IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON THEORY, ALGORITHMS, AND APPLICATIONS OF SPARSE RECOVERY

Received October 31, 2018, accepted December 3, 2018, date of publication December 14, 2018,
date of current version January 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2886471

A Review of Sparse Recovery Algorithms

ELAINE CRESPO MARQUES[1], NILSON MACIEL[1], LÍRIDA NAVINER[1], (Senior Member, IEEE), HAO CAI[1,2], (Member, IEEE), AND JUN YANG[2], (Member, IEEE)

[1]Télécom ParisTech, Université Paris-Saclay, 91405 Paris, France
[2]National ASIC System Engineering Center, Southeast University, Nanjing 210096, China

Corresponding author: Elaine Crespo Marques (ecrespo@telecom-paristech.fr)

This work was supported in part by Télécom ParisTech and in part by the Brazilian Ministry of Defense.

ABSTRACT Nowadays, a large amount of information has to be transmitted or processed. This implies high-power processing, large memory density, and increased energy consumption. In several applications, such as imaging, radar, speech recognition, and data acquisition, the signals involved can be considered sparse or compressive in some domain. The compressive sensing theory could be a proper candidate to deal with these constraints. It can be used to recover sparse or compressive signals with fewer measurements than the traditional methods. Two problems must be addressed by compressive sensing theory: design of the measurement matrix and development of an efficient sparse recovery algorithm. These algorithms are usually classified into three categories: convex relaxation, non-convex optimization techniques, and greedy algorithms. This paper intends to supply a comprehensive study and a state-of-the-art review of these algorithms to researchers who wish to develop and use them. Moreover, a wide range of compressive sensing theory applications is summarized and some open research challenges are presented.

INDEX TERMS
Bayesian compressive sensing, compressive sensing, convex relaxation, greedy algorithms, sparse recovery algorithms, sparse signals.

I. INTRODUCTION

With the increasing amount of information available in the age of big data, the complexity and the cost of processing high-dimensional data systems become very critical [1]. Therefore, developing methods to acquire the previous information of the signals is very important and useful [2].

The Shannon-Nyquist sampling theorem is traditionally used to reconstruct images or signals from measured data. According to this theorem, a signal can be perfectly reconstructed from its samples if it is sampled by the Nyquist rate, that is, a rate equal to twice the bandwidth of the signal [3].

In several applications, especially those involving ultra-wideband communications and digital image, the Nyquist rate can be very high, resulting in too many samples, which makes it difficult to store or transmit them [4]–[6]. Furthermore, it can become unfeasible to implement these scenarios.

Most real signals can be considered sparse or compressive in some domain. Such signals have a lot of coefficients which are equal to or close to zero. For example, many communication signals can be compressible in a Fourier basis, while discrete cosine and wavelet bases tend to be suitable for compressing natural images [7]. Moreover, a sparse representation of a signal allows more efficient signal processing [8].

Compressive Sensing (CS) theory can be very useful when signals are sparse or compressible. This theory was developed by Candès *et al.* [9] and Donoho [10]. CS combines the acquisition and compression processes, exploiting the signals' sparsity. It allows reducing time processing and energy consumption as well as improving storage capacities [5]. While tradition methods use the Nyquist rate, that is, the rate depends on the highest frequency component of the signal, CS relies on sampling rate related to the signal's sparsity.

Researchers have invested effort in developing an efficient algorithm for sparse signal estimation. In fact, there are some survey papers in the literature that address greedy algorithms for sparse signal recovery [11]–[15], measurement matrices used in compressive sensing [16], [17], and compressive sensing applications [18]–[24]. Various works focus on one specific category of sparse recovery algorithm or one specific application of CS theory. Moreover, the recently work [25] reviews the basic theoretical concepts related to CS and the CS acquisition strategies. However, although [25] presents some reconstruction algorithms, it is not focused on them.

Due to the significant amount of literature available, this work aims to review some concepts and applications of compressive sensing, and to provide a survey on the most important sparse recovery algorithms from each category.

TABLE 1. Definitions of acronyms and notations.

Acronym / Notation	Definition	Acronym / Notation	Definition
ADC	Analog to Digital Conversion	NMSE	Normalized Mean Squared Error
AIC	Analog to Information Conversion	OMP	Orthogonal Matching Pursuit
AMP	Approximate Message Passing	RIC	Restricted Isometry Constant
AST	Affine Scaling Transformation	RIP	Restricted Isometry Property
BAOMP	Back-tracking based Adaptive Orthogonal Matching Pursuit	ROMP	Regularized OMP
BCS	Bayesian Compressive Sensing	RVM	Relevance Vector Machine
BOMP	Block Orthogonal Matching Pursuit	SP	Subspace Pursuit
BP	Basis Pursuit	SBL	Sparse Bayesian Learning
BPDN	BP de-noising	SGP	Stochastic Gradient Pursuit
CoSaMP	Compressive Sampling Matching Pursuit	SLSMP	Sequential Least Squares Matching Pursuit
CP	Chaining Pursuit	SpAdOMP	Sparse Adaptive Orthogonal Matching Pursuit
CR	Cognitive Radio	SpaRSA	Sparse Reconstruction by Separable Approximation
CS	Compressive Sensing	StOMP	Stagewise Orthogonal Matching Pursuit
DOA	Direction-of-Arrival	TOMP	Tree-based Orthogonal Matching Pursuit
D-OMP	Differential Orthogonal Matching Pursuit	TSMP	Tree Search Matching Pursuit
DS	Dantzig Selector	VAMP	Vector Approximate Message Passing
EM	Expectation-Maximization	WSN	Wireless Sensors Network
FBP	Forward-Backward Pursuit	$(\cdot)^H$	hermitian transpose matrix
FISTA	Fast Iterative Shrinkage Thresholding Algorithm	$(\cdot)^\dagger$	pseudo-inverse matrix
FOCUSS	Focal Underdetermined System Solution	$\mu()$	coherence
FPGA	Field Programmable Gate Array	$(\cdot)^{-1}$	inverse matrix
GBP	Greedy Basis Pursuit	$S_\tau()$	soft thresholding function defined by (32)
GOAMP	Generalized Orthogonal Adaptive Matching Pursuit	λ	the absolute correlation of the active columns
GOMP	Generalized Orthogonal Matching Pursuit	ϵ	error tolerance
GP	Gradient Pursuit	s	signal's sparsity
HDTV	High-Definition Television	J	set of index selected
HHS	Heavy Hitters on Steroids	N	length of the signal $\mathbf{h}$
IBTMC	Inter-Burst Translational Motion Compensation	M	length of the signal $\mathbf{y}$
IHT	Iterative Hard Thresholding	T	maximum of iterations
IoT	Internet of Things	Λ_i	set of the indices chosen untill iteration i
IRLS	Iterative Reweighted Least Squares	C	constant
ISAR	Inverse Synthetic Aperture Radar	δ_s	the smallest number that achieves RIP
IST	Iterative Soft Thresholding	N_{it}	number of iterations
ISDB-T	Integrated Services Digital Broadcasting-Terrestrial	$\|\|\cdot\|\|_p$	l_p norm - $\|\|\mathbf{x}\|\|_p = \left(\sum_{i=1}^{n} \|\mathbf{x}_i\|^p\right)^{1/p}$
LARS	Least Angle Regression	t_S	threshold value
LASSO	Least Absolute Shrinkage and Selection Operator	γ_i	step size at iteration i
LP	Linear Programming	$\mathbf{h}$	N $\times$ 1 sparse signal to be estimated
LS	Least Square	$\hat{\mathbf{h}}$	estimation of $\mathbf{h}$
MAP	Maximum a posteriori	$\mathbf{\Psi}$	N $\times$ N basis matrix
ML	Maximum Likelihood	$\mathbf{A}$	M $\times$ N measurement matrix
MMP	Multipath Matching Pursuit	$\mathbf{y}$	M $\times$ 1 received signal
MMV	Multiple Measurement Vectors	$\mathbf{b}$	residual vector
MP	Matching Pursuit	$\mathbf{c}$	projection vector
MPLS	Matching Pursuit based on Least Squares	$\mathbf{d}$	direction vector
MRI	Magnetic Resonance Imaging	$< .,. >$	inner product
$H_s()$	sets all components to zero except the s largest magnitude components	$\mathbf{A}(\Lambda_i)$	submatrix of $\mathbf{A}$ containing only those columns of $\mathbf{A}$ with indices in Λ_i

This work focuses on the single measurement vector problem. Other works focused on Multiple Measurement Vectors (MMV) can be found in [26]–[29]. Some open research challenges related to sparse signal estimation are also discussed. The main contributions of this paper are:

- A review of the most important concepts related to compressive sensing and sparse recovery algorithms;
- A list of some applications of CS;
- Comparison of some sparse recovery algorithms;
- Open research challenges.

To facilitate the reading of this paper, Table 1 provides the definitions of acronyms and notations. Moreover, vectors are denoted by bolded lowercase letters, while bolded capital letters represent matrices.

The paper is structured as follows. Section II introduces the key concepts of the CS theory. Its applications are illustrated in Section III. Section IV presents an overview of sparse recovery algorithms. Discussion of these algorithms and some open research challenges are explained in Section V. Finally, conclusions are presented in Section VI.

II. COMPRESSIVE SENSING (CS)

The main idea of compressive sensing is to recover signals from fewer measurements than the Nyquist rate [9], [10]. The underlying assumption is that signals are sparse or compressible by some transforms (e.g., Fourier, wavelets).

A s-sparse signal has only s non-zero coefficients. Otherwise, the coefficients $z(i)$ of a compressible signal

decrease in magnitude according to:

$$|z(I(k))| \leqslant Ck^{-1/r} \quad k = 1, \ldots, n \tag{1}$$

where $I(k)$ is the k^{th} largest component of $\mathbf{z}$ sorted by magnitude from largest to smallest. Due to their rapid decay, such signals can be well approximated by s-sparse signals, keeping just the s largest coefficients of $\mathbf{z}$.

Fig. 1 shows a 200 samples length time-domain signal (Fig. 1(a)) representing 8 distinct sinusoids (Fig. 1(b)). This figure is an example of 8-sparse signal in frequency domain, that is, it can be seen in Fig. 1(b) that only 8 non-zero values exist among the 200 frequencies.

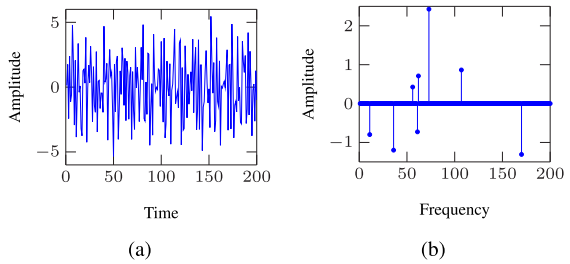

FIGURE 1. Samples of 8 sinusoids in (a) time and (b) frequency domains.

CS allows two main advantages: reduce the energy for transmission and storage through the projection of the information into a lower dimensional space; and reduce the power consumption by reducing the sampling rate to the signal's information content rather than to its bandwidth [10].

CS includes three main steps: sparse representation, CS acquisition (measurement), and CS reconstruction (sparse recovery) as illustrated in Fig. 2 [30], [31].

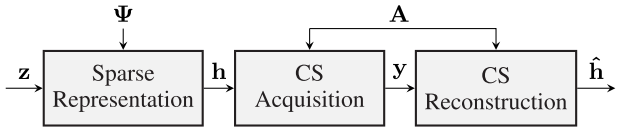

FIGURE 2. Compressive sensing main processes.

In a sparse representation, the signal is represented as a projection on a suitable basis, i.e., a linear combination of only s basis vectors, with $s \ll N$. It means that a signal $\mathbf{z}$ with $N \times 1$ column vector in its original representation can be represented with a basis of $N \times 1$ vectors $\{\boldsymbol{\psi}_i\}_{i=1}^{N}$. Let $\boldsymbol{\Psi}$ be the $N \times N$ basis matrix, the signal can be represented in its sparse form $\mathbf{h}$ by:

$$\mathbf{z} = \boldsymbol{\Psi}\mathbf{h} \tag{2}$$

Next, in the second step (measurement - CS Acquisition), the signal $\mathbf{z}$ is measured by sampling it according to a matrix $\boldsymbol{\Phi} \in \mathbb{C}^{M \times N}$, where $\boldsymbol{\phi}_i$ denotes the i^{th} column of the matrix $\boldsymbol{\Phi}$. The system model is defined by:

$$\mathbf{y} = \boldsymbol{\Phi}\mathbf{z} + \mathbf{n} = \boldsymbol{\Phi}\boldsymbol{\Psi}\mathbf{h} + \mathbf{n} = \mathbf{A}\mathbf{h} + \mathbf{n} \tag{3}$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_M]^T$ denotes the received signal, $\mathbf{h} = [h_1, h_2, \ldots, h_N]^T$ is the sparse signal vector with $N > M$ and $\mathbf{n}$ is the noise.

Recovery is possible if the following two fundamental premises underlying CS are attended [10]:

- *Sparsity* - means that the signal could be characterized by few significant terms in some domain.
- *Incoherence* - states that distances between sparse signals are approximately conserved as distances between their respective measurements generated by the sampling process.

The largest correlation between any two elements of $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ is measured by the coherence between these matrices and defined by:

$$\mu(\boldsymbol{\Phi}, \boldsymbol{\Psi}) = \sqrt{N} \max_{1 \leq k,j \leq N} |< \varphi_k, \psi_j >| \tag{4}$$

If $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ contain correlated elements, the coherence is large. On the contrary, the coherence is small. Compressive sensing is mainly concerned with low coherence pairs. In [32], considering C as a constant, the authors showed that if (5) holds, then with overwhelming probability one sparse recovery algorithm will recover the signal.

$$M \geq C\mu^2(\boldsymbol{\Phi}, \boldsymbol{\Psi}) s \log N \tag{5}$$

Equation (5) shows that fewer measurements will be required to recover the signal if the coherence between $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ is small [2].

As illustrated in Fig.2, the last step (sparse recovery - CS Reconstruction) recovers the sparse signal from a small set of measurements $\mathbf{y}$ through a specific sparse recovery algorithm [31]. This step concerns the development of efficient sparse recovery algorithms. Some of them are addressed in Section IV.

Fig. 3 illustrates the relationship between the variables in a noiseless scenario. This work considers that the signal to be estimated is already in its sparse representation in a noisy scenario. The system is defined by:

$$\mathbf{y} = \mathbf{A}\mathbf{h} + \mathbf{n} \tag{6}$$

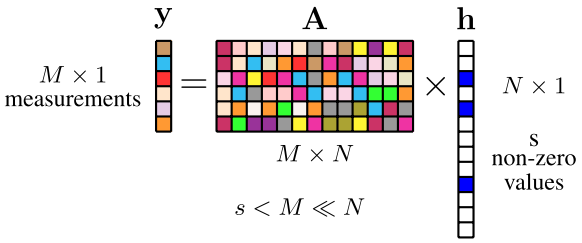

FIGURE 3. Representation of measurements used in compressive sensing.

One of the challenges associated with the sparse signal estimation is to identify the locations of the non-zero signal components. In other words, this is finding the subspace generated by no more than s columns of the matrix $\mathbf{A}$, related to

the received signal $\mathbf{y}$. After finding these positions, the non-zero coefficients can be calculated by applying the pseudoinversion process.

CS theory addresses two main problems:

- Design of the measurement matrix $\mathbf{A}$.
- Development of a sparse recovery algorithm for the efficient estimation of $\mathbf{h}$, given only $\mathbf{y}$ and $\mathbf{A}$.

In the first problem, the goal is to design a measurement matrix $\mathbf{A}$ which assures that the main information of any s-sparse or compressible signal is in this matrix [2]. The ideal goal is to design an appropriate measurement matrix with $M \approx s$.

The measurement matrix is very important in the process of recovering the sparse signal. According to [10], if the Restricted Isometry Property (RIP) defined in (7) is satisfied, using some recovery algorithm, it is possible to obtain an accurate estimation of the sparse signal $\mathbf{h}$, for example solving an l_p-norm problem [33]. $\delta_s \in (0, 1)$ is the RIC (Restricted Isometry Constant) value and corresponds to the smallest number that achieves (7).

$$(1 - \delta_s)\|\mathbf{h}\|_2^2 \leq \|\mathbf{A}\mathbf{h}\|_2^2 \leq (1 + \delta_s)\|\mathbf{h}\|_2^2 \tag{7}$$

Table 2 reproduces a comparison between deterministic sensing and random sensing for the measurement matrix $\mathbf{A}$ presented in [2]. The random matrices are one approach to obtain a measurement matrix $\mathbf{A}$ that obeys the RIP condition. Many works deal with random measurement matrices generated by identical and independent distributions (i.i.d.) such as Bernoulli, Gaussian, and random Fourier ensembles [34]–[37]. However, these matrices require significant space for storage and they have excessive complexity in reconstruction [2]. Furthermore, it is difficult to verify whether these matrices satisfy the RIP property with a small RIC value [2].

TABLE 2. Comparison between random and deterministic sensing [2].

Random Sensing	Deterministic Sensing
Outside the mainstream of signal processing: worst case signal processing	Aligned with the mainstream of signal processing: average case signal processing
Less efficient recovery time	More efficient recovery time
No explicit constructions	Explicit constructions
Larger storage	Efficient storage
Looser recovery bounds	Tighter recovery bounds

Therefore, deterministic matrices have been studied to be used as measurement matrices. In [38] and [39], the authors propose deterministic measurement matrices based on coherence and based on RIP, respectively. Moreover, deterministic measurement matrices are constructed via algebraic curves over finite fields in [40]. Furthermore, a survey on deterministic measurement matrices for CS can be found in [16].

Defined the appropriate measurement matrix $\mathbf{A}$, $\mathbf{h}$ can be estimated by the least squares (LS) solution of (6), i.e., solving the problem (8), where ϵ is a predefined error tolerance.

$$\min \|\hat{\mathbf{h}}\|_2 \quad \textit{subject to } \|\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}\|_2^2 < \epsilon \tag{8}$$

This system is "underdetermined" (the matrix $\mathbf{A}$ has more columns than rows). Let $\mathbf{A}^\dagger$ be the pseudo-inverse matrix of $\mathbf{A}$ and $\mathbf{A}\mathbf{A}^H$ has an inverse matrix, according to the LS algorithm, the unique solution $\hat{\mathbf{h}}$ of this optimization problem is given by (9) [41].

$$\hat{\mathbf{h}}_{LS} = \mathbf{A}^\dagger\mathbf{y} = \mathbf{A}^H(\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{y} \tag{9}$$

It is worth noting that the least squares minimization problem cannot return a sparse vector, so alternatives have been sought. By focusing on the sparsity constraint on the solution and solving the l_0 norm minimization described by (10), it is possible to obtain a sparse approximation $\hat{\mathbf{h}}$.

$$\min \|\hat{\mathbf{h}}\|_0 \quad \textit{subject to } \|\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}\|_2^2 < \epsilon \tag{10}$$

The Lemma 1.2 of [42] shows that if the matrix $\mathbf{A}$ obeys the RIP condition with constant $\delta_{2s} < 1$, (10) has an unique solution and $\mathbf{h}$ can be reconstructed exactly from $\mathbf{y}$ and $\mathbf{A}$.

Unfortunately, an exhaustive search over all $\binom{N}{s}$ possible sparse combinations is required in the l_0 minimization problem, which is computationally intractable for some practical applications. Thus, although this gives the desired solution, in practice it is not feasible to solve this equation. The excessive complexity of such a formulation can be avoided with the minimization of the l_1 problem (11), which can efficiently compute (10) under certain conditions, as demonstrated in [43].

$$\min \|\hat{\mathbf{h}}\|_1 \quad \textit{subject to } \|\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}\|_2^2 < \epsilon \tag{11}$$

One of the advantages of the l_1 norm minimization approach is that it can be solved efficiently by linear programming techniques [44]. Moreover, Donoho and Tanner [45] say that sparse signals can be recovered through l_1 minimization if $M \approx 2s\log(N)$.

III. APPLICATION OF COMPRESSIVE SENSING

This section overviews some application areas for the CS theory and its sparse recovery algorithms.

A. IMAGE AND VIDEO

1) COMPRESSIVE IMAGING

Natural images can be sparsely represented in wavelet domains, so the required number of measurements in compressive imaging can be reduced using CS [46], [47]. One example of application is the single-pixel camera that allows reconstructing an image in a sub-Nyquist image acquisition, that is, from fewer measurements than the number of reconstructed pixels [48].

2) MEDICAL IMAGING

CS can be very useful for medical imaging. For example, the magnetic resonance imaging (MRI) is a time-consuming and costly process. CS allows to decrease the number of samples, and then to reduce the time of acquisition [49]. Similarly, bio-signals such as ECG signals are sparse in either

wavelet or Fourier domain [50]. CS allows to take advantage of the sparsity and reduces the required number of collected measurements [49]–[52]. A hardware implementation on a system on chip (SoC) platform of a solution to tackle big data transmission and privacy issues is presented in [53].

3) VIDEO CODING

Due to the development and the increase of video surveillance, mobile video, and wireless camera sensor networks, wireless video broadcasting is becoming more popular and finding several real-time applications [54], [55]. In these cases, a single video stream is simultaneously transmitted to several receivers with different channel conditions [55]. In order to do this, many new video codecs have been proposed using compressive sensing [55]–[58].

4) COMPRESSIVE RADAR

Radar imaging systems aim to determine the direction, altitude, and speed of fixed and moving objects [15]. By solving an inverse problem using the compressive sensing theory, the received radar signal can be recovered from fewer measurements [15]. Therefore, the cost and the complexity of the hardware of the receiver are extremely reduced [15], [59]. Moreover, the CS has been a novel way to deal with the Inter-Burst Translational Motion Compensation (IBTMC) to achieve the exact recovery of Inverse Synthetic Aperture Radar (ISAR) images from limited measurements [60].

B. COMPRESSIVE TRANSMISSION DATA

1) WIRELESS SENSORS NETWORKS (WSNs)

Wireless sensor networks (WSNs) require high communication costs and energy consumption. Due to critically resource constraints as limited power supply, communication bandwidth, memory, and processing performance, CS can be used to reduce the number of bits to be transmitted or to represent the sensed data in WSNs [61]–[64].

2) INTERNET OF THINGS (IoT)

The use of internet of things (IoT) devices has increased and it is estimated that it will continue to do so in the following years. This includes home automation/control devices, security cameras, mobile phones, and sensing devices [65]. However, the IoT devices have computation, energy, and congestion constraints. Even if they need to transmit large amounts of data, they usually have limited power and low-computation capabilities. Moreover, given the large number of devices connected, they can suffer from congestion and packet drops [65]. Thus, special data transmission strategies have to be developed to enable low-power and low-cost signal processing operations, and energy-efficient communications [65]. Multimedia data usually possesses sparse structures. Therefore, the CS theory emerges as a good strategy to reduce the amount of data that the IoT devices need to transmit with a high fidelity recovery data [66].

3) ASTROPHYSICAL SIGNALS

Radio receivers located in outer space suffer from strong restrictions on storage capacity, energy consumption, and transmission rate. To overcome these challenges, sampling architectures using CS provide a data acquisition technique with fewer measurements. Thus, the amount of collected data to be downloaded to Earth and the energy consumption are reduced. The simple coding process with low computational cost provided by the CS promotes its use in real-time applications often found onboard spacecrafts. Moreover, the reconstruction of the signals will be done on Earth where there are much more computing and energy resources than onboard a satellite [67], [68].

4) MACHINE LEARNING

Machine learning algorithms perform pattern recognition (e.g., classification) on data that is too complex to model analytically to solve high-dimensional problems. However, the amount of information generated by acquisition devices is always huge and ever-growing. It can achieve gigabytes of data or more that exceeds the processing capacity of the most sophisticated machine learning algorithms [69]. To reduce the energy consumption of the applications, as in low-power wireless neural recording tasks, signals must be compressed before transmission to extend battery life. In these cases, the CS can be used and it was demonstrated its potential in neural recording applications [69]–[71].

C. COMMUNICATION SYSTEMS

1) COGNITIVE RADIOS (CRS)

Cognitive radios aim to provide a solution to the inefficient usage of the frequency spectrum. Spectrum sensing techniques suffer from computational complexity, hardware cost, and high processing time [31]. Since usually only some of the available channels are occupied by the users, the signal of interest is normally sparse in the frequency domain. Hence, the CS can be used to sense a wider spectrum with reduced sampling requirements, resulting in more power efficient systems [4], [18], [72], [73].

2) SPARSE CHANNEL ESTIMATION

Channels of several communication systems, e.g., underwater communication systems [74], [75], WideBand HF [76], [77], high-definition television (HDTV) [78], [79], Ultra-WideBand communications [6], [80], [81], and mmWave system [82], [83], can be considered or well modelled as sparse channels. That is, the impulse response of these channels are mainly characterized by few significant components widely separated in some domain. In these cases, better results can be achieved using the compressive sensing theory to estimate these sparse channels [84]–[86]. In [87], a low-complexity CS hardware implementation for channel estimation in the integrated services digital broadcasting-terrestrial (ISDB-T) system is proposed in FPGA.

3) ANALOG TO INFORMATION CONVERSION (AIC)

The analog to digital conversion (ADC) is based on the Nyquist sampling theorem in order to have a perfectly reconstruction of the information. That is, the signal is uniformly sampled at a rate at least twice its bandwidth. In several applications, the information of the signal is much smaller than its bandwidth. In these cases, this represents a waste of hardware and software resources to sample the whole signal. To deal with this, an analog to information conversion (AIC) can use the CS theory to acquire a large bandwidth with relaxed sampling rate requirements, enabling faster, less expensive, and more energy-efficient solutions [88]–[92]. Examples of AIC are: random demodulator [91], [93], [94], modulated wideband converter [95] and non-uniform sampling [90], [96]–[98]. All these architectures have advantages and limitations. While the random demodulator AIC employs finite temporal sampling functions with infinite spectral support, the modulated wideband converter AIC has finite spectral sampling functions with infinite temporal support. Moreover, the modulated wideband converter AIC requires a large number of branches, so synchronization among the branches is also needed, thus consuming more area and power. On the other hand, the non-uniform sampling AIC is sensitive to timing jitter, i.e., a sampling time with a small error can lead to a big error in the sample value for input signals that change rapidly.

D. DETECTION AND RECOGNITION SYSTEMS

1) SPEECH RECOGNITION

Dictionary of example speech tokens can be used to sparsely represent speech signals [99]. Moreover, the speech signal can have sparse representation for a suitable selection of sparse basis functions, but for the noise, it will be difficult to derive a sparse representation. So, it is possible to exploit this characteristic and through the CS theory achieve a better speech recognition performance [20], [99], [100].

2) SEISMOLOGY

The compressive sensing theory has an important use in data acquisition, that is, situations when it is intricate to obtain a lot of samples, for example in the case of seismic data [101]. The layers of the Earth can be estimated by measuring the reflections of a signal from different layers of the Earth. However, this requires a large data collection that is a time-consuming and expensive process. To deal with this, several works have proposed the CS for different seismic applications [101]–[103].

3) DIRECTION-OF-ARRIVAL (DOA)

Direction-of-Arrival (DOA) estimation is the process of determining which direction a signal impinging on an array has arrived from [104] and [105]. Because there are only a few non-zeros in the spatial spectrum of array signals, which represent their corresponding spatial locations, this sparsity can be applied to the DOA estimation [106]. Hence, the compressive sensing theory can be applied to the problem of DOA estimation by splitting the angular region into N potential DOAs, where only $s \ll N$ of the DOAs have an impinging signal (alternatively $N - s$ of the angular directions have a zero-valued impinging signal present) [107], [108]. These DOAs are then estimated by finding the minimum number of DOAs with a non-zero valued impinging signal that still gives an acceptable estimate of the array output [23], [104].

IV. SPARSE RECOVERY ALGORITHMS

Several sparse recovery algorithms have been proposed in the last years. They have to recover a sparse signal from an undersampled set of measurements. They are usually classified into three main categories: convex relaxations, non-convex optimization techniques, and greedy algorithms [109]. Fig. 4 shows the algorithms that will be addressed in more details in this work. For the following algorithms, the system model is defined by (6) and the notation is presented in Table 1.

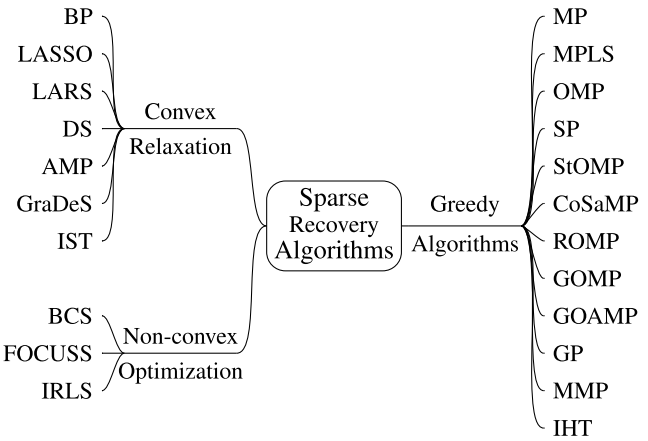

FIGURE 4. Classification of sparse recovery algorithms.

Section IV-A presents some algorithms from the first category. These algorithms result in convex optimization problems whose efficient solutions exist relying on advanced techniques, such as projected gradient methods, interior-point methods, or iterative thresholding [84].

On the other hand, non-convex optimization approaches can recovery the signal by taking into account a previous knowledge of its distribution (see Section IV-B) [31]. Thanks to a posterior probability density function, these solutions offer complete statistics of the estimate. Nonetheless, they can be unsuitable for high-dimensional problems due to their intensive computational requirements [110].

The third category is composed of the greedy algorithms. They recover the signal in an iterative way, making a local optimal selection at each iteration hoping to find the global optimum solution at the end of the algorithm (see Section IV-C).

A. CONVEX RELAXATIONS

1) BASIS PURSUIT (BP)

Basis Pursuit (BP) is a signal processing technique that decomposes the signal into an superposition of basic elements. This decomposition is optimal in the sens that it leads to the smallest l_1 norm of coefficients among all such decompositions [111]. The BP algorithm seeks to determine a signal's representation that solves the problem:

$$\min \|\mathbf{h}\|_1 \quad subject\ to\ \mathbf{y} = \mathbf{A}\mathbf{h} \tag{12}$$

BP is a principle of global optimization without any specified algorithm. One of a possible algorithm to be used is the BP-simplex [111] that is inspired by the simplex method of linear programming [112]. For the BP-simplex, first, an initial basis $\mathbf{A}(\Lambda)$ is found by selecting M linearly independent columns of $\mathbf{A}$. Then, at each step, the swap which best improves the objective functions is chosen to update the current basis, that is, one term in the basis is swapped for one term that is not in the basis [111].

Huggins and Zucker [113] propose an algorithm for BP called Greedy Basis Pursuit (GBP). Unlike standard linear programming methods for BP, the GBP algorithm proceeds more like the MP algorithm, that is, it builds up the representation by iteratively selecting columns based on computational geometry [113]. Moreover, the GBP allows discarding columns that have already been selected [113].

2) BP DE-NOISING (BPDN) / LEAST ABSOLUTE SHRINKAGE AND SELECTION OPERATOR (LASSO)

The Basis Pursuit Denoising (BPDN) [111] / Least Absolute Shrinkage and Selection Operator (LASSO) [114] algorithm considers the presence of the noise $\mathbf{n}$:

$$\min \|\mathbf{h}\|_1 \quad subject\ to\ \mathbf{y} = \mathbf{A}\mathbf{h} + \mathbf{n} \tag{13}$$

and aims to solve the optimization problem defined by:

$$\min(\frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{h}\|_2^2 + \lambda_p \|\mathbf{h}\|_1) \tag{14}$$

where $\lambda_p > 0$ is a scalar parameter [111], [114]. Its value greatly influences on the performance of the LASSO algorithm and therefore should be chosen carefully. In [111], the authors suggest:

$$\lambda_p = \sigma\sqrt{2\log(p)} \tag{15}$$

where $\sigma > 0$ is the noise level and p is the cardinality of the dictionary [111].

Comparing with the LS cost function, it is possible to see that (14) basically includes a l_1 norm penalty term. Hence, under certain conditions, the solution would achieve the minimal LS error [115]. Since $\|\mathbf{h}\|_1$ is not differentiable for any zero position of $\mathbf{h}$, it is not possible to obtain an analytical solution for the global minimum of (14).

There are several iterative techniques to find the minimum of (14) [111], [114]. One of these is called "Shooting" [116] and starts by the solution:

$$\hat{\mathbf{h}} = (\mathbf{A}^H\mathbf{A} + \mathbf{I})^{-1}\mathbf{A}^H\mathbf{y} \tag{16}$$

where $\mathbf{I}$ is the identity matrix. Let a_j be the j^{th} column of the matrix A and B_j be defined by (18), each j^{th} element of $\hat{\mathbf{h}}$ is updated by:

$$\hat{h}_j = \begin{cases} \dfrac{\lambda - B_j}{a_j^T a_j}, & if\ B_j > \lambda \\ \dfrac{-\lambda - B_j}{a_j^T a_j}, & if\ B_j < -\lambda \\ 0, & if\ |B_j| \le \lambda \end{cases} \tag{17}$$

$$B_j = -a_j^T y + \sum_{l \neq j} a_j^T a_l \hat{h}_l \tag{18}$$

The original Shooting method is applied to real variables. For complex variables, an adaptation is necessary. In [117], two schemes are presented to adapt the LASSO algorithm to estimate a complex signal $\mathbf{h}$:

- r-LASSO: Let $imag(.)$ and $real(.)$ be the imaginary and real parts of a complex vector, respectively, it is defined by [117]:

$$\mathbf{y}^R = \begin{pmatrix} real(\mathbf{y}) \\ imag(\mathbf{y}) \end{pmatrix}, \quad \mathbf{h}^R = \begin{pmatrix} real(\mathbf{h}) \\ imag(\mathbf{h}) \end{pmatrix},$$
$$\mathbf{A}^R = \begin{pmatrix} real(\mathbf{A}) & -imag(\mathbf{A}) \\ imag(\mathbf{A}) & real(\mathbf{A}) \end{pmatrix} \tag{19}$$

  These definitions are used in the Shooting method in (16) and each j^{th} element of $\hat{\mathbf{h}}$ is calculated by [117]:

$$\hat{h}_j = \hat{h}_j^R + \sqrt{-1}\hat{h}_{j+N}^R \tag{20}$$

- c-LASSO: The complex l_1-norm can be solved by some methods [117], [118]. It is defined by:

$$\|\mathbf{h}\|_1 = \sum_i |h_i| = \sum_i \sqrt{real(h_i)^2 + imag(h_i)^2} \tag{21}$$

In many applications, the imaginary and real components tend to be either zero or non-zero simultaneously [117]. However, the r-LASSO does not take into account the information about any potential grouping of the real and imaginary parts [117]. On the other hand, the c-LASSO considers this extra information [117]. A comparison between r-LASSO and c-LASSO performed in [117] concludes that the c-LASSO outperforms the r-LASSO since it exploits the connection between the imaginary and the real parts.

3) LEAST ANGLE REGRESSION (LARS)

The Least Angle Regression (LARS) algorithm begins with $\hat{\mathbf{h}} = \mathbf{0}$, the residual vector $\mathbf{b}_0 = \mathbf{y}$, and the active set $\Lambda = \emptyset$. This algorithm selects a new column from the matrix $\mathbf{A}$ at each iteration i and adds its index to the set Λ_i [119]. The column $\mathbf{a}_{j_1}$ that has a smaller angle with $\mathbf{b}_0$ is selected at the first iteration. Then, the coefficient $\hat{h}_1(j_1)$ associated with the selected column $\mathbf{a}_{j_1}$ is increased [119]. Next, the smallest possible steps in the direction of the column $\mathbf{a}_{j_1}$ is taken until another column $\mathbf{a}_{j_2}$ has as much absolute correlation value with the current residual as the column $\mathbf{a}_{j_1}$. The algorithm continues in a direction equiangular between

the two active columns ($\mathbf{a}_{j_1}$,$\mathbf{a}_{j_2}$) until a third column $\mathbf{a}_{j_3}$ earns its way into the most correlated set [119]. The algorithm stops when no remaining column has correlation with the current residual [119].

Fig. 5 illustrates the begin of the LARS algorithm considering a two-dimensional system. As said before, LARS starts with $\hat{\mathbf{h}}_0 = \mathbf{0}$ and the residual vector $\mathbf{b}_0 = \mathbf{y}$. Let $\theta_t(i)$ be the angle between the column $\mathbf{a}_{j_i}$ and the current residual vector $\mathbf{b}_i = \mathbf{y} - \mathbf{A}\hat{\mathbf{h}}_i$ at iteration i, the column $\mathbf{a}_{j_1}$ is selected due to its absolute correlation with the initial residual vector compared to $\mathbf{a}_{j_2}$ ($\theta_1(1) < \theta_1(2)$) [120]. Next, the algorithm continues in the direction of $\mathbf{a}_{j_1}$ by adding the step size γ_1. γ_1 is chosen in a way to guarantee that the columns $\mathbf{a}_{j_1}$ and $\mathbf{a}_{j_2}$ have the same absolute correlation with the current residual vector at the next iteration ($\theta_2(1) = \theta_2(2)$). The solution coefficient is $\hat{h}_1(j_1) = \gamma_1$ [120]. The column $\mathbf{a}_{j_2}$ is added to the set Λ at the second iteration, and the LARS continues in a equiangular direction with $\mathbf{a}_{j_1}$ and $\mathbf{a}_{j_2}$. Then, the step size γ_2 that leads to the vector $\mathbf{y}$ is added [120]. Finally, the solution coefficients are equal to: $\hat{h}_2(j_1) = \gamma_1 + \gamma_2\, d_2(j_1)$ and $\hat{h}_2(j_2) = \gamma_2\, d_2(j_2)$, where $\mathbf{d}_2$ is the updated direction at the second iteration that is equiangular with the active columns ($\mathbf{a}_{j_1}$, $\mathbf{a}_{j_2}$). The estimated vector $\hat{\mathbf{h}}$ is updated by multiplying the step size γ with the updated direction $\mathbf{d}$ [120]. The algorithm continues until the residual be zero.

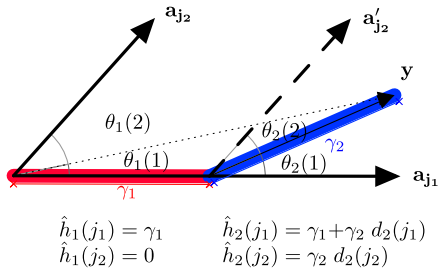

FIGURE 5. LARS approximates the vector y by using $\mathbf{a}_{j_1}$ and $\mathbf{a}_{j_2}$ [120].

A modified LARS called "homotopy algorithm" was proposed by Donoho and Tsaig to find a sparse solution of an underdetermined linear system [44].

These steps can summarize the LARS algorithm [120]:

- Step 1: Initialize the residual vector $\mathbf{b}_0 = \mathbf{y}$, the active set $\Lambda = \emptyset$, $\hat{\mathbf{h}}_0 = \mathbf{0}$ and the iteration counter $i = 1$.
- Step 2: Calculate the correlation vector: $\mathbf{c}_i = \mathbf{A}^T\mathbf{b}_{i-1}$.
- Step 3: Find the maximum absolute value in the correlation vector: $\lambda_i = \|\mathbf{c_i}\|_\infty$.
- Step 4: Stop the algorithm if $\lambda \approx 0$. If not, go to Step 5.
- Step 5: Find the active set: $\Lambda = \{j : |c_i(j)| = \lambda_i\}$.
- Step 6: Solve the following least square problem to find active entries of the updated direction: $\mathbf{A}^T(\Lambda)\mathbf{A}(\Lambda)\mathbf{d}_i(\Lambda) = sign(\mathbf{c}_i(\Lambda))$.
- Step 7: Set the inactive entries of the updated direction to zero: $\mathbf{d}_i(\Lambda^C) = \mathbf{0}$.
- Step 8: Calculate the step size γ_i by:

$$\gamma_i = \min_{j\in\Lambda^c}\left\{\frac{\lambda_i - c_i(j)}{1 - \mathbf{a}_j^T\mathbf{A}(\Lambda)\mathbf{d}_i(\Lambda)}, \frac{\lambda_i + c_i(j)}{1 + \mathbf{a}_j^T\mathbf{A}(\Lambda)\mathbf{d}_i(\Lambda)}\right\}$$

- Step 9: Calculate $\hat{\mathbf{h}}_i = \hat{\mathbf{h}}_{i-1} + \gamma_i\mathbf{d}_i$.
- Step 10: Update $\mathbf{b}_i = \mathbf{y} - \mathbf{A}\hat{\mathbf{h}}_i$.
- Step 11: Stop the algorithm if $\|\mathbf{b}_i\|_2 < \epsilon$. Otherwise, set $i = i + 1$ and return to Step 2.

4) THE DANTZIG SELECTOR (DS)

The Dantzig Selector (DS) is a solution to l_1 minimization problem [121]:

$$\min \|\hat{\mathbf{h}}\|_1 \quad subjet\ to\ \|\mathbf{A}^T\mathbf{b}\|_\infty \le \sqrt{1+\delta_1}\lambda_N\sigma \tag{22}$$

where $\mathbf{b} = \mathbf{y} - \mathbf{A}\hat{\mathbf{h}}$ is the residual vector, σ is the standard deviation of the Additive White Gaussian Noise in (6), $\lambda_N > 0$ and all the columns of $\mathbf{A}$ have norm less than $\sqrt{1+\delta_1}$. $\|\mathbf{A}^T\mathbf{b}\|_\infty$ is defined by:

$$\|\mathbf{A}^T\mathbf{b}\|_\infty = \sup_{1\le i\le N} |(\mathbf{A}^T\mathbf{b})_i| \tag{23}$$

For an orthogonal matrix $\mathbf{A}$, the Dantzig Selector is the l_1-minimizer subject to the constraint $\|\mathbf{A}^T\mathbf{y} - \hat{\mathbf{h}}\|_\infty \le \lambda_N\sigma$, and the i^{th} element of $\hat{\mathbf{h}}$ is calculated by:

$$\hat{h}(i) = \max(|(\mathbf{A}^T\mathbf{y})_i| - \lambda_N\sigma, 0)sgn((\mathbf{A}^T\mathbf{y})_i) \tag{24}$$

5) APPROXIMATE MESSAGE PASSING (AMP)

The Approximate Message Passing (AMP) algorithm is described in [122] and [123]. This algorithm starts by $\hat{\mathbf{h}}_0 = \mathbf{0}$ and $\mathbf{b}_0 = \mathbf{y}$. Then, in each iteration i, it updates these vectors by:

$$\hat{\mathbf{h}}_i = \eta_{i-1}(\hat{\mathbf{h}}_{i-1} + \mathbf{A}^T\mathbf{b}_{i-1}) \tag{25}$$

$$\mathbf{b}_i = \mathbf{y} - \mathbf{A}\hat{\mathbf{h}}_i + \frac{1}{\delta}\mathbf{b}_{i-1}\left\langle\eta'_{i-1}(\mathbf{A}^T\mathbf{b}_{i-1} + \hat{\mathbf{h}}_{i-1})\right\rangle \tag{26}$$

where $\delta = M/N$, $\eta_i(.)$ is the soft thresholding function, $\langle\mathbf{u}\rangle = \sum_{i=1}^N u_i/N$ for $\mathbf{u} = (u_1, \ldots, u_N)$ and $\eta'_i(s) = \frac{\partial}{\partial s}\eta_i(s)$. The term $\frac{1}{\delta}\mathbf{b}_{i-1}\left\langle\eta'_i(\mathbf{A}^T\mathbf{b}_{i-1} + \hat{\mathbf{h}}_{i-1})\right\rangle$ is from theory of belief propagation in graphical model [122].

The thresholding function $\eta_i(.)$ depends on iteration and problem setting. Donoho *et al.* [123] consider the threshold control parameter λ and $\eta_i(.) = \eta(.; \lambda\sigma_i)$ defined by:

$$\eta(u;\lambda\sigma_i) = \begin{cases}(u - \lambda\sigma_i), & if\ u \ge \lambda\sigma_i\\ (u + \lambda\sigma_i), & if\ u \le -\lambda\sigma_i\\ 0, & otherwise\end{cases} \tag{27}$$

where σ_i is the mean square error of the current estimate solution $\hat{\mathbf{h}}_i$ at iteration i. The optimal value of λ is [122]:

$$\lambda(\delta) = \frac{1}{\sqrt{\delta}}\arg\max_{x\le 0}\left\{\frac{1 - (2/\delta)[(1+x^2)\Phi(-x) - x\phi(x)]}{1 + x^2 - 2[(1+x^2)\Phi(-x) - x\phi(x)]}\right\} \tag{28}$$

where $\Phi(x) = \int_{-\infty}^{x}\frac{e^{-t^2/2}}{\sqrt{2\pi}}dt$ and $\phi(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}$.

A high-speed FPGA implementation of AMP is presented in [124]. Moreover, Gallo *et al.* [125] present an implementation of AMP based on memristive crossbar arrays. Furthemore, an adaptive complex approximate message passing (CAMP) algorithm and its hardware implementation in FPGA are proposed in [126].

6) GRADIENT DESCENT WITH SPARSIFICATION (GRADES)

This algorithm was proposed in [127]. It considers a measurement matrix $\mathbf{A}$ which satisfies the RIP with an isometric constant $\delta_{2s} < 1/3$. This algorithm finds a sparse solution for the l_1 minimization problem in an iterative way.

First, the algorithm initializes the signal estimation $\hat{\mathbf{h}}_0 = \mathbf{0}$. Then, in each iteration i, it estimates the signal by:

$$\hat{\mathbf{h}}_i = H_s\left(\hat{\mathbf{h}}_{i-1} + \frac{1}{\gamma}\mathbf{A}^H(\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}_{i-1})\right) \tag{29}$$

where $\gamma > 1$ and the operator $H_s()$ sets all components to zero except the s largest magnitude components.

7) ITERATIVE SOFT THRESHOLDING (IST)

Daubechies *et al.* [128] demonstrate that soft thresholding can be used to minimize equations of the form:

$$\frac{1}{2}\|\mathbf{Ah} - \mathbf{y}\|_2^2 + \tau\|\mathbf{h}\|_1 \tag{30}$$

The solution is given by the limit of the sequence, where each iteration called *Landweber iteration* is defined by [128]:

$$\hat{\mathbf{h}}_i = S_\tau\left(\hat{\mathbf{h}}_{i-1} + \beta\mathbf{A}^H\left(\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}_{i-1}\right)\right) \tag{31}$$

where $\hat{\mathbf{h}}_0 = \mathbf{0}$, β is a stepsize, and $S_\tau(.)$ is the soft thresholding function defined by (32) applied to each element of the vector.

$$S_\tau(x) = \begin{cases} x - \tau, & if\ x > \tau \\ 0, & if\ |x| \leqslant \tau \\ x + \tau, & if\ x < -\tau \end{cases} \tag{32}$$

Let $B_R \subset \mathbb{R}^n$ be the l_1 ball of radius R defined by:

$$B_R = \{\mathbf{h} \in \mathbb{R}^n : \|\mathbf{h}\|_1 \leqslant R\} \tag{33}$$

Daubechies *et al.* [129] suggest a different form of (31) which calls the *projected Landweber iteration* given by:

$$\hat{\mathbf{h}}_i = \mathbb{P}_R\left(\hat{\mathbf{h}}_{i-1} + \mathbf{A}^H\left(\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}_{i-1}\right)\right) \tag{34}$$

where $\mathbb{P}_R(\mathbf{x})$ is the projection of a point $\mathbf{x}$ to the closest point (under the l_2 norm) onto the convex set B_R.

The next steps calculate $\mathbb{P}_R(\mathbf{x})$, that is, calculate the vector $\mathbf{t}$ that is the closest point (under l_2 distance) in the l_1 ball of radius R to $\mathbf{x}$ [11]:

- Step 1: If $\|\mathbf{x}\|_1 \leqslant R$, then $\mathbf{t} = \mathbf{x}$.
- Step 2: Sort the components of $\mathbf{x}$ by magnitude to get the vector $\hat{\mathbf{x}}$ where $|\hat{\mathbf{x}}_1| \geqslant |\hat{\mathbf{x}}_2| \geqslant \ldots \geqslant |\hat{\mathbf{x}}_n|$.
- Step 3: Let $\left\|S_{\hat{\mathbf{x}}_k}(\mathbf{x})\right\|_1 = \sum_{i=1}^{k-1}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_k)$, find k such that:

$$\begin{aligned} \left\|S_{\hat{\mathbf{x}}_k}(\mathbf{x})\right\|_1 \leqslant R \leqslant \left\|S_{\hat{\mathbf{x}}_{k+1}}(\mathbf{x})\right\|_1, \\ \sum_{i=1}^{k-1}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_k) \leqslant R \leqslant \sum_{i=1}^{k}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{k+1}) \end{aligned} \tag{35}$$

- Step 4: Calculate $\mu = \hat{\mathbf{x}}_k + \frac{1}{k}(R - \left\|S_{\hat{\mathbf{x}}_k}(\mathbf{x})\right\|_1)$. Then $\mathbf{t} = S_\mu(\mathbf{x})$.

According to [129], the projected Landweber iterative step with an adaptive descent parameter $\beta_i > 0$ as in (36) will converge to solve $argmin_{\hat{\mathbf{h}}}\|\mathbf{A}\hat{\mathbf{h}} - \mathbf{y}\|_2$, that is, minimize $\hat{\mathbf{h}}$ in the l_1 ball B_R.

$$\hat{\mathbf{h}}_i = \mathbb{P}_R\left(\hat{\mathbf{h}}_{i-1} + \beta_{i-1}\mathbf{A}^H\left(\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}_{i-1}\right)\right) \tag{36}$$

β_i can be chosen by [118]:

- $\beta_i = 1, \quad \forall i.$
- $\beta_i = \dfrac{\left\|\mathbf{A}^H(\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}_i)\right\|_2^2}{\left\|\mathbf{A}\mathbf{A}^H(\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}_i)\right\|_2^2}$

Although IST is guaranteed to converge [128], it converges slowly. Therefore, several modifications have been proposed to speed it up such as the "fast ISTA" (FISTA) [130].

B. NON-CONVEX OPTIMIZATION TECHNIQUES

1) BAYESIAN COMPRESSIVE SENSING (BCS)

Let σ^2 be the noise variance, the sparse Bayesian learning (SBL) assumes the Gaussian likelihood model [131]:

$$p(\mathbf{y}|\mathbf{h};\sigma^2) = (2\pi\sigma^2)^{-M/2}\exp\left(\frac{-1}{2\sigma^2}\|\mathbf{y} - \mathbf{Ah}\|^2\right) \tag{37}$$

In a Bayesian formulation, the formalization that $\mathbf{h}$ is sparse is made by placing a sparseness-promoting prior on $\mathbf{h}$ [132]. The Laplace density function is a widely used sparseness prior [133], [134]:

$$p(\mathbf{h}|\lambda) = \left(\frac{\lambda}{2}\right)^N exp\left(-\lambda\sum_{i=1}^{N}|h_i|\right) \tag{38}$$

and henceforth the subscript s on $\mathbf{h}$ is dropped, recognizing that the interest is in a sparse solution for the weights [132]. Thus, the solution of (6) corresponds to a maximum a posteriori (MAP) estimate for using the prior in (38) [114], [133].

According to the Bayesian probability theory, we consider that a class of prior probability distributions $p(\theta)$ is conjugate to a class of likelihood functions $p(x|\theta)$ if the resulting posterior distributions $p(\theta|x)$ are in the same family as $p(\theta)$ [132]. Since the Laplace prior is not conjugate to the Gaussian likelihood, the relevance vector machine (RVM) is used.

Assuming the hyperparameters $\boldsymbol{\alpha}$ and α_0 are known, a multivariate Gaussian distribution with mean and covariance given by (39) and (40) can express the posterior for $\mathbf{h}$ [132].

$$\boldsymbol{\mu} = \alpha_0\boldsymbol{\Sigma}\mathbf{A}^T\mathbf{y} \tag{39}$$

$$\boldsymbol{\Sigma} = (\alpha_0\mathbf{A}^T\mathbf{A} + \mathbf{D})^{-1} \tag{40}$$

where $\mathbf{D} = diag(\alpha_1, \alpha_2, \ldots \alpha_N)$. Therefore, the search for the hyperparameters $\boldsymbol{\alpha}$ and α_0 can be seen as a learning problem in the context of the RVM. A type-II maximum likelihood (ML) procedure can be used to estimate these hyperparameters from the data [135].

The logarithm of the marginal likelihood for $\boldsymbol{\alpha}$ and α_0, noted $L(\boldsymbol{\alpha}, \alpha_0)$, is given by [132]:

$$\begin{aligned} \log\ p(\boldsymbol{y}|\boldsymbol{\alpha}, \alpha_0) &= \log\int p(\boldsymbol{y}|\boldsymbol{h}, \alpha_0)p(\boldsymbol{h}|\boldsymbol{\alpha})d\mathbf{h} \\ &= -\frac{1}{2}\left[M\log 2\pi + \log|\mathbf{C}| + \mathbf{y}^T\mathbf{C}^{-1}\mathbf{y}\right] \end{aligned} \tag{41}$$

with $\mathbf{C} = \sigma^2\mathbf{I} + \mathbf{A}\mathbf{D}^{-1}\mathbf{A}^T$. The maximization of (41) can be obtained with a type-II ML approximation that uses the point estimates for $\boldsymbol{\alpha}$ and α_0. This can be achieved through the Expectation-Maximization (EM) algorithm [132], [135], to yield:

$$\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2} \tag{42}$$

where μ_i is the i^{th} posterior mean weight from (39) and $\gamma_i = 1 - \alpha_i \boldsymbol{\Sigma}_{ii}$ with $\boldsymbol{\Sigma}_{ii}$ the i^{th} diagonal element of (40).

2) FOCAL UNDERDETERMINED SYSTEM SOLUTION (FOCUSS)

The Focal Underdetermined System Solution (FOCUSS) was proposed in [136] to solve (6). First, a low-resolution initial estimate of the real signal is made. Then, the iteration process refines the initial estimate to the final localized energy solution [136]. The FOCUSS iterations are based on a weighted minimum norm solution defined as the solution minimizing a weighted norm $\|\mathbf{W}^{-1}\mathbf{h}\|_2$. It is given by [136]:

$$\hat{\mathbf{h}} = \mathbf{W}(\mathbf{A}\mathbf{W})^{\dagger}\mathbf{y} \tag{43}$$

where the definition of a weighted minimum norm solution is to find $\mathbf{h} = \mathbf{W}\mathbf{q}$ where $\mathbf{q} : min\|\mathbf{q}\|_2$, subject to $\mathbf{A}\mathbf{W}\mathbf{q} = \mathbf{y}$. When $\mathbf{W}$ is diagonal, the cost objective simply becomes $\left\|\mathbf{W}^{\dagger}\mathbf{h}\right\| = \sum_{i=1,w_i\neq 0}^{N}\left(\frac{h_i}{w_i}\right)^2$, where w_i are the diagonal entries of $\mathbf{W}$ [136].

The basis of the basic FOCUSS algorithm lies the Affine Scaling Transformation (AST):

$$\mathbf{q} = \hat{\mathbf{H}}_{k-1}^{\dagger}\hat{\mathbf{h}} \tag{44}$$

where $\hat{\mathbf{H}}_{k-1}^{\dagger} = diag(\hat{\mathbf{h}}_{k-1})$ [136]. Let $\mathbf{W}_{p_k}$ be the *a posteriori* weight in each iteration, the AST is used in the basic FOCUSS algorithm to construct the weighted minimum norm constraint (45) by setting $\mathbf{W}_{p_k} = \hat{\mathbf{H}}_{k-1}$ [136].

$$\|\mathbf{W}^T\mathbf{h}\|_2^2 = \|\mathbf{q}\|_2^2 = \sum_{i=1,w_i\neq 0}^{n}\left(\frac{h(i)}{w(i)}\right)^2 \tag{45}$$

Let $\hat{\mathbf{h}}_0 = \mathbf{0}$, the steps of the algorithm are:

$$\text{Step 1: } \mathbf{W}_{p_k} = (diag(\hat{\mathbf{h}}_{k-1})) \tag{46}$$

$$\text{Step 2: } \mathbf{q}_k = (\mathbf{A}\mathbf{W}_{p_k})^{\dagger}\mathbf{y} \tag{47}$$

$$\text{Step 3: } \hat{\mathbf{h}}_k = \mathbf{W}_{p_k}\mathbf{q}_k \tag{48}$$

The algorithm continues until a minimal set of the columns of $\mathbf{A}$ that describe $\mathbf{y}$ is obtained [136].

By introducing two parameters, the authors extend the basic FOCUSS into a class of recursively constrained optimization algorithms in [136]. In the first extension, $\hat{\mathbf{h}}_{k-1}$ is raised to some power l [136]. While in the second extension an additional weight matrix $\mathbf{W}_{a_k}$ which is independent of the *a posteriori* constraints is used [136]. The follow steps describe the algorithm:

$$\text{Step 1: } \mathbf{W}_{p_k} = (diag(\hat{\mathbf{h}}_{k-1}^{l})), \quad l \in \mathbb{N}^+ \tag{49}$$

$$\text{Step 2: } \mathbf{q}_k = (\mathbf{A}\mathbf{W}_{a_k}\mathbf{W}_{p_k})^{\dagger}\mathbf{y} \tag{50}$$

$$\text{Step 3: } \hat{\mathbf{h}}_k = \mathbf{W}_{a_k}\mathbf{W}_{p_k}\mathbf{q}_k \tag{51}$$

It can be assumed that $\mathbf{W}_{a_k}$ is constant for all iterations. According to [136], $l > 0.5$ when $h(i) > 0$ is imposed.

3) ITERATIVE REWEIGHTED LEAST SQUARES (IRLS)

The Iterative Reweighted Least Squares (IRLS) algorithm is used for solving (52) through a weigthed l_2 norm given by (53), where the weights are computated from the previous iterate $\mathbf{h}_{n-1}$, so $w_i = |h_{n-1}(i)|^{p-2}$ [137].

$$\min_{\mathbf{h}} \|\boldsymbol{h}\|_p^p \quad subject\ to\ \mathbf{A}\mathbf{h} = \mathbf{y} \tag{52}$$

$$\min_{\mathbf{h}} \sum_{i=1}^{N} w_i h^2(i) \quad subject\ to\ \mathbf{A}\mathbf{h} = \mathbf{y} \tag{53}$$

Let $\mathbf{Q}_n$ be the diagonal matrix with entries $1/w_i = |h_{n-1}(i)|^{2-p}$, the solution of (53) can be given by:

$$\mathbf{h}_n = \mathbf{Q}_n\mathbf{A}^T(\mathbf{A}\mathbf{Q}_n\mathbf{A}^T)^{-1}\mathbf{y} \tag{54}$$

To deal with the case $0 \leq p \leq 1$, where w_i will be undefined for $h_{n-1}(i) = 0$, the authors in [137] regularize the optimization problem by incorporating a small $\epsilon > 0$:

$$w_i = ((h_{n-1}(i))^2 + \epsilon)^{p/2-1} \tag{55}$$

C. GREEDY ALGORITHMS

Several greedy algorithms follow the steps showed in Fig. 6. There are some differences in the choice of the quantity of the column in each iteration, that is, the way to choose the indices j to compose the set J_i. For the MP, the OMP, and the MPLS algorithms only one column is chosen in each iteration. In contrast, the StOMP algorithm chooses all columns whose the projection value is bigger than the threshold value t_S. The calculation of the residual vector $\mathbf{b}_i$ and the estimation of the non-zero values of $\hat{\mathbf{h}}$ in each iteration are other differences between the algorithms. For example, the MPLS and the SP algorithms estimate $\hat{\mathbf{h}}$ only at the end of the algorithms as is explained in the subsections below.

Table 3 summarizes the inputs, the calculation of the residual vector $\mathbf{b}_i$ and the signal estimate components $\hat{\mathbf{h}}_i$ in each iteration. In the next subsections, the algorithms presented in Fig. 4 are explained.

1) MATCHING PURSUIT (MP)

The Matching Pursuit (MP) algorithm is proposed in [138]. Let $\hat{\mathbf{h}}_0 = \mathbf{0}$, each iteration i of the MP algorithm consists in finding the column $\mathbf{a}_{k_i} \in \mathbf{A}$ which is best aligned with the residual vector $\mathbf{b}_{i-1}$ ($\mathbf{b}_0 = \mathbf{y}$) according to (56) [138].

$$k_i = \arg\max_{l} |\mathbf{a}_l{}^H\mathbf{b}_{\mathbf{i-1}}|, \quad l = 1, 2, \ldots, N \tag{56}$$

TABLE 3. Main parameters and calculations of Greedy Algorithms.

Algorithm	Inputs	j	$\mathbf{b}_i$	$\hat{\mathbf{h}}_i$
MP	$\mathbf{A}, \mathbf{y}$	$\max_j \|\|c_j\|\|$	$\mathbf{b}_{i-1} - \frac{(\mathbf{a}_{l_i}{}^H \mathbf{b}_{i-1})\mathbf{a}_{l_i}}{\|\|\mathbf{a}_{l_i}\|\|_2^2}$	$\frac{(\mathbf{a}_{l_i}{}^H \mathbf{b}_{i-1})}{\|\|\mathbf{a}_{l_i}\|\|_2^2}$
MPLS	$\mathbf{A}, \mathbf{y}$	$\max_j \|\|c_j\|\|$	$\mathbf{b}_{i-1} - \frac{(\mathbf{a}_{l_i}{}^H \mathbf{b}_{i-1})\mathbf{a}_{l_i}}{\|\|\mathbf{a}_{l_i}\|\|_2^2}$	–
OMP	$\mathbf{A}, \mathbf{y}$	$\max_j \|\|c_j\|\|$	$\mathbf{y} - \mathbf{A}(\Lambda_i)\hat{\mathbf{h}}_i$	$\mathbf{A}(\Lambda_i)^\dagger \mathbf{y}$
SP	$\mathbf{A}, \mathbf{y}, s$	s biggest $\|\|c_j\|\|$	$\mathbf{y} - \mathbf{A}(\Lambda_i)\mathbf{A}^\dagger(\Lambda_i)\mathbf{y}$	–
StOMP	$\mathbf{A}, \mathbf{y}, T, t_S$	$j : \|\|c_j\|\| > t_S$	$\mathbf{y} - \mathbf{A}(\Lambda_i)\hat{\mathbf{h}}_i$	$\mathbf{A}(\Lambda_i)^\dagger \mathbf{y}$
CoSaMP	$\mathbf{A}, \mathbf{y}, s$	$2s$ biggest $\|\|c_j\|\|$	$\mathbf{y} - \mathbf{A}(\Lambda_i)\hat{\mathbf{h}}_i$	$supp_s(\mathbf{A}(\Lambda_i)^\dagger \mathbf{y})$
ROMP	$\mathbf{A}, \mathbf{y}, s$	s biggest $\|\|c_j\|\|$	$\mathbf{y} - \mathbf{A}(\Lambda_i)\hat{\mathbf{h}}_i$	$\mathbf{A}(\Lambda_i)^\dagger \mathbf{y}$
GOMP	$\mathbf{A}, \mathbf{y}, Q, s$	Q biggest $\|\|c_j\|\|$	$\mathbf{y} - \mathbf{A}(\Lambda_i)\hat{\mathbf{h}}_i$	$\mathbf{A}(\Lambda_i)^\dagger \mathbf{y}$
GOAMP	$\mathbf{A}, \mathbf{y}, Q$	Q biggest $\|\|c_j\|\|$	$\mathbf{y} - \mathbf{A}(\Lambda_i)\hat{\mathbf{h}}_i$	$\mathbf{A}(\Lambda_i)^\dagger \mathbf{y}$
GP	$\mathbf{A}, \mathbf{y}$	$\max_j \|\|c_j\|\|$	$\mathbf{b}_{i-1} - a_i \mathbf{A}(\Lambda_i)\mathbf{d}_i$	$\hat{\mathbf{h}}_{i-1} + a_i \mathbf{d}_i$

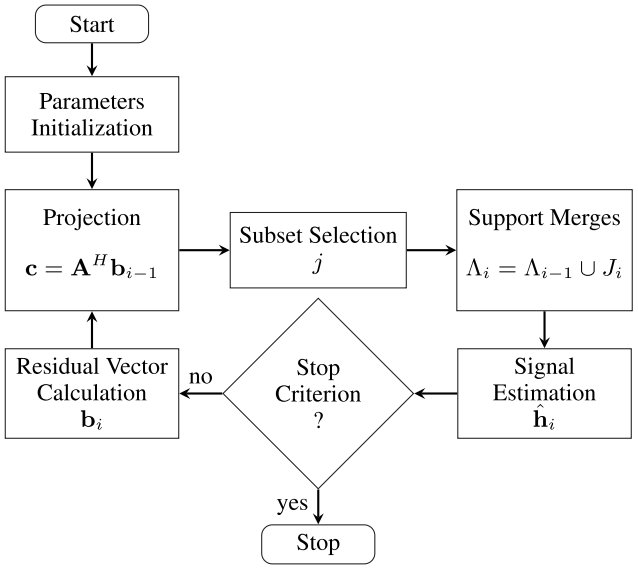

FIGURE 6. Greedy algorithms diagram.

The index set Λ_i stores the indices of the best aligned columns after i iterations. Let $\mathbf{D}_i$ be the matrix formed by the columns $\mathbf{a}_{k_i}$ chosen until iteration i, the next step is $\Lambda_i = \Lambda_{i-1} \cup k_i$ and $\mathbf{D}_i = [\mathbf{D}_{i-1}, a_{k_i}]$, if $k_i \notin \Lambda_{i-1}$. Otherwise, $\Lambda_i = \Lambda_{i-1}$ and $\mathbf{D}_i = \mathbf{D}_{i-1}$.

Then, a new residual vector is computed as (57) by removing the projection of $\mathbf{b}_{i-1}$ along this direction, and the estimated coefficient is calculated by (58).

$$\mathbf{b}_i = \mathbf{b}_{i-1} - P_{\mathbf{a}_{k_i}} \mathbf{b}_{i-1} = \mathbf{b}_{i-1} - \frac{(\mathbf{a}_{k_i}{}^H \mathbf{b}_{i-1})\mathbf{a}_{k_i}}{\|\mathbf{a}_{k_i}\|_2^2} \quad (57)$$

$$\hat{h}_i(k_i) = \hat{h}_{i-1}(k_i) + \frac{(\mathbf{a}_{k_i}{}^H \mathbf{b}_{i-1})}{\|\mathbf{a}_{k_i}\|_2^2} \quad (58)$$

The stop criterion of the algorithm can be, for example, $\|\mathbf{b}_i\| \leq \epsilon$. The signal estimate corresponds to the projections of the best columns of the matrix $\mathbf{A}$. An ASIC implementation of MP algorithm is proposed in [139].

2) MATCHING PURSUIT BASED ON LEAST SQUARES (MPLS)

Similarly to the MP algorithm, in each iteration of the Matching Pursuit based on Least Squares (MPLS) [140] algorithm, the column $\mathbf{a}_{k_i} \in \mathbf{A}$ which is best aligned with the residual vector $\mathbf{b}_{i-1}$ (where $\mathbf{b}_0 = \mathbf{y}$) is selected according to (59).

$$k_i = \arg\max_l |\mathbf{a}_l{}^H \mathbf{b_{i-1}}|, \quad l = 1, 2, \ldots, N \quad (59)$$

Let Λ_i be the index set of the best aligned columns of $\mathbf{A}$ until the iteration i, Λ_i is updated by $\Lambda_i = \Lambda_{i-1} \cup k_i$ if $k_i \notin \Lambda_{i-1}$. Otherwise, $\Lambda_i = \Lambda_{i-1}$.

Then, the new residual vector is computed as:

$$\mathbf{b}_i = \mathbf{b}_{i-1} - P_{\mathbf{a}_{k_i}} \mathbf{b}_{i-1} = \mathbf{b}_{i-1} - \frac{(\mathbf{a}_{k_i}{}^H \mathbf{b}_{i-1})\mathbf{a}_{k_i}}{\|\mathbf{a}_{k_i}\|_2^2} \quad (60)$$

The MP and the MPLS algorithms are different in the way that they calculate the non-zero signal components. In the MPLS algorithm, these components are estimated through the LS calculation only in the end of the algorithm.

After reaching the stop criterion, the signal is estimated by (61), where T is the number of iterations and $\mathbf{A}(\Lambda_T)$ is a submatrix of $\mathbf{A}$ consisting of the $\mathbf{a}_i$ columns with $i \in \Lambda_T$.

$$\hat{\mathbf{h}} = \mathbf{A}^\dagger(\Lambda_T)\mathbf{y} \quad (61)$$

3) ORTHOGONAL MATCHING PURSUIT (OMP)

The Orthogonal Matching Pursuit (OMP) algorithm is an improvement of the MP [141]. It can be stated as follows:

- Step 1: Initialize $\mathbf{b}_0 = \mathbf{y}$, $\Lambda_i = \emptyset$, and $i = 1$.
- Step 2: Find l that solves the maximization problem $\max_l \|P_{a_l} \mathbf{b_{i-1}}\|_2 = \max_l \frac{\mathbf{a}_l{}^H \mathbf{b}_{i-1}}{\|\mathbf{a}_l\|_2^2}$ and update $\Lambda_i = \Lambda_{i-1} \cup \{l\}$.
- Step 3: Calculate $\hat{\mathbf{h}}_i = \mathbf{A}^\dagger(\Lambda_i)\mathbf{y}$ and update $\mathbf{b}_i = \mathbf{y} - \mathbf{A}(\Lambda_i)\hat{\mathbf{h}}_i$.
- Step 4: Stop the algorithm if the stopping condition is achieved (e.g. $\|\mathbf{b}_i\| \leq \epsilon$). Otherwise, set $i = i + 1$ and return to Step 2.

In the OMP, the residual vector $\mathbf{b}_i$ is always orthogonal to the columns that have already been selected. Therefore, there will be no columns selected twice and the set of selected columns is increased through the iterations. Moreover, the sufficient and worst-case necessary conditions for recovering the signal sparsity are investigated in [142]. Furthermore, the condition for the exact support recovery with the OMP algorithm based on RIP and the minimum

magnitude of the non-zero taps of the signal are studied in [143] and [144].

Kulkarni and Mohsenin [145] propose two modifications to the OMP in order to reduce the hardware complexity of the OMP: Thresholding technique for OMP and Gradient Descent OMP. Reconfigurable, parallel, and pipelined architectures for the OMP and its two modifications are implemented on 65 nm CMOS technology operating at 1 V supply voltage to reconstruct data vector sizes ranging from 128 to 1024. These modifications lead to a 33% reduction in reconstruction time and to a 44% reduction in chip area when compared to the OMP ASIC implementation.

However, several other OMP hardware implementations are proposed in the literature [124], [146]–[152]. The Step 3, specifically the least squares operation, is the most costly part of the OMP implementation. The most used methods to deal with this are the QR decomposition and the Cholesky decomposition.

Due to the OMP selects only one column in each iteration, it is very sensitive to the selection of the index [153]. Alternatively, various approaches investigating multiple columns chosen in each iteration have been proposed such as the SP, the StOMP, the CoSaMP, the ROMP, the GOMP, the GOAMP, the MMP, and the GP algorithms. Furthermore, the Block Orthogonal Matching Pursuit (BOMP) algorithm [154] was developed to recover block sparse signals and its performance was investigated in [155] and [156].

4) SUBSPACE PURSUIT (SP)

At each stage, in order to refine an initially chosen estimate for the subspace, the Subspace Pursuit (SP) algorithm tests subsets of s columns in a group [157]. That is, maintaining s columns of $\mathbf{A}$, the algorithm executes a simple test in the spanned list of space, and after refines the list by discarding the unreliable candidates, retaining reliable ones while adding the same number of new candidates [157]. Basically, the steps of the SP are:

- Step 1: Initialize the support set Λ_0 with the s indices corresponding to the largest magnitude entries in the vector $\mathbf{A}^H\mathbf{y}$, the residual vector $\mathbf{b}_0 = \mathbf{y} - \mathbf{A}(\Lambda_0)\mathbf{A}(\Lambda_0)^\dagger\mathbf{y}$ and the iteration counter $i = 1$.
- Step 2: $\hat{\Lambda}_i = \Lambda_{i-1} \cup J_i$, where J_i is the set of the s indices corresponding to the largest magnitude entries in the vector $\mathbf{c}_i = \mathbf{A}^H\mathbf{b}_{i-1}$.
- Step 3: Calculate $\mathbf{x}_i = \mathbf{A}^\dagger(\hat{\Lambda}_i)\mathbf{y}$.
- Step 4: Update $\Lambda_i = \{$s indices corresponding to the largest magnitude elements of $\mathbf{x}_i\}$.
- Step 5: Update $\mathbf{b}_i = \mathbf{y} - \mathbf{A}(\Lambda_i)\mathbf{A}^\dagger(\Lambda_i)\mathbf{y}$.
- Step 6: Stop the algorithm if the stopping condition is achieved. Otherwise, set $i = i + 1$ and return to Step 2.

After T iterations, the signal estimated is given by $\hat{\mathbf{h}} = \mathbf{A}^\dagger(\Lambda_T)\mathbf{y}$.

When the signal is very sparse, the SP algorithm has computational complexity upper-bounded by $O(sMN)$ ($s \leq const.\sqrt{N}$), that is, lower computational complexity than the OMP algorithm [157]. However, when the non-zero components of the sparse signal decay slowly, the computational complexity of the SP can be further reduced to $O(MNlogs)$ [157].

5) STAGEWISE ORTHOGONAL MATCHING PURSUIT (STOMP)

The Stagewise Orthogonal Matching Pursuit (StOMP) [158] algorithm is inspired by the OMP. Different from the OMP algorithm, the StOMP algorithm selects multiple columns at each iteration. That is, according to a threshold, the StOMP algorithm selects the subspaces composed of the columns with the highest coherence between the remaining columns and the residual vector [158]. The number of iterations is fixed.

The input parameters are: the number of iterations T to perform, the threshold value t_S, the received signal $\mathbf{y}$, and the measurement matrix $\mathbf{A}$. The StOMP algorithm can be stated as follows:

- Step 1: Initialize the residual vector $\mathbf{b}_0 = \mathbf{y}$, $\Lambda_0 = \emptyset$, and $i = 1$.
- Step 2: Find $\mathbf{a}_l$ that $\|P_{a_l}\mathbf{b_{i-1}}\| > t_S$, that is, $\max_l \frac{\mathbf{a}_l{}^H\mathbf{b}_{i-1}}{\|\mathbf{a}_l\|^2} > t_S$ and add the $\mathbf{a}_l$ columns to the set of selected columns. Update $\Lambda_i = \Lambda_{i-1} \cup \{l\}$
- Step 3: Let $\hat{\mathbf{h}}_i = \mathbf{A}(\Lambda_i)^\dagger\mathbf{y}$. Update $\mathbf{b}_i = \mathbf{y} - \mathbf{A}(\Lambda_i)\hat{\mathbf{h}}_i$
- Step 4: If the stopping condition is achieved ($i = N_{it} = T$), stop the algorithm. Otherwise, set $i = i + 1$ and return to Step 2.

6) COMPRESSIVE SAMPLING MATCHING PURSUIT (COSAMP)

The Compressive Sampling Matching Pursuit (CoSaMP) algorithm is presented in [159] to mitigate the unstability of the OMP algorithm. Similarly to the OMP, it starts by initializing a residual vector as $\mathbf{b}_0 = \mathbf{y}$, the support set as $\Lambda_0 = \emptyset$, the iteration counter as $i = 1$, and additionally sets $\hat{\mathbf{h}}_0 = \mathbf{0}$. The CoSaMP performs these steps [159]:

- Step 1 - Identification: a proxy of the residual vector from the current samples is formed and the largest components of the proxy $\mathbf{c}_i = |\mathbf{A}^H\mathbf{b}_{i-1}|$ are located. The first $2s$ entries of $\mathbf{c}_i$ with largest absolute values are selected, and the indices selected compose J_i.
- Step 2 - Support merger: the set of newly identified components is united with the set of components that appears in the current approximation. $\Lambda_i = J_i \cup supp(\hat{\mathbf{h}}_{i-1})$ is defined as the augmentation of the support of the previous estimate $\hat{\mathbf{h}}_{i-1}$ with the $2s$ indices corresponding to the entries of $\mathbf{c}_i$ with largest absolute values.
- Step 3 - Estimation: a least-squares problem to approximate the target signal on the merged set of components is solved. $\hat{\mathbf{x}}_i = \mathbf{A}(\Lambda_i)^\dagger\mathbf{y}$.
- Step 4 - Pruning: a new approximation by retaining only the largest entries in this least-squares signal approximation is produced. $\hat{\mathbf{h}}_i$ is the first s entries of $\hat{\mathbf{x}}_i$ with largest absolute values.
- Step 5 - Sample update: update $\mathbf{b}_i = \mathbf{y} - \mathbf{A}(\Lambda_i)\hat{\mathbf{h}}_i$.

FPGA implementations of CoSaMP are presented in [160] and [161]. While an iterative Chebyshev-type method is used in [161] to calculate the matrix inversion process during the algorithm, [160] uses a QR decomposition method.

7) REGULARIZED OMP (ROMP)

The Regularized OMP (ROMP) algorithm was proposed in [162]. Firstly, the ROMP algorithm initializes $\Lambda_0 = \emptyset$ and the residual vector $\mathbf{b}_0 = \mathbf{y}$. Then, during each iteration i, the ROMP performs these three steps:

- Step 1 - Identification: $\hat{\Lambda}_i = \{$s biggest indices in magnitude of the projection vector $\mathbf{c}_i = \mathbf{A}^H \mathbf{b}_{i-1}\}$.
- Step 2 - Regularization: Among all subsets $J_i \subset \hat{\Lambda}_i$ with comparable coordinates $|\mathbf{c}(l)| \leq 2|\mathbf{c}(j)|$ for all $l, j \in J_i$, choose J_i with the maximal energy $\|\mathbf{c}(J_i)\|_2$.
- Step 3 - Updating: Add the set J_i to the index set: $\Lambda_i = \Lambda_i \cup J_i$. Calculate $\hat{\mathbf{h}}_i = \mathbf{A}(\Lambda_i)^\dagger \mathbf{y}$ and update the residual vector $\mathbf{b}_i = \mathbf{y} - \mathbf{A}(\Lambda_i)\hat{\mathbf{h}}_i$.

The regularization step can be done in linear time. The running time of the ROMP is comparable to that of the OMP in theory, but it is often better than the OMP in practice [162].

8) GENERALIZED ORTHOGONAL MATCHING PURSUIT (GOMP)

The Generalized Orthogonal Matching Pursuit (GOMP) algorithm is a direct extension of the OMP algorithm [163]. The GOMP selects $Q \geq 1$ largest correlation columns of the matrix $\mathbf{A}$ with the residual vector $\mathbf{b}$. When $Q = 1$, the GOMP becomes the OMP. Moreover, $Q \leq s$ and $Q \leq \sqrt{M}$. The steps of the GOMP are:

- Step 1: Initialize the residual vector $\mathbf{b}_0 = \mathbf{y}$, $\Lambda_0 = \emptyset$ and $i = 1$.
- Step 2: Find the Q biggest $\mathbf{a}_{l_1}, .., \mathbf{a}_{l_Q}$ columns that solves the maximization problem $\max_k \|P_{a_{l_k}} \mathbf{b_{i-1}}\|_2 = \max_k \frac{\mathbf{a}_{l_k}{}^H \mathbf{b}_{i-1}}{\|\mathbf{a}_{l_k}\|_2^2}$ and add the $\mathbf{a}_{l_i}$ columns to the set of selected columns. Update $\Lambda_i = \Lambda_{i-1} \cup \{l_1, \ldots, l_Q\}$.
- Step 3: Calculate $\hat{\mathbf{h}}_i = \mathbf{A}(\Lambda_i)^\dagger \mathbf{y}$. Update $\mathbf{b}_i = \mathbf{y} - \mathbf{A}(\Lambda_i)\hat{\mathbf{h}}_i$.
- Step 4: Stop the algorithm if the stopping condition is achieved ($N_{it} = \min(s, M/Q)$ or $\|\mathbf{b}_i\|_2 \leq \epsilon$). Otherwise, set $i = i + 1$ and return to Step 2.

The complexity of the GOMP algorithm is approximately $2N_{it}MN + (2Q^2 + Q)N_{it}^2 M$ [163]. The RIP based sufficient conditions for the exact support recovery with the GOMP in the noisy case are investigated in [164].

9) GENERALIZED ORTHOGONAL ADAPTIVE MATCHING PURSUIT (GOAMP)

The Generalized Orthogonal Adaptive Matching Pursuit (GOAMP) algorithm considers that the signal's sparsity is not known, so it adapts the variable Q of the GOMP algorithm during the iterations [165]. Basically, the GOAMP inserts a new Step after the update of the residual vector:

- Step 1: Initialize the residual vector $\mathbf{b}_0 = \mathbf{y}$, $\Lambda_0 = \emptyset$ and $i = 1$.
- Step 2: Find the Q biggest $\mathbf{a}_{l_1}, .., \mathbf{a}_{l_Q}$ columns that solve the maximization problem $\max_k \|P_{a_{l_k}} \mathbf{b_{i-1}}\|_2 = \max_k \frac{\mathbf{a}_{l_k}{}^H \mathbf{b}_{i-1}}{\|\mathbf{a}_{l_k}\|_2^2}$ and add the $\mathbf{a}_{l_i}$ columns to the set of selected columns. Update $\Lambda_i = \Lambda_{i-1} \cup \{l_1, \ldots, l_Q\}$.
- Step 3: Calculate $\hat{\mathbf{h}}_i = \mathbf{A}(\Lambda_i)^\dagger \mathbf{y}$. Update $\mathbf{b}_i = \mathbf{y} - \mathbf{A}(\Lambda_i)\hat{\mathbf{h}}_i$.
- Step 4: If $\|\mathbf{b}_{i-1} - \mathbf{b}_i\|_2^2 / \|\mathbf{b}_{i-1}\|_2^2 < \epsilon_2$, $Q = f(Q)$. Otherwise, go to Step 5.
- Step 5: Stop the algorithm if the stopping condition is achieved ($\|\mathbf{b}_i\|_2 \leq \epsilon_1$). Otherwise, set $i = i + 1$ and return to Step 2.

where $f(Q)$ is a function that increases the value of Q. According to [165], ϵ_2 is about $0.7 - 0.9$.

10) GRADIENT PURSUIT (GP)

The Gradient Pursuit (GP) algorithms were proposed in [166] as variations of the MP algorithm. In the GP, at iteration i, the signal estimate $\hat{\mathbf{h}}_i$ is:

$$\hat{\mathbf{h}}_i = \hat{\mathbf{h}}_{i+1} + \gamma_i \mathbf{d}_i \tag{62}$$

where $\mathbf{d}_i$ is the update direction and γ_i is the optimal step size defined by:

$$\gamma_i = \frac{< \mathbf{b}_{i-1}, \mathbf{A}(\Lambda_i)\mathbf{d}_i >}{\|\mathbf{A}(\Lambda_i)\mathbf{d}_i\|} \tag{63}$$

In the MP and the OMP algorithms, the update direction is taken to be in the direction of the best aligned column of the matrix $\mathbf{A}$. In the OMP, once added, the column will not be selected again as the process of orthogonalisation ensures that all future residuals will remain orthogonal to all currently selected columns. However, in the MP and the GP the orthogonality is not ensured. Hence, it is possible select again the same column.

Each iteration i consists into find the column $\mathbf{a}_{l_i} \in \mathbf{A}$ which is best aligned with the signal vector residual $\mathbf{b}_{i-1}$. The GP algorithms perform these steps:

- Step 1: Initialize $\mathbf{b}_0 = \mathbf{y}$, $\Lambda_0 = \emptyset$ and $i = 1$.
- Step 2: Find l_i that solves the maximization problem $\max_{l_i} \|P_{a_{l_i}} \mathbf{b_{i-1}}\|_2 = \max_{l_i} \frac{\mathbf{a}_{l_i}{}^H \mathbf{b}_{i-1}}{\|\mathbf{a}_{l_i}\|_2^2}$. Update $\Lambda_i = \Lambda_{i-1} \cup \{l_i\}$.
- Step 3: Update the direction $\mathbf{d}_i$. Calculate $\gamma_i = \frac{<\mathbf{b}_{i-1}, \mathbf{A}(\Lambda_i)\mathbf{d}_i>}{\|\mathbf{A}(\Lambda_i)\mathbf{d}_i\|}$ and $\hat{\mathbf{h}}_i = \hat{\mathbf{h}}_{i-1} + \gamma_i \mathbf{d}_i$. Update $\mathbf{b}_i = \mathbf{b}_{i-1} - \gamma_i \mathbf{A}(\Lambda_i)\mathbf{d}_i$.
- Step 4: Stop the algorithm, if the stopping condition is achieved. Otherwise, set $i = i + 1$ and return to Step 2.

There are three different methods for calculating the update direction $\mathbf{d}_i$ [11], [166]:

- Gradient Pursuit: uses the direction that minimises $\|\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}_{i-1}\|_2$, that is:

$$\mathbf{d}_i = \mathbf{A}^T(\Lambda_i)\left(\mathbf{y} - \mathbf{A}(\Lambda_i)\hat{\mathbf{h}}_{i-1}(\Lambda_i)\right) \tag{64}$$

- Conjugate Gradient Pursuit: it is a directional optimization algorithm that is guaranteed to solve quadratic optimization problems in as many steps as the dimension of the problem [167]. Let $\phi(\mathbf{h}) = \frac{1}{2}\mathbf{h}^T\mathbf{G}\mathbf{h} - \mathbf{f}^T\mathbf{h}$ be the cost function to be minimised, this method chooses $\mathbf{d}_i$ that is $\mathbf{G}$-conjugate to all the previous directions, that is:

$$\mathbf{d}_i\mathbf{G}\mathbf{d}_k = 0, \quad \forall k < i \tag{65}$$

  In this case, $\mathbf{G} = \mathbf{A}^T(\Lambda_i)\mathbf{A}(\Lambda_i)$. Let $\mathbf{D}_i$ be the matrix whose columns are the update directions for the first i iterations and let $\mathbf{g}_i$ be the gradient of the the cost function in iteration i, the new update direction $\mathbf{d}_i$ in iteration i is given by [167]:

$$\mathbf{d}_i = \mathbf{g}_i + \mathbf{D}_{i-1}\mathbf{f} \tag{66}$$

  where $\mathbf{f} = -\left(\mathbf{D}_{i-1}^T\mathbf{G}\mathbf{D}_{i-1}\right)^{-1}\left(\mathbf{D}_{i-1}^T\mathbf{G}\mathbf{g}_{i-1}\right)$.
  The OMP uses a full conjugate gradient solver at every iteration. Instead, in this method, only a directional update step occurs for each new added element.
- Approximate Conjugate Gradient Pursuit: the new direction is conjugate to the previous direction, but this can be extended to a larger number of directions:

$$\mathbf{d}_i = \mathbf{g}_i + \mathbf{d}_{i-1}f \tag{67}$$

  The $\mathbf{G}$-conjugacy implies that:

$$\langle(\mathbf{G}\mathbf{d}_{i-1}), (\mathbf{g}_i + b\mathbf{d}_{i-1})\rangle = 0 \tag{68}$$

$$f = -\frac{\langle(\mathbf{A}(\Lambda_i)\mathbf{d}_{i-1}), (\mathbf{A}(\Lambda_i)\mathbf{g}_i)\rangle}{\|\mathbf{A}(\Lambda_i)\mathbf{d}_{i-1}\|_2^2} \tag{69}$$

11) MULTIPATH MATCHING PURSUIT (MMP)

With the help of the greedy strategy, the Multipath Matching Pursuit (MMP) algorithm executes the tree search [153]. First, the MMP algorithm searches multiple promising columns of the matrix $\mathbf{A}$ candidates and then it chooses one minimizing the residual in the final moment. The MMP algorithm can not be represented by Fig. 6. Let L be the number of child paths of each candidate, f_i^k be the k^{th} candidate in the i^{th} iteration, $F_i = \{f_i^1, \ldots, f_i^u\}$ be the set of candidates in the i^{th} iteration and $|F_i|$ be the number of elements of F_i, Ω^k is the set of all possible combinations of k columns in $\mathbf{A}$, for example, if $\Omega = \{1, 2, 3\}$ and $k = 2$, then $\Omega^k = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$ [153].

Fig. 7 shows a comparison from an hypothetical choice of columns in the 3 first iterations of the OMP and the MMP algorithms. In this figure, the OMP selects the column with index 2 in the first iteration, then the index 1 in the next iteration and in the third iteration, it selects the index 4. On the other hand, the MMP algorithm selects the index 2 and 4 in the first iteration, after for each index selected, the algorithm will select others $L = 2$ index in each iteration. Then, in the second iteration, it selects the index 1 and 5 for the index 2 and for the index 4, but it is not necessary select the same index as can be noted in the third iteration where the MMP selects the index 4 and 5 for the $\{2, 1\}$ composing $f_3^1 = \{2, 1, 4\}$ and $f_3^2 = \{2, 1, 5\}$, and the index 2 and 3 for the $\{4, 1\}$ composing $f_3^1 = \{2, 1, 4\}$ and $f_3^5 = \{4, 1, 3\}$. Moreover, it can be noticed that although the number of candidates increases as an iteration goes on (each candidate brings forth multiple children), the increase is actually moderate since many candidates are overlapping in the middle of search as the case of f_3^1, f_3^2 and f_3^3 in [153, Fig. 7].

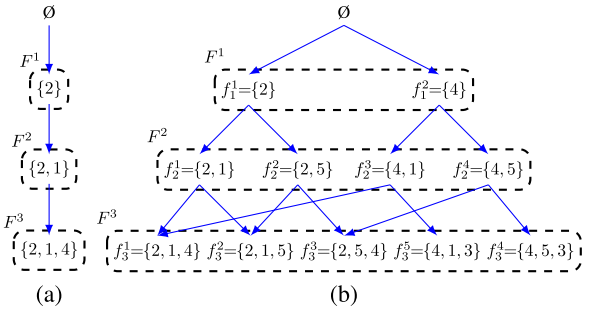

FIGURE 7. Comparison between the OMP and the MMP algorithms ($L = 2$): (a) OMP (b) MMP [153].

The residual vector of the k^{th} candidate in the i^{th} iteration is $\mathbf{b}_i^k = \mathbf{y} - \mathbf{A}(f_i^k)\hat{\mathbf{h}}_i^k$, where $\mathbf{A}(f_i^k)$ is the matrix $\mathbf{A}$ using only the columns indexed by f_i^k. Given the measurement matrix $\mathbf{A}$, the received signal $\mathbf{y}$, the signal's sparsity s and the parameter L, the MMP follows the steps bellow:

- Step 1: Initialize $\mathbf{b}_0 = \mathbf{y}$, $F_0 = \emptyset$ and $i = 1$.
- Step 2: Set $F_i = \emptyset$, $u = 0$ and $k = 1$.
- Step 3: Choose L best indices of columns that solve the maximization problem $\arg\max \|\mathbf{A}^H\mathbf{b}_{i-1}^k\|_2^2$ to compose π and set $j = 1$.
- Step 4: Set $f^{temp} = f_{i-1}^k \cup \{\pi_j\}$, where π_j is the j^{th} element of the set π.
- Step 5: If $f^{temp} \notin F_i$ then $u = u + 1$, $f_i^u = f^{temp}$, $F_i = F_i \cup \{f_i^u\}$, update $\hat{\mathbf{h}}_i^u = \mathbf{A}^\dagger(f_i^u)\mathbf{y}$ and $\mathbf{b}_i^u = \mathbf{y} - \mathbf{A}(f_i^u)\hat{\mathbf{h}}_i^u$. Otherwise, go to Step 6.
- Step 6: Set $j = j + 1$. If $j \leq L$ then go to Step 4. Otherwise, go to Step 7.
- Step 7: Set $k = k + 1$. If $k \leq |F_{i-1}|$ then go to Step 3. Otherwise, go to Step 8.
- Step 8: Set $i = i+1$. If $i > s$ then go to Step 9. Otherwise, go to Step 2.
- Step 9: Find the index of the best candidate, that is, $u^* = \arg\min_u \|\mathbf{b}_s^u\|_2^2$. Set $\Lambda = f_s^{u^*}$ and calculate the estimate signal $\hat{\mathbf{h}} = \mathbf{A}^\dagger(\Lambda)\mathbf{y}$.

If the $\arg\max \|\mathbf{A}^H\mathbf{b}_{i-1}^k\|_2^2$ in the Step 3 is calculated as in the OMP algorithm, the MMP algorithm is called Tree-based Orthogonal Matching Pursuit (TOMP) algorithm [168].

12) ITERATIVE HARD THRESHOLDING (IHT)

The Iterative Hard Thresholding (IHT) algorithm [169] is an iterative method that performs some thresholding function on each iteration. This algorithm can't be represented by Fig. 6. Let $\hat{\mathbf{h}}_0 = \mathbf{0}$, $i = 1$, for each iteration:

$$\hat{\mathbf{h}}_i = H_s(\hat{\mathbf{h}}_{i-1} + \mathbf{A}^H(\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}_{i-1})) \tag{70}$$

where $H_s()$ is a non-linear operator that sets all elements to zero except the s elements having largest amplitudes.

The IHT algorithm can stop after a fixed number of iterations or it can terminate when the sparse vector does not change much between consecutive iterations, for example [170].

D. OTHER ALGORITHMS

This work presents some sparse recovery algorithms. However, if the reader wants to know other algorithms, in addition to the vast list presented above, some of them can be found in: Back-tracking based Adaptive Orthogonal Matching Pursuit (BAOMP) [171], Chaining Pursuit (CP) [172], Conjugate Gradient Iterative Hard Thresholding [173], Differential Orthogonal Matching Pursuit (D-OMP) [174], Fast Iterative Shrinkage Thresholding Algorithm (FISTA) [130], Forward-Backward Pursuit (FBP) [175], Fourier sampling algorithm [176], Hard Thresholding Pursuit [177], Heavy Hitters on Steroids (HHS) [178], Normalized Iterative Hard Thresholding [179], l_p-Regularized Least-Squares Two Pass [180], Sequential Least Squares Matching Pursuit (SLSMP) [115], Sparse Adaptive Orthogonal Matching Pursuit (SpAdOMP) [181], Sparse Reconstruction by Separable Approximation (SpaRSA) [182], Stochastic Gradient Pursuit (SGP) [183], Stochastic Search Algorithms [184], Tree Search Matching Pursuit (TSMP) [185], and Vector Approximate Message Passing (VAMP) [186].

V. ALGORITHM DISCUSSION

This section presents a generic comparison of some algorithms previously mentioned and some performance comparison found in the literature. Moreover, open research challenges related to the CS theory, especially concerned to sparse recovery algorithms, are presented.

A. GENERIC DISCUSSION

1) BP × OMP

While the OMP algorithm begins with an empty set of columns and adds to the support set only the most important new column among all those in each step, the BP-simplex begins with a ''full'' index set and then iteratively improves this set by placing negligible terms with useful new ones [111].

2) OMP × SP

The difference between the OMP and the SP algorithms lies in the way that they generate the index set Λ_i. In the case of the OMP, after an index is included in the set Λ_i, it stays there during all the process. In a different way from this, SP holds an estimate Λ_i with size s that is refined in each iteration [157]. Hence, the index can be added to or removed from the estimated support set at any iteration of the algorithm [157]. Moreover, the SP solves two least squares problems in each iteration while the OMP solves only one.

3) SP × CoSaMP

The SP and the CoSaMP algorithms add the new candidate columns in a different way. With the SP algorithm, only s index are added at each iteration, while $2s$ columns are added in the CoSaMP algorithm [157]. Futhermore, the signal estimate $\hat{\mathbf{h}}$ is different for SP and CoSaMP. While the SP solves a LS problem again to get the final approximation of the current iteration, the CoSaMP algorithm keeps the s largest components of $\hat{\mathbf{x}}$ [157]. Therefore, the SP solves two least squares problems in each iteration whereas the CoSaMP solves only one [157], [187].

4) STOMP × OMP

The StOMP and the OMP algorithms differ in the number of columns selected at each iteration: the OMP selects one column, while the StOMP selects several columns. Thus, the StOMP algorithm is generally faster than the OMP. StOMP can produce a good approximation with a small number of iterations [15], but it has to determine an appropriate threshold value since different threshold values could lead to different results [15], [158].

5) ROMP × SP

The ROMP and the SP algorithms generate the support set Λ_i in a different way. The ROMP algorithm generates it sequentially, by adding one or many reliable indexes to the existing list in each iteration. While the SP algorithm re-evaluates all the indexes at each iteration, in the ROMP algorithm, an index added to the list can not be removed [162].

6) ROMP × StOMP

The threshold value is the difference between the ROMP and the StOMP algorithms. While the ROMP uses all columns that have a dot product above half the size of the largest dot product, the StOMP uses a preset threshold value [162].

7) GP × IHT

The difference between the GP and the IHT algorithms is in how the sparsity constraint is enforced. For the gradient pursuit algorithms, in each iteration, a new dictionary element is added and cannot be removed afterwards. Otherwise, in the IHT algorithm, the indices can be added and removed because it keeps only the most important (decided by the largest magnitude) dictionary elements [166].

8) TOMP × OMP

The major difference between the TOMP and the OMP algorithms is the manner (quantity) of selecting the columns of the measurement matrix $\mathbf{A}$. While the TOMP algorithm sequentially selects the whole next ''good'' family of columns, the OMP algorithm sequentially selects the next ''good'' column [168].

B. PERFORMANCE DISCUSSION

A performance comparison between algorithms from each category is analyzed below. From the Convex Relaxation category, the AMP and FISTA algorithms were implemented.

The BCS via RVM was implemented representing the Non-convex Optimization category. And finally, from the Greedy algorithms, the MP and OMP were implemented.

Let N_s be the number of realizations, the average normalized mean squared error (NMSE) described by (71) is used to evaluate the algorithms in terms of the size of the measured signal y (M) and the signal's sparsity.

$$NMSE = \frac{1}{N_s} \sum \frac{\|h - \hat{h}\|_2^2}{\|h\|_2^2} \tag{71}$$

The system model is defined by (6). For these simulations, $N_s = 1000$, $N = 1024$, A is i.i.d. Gaussian, with elements distributed $\mathcal{N}(0, M^{-1})$. The sparse signal h to be estimated is Bernoulli-Gaussian, that is, its elements are i.i.d $\mathcal{N}(0, 1)$ with probability γ and the others are set to 0. Signal-to-noise ratio (SNR) is 30 dB. The results are compared to the theoretical performance bound "Oracle". It has the previous knowledge of the non-zero tap positions. The non-zero coefficients are calculated by applying the Least Square algorithm using the submatrix A_s composed of the column related to the non-zero tap positions of the signal to be estimated.

First, the algorithms performances are analyzed varying the size of M for $\gamma = 0.05$ as shown in Fig. 8.

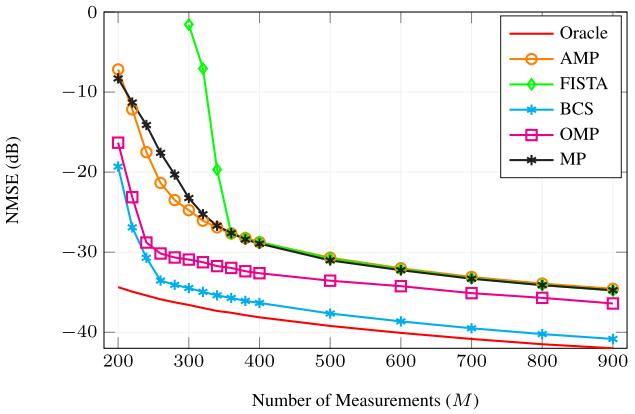

FIGURE 8. Algorithms performances varying M for $\gamma = 0.05$.

It can be seen that the performances of all the algorithms increase when the number of measurements M increases. However, it can be noticed that a low M value ($M < N$) allows the algorithms to recover the sparse signal resulting in low NMSE values. Among the algorithms analyzed, the BCS presents the best performance. Furthermore, its performance is close to the one achieved by the "Oracle". It confirms the good results achieved by the algorithms from the Bayesian theory.

The algorithms performances are also analyzed varying γ for $M = 512$ as shown in Fig. 9.

According to Fig. 9, as the signal becomes less sparse (i.e. γ increases), the performances of all algorithms decrease, that is, the NMSE values increase. When γ is low, BCS is the algorithm that achieves the best performance (lower NMSE value), which is close to the one achieved by

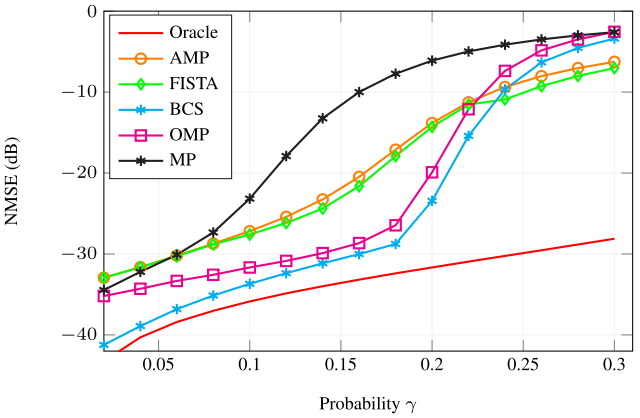

FIGURE 9. Algorithms performances varying γ for $M = 512$.

the "Oracle". However, when the signal to be estimated is less sparse (big γ values), FISTA shows a better performance in recovering the signal.

Table 4 shows the percentage of non-zero tap positions correctly found for the five algorithms analyzed. The result of FISTA is not presented for $M = 200$ because in this scenario this algorithm did not converge. It can be observed that when the M value increases, the percentage of non-zero tap positions correctly found increases. Moreover, notice that although AMP and FISTA algorithms present the highest percentage values for $M = 400$ and $\gamma = 0.05$, the algorithms BCS and OMP are the ones that achieve the best results in terms of NMSE (see Fig. 8). It means that even if BCS and OMP correctly find less non-zero tap positions than the algorithms AMP and FISTA, BCS and OMP are better able to estimate the non-zero coefficients resulting in lower NMSE values.

TABLE 4. Percentage of non-zero tap positions correctly found.

Algorithm	$\gamma = 0.05$		$\gamma = 0.1$	$\gamma = 0.2$
	$M = 200$	$M = 400$	$M = 512$	
AMP	81.7%	98.4%	97.7%	88.5%
FISTA	-	98.5%	98.2%	89.5%
BCS	93.9%	97.0%	95.8%	91.9%
OMP	92.1%	96.7%	95.8%	92.9%
MP	67.6%	96.8%	96.2%	70.1%

Moreover, it can be observed that when the γ value increases, that is, the signal become less sparse, the percentage of non-zero tap positions correctly found decreases. This occurs for all the algorithms analyzed and confirms what was suggested in Fig. 9.

Other performance comparisons between other algorithms can be found in the literature. Some of them are presented below.

A performance comparison of the SP, the OMP, the ROMP, the GOMP, and the GOAMP algorithms is made in [12] for the reconstruction of an image. The recovery performance was analyzed in the form of Peak Signal to Noise Ratio (PSNR) value achieved and running time elapsed.

From these simulations, the PSNR value is better when the GOAMP algorithm is used.

Arjoune *et al.* [31] compare the BCS, the BP, the GraDeS, the OMP, and the IHT algorithms to estimate a noisy sparse signal of length $N = 1024$. The metrics used were: phase transition diagram, recovery time, recovery error, and covariance. The results show that techniques of convex relaxation perform better in terms of recovery error, while greedy algorithms are faster, and Bayesian based techniques appear to have an advantageous balance of small recovery error and a short recovery time [31].

A comparison between the OMP and the modified LARS for solving LASSO algorithms is made in [120] considering the solution accuracy and the convergence time. The results show that generally the OMP requires fewer iterations than the LARS to converge to the final solution, suggesting that the OMP is much faster than the LARS [120]. However, for the cases where some columns of $\mathbf{A}$ are highly correlated, the OMP was considered less accurate than the LARS [120].

Wang *et al.* [163] compare the GOMP, the OMP, the StOMP, the ROMP, and the CoSaMP algorithms for a measurement matrix $\mathbf{A}$ 128×256 generated by a Gaussian distribution $N(0, 1/128)$. The sparse signal varies from $s = 1$ to $s = 70$ and it is generated in two ways: Gaussian signals and pulse amplitude modulation (PAM) signals. The results show that the critical sparsity of the GOMP algorithm is larger than that of the OMP, the ROMP, the StOMP, and the CoSaMP algorithms [163].

Algorithms OMP, StOMP, CoSaMP, MMP, and BPDN are compared in [153] varying the SNR for two different sparsity values ($s = 20$ and $s = 30$). The 100×256 measurement matrix is generated by a Gaussian distribution. The results show that the MMP performs close to the OMP to $s = 20$, but for $s = 30$, the performance of the MMP is better [153]. Moreover, the running time of these algorithms is shown as a function of s. The MMP algorithm has the highest running time and the OMP and the StOMP algorithms have the lowest running time among algorithms under test [153].

In [137], the authors compare the performance of the IRLS algorithm using the regularization. The results show that for $p = 1$ the unregularized IRLS and regularized IRLS are almost identical but for $p = 0$ and $p = 1/2$, the regularized IRLS algorithm recovers the greatest range of signals [137].

The authors in [168] compare the performance of the TOMP, the BP, and the OMP algorithms. According to their results, TOMP needs less iteration than the OMP because the TOMP algorithm selects the whole tree at a time and not only one element. Moreover, the TOMP can achieve better results than the BP and the OMP in reconstruction quality [168].

In [124], the authors implement the algorithms OMP and AMP in FPGA. As the OMP processing time increases quadratically with the number of non-zero coefficients of the signal to be estimated, this algorithm is more suitable to recover very sparse signals. On the other hand, if the signal to be estimated has several non-zero components it is more efficient to use the AMP algorithm to recover the signal than the OMP.

The algorithms GraDeS and LARS have complexity $O(MN)$. Table 5 presents the complexity of other algorithms as well as the minimum measurement (M) requirement.

TABLE 5. Complexity and minimum measurement (M) requirement.

Algorithm	Complexity	M	Reference
BP	$O(N^3)$	$O(slogN)$	[5], [111], [157]
CoSaMP	$O(MN)$	$O(slogN)$	[5], [18], [159]
IHT	$O(MN)$	$O(slog(N/s))$	[170]
MP	$O(MNN_{it})$	$O(slog(N/s))$	[138], [188]
OMP	$O(sMN)$	$O(slogN)$	[5], [18], [157], [162]
ROMP	$O(sMN)$	$O(slog^2N)$	[5], [18], [159], [162]
SP	$O(sMN)$	$O(slog(N/s))$	[5], [157]
StOMP	$O(NlogN)$	$O(NlogN)$	[5], [18], [158]

The storage cost per iteration (number of floating point numbers) of some sparse recovery algorithms presented in [189] are reproduced in Table 6, where E is the computational cost of storing $\mathbf{A}$ or $\mathbf{A}^T$, k is the size of the support set Λ_i in the iteration i, and η is the number of conjugate gradient steps used per iteration that is lower or equal to the number of elements selected.

TABLE 6. Storage cost [189].

Algorithm	Storage cost
GP	$2m + E + 2k + N$
MP	$E + M + 2k + N$
OMP	$2Mk + 0.5k^2 + 2.5k + E + N$
StOMP	$2M + E + 2k + N$

Finally, Table 7 presents the recovery condition related to the RIC value of the matrix A of some algorithms.

TABLE 7. The RIC value of the matrix *A*.

Algorithm	The RIC value	Reference
BP	$\delta_{2s} < \sqrt{2} - 1$ or $\delta_s + \delta_{2s} + \delta_{3s} < 1$	[127]
CoSaMP	$\delta_{4s} < 0.1$	[127]
GraDeS	$\delta_{2s} < 1/3$	[127]
IHT	$\delta_{3s} < 1/\sqrt{32}$ and $\delta_{2s} < 0.25$	[190]
LARS	$\delta_{2s} < \sqrt{2} - 1$ or $\delta_s + \delta_{2s} + \delta_{3s} < 1$	[127]
OMP	$\delta_{s+1} < \frac{1}{\sqrt{s+1}}$	[142]
ROMP	$\delta_{2s} < \frac{0.03}{\sqrt{log(s)}}$	[127]
SP	$\delta_{3s} < 0.06$	[127]

C. RESEARCH CHALLENGES

As can be observed, several papers have addressed CS applications, designing of the measurement matrix, and sparse recovery algorithms. However, there are still many research challenges to overcome.

Each domain of application has its characteristics and these should be used to improve the estimation of the sparse signal. For example, Pareschi *et al.* [50] optimize the sensing matrix

by a proper specialization of a specific sparsity matrix taking advantage of the input signal statistical features.

As said before, the two important challenges addressed to compressive sensing are the design of the measurement matrix and the development of an efficient recovery algorithm.

Concerning the first challenge, while random measurement matrices have been widely studied, only a few deterministic measurement matrices have been considered [2]. However, in structures that allow fast implementation with reduced storage requirements, deterministic measurement matrices are highly desirable [2]. Therefore, this domain can be improved.

Addressed to the second challenge, a lot of CS approaches assume that the signal's sparsity is known. However, in several applications such as the cognitive radio networks it is not true. Thus, it is necessary to develop sparse recovery algorithms that do not need this information and that are able to be adaptive to time changes. Another alternative is to develop a sparsity order estimation method, so the sparsity order can be accurately estimated before using a recovery algorithm [191].

Moreover, in some cases, the signal's sparsity can be time-varying. Hence, the investigation of adaptive sparsity order estimation methods to capture the dynamicity of the signal of interest constitutes an important research challenge [72].

Another opportunity is the development of the sparse recovery algorithm on a distributed platform such as a sensor network as is made in [192].

Furthermore, sparse recovery algorithms can be combined with deep learning to improve sparse signal recovery. Some different deep-learning approaches to solve sparse linear inverse problems have already been reported in the literature [193]–[197]. However, some improvements in their performances can be suggested. For example vary the optimization algorithm used in the neural network, the loss function, or the activation function of some of these neural network, or even suggest a new neural network to sparse signal estimation in order to produce faster and accurate results.

VI. CONCLUSION

The compressive sensing and its sparse recovery algorithms are used in several areas and have been extensively studied in this work. With growing demand for cheaper, faster, and more efficient devices, the usefulness of compressive sensing theory is progressively greater and more important. This paper has provided a review of this theory. We have presented mathematical and theoretical foundations of the key concepts. More specifically, we have focused on the sparse recovery algorithms illustrating numerous algorithms. Some comparisons of them were also discussed. Furthermore, several applications of compressive sensing have been presented such as Image and Video, Compressive Transmission Data, Systems Communication, and Detection and Recognition Systems. The importance of choosing an efficient sparse recovery algorithm to increase the performance of the sparse signal estimation was also highlighted. As shown in the previous sections of this paper, the compressive sensing theory can provide useful and promising techniques in future. Indeed, this theme is in significant and wide development in several applications. However, it still faces a number of open research challenges. For example, to determine the suitable measurement matrix and develop a sparse recovery algorithm that does not need know the signal's sparsity and can be adaptive to time-varying sparsity. Moreover, signal statistical information can be added in the CS acquisition or CS reconstruction to reduce the amount of required resources (time, hardware, energy, etc.).

REFERENCES

[1] M. Amin, *Compressive Sensing for Urban Radar*, 1st ed. Boca Raton, FL, USA: CRC Press, 2017.

[2] M. M. Abo-Zahhad, A. T. Hussein, and A. M. Mohamed, "Compressive sensing algorithms for signal processing applications: A survey," *Int. J. Commun., Netw. Syst. Sci.*, vol. 8, no. 6, pp. 197–216, 2015.

[3] C. E. Shannon, "Communication in the presence of noise," *Proc. Inst. Radio Eng.*, vol. 37, no. 1, pp. 10–21, Jan. 1949.

[4] S. K. Sharma, E. Lagunas, S. Chatzinotas, and B. Ottersten, "Application of compressive sensing in cognitive radio communications: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1838–1860, 3rd Quart., 2016.

[5] S. Qaisar, R. M. Bilal, W. Iqbal, M. Naureen, and S. Lee, "Compressive sensing: From theory to applications, a survey," *J. Commun. Netw.*, vol. 15, no. 5, pp. 443–456, 2013.

[6] S. Sharma, A. Gupta, and V. Bhatia, "A new sparse signal-matched measurement matrix for compressive sensing in UWB communication," *IEEE Access*, vol. 4, pp. 5327–5342, 2016.

[7] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd ed. New York, NY, USA: Academic, 2008.

[8] Z. Qin, J. Fan, Y. Liu, Y. Gao, and G. Y. Li, "Sparse representation for wireless communications: A compressive sensing approach," *IEEE Signal Process. Mag.*, vol. 35, no. 3, pp. 40–58, May 2018.

[9] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[10] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[11] G. Pope, "Compressive sensing: A summary of reconstruction algorithms," M.S. thesis, Dept. Comput. Sci., Eidgenössische Technische Hochschule, Zürich, Switzerland, 2009.

[12] S. B. Budhiraja, "A survey of compressive sensing based greedy pursuit reconstruction algorithms," *Int. J. Image, Graph. Signal Process.*, vol. 7, no. 10, pp. 1–10, 2015.

[13] T. Akhila and R. Divya, "A survey on greedy reconstruction algorithms in compressive sensing," *Int. J. Res. Comput. Commun. Technol.*, vol. 5, no. 3, pp. 126–129, 2016.

[14] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[15] K. V. Siddamal, S. P. Bhat, and V. S. Saroja, "A survey on compressive sensing," in *Proc. 2nd Int. Conf. Electron. Commun. Syst. (ICECS)*, Feb. 2015, pp. 639–643.

[16] T. L. Nguyen and Y. Shin, "Deterministic sensing matrices in compressive sensing: A survey," *Sci. World J.*, vol. 2013, Sep. 2013, Art. no. 192795.

[17] A. Gilbert and P. Indyk, "Sparse recovery using sparse matrices," *Proc. IEEE*, vol. 98, no. 6, pp. 937–947, Jun. 2010.

[18] F. Salahdine, N. Kaabouch, and H. El Ghazi, "A survey on compressive sensing techniques for cognitive radio networks," *Phys. Commun.*, vol. 20, pp. 61–73, Sep. 2016.

[19] J. W. Choi, B. Shim, Y. Ding, B. Rao, and D. I. Kim, "Compressed sensing for wireless communications: Useful tips and tricks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1527–1550, 3rd Quart., 2017.

[20] U. P. Shukla, N. B. Patel, and A. M. Joshi, "A survey on recent advances in speech compressive sensing," in *Proc. Int. Mutli-Conf. Autom., Comput., Commun., Control Compressed Sens. (iMac4s)*, Mar. 2013, pp. 276–280.
[21] G. Wunder, H. Boche, T. Strohmer, and P. Jung, "Sparse signal processing concepts for efficient 5G system design," *IEEE Access*, vol. 3, pp. 195–208, 2015.
[22] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen, and X. He, "A review of compressive sensing in information security field," *IEEE Access*, vol. 4, pp. 2507–2519, 2016.
[23] Q. Shen, W. Liu, W. Cui, and S. Wu, "Underdetermined DOA estimation under the compressive sensing framework: A review," *IEEE Access*, vol. 4, pp. 8865–8878, 2016.
[24] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: Algorithms and applications," *IEEE Access*, vol. 3, no. 1, pp. 490–530, May 2015.
[25] M. Rani, S. Dhok, and R. Deshmukh, "A systematic review of compressive sensing: Concepts, implementations and applications," *IEEE Access*, vol. 6, pp. 4875–4894, 2018.
[26] S.-W. Hu, G.-X. Lin, S.-H. Hsieh, and C.-S. Lu, "Performance analysis of joint-sparse recovery from multiple measurement vectors via convex optimization: Which prior information is better?" *IEEE Access*, vol. 6, pp. 3739–3754, 2018.
[27] H. Palangi, R. Ward, and L. Deng, "Distributed compressive sensing: A deep learning approach," *IEEE Trans. Signal Process.*, vol. 64, no. 17, pp. 4504–4518, Sep. 2016.
[28] J. Ziniel and P. Schniter, "Efficient high-dimensional inference in the multiple measurement vector problem," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 340–354, 2013.
[29] J. Wen, J. Tang, and F. Zhu, "Greedy block coordinate descent under restricted isometry property," *Mobile Netw. Appl.*, vol. 22, no. 3, pp. 371–376, Jun. 2017.
[30] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
[31] Y. Arjoune, N. Kaabouch, H. El Ghazi, and A. Tamtaoui, "Compressive sensing: Performance comparison of sparse recovery algorithms," in *Proc. IEEE 7th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2017, pp. 1–7.
[32] E. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, no. 3, pp. 969–985, 2007.
[33] J. Wen, D. Li, and F. Zhu, "Stable recovery of sparse signals via l_p-minimization," *Appl. Comput. Harmon. Anal.*, vol. 38, no. 1, pp. 161–176, 2015.
[34] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007.
[35] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
[36] Z. Chen and J. J. Dongarra, "Condition numbers of Gaussian random matrices," *SIAM J. Matrix Anal. Appl.*, vol. 27, no. 3, pp. 603–620, Jul. 2005.
[37] W. U. Bajwa, J. D. Haupt, G. M. Raz, S. J. Wright, and R. D. Nowak, "Toeplitz-structured compressed sensing matrices," in *Proc. IEEE/SP 14th Workshop Stat. Signal Process.*, Aug. 2007, pp. 294–298.
[38] A. Amini, V. Montazerhodjat, and F. Marvasti, "Matrices with small coherence using p-ary block codes," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 172–181, Jan. 2012.
[39] R. Calderbank, S. Howard, and S. Jafarpour, "Construction of a large class of deterministic sensing matrices that satisfy a statistical isometry property," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 358–374, Apr. 2010.
[40] S. Li, F. Gao, G. Ge, and S. Zhang, "Deterministic construction of compressed sensing matrices via algebraic curves," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5035–5041, Aug. 2012.
[41] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
[42] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.
[43] M. Elad and A. M. Bruckstein, "A generalized uncertainty principle and sparse representation in pairs of bases," *IEEE Trans. Inf. Theory*, vol. 48, no. 9, pp. 2558–2567, Sep. 2002.
[44] D. L. Donoho and Y. Tsaig, "Fast solution of ℓ_1-norm minimization problems when the solution may be sparse," *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 4789–4812, Nov. 2008.
[45] D. L. Donoho and J. Tanner, "Counting faces of randomly projected polytopes when the projection radically lowers dimension," *J. Amer. Math. Soc.*, vol. 22, no. 1, pp. 1–53, 2009.
[46] G. Satat, M. Tancik, and R. Raskar, "Lensless imaging with compressive ultrafast sensing," *IEEE Trans. Comput. Imag.*, vol. 3, no. 3, pp. 398–407, Sep. 2017.
[47] U. V. Dias and M. E. Rane, "Block based compressive sensed thermal image reconstruction using greedy algorithms," *Int. J. Image, Graph. Signal Process.*, vol. 6, no. 10, pp. 36–42, 2014.
[48] M. F. Duarte *et al.*, "Single-pixel imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 83–91, Mar. 2008.
[49] A. Saucedo, S. Lefkimmiatis, N. Rangwala, and K. Sung, "Improved computational efficiency of locally low rank MRI reconstruction using iterative random patch adjustments," *IEEE Trans. Med. Imag.*, vol. 36, no. 6, pp. 1209–1220, Jun. 2017.
[50] 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 Trans. Biomed. Circuits Syst.*, vol. 10, no. 1, pp. 149–162, Feb. 2016.
[51] S. Vasanawala *et al.*, "Practical parallel imaging compressed sensing MRI: Summary of two years of experience in accelerating body MRI of pediatric patients," in *Proc. IEEE Int. Symp. Biomed. Imag. Nano Macro*, Apr. 2011, pp. 1039–1043.
[52] D. Craven, B. McGinley, L. Kilmartin, M. Glavin, and E. Jones, "Adaptive dictionary reconstruction for compressed sensing of ECG signals," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 3, pp. 645–654, May 2017.
[53] H. Djelouat, X. Zhai, M. A. Disi, A. Amira, and F. Bensaali, "System-on-chip solution for patients biometric: A compressive sensing-based approach," *IEEE Sensors J.*, vol. 18, no. 23, pp. 9629–9639, Dec. 2018.
[54] S. Pudlewski, T. Melodia, and A. Prasanna, "Compressed-sensing-enabled video streaming for wireless multimedia sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 6, pp. 1060–1072, Jun. 2012.
[55] B. K. N. Srinivasarao, V. C. Gogineni, S. Mula, and I. Chakrabarti, "A novel framework for compressed sensing based scalable video coding," *Signal Process., Image Commun.*, vol. 57, pp. 183–196, Sep. 2017.
[56] C. Li, H. Jiang, P. Wilford, Y. Zhang, and M. Scheutzow, "A new compressive video sensing framework for mobile broadcast," *IEEE Trans. Broadcast.*, vol. 59, no. 1, pp. 197–205, Mar. 2013.
[57] T. Goldstein, L. Xu, K. F. Kelly, and R. Baraniuk, "The STOne transform: Multi-resolution image enhancement and compressive video," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5581–5593, Dec. 2015.
[58] R. G. Baraniuk, T. Goldstein, A. C. Sankaranarayanan, C. Studer, A. Veeraraghavan, and M. B. Wakin, "Compressive video sensing: Algorithms, architectures, and applications," *IEEE Signal Process. Mag.*, vol. 34, no. 1, pp. 52–66, Jan. 2017.
[59] M. Herman and T. Strohmer, "Compressed sensing radar," in *Proc. IEEE Radar Conf.*, May 2008, pp. 1–6.
[60] M.-S. Kang, S.-J. Lee, S.-H. Lee, and K.-T. Kim, "ISAR imaging of high-speed maneuvering target using gapped stepped-frequency waveform and compressive sensing," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 5043–5056, Oct. 2017.
[61] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.
[62] M. A. Razzaque, C. Bleakley, and S. Dobson, "Compression in wireless sensor networks: A survey and comparative evaluation," *ACM Trans. Sensor Netw.*, vol. 10, no. 1, p. 5, 2013.
[63] C. Karakus, A. C. Gurbuz, and B. Tavli, "Analysis of energy efficiency of compressive sensing in wireless sensor networks," *IEEE Sensors J.*, vol. 13, no. 5, pp. 1999–2008, May 2013.
[64] M. Hooshmand, M. Rossi, D. Zordan, and M. Zorzi, "Covariogram-based compressive sensing for environmental wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 6, pp. 1716–1729, Mar. 2016.
[65] Z. Li, H. Huang, and S. Misra, "Compressed sensing via dictionary learning and approximate message passing for multimedia Internet of Things," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 505–512, Apr. 2017.

[66] M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "Low-cost security of IoT sensor nodes with rakeness-based compressed sensing: Statistical and known-plaintext attacks," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 2, pp. 327–340, Feb. 2018.

[67] Y. Gargouri, H. Petit, P. Loumeau, B. Cecconi, and P. Desgreys, "Compressed sensing for astrophysical signals," in *Proc. IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Dec. 2016, pp. 313–316.

[68] J. Bobin, J.-L. Starck, and R. Ottensamer, "Compressed sensing in astronomy," *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 5, pp. 718–726, Oct. 2008.

[69] J. Yang, "A machine learning paradigm based on sparse signal representation," M.S. thesis, School Elect., Comput. Telecommun. Eng., Univ. Wollongong, Wollongong, NSW, Australia, 2013. [Online]. Available: https://ro.uow.edu.au/cgi/viewcontent.cgi?referer=https://www.google.com.br/&httpsredir=1&article=4905&context=theses

[70] J. Lu, N. Verma, and N. K. Jha, "Compressed signal processing on Nyquist-sampled signals," *IEEE Trans. Comput.*, vol. 65, no. 11, pp. 3293–3303, Nov. 2016.

[71] B. Sun, H. Feng, K. Chen, and X. Zhu, "A deep learning framework of quantized compressed sensing for wireless neural recording," *IEEE Access*, vol. 4, pp. 5169–5178, 2016.

[72] H. Sun, A. Nallanathan, C.-X. Wang, and Y. Chen, "Wideband spectrum sensing for cognitive radio networks: A survey," *IEEE Wireless Commun.*, vol. 20, no. 2, pp. 74–81, Apr. 2013.

[73] A. Ali and W. Hamouda, "Advances on spectrum sensing for cognitive radio networks: Theory and applications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1277–1304, 2nd Quart., 2016.

[74] E. Panayirci, H. Senol, M. Uysal, and H. V. Poor, "Sparse channel estimation and equalization for OFDM-based underwater cooperative systems with amplify-and-forward relaying," *IEEE Trans. Signal Process.*, vol. 64, no. 1, pp. 214–228, Jan. 2016.

[75] C. Li, K. Song, and L. Yang, "Low computational complexity design over sparse channel estimator in underwater acoustic OFDM communication system," *IET Commun.*, vol. 11, no. 7, pp. 1143–1151, May 2017.

[76] J. Ying, J. Zhong, M. Zhao, and Y. Cai, "Turbo equalization based on compressive sensing channel estimation in wideband HF systems," in *Proc. Int. Conf. Wireless Commun. Signal Process.*, Oct. 2013, pp. 1–5.

[77] E. C. Marques, N. Maciel, L. A. B. Naviner, H. Cai, and J. Yang, "Compressed sensing for wideband HF channel estimation," in *Proc. Int. Conf. Frontiers Signal Process.*, Sep. 2018, pp. 1–5.

[78] W. Schreiber, "Advanced television systems for terrestrial broadcasting: Some problems and some proposed solutions," *Proc. IEEE*, vol. 83, no. 6, pp. 958–981, Jun. 1995.

[79] Z. Fan, Z. Lu, and Y. Han, "Accurate channel estimation based on Bayesian compressive sensing for next-generation wireless broadcasting systems," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast.*, Jun. 2014, pp. 1–5.

[80] P. Zhang, Z. Hu, R. C. Qiu, and B. M. Sadler, "A compressed sensing based ultra-wideband communication system," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2009, pp. 1–5.

[81] K. M. Cohen, C. Attias, B. Farbman, I. Tselniker, and Y. C. Eldar, "Channel estimation in UWB channels using compressed sensing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 1966–1970.

[82] Z. Marzi, D. Ramasamy, and U. Madhow, "Compressive channel estimation and tracking for large arrays in mm-wave picocells," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 514–527, Apr. 2016.

[83] X. Ma, F. Yang, S. Liu, J. Song, and Z. Han, "Design and optimization on training sequence for mmWave communications: A new approach for sparse channel estimation in massive MIMO," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 7, pp. 1486–1497, Jul. 2017.

[84] C. R. Berger, Z. Wang, J. Huang, and S. Zhou, "Application of compressive sensing to sparse channel estimation," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 164–174, Nov. 2010.

[85] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, no. 6, pp. 1058–1076, Jun. 2010.

[86] B. Mansoor, S. J. Nawaz, and S. M. Gulfam, "Massive-MIMO sparse uplink channel estimation using implicit training and compressed sensing," *Appl. Sci.*, vol. 7, no. 1, p. 63, 2017.

[87] R. Ferdian, Y. Hou, and M. Okada, "A low-complexity hardware implementation of compressed sensing-based channel estimation for ISDB-T system," *IEEE Trans. Broadcast.*, vol. 63, no. 1, pp. 92–102, Mar. 2017.

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

[89] X. Chen, Z. Yu, S. Hoyos, B. M. Sadler, and J. Silva-Martinez, "A sub-Nyquist rate sampling receiver exploiting compressive sensing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 3, pp. 507–520, Mar. 2011.

[90] M. Pelissier and C. Studer, "Non-uniform wavelet sampling for RF analog-to-information conversion," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 2, pp. 471–484, Feb. 2018.

[91] W. Guo, Y. Kim, A. H. Tewfik, and N. Sun, "A fully passive compressive sensing SAR ADC for low-power wireless sensors," *IEEE J. Solid-State Circuits*, vol. 52, no. 8, pp. 2154–2167, Aug. 2017.

[92] Y. Zhang, X. Fu, Q. Zhang, Z. Feng, and X. Liu, "Novel schemes to optimize sampling rate for compressed sensing," *J. Commun. Netw.*, vol. 17, no. 5, pp. 517–524, Oct. 2015.

[93] J. N. Laska, S. Kirolos, M. F. Duarte, T. S. Ragheb, R. G. Baraniuk, and Y. Massoud, "Theory and implementation of an analog-to-information converter using random demodulation," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 1959–1962.

[94] T. Ragheb, J. N. Laska, H. Nejati, S. Kirolos, R. G. Baraniuk, and Y. Massoud, "A prototype hardware for random demodulation based compressive analog-to-digital conversion," in *Proc. 51st Midwest Symp. Circuits Syst.*, Aug. 2008, pp. 37–40.

[95] Y. Chen, M. Mishali, Y. C. Eldar, and A. O. Hero, "Modulated wideband converter with non-ideal lowpass filters," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 3630–3633.

[96] D. E. Bellasi, L. Bettini, C. Benkeser, T. Burger, Q. Huang, and C. Studer, "VLSI design of a monolithic compressive-sensing wideband analog-to-information converter," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 3, no. 4, pp. 552–565, Dec. 2013.

[97] Y. Gargouri, H. Petit, P. Loumeau, B. Cecconi, and P. Desgreys, "Analog-to-information converter design for low-power acquisition of astrophysical signals," in *Proc. 15th IEEE Int. New Circuits Syst. Conf. (NEWCAS)*, Jun. 2017, pp. 113–116.

[98] M. Trakimas, R. D'Angelo, S. Aeron, T. Hancock, and S. Sonkusale, "A compressed sensing analog-to-information converter with edge-triggered SAR ADC core," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 5, pp. 1135–1148, May 2013.

[99] J. F. Gemmeke, H. Van Hamme, B. Cranen, and L. Boves, "Compressive sensing for missing data imputation in noise robust speech recognition," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 272–287, Apr. 2010.

[100] M. Gavrilescu, "Improved automatic speech recognition system by using compressed sensing signal reconstruction based on l_0 and l_1 estimation algorithms," in *Proc. 7th Int. Conf. Electron., Comput. Artif. Intell. (ECAI)*, Jun. 2015, pp. S-23–S-28.

[101] A. Latif and W. A. Mousa, "An efficient undersampled high-resolution radon transform for exploration seismic data processing," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 1010–1024, Feb. 2017.

[102] J. Cao, Y. Wang, J. Zhao, and C. Yang, "A review on restoration of seismic wavefields based on regularization and compressive sensing," *Inverse Problems Sci. Eng.*, vol. 19, no. 5, p. 679–704, 2011.

[103] C. Stork and D. Brookes, "The decline of conventional seismic acquisition and the rise of specialized acquisition: This is compressive sensing," in *Proc. Soc. Explor. Geophysicists Annu. Meeting*, 2014, pp. 4386–4392.

[104] M. Hawes, L. Mihaylova, F. Septier, and S. Godsill, "Bayesian compressive sensing approaches for direction of arrival estimation with mutual coupling effects," *IEEE Trans. Antennas Propag.*, vol. 65, no. 3, pp. 1357–1368, Mar. 2017.

[105] Q. Wang, T. Dou, H. Chen, W. Yan, and W. Liu, "Effective block sparse representation algorithm for DOA estimation with unknown mutual coupling," *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2622–2625, Dec. 2017.

[106] L. Zhao, J. Xu, J. Ding, A. Liu, and L. Li, "Direction-of-arrival estimation of multipath signals using independent component analysis and compressive sensing," *PLoS ONE*, vol. 12, no. 7, pp. 1–17, 2017.

[107] D. Malioutov, M. Çetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3010–3022, Aug. 2005.

[108] I. Bilik, T. Northardt, and Y. Abramovich, "Expected likelihood for compressive sensing-based DOA estimation," in *Proc. IET Int. Conf. Radar Syst. (Radar)*, Oct. 2012, pp. 1–4.

[109] A. Y. Carmi, L. S. Mihaylova, and S. J. Godsill, *Compressed Sensing and Sparse Filtering* (Signals and Communication Technology). Berlin, Germany: Springer-Verlag, 2013.
[110] D. Kanevsky, A. Carmi, L. Horesh, P. Gurfil, B. Ramabhadran, and T. N. Sainath, "Kalman filtering for compressed sensing," in *Proc. 13th Int. Conf. Inf. Fusion*, Jul. 2010, pp. 1–8.
[111] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1999.
[112] R. J. Vanderbei, *Linear Programming: Found. Extensions*, 2nd ed. Norwell, MA, USA: Kluwer, 2001.
[113] P. S. Huggins and S. W. Zucker, "Greedy basis pursuit," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3760–3772, Jul. 2007.
[114] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc., B (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.
[115] W. Li, "Estimation and tracking of rapidly time-varying broadband acoustic communication channels," Ph.D. dissertation, Dept. Mech. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 2006.
[116] W. J. Fu, "Penalized regressions: The bridge versus the lasso," *J. Comput. Graph. Statist.*, vol. 7, no. 3, pp. 397–416, 1998.
[117] A. Maleki, L. Anitori, Z. Yang, and R. G. Baraniuk, "Asymptotic analysis of complex LASSO via complex approximate message passing (CAMP)," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4290–4308, Jul. 2013.
[118] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.
[119] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.
[120] M. A. Hameed, "Comparative analysis of orthogonal matching pursuit and least angle regression," M.S. thesis, Master Sci. Elect. Eng., Michigan State Univ., Lansing, MI, USA, 2012. [Online]. Available: https://d.lib.msu.edu/etd/1711/datastream/OBJ/View/
[121] E. Candes and T. Tao, "The dantzig selector: statistical estimation when p is much larger than n," *Ann. Statist.*, vol. 35, no. 6, pp. 2313–2351, 2007.
[122] M. A. Maleki, "Approximate message passing algorithms for compressed sensing," Ph.D. dissertation, Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, 2011.
[123] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 45, pp. 18914–18919, 2009.
[124] L. Bai, P. Maechler, M. Muehlberghuber, and H. Kaeslin, "High-speed compressed sensing reconstruction on FPGA using OMP and AMP," in *Proc. 19th IEEE Int. Conf. Electron., Circuits, Syst. (ICECS)*, Dec. 2012, pp. 53–56.
[125] M. L. Gallo, A. Sebastian, G. Cherubini, H. Giefers, and E. Eleftheriou, "Compressed sensing with approximate message passing using in-memory computing," *IEEE Trans. Electron Devices*, vol. 65, no. 10, pp. 4304–4312, Oct. 2018.
[126] L. Zheng, Z. Wu, M. Seok, X. Wang, and Q. Liu, "High-accuracy compressed sensing decoder based on adaptive (ℓ_0, ℓ_1) complex approximate message passing: Cross-layer design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 10, pp. 1726–1736, Oct. 2016.
[127] R. Garg and R. Khandekar, "Gradient descent with sparsification: An iterative algorithm for sparse recovery with restricted isometry property," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 337–344.
[128] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.
[129] I. Daubechies, M. Fornasier, and I. Loris, "Accelerated projected gradient method for linear inverse problems with sparsity constraints," *J. Fourier Anal. Appl.*, vol. 14, pp. 764–792, Dec. 2008.
[130] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 693–696.
[131] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.
[132] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.
[133] M. A. T. Figueiredo, "Adaptive sparseness using Jeffreys prior," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst., Natural Synth. (NIPS)*. Cambridge, MA, USA: MIT Press, 2001, pp. 697–704.
[134] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*. New York, NY, USA: Wiley, 1994.
[135] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Sep. 2001.
[136] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 600–616, Mar. 1997.
[137] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2008, pp. 3869–3872.
[138] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
[139] P. K. Meher, B. K. Mohanty, and T. Srikanthan, "Area-delay efficient architecture for MP algorithm using reconfigurable inner-product circuits," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2014, pp. 2628–2631.
[140] N. M. de Paiva, Jr., E. C. Marques, and L. A. de Barros Naviner, "Sparsity analysis using a mixed approach with greedy and LS algorithms on channel estimation," in *Proc. 3rd Int. Conf. Frontiers Signal Process. (ICFSP)*, Sep. 2017, pp. 91–95.
[141] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conf. Signals, Syst. Comput.*, vol. 1, Nov. 1993, pp. 40–44.
[142] J. Wen, Z. Zhou, J. Wang, X. Tang, and Q. Mo, "A sharp condition for exact support recovery of sparse signals with orthogonal matching pursuit," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 2364–2368.
[143] J. Wen, Z. Zhou, J. Wang, X. Tang, and Q. Mo, "A sharp condition for exact support recovery with orthogonal matching pursuit," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1370–1382, Mar. 2017.
[144] J. Wang, "Support recovery with orthogonal matching pursuit in the presence of noise," *IEEE Trans. Signal Process.*, vol. 63, no. 21, pp. 5868–5877, Nov. 2015.
[145] A. Kulkarni and T. Mohsenin, "Low overhead architectures for OMP compressive sensing reconstruction algorithm," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 6, pp. 1468–1480, Jun. 2017.
[146] H. Rabah, A. Amira, B. K. Mohanty, S. Almaadeed, and P. K. Meher, "FPGA implementation of orthogonal matching pursuit for compressive sensing reconstruction," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 10, pp. 2209–2220, Oct. 2015.
[147] B. Knoop, J. Rust, S. Schmale, D. Peters-Drolshagen, and S. Paul, "Rapid digital architecture design of orthogonal matching pursuit," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*, Aug./Sep. 2016, pp. 1857–1861.
[148] C. Morales-Perez, J. Rangel-Magdaleno, H. Peregrina-Barreto, J. Ramirez-Cortes, and I. Cruz-Vega, "FPGA-based broken bar detection on IM using OMP algorithm," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2017, pp. 1–6.
[149] S. Liu, N. Lyu, and H. Wang, "The implementation of the improved OMP for AIC reconstruction based on parallel index selection," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 2, pp. 319–328, Feb. 2018.
[150] Ö. Polat and S. K. Kayhan, "High-speed FPGA implementation of orthogonal matching pursuit for compressive sensing signal reconstruction," *Comput. Elect. Eng.*, vol. 71, pp. 173–190, Oct. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790617337904
[151] Z. Yu *et al.*, "Fast compressive sensing reconstruction algorithm on FPGA using orthogonal matching pursuit," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 249–252.
[152] J. L. V. M. Stanislaus and T. Mohsenin, "High performance compressive sensing reconstruction hardware with QRD process," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2012, pp. 29–32.
[153] S. Kwon, J. Wang, and B. Shim, "Multipath matching pursuit," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2986–3001, May 2014.
[154] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3042–3054, Jun. 2010.
[155] J. Wen, H. Chen, and Z. Zhou, "An optimal condition for the block orthogonal matching pursuit algorithm," *IEEE Access*, vol. 6, pp. 38179–38185, 2018.

[156] J. Wen, Z. Zhou, Z. Liu, M.-J. Lai, and X. Tang, "Sharp sufficient conditions for stable recovery of block sparse signals by block orthogonal matching pursuit," *Appl. Comput. Harmon. Anal.*, 2018, doi: 10.1016/j.acha.2018.02.002.

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

[158] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 1094–1121, Feb. 2012.

[159] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 2008-01, 2008.

[160] J. Lu, H. Zhang, and H. Meng, "Novel hardware architecture of sparse recovery based on FPGAs," in *Proc. 2nd Int. Conf. Signal Process. Syst.*, vol. 1, Jul. 2010, pp. V1-302–V1-306.

[161] H. D. R. Aniles, "FPGA-based compressed sensing reconstruction of sparse signals," M.S. thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, Cholula, Mexico, 2014.

[162] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Found. Comput. Math.*, vol. 9, no. 3, pp. 317–334, Jun. 2009.

[163] J. Wang, S. Kwon, and B. Shim, "Generalized orthogonal matching pursuit," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6202–6216, Dec. 2012.

[164] J. Wen, Z. Zhou, D. Li, and X. Tang, "A novel sufficient condition for generalized orthogonal matching pursuit," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 805–808, Apr. 2017.

[165] H. Sun and L. Ni, "Compressed sensing data reconstruction using adaptive generalized orthogonal matching pursuit algorithm," in *Proc. 3rd Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*, Oct. 2013, pp. 1102–1106.

[166] T. Blumensath and M. E. Davies, "Gradient pursuits," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2370–2382, Jun. 2008.

[167] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MA, USA: Johns Hopkins Univ. Press, 1996.

[168] A. C. Gilbert, S. Muthukrishnan, and M. Strauss, "Improved time bounds for near-optimal sparse Fourier representations," *Proc. SPIE, Wavelets XI*, vol. 5914, p. 59141A, Sep. 2005, doi: 10.1117/12.615931.

[169] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *J. Fourier Anal. Appl.*, vol. 14, nos. 5–6, pp. 629–654, Dec. 2008.

[170] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 265–274, Nov. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1063520309000384

[171] H. Huang and A. Makur, "Backtracking-based matching pursuit method for sparse signal reconstruction," *IEEE Signal Process. Lett.*, vol. 18, no. 7, pp. 391–394, Jul. 2011.

[172] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin, "Algorithmic linear dimension reduction in the ℓ_1 norm for sparse vectors," in *Proc. 44th Annu. Allerton Conf. Commun., Control, Comput. (ALLERTON)*, 2006, pp. 1411–1418.

[173] J. D. Blanchard, J. Tanner, and K. Wei, "CGIHT: Conjugate gradient iterative hard thresholding for compressed sensing and matrix completion," *Inf. Inference, J. IMA*, vol. 4, no. 4, pp. 289–327, 2015.

[174] X. Zhu, L. Dai, W. Dai, Z. Wang, and M. Moonen, "Tracking a dynamic sparse channel via differential orthogonal matching pursuit," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Oct. 2015, pp. 792–797.

[175] N. B. Karahanoglu and H. Erdogan, "Compressed sensing signal recovery via forward–backward pursuit," *Digital Signal Process.*, vol. 23, no. 5, pp. 1539–1548, 2013.

[176] A. C. Gilbert, S. Muthukrishnan, and M. Strauss, "Improved time bounds for near-optimal sparse Fourier representations," *Proc. SPIE*, vol. 5914, p. 59141A, Sep. 2005.

[177] S. Foucart, "Hard thresholding pursuit: An algorithm for compressive sensing," *SIAM J. Numer. Anal.*, vol. 49, no. 6, pp. 2543–2563, 2011.

[178] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin, "One sketch for all: Fast algorithms for compressed sensing," in *Proc. 39th ACM Symp. Theory Comput.*, 2007, pp. 237–246.

[179] J. Tanner and K. Wei, "Normalized iterative hard thresholding for matrix completion," *SIAM J. Sci. Comput.*, vol. 35, no. 5, pp. S104–S125, 2013.

[180] J. K. Pant and S. Krishnan, "Two-pass ℓ_p-regularized least-squares algorithm for compressive sensing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.

[181] G. Mileounis, B. Babadi, N. Kalouptsidis, and V. Tarokh, "An adaptive greedy algorithm with application to nonlinear communications," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 2998–3007, Jun. 2010.

[182] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2479–2493, Jul. 2009.

[183] E. Vlachos, A. S. Lalos, and K. Berberidis, "Stochastic gradient pursuit for adaptive equalization of sparse multipath channels," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 2, no. 3, pp. 413–423, Sep. 2012.

[184] B. A. Olshausen and K. J. Millman, "Learning sparse codes with a mixture-of-Gaussians prior," in *Advances in Neural Information Processing Systems 12*. Cambridge, MA, USA: MIT Press, 2000.

[185] J. Lee, J. W. Choi, and B. Shim, "Sparse signal recovery via tree search matching pursuit," *J. Commun. Netw.*, vol. 18, no. 5, pp. 699–712, Oct. 2016.

[186] S. Rangan, P. Schniter, and A. K. Fletcher, "Vector approximate message passing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 1588–1592, doi: 10.1109/ISIT.2017.8006797.

[187] C.-B. Song, S.-T. Xia, and X.-J. Liu, "Improved analyses for SP and CoSaMP algorithms in terms of restricted isometry constants," *CoRR*, vol. abs/1309.6073, Sep. 2013, pp. 1–9.

[188] A. K. Mishra and R. S. Verster, *Compressive Sensing Based Algorithms for Electronic Defence* (Signals and Communication Technology). Springer, 2017.

[189] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[190] S. Foucart, "Sparse recovery algorithms: Sufficient conditions in terms of restricted isometry constants," in *Approximation Theory XIII: San Antonio*, M. Neamtu and L. Schumaker, Eds. New York, NY, USA: Springer, 2012, pp. 65–77.

[191] S. K. Sharma, S. Chatzinotas, and B. Ottersten, "Compressive sparsity order estimation for wideband cognitive radio receiver," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 1361–1366.

[192] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel, "Distributed basis pursuit," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1942–1956, Apr. 2012.

[193] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep./Oct. 2015, pp. 1336–1343.

[194] M. Borgerding, P. Schniter, and S. Rangan, "AMP-inspired deep networks for sparse linear inverse problems," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4293–4308, Aug. 2017.

[195] Z. Wang, Q. Ling, and T. Huang, "Learning deep ℓ_0 encoders," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2194–2200.

[196] M. Borgerding and P. Schniter, "Onsager-corrected deep learning for sparse linear inverse problems," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2016, pp. 227–231.

[197] U. S. Kamilov and H. Mansour, "Learning optimal nonlinearities for iterative thresholding algorithms," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 747–751, May 2016.

ELAINE CRESPO MARQUES received the B.Sc. degree from the Military Institute of Engineering, Rio de Janeiro, Brazil, in 2011. She is currently pursuing the Ph.D. degree with Télécom ParisTech, Paris, France.

From 2012 to 2016, she was a R&D Engineer at the Brazilian Defense SDR Project, Rio de Janeiro, Brazil. Her research interests include sparse channel estimation, software-defined radio, cognitive radio, and compressive sensing.

NILSON MACIEL received the B.Sc. degree from the Military Institute of Engineering, Rio de Janeiro, Brazil, in 2011. He is currently pursuing the Ph.D. degree with Télécom ParisTech, Paris, France.

From 2012 to 2016, he was a R&D Engineer with the Brazilian Defense SDR Project, Rio de Janeiro, Brazil. His research interests include physical layer of wireless communication systems, stochastic modeling applied to communications systems, compressive sensing algorithms, and hardware implementation.

LÍRIDA NAVINER (M'98–SM'06) received the B.Sc. and M.Sc. degrees from the Federal University of Paraiba (UFPB), Paraíba, Brazil, in 1988 and 1990, respectively, and the Ph.D. degree from the Ecole Nationale Supérieure des Télécommunications, Paris, France, in 1994. She received the Habilitation à Diriger des Recherches from Pierre and Marie Curie University, Paris, in 2006.

From 1994 to 1997, she was a Faculty Member with UFPB. Since 1998, she has been with the Department of Electronics and Communications, Télécom ParisTech, Paris, France, where she is currently a Full Professor. She has supervised over 20 Ph.D. theses in addition to many M.Sc. projects. She is the inventor/co-inventor of eight patents and has authored/co-authored over 200 scientific papers. Her research interests are currently focused on deep-submicron CMOS technologies, design for reliability, and hybrid nonvolatile devices/CMOS circuits.

Dr. Naviner has served as a Technical Program Committee Member for numerous international conferences, and is regularly invited to review papers for scientific journals and projects proposals for funding agencies. She has led several academic and industrial research projects with national/international partners for Télécom ParisTech. She is also a Member of several think tanks, including the start-up incubator of ParisTech. She has been honored with the title Chevalier dans l'Ordre des Palmes Académiques by the French government.

HAO CAI (M'15) received the Ph.D. degree in electrical engineering from Télécom Paristech, Université Paris-Saclay, France, in 2013.

From 2012 to 2014, he was involved in the European EUREKA Programme CATRENE-RELY for high reliability nanoscale integrated circuits and systems.

He is currently an Associate Professor with the National ASIC System Engineering Center, Southeast University, Nanjing, China. His research interests include circuit techniques for emerging technologies, ultra-low-power VLSI, and reliability-aware design.

JUN YANG (M'14) received the B.Sc., M.Sc., and Ph.D. degrees from Southeast University, in 1999, 2001, and 2004, respectively. He is currently a Professor with the National ASIC Center, Southeast University, Nanjing, China.

His research interests include near threshold circuit design and ultra-low power indoor/outdoor position algorithm and chips. He is now close collaborated with SMIC and TSMC to develop near threshold IP, such as embedded memories, standard cell library, and other RF components. He owned three US patents, one EU patent, over 30 Chinese patents, and has co-authored over 30 academic papers. He has supervised and graduated over 30 master's and Ph.D. students. He is a recipient of several national awards, including the State Science and Technology Awards.