Adapted Compressed Sensing: A Game Worth Playing

Adapted Compressed Sensing: a Game Worth Playing

Mauro Mangia, Fabio Pareschi, Riccardo Rovatti, Gianluca Setti

***Abstract*—Despite the *universal* nature of the compressed sensing mechanism, additional information on the class of sparse signals to acquire allows adjustments that yield substantial improvements. In facts, proper exploitation of these priors allows to significantly increase compression for a given reconstruction quality.**

Since one of the most promising scopes of application of compressed sensing is that of IoT devices subject to extremely low resource constraint, adaptation is especially interesting when it can cope with hardware-related constraint allowing low complexity implementations.

We here review and compare many algorithmic adaptation policies that focus either on the encoding part or on the recovery part of compressed sensing. We also review other more hardware-oriented adaptation techniques that are actually able to make the difference when coming to real-world implementations. In all cases, adaptation proves to be a tool that should be mastered in practical applications to unleash the full potential of compressed sensing.

I. Introduction

All the *magic* of Compressed Sensing (CS) [1] is in the possibility of going back and forth between two vectors $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$ with $m < n$ providing the first is κ-*sparse* ($\kappa < m$). This means that an $n \times d$ matrix $\boldsymbol{D}$ called *dictionary* exists such that the instances of $\boldsymbol{x}$ can be expressed as $\boldsymbol{x} = \boldsymbol{D\xi}$ with $\boldsymbol{\xi}$ having not more than κ non-zero entries.

We go from $\boldsymbol{x}$ to $\boldsymbol{y}$ (the *encoding* step) with a linear transformation $\boldsymbol{y} = \boldsymbol{Ax}$ for a certain $m \times n$ matrix. We go from $\boldsymbol{y}$ to $\boldsymbol{x}$ (the *decoding* step) by exploiting the sparsity prior on $\boldsymbol{x}$ to pick the right element among the infinite set of solutions in $\boldsymbol{\xi}$ of the linear systems of equalities $\boldsymbol{y} = \boldsymbol{AD\xi}$. The key point in decoding is that the search for a sparse solution can be translated into a convex optimization problem that can be tackled either exactly or by means of some iterative approximation [2].

Compressed sensing is no longer an utterly new topic as the seminal papers of Donoho, Candes and Tao [3], [4], [2] that shed light upon this double path date back not less than 12 years.

The amount of theoretical development in this field is impressive as is the number of techniques used for reconstructing the so called *original signal* $\boldsymbol{x}$ from the overly famous *small number of linear measurements* in the vector $\boldsymbol{y}$ [5], [6], [7], [8], [9].

Among all the methodological results, *adaptation* has always been given a secondary role. The very main theory of CS centers on words such as *universal*, *democratic*, *non-adaptive*, etc. Actually, the pairing of sparsity with linear encoding is such a powerful concept that all the theoretical guarantees that ensure CS to work are basically independent of the specific features of the signals involved in the process.

This is a key observation. In fact, a direct consequence of this approach is that typical theoretical guarantees are upper bounds on reconstruction errors, and upper-bounding means that a worst-case analysis has been carried out, and worst-case analysis implicitly considers also the *worst* possible signals, whatever it may mean in each specific context.

From a more applicative point of view, guarantees are fundamental, but signals are not as bad as one is forced to assume in the worst-case analysis. On the contrary, most of the times they have features in addition to sparsity that can be leveraged through adaptation to increase system performance. This is especially important in case of hardware implementations that, by their nature, must maximize performance while complying with possibly severe resource constraints.

Hence, while universal theories deal with asymptotic trends, adaptation allows the tuning of the *constant coefficients* hidden in the asymptotic trend formulas that are irrelevant in the run to infinity, but largely affect the cost and the performance of real-world systems.

This playing with constants can be a key factor in expressing the practical potential of CS. In fact, the ultimate simplicity of the encoding step hints at applications in which resources at the acquisition side are scarce while computational power is available at the receiver of the sensed information. This perfectly fits within the general framework that most of the information engineering community is working on, i.e., the grand view implied by "memes" like *Internet of Things* (IoT) or *Cyber-Physical Systems* (CPS).

Networks of ubiquitous sensors and actuators whose activity entails an intense exchange of data between them and to local hub infrastructures, that act as a gateway to cloud-based processing and decision, and in which acquired signals follow a path that goes from extremely simple sensing units, to concentrators, to server farms. The ability to compress with a limited resource budget is appealing both at individual nodes and in gateways, and CS can be a winning option.

This review article aims at collecting the most widespread adaptation techniques presented in literature which proved capable of improving CS performance on real-world signals and of making the implementation of the acquisition subsystems of IoT networks more effective. We concentrate on *adapted* CS and not on *adaptive* CS, meaning that *adaptation is performed at design-time* considering the class of signal to acquire and *not at run-time* on each signal instance [10]. Roughly speaking, adapted CS methods do not require any resource trade-off. Encoding procedures are still based on the

matrix multiplication $\boldsymbol{Ax}$, with the advantage to (strongly) reduce the number of rows in $\boldsymbol{A}$. With respect to the adoption of the standard CS theory, the produced benefits are both higher data compression and lower computational burden of the encoder. Though we tried to be as exhaustive as possible in bibliographic search, not every contribution is reported here as we concentrate on those that appear to give better performance.

As a teaser on what can be done following this path, consider a case leveraging on the simplicity and flexibility of CS, whose performance can be substantially boosted by adaptation. Electro Cardio Graphics (ECG) signals are of interest in both healthcare and wellness oriented applications as they give information on the status of the hearth as well as on the activity of the subject.

The general effectiveness of CS in ECG monitoring is discussed, for example, in [11], that shows a $\approx 40\%$ improved battery lifetime compared to state-of-the-art compression techniques for an embedded ECG monitor although no adaptation is considered.

The approach discussed in [12] uses CS as a basic building block for ECG compression simultaneously supporting both health care and wellness applications in a dual-mode wearable monitor. More specifically, CS is employed as low-resource scalable lossy compression stage working immediately after signal digitalization and before data dispatching. From $\boldsymbol{y} = \boldsymbol{Ax}$ one gets that each measurement in the vector $\boldsymbol{y}$ contains information on the whole vector $\boldsymbol{x}$ and thus the number of measurements passed to the decoder is a simple but effective way of administering the rate-distortion trade-off.

As reported in Figure 1, the monitoring device is equipped with a Non-Volatile-Memory (NVM) that stores the m^{HQ} measurements needed to reconstruct an High-Quality medical-grade ECG (HQ) and with a transmitter sending only $m^{\mathrm{LQ}} < m^{\mathrm{HQ}}$ of those measurements to a personal device, such as a smartwatch or a smartphone, whose functionalities depend on the reconstruction of a Low-Quality ECG (LQ), e.g., sufficient to reliably estimate heart rate. Actually, the proposed device can switch from LQ to HQ due to an external trigger either from the patient or from the heart rate monitoring device in case of critical events whose scrutiny is needed at medical-grade precision.

Clearly, both storage and transmission take advantage of compression in terms of hardware cost, memory footprint, computation time and, most important, energy consumption. Adaptation plays a fundamental role in this. In fact, once target qualities are fixed, the adoption of classical CS like in [11] compresses with a factor 2.2:1 in the HQ setting and 3.7:1 in the LQ setting. If methods among these presented in this paper are employed, the above figures become a compression ratio of 6.0:1 for the HQ setting and 14.2:1 in the LQ setting.

The illustrated application briefly highlights how the adaptation can be directly employable in the design of market-ready systems. This is also an example of the way in which we expect adaptive CS to play a significant role e in the incoming future.

As an additional example, it is also worth mentioning that the U.S. Food and Drug Administration has recently approved a magnetic resonance imaging scan[1] that uses CS to speed up the images acquisition.

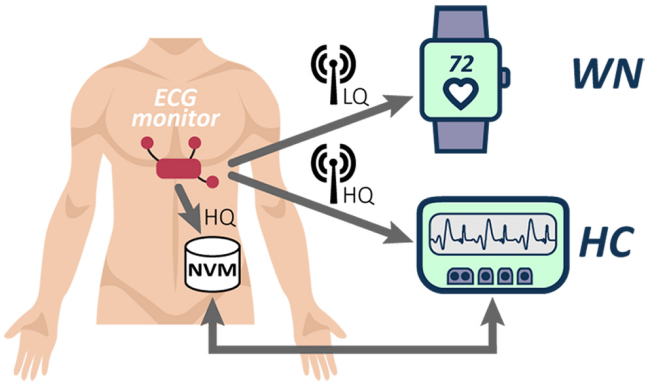

Fig. 1. A dual mode ECG monitor that locally stores compressed data for medical-grade analysis wile a subset of the compressed data are transmitted to a gateway for wellness oriented applications.

The rest of the paper is organized as follows. Section II reports the basics of CS and three of the most useful explanation of its working principle. Section III is devoted to the review of methods that adapt the encoder to the class of signals to acquire, distinguishing between methods inspired by mutual coherence arguments and methods driven by average energy considerations. Improvements with respect to non-adaptive CS provided by encoder adaption are assessed in Section IV. Section V deals with adaptation at the decoder. Section VI develops the ECG monitoring application sketched above by applying the best performing methods both at the encoder and at the decoder. Section VIII is finally devoted to describe adaptation policies that were devised in CS hardware implementations.

II. Basics of CS

The most general model for CS signal encoding is $\boldsymbol{y} = \boldsymbol{A}\left(\boldsymbol{x} + \boldsymbol{\nu}^x\right) + \boldsymbol{\nu}^y$, where $\boldsymbol{\nu}^x \in \mathbb{R}^n$ and $\boldsymbol{\nu}^y \in \mathbb{R}^m$ are vectors of disturbances corresponding to the errors implicitly attached to the signal ($\boldsymbol{\nu}^x$) and those due to the processing producing the measurements ($\boldsymbol{\nu}^y$).

For simplicity's sake, most of the contributions that tune $\boldsymbol{A}$ neglect $\boldsymbol{\nu}^x$ as it makes the overall disturbance contribution $\boldsymbol{A\nu}^x + \boldsymbol{\nu}^y$ dependent on $\boldsymbol{A}$. We adhere to such a simplification, neglect such a dependency, and assume $\boldsymbol{y} = \boldsymbol{Ax} + \boldsymbol{\nu}$ where $\boldsymbol{\nu}$ is a vector of white and Gaussian disturbances $\boldsymbol{\nu} \sim \mathrm{N}\left(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_m\right)$, with $\boldsymbol{I}_m$ the $m \times m$ identity matrix and σ^2 the power of the equivalent noise affecting each measurement. Paired with that, the prototype decoding algorithm is the so called basis pursuit denoising (BPDN) that estimates $\hat{\boldsymbol{x}} = \boldsymbol{D}\hat{\boldsymbol{\xi}}$ with

$$\hat{\boldsymbol{\xi}} = \mathrm{argmin}_{\boldsymbol{\xi} \in \mathbb{R}^d} \|\boldsymbol{\xi}\|_1 \quad \text{s.t.} \quad \|\boldsymbol{y} - \boldsymbol{B\xi}\|_2 \leq \eta \tag{1}$$

where $\|\cdot\|_p$ indicates the p-norm, $\boldsymbol{B} = \boldsymbol{AD}$ and $\eta = \max \|\boldsymbol{\nu}\|_2$ takes into account the maximum deviation from the original

[1] online available on: https://www.healthimaging.com/topics/cardiovascular-imaging/fda-clears-compressed-sensing-mri-acceleration-technology-siemens

signal due to disturbances. Assuming that $\boldsymbol{x} = \boldsymbol{D}\bar{\boldsymbol{\xi}}$ for a certain $\bar{\boldsymbol{\xi}}$ one aims at $\hat{\boldsymbol{\xi}} = \bar{\boldsymbol{\xi}}$.

To grasp the core of CS theory, assume first that $\eta = 0$, the noiseless case in which BPDN reduces to Basis Pursuit (BP). Since $m < n$, then $\boldsymbol{A}$ and $\boldsymbol{B}$ are slanted matrices and, assuming that they are full rank, finding $\bar{\boldsymbol{\xi}}$ means picking the right solution among the infinite number of candidates that satisfy the ill-conditioned system of equations $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$.

A. Explaining CS with restricted isometries

First, let us try to explain how the previous problem may admit a solution by assuming that matrix B satisfies the well-known Restricted Isometry Property (RIP) [13]. Start by noting that an obvious requirement is that no two κ-sparse vectors $\boldsymbol{\xi}' \neq \boldsymbol{\xi}''$ solve $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}_j$. In fact, if $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}' = \boldsymbol{B}\boldsymbol{\xi}''$, i.e., if $\boldsymbol{B}(\boldsymbol{\xi}' - \boldsymbol{\xi}'') = 0$, it is impossible to tell from $\boldsymbol{y}$ which of the two solutions is the true signal.

Going back to the noisy case $\eta > 0$, not to be fooled by disturbances, one should require that for any two κ-sparse vectors $\boldsymbol{\xi}' \neq \boldsymbol{\xi}''$, the two vectors $\boldsymbol{B}\boldsymbol{\xi}'$ and $\boldsymbol{B}\boldsymbol{\xi}''$ are sufficiently far apart, i.e., that $\boldsymbol{B}(\boldsymbol{\xi}' - \boldsymbol{\xi}'')$ is not too small.

Since if $\boldsymbol{\xi}'$ and $\boldsymbol{\xi}''$ are κ-sparse, then $\boldsymbol{\xi}' - \boldsymbol{\xi}''$ is up to 2κ-sparse, and we need $\boldsymbol{B}$ to behave in a *proper way* when applied to 2κ-sparse vectors.

This is formalized by the definition of the RIP that requires $\boldsymbol{B}$ to be almost an isometry (i.e., a transformation that preserves length) when it is applied to 2κ-sparse vectors. RIP is quantified by a Restricted Isometry Constant (RIC) $\delta_{2\kappa}$ that is 0 when $\boldsymbol{B}$ is a true isometry and increases as $\boldsymbol{B}$ departs from that condition.

These ideas have been able to originate the most widely known guarantees on the possibility of reconstructing $\boldsymbol{x}$ from $\boldsymbol{y}$ by means of (1) [3]. By largely simplifying the sophisticated machinery needed to prove this result, one knows that if $\delta_{2\kappa} \leq \sqrt{2} - 1$ the reconstruction error is bounded from above and thus cannot completely disrupt the signal. In the noiseless case $\eta = 0$ the reconstruction can be perfect, and the guarantees substantially depend on the same assumption on $\delta_{2\kappa}$.

Furthermore, one can also prove that, if $m = \mathrm{O}(\kappa \log(n/\kappa))$ and one chooses $\boldsymbol{A}$ as a $m \times n$ random matrix whose entries $\boldsymbol{A}_{j,k}$ are independent and identically distributed (i.i.d.), for example, as normals (i.e., $\boldsymbol{A}_{j,k} \sim \mathrm{N}(0,1)$), then the probability of producing a matrix $\boldsymbol{B}$ that satisfies the RIP property is extremely high independently of the choice of the dictionary D.

These good news imply a very simple design flow that is completely agnostic of the features of the signal to acquire with the exception of its sparsity κ, and basically employs i.i.d. random matrices with the proper size.

Yet, since such a design flow is based on guarantees and thus on worst-case derivations, *in practice the performance of this kind of CS systems on typical signals is much better than what is ensured by theory*.

B. Explaining CS with mutual coherence

Figure 2 helps understanding how CS works by applying it to the particular case $n = d = 3$, $m = 2$ and $\kappa = 1$.

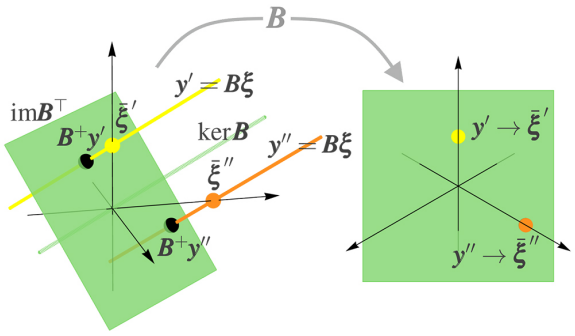

Fig. 2. The mutual-coherence point of view on CS

As a general property, any $\boldsymbol{B}$ decomposes its n-dimensional domain into two orthogonal subspaces: its $(n-m)$-dimensional kernel $\ker \boldsymbol{B}$ (the green line on the left of Figure 2) and the m-dimensional image subspace $\mathrm{im}\boldsymbol{B}^\top$ (the green plane on both sides of Figure 2). Since vectors in $\ker \boldsymbol{B}$ disappear in the mapping, multiplication by $\boldsymbol{B}$ amounts to projection onto $\mathrm{im}\boldsymbol{B}^\top$. Vice versa, once we are given a measurement vector $\boldsymbol{y}'$, the equation $\boldsymbol{y}' = \boldsymbol{B}\boldsymbol{\xi}$ defines the affine $(n-m)$-dimensional subspace spanned by the sum $\boldsymbol{B}^+\boldsymbol{y}' + \boldsymbol{e}$ (where $\cdot^+$ indicates pseudo-inversion) for any $\boldsymbol{e} \in \ker \boldsymbol{B}$ and represented, for example, by the yellow straight line in Figure 2. Since $\kappa = 1$, among all the points of such subspace, the one corresponding to the signal is that sitting on a coordinate axis, i.e. the $\boldsymbol{\xi}'$ represented by a yellow dot in Figure 2.

The same happens for the other measurement vector $\boldsymbol{y}''$, that defines the orange affine subspace $\boldsymbol{B}^+\boldsymbol{y}'' + \boldsymbol{e}$ of Figure 2 which contains only one point sitting on a coordinate axis, i.e., the orange dot representing $\boldsymbol{\xi}''$.

Clearly, this going from $\boldsymbol{y}$ to $\boldsymbol{\xi}$ is possible since the projections of the coordinate axis on the green plane are distinct, as shown on the right of Figure 2 that represents the co-domain of $\boldsymbol{B}$.

In more general terms, since $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$, such projections are the columns of $\boldsymbol{B}$. Since we are dealing with n vectors in an $m < n$-dimensional space, they cannot be orthogonal. Yet, we may ask them to be "as orthogonal as possible". This is where *mutual coherence* [14] comes into play. It is defined considering the columns vectors $\boldsymbol{B}_{\cdot,0}, \ldots, \boldsymbol{B}_{\cdot,n-1}$ and setting

$$\mu(\boldsymbol{B}) = \max_{j \neq k} \frac{\left|\boldsymbol{B}_{\cdot,j}^\top \boldsymbol{B}_{\cdot,k}\right|}{\|\boldsymbol{B}_{\cdot,j}\|_2 \, \|\boldsymbol{B}_{\cdot,k}\|_2} \tag{2}$$

Mutual coherence is the cosine of the smallest angle between any two vectors $\boldsymbol{B}_{\cdot,j}$ and is bounded by $\mu_{\min} \leq \mu(\boldsymbol{B}) \leq 1$ with $\mu_{\min} = \sqrt{\frac{(d-m)}{(d-1)m}}$ [15]. In our toy case, the perfectly symmetric disposition of the 3 projections in the 2-dimensional plane, implies $\mu(\boldsymbol{B}) = 1/2$[2] that matches the lower bound for $d = n = 3$ and $m = 2$ thus making $\boldsymbol{B}$ a very good matrix for CS.

The most well-known result linking mutual coherence with the possibility of reconstructing $\boldsymbol{x}$ from $\boldsymbol{y}$ is that, in the noise-

[2] that corresponds to angles of $2/3\pi$ between each pair of vectors in Figure 2

less case, BP recovers the correct original signal providing that $\mu(\boldsymbol{B}) \leq {}^{1}/_{(2\kappa-1)}$ [16].

In general, to cope with possible disturbances that offset the measurement vectors from its ideal position, one should choose the matrix $\boldsymbol{A}$ so that the mutual coherence of the columns of $\boldsymbol{B} = \boldsymbol{AD}$ is as low as possible to keep the projections of the axes as far as possible from each other.

C. Explaining CS with polytopes

An interesting alternative view on noiseless recovery comes from a polytope interpretation of BP whose working principle can be exemplified using the same toy case used above. The 1-norm sphere of radius r, i.e., $\{\boldsymbol{\xi} \in \mathbb{R}^n | \, \|\boldsymbol{\xi}\|_1 \leq r\}$, is the so-called n-dimensional *cross-polytope*. For $n = 3$ is its the blue diamond-like shape in Figure 3.

In that figure, the radius of the 1-norm sphere is the minimum allowing a non-empty intersection between the sphere itself and the yellow line, an intersection that is the solution of BP. Note that this intersection contains the true $\bar{\boldsymbol{\xi}}$ but only a properly designed $\boldsymbol{B}$ can guarantee that other solutions do not exist.

In particular, since $\kappa = 1$, the solution is on a vertex of the cross-polytope that must be identified starting from the projection $\boldsymbol{B}^{+}\boldsymbol{y}$ on the green plane.

In the case of Figure 3-(a) the projection of the cross-polytope on the green plane yields the red hexagon in which the 6 vertices of the cross-polytope are still distinguishable.

On the contrary, in Figure 3-(b) the projection of cross-polytope on the green plane is the red rectangle in which 2 of the 6 vertices of the cross-polytope disappear.

When this happens, more than one point of the minimum-radius cross-polytope projects on the same $\boldsymbol{B}^{+}\boldsymbol{y}$ and BP is unable to ensure that its solution is sparse and thus coincides with the true signal.

This can be generalized to generic κ-sparse signals sitting on κ-dimensional facets of the cross-polytope, which in this case should be projected on the subspace $\mathrm{im}\boldsymbol{B}^{\top}$ and remain distinguishable. This leads to the estimation of the probability that reconstruction is possible as the ratio between the number of facets that are still recognizable after projection over the total number of facets.

Overall, if one adopts this point of view, good matrices $\boldsymbol{A}$ are those for which $\boldsymbol{B} = \boldsymbol{AD}$ preserves distinguishability of κ-dimensional facets of the n-dimensional cross-polytope.

Adaptation methods start from the theoretical developments that we have just sketched and identify promising merit figures that are related with the capability of effectively reconstructing $\bar{\boldsymbol{\xi}}$ from $\boldsymbol{y}$. Then, propose heuristics to improve such merit figures.

III. Adaptation at the encoder side

In describing all the methods we adopt some common notation. Given a possible sensing matrix $\boldsymbol{A}$, the matrix $\boldsymbol{B} = \boldsymbol{AD}$ and the Gram matrix $\boldsymbol{G} = \boldsymbol{B}^{\top}\boldsymbol{B}$ remain implicitly defined. Given any matrix $\boldsymbol{M}$, we indicate its Singular Value Decomposition (SVD) as $\boldsymbol{M} = \boldsymbol{U_M}\boldsymbol{\Lambda_M}\boldsymbol{V_M}^{\top}$ with $\boldsymbol{U_M}$ and $\boldsymbol{V_M}$ orthonormal matrices and where $\boldsymbol{\Lambda_M}$ the diagonal matrix

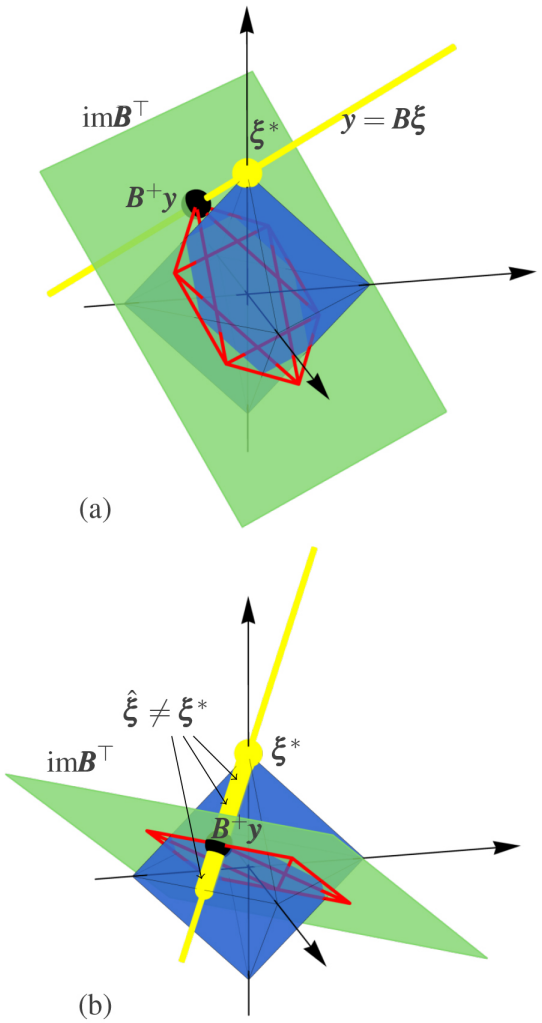

Fig. 3. The polytope point of view of CS

aligning the singular values $\lambda_{\boldsymbol{M}0} \geq \lambda_{\boldsymbol{M}1} \geq \cdots \geq 0$. In what follows, when a matrix $\boldsymbol{M}$ is symmetric it is also positive-semidefinite and we may write its spectral decomposition as $\boldsymbol{M} = \boldsymbol{Q_M}\boldsymbol{\Lambda_M}\boldsymbol{Q_M}^{\top}$ where $\boldsymbol{Q_M}$ is orthonormal.

To avoid that scaling of $\boldsymbol{A}$ in $\boldsymbol{y} = \boldsymbol{Ax} + \boldsymbol{\nu}$ alters the impact of the disturbances $\boldsymbol{\nu}$ we assume that its energy is normalized $\sum_{j=0}^{m-1}\sum_{k=0}^{n-1}\boldsymbol{A}_{j,k}^2 = n$, a constraint that is commonly written in terms of the Frobenius norm $\|\boldsymbol{A}\|_F^2 = n$. Whenever such a normalization is not explicit in the design flow we assume it to be applied as last step.

A. Optimizing mutual coherence

From the theory concerning mutual coherence, one is led to think that the lower the $\mu(\boldsymbol{B})$ the easier and better should be signal recovery. This is the common guideline of a plethora of methods that try to optimize $\boldsymbol{A}$ so that $\boldsymbol{B}$ has the least possible coherence thus adapting sensing to the signal sparsity dictionary.

As a common starting point for all methods, note that, if the columns of $\boldsymbol{B}$ are normalized to unit length, the very same definition of $\boldsymbol{G}$ implies that $\mu(\boldsymbol{B})$ is the maximum of the

magnitudes of the off-diagonal entries in $\boldsymbol{G}$. The prototype optimization problem that is being solved is

$$\min_{\boldsymbol{G}\in\mathfrak{G},\boldsymbol{F}\in\mathfrak{F}} \|\boldsymbol{G}-\boldsymbol{F}\|_{\times} \tag{3}$$

where the $\|\cdot\|_{\times}$ norm can be either the sup norm $\|\cdot\|_{\infty}$ or the Frobenius norm $\|\cdot\|_{F}$ and the matrix set $\mathfrak{F}$ is suitably defined in every variant of (3). As far as $\mathfrak{G}$ is concerned, we need to limit the search to symmetric, positive semidefinite, low-rank matrices with a unit diagonal, i.e.,

$$\mathfrak{G}=\{\boldsymbol{G}|\boldsymbol{G}^{\top}=\boldsymbol{G}\wedge\boldsymbol{G}\succeq 0\wedge\operatorname{rank}(\boldsymbol{G})=m\wedge\operatorname{diag}(\boldsymbol{G})=\boldsymbol{1}\} \tag{4}$$

Since $\boldsymbol{G}$ is symmetric, we have $\boldsymbol{G}=\boldsymbol{Q}_{\boldsymbol{G}}\sqrt{\boldsymbol{\Lambda}_{\boldsymbol{G}}}\sqrt{\boldsymbol{\Lambda}_{\boldsymbol{G}}}^{\top}\boldsymbol{Q}_{\boldsymbol{G}}^{\top}$. If $\boldsymbol{\Lambda}'_{\boldsymbol{G}}$ is the $m\times m$ upper left submatrix of $\boldsymbol{\Lambda}_{\boldsymbol{G}}$ containing the m non-zero eigenvalues and $\boldsymbol{Q}'_{\boldsymbol{G}}$ is the $d\times m$ submatrix of $\boldsymbol{Q}_{\boldsymbol{G}}$ containing the m leftmost columns, then we also have $\boldsymbol{G}=\boldsymbol{Q}'_{\boldsymbol{G}}\sqrt{\boldsymbol{\Lambda}'_{\boldsymbol{G}}}\sqrt{\boldsymbol{\Lambda}'_{\boldsymbol{G}}}^{\top}(\boldsymbol{Q}'_{\boldsymbol{G}})^{\top}$ implying $\boldsymbol{B}=\sqrt{\boldsymbol{\Lambda}'_{\boldsymbol{G}}}^{\top}(\boldsymbol{Q}'_{\boldsymbol{G}})^{\top}$. With this, we may finally set $\boldsymbol{A}=\boldsymbol{B}\boldsymbol{D}^{+}$.

As far as $\mathfrak{G}$ and $\mathfrak{F}$ are convex sets defined as the intersection of elementary convex sets, the key technique for solving (3) is a mix of projected gradient descent [17] or shrinking in which the projection on $\mathfrak{G}$ and $\mathfrak{F}$ is computed by the method of alternating projections [18].

What follows is a brief overview of the proposals that use the above setting, each of them labeled with the prefix coh- followed by the initial of one of the proposing authors.

1) The "coh-S" method in [19]: The method is equivalent to set $\|\cdot\|_{\times}=\|\cdot\|_{F}$ and $\mathfrak{F}=\{\boldsymbol{I}_d\}$, with $\boldsymbol{I}_d$ the $d\times d$ identity matrix, thus simply pursuing the reduction of all off-diagonal entries of $\boldsymbol{G}$.

As noted in [20], pushing $\boldsymbol{G}$ towards $\boldsymbol{I}_d$ can be also interpreted, under suitable assumptions on the signal to acquire, in terms of minimization of the average squared error committed by an *oracle* estimator of $\boldsymbol{\xi}$ that knows in advance which are the non-zero entries.

2) The "coh-C" method in [21]: The author notices that when $d>n$ the dictionary is redundant and this implies some coherence between the columns of $\boldsymbol{D}$. Since vectors forming a small angle get projected into vectors forming a small angle, such a coherence is imported in $\boldsymbol{AD}$ whatever the $\boldsymbol{A}$.

Hence, instead of trying to reduce cross correlation, it is more sensible to make $\boldsymbol{G}$ as close as possible to the Gram matrix of the dictionary alone $\boldsymbol{D}^{\top}\boldsymbol{D}$. Hence, the method sets $\mathfrak{F}=\{\boldsymbol{D}^{\top}\boldsymbol{D}\}$ and considers both $\|\cdot\|_{\times}=\|\cdot\|_{F}$ and $\|\cdot\|_{\times}=\|\cdot\|_{\infty}$.

3) The "coh-X" method in [22]: The authors note that pushing $\boldsymbol{G}$ towards $\boldsymbol{I}_d$ is not completely justified by the objective of making the columns of $\boldsymbol{B}$ as distinguishable as possible. In fact, assuming that *distinguishable* can be interpreted as *orthogonal*, a set of d vectors in $\mathbb{R}^m$ that are *as orthogonal as possible* is a Grassmannian frame (GF) [23].

Then it would be convenient to define $\mathfrak{F}$ as the set of all the possible Gram matrices corresponding to a GF. From [23] we know that, if all the columns are normalized to unit length, the absolute value of the scalar product of every pair of vectors in a GF matches $\mu_{\min}$. Hence, the corresponding Gram matrix $\boldsymbol{F}$ is such that $\boldsymbol{F}_{j,k}=\pm\mu_{\min}$ for every $j\neq k$. Regrettably the set of such matrices is not convex and [22] relaxes it to one of its convex supersets. In particular it considers the set of symmetric, unit diagonal matrices, whose off-diagonal entries have a magnitude not larger than $\mu_{\min}$, i.e., $\mathfrak{F}=\{\boldsymbol{F}|\boldsymbol{F}^{\top}=\boldsymbol{F}\wedge\boldsymbol{F}\succeq 0\wedge\operatorname{diag}(\boldsymbol{F})=1\wedge|\boldsymbol{F}_{j,k}|\leq\mu_{\min}\ \forall j\neq k\}$.

4) The "coh-B" method in [24]: This method puts an even stronger emphasis on frame-based design by considering Equiangular Tight Frames (ETFs) [25], [26]. Tightness means that $\left\|\boldsymbol{B}^{\top}\boldsymbol{y}\right\|_2^2=\alpha\|\boldsymbol{y}\|_2^2$ for some $\alpha>0$. Equiangularity implies $\alpha=d/m$. Hence, the SVD of $\boldsymbol{B}$ then becomes $\boldsymbol{B}=\boldsymbol{U}_{\boldsymbol{B}}\left[\sqrt{d/m}\boldsymbol{I}_m\ \ \boldsymbol{0}\right]\boldsymbol{V}_{\boldsymbol{B}}^{\top}$ and this forces $\boldsymbol{G}=\boldsymbol{B}^{\top}\boldsymbol{B}$ to have a well defined spectral structure that is exploited to redefine $\mathfrak{G}$ as

$$\mathfrak{G}=\left\{\boldsymbol{G}|\boldsymbol{G}=\boldsymbol{Q}_{\boldsymbol{G}}\begin{bmatrix}\frac{d}{m}\boldsymbol{I}_m & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\boldsymbol{Q}_{\boldsymbol{G}}^{\top}\ \wedge\ \boldsymbol{Q}_{\boldsymbol{G}}\in\mathbb{R}^{d\times d}\ \wedge\ \boldsymbol{Q}_{\boldsymbol{G}}\boldsymbol{Q}_{\boldsymbol{G}}^{\top}=\boldsymbol{I}_d\right\}$$

where $\boldsymbol{Q}_{\boldsymbol{G}}=\boldsymbol{V}_{\boldsymbol{B}}$.

Further to that, [24] balances the goal of low mutual coherence with the need for minimizing the effect of non-perfect sparsity, something that we will not address in this overview.

5) The "coh-L" method in [27]: This method mixes coh-S and coh-X. It first notices that, for a full-rank dictionary $\boldsymbol{D}$ whose SVD features $\boldsymbol{\Lambda}_{\boldsymbol{D}}=\begin{bmatrix}\boldsymbol{\Lambda} & \boldsymbol{0}\end{bmatrix}$ for some $n\times n$ diagonal and non-singular matrix $\boldsymbol{\Lambda}$, the solution of the optimization problem entailed by the coh-S method is $\boldsymbol{A}=\begin{bmatrix}\boldsymbol{P}' & \boldsymbol{0}\end{bmatrix}\boldsymbol{P}''\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_{\boldsymbol{D}}^{\top}$ with $\boldsymbol{P}'\in\mathbb{R}^{m\times m}$ and $\boldsymbol{P}''\in\mathbb{R}^{n\times n}$ arbitrary orthonormal matrices. The corresponding Gram matrix is

$$\boldsymbol{G}=\boldsymbol{V}_{\boldsymbol{D}}\begin{bmatrix}\boldsymbol{P}''^{\top}\\ \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{I}_m & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{P}'' & \boldsymbol{0}\end{bmatrix}\boldsymbol{V}_{\boldsymbol{D}}^{\top}$$

that can be further optimized according to coh-X using $\boldsymbol{P}''$ as a degree of freedom.

6) The "coh-E" method in [28]: As a slight variation, one may think that reducing *all* the pairwise correlations between columns of $\boldsymbol{B}$ is a too ambitious goal. This method departs from such a worst-case target and concentrates on the largest correlations, replacing the merit figure in (3) with the average of the off-diagonal entries in $\boldsymbol{G}$ that exceed a certain threshold t, i.e., $\frac{\sum_{|\boldsymbol{G}_{j,k}|\geq t}|\boldsymbol{G}_{j,k}|}{\sum_{|\boldsymbol{G}_{j,k}|\geq t}1}$.

B. Adaptation considering second-order statistics

The common characteristic of the adaptation methods considered so far is that they rely on their design of $\boldsymbol{A}$ starting from the knowledge of $\boldsymbol{D}$ with algorithms metrics that do not consider how both signal and noise energy are processed by $\boldsymbol{A}$. Conversely, the adaptation methods we discuss in this section base their effectiveness on how such energy is processed in the encoder block. In particular, the last two methods we present use also the knowledge of the second-order statistical characterization of the input signals as an additional prior. Depending on the application, suck knowledge could be either

possessed in advance, or it must be obtained by estimation of the second-order statistics on a proper set of possible signals.

What follows is a brief overview of the corresponding proposals, each of them labeled with the prefix pow- followed by the initial of one of the proposing authors.

1) The "pow-C" method in [20]: This method shares the attention to mutual coherence that characterizes coh-S and aims at obtaining $\boldsymbol{G}$ as close as possible to $\boldsymbol{I}_d$. Yet, it notices that if two matrices $\boldsymbol{A}'$ and $\boldsymbol{A}''$ exist such that setting $\boldsymbol{B}' = \boldsymbol{A}'\boldsymbol{D}$ and $\boldsymbol{B}'' = \boldsymbol{A}''\boldsymbol{D}$ one has $\boldsymbol{B}'^{\top}\boldsymbol{B}' = \boldsymbol{B}''^{\top}\boldsymbol{B}'' = \boldsymbol{I}_d$, then, $\boldsymbol{A}''$ is preferable to $\boldsymbol{A}'$ whenever $\|\boldsymbol{A}''\|_F < \|\boldsymbol{A}'\|_F$.

To see why, assume, for simplicity's sake, that the sparse representation of the signal is white, so that $\mathbf{E}\left[\boldsymbol{\xi}\boldsymbol{\xi}^{\top}\right] = \boldsymbol{I}_d$. The power of the signal-related components in the measurement vector $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi} + \boldsymbol{\nu}$ is $\mathbf{E}\left[\|\boldsymbol{B}\boldsymbol{\xi}\|_2^2\right] = \mathbf{E}\left[\boldsymbol{\xi}^{\top}\boldsymbol{B}^{\top}\boldsymbol{B}\boldsymbol{\xi}\right] = \mathrm{tr}\left(\boldsymbol{B}^{\top}\boldsymbol{B}^{\top}\mathbf{E}\left[\boldsymbol{\xi}\boldsymbol{\xi}^{\top}\right]\right) = \|\boldsymbol{B}\|_F^2$.

To enforce the energy constraint $\|\boldsymbol{A}\|_F^2 = n$ one sets $\boldsymbol{A} = \frac{\sqrt{n}}{\|\boldsymbol{A}'\|_F}\boldsymbol{A}'$ and thus $\boldsymbol{B} = \frac{\sqrt{n}}{\|\boldsymbol{A}'\|_F}\boldsymbol{B}'$ so that the power of the signal-related component is $\frac{n}{\|\boldsymbol{A}'\|_F^2}\|\boldsymbol{B}'\|_F^2 = \frac{nd}{\|\boldsymbol{A}'\|_F^2}$ since the columns of $\boldsymbol{B}'$ are normalized to unit length. Alternatively, by choosing $\boldsymbol{A}''$ instead of $\boldsymbol{A}'$ one obtains that the power of the signal-related component becomes $\frac{n}{\|\boldsymbol{A}''\|_F^2}\|\boldsymbol{B}''\|_F^2 = \frac{nd}{\|\boldsymbol{A}''\|_F^2} > \frac{nd}{\|\boldsymbol{A}'\|_F^2}$.

As the disturbance $\boldsymbol{\nu}$ is independent of $\boldsymbol{A}$, increasing the signal-related component is surely beneficial and the method designs $\boldsymbol{A}$ by solving

$$\min_{\boldsymbol{A}\in\mathbb{R}^{m\times n}} \|\boldsymbol{A}\|_F \;\text{ s.t. }\; \boldsymbol{D}^{\top}\boldsymbol{A}^{\top}\boldsymbol{A}\boldsymbol{D} = \boldsymbol{I}_d \tag{5}$$

and then normalizing $\boldsymbol{A}$ to satisfy the sensing energy constraint.

The solution of (5) can be written in terms of the SVD of $\boldsymbol{D}$ and of the n-dimensional counter-diagonal unit matrix $\boldsymbol{J}_n$ to set

$$\boldsymbol{A} = \boldsymbol{P}\left[\mathrm{diag}\left(\lambda_{\boldsymbol{D}m-1}^{-1}, \lambda_{\boldsymbol{D}m-2}^{-1}, \ldots, \lambda_{\boldsymbol{D}0}^{-1}\right) \quad \boldsymbol{0}\right]\boldsymbol{J}_n\boldsymbol{U}_{\boldsymbol{D}}$$

where $\boldsymbol{P}$ is an arbitrary orthonormal matrix.

2) The "pow-P" method in [29]: This method was devised for radar application. Here, dictionaries are typically built from the collection of responses from every possible target position and thus features a number of vectors d much larger than the number κ of non-null components in $\boldsymbol{\xi}$, which correspond to the number of actual targets and is in the order of few units. It will show that this aspect also reflects to the fact that this method guarantees very high performance when d/k is very high while a strong performance degradation is observed for cases with lower d/k values. The general setting comprises a signal $\boldsymbol{D}\boldsymbol{\xi}$, the disturbances $\boldsymbol{\nu}$ and an interference contribution that we do not consider here. The optimization procedure for $\boldsymbol{A}$ is heuristically derived by assuming that signal and noise are Gaussian with $\boldsymbol{\xi} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\xi}})$ for a certain covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}$ and $\boldsymbol{\nu} \sim \mathrm{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I}_n)$. Since $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$ then one has $\boldsymbol{x} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$ with $\boldsymbol{\Sigma}_{\boldsymbol{x}} = \boldsymbol{D}\boldsymbol{\Sigma}_{\boldsymbol{\xi}}\boldsymbol{D}^{\top}$ that can be decomposed as $\boldsymbol{\Sigma}_{\boldsymbol{x}} = \boldsymbol{Q}_{\boldsymbol{\Sigma}_{\boldsymbol{x}}}\boldsymbol{\Lambda}_{\boldsymbol{\Sigma}_{\boldsymbol{x}}}\boldsymbol{Q}_{\boldsymbol{\Sigma}_{\boldsymbol{x}}}^{\top}$.

With these assumptions, the method considers the estimation of $\boldsymbol{\xi}$ from $\boldsymbol{y}$ and the corresponding confidence ellipsoid, i.e. the set in which estimations fall with a certain probability. In our case the shape and size of such an ellipsoid depend on the conditioned covariance $\boldsymbol{\Sigma}_{\boldsymbol{\xi}|\boldsymbol{y}}$. In particular, since the axes of the ellipsoid are proportional to the eigenvalues of $\boldsymbol{\Sigma}_{\boldsymbol{\xi}|\boldsymbol{y}}$ (that are non-negative), the method aims at making the distribution of estimations more concentrated, hopefully along the right direction, by minimizing

$$\min_{\boldsymbol{A}\in\mathbb{R}^{m\times n}} \mathrm{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{\xi}|\boldsymbol{y}}\right) \quad \text{s.t.} \quad \begin{cases} \|\boldsymbol{A}\|_F^2 = n \\ \boldsymbol{A}\boldsymbol{A}^{\top} \text{ and } \boldsymbol{\Sigma}_{\boldsymbol{x}} \text{ have the same eigenvectors} \end{cases} \tag{6}$$

The solution of such a problem is given in terms of the eigenvalues of $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ and of the squared lengths t_k of the columns of $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}\boldsymbol{D}^{\top}\boldsymbol{Q}_{\boldsymbol{\Sigma}_{\boldsymbol{x}}}$. More specifically, the method in [29] sets $\boldsymbol{A} = \mathrm{diag}\left(\sqrt{\mu_0}, \ldots, \sqrt{\mu_{m-1}}\right)\boldsymbol{Q}_{\boldsymbol{\Sigma}_{\boldsymbol{x}}\cdot,0:m-1}^{\top}$ where $\boldsymbol{Q}_{\boldsymbol{\Sigma}_{\boldsymbol{x}}\cdot,0:m-1}$ is the matrix made of the first m columns of $\boldsymbol{Q}_{\boldsymbol{\Sigma}_{\boldsymbol{x}}}$ and

$$\mu_j = \frac{t_j}{\lambda_{\boldsymbol{\Sigma}_{\boldsymbol{x}}j}} \frac{n + \sum_{k=0}^{L-1}\frac{\sigma}{\lambda_{\boldsymbol{\Sigma}_{\boldsymbol{x}}k}}}{\sum_{k=0}^{L-1}\frac{t_k}{\lambda_{\boldsymbol{\Sigma}_{\boldsymbol{x}}k}}} - \frac{\sigma}{\lambda_{\boldsymbol{\Sigma}_{\boldsymbol{x}}j}}$$

3) The "pow-R" method in [30]: Real world signals are not white and this method aims at exploiting this prior. Non-whiteness depends on how average energy is distributed in the signal space. Such an information is contained in the correlation matrix $\boldsymbol{K}_{\boldsymbol{x}} = \mathbf{E}\left[\boldsymbol{x}\boldsymbol{x}^{\top}\right]$ that is the starting point of this method. Consider $\boldsymbol{K}_{\boldsymbol{x}}$ and note that if $\boldsymbol{x}$ were white, then all the eigenvalues of $\boldsymbol{K}_{\boldsymbol{x}}$ would be equal. Hence, one can measure non-whiteness with *localization* [31], that quantifies the deviation of the actual eigenvalues of $\boldsymbol{K}_{\boldsymbol{x}}$ from their equidistributed version

$$\mathcal{L}_{\boldsymbol{x}} = \sum_{j=0}^{n-1}\left(\frac{\lambda_{\boldsymbol{K}_{\boldsymbol{x}}j}}{\mathrm{tr}\left(\boldsymbol{K}_{\boldsymbol{x}}\right)} - \frac{1}{n}\right)^2 = \frac{\mathrm{tr}\left(\boldsymbol{K}_{\boldsymbol{x}}^2\right)}{\mathrm{tr}^2\left(\boldsymbol{K}_{\boldsymbol{x}}\right)} - \frac{1}{n} \tag{7}$$

Localization goes from 0 (white signals) to $1 - 1/n$ (signals whose energy concentrates along a single direction).

The method does not yield deterministic matrices $\boldsymbol{A}$. This slightly complicates the design flow but provides some advantages that can be exploited, for example, for effective implementations.

It assumes that $\boldsymbol{A}$ is a random matrix made of independent and identically distributed non-white rows. If we indicate the generic row as $\boldsymbol{a}^{\top}$ and the corresponding measurement with $y = \boldsymbol{a}^{\top}\boldsymbol{x}$, then the method aims at identifying rows with an high *rakeness*, i.e., with the ability of collecting the largest possible amount of energy from the signal and transfer it to the measurement[3].

[3]This is similar to what some of the authors employed in (chaos-based) DS-CDMA communication, where chip waveforms, spreading sequence statistics and rake receivers taps were jointly selected to collect (rake) as much energy as possible at the received side [32], [33]. In other words, the underlying idea behind the rakeness-based CS is to adapt the statistics of the sensing sequences to the class of input signals by exploiting the fact that the energy and thus the information content of the signal is not uniformly distributed over its whole domain.

Since the power of the measurement is $\mathbf{E}\left[y^2\right] = \mathbf{E}\left[\boldsymbol{x}^\top \boldsymbol{a}\boldsymbol{a}^\top \boldsymbol{x}\right] = \mathrm{tr}\left(\boldsymbol{K}_{\boldsymbol{x}}\boldsymbol{K}_{\boldsymbol{a}}\right)$ the natural design parameter is the correlation matrix $\boldsymbol{K}_{\boldsymbol{a}} = \mathbf{E}\left[\boldsymbol{a}\boldsymbol{a}^\top\right]$ that is fixed by solving

$$\max_{\boldsymbol{K}_{\boldsymbol{a}} \in \mathbb{R}^{n\times n}} \mathrm{tr}\left(\boldsymbol{K}_{\boldsymbol{x}}\boldsymbol{K}_{\boldsymbol{a}}\right) \quad \text{s.t.} \quad \begin{cases} \boldsymbol{K}_{\boldsymbol{a}} \succeq 0 \\ \boldsymbol{K}_{\boldsymbol{a}} = \boldsymbol{K}_{\boldsymbol{a}}^\top \\ \mathrm{tr}\left(\boldsymbol{K}_{\boldsymbol{a}}\right) = \frac{n}{m} \\ \mathcal{L}_{\boldsymbol{a}} \leq \mathcal{L}_{\boldsymbol{a}}^{\max} \end{cases} \tag{8}$$

where the first two constraints ensure that $\boldsymbol{K}_{\boldsymbol{a}}$ is a proper correlation matrix, and the trace constraint makes the power of each row equal to n/m so that the m-rows matrix $\boldsymbol{A}$ satisfies the sensing energy constraint $\|\boldsymbol{A}\|_F = n$ on average.

The localization constraint is used to control the amount of adaptation of the sensing matrix. When $\mathcal{L}_{\boldsymbol{a}}^{\max} = 0$ then the rows are forced to be white and there is no adaptation. When $\mathcal{L}_{\boldsymbol{a}}^{\max} = 1 - 1/n$ all the rows come up lying along the principal component of $\boldsymbol{K}_{\boldsymbol{x}}$ thus making all the measurements maximally energetic but indistinguishable and thus useless.

The problem in (8) can be given an analytical solution that, under the assumption $\mathcal{L}_{\boldsymbol{a}}^{\max} \leq \mathcal{L}_{\boldsymbol{x}}$, reads

$$\boldsymbol{K}_{\boldsymbol{a}} = \frac{n}{m}\frac{\boldsymbol{K}_{\boldsymbol{x}}}{\mathrm{tr}\left(\boldsymbol{K}_{\boldsymbol{x}}\right)}\sqrt{\frac{\mathcal{L}_{\boldsymbol{a}}^{\max}}{\mathcal{L}_{\boldsymbol{x}}}} + \frac{1}{m}\boldsymbol{I}_n\left(1 - \sqrt{\frac{\mathcal{L}_{\boldsymbol{a}}^{\max}}{\mathcal{L}_{\boldsymbol{x}}}}\right)$$

Once that $\boldsymbol{K}_{\boldsymbol{a}}$ is fixed, one may draw $\boldsymbol{a}' \sim \mathrm{N}\left(0, \boldsymbol{I}_n\right)$ and set $\boldsymbol{a} = \boldsymbol{Q}_{\boldsymbol{K}_{\boldsymbol{a}}}\sqrt{\boldsymbol{\Lambda}_{\boldsymbol{K}_{\boldsymbol{a}}}}\boldsymbol{a}'$ to produce a Gaussian row $\boldsymbol{a}^\top$ with the proper correlation.

Further to that, methods exist [34][35] to reproduce the same statistical behavior even if the the entries of $\boldsymbol{A}$ are constrained to some low-cardinality, hardware-friendly [36], [37], set of values like $\boldsymbol{A}_{j,k} \in \{-1, 1\}$, $\boldsymbol{A}_{j,k} \in \{-1, 0, 1\}$, or $\boldsymbol{A}_{j,k} \in \{0, 1\}$. If hardware friendliness is a major issue, the constraint on the kind of entries in $\boldsymbol{A}$ can be plugged directly into the rakeness-based design flow by adjusting (8) [31] to yield sensing matrices that allow substantial savings in computational complexity.

C. Power-based adaptation is not only about noise

The most obvious rationale beyond pow-C and pow-R methods is that they increase the power of the signal-related component $\boldsymbol{A}\boldsymbol{x}$ in $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\nu}$ thus countering the effect of noise. Yet, there is a deeper reason regulating their effectiveness that can be illustrated using the polytope interpretation of CS when $\boldsymbol{D}$ is an orthonormal basis.

To illustrate the core concept rotate Figures 3-(b) until it appears as Figure 4. From this point of view, it is clear that BP is not effective as the angle θ between $\ker \boldsymbol{B}$ and the signal $\boldsymbol{\xi}^*$ is equal to the angle between the sparse representation of the signal itself and a facet of the cross-polytope.

To generalize such an unfavorable condition, indicate with $A(\boldsymbol{\xi}^*)$ the union of the facets that are adjacent to $\boldsymbol{\xi}^*$. As an example, in Figure 3, $A(\boldsymbol{\xi}^*)$ is made of the four segments (1-dimensional facets) and the 4 triangles (2-dimensional facets) on the surface of the blue cross-polytope that have $\boldsymbol{\xi}^*$ as a vertex. Based on $A(\boldsymbol{\xi}^*)$, define the set $\Theta(\boldsymbol{\xi}^*)$ of the angles formed by the sparse representation of the signal with any

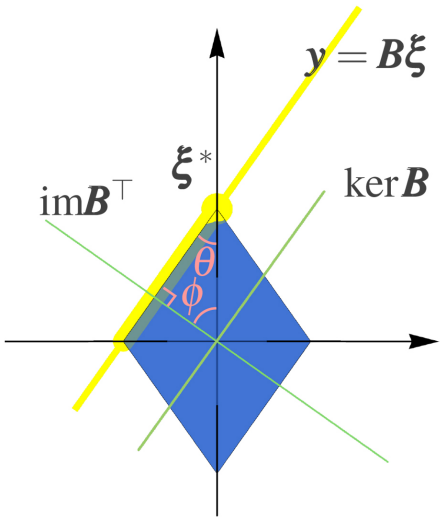

Fig. 4. Angles deciding whether or not reconstruction by BP is possible

segment having $\boldsymbol{\xi}^*$ as a vertex and lying on $A(\boldsymbol{\xi}^*)$. Problems arise when the angle θ between $\ker \boldsymbol{B}$ and $\boldsymbol{\xi}^*$ belongs to $\Theta(\boldsymbol{\xi}^*)$.

To prevent this from happening, we may choose a $\boldsymbol{B}$ such that the angle θ is larger than $\max \Theta(\boldsymbol{\xi}^*)$. Since $\mathrm{im}\ \boldsymbol{B}^\top$ is orthogonal to $\ker\ \boldsymbol{B}$ this translates into the requirement that the angle ϕ between the rows of $\boldsymbol{B}$ (whose linear combinations yield $\mathrm{im}\ \boldsymbol{B}^\top$) and $\boldsymbol{\xi}^*$ is smaller than $\pi/2 - \max \Theta(\boldsymbol{\xi}^*)$. If $\boldsymbol{D}$ is orthonormal, the angles between the rows of $\boldsymbol{B}$ and $\boldsymbol{\xi}^*$ are the same as those between the rows of $\boldsymbol{B}\boldsymbol{D}^\top = \boldsymbol{A}$ and $\boldsymbol{D}\boldsymbol{\xi}^* = \boldsymbol{x}^*$.

Finally, given $\boldsymbol{x}^*$ and assuming that the rows of $\boldsymbol{A}$ are normalized to the same length, reducing the angle means increasing the magnitude of each entry of $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, i.e., its energy.

IV. Encoder adaptation at work

We test the above methods in a common environment with $n = 128$ and in which $\boldsymbol{D}$ is either a random orthonormal matrix or a random dictionary with $d = 256$ and normalized columns. Sparsity is set to $\kappa = 6$.

Each signal window is generated starting from a random vector $\boldsymbol{x}' \sim \mathrm{N}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{x}'}\right)$ for a certain $\boldsymbol{\Sigma}_{\boldsymbol{x}'}$. Such a vector is then decomposed along $\boldsymbol{D}$ by setting $\boldsymbol{\xi}' = \mathrm{argmin}_{\boldsymbol{\xi}\in\mathbb{R}^d} \|\boldsymbol{\xi}\|_1 \quad \text{s.t.} \quad \boldsymbol{x}' = \boldsymbol{D}\boldsymbol{\xi}'$. The vector $\boldsymbol{\xi}'$ is then sparsified into the vector $\boldsymbol{\xi}''$ by keeping only the κ largest components while setting the others to 0. The signal is finally generated as $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}''$. The matrix $\boldsymbol{\Sigma}_{\boldsymbol{x}'}$ is chosen to make $\boldsymbol{x}$ slightly low-pass and $\mathcal{L}_{\boldsymbol{x}} \simeq 0.03$ of the same magnitude of some classical real-world signals.

Whatever method is used to build the matrix $\boldsymbol{A}$, we compute the measurement vector $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\nu}$ with $\boldsymbol{\nu} \sim \mathrm{N}\left(0, 10^{-6}\boldsymbol{I}_m\right)$ and go from $\boldsymbol{y}$ to an estimation $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ by means of BPDN as implemented in [38].

We evaluate the quality of reconstruction with the Reconstruction Signal-to-Noise-Ratio (RSNR) $\|\boldsymbol{x}\|_2^2 / \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2$. Performance is assessed by considering 4000 Montecarlo trials and computing the average RSNR that we indicate with ARSNR. Though average performance is not a complete

characterization of the effectiveness of CS it will suffice here to give a general idea of what can be obtained by adaptation.

As a reference case we assume the one in which the entries of $\boldsymbol{A}$, before energy normalization, are independent normals $\mathrm{N}(0,1)$. It is a classical setting that we label as "i.i.d." and allows to quantify the improvements due to "different adaptation policies[4]

Since hardware implementations greatly benefit from constraining the $\boldsymbol{A}_{j,k}$ to a small number of possible values, we also consider, as a second setting, the option of substituting $\boldsymbol{A}_{j,k}$ with $m^{-1/2}\mathrm{sign}(\boldsymbol{A}_{j,k})$ to obtain antipodal adapted matrices.

Figure 5 and Figure 6 plot the number of measurements against the ARSRN that they allow to achieve. Since $n = 128$, the vertical span is limited to $m \leq n/2 = 64$ to zoom in the area in which the compression rate is at least 2:1. From the shape of all the curves in the figures it is clear that performance tends to saturate when $m > 64$.

In all plots, the i.i.d. reference case is the black line. Plots are from the designer's point of view: one chooses the reconstruction quality along the horizontal axis and finds the minimum number of measurements (and thus the compression ratio) needed to obtain it. Hence, the lower the curve, the better the method in exploiting the features of the signal to optimize CS performance.

Note that the methods behave differently in the dictionary and orthonormal basis cases. In particular, it is very interesting to notice that the best option in the dictionary case (pow-P), yields extremely low performance in the basis case, in which pow-R consistently delivers the best performance.

It is also to notice that coherence-based methods (dashed lines) seem to provide smaller improvements with respect to power-based methods (solid lines) and to suffer more from antipodal quantization passing from Figure 5 to Figure 6. On the contrary, antipodal quantization does not cause any severe loss in performance to the i.i.d. reference case and to the power-based method.

Overall, adaptation proves to be quite effective. As an example, consider a target ARSNR of 60 dB for a dictionary-based CS. In both unconstrained or antipodal $\boldsymbol{A}$ cases, non-adpated CS needs $m^{\mathrm{i.i.d.}} = 57$ measurements for a 2.2:1 compression ratio. The pow-P method reduces it to $m^{\mathrm{pow-P}} = 38$ for unconstrained $\boldsymbol{A}$ and $m^{\mathrm{pow-P}} = 40$ for antipodal $\boldsymbol{A}$ thus yielding compression ratios of 3.4:1 and 3.2:1 respectively.

In the basis case, non-adapted CS needs $m^{\mathrm{i.i.d.}} = 39$ measurements for a 3.3:1 compression ratio. Yet, the pow-R method yields $m^{\mathrm{pow-P}} = 32$ in both the unconstrained and antipodal $\boldsymbol{A}$ cases, thus bringing compression ratio to 4:1.

V. Adaptation at the decoder side

Though, in principle, any convex optimization solver can be used to solve BPDN, there is a flourishing literature developing alternative reconstruction algorithms. For example, 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 [39], [40], [41]. Moreover, procedures exist that retrieve the original signal by generating solutions to $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$ iteratively and adjusting their sparsity at each step. Different heuristics may be used to promote sparsity and this gives rise to different methods [5], [6], [7]. The simple structure of these methods, and their good performance, make them ideal for CS embodiments in which the resources devoted to signal reconstruction are limited. As an example, the approach of Orthogonal Matching Pursuit (OMP) tries to reconstruct $\mathrm{supp}(\boldsymbol{\xi}) = \{j|\boldsymbol{\xi}_j \neq 0\}$ iteratively by looking for the columns of $\boldsymbol{B}$ that have the highest correlation with the measurements vector $\boldsymbol{y}$, a simplified pseudo-code being in Table I.

TABLE I
Pseudo-code of OMP

$\boldsymbol{\zeta} \leftarrow [\,]$ ▷ initialize the vector that will contain the non-zero components of $\hat{\boldsymbol{\xi}}$
$J \leftarrow [\,]$ ▷ initialize the vector that will contain $\mathrm{supp}\left(\hat{\boldsymbol{\xi}}\right)$
repeat
  $\Delta\boldsymbol{y} \leftarrow \boldsymbol{y} - \boldsymbol{B}_{\cdot,J}\boldsymbol{\zeta}$ ▷ $\boldsymbol{B}_{\cdot,J}$ is the submatrix of $\boldsymbol{B}$ with columns indexed by J
  $j = \mathrm{argmax}_k \left|\boldsymbol{B}_{\cdot,k}^{\top}\Delta\boldsymbol{y}\right|$ ▷ column $\boldsymbol{B}_{\cdot,k}$ of $\boldsymbol{B}$ that best matches the measurements residual
  $J \leftarrow \begin{bmatrix} J & j \end{bmatrix}$ ▷ include it in J
  $\boldsymbol{\zeta} \leftarrow \boldsymbol{B}_{\cdot,J}^{+}\boldsymbol{y}$ ▷ re-estimate $\boldsymbol{\zeta}$ by pseudo-inversion
until convergence
$\hat{\boldsymbol{\xi}}_j \leftarrow \begin{cases} \boldsymbol{\zeta}_k & \text{if } j = J_k \\ 0 & \text{otherwise} \end{cases}$ ▷ put non-zero components back into $\hat{\boldsymbol{\xi}}$

Among this abundance, few methods concentrate on adapting the decoding procedure to the features of the signal. Since the task of the decoder is to retrieve the sparse representation $\boldsymbol{\xi}$, it is most natural to exploit priors applying to that space.

In particular, the priors we consider concern $\mathrm{supp}(\boldsymbol{\xi}) = \{j|\boldsymbol{\xi}_j \neq 0\}$, i.e., the positions of the non-zero elements in $\boldsymbol{\xi}$, or the magnitudes of the non-zero elements in $\boldsymbol{\xi}$.

1) The "dec-ZZ" method in [8]: This approach assumes that the signal is not only sparse but also *block-sparse*. Blocks are subvectors $\boldsymbol{\xi}^{[j]}$ of $\boldsymbol{\xi}$ containing adjacent entries so that one may partition $\boldsymbol{\xi}$ in β blocks $\boldsymbol{\xi}^{\top} = \left[\boldsymbol{\xi}^{[0]\top} \quad \dots \quad \boldsymbol{\xi}^{[\beta-1]\top}\right]$. The signal is block-sparse if $\mathrm{supp}(\boldsymbol{\xi})$ is always contained in the union of a number of blocks $\ll \beta$.

Block-sparsity is a stronger prior than simple sparsity [42] and can be paired with Bayesian learning [8] to yield effective reconstruction algorithms. The core idea is to assume that each block follows a parameterized multivariate Gaussian distribution $\boldsymbol{\xi}^{[j]} \sim \mathrm{N}\left(0, \gamma_j \boldsymbol{\Sigma}_{\boldsymbol{\xi}^{[j]}}\right)$ and is independent of the other blocks so that $\boldsymbol{\xi}_j \sim \mathrm{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\xi}})$ with $\boldsymbol{\Sigma}_{\boldsymbol{\xi}} = \mathrm{diag}\left(\gamma_0\boldsymbol{\Sigma}_{\boldsymbol{\xi}^{[0]}}, \dots, \gamma_{\beta-1}\boldsymbol{\Sigma}_{\boldsymbol{\xi}^{[\beta-1]}}\right)$. With this model, $\gamma_j = 0$ implies that the block does not cover $\mathrm{supp}(\boldsymbol{\xi})$ and thus controls the block sparsity.

Decoding is then divided in two steps. In the first step the parameters γ_j and $\boldsymbol{\Sigma}_{\boldsymbol{\xi}^{[j]}}$ are *learnt* for $j = 0, \dots, \beta - 1$. In the second step they are used to decode $\boldsymbol{\xi}$ by means of straightforward Maximum-A-Posteriori estimation

$$\hat{\boldsymbol{\xi}} = \boldsymbol{\Sigma}_{\boldsymbol{\xi}}\boldsymbol{B}^{\top}\left(\sigma^2\boldsymbol{I}_m + \boldsymbol{B}\boldsymbol{\Sigma}_{\boldsymbol{\xi}}\boldsymbol{B}^{\top}\right)^{-1}\boldsymbol{y}$$

[4]The MATLAB code used to obtain these results is available at `https://goo.gl/6hknan`.

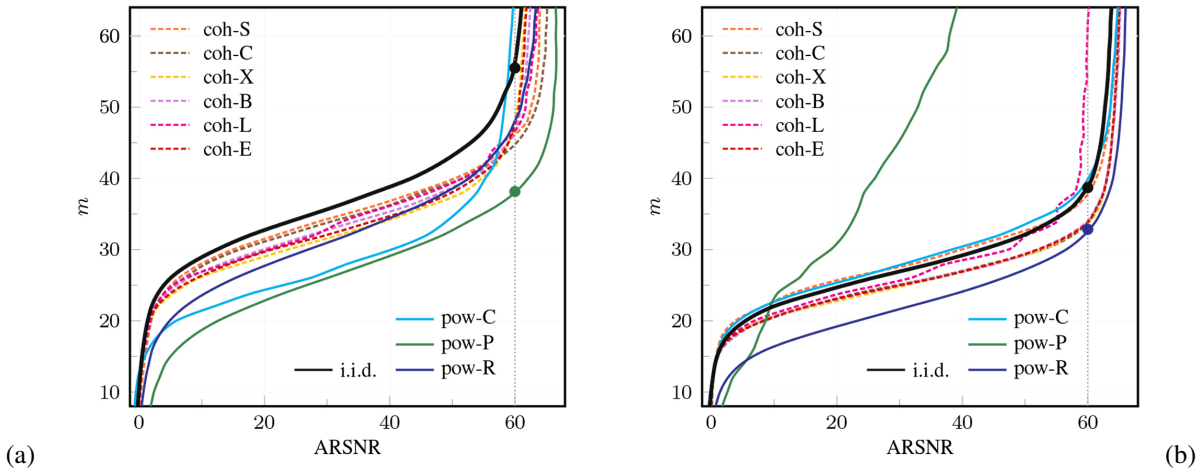

Fig. 5. Performance of different encoder adaptation policies when $\boldsymbol{A}$ is unconstrained: (a) $\boldsymbol{D}$ is a random dictionary; (b) $\boldsymbol{D}$ is a random orthonormal basis.

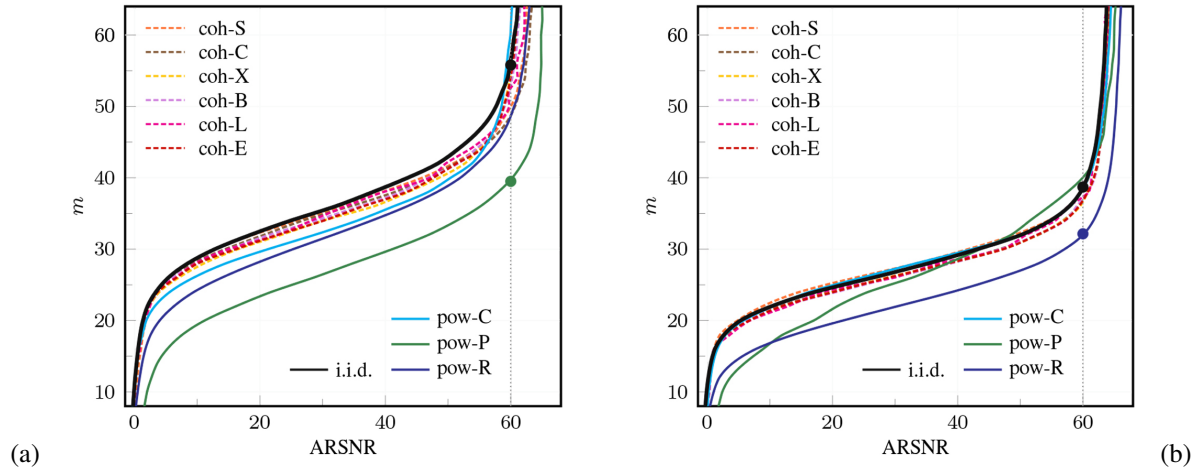

Fig. 6. Performance of different encoder adaptation policies when $\boldsymbol{A}_{j,k} = \pm m^{-1/2}$: (a) $\boldsymbol{D}$ is a random dictionary; (b) $\boldsymbol{D}$ is a random orthonormal basis.

Depending on the different strategies for learning the γ_j and $\boldsymbol{\Sigma}_{\boldsymbol{\xi}[j]}$, this approach gives rise to different methods that differ in computational complexity more than in final performance.

2) The "dec-JZ" method in [43]: In this case one assumes that, when $\boldsymbol{\xi}_j \neq 0$ then its average magnitude varies with j. This is most natural when, for example, $\boldsymbol{D}$ is a wavelet-like orthonormal basis that decomposes the signal into a sequence of *approximation/detail* pairs whose typical decay is known by design of can be identified.

This decay information can be plugged into BPDN by altering the sparsity promoting norm from $\|\boldsymbol{\xi}\|_1$ to $\left\|\boldsymbol{W}^{-1}\boldsymbol{\xi}\right\|_1$ where $\boldsymbol{W}$ is the diagonal matrix aligning the coefficients modeling the decay.

The method appears to be most effective when BPDN is solved considering its *lasso* relaxation, i.e., in

$$\min_{\boldsymbol{\xi}\in\mathbb{R}^d} (1-\alpha)\left\|\boldsymbol{W}^{-1}\boldsymbol{\xi}\right\|_1 + \alpha\left\|\boldsymbol{y}-\boldsymbol{B}\boldsymbol{\xi}\right\|_2^2$$

where the parameter α administers the weight of the two components in the relaxation.

3) The "dec-P" method in [9], [44]: In OMP, the new columns of $\boldsymbol{B}$ to be inserted in the set that is deemed to be necessary to reproduce the measurements are selected for their alignment with the residual measurement. A prior of the kind used in the previous method can alter this selection by altering the opportunity of choosing a column depending on the decay coefficient of the corresponding entry in $\boldsymbol{\xi}$.

This can be done by changing Line 5: in Table I with $j = \operatorname{argmax}_k \left|\boldsymbol{B}_{:,k}^{\top}\left[(1-\alpha)\boldsymbol{W}^{-1} + \alpha\boldsymbol{I}_d\right]\Delta\boldsymbol{y}\right|$, where the parameter α administers the trade-off between following the correct decay and aligning with the residual.

VI. Back to ECGs

The ECG signals we mentioned in the introduction offer the opportunity of testing the effect of adaptation at both the encoder and decoder side. In fact, they can be given an approximately sparse representation along, for example, a Daubechies-6 wavelet basis $\boldsymbol{D}$ that has a dyadic scaling and thus induces a blockwise halving decay in the typical magnitude of the coefficients.

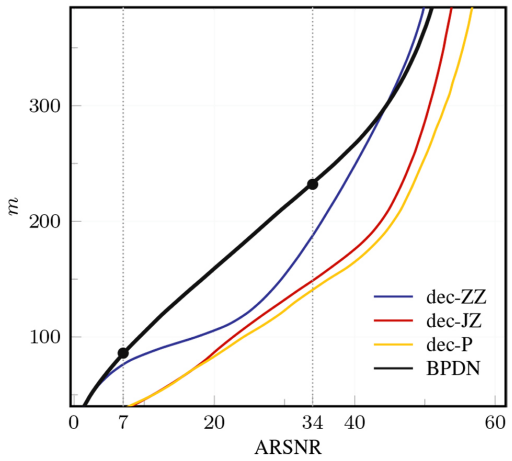

Fig. 7. Performance of different decoder adaptation policies.

We consider windows containing $n = 512$ samples taken at 360 sample/s from a classical procedure generating clean and realistic ECG tracks [45] and superimpose a noise vector $\boldsymbol{\nu} \sim \mathrm{N}\left(\mathbf{0}, 10^{-6}\boldsymbol{I}_m\right)$ to the measurement vector $\boldsymbol{y}$.

Figure 7 compares the performance of the three above decoding strategies against non-adapted BPDN. In all cases $\boldsymbol{A}$ is made of independent Gaussian random variables. Though with different profiles, all methods give definite advantages over non-adapted BPDN.

With reference to the hearth monitoring device briefly sketched in the introduction, we may also evaluate what could be the overall impact of adaptation both at the encoder and decoder side.

Since we are dealing with a portable device in which the resources for the computation of $\boldsymbol{y}$ are limited, we constrain $\boldsymbol{A} = \pm m^{-1/2}$.

The previous analysis (see Figure 6-(b)) suggests that, since $\boldsymbol{D}$ is an orthonormal basis, it is convenient to us the pow-R method that yields consistent improvement across multiple target quality levels. From [12] we get that LQ reconstruction supporting, for example, heart rate estimation may feature an ARSNR as low as 7 dB. Yet, from [46] we get that HQ reconstruction must provide an ARSNR not smaller than 34 dB.

From Figure 7 we get that, when neither the encoder nor the decoder is adapted, an LQ recovery of the signal requires $m^{\mathrm{LQ}} = 86$ thus yielding a 6.0:1 compression, while HQ recovery requires $m^{\mathrm{HQ}} = 232$ thus yielding a 2.2:1 compression. This second result is a pretty standard one as it is accepted that straightforward CS applied to ECGs is a computationally effective way of enjoying a compression in the order of 2:1.

Furthermore, introducing pow-R at the encoder and dec-P at the decoder, i.e., the case where both encoder and decoder are designed in according to an adapted CS method, the number of measurements is reduced to $m^{\mathrm{LQ}} = 36$ and $m^{\mathrm{HQ}} = 137$ thus increasing the compression ratio from 6.0:1 to 14.2:1 in the LQ case and from 2.2:1 to 3.7:1 in the HQ case. To give an intuitive feeling on the improvement due to adaptation we

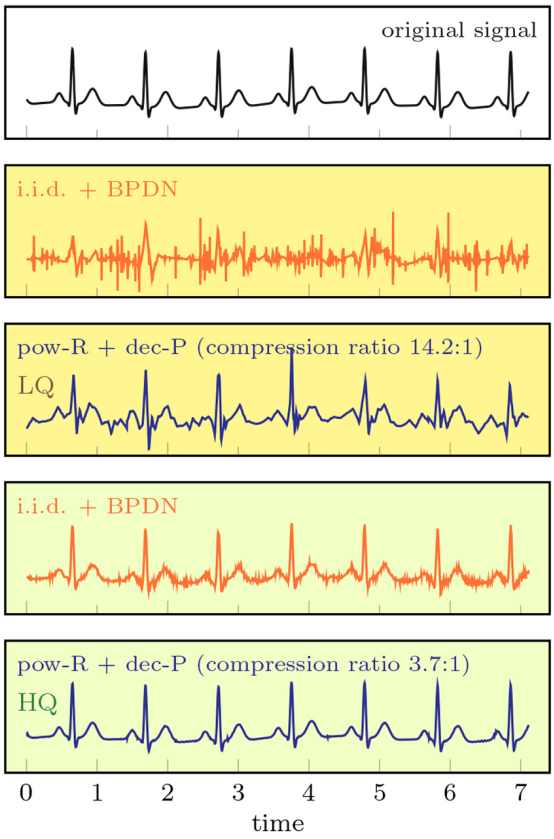

Fig. 8. Example reconstruction of an ECG signal (black track) by non-adapted (red tracks) and adapted (blue tracks) CS for different compression ratios.

may consider 5 windows of an ideal ECG profile as reported at the top of Figure 8. Based on that profile we test both the non-adapted encoder-decoder pair and the adapted one, using the number of measurements that make the adapted pair work either at LQ ($m^{\mathrm{LQ}} = 36$) or HQ ($m^{\mathrm{HQ}} = 137$).

In Figure 8, the yellow-background plots refer to the $m^{\mathrm{LQ}} = 36$, while the green-background plots refer to $m^{\mathrm{HQ}} = 137$. Independently of the compression, adapted CS widely outperforms the non-adapted option.

VII. Adaptation for Hardware Implementation

When dealing with implementations of CS-based acquisition systems, the concept of *adaptation* can be considered also from an "hardware-oriented" perspective. As an example, CS has been characterized so far by the set of linear projections expressed by $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$. However, when dealing with a real world input signal, the very concept of projection and scalar product is heavily dependent on the way in which the input signal information is encoded. In other word, it is fundamental to keep into account the *signal representation*.

The aim of this section is to review both standard and innovative approaches to examine practical problems that can arise in hardware implementations of CS-based acquisition systems and present an overview of solutions employed in integrated circuits presented so far in the literature. Referring to the above example, a brief survey is enough to identify

the many different possibilities in which the linear projections $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ are computed according to the different input signal model. The solution for this and other issues usually requires the adaptation of the CS framework either at circuital or even at system level.

Regrettably, a rapid scan of the recent literature on CS related works reveals that among the overwhelming number of works that can be found, only a negligible fraction of them deals with the actual circuital implementation of the proposed algorithm or technique.

To the best of authors' knowledge, the first prototype capable of implementing a CS-based system via the generic $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ product has been presented in [47]. The circuit is a sub-Nyquist rate receiver for radar pulse signal designed in $90\,\mathrm{nm}$ technology, capable to acquire signals up to $2\,\mathrm{GHz}$. In [48], the authors describe a CS-based data acquisition front-end for a radio frequency (RF) communication system implemented in $90\,\mathrm{nm}$ CMOS process. The work in [49] presents an analog front-end for ECG signals designed in $180\,\mathrm{nm}$ CMOS process, while [50] reports an area and power efficient multi-electrode arrays acquisition system based on CS designed in $180\,\mathrm{nm}$ CMOS process, outperforming previously presented works in terms of compression rate and reconstruction quality by a run-time adaptation. The work [51] describes a low-power sub-Nyquist sampler for the multichannel acquisition of cortical intracranial electroencephalographic (iEEG) signals. The peculiarity of this architecture, which has been fabricated in $180\,\mathrm{nm}$ CMOS process, is to consider the signal features not only in the temporal domain, but also in the spatial domain. The architecture presented in [37], designed in $180\,\mathrm{nm}$ CMOS process, is an analog-to-information converter for generic biomedical signals. It introduces a smart saturation checking mechanism with which it is possible to reconstruct the acquired signal even if many measurements suffer saturation, and exploits the aforementioned pow-R approach introduced in Section III-B. In [52], the authors propose a run-time signal evaluation module (indicated as *dynamic knob*) designed in $130\,\mathrm{nm}$ CMOS process, with the aim of improving the quality of the following CS encoder by adapting a few parameters towards the input biosignal dynamics. Finally, in [53], the complexity of the VLSI implementation of the generator of the sensing matrix $\boldsymbol{A}$ is investigated, and the hardware efficient generation of deterministic sparse sensing matrices is considered.

A. Computation of compressed measurements

All aforementioned implementations share the same philosophy: in order to compute m different measurements, the very same hardware block is either replicated m times or used in an interleaved way m times, and driven by the input signal $\boldsymbol{x}$ and the m different instances corresponding to the rows of $\boldsymbol{A}$. For the sake of simplicity, in the following we will focus on the scalar product $y = \boldsymbol{a}^\top \boldsymbol{x}$, where $\boldsymbol{a}$. is a generic row of $\boldsymbol{A}$ and y the corresponding element of the measurement vector.

A first group of works [47], [48], [49], [51], [37], [54] deal with an *analog* input signal represented as a function $x(t)$ of the time variable t. As an example, $x(t)$ may be the output of a sensor (typically a voltage), or, more frequently, $x(t)$ is differentially encoded, i.e., it is represented as the difference between two (voltage) quantities as $x(t) = x^+(t) - x^-(t)$. In a few cases, as in [51], the authors consider a multi-channel scenario, and the input signal $\boldsymbol{x}(t)$ is actually an array of p signals coming from different sensors.

For the sake of simplicity, in the following we limit ourselves to the scalar case regardless of the fact that the actual implementation of x is differential or single-ended.

When dealing with an analog signal, two issues immediately arise if one thinks about the computation of $y = \boldsymbol{a}^\top \boldsymbol{x}$: $i)$ according to the standard CS model, $\boldsymbol{x}$ is assumed to be a vector, while in a real implementation the best fit for the input signal is a function of time $x(t)$; $ii)$ $\boldsymbol{x}$ is n-dimensional, while $x(t)$ is defined over the whole $\mathbb{R}$, i.e., it has an infinite dimensionality.

The standard solutions adopted to reconcile the different signal models are sketched in Figure 9. The common way to cope with dimensionality is by *windowing* $x(t)$ [47], [48], [49], [37]. The input signal is sliced to get the functions $x^{(l)}(t), x^{(l+1)}(t), x^{(l+2)}(t), \ldots$, each of them defined on contiguous and non overlapping time intervals $I^{(l)}, I^{(l+1)}, I^{(l+2)}, \ldots$, and such that $x^{(l)} : I^{(l)} \mapsto \mathbb{R}$. The l-th slice of the input signal $x^{(l)}(t)$ gives rise to measurements $\boldsymbol{y}^{(l)}$, that are used to reconstruct $\hat{x}^{(l)}(t)$. The complete input signal can then be achieved by joining all the reconstructed slices.

Conversely, coping with the first issue has not a unique solution. The most general approach is that adopted by [47], [48], where the generic measurement y is achieved by a continuous time *multiply-and-integrate* architecture as in the "analog continuous-time" case of Figure 9. The l-th slice of the input signal $x^{(l)}(t)$ is first multiplied by a sensing function $a(t)$ (assumed defined over $I^{(l)}$) and then integrated over $I^{(l)}$. Focusing for simplicity on the case $l = 0$ where $I^{(0)} = [0, T_w]$, the measurement is expressed as

$$y^{(0)} = \int_0^{T_w} a(t) x^{(0)}(t) dt \tag{9}$$

It is interesting to notice that also this case can be easily incorporated into the standard framework $y = \boldsymbol{a}^\top \boldsymbol{x}$ under the reasonable assumption that $a(t)$ is generated starting from n coefficients $\boldsymbol{a}_k$ stored into a local memory as the pulse-amplitude modulated (PAM) function $a(t) = \sum_{k=0}^{n-1} \boldsymbol{a}_k g(t \cdot n/T_w - k)$, being $g(\cdot)$ a normalized pulse[5].

By replacing the definition of $a(t)$ in (9), we get

$$\begin{aligned} y &= \sum_{k=0}^{n-1} \boldsymbol{a}_k \int_0^{T_w} g\left(t\frac{n}{T_w} - k\right) x^{(0)}(t) dt \\ &= \sum_{k=0}^{n-1} \boldsymbol{a}_k \tilde{\boldsymbol{x}}_k^{(0)} = \boldsymbol{a}^\top \tilde{\boldsymbol{x}}^{(0)} \end{aligned}$$

where we have implicitly defined a generalized Nyquist-rate samples vector $\tilde{\boldsymbol{x}}^{(0)} = \left[\tilde{\boldsymbol{x}}_0^{(0)}, \tilde{\boldsymbol{x}}_1^{(0)}, \ldots, \tilde{\boldsymbol{x}}_{n-1}^{(0)}\right]$ where samples

[5] A typically, but not necessary, choice for $g(\cdot)$ is the rectangular pulse

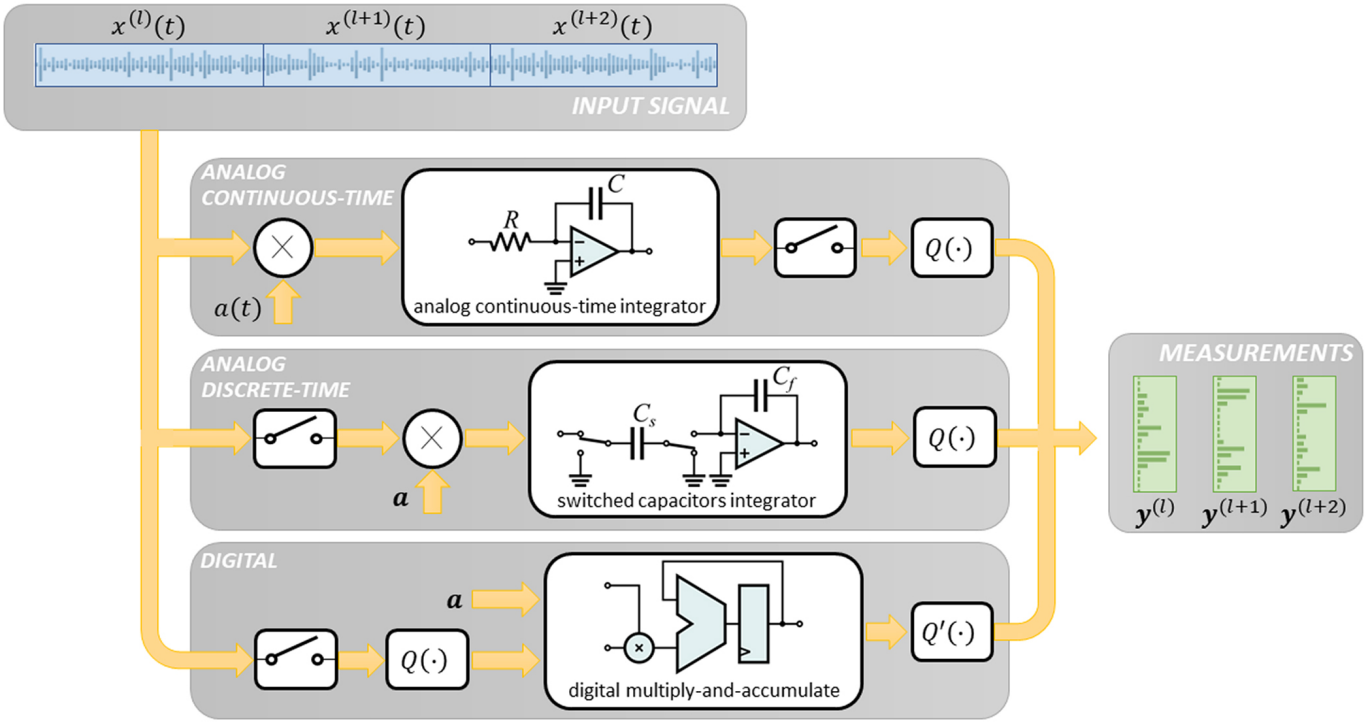

Fig. 9. Grand view of a possible actual implementation of a CS-based acquisition system. The input signal $x(t)$ is first windowed into non overlapping slices $x^{(l)}(t)$, $x^{(l+1)}(t)$, $x^{(l+2)}(t)$, ..., that generate the measurements $y^{(l)}$, $y^{(l+1)}$, $y^{(l+2)}$, ... respectively. The way how measurements are computed depends on the input signal. For high frequency signals, the preferred solution is to mix $x(t)$ with $a(t)$ and then integrate the result with a continuous-time integrator. Results are then quantized by $Q(\cdot)$. For low frequency signals, using a switching capacitors integration approach is a more common choice. Another possible solution for low frequency signals is the fully digital approach, where $x(t)$ is first sampled, quantized by $Q(\cdot)$, processed by a standard multiply-and-accumulate architecture in the digital domain, and optionally re-quantized by $Q'(\cdot)$.

are taken are take at Nyquist rate but their amplitude is related to

$$\hat{\boldsymbol{x}}_k^{(0)} = \int_0^{T_w} g\left(t\frac{n}{T_w} - k\right) x^{(0)}(t)$$

Note that, in many practical cases, $\tilde{\boldsymbol{x}}$ is not so different from the Nyquist rate sample vector $\boldsymbol{x}$. In fact, we have $\tilde{\boldsymbol{x}} = \boldsymbol{x}$ when $g(\tau)$ is the standard Dirac delta operator $\delta(\tau)$. Yet, in practical implementations, it is common to replace $\delta(\tau)$ with a normalized pulse equal to the ideal rectangular pulse $\chi(\tau) = 1$ when $0 \leq \tau < 1$, and $\chi(\tau) = 0$ elsewhere. In this case the generalized coefficient vector can be considered a good approximation of Nyquist-rate sample vector, i.e., $\tilde{\boldsymbol{x}} \approx \boldsymbol{x}$ if the input signal is quasi-stationary. Note also that the CS reconstruction procedure will retrieve the generalized sample vector $\tilde{\boldsymbol{x}}$.

Moreover, if the input signal is low frequency, different approaches are possible. A solution adopted in [49], [51], [37] is to consider a switched capacitor architecture. In this case, we are dealing with a *discrete-time* input signal model thanks to the intrinsic sampling capabilities guaranteed by this class of circuits. The generic l-th time windows of length T_w, defined in the interval $I^{(l)}$ is sampled at rate n/T_w to generate a n-length analog samples vector. Focusing, for the sake of simplicity, on $I^{(0)} = [0, T_w]$, we get

$$\boldsymbol{x}^{(0)} = \left[x^{(0)}(0),\ x^{(0)}\left(\tfrac{T_w}{n}\right), \ldots, x^{(0)}\left((n-1)\tfrac{T_w}{n}\right)\right]^\top$$

The Nyquist samples of the input signal are then processed with a standard *multiply-and-accumulate* (MAC) analog architecture, shown in the "analog discrete-time" case of Figure 9, where a clock signal is assumed whose period is divided into two phases.

In the first one, each sample of $\boldsymbol{x}_k^{(0)} = x^{(0)}(kT_w/n)$ is multiplied by $\boldsymbol{a}_k$, and the results sampled by the C_s. In the second phase, all the charge stored in C_s is transferred to the feedback capacitor C_f, that accumulates the voltage level to be provided as circuit output. After n integration time steps, and assuming $C_s = C_f$, we get at the circuit output

$$\sum_{k=0}^{n-1} \boldsymbol{a}_k \boldsymbol{x}_k^{(0)} = \boldsymbol{a}^\top \boldsymbol{x}^{(0)} = y^{(0)}$$

that is the measurement associated to the generic row $\boldsymbol{a}^\top$ for the time windows I^0.

As final step, in both the considered cases (i.e., when $x(t)$ is high-frequency and measurements are computed by means of a continuous-time integrator [47], [48], and when $x(t)$ is low-frequency and measurements are achieved with a switched capacitor integrator [49], [51], [37]) y is finally converted in digital words by a proper ADC (represented by the $Q(\cdot)$ function of Figure 9) operating at a sub-Nyquist rate[6].

[6]While the Nyquist rate is defined as n/T_w, with this solution the ADC is working at a rate given by $1/T_w$, or m/T_w in case a single shared ADC is used to convert all m measurements.

Another group of works [50], [52] assume to operate directly on *digital* input signals. This approach has recently been receiving increasing attention, and aims at using CS as an early digital processing stage replacing complex and expensive (either in terms of required energy or hardware resources) classic compression algorithms. The corresponding architecture is shown as the "digital" case in Figure 9. The vector $\boldsymbol{x}$ is made of digital words after windowing, sampling and quantizing the input signal $x(t)$. In order to compute y, it is enough to process $\boldsymbol{x}$ with a common digital MAC architecture. Even if, in this case, the measurement y is already a digital quantity that can be delivered "as is" to the reconstruction algorithm, it is a common practice to apply an additional re-quantization function (such as the $Q'(\cdot)$ in the figure) to ensure, for example, a better adaptation to the statistics of y.

B. Multiplication by $\boldsymbol{a}_k$

Notwithstanding the actual implementation as a multiply-and-integrate [47], [48], or a as multiply-and-accumulate stage [49], [50], [51], [37], [53], one of the main difficulties in realizing a CS signal acquisition stage is multiplying the input signal by the sensing sequence/function. In fact, a multiplier is one of the most complicated circuital block in a signal processing chain, both in the analog and in the digital domain.

To tackle this issue, almost all the considered works constrains the elements of vectors $\boldsymbol{a}_k$ to assume a very limited number of values. In [47], [48], [49], [37], it is required that $\boldsymbol{a}_k \in \{-1, +1\}$. The advantage of this approach is clear: the multiplier block can be replaced by a simple sign inversion circuit. More specifically, in the analog domain, and assuming a differential encoding for $x(t)$ [47], [48], [49], [51], [37], a few pass transistors capable of exchanging the $x^+(t)$ and $x^-(t)$ line are enough to perform multiplication by -1. In the digital domain, the solution is similarly simple since a straightforward two's complement allows a multiplication by -1. The multiplication by $+1$ is, of course, trivial in both cases.

Another possible solution is to ask that $\boldsymbol{a}_k \in \{0, +1\}$ [50], [51], [53]. In this case the situation is even simpler, since the multiplication by $+1$ or 0 is simply achieved by allowing the input signal to be summed/integrated or by disconnecting it from the rest of the circuit, respectively.

When multiplication by an arbitrary value is desired, the resulting circuit complexity is expected to substantially increase and several authors have proposed remedies to cope with this. As an example, in the discrete-time analog input signal case, the authors of [49] describe a solution to achieve the multiplication by a 6-bit integer value at virtually no cost in terms of energy. More specifically, instead of relying on a simple switched capacitor integrator based on two capacitors such that shown in the "analog discrete-time" case of Figure 9, they replace C_s with a 5-bit C-2C split capacitor array circuit typically used in digital-to-analog converter. The effect is that only a part (that is proportional to the 5-bits control value) of the charge accumulated on the C-2C split capacitor array is transferred to the C_f, thus performing at the same time both a multiplication and the integration without any additional active device. The 6-th bit of the control word is used as sign bit, and decides if the signal to be integrated is $x(t)$ or $-x(t)$ by exchanging the $x^+(t)$ and $x^-(t)$ differential lines.

C. Time continuity

In the windowing approach illustrated so far, the input signal $x(t)$ is sliced with respect to contiguous and non overlapping time intervals of length T_w. From a circuital point of view, let us refer again to Figure 9 and consider the time interval $I^{(l)}$. The measurement $y^{(l)}$ is computed by considering the input signal slice $x^{(l)}(t)$ and $\boldsymbol{a}$. At the end of $I^{(l)}$, $y^{(l)}$ is available, and can be converted into a digital word (or requantized assuming a digital architecture). After that, the integrator circuit have to be reset in order to be ready to start the computation of a new measurement $y^{(l+1)}$. Note that these two operations (conversion to digital word/requantization and reset) may require a non-negligible amount of time.

Yet, at the same time instant when $I^{(l)}$ ends, $I^{(l+1)}$ starts and all the aforementioned processes need to be repeated on the successive slice $x^{(l+1)}(t)$ of the input signal to compute $y^{(l+1)}$. While it is reasonable to assume that both $y^{(l)}$ and $y^{(l+1)}$ should be computed by the same hardware block, it is straightforward to understand that the two above requirements are conflicting: i) a certain amount of time is needed for the hardware block to be ready for start the computation of the new measurement, and ii) the computation of the new measurement should start immediately after the end of the elaboration of the previous one.

While the solution for a digital architecture is trivial (a latch is enough to store the accumulated value so that the multiply-and-accumulate block is ready for the next computation), the situation is more complicated when considering analog hardware. In the latter case, a complex (and energy-hungry) analog sample/hold should in principle be required. To avoid it, [48] and [37] propose two similar solutions to be applied to the continuous-time and the discrete-time model, respectively.

Referring to the "analog continuous-time" case of Figure 9, in [48] the proposed multiply-and-integrate block is composed by a single multiplication block and by two integration paths. When the first block is integrating, the second one is disconnected from the multiplier and is working as a sample/hold block. This allows enough time to the cascade circuit to digitize the results, and also to clear the charge accumulated on C_f before starting a new integration. At the end of the time window, the role of the two paths is exchanged: the first one works as a sample/hold, while the second path starts a new integration phase for computing the next measurement.

An even simpler solution is presented in [37], where the proposed design is based on a switched capacitor architecture. Referring to the circuit sketched in the "analog discrete-time" case of Figure 9, the authors created two integration paths by replicating only the feedback capacitors C_f, while all other element (the multiplier, C_s and the active amplifier) are shared. The working principle is exactly the same as in the previous case: when the first C_f is used for computing the actual measurement, the second C_f is sampling the previous

measurement and may be connected to the cascading analog-to-digital converter, leaving enough time for the conversion and for removing the charge accumulated on it. At the end of each time window, the role of the two capacitors is exchanged. Note also that the further advantage of this solution is that no active (and so, energy consuming) devices are replicated.

D. Saturation

Independently of the architecture, measurements must be quantized/requantized before dispatching them. Unfortunately, such an operation may lead to saturation problems. Even if this has been rarely considered, it is a severe issue both for analog and digital CS implementations.

Since $y = \sum_{k=0}^{n-1} \boldsymbol{a}_k \boldsymbol{x}_k$, and assuming a large n (or more precisely, assuming that the number of terms $\boldsymbol{a}_k \boldsymbol{x}_k \neq 0$ is large), the central limit theorem can be applied to the sum resulting into y. As a consequence, the distribution of the result is expected to be approximately Gaussian, so that y may indeed assume very large values, while the majority of the observed cases will be practically located around the mean value.

This is an important issue for a twofold reason: i) the applied quantization function is uniform, i.e., all quantization steps have the same size, and ii) the conversion range is limited by an upper and a lower threshold, which identify the interval where conversion is correctly preformed, while outside it saturation occurs.

When an analog CS architecture needs to be implemented, the two above mentioned quantization thresholds must carefully be selected. In fact, it is obvious that using very different values can decrease (ideally, down to zero) the probability of a saturation event. This has however the drawback, due to the large quantization step, to increase the quantization error and reduces the reconstruction quality. Conversely, making the values of the two thresholds closer to each other results in a lower quantization error, but increases to a non-negligible value the probability of a saturation event.

A similar problem has to be considered also when computing y by means of a digital multiply-and-accumulate block. Unless a complex and non-efficient floating point representation is used, a high number of bits in the representation of y will result in negligible probability of saturation. The drawback is the non-efficient coding, since the most significant bits will hardly ever be used. It is worth stressing that a non-efficient coding of measurements is actually a very serious problem, since it is in contrast with the ambition of CS to work as an efficient compression algorithm.

As an additional problem, saturation may actually be observed at *any time* in the computation of y. Let us define the succession of values

$$y[j] = \sum_{k=0}^{j} \boldsymbol{a}_k \boldsymbol{x}_k, \quad j = 0, 1, \ldots, n-1 \tag{10}$$

with $y = y[n-1]$. This is the sequence of all intermediate partial sums that lead to y. Considerations similar to those mentioned above for y, hold also for each $y[j]$. When analog hardware is the choice, the integrator computing (10) may saturate (i.e. compromising the correct operation of the integrator in its linear region). Similarly, in a digital architecture, the multiply-and-accumulate block may overflow. In both cases, the outcome of the computation (i.e., the final y) is unpredictable. Using this value for reconstructing the input signal leads to an error that will impair the signal reconstruction.

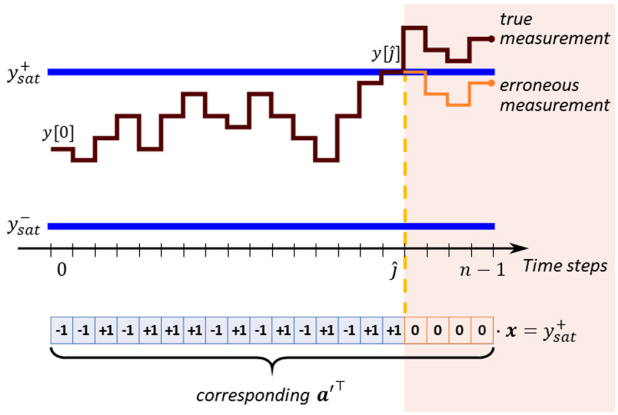

Fig. 10. During the computation of a generic measurement y, when the intermediate accumulated value $y[j]$ leaves the safe interval $\left[y^-_{sat}, y^+_{sat}\right]$, a non-linear phenomenon occurs (saturation, overflow, etc.). In this case, the final accumulated value may be different from the expected one. Stopping the integration at time step $\hat{j}$ when the safe region is left is a smart solution to get a non-erroneous measurement.

A workaround for this problem was first proposed in [55] and adopted in [37] for the "analog discrete-time" case, and can be easily extended to any other architecture. The authors added two analog comparators to the multiply-and-integrate block with the aim of checking whether $y[j]$, $\forall j \in \{0, 1, \ldots, n-1\}$ is in a safe range of linearity of the integrator block, defined by an upper threshold y^+_{sat} and a lower one y^-_{sat}. As soon as this region is left, the time step $\hat{j}$ is recorded along with the fact that the threshold being reached is y^+_{sat} or y^-_{sat}. These data are transmitted to the reconstruction algorithm instead of the corrupted measurement.

More specifically, it is reasonable to assume that the expected value of each step $\boldsymbol{a}_k \boldsymbol{x}_k$ is small with respect to the integrator linearity range. In this way, the probability of a saturation event would be limited even for n large. As a consequence, we have either

$$\sum_{k=0}^{\hat{j}} \boldsymbol{a}_k \boldsymbol{x}_k \approx y^-_{sat} \quad \text{or} \quad \sum_{k=0}^{\hat{j}} \boldsymbol{a}_k \boldsymbol{x}_k \approx y^+_{sat} \tag{11}$$

By introducing a sensing vector $\boldsymbol{a}' \in \mathbb{R}^n$ defined as

$$\boldsymbol{a}' = \left[\boldsymbol{a}_0, \boldsymbol{a}_0, \ldots, \boldsymbol{a}_{\hat{j}}, 0, \ldots, 0\right]^{\top}$$

the two expressions in (11) can be written as $\boldsymbol{a}'^{\top}\boldsymbol{x} \approx y^-_{sat}$ or $\boldsymbol{a}'^{\top}\boldsymbol{x} \approx y^+_{sat}$, respectively, and can be used in the reconstruction algorithm to replace the erroneous $y = \boldsymbol{a}^{\top}\boldsymbol{x}$. This approach is schematically represented in Figure 10.

The advantage of this solution with respect to the simple approach where measurements characterized by a saturation event are marked as invalid and not used in the reconstruction

algorithm [56] can be explained by considering that the philosophy underlying CS is to reconstruct the input signal with the minimum amount of information. Accordingly, being able to recover even a small quantity of information from saturated measurements is an advantage. The authors of [37] showed that, using this approach, when a small ratio of measurements present saturation, it is still possible to recover the input signal with the same quality one get when no saturation events occur. Furthermore, they also showed that when up to 60% of the measurements are characterized by a saturation event, it is still possible to reconstruct the input signal with an almost negligible drop in performance.

E. Generation of sensing sequences

The generation of $\boldsymbol{a}$ appears as a straightforward operation in a CS-based system. Nevertheless, when dealing with a practical CS systems implementation, this operation surprisingly requires appreciable care.

While in test prototypes it is commonly allowed to externally generate and load the sensing sequences [47], [37], any IoT CS-based not must have the elements of $\boldsymbol{a}$ available on-chip. With respect to this, it is immediate to conclude that storing all the sensing sequences in an internal memory is in general not an option. In fact, referring for instance to [49] where $n = 256$, $m = 64$, and each entry of the sensing matrix is a 6-bit quantized value, storing all different elements of all vectors $\boldsymbol{a}$ would require 100-Kbit of dedicated memory, something that should be avoided in the implementation of an analog circuit. So the path to follow is the on-chip generation of the sensing sequences.

Yet, as extensively shown in this paper, CS performance strongly depends on the choice of the acquisition sequences. We have seen that sensing sequences need to be randomly drawn or generated by adopting one of the discussed approaches. While the complexity of the second solution is obvious, the first case may present drawbacks when using a simple linear feedback shift register (LFSR) due to the low quality of the generated stream. The problem is relevant, in particular, if the number of channels is high, and so the number of different elements in $\boldsymbol{a}$ to be generated at the same time.

While in [48], [51] the generation of sequences $\boldsymbol{a}$ is achieved by a simple LFSR with no additional details, in [49] a complex Fibonacci—Galois 384-bit LFSR is designed. Basically, 64 6-bit Fibonacci LFSRs have been integrated into the circuit, each one generating a different $\boldsymbol{a}$. Then, the 64 LFSRs are further randomized by dithering their less significant bits in a Galois fashion, each LFSR using the most significant bits of another stage. An external trigger signal enables a 384-bit seed load at the beginning of each integration window[7].

A completely different problem was instead faced in [47]. The proposed integrated circuit is a sub-Nyquist sampler for a 2 GHz bandwidth input signal. The circuit has a continuous-time analog architecture, and uses PAM sampling signal $a(t)$ obtained from antipodal sequences $\boldsymbol{a}$, where the multiplication by $a(t)$ is simply achieved by exchanging the differential lines of the differential input signal. Due to the 2 GHz input signal bandwidth, the $\boldsymbol{a}_k$ symbols must be generated at a rate equal to 4 Gbit/s. The use of an internal serial memory for storing the $\boldsymbol{a}$ vector, built upon a programmable shift register, even with all its implementation drawbacks, has been found to be the only solution allowing versatility at this speed.

The problem of the efficient generation of the elements of $\boldsymbol{a}$ has also been considered in [53]. The authors propose a simple and deterministic algorithm to generate binary vectors $\boldsymbol{a}$ (i.e., $\boldsymbol{a}_k \in \{0, 1\}$) that, once collected into the sensing matrix $\boldsymbol{A}$, ensures that $i)$ $\boldsymbol{A}$ satisfies the theoretically requirements for input signal reconstruction at the decoder side; and $ii)$ $\boldsymbol{A}$ is easily obtained with a finite state machine.

The method proposed in [53] relies on the quasi-cyclic array code based binary matrix framework. In particular, the authors aim to utilize the parity-check matrices of array codes and their submatrices to construct $\boldsymbol{A}$. In more details, let us indicate with $\boldsymbol{I}_q$ the $q \times q$ identity matrix, and $\boldsymbol{P}_q$ the $q \times q$ cyclic permutation matrix defined as

$$\boldsymbol{P}_q = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Given an integer r, the sensing matrix $\boldsymbol{A}$ is the $rq \times q^2$ binary matrix given by

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{I}_q & \boldsymbol{I}_q & \cdots & \boldsymbol{I}_q \\ \boldsymbol{I}_q & \boldsymbol{P}_q & \cdots & \boldsymbol{P}_q^{q-1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{I}_q & \boldsymbol{P}_q^{r-1} & \cdots & \boldsymbol{P}_q^{(r-1)(q-1)} \end{bmatrix}$$

According to [53], the proposed approach shows comparable recovery performance for EEG and spike data compression with respect to standard approach at a reduced hardware complexity.

F. The spatio-temporal approach

In the works [50], [51] authors consider a multichannel EEG recording as input. A multichannel signal may be modeled as an array $\boldsymbol{x}(t)$ composed of p real functions, with $\boldsymbol{x} : \mathbb{R} \mapsto \mathbb{R}^p$. By windowing it and sampling it at rate T_w/n, we get for each $I^{(l)}$, a slice of signal represented by a $p \times n$ matrix, that can be easily unrolled to get a $p \times n$ vector. The standard CS framework can then be applied to this vector with no other modification.

Indeed, a spatio-temporal approach may lead to several advantages with respect to a standard approach, since it make possible to exploit the input signal features in two different domains (i.e. spatial and temporal one).

This is an open research topic. Yet, by limiting ourselves to a pure circuital level consideration, it interesting to see how in [51] this model is used to reduce the hardware complexity of $y = \boldsymbol{a}^\top \boldsymbol{x}$. In fact, at each time step, a number

[7]Note that, even if [49] is the only work where the elements of each $\boldsymbol{a}$ are the approximations of real quantities, the authors considers only uniformly distributed random values due to the complexity of generating a Gaussian distribution at hardware level.

of measurements is computed as a linear combination of the samples coming from all input signal channels sampled at that time step. Mathematically, at time step k, we get the generic measurement

$$y = \sum_{j=0}^{p-1} \boldsymbol{a}_j x^{(j)} \left(k \frac{T_w}{n} \right) \tag{12}$$

Then, all measurements generated at the k-th time step are collected by the reconstruction algorithm and joined to all measurements generated for all other time steps belonging to $I^{(l)}$. During reconstruction, both sparsity (on the time domain) and correlation (on the spatial domain) are used to improve the input signal reconstruction quality.

Computing measurements with a multiply-and-accumulate operation in the spatial domain only as in (12) has a twofold advantage.

On the one hand, this avoids the time continuity problem described so far. At a generic time step k the input signal is sampled, and each sample is multiplied by $\boldsymbol{a}_j$ and accumulated to get y as in (12). The measurements have to be quantized and dispatched before the end of the time step. This approach imposes time constraints tighter with respect to a standard approach and due to the necessity to deal at the same time with p different input signals (the time available for all the required operations is T_w/n instead of T_w). However, at the end of the time windows all measurements have already been converted and dispatched, and no additional actions are required between the end of a time windows and the beginning of the next one.

On the other hand, this approach makes the computation of y independent of the length of T_w. As observed in [37], a T_w that is too long (that is actually a common situation for biomedical signals such as ECG or EEG) may lead to leakage problems in analog implementation due to the discharge of the capacitors. In the light of this, an architecture where performance of the integrator is independent of T_w is a sure advantage.

G. Dynamic CS approaches

In many cases, it may be useful to tune some parameters of a CS system on the particular input signal. For example, in biomedical signal acquisition, the dictionary $\boldsymbol{D}$ used for reconstruction can be trained on the particular patient for improving performance. The training phase, however, cannot be done by using compressed measurements, but needs to be established before the starting of the CS operating mode.

Such an approach is considered in [50]. The interesting aspect from the circuital point of view is that the designed system works with a feedback signal generated by the reconstruction algorithm capable of enabling or disabling the CS operating mode. Initially, the CS mode is disabled (i.e., the device is working as a standard Nyquist-rate converter) and the uncompressed input EEG signal is sent to the receiver side which analyses it. When enough data is collected to allow a good estimation of a trained dictionary, a signal is sent to the acquisition device to start the CS operating mode. [8]

A different case is taken into account in [52], where no CS architectures are considered. Indeed, the design of a low-power companion chip capable of detecting features of the input signal is presented. The aim of the work is to identify the type of the input signal among many possible biosignals. In this way, it should be possible to adapt parameters of the CS (for example, the number of measurements m, the time windows T_w, the number of samples n per time window, and so on) to that of the input signal, with advantages in terms of reconstruction quality and/or energy required to sample the input signal.

VIII. Conclusion

CS is often thought of as a technique in which adaptation cannot play a significant role. Yet, a careful scan of the literature reveals that many methods have been developed to adjust either the encoding or the decoding side of CS depending on the class of signals that has to be acquired. Moreover, every hardware implementation has to adapt the general paradigm to cope with realization constraints or resource budgets.

Our review shows that these designs are able to yield significant improvements that cannot be predicted by classical theoretical guarantees dealing with worst-case analysis.

Yet, the possibility of substantially increasing the compression ratio while obeying to the same requirements in terms of quality of the recovered signal, is a key point to allow the effective introduction of CS in application in which resources must be carefully administrated. From this point of view, adapted CS is definitely worth pursuing whenever designing the low-resources, autonomous, ubiquitous sensing subsystems that will be the backbone of future IoT applications.

References

[1] R. G. Baraniuk, E. Candes, R. Nowak, and M. Vetterli, "Compressive sampling [from the guest editors]," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 12–13, March 2008.

[2] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

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

[4] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[5] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, Mar. 1997.

[6] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, May 2009.

[7] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 629–654, Dec. 2008.

[8] Z. Zhang and D. B. Rao, "Extension of sbl algorithms for recovery of block sparse signals with intra-block correlation," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 2009–2015, Apr. 2009.

[8] Note also that the trained dictionary characterizes the first of two phases of the proposed decoding procedure. Here, the main signal shape is approximately reconstructed by using a single atom from the trained dictionary. In the second phase, signal details are recovered using Daubechies wavelet transformation as sparsity basis.

[9] F. Pareschi, M. Mangia, D. Bortolotti, A. Bartolini, L. Benini, R. Rovatti, and G. Setti, "Energy analysis of decoders for rakeness-based compressed sensing of ecg signals," *IEEE Transactions on Biomedical Circuits and Systems*, 2017.

[10] M. Mangia, F. Pareschi, V. Cambareri, R. Rovatti, and G. Setti, *Adapted compressed sensing for effective hardware implementations: A design flow for signal-level optimization of compressed sensing stages*. Springer, 2018.

[11] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2456–2466, Sept 2011.

[12] D. Bortolotti, M. Mangia, A. Bartolini, R. Rovatti, G. Setti, and L. Benini, "An ultra-low power dual-mode ecg monitor for healthcare and wellness," in *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1611–1616, Mar. 2015.

[13] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9, pp. 589–592, May 2008.

[14] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *Information Theory, IEEE Transactions on*, vol. 47, no. 7, pp. 2845–2862, Nov. 2001.

[15] L. Welch, "Lower bounds on the maximum cross correlation of signals (corresp.)," *IEEE Transactions on Information Theory*, vol. 20, no. 3, pp. 397–399, May 1974.

[16] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Transactions on Information Theory*, vol. 49, no. 12, pp. 3320–3325, Dec. 2003.

[17] A. A. Goldstein, "Convex programming in Hilbert space," *Bullettin of the americal Mathematical Society*, vol. 70, pp. 709–710, 1964.

[18] J. P. Boyle and R. L. Dykstra, *A Method for Finding Projections onto the Intersection of Convex Sets in Hilbert Spaces*. New York, NY: Springer New York, pp. 28–47, 1986.

[19] J. Duarte-Carvajalino and G. Sapiro, "Learning to Sense Sparse Signals: Simultaneous Sensing Matrix and Sparsifying Dictionary Optimization," *Image Processing, IEEE Transactions on*, vol. 18, no. 7, pp. 1395–1408, 2009.

[20] W. Chen, M. Rodrigues, and I. Wassell, "Projection design for statistical compressive sensing: A tight frame based approach," *IEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 2016–2028, Apr. 2013.

[21] N. Cleju, "Optimized projections for compressed sensing via rank-constrained nearest correlation matrix," *Applied and Computational Harmonic Analysis*, vol. 36, no. 3, pp. 495–507, May 2014.

[22] J. Xu, Y. Pi, and Z. Cao, "Optimized projection matrix for compressive sensing," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, p. 560349, 2010.

[23] T. Strohmer and R. W. Heath Jr, "Grassmannian frames with applications to coding and communication," *Applied and computational harmonic analysis*, vol. 14, no. 3, pp. 257–275, 2003.

[24] H. Bai, S. Li, and X. He, "Sensing matrix optimization based on equiangular tight frames with consideration of sparse representation error," *IEEE Transactions on Multimedia*, vol. 18, no. 10, pp. 2040–2053, Oct. 2016.

[25] J. Kovačević and A. Chebira, "Life beyond bases: the advent of frames - part i," *IEEE Signal processing magazine*, vol. 24, no. 4, pp. 86–104, Jul. 2007.

[26] ——, "Life beyond bases: the advent of frames - part ii," *IEEE Signal processing magazine*, vol. 24, no. 6, pp. 115–125, Sep. 2007.

[27] G. Li, Z. Zhu, D. Yang, L. Chang, and H. Bai, "On projection matrix optimization compressive sensing systems," *IEEE Transactions on Signal Processing*, vol. 61, no. 11, pp. 2887–2898, Jun. 2013.

[28] M. Elad, "Optimized projections for compressed sensing," *IEEE Transactions on Signal Processing*, vol. 55, no. 12, pp. 5695–5702, Dec. 2007.

[29] S. Pazos, M. Hurtado, C. H. Muravchik, and A. NEhorai, "Projection matrix optimization for sparse signals in structured noise," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 3902–3913, Aug. 2015.

[30] M. Mangia, R. Rovatti, and G. Setti, "Rakeness in the design of analog-to-information conversion of sparse and localized signals," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 5, pp. 1001–1014, May 2012.

[31] M. Mangia, F. Pareschi, V. Cambareri, R. Rovatti, and G. Setti, "Rakeness-based design of low-complexity compressed sensing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 5, 2017.

[32] R. Rovatti, G. Mazzini, and G. Setti, "Enhanced rake receivers for chaos-based ds-cdma," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 7, pp. 818–829, July 2001.

[33] G. Setti, R. Rovatti, and G. Mazzini, "Performance of chaos-based asynchronous ds-cdma with different pulse shapes," *IEEE Communications Letters*, vol. 8, no. 7, pp. 416–418, July 2004.

[34] R. Rovatti, G. Mazzini, and G. Setti, "Memory-m antipodal processes: spectral analysis and synthesis," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 1, Jan. 2009.

[35] A. Caprara, F. Furini, A. Lodi, M. Mangia, R. Rovatti, and G. Setti, "Generation of Antipodal Random Vectors With Prescribed Non-Stationary 2-nd Order Statistics," *IEEE Transactions on Signal Processing*, vol. 62, no. 6, pp. 1603–1612, Mar. 2014.

[36] D. E. Bellasi, R. Rovatti, L. Benini, and G. Setti, "A low-power architecture for punctured compressed sensing and estimation in wireless sensor-nodes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 5, pp. 1296–1305, May 2015.

[37] F. Pareschi, P. Albertini, G. Frattini, M. Mangia, R. Rovatti, and G. Setti, "Hardware-Algorithms Co-Design and Implementation of an Analog-to-Information Converter for Biosignals Based on Compressed Sensing," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 1, pp. 149–162, Feb. 2016.

[38] E. van den Berg and M. P. Friedlander, "SPGL1: A solver for large-scale sparse reconstruction," Jun. 2007, http://www.cs.ubc.ca/labs/scl/spgl1.

[39] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *2011 IEEE International Symposium on Information Theory Proceedings*. IEEE, Jul. 2011, pp. 2168–2172.

[40] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, Jan. 2010.

[41] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.

[42] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.

[43] J. Zhang, Z. Gu, Z. L. Yu, and Y. Li, "Energy-efficient ecg compression on wireless biosensors via minimal coherence sensing and weighted ℓ_1 minimization reconstruction," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 2, pp. 520–528, Mar. 2015.

[44] A. Marchioni, M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "Low-complexity greedy algorithm in compressed snesing for the adapated decoding of ecgs," in *IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2017, pp. 1–4.

[45] P. E. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith, "A dynamical model for generating synthetic electrocardiogram signals," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 3, pp. 289–294, Mar. 2003.

[46] Y. Zigel, A. Cohen, and A. Katz, "The weighted diagnostic distortion (wdd) measure for ecg signal compression," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 11, pp. 1422–1430, nov 2000.

[47] J. Yoo, S. Becker, M. Monge, M. L. andEmmanuel Candès, and A. Emami-Neyestanak, "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.

[48] X. Chen, E. A. Sobhy, Z. Yu, S. Hoyos, J. Silva-Martinez, S. Palermo, and B. M. Sadler, "A Sub-Nyquist Rate Compressive Sensing Data Acquisition Front-End," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 542–551, Sep. 2012.

[49] D. Gangopadhyay, E. G. Allstot, A. M. R. Dixon, K. Natarajan, S. Gupta, and D. J. Allstot, "Compressed Sensing Analog Front-End for Bio-Sensor Applications," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 2, pp. 426–438, Feb. 2014.

[50] J. Zhang, Y. Suo, S. Mitra, S. P. Chin, S. Hsiao, R. F. Yazicioglu, T. D. Tran, and R. Etienne-Cummings, "An efficient and compact compressed sensing microsystem for implantable neural recordings," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 4, pp. 485–496, Aug. 2014.

[51] M. Shoaran, M. H. Kamal, C. Pollo, P. Vandergheynst, and A. Schmid, "Compact Low-Power Cortical Recording Architecture for Compressive Multichannel Data Acquisition," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 6, pp. 857–870, Dec. 2014.

[52] A. Wang, F. Lin, Z. Jin, and W. Xu, "Ultra-low power dynamic knob in adaptive compressed sensing towards biosignal dynamics," *IEEE*

Transactions on Biomedical Circuits and Systems, vol. 10, no. 3, pp. 579–592, June 2016.

[53] W. Zhao, B. Sun, T. Wu, and Z. Yang, "On-chip neural data compression based on compressed sensing with sparse sensing matrices," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 1, no. 12, pp. 242–254, Feb. 2018.

[54] J. Haboba, M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "A pragmatic look at some compressive sensing architectures with saturation and quantization," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 443–459, Sep. 2012.

[55] M. Mangia, F. Pareschi, R. Rovatti, G. Setti, and G. Frattini, "Coping with saturating projection stages in RMPI-based Compressive Sensing," in *2012 IEEE International Symposium on Circuits and Systems*, May 2012, pp. 2805–2808.

[56] J. N. Laska, P. T. Boufounos, M. A. Davenport, and R. G. Baraniuk, "Democracy in action: Quantization, saturation, and compressive sensing," *Applied and Computational Harmonic Analysis*, vol. 31, no. 3, pp. 429 – 443, 2011.