noisy $\hat{\boldsymbol{x}} = \boldsymbol{D}\hat{\boldsymbol{\xi}}$.

[7]To be specific, it can be seen as drawn from a ternary-valued random matrix ensemble $\Delta\boldsymbol{A} \in \{-2,0,2\}^{m\times n}$ constructed from all the equiprobable assignments of c nonzero elements verifying (9.27). In a simplifying view, we let it have i.i.d. entries $\forall (j,k) \in \{0,\ldots,m-1\}, \{0,\ldots,n-1\}, \mathbf{P}[\Delta\boldsymbol{A}_{j,k} = -2] = \mathbf{P}[\Delta\boldsymbol{A}_{j,k} = 2] = \frac{\eta}{2}, \mathbf{P}[\Delta\boldsymbol{A}_{j,k} = 0] = 1-\eta$, so the density parameter actually controls the probability assignment.

In terms of recovery guarantees, while upper bounds on the recovery error norm $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2$ have been anticipated in the form of Theorems 1.7 and 9.7, the crucial matter in this section will be finding a lower bound to the error norm, i.e., a *best-case analysis* of the second-class recovery error. We anticipate that this will depend on the perturbation density η, which will be suitably chosen to fix the desired quality range for each class. This is precisely obtained in Sect. 9.4.3.1, together with a quantification of the upper bound by a direct application of Theorem 9.7.

A Multiclass Encryption Scheme

The two-class scheme may be iterated to devise an arbitrary number of user classes: a sparse sign-flipping can be applied on disjoint subsets of index pairs $C^{(u)}$, $u \in \{0, \dots, w-2\}$ of $\boldsymbol{A}^{(0)}$ so that

$$\boldsymbol{A}_{j,k}^{(u+1)} = \begin{cases} \boldsymbol{A}_{j,k}^{(u)}, & (j,k) \notin C^{(u)} \\ -\boldsymbol{A}_{j,k}^{(u)}, & (j,k) \in C^{(u)} \end{cases}$$

yielding the corresponding $\{\boldsymbol{A}^{(u)}\}_{u=0}^{w-1}$, each in turn associated with one of w user classes that progressively complete the knowledge of the true encoding $\boldsymbol{A}^{(w-1)}$. Thus, if the plaintext $\boldsymbol{x}$ is encoded with $\boldsymbol{A}^{(w-1)}$ we may distinguish *high-class* users knowing the complete encoding $\boldsymbol{A}^{(w-1)}$, *low-class* users knowing only $\boldsymbol{A}^{(0)}$, and *mid-class* users knowing $\boldsymbol{A}^{u+1}$ with $u = 0, \dots, w-3$. This simple technique can be applied to provide multiple classes of access to the information in $\boldsymbol{x}$ granting different signal recovery performances at the decoder.

A System Perspective

The strategy described in this section provides a multiclass encryption architecture where the shared secret between the CS encoder and each receiver is distributed depending on the quality level granted to the latter. In particular, the full encryption key of a w-class CS scheme is composed of w seeds, i.e., low-class users are provided the secret $\mathrm{Key}(\boldsymbol{A}^{(0)})$, class-1 users are provided $\mathrm{Key}(\boldsymbol{A}^{(1)}) = (\mathrm{Key}(C^{(0)}), \mathrm{Key}(\boldsymbol{A}^{(0)}))$ up to high-class users, which are given the key

$$\mathrm{Key}(\boldsymbol{A}^{(w-1)}) = \left(\mathrm{Key}(C^{(w-2)}), \cdots, \mathrm{Key}(C^{(0)}), \mathrm{Key}(\boldsymbol{A}^{(0)})\right).$$

An exemplary network implementing this policy is depicted in Fig. 9.13. This is reduced to the simple scheme of Fig. 9.12 in the case of a two-class encryption, where $\mathrm{Key}(C^{(0)}), \mathrm{Key}(\boldsymbol{A}^{(0)})$ fully define the key-agreement.

From the resources point of view, multiclass CS can be implemented with practically zero computational overhead. The encoding matrix generator is substantially a PRNG (e.g., an LFSR) and is structurally identical at both the

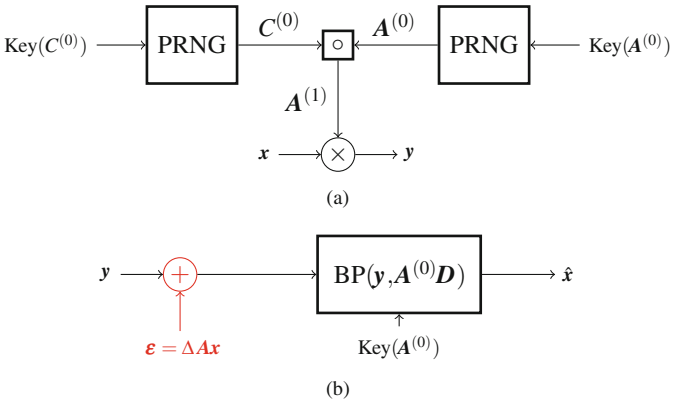


Fig. 9.12 An overview of two-class encryption by CS. (**a**) The encoder; *cross* here denotes the matrix-vector product, *open circle* the composition by (9.25). (**b**) A second-class decoder; the virtual effect of missing information on the encoding matrix at the decoder is highlighted in *red*

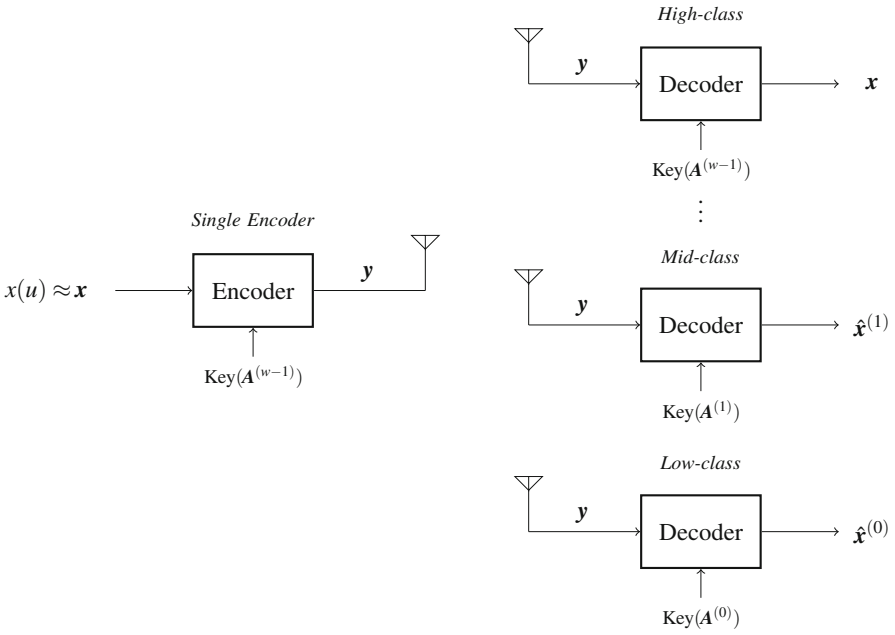


Fig. 9.13 A single-transmitter, multiple-receiver multiclass CS network: the encoder acquires an analog signal $x(u)$ by CS and transmits the measurement vector $\boldsymbol{y}$. Low-quality decoders reconstruct a signal approximation with partial knowledge of the encoding, resulting in perturbation noise and leading to an approximate solution $\hat{\boldsymbol{x}}^{(u)}$ for the u-th user class

encoder and high-class decoder side, whereas lower-class decoders may use the same encoding matrix generation scheme but are unable to rebuild the true one due to the missing pieces of the shared secret, i.e., Key($C^{(u)}$).

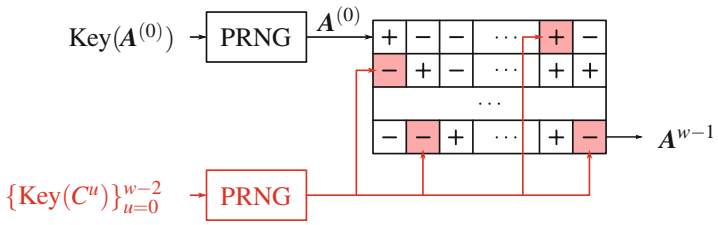


Fig. 9.14 Encoding matrix generator architecture

The initial matrix $\boldsymbol{A}^{(0)}$ is, as anticipated, updated from a pseudorandom binary stream generated by expanding Key($\boldsymbol{A}^{(0)}$) with a PRNG. The introduction of sign-flipping is a simple postprocessing step carried out on the stream buffer by reusing the same PRNG architecture and expanding the corresponding Key($C^{(u)}$), thus having minimal computational cost (see Fig. 9.14). Of course, the PRNGs have to be carefully chosen to avoid cryptanalysis [39]; however, since the values generated by this PRNG are never exposed, cryptographically secure PRNGs [40] or security-enhancing primitives on the output [19] may be avoided to save resources, provided that the period with which the matrices are reused is kept sufficiently large.

9.4.3 Properties and Main Results

9.4.3.1 Recovery Error Guarantees and Bounds

We now analyze the properties of multiclass CS starting from some statistical priors on the signal being encoded. Rather than relying on its a priori distribution, our analysis uses general moment assumptions that may correspond to many probability distributions on the signal domain. In order to quantify the recovery quality performance gap between low- and high-class users receiving the same measurements $\boldsymbol{y} = \boldsymbol{A}^{(1)}\boldsymbol{x}$, we now provide performance bounds on the recovery error in the simple two-class case, starting from the basic intuition that if the sparsity basis of $\boldsymbol{x}$ is not the canonical basis, then most plaintexts $\boldsymbol{x} \notin \ker \Delta\boldsymbol{A}$ so the perturbation noise $\boldsymbol{\epsilon} \neq \boldsymbol{0}_m$.

Main Results

The following results aim at predicting the best-case recovery quality (or equivalently, a recovery error lower bound) of any second-class decoder that assumes $\boldsymbol{y}$ was encoded by $\boldsymbol{A}^{(0)}$, whereas $\boldsymbol{y} = \boldsymbol{A}^{(1)}\boldsymbol{x}$ in absence of other noise sources and regardless of the sparsity of $\boldsymbol{x}$. Since $\boldsymbol{A}^{(1)} \sim \text{RAE}(\boldsymbol{I})$, any exact signal recovery guarantee based on the properties of this matrix ensemble holds when encoding $\boldsymbol{x}$

by $\boldsymbol{A}^{(1)}$. By such guarantees, the dimensionality m of the measurement vector $\boldsymbol{y}$ must exceed the sparsity κ by a quantity depending on the rate $\frac{\kappa}{n}$. In the following, we will assume that $\frac{m}{n}$ and $\frac{\kappa}{n}$ grant that a decoder knowing the true encoding $\boldsymbol{A}^{(1)}$ is able to accurately reconstruct the original signal by $\mathrm{BP}(\boldsymbol{y}, \boldsymbol{A}^{(1)}\boldsymbol{D})$.

We now introduce a result that shows how the recovery error norm suffered by a second-class receiver is *at least* (rather than *at most*, as is usually the case for performance guarantees in CS) a certain quantity essentially depending on the nature of the perturbation $\Delta\boldsymbol{A} = \boldsymbol{A}^{(1)} - \boldsymbol{A}^{(0)}$; this will serve as a basic design guideline for multiclass encryption schemes.

Theorem 9.8 (Second-Class Recovery Error Lower Bound (Non-Asymptotic Case)) *Let:*

- $\boldsymbol{A}^{(0)}, \boldsymbol{A}^{(1)} \in \{-1, +1\}^{m\times n}$ *be drawn from the RAE*$(\boldsymbol{I})$ *and* $\Delta\boldsymbol{A}$ *be as in* (9.27) *with density* $\eta \leq \frac{1}{2}$*;*
- $\boldsymbol{x} \in \mathbb{R}^n$ *be as in* (M_1) *with finite* $E_{\boldsymbol{x}} = \mathbf{E}\left[\sum_{k=0}^{n-1} \boldsymbol{x}_k^2\right]$, $F_{\boldsymbol{x}} = \mathbf{E}\left[\left(\sum_{k=0}^{n-1} \boldsymbol{x}_k^2\right)^2\right]$.

For any $\theta \in (0, 1)$*, and any instance of* $\boldsymbol{y} = \boldsymbol{A}^{(1)}\boldsymbol{x}$*,* $\hat{\boldsymbol{x}}$ *that satisfies* $\boldsymbol{y} = \boldsymbol{A}^{(0)}\hat{\boldsymbol{x}}$ *is such that*

$$\mathbf{P}\left[\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2^2 \geq \frac{4\eta m E_{\boldsymbol{x}}}{\sigma_{\max}(\boldsymbol{A}^{(0)})^2}\,\theta\right] \geq \zeta \tag{9.30}$$

where

$$\zeta = \frac{1}{1 + (1-\theta)^{-2}\left[\left[1 + \frac{1}{m}\left(\frac{3}{2\eta} - 1\right)\right]\frac{F_{\boldsymbol{x}}}{E_{\boldsymbol{x}}^2} - 1\right]} \tag{9.31}$$

This is extended to the asymptotic case (i.e., model (M_2)) as follows.

Theorem 9.9 (Second-Class Recovery Error Lower Bound (Asymptotic Case)) *Let:*

- $\boldsymbol{A}^{(0)}, \boldsymbol{A}^{(1)}, \Delta\boldsymbol{A}, \eta$ *be as in Theorem 9.8 as* $m, n \to \infty$, $\frac{m}{n} \to q$*;*
- $\boldsymbol{\mathcal{X}}$ *be as in* (M_2)*,* α*-mixing [7, (27.25)], with finite* $W_{\boldsymbol{x}} = \lim_{n\to\infty} \frac{1}{n}\mathbf{E}\left[\sum_{k=0}^{n-1} \boldsymbol{x}_k^2\right]$ *and uniformly bounded* $\mathbf{E}[\boldsymbol{x}_k^4] \leq m_{\boldsymbol{x}}$ *for some* $m_{\boldsymbol{x}} > 0$*.*

For any $\theta \in (0, 1)$*, and any instance of* $\boldsymbol{y} = \frac{1}{\sqrt{n}}\boldsymbol{A}^{(1)}\boldsymbol{x}$*,* $\hat{\boldsymbol{x}}$ *that satisfies* $\boldsymbol{y} = \frac{1}{\sqrt{n}}\boldsymbol{A}^{(0)}\hat{\boldsymbol{x}}$ *is such that*[8]

$$\mathbf{P}\left[W_{\hat{\boldsymbol{x}}-\boldsymbol{x}} \geq \frac{4\eta q W_{\boldsymbol{x}}}{(1+\sqrt{q})^2}\,\theta\right] \simeq 1. \tag{9.32}$$

[8]Clearly the recovery error power $W_{\hat{\boldsymbol{x}}-\boldsymbol{x}} = \lim_{n\to\infty} \frac{1}{n}\sum_{k=0}^{n-1}(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)^2$.

The proof of these statements is given below. Simply put, Theorems 9.8 and 9.9 state that a second-class decoder recovering with any algorithm $\hat{\boldsymbol{x}}$ such that $\boldsymbol{y} = \boldsymbol{A}^{(0)}\hat{\boldsymbol{x}}$ is subject to a recovery error whose norm, with high probability, exceeds a quantity depending on the density η of the perturbation $\Delta\boldsymbol{A}$, the undersampling rate $\frac{m}{n}$, and the average energy $E_{\boldsymbol{x}}$ or power $W_{\boldsymbol{x}}$, respectively.

In particular, the non-asymptotic case in (9.30) is a probabilistic lower bound: as a quantitative example, by assuming it holds with probability $\zeta = 0.98$ and that $\frac{F_{\boldsymbol{x}}}{E_{\boldsymbol{x}}^2} = 1.0001, n = 1024, m = 512, \sigma_{\max}(\boldsymbol{A}^{(0)}) \simeq \sqrt{m} + \sqrt{n}$ one could take an arbitrary $\theta = 0.1 \Rightarrow \eta = 0.1594$ to obtain $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2^2 \geq 0.0109$ w.r.t. random vectors having average energy $E_{\boldsymbol{x}} = 1$. In other words, with probability 0.98 a perturbation of density $\eta = 0.1594$ will cause a minimum recovery error norm of 19.61 dB.

A stronger asymptotic result holding with probability 1 on the recovery error power $W_{\hat{\boldsymbol{x}}-\boldsymbol{x}}$ is then reported in Theorem 9.9 under broadly verified assumptions on the random process $\boldsymbol{\mathcal{X}}$, where θ can be arbitrarily close to 1 and only affecting the convergence rate to this lower bound. The bounds in (9.30) and (9.32) are adopted as reference best-cases in absence of other noise sources for the second-class decoder, which actually exhibits higher recovery error for most problem instances and reconstruction algorithms as well illustrated in the exemplary applications of Sect. 9.4.5.

9.4.3.2 Proof of Main Results on Multiclass Encryption

We now give a technical proof of Theorems 9.8 and 9.9. We first introduce a Lemma that gives a self-contained probabilistic result on the Euclidean norm of $\boldsymbol{\epsilon}$ in (9.29).

Lemma 9.1 *Let:*

- $\boldsymbol{\omega} \in \mathbb{R}^n$ *be a random vector with* $E_{\boldsymbol{\omega}} = \mathbf{E}\left[\sum_{k=0}^{n-1} \boldsymbol{\omega}_k^2\right]$, $F_{\boldsymbol{\omega}} = \mathbf{E}\left[\left(\sum_{k=0}^{n-1} \boldsymbol{\omega}_k^2\right)^2\right]$;
- $\Delta\boldsymbol{A} \in \{-2, 0, 2\}^{m\times n}$ *be the sparse random matrix in* (9.27) *drawn from a random matrix ensemble with i.i.d. entries and density* $\eta = \frac{c}{mn} \leq \frac{1}{2}$.

If $\boldsymbol{\omega}$ *and* $\Delta\boldsymbol{A}$ *are independent, then for any* $\theta \in (0, 1)$

$$\mathbf{P}\left[\|\Delta\boldsymbol{A}\boldsymbol{\omega}\|_2^2 \geq 4m\eta\, E_{\boldsymbol{\omega}}\theta\right] \geq \zeta \tag{9.33}$$

with

$$\zeta = \left\{1 + (1-\theta)^{-2}\left[\left(1 + \frac{1}{m}\left(\frac{3}{2\eta} - 1\right)\right)\frac{F_{\boldsymbol{\omega}}}{E_{\boldsymbol{\omega}}^2} - 1\right]\right\}^{-1} \tag{9.34}$$

A proof is given as follows.

Proof (Lemma 9.1) Consider

$$\|\Delta\boldsymbol{A}\boldsymbol{\omega}\|_2^2 = \sum_{j=0}^{m-1}\sum_{k=0}^{n-1}\sum_{i=0}^{n-1}\Delta\boldsymbol{A}_{j,k}\Delta\boldsymbol{A}_{j,i}\boldsymbol{\omega}_k\boldsymbol{\omega}_i$$

We now derive the first and second moments of this positive random variable as follows; $\Delta\boldsymbol{A}$ is drawn from a random matrix ensemble of i.i.d. entries with mean $\mu_{\Delta\boldsymbol{A}} = 0$, variance $\sigma^2_{\Delta\boldsymbol{A}} = 4\eta$, and $\forall(j,k) \in \{0,\ldots,m-1\}\times\{0,\ldots,n-1\}, \mathbf{E}[\Delta\boldsymbol{A}^4_{j,k}] = 16\eta$. Using the independence between $\boldsymbol{\omega}$ and $\Delta\boldsymbol{A}$, and the fact that $\Delta\boldsymbol{A}$ is drawn from a random matrix ensemble with i.i.d. entries we have that the first moment

$$\mathbf{E}\left[\|\Delta\boldsymbol{A}\boldsymbol{\omega}\|_2^2\right] = \sum_{j=0}^{m-1}\sum_{k=0}^{n-1}\sum_{i=0}^{n-1}\mathbf{E}[\Delta\boldsymbol{A}_{j,k}\Delta\boldsymbol{A}_{j,i}]\mathbf{E}[\boldsymbol{\omega}_k\boldsymbol{\omega}_i]$$

$$= \sum_{j=0}^{m-1}\sum_{k=0}^{n-1}\sum_{i=0}^{n-1}\sigma^2_{\Delta\boldsymbol{A}}\delta(l,i)\mathbf{E}[\boldsymbol{\omega}_k\boldsymbol{\omega}_i] = \sum_{j=0}^{m-1}\sigma^2_{\Delta\boldsymbol{A}}\sum_{k=0}^{n-1}\mathbf{E}[\boldsymbol{\omega}_k^2] = 4m\eta E_{\boldsymbol{\omega}}$$

For the aforementioned properties of $\Delta\boldsymbol{A}$ we also have

$$\mathbf{E}[\Delta\boldsymbol{A}_{j,k}\Delta\boldsymbol{A}_{j,i}\Delta\boldsymbol{A}_{v,h}\Delta\boldsymbol{A}_{v,o}] = \begin{cases} \sigma^4_{\Delta\boldsymbol{A}}, & \begin{cases} j\neq v, k=i, h=o \\ j=v, k=i, h=o, l\neq h \\ j=v, k=h, i=o, l\neq i \\ j=v, k=o, i=h, k\neq i \end{cases} \\ \mathbf{E}[\Delta\boldsymbol{A}^4_{j,k}], & j=v, k=i=h=o \\ 0, & \text{otherwise} \end{cases} \tag{9.35}$$

illustrating the expectation of all possible 4-ples of entries of $\Delta\boldsymbol{A}$. After cumbersome but straightforward calculations that involve the substitution of (9.35) into $\mathbf{E}\left[(\|\Delta\boldsymbol{A}\boldsymbol{\omega}\|_2^2)^2\right]$ we obtain

$$\mathbf{E}\left[(\|\Delta\boldsymbol{A}\boldsymbol{\omega}\|_2^2)^2\right] = 16m\eta(\eta(m-1)F_{\boldsymbol{\omega}} + 3\eta(F_{\boldsymbol{\omega}} - G_{\boldsymbol{\omega}}) + G_{\boldsymbol{\omega}})$$

where $G_{\boldsymbol{\omega}} = \mathbf{E}\left[\sum_{k=0}^{n-1}\boldsymbol{\omega}_k^4\right]$. We are now in the position of using a one-sided version of Chebyshev's inequality for positive random variables, i.e., any random variable $z \geq 0$ verifies

$$\forall\theta\in(0,1),\ \mathbf{P}\left[z \geq \theta\mathbf{E}[z]\right] \geq \frac{(1-\theta)^2\mu_z^2}{(1-\theta)^2\mu_z^2 + \sigma_z^2} \tag{9.36}$$

By applying this inequality to $\|\Delta \boldsymbol{A}\boldsymbol{\omega}\|_2^2$ we have that, $\forall \theta \in (0,1)$,

$$\begin{aligned}\mathbf{P}\left[\|\Delta \boldsymbol{A}\boldsymbol{\omega}\|_2^2 \geq \theta \mathbf{E}[\|\Delta \boldsymbol{A}\boldsymbol{\omega}\|_2^2]\right] &\geq \left[1+(1-\theta)^{-2}\left[\frac{\mathbf{E}[(\|\Delta \boldsymbol{A}\boldsymbol{\omega}\|_2^2)^2]}{\mathbf{E}[\|\Delta \boldsymbol{A}\boldsymbol{\omega}\|_2^2]^2}-1\right]\right]^{-1} \\ &= \left[1+(1-\theta)^{-2}\left[\left(1-\frac{1}{m}\right)\frac{F_{\boldsymbol{\omega}}}{E_{\boldsymbol{\omega}}^2}+\frac{3\eta(F_{\boldsymbol{\omega}}-G_{\boldsymbol{\omega}})+G_{\boldsymbol{\omega}}}{\eta m E_{\boldsymbol{\omega}}^2}-1\right]\right]^{-1}\end{aligned}$$

which yields (9.34) by considering that when $\eta \leq \frac{1}{2}$, $3\eta(F_{\boldsymbol{\omega}}-G_{\boldsymbol{\omega}})+G_{\boldsymbol{\omega}} \leq \frac{3}{2}F_{\boldsymbol{\omega}}$. We are now in the position of proving Theorem 9.8.

Proof (Theorem 9.8) Since all decoders receive in absence of other noise sources the same measurements $\boldsymbol{y} = \boldsymbol{A}^{(1)}\boldsymbol{x}$, a second-class decoder would naively assume $\boldsymbol{y} = \boldsymbol{A}^{(0)}\hat{\boldsymbol{x}}$ with $\hat{\boldsymbol{x}}$ an approximation of $\boldsymbol{x}$ obtained by a decoder that satisfies this equality, e.g., as the naive BP in Sect. 9.4.1.1. Since $\boldsymbol{A}^{(1)} = \boldsymbol{A}^{(0)} + \Delta \boldsymbol{A}$, if we define $\Delta \boldsymbol{x} = \hat{\boldsymbol{x}} - \boldsymbol{x}$ we may write $\boldsymbol{A}^{(0)}\boldsymbol{x} + \Delta \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^{(0)}\hat{\boldsymbol{x}}$ and thus $\boldsymbol{A}^{(0)}\Delta \boldsymbol{x} = \Delta \boldsymbol{A}\boldsymbol{x}$. $\|\Delta \boldsymbol{x}\|_2^2$ can then be bounded straightforwardly as $\sigma_{\max}(\boldsymbol{A}^{(0)})^2\|\Delta \boldsymbol{x}\|_2^2 \geq \|\Delta \boldsymbol{A}\boldsymbol{x}\|_2^2$ yielding

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2^2 \geq \frac{\|\Delta \boldsymbol{A}\boldsymbol{x}\|_2^2}{\sigma_{\max}(\boldsymbol{A}^{(0)})^2} \tag{9.37}$$

By applying the probabilistic lower bound of Lemma 9.1 on $\|\Delta \boldsymbol{A}\boldsymbol{x}\|_2^2$ in (9.37), we have that $\|\Delta \boldsymbol{A}\boldsymbol{x}\|_2^2 \geq 4mn\,E_{\boldsymbol{x}}\theta$ for $\theta \in (0,1)$ and a given probability value exceeding ζ in (9.34). Plugging the r.h.s. of this inequality in (9.37) yields (9.30).
The following lemma applies to finding the asymptotic result (9.32) of Theorem 9.9.

Lemma 9.2 *Let $\mathcal{X}$ be an α-mixing random process with uniformly bounded fourth moments $\mathbf{E}[\boldsymbol{x}_k^4] \leq m_{\boldsymbol{x}}$ for some $m_{\boldsymbol{x}} > 0$. Define*

$$E_{\boldsymbol{x}} = \mathbf{E}\left[\sum_{k=0}^{n-1}\boldsymbol{x}_k^2\right] \qquad \textit{and} \qquad F_{\boldsymbol{x}} = \mathbf{E}\left[\left(\sum_{k=0}^{n-1}\boldsymbol{x}_k^2\right)^2\right].$$

If

$$W_{\boldsymbol{x}} = \lim_{n\to\infty}\frac{1}{n}E_{\boldsymbol{x}} > 0,$$

then

$$\lim_{n\to\infty}\frac{F_{\boldsymbol{x}}}{E_{\boldsymbol{x}}^2} = 1.$$

Proof (Lemma 9.2) Note first that from Jensen's inequality $F_{\boldsymbol{x}} \geq E_{\boldsymbol{x}}^2$, so $\lim_{n\to\infty} \frac{1}{n} E_{\boldsymbol{x}} > 0$ also implies that $\lim_{n\to\infty} \frac{1}{n^2} E_{\boldsymbol{x}}^2 > 0$ and $\lim_{n\to\infty} \frac{1}{n^2} F_{\boldsymbol{x}} > 0$. Since $\lim_{n\to\infty} \frac{1}{n^2} E_{\boldsymbol{x}}^2 = W_{\boldsymbol{x}}^2 > 0$ we may write

$$\lim_{n\to\infty} \frac{F_{\boldsymbol{x}}}{E_{\boldsymbol{x}}^2} = 1 + \frac{\lim_{n\to\infty} \frac{1}{n^2} F_{\boldsymbol{x}} - \frac{1}{n^2} E_{\boldsymbol{x}}^2}{W_{\boldsymbol{x}}^2} \tag{9.38}$$

and observe that

$$\left| \frac{1}{n^2} F_{\boldsymbol{x}} - \frac{1}{n^2} E_{\boldsymbol{x}}^2 \right| \leq \frac{1}{n^2} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} |\Xi_{j,k}|$$

where

$$\Xi_{j,k} = \mathbf{E}[\boldsymbol{x}_k^2 \boldsymbol{x}_k^2] - \mathbf{E}[\boldsymbol{x}_k^2]\mathbf{E}[\boldsymbol{x}_k^2] = \mathbf{E}[(\boldsymbol{x}_k^2 - \mathbf{E}[\boldsymbol{x}_k^2])(\boldsymbol{x}_k^2 - \mathbf{E}[\boldsymbol{x}_k^2])]$$

From the α-mixing assumption we know that $\left|\Xi_{j,k}\right| \leq \alpha(|j-l|) \leq m_{\boldsymbol{x}}$ with the sequence $\alpha(h)$ vanishing to 0 as $h \to \infty$. Hence,

$$\begin{aligned} \left| \frac{1}{n^2} F_{\boldsymbol{x}} - \frac{1}{n^2} E_{\boldsymbol{x}}^2 \right| &\leq \frac{1}{n^2} \sum_{j=0}^{n-1} \left|\Xi_{j,j}\right| + \frac{2}{n^2} \sum_{h=1}^{n-1} \sum_{j=0}^{n-h-1} \left|\Xi_{j,j+h}\right| \\ &\leq \frac{n\, m_{\boldsymbol{x}}}{n^2} + \frac{2}{n^2} \sum_{h=1}^{n-1} (n-h)\alpha(h) \leq \frac{m_{\boldsymbol{x}}}{n} + \frac{2}{n} \sum_{h=1}^{n-1} \alpha(h) \end{aligned}$$

The thesis of this lemma follows from the fact that the above upper bound vanishes to 0 as $n \to \infty$. This is obvious when $\sum_{h=0}^{+\infty} \alpha(h)$ is convergent. Otherwise, if $\sum_{h=0}^{+\infty} \alpha(h)$ is divergent we may resort to the Stolz–Cesàro theorem to find $\lim_{n\to\infty} \frac{1}{n} \sum_{h=1}^{n-1} \alpha(h) = \lim_{n\to\infty} \alpha(n) = 0$.
We are now in the position of proving Theorem 9.9, that is a mere extension of the proof of Theorem 9.8 to the asymptotic case.

Proof (Theorem 9.9) The inequality (9.37) in the proof of Theorem 9.8 is now modified for the asymptotic case, i.e., for a random process $\boldsymbol{\mathcal{X}}$. Note that $\boldsymbol{A}^{(0)}$ is drawn from the RAE($\boldsymbol{I}$) with zero-mean, unit-variance entries; thus, when $m, n \to \infty$ with $\frac{m}{n} \to q$ the value $\sqrt{n}\,\sigma_{\max}(\boldsymbol{A}^{(0)})$ is known from [22] since all its singular values belong to the interval $\left[1 - \frac{1}{\sqrt{q}}, 1 + \frac{1}{\sqrt{q}}\right]$. We therefore assume $\sigma_{\max}(\boldsymbol{A}^{(0)}) \simeq \sqrt{m} + \sqrt{n}$ and take the limit of (9.37) normalized by $\frac{1}{n}$ for $m, n \to \infty$, i.e., the recovery error power

$$W_{\hat{\boldsymbol{x}}-\boldsymbol{x}} = \lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} (\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)^2 \geq \lim_{m,n\to\infty} \frac{\left\| \Delta \boldsymbol{A} \frac{\boldsymbol{x}^{(n)}}{\sqrt{n}} \right\|_2^2}{\left(\sqrt{m} + \sqrt{n}\right)^2} \tag{9.39}$$

with $\boldsymbol{x}^{(n)}$ the n-th finite-length term in a plaintext $\boldsymbol{x} = \{\boldsymbol{x}^{(n)}\}_{n=0}^{+\infty}$ of $\boldsymbol{\mathcal{X}}$. We may now apply Lemma 9.1 in $\boldsymbol{\omega} = \frac{\boldsymbol{x}^{(n)}}{\sqrt{n}}$ for each $\|\Delta \boldsymbol{A}\boldsymbol{\omega}\|_2^2$ at the numerator of the r.h.s. of (9.39) with $F_{\boldsymbol{\omega}} = \frac{1}{n^2} F_{\boldsymbol{x}}$, $E_{\boldsymbol{\omega}} = \frac{1}{n} E_{\boldsymbol{x}}$, and $E_{\boldsymbol{x}}, F_{\boldsymbol{x}}$ as in Lemma 9.2. For $m, n \to \infty$ and $\eta \leq \frac{1}{2}$, the probability in (9.34) becomes

$$\forall \theta \in (0,1),\ \lim_{m,n\to\infty} \zeta = \left[1 + (1-\theta)^{-2} \left[\lim_{n\to\infty} \frac{\frac{1}{n^2} F_{\boldsymbol{x}}}{\frac{1}{n^2} E_{\boldsymbol{x}}^2} - 1\right]\right]^{-1}$$

Since $\boldsymbol{\mathcal{X}}$ also satisfies by hypothesis the assumptions of Lemma 9.2, we have that

$$\lim_{n\to\infty} \frac{F_{\boldsymbol{\omega}}}{E_{\boldsymbol{\omega}}^2} = 1$$

and thus $\lim_{m,n\to\infty} \zeta = 1$. Hence, with $\frac{m}{n} \to q$ and probability 1 the r.h.s. of (9.39) becomes

$$\forall \theta \in (0,1),\ \lim_{m,n\to\infty} \frac{\|\Delta \boldsymbol{A}\boldsymbol{\omega}\|_2^2}{n(1+\sqrt{\frac{m}{n}})^2} = \lim_{m,n\to\infty} \frac{4m\eta E_{\boldsymbol{x}}}{n^2(1+\sqrt{\frac{m}{n}})^2}\theta$$

and the recovery error power is shown to satisfy (9.32).
Thus, Theorems 9.8 and 9.9 were shown to hold in the respective cases.

An Upper Bound for the Second-Class Recovery Error

The second-class recovery error norm is substantially bounded from above by direct application of Theorem 9.7. To apply it, we have to compute $\epsilon_{\boldsymbol{A}^{(1)}}^{(\kappa)}, \epsilon_{\boldsymbol{A}^{(1)}}^{(2\kappa)}$ in our particular case. Theoretical results exist for estimating their value by bounding the extreme singular values in (9.16), since both $\boldsymbol{A}^{(1)}$ and $\Delta \boldsymbol{A}$ are drawn from i.i.d. random matrix ensembles.

To estimate $\epsilon_{\boldsymbol{A}^{(1)}}^{(\kappa)}$ we may proceed in the following fashion: since $\Delta \boldsymbol{A}$ is drawn from a random matrix ensemble with i.i.d. zero-mean entries for which

$$\forall (j,k) \in \{0,\ldots,m-1\} \times \{0,\ldots,n-1\},\ \mathbf{E}[\Delta \boldsymbol{A}_{j,k}^2] = 4\eta, \mathbf{E}[\Delta \boldsymbol{A}_{j,k}^4] = 16\eta \tag{9.40}$$

we may use [33, Theorem 2] to find

$$\mathbf{E}[\sigma_{\max}^{(\kappa)}(\Delta \boldsymbol{A})] = 2\underline{c}' \left(\sqrt{\kappa\eta} + \sqrt{m\eta} + (m\kappa\eta)^{\frac{1}{4}}\right) \tag{9.41}$$

for $\underline{c}' > 0$ a universal constant. Then, using the non-asymptotic estimate given in [55, Theorem 5.39], we may assume $\sigma_{\max}^{(\kappa)}(\boldsymbol{A}^{(1)}) = \underline{c}''(\sqrt{\kappa} + \sqrt{m})$ for another universal constant $\underline{c}'' > 0$. Thus, we have

$$\epsilon_{\boldsymbol{A}^{(1)}}^{(\kappa)} \simeq 2\underline{c}\frac{\sqrt{\kappa\eta} + \sqrt{m\eta} + (m\kappa\eta)^{\frac{1}{4}}}{\sqrt{\kappa} + \sqrt{m}} \tag{9.42}$$

for $\underline{c} = \frac{\underline{c}'}{\underline{c}''} > 0$ a universal constant, where the approximation is due to the fact that (9.41) actually yields an expectation of the maximum. However, this estimate is easily applicable only when $\boldsymbol{D}$ is the canonical basis; since in many practical cases this does not hold, we simply resort to a Montecarlo simulation of $\epsilon_{\boldsymbol{A}^{(1)}}^{(\kappa)}$ with, e.g., $\boldsymbol{D}$ a random orthonormal basis. As an example of such a numerical analysis, we calculate (9.16) for 10^4 instances of sub-matrices of $\boldsymbol{A}^{(1)}$ and $\Delta\boldsymbol{A}$ with $m = 512, \kappa = 1, 4, 16$ and $\eta \in [5 \cdot 10^{-4}, 10^{-2}]$. This allows us to find typical values of $\epsilon_{\boldsymbol{A}^{(1)}}^{(\kappa)}$ as reported in Fig. 9.15a. In this test case, we have found that $\underline{c} \approx 0.5741$ in (9.42) would match the simulations. In the same setting $\epsilon_{\boldsymbol{A}^{(1)}}^{(\kappa)} < 2^{\frac{1}{4}} - 1$ only when $\eta \leq 8 \cdot 10^{-3}$. In Fig. 9.15b we report the corresponding range of allowed constants $\delta^{(2\kappa)} \leq \delta_{\max}^{(2\kappa)}$ that comply with Theorem 9.7, i.e., the RIP constraints of the encoding matrices must be met so that (9.17) holds.

Once again, RIP-based analyses provide very strong sufficient conditions for signal recovery, which in our case result in establishing a formal upper bound for a small range of η. As observed by the very authors of [24], typical recovery errors are substantially smaller than this upper bound. We will therefore rely on another less rigorous, yet practically effective least-squares approach using the same hypotheses of Theorem 9.8 to bound the average recovery quality performances, as presented in the following section.

Average Signal-to-Noise Ratio Bounds

We have already discussed how the perturbation density η is the main design parameter for the proposed multiclass encryption by CS, and have presented in Sect. 9.4.1 a method to predict the average recovery performances under a variety of perturbations, including the sparse sign-flipping which is at the heart of our encryption scheme. To provide *criteria* for the choice of η we adopt two ARSNR bounds derived as follows.

The Lower Bound Although rigorous, the second- (or lower-) class recovery error upper bound derived by applying Theorem 9.7 is only compatible with small values of (κ, η), as shown by the evidence gathered in Fig. 9.15. To bound the typical recovery performances in a larger range we follow a method similar to the one used in Sect. 9.4.1.3, i.e., we analyze the behavior of a lower-class decoder that naively recovers $\hat{\boldsymbol{x}}$ such that $\boldsymbol{y} = \boldsymbol{A}^{(0)}\hat{\boldsymbol{x}} = (\boldsymbol{A}^{(0)} + \Delta\boldsymbol{A})\boldsymbol{x}$ and $\boldsymbol{A}^{(0)}(\hat{\boldsymbol{x}} - \boldsymbol{x}) = \Delta\boldsymbol{A}\boldsymbol{x}$. In most cases, such a recovery produces $\hat{\boldsymbol{x}}$ lying close to $\boldsymbol{x}$, so we approximate $\hat{\boldsymbol{x}} - \boldsymbol{x} = (\boldsymbol{A}^{(0)})^{+}\Delta\boldsymbol{A}\boldsymbol{x}$, i.e.,

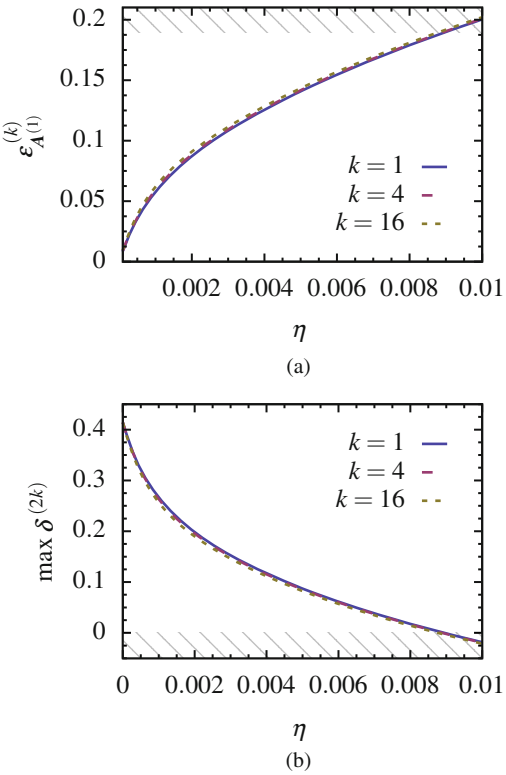


Fig. 9.15 Empirical evaluation of the constants in Theorem 9.7 based on a large number of $\boldsymbol{A}^{(1)}$, $\Delta\boldsymbol{A}$ with $m = 512, \eta \in [5{\cdot}10^{-4}, 10^{-2}]$ and $\boldsymbol{D}$ a random orthonormal basis. The forbidden areas in the statement of Theorem 9.7 are marked with stripes. (**a**) Empirical values of $\epsilon^{(\kappa)}_{\boldsymbol{A}^{(1)}}$. (**b**) Maximum allowed values of $\delta^{(2\kappa)}$

$$\frac{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2^2}{\|\boldsymbol{x}\|_2^2} \leq \sigma_{\max}((\boldsymbol{A}^{(0)})^+ \Delta\boldsymbol{A})^2$$

By taking an empirical expectation on both sides, our criterion becomes $\texttt{ARSNR} > \mathrm{LB}(m, n, \eta)$ where

$$\mathrm{LB}(m, n, \eta) = -10 \log_{10} \hat{\mathbf{E}} \left(\sigma_{\max}((\boldsymbol{A}^{(0)})^+ \Delta\boldsymbol{A})^2 \right) \text{ dB} \tag{9.43}$$

(9.43) is then calculated by a thorough Montecarlo simulation of $\sigma_{\max}((\boldsymbol{A}^{(0)})^+ \Delta\boldsymbol{A})$.

The Upper Bound The opposite criterion is found by assuming $\texttt{ARSNR} < \mathrm{UB}(m, n, \eta)$ where

$$\mathrm{UB}(m, n, \eta) = -10 \log_{10} \frac{4\eta m}{(\sqrt{m} + \sqrt{n})^2} \text{ dB} \tag{9.44}$$

that is obtained from a simple rearrangement of (9.32) with $\theta \simeq 1$. We will see how (9.43) and (9.44) fit well the ARSNR performances of the examples, and enable

a sufficiently reliable estimate of the range of performances of lower-class receivers from a given configuration of (m, n, η)

9.4.4 Application Examples

In this section we detail some example applications for the multiclass CS scheme we proposed. For each exemplary case, we study the recovery quality attained by first-class receivers against second-class ones in a two-class scheme; these results encompass the multiclass setting since high-class receivers will correspond to first-class recovery performances (i.e., at a perturbation density $\eta = 0$), while lower-class users will attain the performances of a second-class receiver at a fixed $\eta > 0$.

9.4.4.1 Experimental Framework

For each plaintext $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$ being reconstructed and each approximation $\hat{\boldsymbol{x}} = D\hat{s}$, we evaluate once again the ARSNR of (2.4); this average performance index is compared against (9.43) and (9.44) with the purpose of choosing a suitable perturbation density η so that lower-class recovery performances are set to the desired quality level. In particular, each example reports (9.43) obtained by a Montecarlo simulation of the singular values of $(\boldsymbol{A}^{(0)})^{+}\boldsymbol{\Delta A}$ over $5 \cdot 10^3$ cases.

Since our emphasis is on showing that, despite its simplicity, this method is effective in avoiding the access to high-quality information content for lower-class receivers, we complement the ARSNR evidence of each example with an automated assessment of the information content intelligible from $\hat{\boldsymbol{x}}$ by means of feature-extraction algorithms. These are equivalent to partially informed attacks to the encryption, attempting to expose the sensitive content inferred from the recovered signal. More specifically, we will try to recover an English sentence from a speech segment, the location of the PQRST peaks in an ECG, and printed text in an image.

9.4.4.2 Recovery Algorithms

While we have widely discussed the use of BP and BPDN in this book, and in particular w.r.t. their sensitivity to matrix perturbations, these convex problems are often replaced in practice by a variety of high-performance algorithms. In detail, probabilistic inference algorithms such as those in [17, 50] are capable of solving essentially the same problem as BPDN with statistical priors on the nature of the additive noise affecting the measurements. Thus, they are particularly well-fit to our application if we want to assess the best achievable performances of lower-class decoders. For completeness, as reference cases for most common algorithmic classes we preliminarily tested the solution of BPDN as implemented in SPGL1; this was compared to the greedy algorithm CoSaMP [43] and the GAMP algorithm [50].

To optimize these preliminary tests, the algorithms were optimally tuned in a "genie" fashion: BPDN was solved as BPDN($\boldsymbol{y}, \boldsymbol{A}^{(0)}\boldsymbol{D}, \varepsilon^\star$), i.e., as if $\varepsilon^\star = \|\Delta\boldsymbol{A}\boldsymbol{x}\|_2$ was known beforehand; CoSaMP was initialized with the exact sparsity level κ for each case; GAMP was run with the sparsity-enforcing, i.i.d. Bernoulli–Gaussian prior (see, e.g., [56]) and initialized with the exact sparsity ratio $\frac{\kappa}{n}$ of each instance, and the exact mean and variance of each considered test set. Moreover, signal-independent parameters were hand-tuned in each case to yield optimal recovery performances.

For the sake of brevity, in each example we select and report the algorithm that yields the most accurate recovery quality at a lower-class decoder as the amount of perturbation varies. We found that GAMP achieves the highest ARSNR in all the settings explored in the examples, consistently with the observations in [56] that assess the robust recovery capabilities of this algorithm under a broadly applicable sparsity-enforcing prior. Moreover, as $\Delta\boldsymbol{A}$ verifies [47, Proposition 2.1] the perturbation noise $\boldsymbol{\epsilon}$ is approximately Gaussian for large (m, n) and thus GAMP tuned as above yields the optimal performances as expected.

Note that recovery algorithms which attempt to jointly identify $\boldsymbol{x}$ and $\Delta\boldsymbol{A}$ [47, 60] can be seen as explicit attacks to multiclass encryption and are evaluated in more detail in Sect. 9.4.6.3, anticipating that their performances are compatible with those of GAMP.

9.4.4.3 Speech Signals

We consider a subset of spoken English sentences from the PTDB-TUG database [48] with original sampling frequency $f_s = 48\,\text{kHz}$, variable duration, and sentence length. Each speech signal is divided into segments of $n = 512$ samples and encoded by two-class CS with $m = \frac{n}{2}$ measurements. We obtain the sparsity basis $\boldsymbol{D}$ by applying principal component analysis to 500 n-dimensional segments yielding an orthonormal basis. The encoding matrix $\boldsymbol{A}^{(1)}$ is generated from $\boldsymbol{A}^{(0)} \sim \text{RAE}(\boldsymbol{I})$, by adding to the latter a sparse sign-flipping perturbation $\Delta\boldsymbol{A}$ chosen as in (9.27) with density η. The encoding in (9.29) is simulated in a realistic setting, where each window $\boldsymbol{x}$ of n samples is acquired with a different instance of $\boldsymbol{A}^{(1)}$ yielding m measurements per speech segment. As for the decoding stage, we apply GAMP as specified above to recover $\hat{\boldsymbol{x}}$ given $\boldsymbol{A}^{(1)}$ (first-class) and $\boldsymbol{A}^{(0)}$ (second-class).

For a given encoding matrix a first-class receiver is capable of decoding a clean speech signal with $\texttt{ARSNR} = 38.76\,\text{dB}$, whereas a second-class receiver is subject to significant ARSNR degradation when η increases, as shown in Fig. 9.16a. Note that while the $\texttt{RSNR}_{\hat{\boldsymbol{x}},\boldsymbol{x}}$ for $\eta = 0$ has a relative deviation of 2.14 dB around its mean (i.e., the ARSNR), as η increases the observed relative deviation is less than 0.72 dB due to the perturbation becoming the dominant effect in limiting the recovery quality w.r.t. the fact that $\boldsymbol{x}$ are compressible, but not κ-sparse. Note how the ARSNR values lie in the highlighted range between (9.43), (9.44).

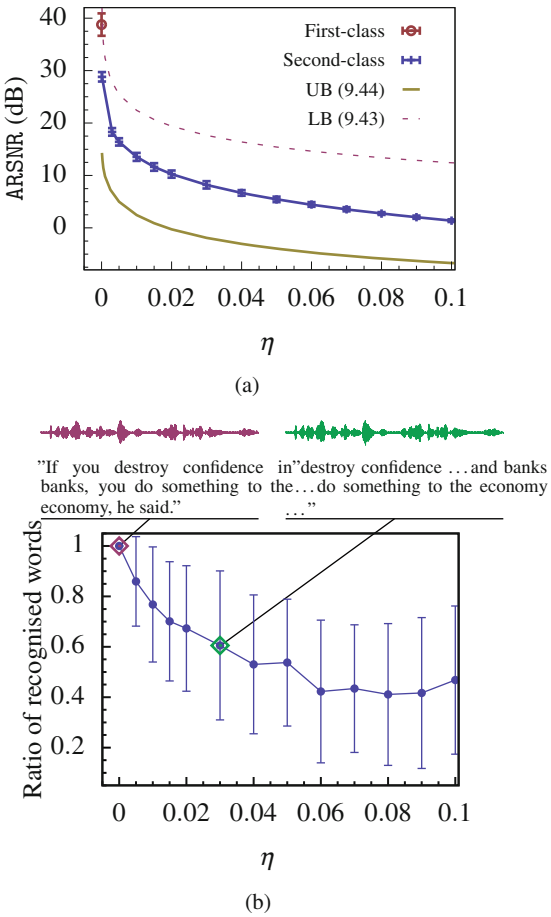


Fig. 9.16 Multiclass CS of speech signals: (**a**) ARSNR as a function of the perturbation density $\eta \in [0, 0.1]$ (*solid*) and second-class RSNR upper bound (*dashed*); (**b**) Ratio of words correctly recognized by ASR in $\eta \in [0, 0.1]$ (*bottom*) and typical recovered instances for $\eta = \{0, 0.03\}$ (*top*)

To further quantify the limited quality of attained recoveries, we process the recovered signal with the Google Web Speech interface [25, 53] which provides basic Automatic Speech Recognition (ASR). The ratio of words correctly recognized by automatic speech recognition for different values of η is reported in Fig. 9.16b; there we also depict a typical recovered signal instance, on which a first-class user (i.e., $\eta = 0$) attains $\mathtt{RSNR} = 36.58\,\mathrm{dB}$, whereas a second-class decoder only achieves $\mathtt{RSNR} = 8.42\,\mathrm{dB}$ when $\eta = 0.03$. The corresponding ratio of recognized words is $\frac{14}{14}$ against $\frac{8}{14}$. In both cases the sentence is intelligible to a human listener, yet the second-class decoder recovers a signal that is sufficiently corrupted to avoid straightforward automatic speech recognition.

9.4.4.4 ECG Signals

We now process a large subset of ECGs from the PhysioNet database [23] sampled at $f_s = 256$ Hz. In particular, we report the case of a typical 25-min ECG (sequence `e0108`) and encode windows of $n = 256$ samples by two-class CS with $m = 90$ measurements, amounting to a dataset of 1500 ECG instances. The encoding and decoding scheme is identical to that of Sect. 9.4.4.3, and we assume the Symmlet-6 orthonormal DWT [37] as the sparsity basis $\boldsymbol{D}$.

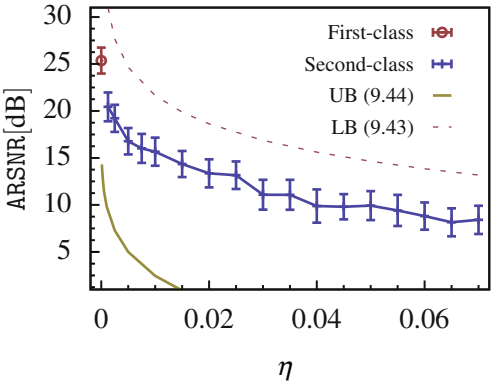
In this configuration the first-class decoder is able to reconstruct the original signal with $\texttt{ARSNR} = 25.36\,\text{dB}$, whereas a second-class decoder subject to a perturbation of density $\eta = 0.03$ achieves an $\texttt{ARSNR} = 11.08\,\text{dB}$; the recovery degradation depends on η as reported in Fig. 9.17a. As an additional quantification of the encryption at second-class decoders we apply PUWave [29], an Automatic Peak Detection algorithm (APD), to first- and second-class signal reconstructions. In more detail, PUWave is used to detect the position of the P,Q,R,S and T peaks, i.e., the sequence of pulses whose positions and amplitudes summarize the diagnostic properties of an ECG.

The application of this APD yields the estimated peak instants $\hat{t}_{\text{P,Q,R,S,T}}$ for each of $J = 1500$ reconstructed signal windows and each decoder class, which are afterwards compared to the corresponding peak instants as detected on the original signal prior to encoding. Thus, we define the average time displacement $\sigma_t = \sqrt{\frac{1}{J}\sum_{i=0}^{J-1}(\hat{t}^{(i)} - t^{(i)})^2}$ and evaluate it for t_{R} and t_{PQST}. A first-class receiver is subject to a displacement $\sigma_{t_{\text{R}}} = 0.6\,\text{ms}_{\text{rms}}$ of the R-peak and $\sigma_{t_{\text{PQST}}} = 9.8\,\text{ms}_{\text{rms}}$ of the remaining peaks w.r.t. the original signal. On the other hand, a second-class user is able to determine the R-peak with $\sigma_{t_{\text{R}}} = 4.4\,\text{ms}_{\text{rms}}$ while the displacement of the other peaks is $\sigma_{t_{\text{PQST}}} = 55.3\,\text{ms}_{\text{rms}}$. As η varies in $[0, 0.05]$ this displacement increases as depicted in Fig. 9.17b, thus confirming that a second-class user will not be able to accurately determine the position and amplitude of the peaks with the exception of the R-peak.

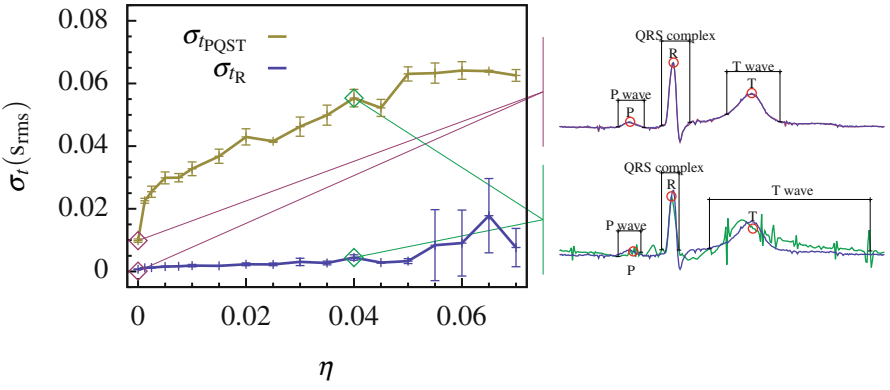
9.4.4.5 Sensitive Text in Images

In this final example we consider an image dataset of people holding printed identification text and apply multiclass CS to selectively hide this sensitive content to lower-class users. The 640×512 pixel images are encoded by CS in 10×8 blocks each of 64×64 pixel while the two-class strategy is only applied to a relevant image area of 3×4 blocks. We adopt as sparsity basis the bidimensional Daubechies-4 orthonormal DWT [37] and encode each block of $n = 4096$ pixels with $m = 2048$ measurements; two-class encoding is then applied with a sparse sign-flipping perturbation density $\eta \in [0, 0.4]$.

The `ARSNR` performances of this example are reported in Fig. 9.18a as averaged on 20 instances per case, showing a rapid degradation of the `ARSNR[dB]` as η is increased. This degradation is highlighted in the typical case of Fig. 9.18b for $\eta \in \{0.03, 0.2\}$.



(a)



(b)

Fig. 9.17 Multiclass CS of ECG signals: (**a**) ARSNR as a function of the perturbation density $\eta \in [0, 0.05]$ (*solid*) and second-class RSNR upper bound (*dashed*); (**b**) Time displacement (*left*) of the R (*solid*) and P,Q,S,T (*dashed*) peaks as evaluated by APD for $\eta \in [0, 0.05]$ with typical recovered instances (*right*) for first-class (*top*) and second-class (*bottom*) users

In order to assess the effect of our encryption method with an automatic information extraction algorithm, we have applied Tesseract [54], an optical character recognition algorithm (OCR), to the images reconstructed by a second-class user. The text portion in the recovered image data is preprocessed to enhance their quality prior to OCR: the images are first rotated, then we apply standard median filtering to reduce the high-pass noise components. Finally, contrast adjustment and thresholding yield the two-level image which is processed by Tesseract. To assess the attained OCR quality we have measured the average number of correctly

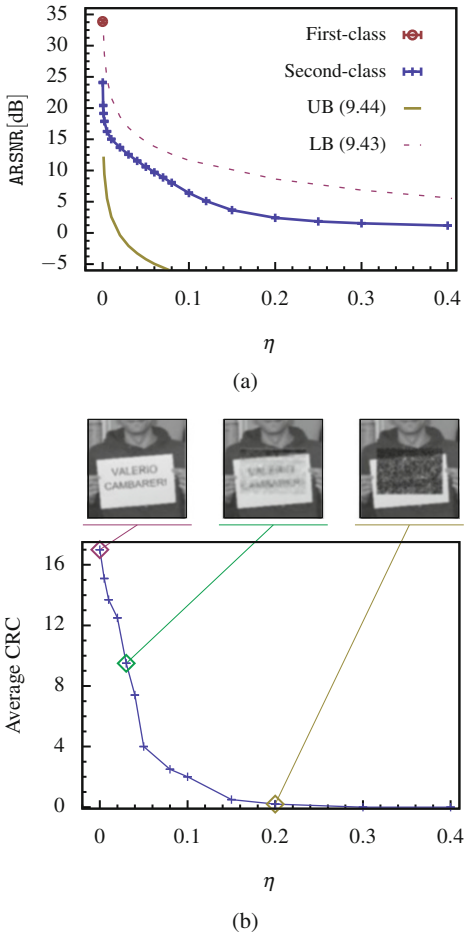


Fig. 9.18 Multiclass CS of images: (**a**) ARSNR as a function of the perturbation density $\eta \in [0, 0.4]$ (*solid*) and second-class RSNR upper bound (*dashed*); (**b**) Average CRC by OCR for $\eta \in [0, 0.4]$ (*bottom*) and typical recovered instances for $\eta \in \{0, 0.03, 0.2\}$ (*top*)

recovered characters (CRC) from the decoded text image. In Fig. 9.18b the average CRC is reported as a function of η: as the perturbation density increases the OCR fails to recognize an increasing number of ordered characters, i.e., a second-class user progressively fails to extract text content from the decoded image.

9.4.5 Resilience Against Known-Plaintext Attacks

We now focus on how robust is our two- to multi-class encryption scheme to malicious attempts at the recovery of the exact encoding matrix by a partially informed user.

9.4.5.1 Preliminary Considerations

Still in computational security terms, since missing information on the encoding matrices might be treated as a perturbation matrix, we attempt an additional computational attack specifically targeted to a multiclass scheme and attempting to nullify its effect. This form of attack is carried out by a second-class user that attempts to upgrade its knowledge by using signal recovery algorithms specifically accounting for encoding matrix uncertainty [47, 60]. As expected from the random nature of the SSF perturbation introduced in Sect. 9.4.1.4 the results will, however, show no practical improvement w.r.t. the bounds and performances illustrated in Sect. 9.4.4.

Practical computational attacks are then exemplified by applying CS as an encryption scheme to the same signal classes of Sect. 9.4.4, showing how the extracted information on the true encoding matrix from a plaintext–ciphertext pair leads to no significant signal recovery quality increase. This theoretical and empirical evidence clarifies that, although not perfectly secure, both standard CS and multiclass encryption based on it feature a noteworthy level of security against KPAs, thus increasing its appeal as a zero-cost encryption method for resource-limited sensor nodes.

Contextualizing the above findings to multiclass encryption by CS, we have shown how a malicious eavesdropper attempting to break the encoding by means of a straightforward statistical analysis of $\boldsymbol{y}$ is effectively presented with Gaussian-distributed ciphertexts when the encoding matrix is drawn from an i.i.d. sub-Gaussian random matrix ensemble.

In addition, one could consider the threat of a malicious second-class user attempting to upgrade itself to the knowledge of the true encoding matrix $\boldsymbol{A}^{(1)}$ given $\boldsymbol{A}^{(0)}$. Letting $\boldsymbol{A}^{(0)}, \boldsymbol{A}^{(1)}$ be drawn from RAE($\boldsymbol{I}$) encoding matrix ensembles, in the worst-case we may also assume that this attacker has access to $\boldsymbol{\epsilon} = \Delta\boldsymbol{A}\boldsymbol{x}$, and is able to compute $f(\boldsymbol{\epsilon})$ for a statistical cryptanalysis. Clearly, this will depend on the density of $\Delta\boldsymbol{A} = \boldsymbol{A}^{(1)} - \boldsymbol{A}^{(0)}$, that is a sparse sign-flipping drawn from a random matrix ensemble with i.i.d. entries. Informally and intuitively, this will result in $f(\boldsymbol{\epsilon}|\boldsymbol{x}) \longrightarrow_{\text{dist.}} \mathrm{N}(\boldsymbol{0}_m, \mathscr{C}_{\boldsymbol{\epsilon}})$ where $\mathscr{C}_{\boldsymbol{\epsilon}} = \sigma^2_{\Delta\boldsymbol{A}} E_{\boldsymbol{x}} \boldsymbol{I}_m$ where $\sigma^2_{\Delta\boldsymbol{A}} = 4\eta$ and $E_{\boldsymbol{x}} = \|\boldsymbol{x}\|_2^2$, i.e., the information that leaks to a malicious second-class user is the sparse sign-flipping density η as well as the energy of the plaintext. A more thorough verification can be derived by application of the procedures detailed in this section. Hence, the ciphertext is statistically indistinguishable from the one that could be produced by encoding the same plaintext with $\boldsymbol{A}^{(0)}$ instead of $\boldsymbol{A}^{(1)}$, and such second-class users will be unable to exploit the statistical properties of $\boldsymbol{y}$ to upgrade their encoding matrix to $\boldsymbol{A}^{(1)}$.

Thus, we may safely conclude that straightforward statistical attacks to multiclass encryption based on CS only extract very limited information from the ciphertext; the more threatening case of known-plaintext attacks is expanded in the next section.

9.4.5.2 Class-Upgrade Known-Plaintext Attack

A KPA may also be attempted by Steve, a malicious second-class receiver aiming to improve its signal recovery performances with the intent of reaching the same quality of a first-class receiver. In this KPA, a partially correct encoding matrix $\boldsymbol{A}^{(0)}$ that differs from $\boldsymbol{A}^{(1)}$ by c entries is also known in addition to $\boldsymbol{x}$ and $\boldsymbol{y}$. With this prior, Steve may compute $\boldsymbol{\epsilon} = \boldsymbol{y} - \boldsymbol{A}^{(0)}\boldsymbol{x} = \Delta\boldsymbol{A}\boldsymbol{x}$ where $\Delta\boldsymbol{A} = \boldsymbol{A}^{(1)} - \boldsymbol{A}^{(0)}$ here is an unknown matrix with ternary-valued entries, i.e., $\Delta\boldsymbol{A} \in \{-2, 0, 2\}^{m\times n}$. Hence, Steve performs a KPA by searching for a set of ternary symbols $\{\Delta\boldsymbol{A}_{j,k}\}_{k=0}^{n-1}$ such that each entry of $\boldsymbol{\epsilon}$,

$$\epsilon_j = \sum_{k=0}^{n-1} \Delta\boldsymbol{A}_{j,k}\boldsymbol{x}_k \tag{9.45}$$

of which it is known a priori that $\Delta\boldsymbol{A}_{j,k} \neq 0$ only in c cases. Moreover, to ease the solution of this problem and make it row-wise separable, we assume that Steve gains access to an even more accurate information, i.e., the exact number c_j of nonzero entries for each row $\Delta\boldsymbol{A}_j$ or equivalently the number of sparse sign-flippings mapping $\boldsymbol{A}_{j,\cdot}^{(0)}$ into the corresponding[9] ${\boldsymbol{A}^{(1)}}_{j,\cdot}$. By assuming this, we may prove the equivalence between Steve's KPA to each row of $\boldsymbol{A}^{(1)}$ and a slightly adjusted SSP.

Problem 9.2 (γ-Cardinality Subset-Sum Problem) Let $\{\boldsymbol{u}_k\}_{k=0}^{n-1}$, $\boldsymbol{u}_k \in \{1, \ldots, Q\}$, $\gamma \in \{1, \ldots, n\}$ and $\upsilon \in \mathbb{Z}_+$. We define *$\gamma$-cardinality* subset-sum problem (γ-SSP) the optimization problem of assigning n binary variables $\boldsymbol{b}_k \in \{0, 1\}$, $k = 0, \ldots, n-1$ so that

$$\upsilon = \sum_{k=0}^{n-1} \boldsymbol{b}_k\boldsymbol{u}_k \tag{9.46}$$

$$\gamma = \sum_{k=0}^{n-1} \boldsymbol{b}_k \tag{9.47}$$

We define *solution* any $\{\boldsymbol{b}_k\}_{k=0}^{n-1}$ verifying (9.46) and (9.47).
Again, a mapping of Steve's KPA to Problem 9.2 is easily obtained.

Theorem 9.10 (Steve's Known-Plaintext Attack) *The KPA to ${\boldsymbol{A}^{(1)}}_{j,\cdot}$ given $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{A}^{(0)}$, and c_j is equivalent to a γ-SSP where $\gamma = c_j$, $Q = 2L$, $\boldsymbol{u}_k = -\boldsymbol{A}_{j,k}^{(0)}\boldsymbol{x}_k + L$, the variables*

$$\boldsymbol{b}_k = \frac{1}{2}\left(1 - \frac{\hat{\boldsymbol{A}}_{j,k}^{(1)}}{\boldsymbol{A}_{j,k}^{(0)}}\right)$$

[9] Clearly, the total number of nonzero entries in $\Delta\boldsymbol{A}$ is $c = \sum_{j=0}^{m-1} c_j$.

and the sum

$$\upsilon = \frac{1}{2}\epsilon_j + L\,c_j$$

This SSP has a true solution $\{\bar{\boldsymbol{b}}_k\}_{k=0}^{n-1}$ *that is mapped to the row* $\boldsymbol{A}^{(1)}{}_{j,\cdot}$*, and other* candidate solutions *that verify* (9.46) *and* (9.47) *but correspond to matrix rows* $(\hat{\boldsymbol{A}})_j \neq \boldsymbol{A}^{(1)}{}_{j,\cdot}$.

We also define $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{A}^{(0)}_{j,\cdot}, \boldsymbol{A}^{(1)}{}_{j,\cdot})$ a *problem instance*; Steve can therefore use the result of (9.46) to obtain the perturbation entries $\Delta\boldsymbol{A}_{j,k} = -2\boldsymbol{A}^{(0)}_{j,k}\boldsymbol{b}_k$. The derivation of Theorem 9.10 is obtained as follows.

Proof (Theorem 9.10) In this case the attacker knows $(\boldsymbol{A}^{(0)}, \boldsymbol{x}, \boldsymbol{y})$, and is able to calculate $\boldsymbol{\epsilon} = \boldsymbol{y} - \boldsymbol{A}^{(0)}\boldsymbol{x}$, i.e., $\epsilon_j = \boldsymbol{y}_j - \sum_{k=0}^{n-1}\boldsymbol{A}^{(0)}_{j,k}\boldsymbol{x}_k = \sum_{k=0}^{n-1}\Delta\boldsymbol{A}_{j,k}\boldsymbol{x}_k$ where all the entries $\Delta\boldsymbol{A}_{j,k}$ are unknown. For the j-th row, the attacker also knows there are c_j non-zero elements in $\Delta\boldsymbol{A}_{j,k} = -2\boldsymbol{A}^{(0)}_{j,k}\boldsymbol{b}_k$ with $\boldsymbol{b}_k \in \{0, 1\}$ binary variables that are 1 if the flipping occurred and 0 otherwise. Note that from the above information $c_j = \sum_{k=0}^{n-1}\boldsymbol{b}_k$. With this we define a set of even weights $D_k = -2\boldsymbol{A}^{(0)}_{j,k}\boldsymbol{x}_k$, i.e., $D_k \in \{-2L, \ldots, -2, 0, 2, \ldots, 2L\}$ so the KPA is defined by satisfying the equalities

$$\epsilon_j = \sum_{k=0}^{n-1} D_k\boldsymbol{b}_k \tag{9.48}$$

$$c_j = \sum_{k=0}^{n-1}\boldsymbol{b}_k \tag{9.49}$$

To obtain a standard γ-SSP with positive weights and $\gamma = c_j$ we sum $2L$ to all D_k so (9.48) becomes $\epsilon_j + 2L\sum_{k=0}^{n-1}\boldsymbol{b}_k = \sum_{k=0}^{n-1}(D_k + 2L)\boldsymbol{b}_k$. Multiplying both sides by $\frac{1}{2}$ and using (9.49) yields $\upsilon = \frac{1}{2}\epsilon_j + Lc_j = \sum_{k=0}^{n-1}\boldsymbol{u}_k\boldsymbol{b}_k$ where $\boldsymbol{u}_k = -\boldsymbol{A}^{(0)}_{j,k}\boldsymbol{x}_k + L \in \{0, \ldots, Q\}$. $Q = 2L$. Finally, we note the exclusion of $\boldsymbol{u}_k = 0$ to facilitate the attack.

In the following, we will denote with $r = \frac{c_j}{n}$ the row-density of perturbations. Since in [51] the γ-cardinality SSP case is obtained as an extension of the results on the unconstrained SSP, we obtain the following theorem.

Theorem 9.11 (Expected Number of Solutions to Steve's Known-Plaintext Attack) *For large n, the expected number of candidate solutions of the KPA in Theorem 9.10, in which (i) all the coefficients* $\{\boldsymbol{u}_k\}_{k=0}^{n-1}$ *are i.i.d. uniformly drawn from* $\{1, \ldots, 2L\}$*, and (ii) the true solution* $\{\bar{\boldsymbol{b}}_k\}_{k=0}^{n-1}$ *is drawn with equiprobable independent binary values, is*

$$\mathcal{S}_{\text{Steve}}(n, L, r) \simeq \sqrt{\frac{3}{2}}\frac{r^{-1-nr}\,(1-r)^{-1-n(1-r)}}{2\pi nL} \tag{9.50}$$

The proof of Theorem 9.11 is reported below. The number of candidate solutions found by Steve's KPA is by many orders of magnitude smaller than Eve's KPA, the reason being that Steve requires much less information to achieve complete knowledge of the true encoding $\boldsymbol{A}^{(1)}$. In order to provide numerical evidence, we simulate Steve's KPA on a set of 50 randomly generated problem instances with row-density of perturbations $r = \left\{\frac{5}{n}, \frac{10}{n}, \frac{15}{n}\right\}$ for $n = \{20, \ldots, 32\}$ and $L = 5 \cdot 10^3$; the problem is still formulated as binary programming in CPLEX, albeit with the additional equality constraint (9.49); the full solution pool can still be populated for the given dimensions.[10]

The empirical average number of solutions $\hat{\mathcal{S}}_{\text{Steve}}(n, L, r)$ reported in Fig. 9.19 is well predicted by the theoretical value in (9.50); note that this approximation is increasingly accurate for large n. Moreover, by resuming the previous example our $n = 64 \times 64$ pixel gray-scale image quantized at $\mathfrak{b}_{\boldsymbol{x}} = 8$ bit and encoded with two-class CS using $\Delta\boldsymbol{A}$ with $r = 0.03$ will have on-average $6.25 \cdot 10^{234}$ candidate solutions of indistinguishable quality.

The previous analysis hinges on a counting argument in a general setting, without any other prior assumption on the structure of $\boldsymbol{A}^{(1)}$ or $\Delta\boldsymbol{A}$. This class-upgrade KPA has been examined by assuming very accurate prior information on the number of perturbations per row, thus implying a best-case situation for the attacker. As we will show in the experiments of Sect. 9.4.6, these attacks yield no advantage in terms of recovery performances to unintended receivers.

Let us now prove Theorem 9.11; the proof draws again from the work of Sasamoto et al. [51] and is therefore similar in principle to that of Theorem 9.5, i.e., it is merely an interface to existing results on the γ-SSP. It is worth noting that the proof draws on Definition 9.3.

Proof (Theorem 9.11) Assume $F_p(a, b)$ and $G_p(a, b)$ as in (9.11), (9.12). Define the normalized constraint $r = \frac{cj}{n}$ and two quantities $a(\tau, r)$ and $b(\tau, r)$ that are the solutions of the following system of equalities

$$r = F_0(a, b)$$
$$\tau = F_1(a, b)$$

that are, respectively, equivalent to [51, (5.3-4)]. We also define

$$G(\tau, r) = \begin{bmatrix} G_0(a(\tau, r), b(\tau, r)) & G_1(a(\tau, r), b(\tau, r) \\ G_1(a(\tau, r), b(\tau, r) & G_2(a(\tau, r), b(\tau, r)) \end{bmatrix}$$

[10] In the first case, full enumeration is still feasible in an acceptable computation time up to about $n = 48$.

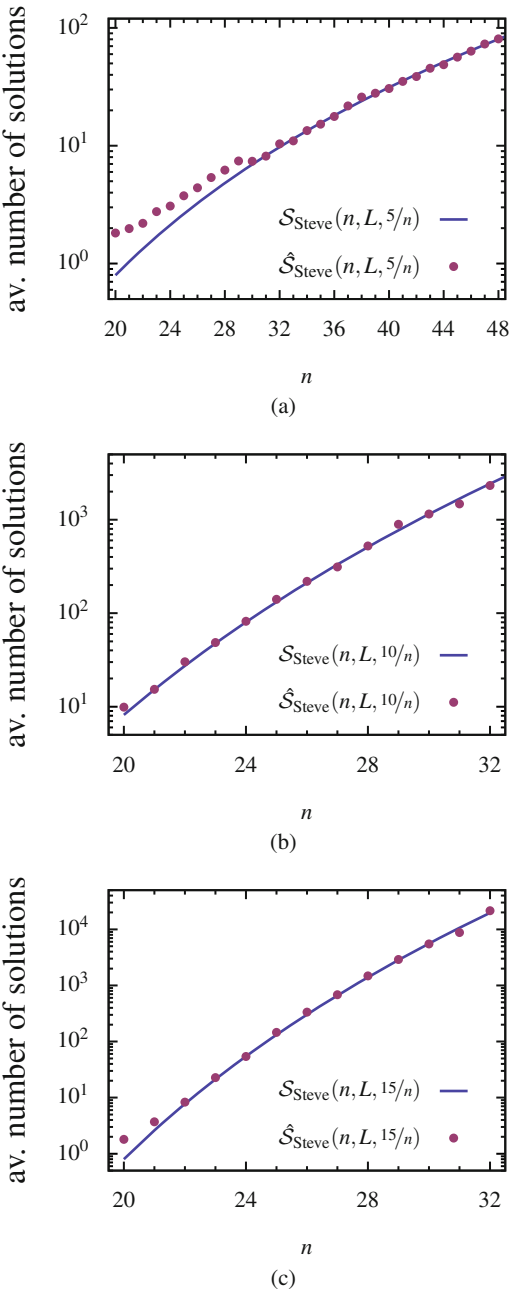


Fig. 9.19 Empirical average number of solutions for Steve's KPA compared to the theoretical approximation of (9.50) for $L = 5 \cdot 10^3$ with row-density of perturbations $r = \frac{5}{n}, \frac{10}{n}, \frac{15}{n}$

With this, [51, (5.8-9)] prove that the number of solutions of a γ-SSP with integer coefficients $\{\boldsymbol{u}_k\}_{k=0}^{n-1}$ uniformly distributed in $\{1,\dots,Q\}$, $Q=2L$, $\gamma=c_j$ is

$$\mathcal{S}_{\text{Steve}}(\tau,n,L,r)=\frac{e^{n(a(\tau,r)\tau-b(\tau,r)r)}}{4\pi nL\sqrt{\det\left(G(\tau,r)\right)}}e^{n\int_0^1\log\left[1+e^{b(\tau,r)-a(\tau,r)\xi}\right]\,\mathrm{d}\xi} \tag{9.51}$$

Using the same arguments as in the proof of Theorem 9.5, we average on τ and obtain an expression identical to (9.13) for the computation of $\mathbf{E}_\tau[\mathcal{S}_{\text{Steve}}(\tau,n,L,r)]$. Since $\mathcal{S}_{\text{Steve}}(\tau,n,L,r)$ has once again an approximately Gaussian profile in τ with a maximum in $\tau=\frac{r}{2}$ we approximate the expectation in τ,

$$\begin{aligned}\mathbf{E}_\tau[\mathcal{S}_{\text{Steve}}(\tau,n,L,r)]&\simeq\mathcal{S}_{\text{Steve}}\left(\frac{r}{2},n,L,r\right)\frac{1}{\sqrt{2}}\\&=\sqrt{\frac{3}{2}}\frac{r^{-1-n\rho}\,(1-r)^{-1-n(1-r)}}{2\pi nL}\end{aligned} \tag{9.52}$$

by using the fact that $a\left(\frac{r}{2},r\right)=0$ and $b\left(\frac{r}{2},r\right)=\log\left(\frac{r}{1-r}\right)$.

9.4.6 *Practical Attack Examples*

In this section we exemplify KPAs in a common framework which entails the following procedure. When Eve is performing a KPA as in Sect. 9.3.1, it knows a single plaintext–ciphertext pair $(\boldsymbol{x}',\boldsymbol{y}')$ and attacks a matrix $\boldsymbol{A}^{(1)}$ row-by-row; we here infer each row $\boldsymbol{A}^{(1)}_{j,\cdot}$ by generating random instances of a RAE($\boldsymbol{I}$) encoding matrix until a chosen number of candidate rows $\hat{\boldsymbol{A}}^{(1)}_{j,\cdot}$ that verify $\boldsymbol{y}'_j=\hat{\boldsymbol{A}}^{(1)}_{j,\cdot}\boldsymbol{x}'$ has been found. Thus, the inferred $\hat{\boldsymbol{A}}^{(1)}$ is actually composed by collecting the outputs of m random searches. This approach is preferable to solving Eve's KPA by means of linear programming as in Sect. 9.3.1 for two reasons.

Firstly, it is known from Theorem 9.5 that the expected number of solutions is very large and thus the probability of success of a random search is far from being negligible, while its computational cost is relatively low.

Secondly, the theoretical conditions that guarantee when $\boldsymbol{x}'$ can be retrieved from $\boldsymbol{y}'$ despite the dimensionality reduction are applicable when $\boldsymbol{A}^{(1)}$ is comprised of sensing sequences with i.i.d. antipodal symbols. On the contrary, the chosen integer programming solver explores solutions in a systematic way and, while crucial in the enumeration of *all* candidate solutions as in Sect. 9.3.1 (with computational cost growing exponentially in n), it tends to generate them in an ordered fashion. When only some of these solutions are considered (as necessary when n is large and the

number of solutions scales according to our results) this results in sets of $\hat{\boldsymbol{A}}_{j,\cdot}^{(1)}$ that could be very distant from $\boldsymbol{A}_{j,\cdot}^{(1)}$.

To test the obtained guess $\hat{\boldsymbol{A}}^{(1)}$, Eve may then *pretend* to ignore $\boldsymbol{x}'$ and recover its approximation $\hat{\boldsymbol{x}}'$ from $(\boldsymbol{y}', \hat{\boldsymbol{A}}^{(1)})$ by using a high-performance signal recovery algorithm such as GAMP [50] optimally tuned as in Sect. 9.4.6.3. This sets[11] the $\mathtt{RSNR}_{\hat{\boldsymbol{x}}',\hat{\boldsymbol{x}}}$ level which is adopted as a quality indicator for $\hat{\boldsymbol{A}}^{(1)}$. Then, Eve attempts signal recovery from a second ciphertext $\boldsymbol{y}'' = \boldsymbol{A}^{(1)}\boldsymbol{x}''$ where the plaintext $\boldsymbol{x}''$ is unknown, i.e., as if somehow $\boldsymbol{A}^{(1)}$ was reused twice. In this case, and if Eve's KPA was successful in retrieving $\hat{\boldsymbol{A}}^{(1)}$, the recovery $\hat{\boldsymbol{x}}''$ obtained by means of GAMP would yield a new $\mathtt{RSNR}_{\hat{\boldsymbol{x}}'',\boldsymbol{x}''} \approx \mathtt{RSNR}_{\hat{\boldsymbol{x}}',\boldsymbol{x}'}$. To remark what is shown below, we evaluate how the $(\mathtt{RSNR}_{\hat{\boldsymbol{x}}',\boldsymbol{x}'}, \mathtt{RSNR}_{\hat{\boldsymbol{x}}'',\boldsymbol{x}''})$ pairs are distributed w.r.t. fixed plaintexts $\boldsymbol{x}', \boldsymbol{x}''$ encoded with the same $\boldsymbol{A}^{(1)}$ and candidate solutions $\hat{\boldsymbol{A}}^{(1)}$ are considered in the decoding; if Eve is successful, an $\mathtt{RSNR}_{\hat{\boldsymbol{x}}'',\boldsymbol{x}''}$ compatible with $\mathtt{RSNR}_{\hat{\boldsymbol{x}}',\boldsymbol{x}'}$ must be observed.

The examples of class-upgrade KPAs follow the same procedure as those performed by Eve, with the exception that Steve generates the rows of $\hat{\boldsymbol{A}}^{(1)}$ by random search of the index set that maps the known $\boldsymbol{A}_{j,\cdot}^{(1)}$ to $\hat{\boldsymbol{A}}_{j,\cdot}^{(1)}$ that verifies $\boldsymbol{y}'_j = \hat{\boldsymbol{A}}_{j,\cdot}^{(1)}\boldsymbol{x}'$. Coherently with the theoretical setting of Sect. 9.4.5.2, we also assume that Steve knows that exactly c_j entries are flipped in each row. Repeating this search for m rows provides the candidate solutions $\hat{\boldsymbol{A}}^{(1)}$, of which we will study how the corresponding $(\mathtt{RSNR}_{\hat{\boldsymbol{x}}',\boldsymbol{x}'}, \mathtt{RSNR}_{\hat{\boldsymbol{x}}'',\boldsymbol{x}''})$ pairs are distributed as mentioned above.

9.4.6.1 ECG Signals

We now consider ECG signals in the same conditions of Sect. 9.4.5, focusing on two windows $\boldsymbol{x}', \boldsymbol{x}''$ of $n = 256$ samples quantized with $\mathtt{b}_{\boldsymbol{x}} = 12$ bit; these correspond to the measurement vectors $\boldsymbol{y}', \boldsymbol{y}''$ of dimensionality $m = 90$. Signal recovery is allowed by the sparsity level of the windowed signal when decomposed with $\boldsymbol{D}$ chosen as a Symmlet-6 orthonormal DWT [37].

We generate 2000 candidate solutions for both Eve and Steve's KPA that correspond to the recovery performances reported in Fig. 9.20. While both malicious users are able to reconstruct the known plaintext $\boldsymbol{x}'$ with a relatively high[12] average $\mathtt{RSNR}_{\hat{\boldsymbol{x}}',\boldsymbol{x}'} \approx 25$ dB, on the second window of samples $\boldsymbol{x}''$ the eavesdropper achieves an average $\mathtt{RSNR}_{\hat{\boldsymbol{x}}'',\boldsymbol{x}''} \approx -0.20$ dB (Fig. 9.20a), whereas the second-class decoder achieves an average $\mathtt{RSNR}_{\hat{\boldsymbol{x}}'',\boldsymbol{x}''} \approx 12.15$ dB (Fig. 9.20b) when the two-class encryption scheme is set to a sign-flipping density $\eta = \frac{c}{mn} = 0.03$ between $\boldsymbol{A}^{(1)}$ and

[11] Hereafter, we specify the $\mathtt{RSNR}_{\hat{\boldsymbol{u}},\boldsymbol{u}}$ between a signal $\boldsymbol{u}$ and its approximation $\hat{\boldsymbol{u}}$.

[12] Their KPAs indeed yield solutions to $\boldsymbol{y}' = \hat{\boldsymbol{A}}^{(1)}\boldsymbol{x}'$.

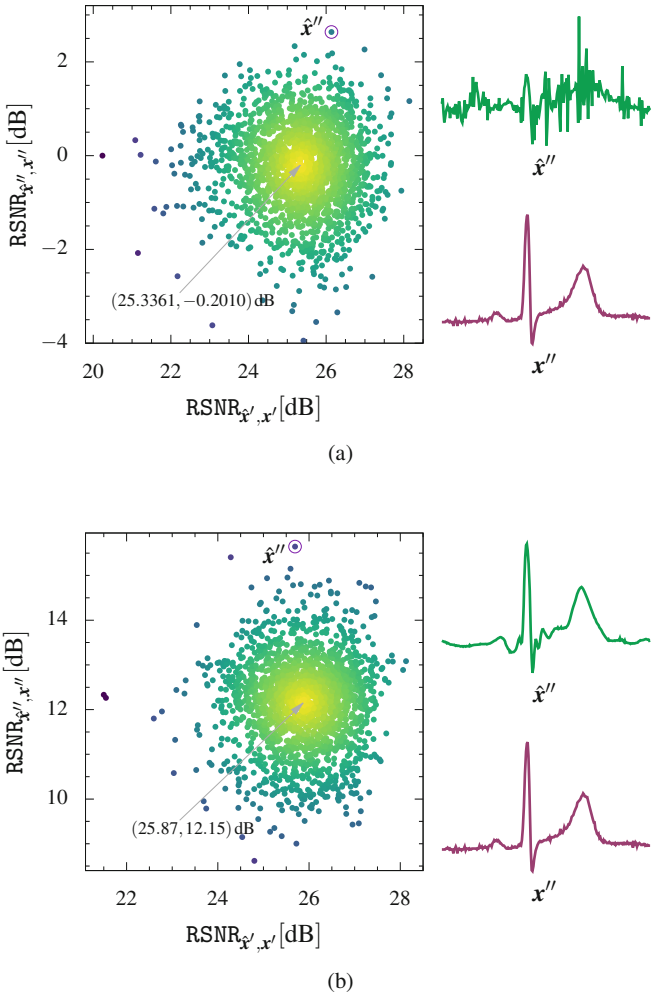


Fig. 9.20 Effectiveness of (**a**) Eve and (**b**) Steve's KPA in recovering a hidden ECG. Each point is a guess of the encoding matrix $\boldsymbol{A}^{(1)}$ whose quality is assessed by decoding the ciphertext $\boldsymbol{y}'$ corresponding to the known plaintext x' ($\mathtt{RSNR}_{\hat{x}',x'}$) and by decoding a new ciphertext $\boldsymbol{y}''$ ($\mathtt{RSNR}_{\hat{x}'',x''}$). The Euclidean distance from the average ($\mathtt{RSNR}_{\hat{x}',x'}, \mathtt{RSNR}_{\hat{x}'',x''}$) is highlighted by color gradient

$\boldsymbol{A}^{(1)}$. In this case, the nominal second-class RSNR $= 11.08$ dB when reconstructing $\boldsymbol{x}''$ from $\boldsymbol{y}''$ with $\boldsymbol{A}^{(1)}$, while the correlation coefficient between $\mathtt{RSNR}_{\hat{x}',x'}$ and $\mathtt{RSNR}_{\hat{x}'',x''}$ is 0.0140; these figures clearly highlight the ineffectiveness of KPAs at inferring $\boldsymbol{A}^{(1)}$ in this case. This is also confirmed by the perceptual quality of $\hat{\boldsymbol{x}}''$ corresponding to the maximum $\mathtt{RSNR}_{\hat{x}'',x''}$ highlighted in Fig. 9.20.

9.4.6.2 Sensitive Text in Images

In this example we consider the same test images used in Sect. 9.4.5, i.e., 640×512 pixel gray-scale images of people holding a printed identification text concealed by means of two-class encryption. To reduce the computational burden of KPAs we assume a block size of 64×64 pixel, $\mathtt{b}_x = 8$ bit per pixel, and encode the resulting $n = 4096$ pixels into $m = 2048$ measurements. Signal recovery is performed by assuming the blocks have a sparse representation on a bidimensional Daubechies-4 orthonormal DWT [37]. Two-class encryption is applied on the blocks containing printed text: we choose two adjacent blocks $\boldsymbol{x}', \boldsymbol{x}''$ containing some letters and encoded with the same $\boldsymbol{A}^{(1)}$; in this case, the second-class decoder nominally achieves $\mathtt{RSNR} = 12.57$ dB without attempting class-upgrade due to the flipping of $c = 251658$ entries (corresponding to a perturbation density $\eta = 0.03$) in the encoding matrix.

In order to test Eve and Steve's KPA we randomly generate 2000 solutions for the j-th row of the encoding given $\boldsymbol{x}', \boldsymbol{y}'$: it is worth noting that while in the previous case the signal dimensionality is sufficiently small to produce a solution set in less than two minutes, in this case generating 2000 different solutions for a single row may take up to several hours for particularly hard instances. By using these candidate solutions to find $\hat{\boldsymbol{x}}', \hat{\boldsymbol{x}}''$ we obtain the results of Fig. 9.21: while both attackers attain an average $\mathtt{RSNR}_{\hat{\boldsymbol{x}}', \boldsymbol{x}'} \approx 33$ dB on $\boldsymbol{x}'$, Eve is only capable of reconstructing $\boldsymbol{x}''$ with an average $\mathtt{RSNR}_{\hat{\boldsymbol{x}}'', \boldsymbol{x}''} \approx 0.14$ dB where Steve reaches an average $\mathtt{RSNR}_{\hat{\boldsymbol{x}}'', \boldsymbol{x}''} \approx 12.80$ dB with $\eta = 0.03$.

Note also that, although lucky guesses exist with $\mathtt{RSNR}_{\hat{\boldsymbol{x}}'', \boldsymbol{x}''} > 12.57$ dB, it is impossible to identify them by looking at $\mathtt{RSNR}_{\hat{\boldsymbol{x}}', \boldsymbol{x}'}$ since the correlation coefficient between $\mathtt{RSNR}_{\hat{\boldsymbol{x}}', \boldsymbol{x}'}$ and $\mathtt{RSNR}_{\hat{\boldsymbol{x}}'', \boldsymbol{x}''}$ is -0.0041. Thus, Steve cannot rely on observing the $\mathtt{RSNR}_{\hat{\boldsymbol{x}}', \boldsymbol{x}'}$ to choose the best performing solution $\hat{\boldsymbol{A}}^{(1)}$, so both Eve and Steve's KPAs are inconclusive. As a further perceptual evidence of this, the best recoveries according to the $\mathtt{RSNR}_{\hat{\boldsymbol{x}}'', \boldsymbol{x}''}$ are reported in Fig. 9.21.

9.4.6.3 Signal Recovery-Based Class-Upgrade Attacks

Class-upgrade attacks to two-class encryption by CS are closely related to a recovery problem that has attracted the attention of prior contributions, i.e., *sparse signal recovery under matrix uncertainty*, as was partly introduced in Sect. 9.4.1. In this case, we assume the perspective of Steve and let $\boldsymbol{A}^{(1)} = \boldsymbol{A}^{(0)} + \Delta\boldsymbol{A}$ be the encoding matrix, where $\boldsymbol{A}^{(0)}$ is known a priori and $\Delta\boldsymbol{A}$ is an unknown random sparse sign-flipping perturbation matrix. This qualifies as a class-upgrade known-ciphertext attack, as Steve is given $(\boldsymbol{y}, \boldsymbol{A}^{(0)})$ and no other information—if $\boldsymbol{x}$ was also provided, the best approach would still be the KPA in Theorem 9.10.

Steve's information could be paired with a sparsity prior on $\boldsymbol{x}$ to attempt the *joint recovery* of $\boldsymbol{x}$ and $\Delta\boldsymbol{A}$, eventually leading to a mere refinement of the estimate $\hat{\boldsymbol{x}}$ instead of an actual estimate of $\Delta\boldsymbol{A}$. Two main algorithms specifically address

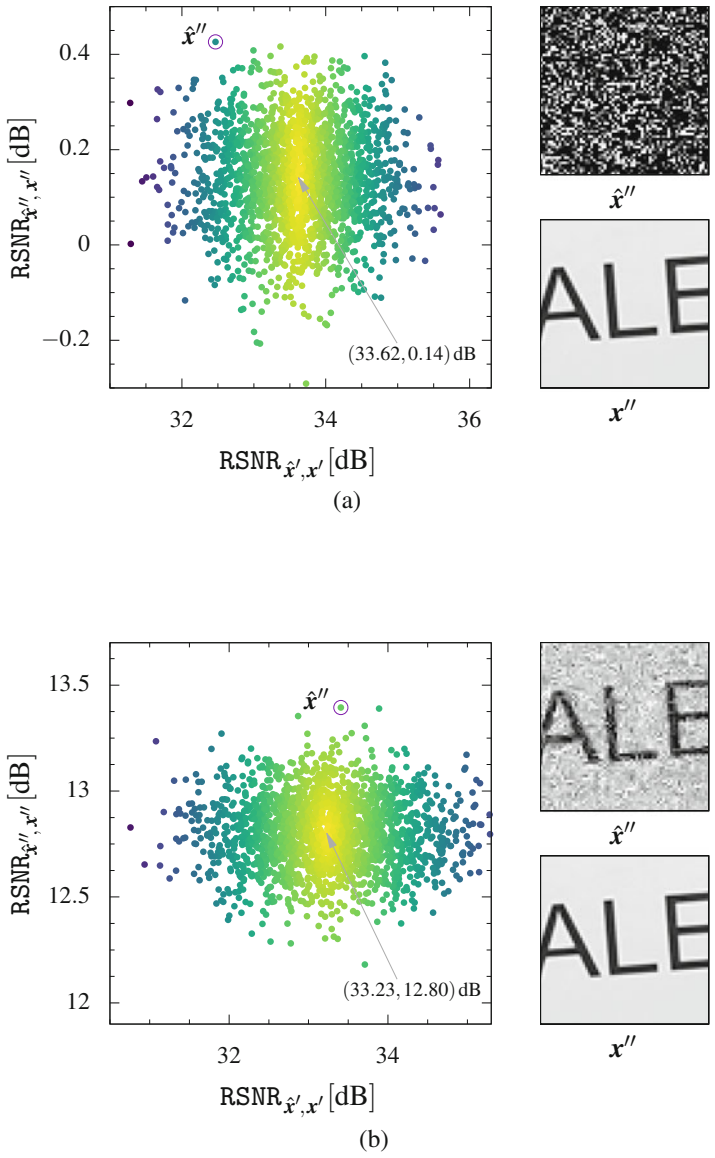


Fig. 9.21 Effectiveness of (**a**) Eve and (**b**) Steve's KPA in recovering hidden image blocks. Each point is a guess of the encoding matrix $\boldsymbol{A}^{(1)}$ whose quality is assessed by decoding the ciphertext $\boldsymbol{y}'$ corresponding to the known plaintext x' ($\mathtt{RSNR}_{\hat{x}',x'}$) and by decoding a new ciphertext $\boldsymbol{y}''$ ($\mathtt{RSNR}_{\hat{x}'',x''}$). The Euclidean distance from the average ($\mathtt{RSNR}_{\hat{x}',x'}, \mathtt{RSNR}_{\hat{x}'',x''}$) is highlighted by color gradient

this problem setup for a generic $\Delta\boldsymbol{A}$, namely Matrix-Uncertainty Generalized Approximate Message Passing (MU-GAMP, [47]) and Sparsity-cognizant Total Least Squares (S-TLS, [60]).

Although appealing in principle, this joint recovery approach can be anticipated to fail for multiple reasons. First, this attack is intrinsically harder than Steve's KPA in that the true plaintext $\boldsymbol{x}$ here is unknown. Whatever $\Delta\boldsymbol{A}$ is a candidate solution to Steve's KPA given $\boldsymbol{x}$, it will also be possible solution of joint recovery with the same $\boldsymbol{x}$. Since we know from Sect. 9.4.5.2 that Steve's KPA typically has a huge number of indistinguishable and equally sparse candidate solutions, at least as many will verify the joint recovery problem when the plaintext is unknown. Hence, this approach has negligible odds of yielding more information on $\Delta\boldsymbol{A}$ than Steve's KPA.

Furthermore, note that joint recovery amounts to solving $\boldsymbol{y} = \boldsymbol{A}^{(0)}\boldsymbol{x} + \Delta\boldsymbol{A}\boldsymbol{x}$ with $\Delta\boldsymbol{A}$ and $\boldsymbol{x}$ unknown, that is clearly a non-linear equality involving non-convex/non-concave operators; in general, this is a hard problem that can only be solved in a relaxed form (as, in fact, does S-TLS).

The aforementioned algorithms are indeed able to compensate matrix uncertainties when $\Delta\boldsymbol{A}$ depends on a low-dimensional, deterministic set of parameters. However, such a model does not apply to two-class encryption by CS: even if $\Delta\boldsymbol{A}$ is c-sparse, it has no deterministic structure to leverage in the attack—to make it so, one would need to know the exact set C_0 of c index pairs at which the sign-flipping randomly occurred, which by itself entails a combinatorial search.

In fact, $\Delta\boldsymbol{A}$ is *uniform* in the sense of [47] since it is a realization of a random matrix ensemble with i.i.d. entries having zero-mean and bounded variance. Hence, we expect the accuracy of the estimate $\hat{\boldsymbol{x}}$ with joint recovery (both using S-TLS and MU-GAMP) to agree with the uniform matrix uncertainty case of [47], where negligible improvement is shown w.r.t. the (standard, non-joint) recovery algorithm Generalized Approximate Message Passing (GAMP, [50]). The advocated reason is that the perturbation noise $\boldsymbol{\epsilon} = \Delta\boldsymbol{A}\boldsymbol{x}$ is asymptotically Gaussian for a given $\boldsymbol{x}$ [47, Proposition 2.1]; thus, it is reasonable that a suitably tuned application of GAMP attains near-optimal performances.

We now provide empirical evidence on the ineffectiveness of joint recovery as a class-upgrade attack for finite n, m and sparsity κ. As an example, we let $n = 256$, $m = 128$, $\kappa = 20$, and $\eta = \frac{c}{mn} \in [0.005, 0.1]$ and generate 100 random instances of $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$ with $\boldsymbol{x}$ being κ-sparse w.r.t. a randomly chosen orthonormal basis $\boldsymbol{D}$. For each η, we also generate 100 pairs of matrices $(\boldsymbol{A}^{(0)}, \boldsymbol{A}^{(1)})$ related as (9.26) and encode $\boldsymbol{x}$ by $\boldsymbol{y} = \boldsymbol{A}^{(1)}\boldsymbol{x}$.

Signal recovery is performed by MU-GAMP, S-TLS, and GAMP. To maximize their performances, each of the algorithms is "genie"-tuned to reveal the exact value of the required features of $\boldsymbol{x}$. In particular, MU-GAMP and GAMP are provided with an i.i.d. Bernoulli–Gaussian sparsity-enforcing signal model [50, 56] having the exact mean, variance, and sparsity level of the instances $\boldsymbol{\xi}$. As far as the perturbation $\Delta\boldsymbol{A}$ is concerned, MU-GAMP is given the PMF of its i.i.d. entries. On the other hand, GAMP is initialized with the noise variance of $\boldsymbol{\epsilon} = \Delta\boldsymbol{A}\boldsymbol{x}$, that is assumed as additive white Gaussian noise. S-TLS is run in its locally optimal, polynomial-time version [60, Section IV-B] and fine-tuned w.r.t. its regularization parameter as η varies.

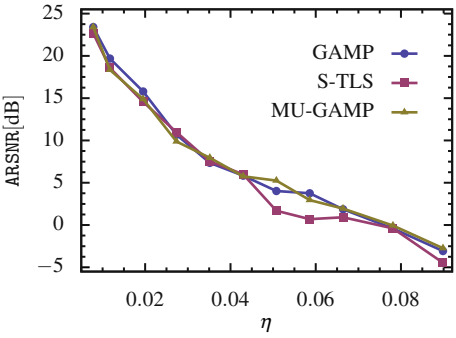


Fig. 9.22 ARSNR performances of a class-upgrade known-ciphertext attack using signal recovery under matrix uncertainty algorithms

Since the typically very low accuracy of the recovered $\Delta \boldsymbol{A}$ is not as relevant to a class-upgrade attack as improving the estimate of $\hat{\boldsymbol{x}}$, we here focus on measuring the usual ARSNR, as reported in Fig. 9.22. The standard deviation from the average is less than 1.71 dB in all the reported curves. The maximum ARSNR performance gap between GAMP and MU-GAMP is 1.22 dB while S-TLS attains generally lower performances for high values of η. These observed performances confirm what is also found in [47], i.e., that GAMP, MU-GAMP, and S-TLS substantially attain the same performances under uniform matrix uncertainty. As expected, class-upgrade attacks based on joint recovery are ineffective even for finite n and m, since GAMP under the same conditions is the reference case adopted in Sect. 9.4.5 for the design of two-class encryption by CS.

References

1. Z. Ben-Haim, Y.C. Eldar, Performance bounds for sparse estimation with random noise, in *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*, IEEE, Aug. 2009, pp. 225–228
2. Z. Ben-Haim, Y.C. Eldar, The Cramér-Rao bound for estimating a sparse parameter vector. IEEE Trans. Signal Process. **58**(6), 3384–3389 (2010)
3. A.C. Berry, The accuracy of the Gaussian approximation to the sum of independent variates. Trans. Am. Math. Soc. **49**(1), 122–136 (1941)
4. T. Bianchi, V. Bioglio, E. Magli, Analysis of one-time random projections for privacy preserving compressed sensing. IEEE Trans. Inf. Forensics Secur. **11**(2), 313–327 (2016)
5. T. Bianchi, V. Bioglio, E. Magli, On the security of random linear measurements, in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, May 2014, pp. 3992–3996
6. T. Bianchi, E. Magli, Analysis of the security of compressed sensing with circulant matrices, in *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, Dec. 2014, pp. 173–178
7. P. Billingsley, *Probability and Measure* (Wiley, New York, 2008)
8. V. Bioglio, T. Bianchi, E. Magli, Secure compressed sensing over finite fields, in *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, Dec. 2014, pp. 191–196
9. V. Cambareri et al., A two-class information concealing system based on compressed sensing, in *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, IEEE, May 2013, pp. 1356–1359

10. V. Cambareri et al., Low-complexity multiclass encryption by compressed sensing. IEEE Trans. Signal Process. **63**(9), 2183–2195 (2015)
11. V. Cambareri et al., On known-plaintext attacks to a compressed sensing-based encryption: A quantitative analysis. IEEE Trans. Inf. Forensics Secur. **10**(10), 2182–2195 (2015)
12. E.J. Candes, J.K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements. Commun. Pure Appl. Math. **59**(8), 1207–1223 (2006)
13. E.J. Candes, T. Tao, Decoding by linear programming. IEEE Trans. Inf. Theory **51**(12), 4203–4215 (2005)
14. E.J. Candes, T. Tao, Near-optimal signal recovery from random projections: Universal encoding strategies? IEEE Trans. Inf. Theory **52**(12), 5406–5425 (2006)
15. B. Chor, R.L. Rivest, A knapsack-type public key cryptosystem based on arithmetic in finite fields. IEEE Trans. Inf. Theory **34**(5), 901–909 (1988)
16. T.M. Cover, J.A. Thomas, *Elements of Information Theory* (Wiley, New York, 2012)
17. D.L. Donoho, A. Maleki, A. Montanari, Message-passing algorithms for compressed sensing. Proc. Natl. Acad. Sci. **106**(45), 18914–18919 (2009)
18. I. Drori, Compressed video sensing, in *BMVA Symposium on 3D Video-Analysis, Display and Applications*, 2008
19. D. Eastlake, P. Jones, in *US Secure Hash Algorithm 1 (SHA1)*, 2001
20. E. Ehrhart, Sur un probleme de géométrie diophantienne linéaire. II. Systemes diophantiens linéaires. (French). J. f̂ur die reine und angewandte Mathematik **227**, 25–49 (1967)
21. R. Fay, Introducing the counter mode of operation to Compressed Sensing based encryption. Inf. Process. Lett. **116**(4), 279–283 (2016)
22. S. Geman et al., A limit theorem for the norm of random matrices. Ann. Probab. **8**(2), 252–261 (1980)
23. A.L. Goldberger et al., Physiobank, Physiotoolkit, and Physionet: components of a new research resource for complex physiologic signals. Circulation **101**(23), 215–220 (2000)
24. M.A. Herman, T. Strohmer, General deviants: An analysis of perturbations in compressed sensing. IEEE J. Sel. Top. Signal Process. **4**(2), 342–349 (2010)
25. G. Hinton et al., Deep neural networks for acoustic modeling in speech recognition: the Shared Views of Four Research Groups. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)
26. ILOG, Inc., *ILOG CPLEX: High-Performance Software for Mathematical Programming and Optimization*. http://www.ilog.com/products/cplex/.2015
27. L. Jacques, D.K. Hammond, J.M. Fadili, Dequantizing compressed sensing: When oversampling and non-Gaussian constraints combine. IEEE Trans. Inf. Theory **57**(1), 559–571 (2011)
28. L. Jacques et al., Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. IEEE Trans. Inf. Theory **59**(4), 2082–2102 (2013)
29. R. Jane et al., Evaluation of an automatic threshold based detector of waveform limits in Holter ECG with the QT database. *Computers in Cardiology 1997*, IEEE, Sept. 1997, pp. 295–298
30. A. Kerckhoffs, La cryptographie militaire. J. des sciences militaires **IX**, 5–38 (1883)
31. B. Klartag, S. Sodin, Variations on the Berry–Esseen Theorem. Theory Probab. Appl. **56**(3), 403–419 (2012)
32. J.C. Lagarias, A.M. Odlyzko, Solving low-density subset sum problems. J. ACM (JACM) **32**(1), 229–246 (1985)
33. R. Latała, Some estimates of norms of random matrices. Proc. Am. Math. Soc. **133**(5), 1273–1282 (2005)
34. P.-L. Loh, M.J. Wainwright, Corrupted and missing predictors: minimax bounds for high-dimensional linear regression, in *2012 IEEE International Symposium on Information Theory Proceedings*, IEEE, July 2012, pp. 2601–2605
35. P.-L. Loh, M.J. Wainwright, et al., High-dimensional regression with noisy and missing data: Provable guarantees with nonconvexity. Ann. Stat. **40**(3), p. 1637 (2012)
36. I.G. MacDonald, Polynomials associated with finite cell-complexes. J. Lond. Math. Soc. **2**(1), 181–192 (1971)
37. S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*. Access Online via Elsevier, 2008

38. S. Martello, P. Toth, *Knapsack Problems: Algorithms and Computer Implementations* (Wiley, New York, 1990)
39. J.L. Massey, Shift-register synthesis and BCH decoding. IEEE Trans. Inf. Theory **15**(1), 122–127 (1969)
40. M. Matsumoto et al., Cryptographic mersenne twister and fubuki stream/block cipher. Cryptographic ePrint Archive (June 2005)
41. R.J. McEliece, *A Public-key Cryptosystem Based on Algebraic Coding Theory*. Tech. rep., Jet Propulsion Laboratory, Pasadena, CA, Jan. 1978, pp. 114–116
42. R. Merkle, M. Hellman, Hiding information and signatures in trapdoor knapsacks. IEEE Trans. Inf. Theory **24**(5), 525–530 (1978)
43. D. Needell, J.A. Tropp, CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Appl. Comput. Harmon. Anal. **26**(3), 301–321 (2009)
44. H. Niederreiter, Knapsack-type cryptosystems and algebraic coding theory. Probl. Control Inf. Theory **15**(2), 159–166 (1986)
45. A.M. Odlyzko, The rise and fall of knapsack cryptosystems. Cryptol. Comput. Number Theory **42**, 75–88 (1990)
46. A. Orsdemir et al., On the security and robustness of encryption via compressed sensing, in *MILCOM 2008 - 2008 IEEE Military Communications Conference*, IEEE, Nov. 2008, pp. 1–7
47. J.T. Parker, V. Cevher, P. Schniter, Compressive sensing under matrix uncertainties: an approximate message passing approach, in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, IEEE, Nov. 2011, pp. 804–808
48. G. Pirker et al., A pitch tracking corpus with evaluation on multipitch tracking scenario, in *Interspeech 2011*, Aug. 2011, pp. 1509–1512
49. Y. Rachlin, D. Baron, The secrecy of compressed sensing measurements, in *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, IEEE, Sept. 2008, pp. 813–817
50. S. Rangan, Generalized approximate message passing for estimation with random linear mixing, in *2011 IEEE International Symposium on Information Theory Proceedings*, IEEE, July 2011, pp. 2168–2172
51. T. Sasamoto, T. Toyoizumi, H. Nishimori, Statistical mechanics of an NP-complete problem: subset sum. J. Phys. A Math. Gen. **34**(44), 9555–9568 (2001)
52. C.E. Shannon, Communication theory of secrecy systems. Bell Syst. Tech. J. **28**(4), 656–715 (1949)
53. G. Shires, H. Wennborg, *Web Speech API Specification*. http://dvcs.w3.org/hg/speech-api/raw-file/tip/speechapi.html, Oct. 2012
54. R. Smith, An overview of the tesseract OCR engine, in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, Sept. 2007, pp. 629–633
55. R. Vershynin, *Introduction to the Non-asymptotic Analysis of Random Matrices* (Cambridge University Press, Cambridge, 2012), pp. 210–268
56. J. Vila, P. Schniter, Expectation-maximization Bernoulli-Gaussian approximate message passing, in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, IEEE, Nov. 2011, pp. 799–803
57. L.C. Washington, W. Trappe, *Introduction to Cryptography: With Coding Theory* (Prentice Hall PTR, Upper Saddle River, 2002)
58. A.D. Wyner, The wire-tap channel. Bell Syst. Tech. J. **54**(8), 1355–1387 (1975)
59. L.Y. Zhang et al., Bi-level protected compressive sampling. IEEE Trans. Multimedia **18**(9), 1720–1732 (2016)
60. H. Zhu, G. Leus, G.B. Giannakis, Sparsity-cognizant total least-squares for perturbed compressive sampling. IEEE Trans. Signal Process. **59**(5), 2002–2016 (2011)