Review Article

Compressive Sensing in Signal Processing: Algorithms and Transform Domain Formulations

Irena Orović,[1] Vladan Papić,[2] Cornel Ioana,[3] Xiumei Li,[4] and Srdjan Stanković[1]

[1]*Faculty of Electrical Engineering, University of Montenegro, Džordža Vašingtona bb, 81000 Podgorica, Montenegro*
[2]*Faculty of Electrical Engineering, Mechanical Engineering & Naval Architecture, University of Split, Split, Croatia*
[3]*Grenoble Institute of Technology, GIPSA-Lab, Saint-Martin-d'Hères, France*
[4]*School of Information Science and Engineering, Hangzhou Normal University, Zhejiang, China*

Correspondence should be addressed to Irena Orović; irenao@ac.me

Compressive sensing has emerged as an area that opens new perspectives in signal acquisition and processing. It appears as an alternative to the traditional sampling theory, endeavoring to reduce the required number of samples for successful signal reconstruction. In practice, compressive sensing aims to provide saving in sensing resources, transmission, and storage capacities and to facilitate signal processing in the circumstances when certain data are unavailable. To that end, compressive sensing relies on the mathematical algorithms solving the problem of data reconstruction from a greatly reduced number of measurements by exploring the properties of sparsity and incoherence. Therefore, this concept includes the optimization procedures aiming to provide the sparsest solution in a suitable representation domain. This work, therefore, offers a survey of the compressive sensing idea and prerequisites, together with the commonly used reconstruction methods. Moreover, the compressive sensing problem formulation is considered in signal processing applications assuming some of the commonly used transformation domains, namely, the Fourier transform domain, the polynomial Fourier transform domain, Hermite transform domain, and combined time-frequency domain.

1. Introduction

The fundamental approach for signal reconstruction from its measurements is defined by the Shannon-Nyquist sampling theorem stating that the sampling rate needs to be at least twice the maximal signal frequency. In the discrete case, the number of measurements should be at least equal to the signal length in order to be exactly reconstructed. However, this approach may require large storage space, significant sensing time, heavy power consumption, and large number of sensors. Compressive sensing (CS) is a novel theory that goes beyond the traditional approach [1–4]. It shows that a sparse signal can be reconstructed from much fewer incoherent measurements. The basic assumption in CS approach is that most of the signals in real applications have a concise representation in a certain transform domain where only few of them are significant, while the rest are zero or negligible [5–7]. This requirement is defined as signal sparsity.

Another important requirement is the incoherent nature of measurements (observations) in the signal acquisition domain. Therefore, the main objective of CS is to provide an estimate of the original signal from a small number of linear incoherent measurements by exploiting the sparsity property [3, 4].

The CS theory covers not only the signal acquisition strategy, but also the signal reconstruction possibilities and different algorithms [8–17]. Several approaches for CS signal reconstruction have been developed and most of them belong to one of three main approaches: convex optimizations [8–11] such as basis pursuit, Dantzig selector, and gradient-based algorithms; greedy algorithms like matching pursuit [14] and orthogonal matching pursuit [15]; and hybrid methods such as compressive sampling matching pursuit [16] and stage-wise OMP [17]. When comparing these algorithms, convex programming provides the best reconstruction accuracy, but at the cost of high computational complexity. The greedy

algorithms bring about low computation complexity, while the hybrid methods try to provide a compromise between these two requirements [18].

The proposed work provides a survey of the general compressive sensing concept supplemented with the several existing approaches and methods for signal reconstruction, which are briefly explained and summarized in the form of algorithms with the aim of providing the readers with an easier and practical insight into the state of the art in this field. Apart from the standard CS algorithms, a few recent solutions have been included as well. Furthermore, the paper provides an overview of different sparsity domains and the possibilities of employing them in the CS problem formulation. Additional contribution is provided through the examples showing the efficiency of the presented methods in practical applications.

The paper is organized as follows. In Section 2, a brief review of the general compressive sensing idea is provided together with the conditions for successful signal reconstruction from reduced set of measurements and the signal recovery formulations using minimization approaches. In Section 3, the commonly used CS algorithms are reviewed. The commonly used domains for CS strategy implementation are given in Section 4, while some of the examples in real applications are provided in Section 5. The concluding remarks are given in Section 6.

2. Compressive Sensing: A General Overview

2.1. Sparsity and Compressibility. Reducing the sampling rate using CS is possible for the case of sparse signals that can be represented by a small number of significant coefficients in an appropriate transform basis. A signal having K nonzero coefficients is called K-sparse. Assume that signal s exhibits sparsity in certain orthonormal basis $\boldsymbol{\Psi}$ defined by the basis vectors $\{\psi_1, \psi_2, \ldots, \psi_N\}$. The signal s can be represented using its sparse transform domain vector $\mathbf{x}$ as follows:

$$s(n) = \sum_{i=1}^{N} x_i \psi_i(n). \tag{1}$$

In matrix notation, the previous relation can be written as

$$\mathbf{s} = \boldsymbol{\Psi}\mathbf{x}. \tag{2}$$

Commonly, the sparsity is measured using the ℓ_0-norm, which represents the cardinality of the support of $\mathbf{x}$:

$$\|\mathbf{x}\|_0 = \text{card}\{\text{supp}(\mathbf{x})\} = K. \tag{3}$$

In real applications, the signals are usually not strictly sparse but only approximately sparse. Therefore, instead of being sparse, these signals are often called compressible, meaning that the amplitudes of coefficients decrease rapidly when arranged in descending order. For instance, if we consider coefficients $|x_1| \geq |x_2| \geq \cdots \geq |x_n|$, then the magnitude decays with a power law if there exist constants C_1 and $q > 0$ satisfying [19]

$$|x_i| \leq C_1 i^{-q}, \tag{4}$$

where larger q means faster decay and consequently more compressible signal. The signal compressibility can be quantified using the minimal error between the original and sparsified signal (obtained by keeping only K largest coefficients):

$$\sigma_K(\mathbf{x})_p = \min \|\mathbf{x} - \mathbf{x}_K\|_p. \tag{5}$$

2.2. Conditions on the CS Matrix: Null Space Property, Restricted Isometry Property, and Incoherence. Instead of acquiring a full set of signal samples of length N, in CS scenario, we deal with a quite reduced set of measurements $\mathbf{y}$ of length M, where $M < N$. The measurement procedure can be modeled by projections of the signal $\mathbf{s}$ onto vectors $\{\phi_1, \phi_2, \ldots, \phi_M\}$ constituting the measurement matrix $\boldsymbol{\Phi}$:

$$\mathbf{y} = \boldsymbol{\Phi}\mathbf{s}. \tag{6}$$

Using the sparse transform domain representation of vector $\mathbf{s}$ given by (2), we have

$$\mathbf{y} = \boldsymbol{\Phi}\boldsymbol{\Psi}\mathbf{x} = \mathbf{A}\mathbf{x}, \tag{7}$$

where $\mathbf{A}$ will be referred to as CS matrix.

In order to define some requirements for the CS matrix $\mathbf{A}$, which are important for successful signal reconstruction, let us introduce the null space of matrix. The null space of CS matrix $\mathbf{A}$ contains all vectors $\mathbf{x}$ that are mapped to 0:

$$\mathbb{N}(\mathbf{A}) = \{\mathbf{x} : \mathbf{A}\mathbf{x} = 0\}. \tag{8}$$

In order to provide a unique solution, it is necessary to provide the notion that two K-sparse vectors $\mathbf{x}$ and $\mathbf{x}'$ do not result in the same measurement vector. In other words, their difference should not be part of the null space of CS matrix $\mathbf{A}$:

$$\mathbf{A}\left(\mathbf{x} - \mathbf{x}'\right) \neq 0. \tag{9}$$

Since the difference between two K-sparse vectors is at most $2K$-sparse, then a K-sparse vector $\mathbf{x}$ is uniquely defined if null space of $\mathbf{A}$ contains no $2K$-sparse vectors. This corresponds to the condition that any $2K$ columns of $\mathbf{A}$ are linearly independent; that is,

$$\text{spark}(\mathbf{A}) > 2K, \tag{10}$$

and since spark($\mathbf{A}$) $\in [2, M+1]$, we obtain a lower bound on the number of measurements:

$$M > 2K. \tag{11}$$

In the case of strictly sparse signals, the spark can provide reliable information about the exact reconstruction possibility. However, in the case of approximately sparse signals, this condition is not sufficient and does not guarantee stable recovery. Hence, there is another property called null space property that measures the concentration of the null space of matrix $\mathbf{A}$. The null space property is satisfied if there is a constant $\gamma \in (0, 1)$ such that [19]

$$\|\mathbf{a}_S\|_1 = \gamma \|\mathbf{a}_{S^c}\|_1 \quad \forall \mathbf{a} \in \mathbb{N}(\mathbf{A}), \tag{12}$$

for all sets $S \subset \{1, \ldots, N\}$ with cardinality K and their complements $S_c = \{1, \ldots, N\} \setminus S$. If the null space property is satisfied, then a strictly K-sparse signal can be perfectly reconstructed by using ℓ_1-minimization. For approximately K-sparse signals, an upper bound of the ℓ_1-minimization error can be defined as follows [20]:

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_1 \leq 2 \frac{1 + \gamma}{1 - \gamma} \sigma_K(\mathbf{x})_1, \tag{13}$$

where $\sigma_K(\mathbf{x})_1$ is defined in (5) for $p = 1$ as the minimal error induced by the best K-sparse approximation.

The null space property is necessary and sufficient for establishing guarantees for recovery. A stronger condition is required in the presence of noise (and approximately sparse signals). Hence, in [8], the restricted isometry property (RIP) of CS matrix $\mathbf{A}$ has been introduced. The CS matrix $\mathbf{A}$ satisfies the RIP property with constant $\delta_K \in (0, 1)$ if

$$(1 - \delta_K) \|\mathbf{x}\|_2^2 \leq \|\mathbf{Ax}\|_2^2 \leq (1 + \delta_K) \|\mathbf{x}\|_2^2, \tag{14}$$

for every K-sparse vector $\mathbf{x}$. This property shows how well the distances are preserved by a certain linear transformation. We might now say that if the RIP is satisfied for $2K$ with $\delta_{2K} < 1$, then there are no two K-sparse vectors $\mathbf{x}$ that can correspond to the same measurement vector $\mathbf{y}$.

Finally, the incoherence condition mentioned before, which is also related to the RIP of matrix $\mathbf{A}$, refers to the incoherence of the projection basis $\mathbf{\Phi}$ and the sparsifying basis $\mathbf{\Psi}$. The mutual coherence can be simply defined by using the combined CS matrix $\mathbf{A}$ as follows [21]:

$$\mu(\mathbf{A}) = \max_{i \neq j, 1 \leq i, j \leq N} \left| \frac{\langle A_i, A_j \rangle}{\|A_i\|^2 \|A_j\|^2} \right|. \tag{15}$$

The mutual coherence is related to the restricted isometry constant using the following bound [22]:

$$\delta_K = (K - 1)\mu. \tag{16}$$

2.3. Signal Recovery Using Minimization Approach.

The signal recovery problem is defined as the reconstruction of vector $\mathbf{x}$ from the measurements $\mathbf{y} = \mathbf{Ax}$. This problem can be generally seen as a problem of solving an underdetermined set of linear equations. However, in the circumstances when $\mathbf{x}$ is sparse, the problem can be reduced to the following minimization:

$$\begin{aligned} \min \quad & \|\mathbf{x}\|_0 \\ \text{s.t.} \quad & \mathbf{y} = \mathbf{Ax}. \end{aligned} \tag{17}$$

The ℓ_0-minimization requires an exhaustive search over all $\binom{N}{K}$ possible sparse combinations, which is computationally intractable. Hence, the ℓ_0-minimization is replaced by convex ℓ_1-minimization, which will provide the sparse result with high probability if the measurement matrix satisfies the previous conditions. The ℓ_1-minimization problem is defined as follows:

$$\begin{aligned} \min \quad & \|\mathbf{x}\|_1 \\ \text{s.t.} \quad & \mathbf{y} = \mathbf{Ax}, \end{aligned} \tag{18}$$

and it has been known as the basis pursuit.

In the situation when the measurements are corrupted by the noise of level $\mathbf{e}$: $\mathbf{y} = \mathbf{\Phi\Psi x} + \mathbf{e} = \mathbf{Ax} + \mathbf{e}$ and $\|\mathbf{e}\|_2 \leq \varepsilon$, the reconstruction problem can be defined in a form:

$$\begin{aligned} \min \quad & \|\mathbf{x}\|_1 \\ \text{s.t.} \quad & \|\mathbf{y} - \mathbf{Ax}\|_2 \leq \varepsilon, \end{aligned} \tag{19}$$

called basis pursuit denoising. The error bound for the solution of (19), where $\mathbf{A}$ satisfies the RIP of order $2K$ with $\delta_{2K} < \sqrt{2} - 1$ and $\mathbf{y} = \mathbf{Ax} + \mathbf{e}$, is given by

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2 \leq C_0 \sigma_K(\mathbf{x})_1 + C_1 2\varepsilon \sqrt{1 + 2\delta_{2K}}, \tag{20}$$

where the constants C_0 and C_1 are defined as [19]

$$\begin{aligned} C_0 &= 2 \frac{1 - (1 - \sqrt{2})\delta_{2K}}{1 - (1 + \sqrt{2})\delta_{2K}}, \\ C_1 &= 2 \frac{1}{1 - (1 + \sqrt{2})\delta_{2K}}. \end{aligned} \tag{21}$$

For a particular regularization parameter $\lambda > 0$, the minimization problem (19) can be defined using the unconstrained version as follows:

$$\min_x \quad \frac{1}{2} \|\mathbf{y} - \mathbf{Ax}\|_2^2 + \lambda \|\mathbf{x}\|_1, \tag{22}$$

which is known as the Lagrangian form of the basis pursuit denoising. These algorithms are commonly solved using primal-dual interior-point methods [22].

Another form of basis pursuit denoising is solved using the least absolute shrinkage and selection operator (LASSO), and it is defined as follows:

$$\begin{aligned} \min_x \quad & \frac{1}{2} \|\mathbf{y} - \mathbf{Ax}\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{x}\|_1 < \tau, \end{aligned} \tag{23}$$

where τ is a nonnegative real parameter. The convex optimization methods usually require high computational complexity and high numerical precision.

When the noise is unbounded, one may apply the convex program based on Dantzig selector (it is assumed that the noise variance is σ^2 per measurement, i.e., the total variance is $M\sigma^2$):

$$\begin{aligned} \min \quad & \|\mathbf{x}\|_1 \\ \text{s.t.} \quad & \|\mathbf{A}^*(\mathbf{y} - \mathbf{Ax})\|_\infty \leq \sqrt{2 \log N}\sigma, \end{aligned} \tag{24}$$

which (for enough measurements) reconstructs a signal with the error bound:

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2 \leq \sigma \sqrt{2 \log(N) K}. \tag{25}$$

The norm ℓ_∞ is infinite norm called also supreme (maximum) norm.

Input:
‣ Measurement matrix $\mathbf{A}$ (of size $M \times N$)
‣ Measurement vector $\mathbf{y}$ (of length M)
Output:
‣ A signal estimate $\hat{\mathbf{x}}$ (of length N)
(1) $\hat{x} = 0, r_0 \leftarrow \mathbf{y}, i = 0$
(2) **while** halting criterion false **do**
(3) $i = i + 1$
(4) $\omega_i \leftarrow \arg\max_{j=1,\dots,N} |\langle \mathbf{r}_{i-1}, A_j \rangle|$ (*max correlation column*)
(5) $\hat{\mathbf{x}}_i \leftarrow \langle \mathbf{r}_{i-1}, \omega_i \rangle$ (*new signal estimate*)
(6) $\hat{\mathbf{r}}_i \leftarrow \mathbf{r}_{i-1} - \omega_i \hat{\mathbf{x}}_i$ (*update residual*)
(7) **end while**
(8) **return** $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}}_i$

ALGORITHM 1: Matching pursuit.

Besides the ℓ_1-norm minimization, there exist some approaches using the ℓ_p-norm minimization, with $0 < p < 1$:

$$\begin{aligned} \min \quad & \|\mathbf{x}\|_p \\ \text{s.t.} \quad & \|\mathbf{y} - \mathbf{Ax}\|_2 \le \varepsilon, \end{aligned} \tag{26}$$

or using ℓ_2-norm minimization, in which case the solution is not rigorously sparse enough [23]:

$$\begin{aligned} \min \quad & \|\mathbf{x}\|_2 \\ \text{s.t.} \quad & \|\mathbf{y} - \mathbf{Ax}\|_2 \le \varepsilon. \end{aligned} \tag{27}$$

3. Review of Some Signal Reconstruction Algorithms

The ℓ_1-minimization problems in CS signal reconstruction are usually solved using the convex optimization methods. In addition, there exist greedy methods for sparse signal recovery which allow faster computation compared to ℓ_1-minimization. Greedy algorithms can be divided into two major groups: greedy pursuit methods and thresholding-based methods. In practical applications, the widely used ones are the orthogonal matching pursuit (OMP) and compressive sampling matching pursuit (CoSaMP) from the group of greedy pursuit methods, while from the thresholding group the iterative hard thresholding (IHT) is commonly used due to its simplicity, although it may not be always efficient in providing an exact solution. Some of these algorithms are discussed in detail in this section.

3.1. Matching Pursuit. The matching pursuit algorithm has been known for its simplicity and was first introduced in [14]. This is the first algorithm from the class of iterative greedy methods that decomposes a signal into a linear set of basis functions. Through the iterations, this algorithm chooses in a greedy manner the basis functions that best match the signal. Also, in each iteration, the algorithm removes the signal component having the form of the selected basis function and obtains the residual. This procedure is repeated until the

norm of the residual becomes lower than a certain predefined threshold value (halting criterion) (Algorithm 1).

The matching pursuit algorithm however has a slow convergence property and generally does not provide efficiently sparse results.

3.2. Orthogonal Matching Pursuit. The orthogonal matching pursuit (OMP) has been introduced [15] as an improved version to overcome the limitations of the matching pursuit algorithm. OMP is based on principle of orthogonalization. It computes the inner product of the residue and the measurement matrix and then selects the index of the maximum correlation column and extracts this column (in each iteration). The extracted columns are included into the selected set of atoms. Then, the OMP performs orthogonal projection over the subspace of previously selected atoms, providing a new approximation vector used to update the residual. Here, the residual is always orthogonal to the columns of the CS matrix, so there will be no columns selected twice and the set of selected columns is increased through the iterations. OMP provides better asymptotic convergence compared to the previous matching pursuit version (Algorithm 2).

3.3. CoSaMP. Compressive sampling matching pursuit (CoSaMP) is an extended version of the OMP algorithm [16].

In each iteration, the residual vector $\mathbf{r}$ is correlated with the columns of the CS matrix $\mathbf{A}$, forming a "proxy signal" $\mathbf{z}$. Then, the algorithm selects $2K$ columns of $\mathbf{A}$ corresponding to the $2K$ largest absolute values of $\mathbf{z}$, where K defines the signal sparsity (number of nonzero components). Namely, all but the largest $2K$ elements of $\mathbf{z}$ are set to zero and the $2K$ support Ω is obtained by finding the positions of nonzero elements. The indices of the selected columns ($2K$ in total) are then added to the current estimate of the support of the unknown vector. After solving the least squares, a $3K$-sparse estimate of the unknown vector is obtained. Then, the $3K$-sparse vector is pruned to obtain the K-sparse estimate of the unknown vector (by setting all but the K largest elements to zero). Thus, the estimate of the unknown vector remains K-sparse and all columns that do not correspond

```
Input:
    ▶ Transform matrix Ψ, Measurement matrix Φ
    ▶ CS matrix A: A = ΦΨ
    ▶ Measurement vector y
Output:
    ▶ A signal estimate x̂ ← x_K
    ▶ An approximation to the measurements y by a_K
    ▶ A residual r_K = y − a_K.
    ▶ A set Ω_K with positions of non-zero elements of x̂.
(1) r_0 ← y, Ω_0 ← ∅, Θ_0 ← []
(2) for i = 1, . . . , K
(3) ω_i ← arg max_{j=1,...,N}|⟨r_{i−1}, A_j⟩|        (max correlation column)
(4) Ω_i ← Ω_{i−1} ∪ ω_i                              (Update set of indices)
(5) Θ_i ← [Θ_{i−1}  A_{ω_i}]                         (Update set of atoms)
(6) x_i = arg min_x‖r_{i−1} − Θ_i x‖²_2
(7) a_i ← Θ_i x_i                                    (new approximation)
(8) r_i ← y − a_i                                    (update residual)
(9) end for
(10) return x_K, a_K, r_K, Ω_K
```

ALGORITHM 2: Orthogonal matching pursuit.

to the true signal components are removed, which is an improvement over the OMP. Namely, if OMP selects in some iteration a column that does not correspond to the true signal component, the index of this column will remain in the final signal support and cannot be removed (Algorithm 3).

The CoSaMP can be observed through five crucial steps:

(i) *Identification (Line 5).* It finds the largest 2s components of the signal proxy.

(ii) *Support Merge (Line 6).* It merges the support of the signal proxy with the support of the solution from the previous iteration.

(iii) *Estimation (Line 7).* It estimates a solution via least squares where the solution lies within a support T.

(iv) *Pruning (Line 8).* It takes the solution estimate and compresses it to the required support.

(v) *Sample Update (Line 9).* It updates the "sample," meaning the residual as the part of signal that has not been approximated.

3.4. Iterative Hard Thresholding Algorithm. Another group of algorithms for signal reconstruction from a small set of measurements is based on iterative thresholding. Generally, these algorithms are composed of two main steps. The first one is the optimization of the least squares term, which is done by solving the optimization problem without ℓ_1-minimization. The other one is the decreasing of the ℓ_1-norm, which is done by applying the thresholding operator to the magnitude of entries in **x**. In each iteration, the sparse vector $\mathbf{x}_i$ is estimated by the previous version $\mathbf{x}_{i-1}$ using negative gradient of the objective function defined as

$$J(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2, \quad (28)$$

while the negative gradient is then

$$-\nabla J(\mathbf{x}) = \mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}). \quad (29)$$

Generally, the obtained estimate $\mathbf{x}_i$ is sparse, and we need to set all but the K largest components to zero using the thresholding operator. Here, we distinguish two types of thresholding. Hard thresholding [24, 25] sets all but the K largest magnitude values to zero, where the thresholding operator can be written as

$$H_K(\mathbf{x}) = \begin{cases} \mathbf{x}(k), & \text{for } |\mathbf{x}(k)| > \xi \\ 0, & \text{otherwise,} \end{cases} \quad (30)$$

where ξ is the K largest component of **x**. The algorithm is summarized in Algorithm 4.

The stopping criterion for IHT can be a fixed number of iterations or the algorithm terminates when the sparse vector does not change much between consecutive iterations.

The soft-thresholding function can be defined as

$$S_\tau(\mathbf{x}(k)) = \begin{cases} \mathbf{x}(k) - \tau, & \mathbf{x} > \tau \\ 0, & |\mathbf{x}(k)| < \tau \\ \mathbf{x}(k) + \tau, & \mathbf{x} < -\tau \end{cases} \quad (31)$$

and it is applied to each element of **x**.

3.5. Iterative Soft Thresholding for Solving LASSO. Let us consider the minimization of the function:

$$J(\mathbf{x}) = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 + \lambda \|\mathbf{x}\|_1, \quad (32)$$

as given by the LASSO minimization problem (22). One of the algorithms that has been used for solving (32) is the iterated

```
Input:
    ▶ Transform matrix Ψ, Measurement matrix Φ
    ▶ CS matrix A: A = ΦΨ
    ▶ Measurement vector y
    ▶ K being the signal sparsity
    ▶ Halting criterion
Output:
    ▶ K-sparse approximation x̂ of the target signal
(1) x₀ ← 0, r ← y, i ← 0
(2) while halting condition false do
(3) i ← i + 1
(4) z ← A*r                                            (signal proxy)
(5) Ω ← supp(z²ᴷ)                      (support of best 2K-sparse approximation)
(6) T ← Ω ∪ supp(xᵢ₋₁)                              (merge supports)
(7) x̄ = arg min_{x̄:supp(x̄)=T}‖Ax̄ − y‖₂²                 (solve least squares)
(8) xᵢ ← x̄ᴷ                            (Prune: best K-sparse approximation)
(9) r ← y − Axᵢ                                    (update current sample)
(10) end while
(11) x̂ ← xᵢ
(12) return x̂
```

<div align="center">ALGORITHM 3: CoSaMP.</div>

```
Input:
    ▶ Transform matrix Ψ, Measurement matrix Φ
    ▶ CS matrix A: A = ΦΨ
    ▶ Measurement vector y
    ▶ K being the signal sparsity
Output:
    ▶ K-sparse approximation x̂
(1) x₀ ← 0
(2) while stopping criterion do
(3) i ← i + 1
(4) xᵢ ← Hₖ(xᵢ₋₁ + Aᵀ(y − Axᵢ₋₁))
(5) end while
(6) x̂ ← xᵢ
(7) return x̂
```

<div align="center">ALGORITHM 4: Iterative hard thresholding.</div>

soft-thresholding algorithm (ISTA), also called the thresholded Landweber algorithm. In order to provide an iterative procedure for solving the considered minimization problem, we use the minimization-maximization [26] approach to minimize $J(\mathbf{x})$. Hence, we should create a function $G_k(\mathbf{x})$ that is equal to $J(\mathbf{x})$ at $\mathbf{x}_k$; otherwise, it upper-bounds $J(\mathbf{x})$. Minimizing a majorizer $G_k(\mathbf{x})$ is easier and avoids solving a system of equations. Hence [27],

$$G_k(\mathbf{x}) = J(\mathbf{x}) + \text{nonnegative function of } \mathbf{x},$$

$$
\begin{aligned}
G_k(\mathbf{x}) &= J(\mathbf{x}) + (\mathbf{x} - \mathbf{x}_k)^T (\alpha \mathbf{I} - \mathbf{A}^T \mathbf{A})(\mathbf{x} - \mathbf{x}_k) \\
&= \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 + \lambda \|\mathbf{x}\|_1 \\
&\quad + (\mathbf{x} - \mathbf{x}_k)^T (\alpha \mathbf{I} - \mathbf{A}^T \mathbf{A})(\mathbf{x} - \mathbf{x}_k),
\end{aligned}
\tag{33}
$$

and α is such that the added function is nonnegative, meaning that α must be equal to or greater than the maximum eigenvalue of $\mathbf{A}^T\mathbf{A}$:

$$\alpha > \max \text{eig} \left\{ \mathbf{A}^T \mathbf{A} \right\}. \tag{34}$$

In order to minimize $G_k(\mathbf{x})$, the gradient of $G_k(\mathbf{x})$ should be zero:

$$
\begin{aligned}
&\frac{\partial G_k(\mathbf{x})}{\partial \mathbf{x}^T} = 0 \Longrightarrow \\
&- 2\mathbf{A}^T \mathbf{y} + 2\mathbf{A}^T \mathbf{A}\mathbf{x} + \lambda \, \text{sign}(\mathbf{x}) \\
&+ 2 \left(\alpha \mathbf{I} - \mathbf{A}^T \mathbf{A} \right)(\mathbf{x} - \mathbf{x}_k) = 0.
\end{aligned}
\tag{35}
$$

The solution is

$$\mathbf{x} + \frac{\lambda}{2\alpha} \text{sign}(\mathbf{x}) = \frac{1}{\alpha}\mathbf{A}^T (\mathbf{y} - \mathbf{A}\mathbf{x}_k) + \mathbf{x}_k. \tag{36}$$

The solution of the linear equation

$$a + \frac{\lambda}{2\alpha} \text{sign}(a) = b \tag{37}$$

is obtained using the soft-thresholding approach:

$$
a = S_\lambda(b) = \begin{cases} b - \lambda, & b > \lambda \\ 0, & |b| < \lambda \\ b + \lambda, & b < -\lambda. \end{cases}
\tag{38}
$$

Hence, we may write the soft-thresholding-based solution of (36) as follows:

$$\mathbf{x} = S_{\lambda/2\alpha} \left(\frac{1}{\alpha} \mathbf{A}^T (\mathbf{y} - \mathbf{A}\mathbf{x}_k) + \mathbf{x}_k \right). \tag{39}$$

In terms of iteration $k + 1$, the previous relation can be rewritten as

$$\mathbf{x}_{k+1} = S_{\lambda/2\alpha}\left(\frac{1}{\alpha}\mathbf{A}^T\left(\mathbf{y} - \mathbf{A}\mathbf{x}_k\right) + \mathbf{x}_k\right). \tag{40}$$

3.6. Automated Threshold Based Iterative Solution. This algorithm (see [12, 13]) starts from the assumption that the missing samples, modeled by zero values in the signal domain, produce noise in the transform domain. Namely, the zero values at the positions of missing samples can be observed as a result of adding noise to these samples, where the noise at the unavailable positions has values of original signal samples with opposite sign. For instance, let us observe the signals that are sparse in the Fourier transform domain. The variance of the mentioned noise when observed in the discrete Fourier transform (DFT) domain can be calculated as

$$\sigma_{MS}^2 \longleftarrow M\frac{N - M}{N - 1}\sum_{i=1}^{M}\frac{|\mathbf{y}(i)|^2}{M}, \tag{41}$$

where M is the number of available and N is the total number of samples and $\mathbf{y}$ is a measurement vector. Consequently, using the variance of noise, it is possible to define a threshold to separate signal components from the nonsignal components in the DFT domain.

For a desired probability $P(T)$, the threshold is derived as

$$T = \sqrt{-\sigma_{MS}^2 \log\left(1 - P(T)^{1/N}\right)}. \tag{42}$$

The automated threshold based algorithm, in each iteration, detects a certain number of DFT components above the threshold. A set of positions $\mathbf{k}^i$ corresponding to the detected components is selected and the contribution of these components is removed from the signal's DFT. This will reveal the remaining components that are below the noise level. Further, it is necessary to update the noise variance and threshold value for the new iteration. Since the algorithm detects a set of components in each iteration, it usually needs just a couple of iterations to recover the entire signal. In the case that all signal components are above the noise level in DFT, the component detection and reconstruction are achieved in single iteration.

3.7. Adaptive Gradient-Based Algorithm. This algorithm belongs to the group of convex minimization approaches [11]. Unlike other convex minimization approaches, this method starts from some initial values of unavailable samples (initial state) which are changed through the iterations in a way to constantly improve the concentration in sparsity domain. In general, it does not require the signal to be strictly sparse in certain transform domain, which is an advantage over other methods. Particularly, the missing samples in the signal domain can be considered as zero values. In each iteration, the missing samples are changed for $+\Delta$ and for $-\Delta$. Then, for both changes, the concentration is measured as ℓ_1-norm of the transform domain vectors $\mathbf{x}^+$ (for $+\Delta$ change) and $\mathbf{x}^-$ (for $-\Delta$ change), while the gradient is determined using their difference (lines 8, 9, and 10). Finally, the gradient is used to update the values of missing samples. Here, it is important to

note that each sample is observed separately in this algorithm and one iteration is finished when all samples are processed.

When the algorithm reaches the vicinity of the sparsity measure minimum, the gradient changes direction for almost 180 degrees (line 16), meaning that the step Δ needs to be decreased (line 17).

The precision of the result in this iterative algorithm is estimated based on the change of the result in the last iteration (lines 18 and 19).

4. Exploring Different Transform Domains for CS Signal Reconstruction

4.1. CS in the Standard Fourier Transform Domain. As the simplest case of CS scenario, we will assume a multicomponent signal that consists of K sinusoids that are sparse in the DFT domain:

$$s(n) = \sum_{i=1}^{K} A_i e^{j2\pi k_i n/N}, \tag{43}$$

where the sparsity level is $K \ll N$, where N is total signal length, while A_i and k_i denote amplitudes and frequencies of signal components, respectively. Since s is sparse in the DFT domain, then we can write

$$\mathbf{s} = \mathbf{\Psi}\mathbf{F} = \mathscr{F}^{-1}\mathbf{F}, \tag{44}$$

where $\mathbf{F}$ is a vector of DFT coefficients where at most K coefficients are nonzero, while $\mathscr{F}^{-1}$ is the inverse Fourier transform matrix of size $N \times N$.

If $\mathbf{s}$ is a signal in compressive sensing application, then only the random measurements $\mathbf{y} \subset \mathbf{s}$ are available and these are defined by the set of M positions $\{n_1, n_2, n_M\}$. Therefore, the measurement process can be modeled by matrix $\mathbf{\Phi}$:

$$\mathbf{y} = \mathbf{\Phi}\mathscr{F}^{-1}\mathbf{F} = \mathbf{A}\mathbf{F}. \tag{45}$$

The CS matrix $\mathbf{A}$ represents a partial random inverse Fourier transform matrix obtained by omitting rows from $\mathscr{F}^{-1}$ that corresponds to unavailable samples positions:

$$\mathbf{A} = \begin{bmatrix} \psi_0(n_1) & \psi_1(n_1) & \cdots & \psi_{N-1}(n_1) \\ \psi_0(n_2) & \psi_1(n_2) & \cdots & \psi_{N-1}(n_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_0(n_M) & \psi_1(n_M) & \cdots & \psi_{N-1}(n_M) \end{bmatrix}, \tag{46}$$

with the Fourier basis functions:

$$\psi_k(n) = e^{j2\pi kn/N}. \tag{47}$$

The CS problem formulation is now given as

$$\begin{aligned} \min \quad & \|\mathbf{F}\|_1 \\ \text{s.t.} \quad & \mathbf{y} = \mathbf{A}\mathbf{F}. \end{aligned} \tag{48}$$

4.2. CS in the Polynomial Fourier Transform Domain. In this section, we consider the possibility of CS reconstruction of polynomial phase signals [28]. Observe the multicomponent polynomial phase signal vector **s**, with elements $s(n)$:

$$s(n) = \sum_{i=1}^{K} s_i(n) = \sum_{i=1}^{K} r_i e^{j(2\pi/N)(na_{1i}+n^2 a_{2i}/2+\cdots+n^p a_{pi}/p!)}, \quad (49)$$

where the polynomial coefficients are assumed to be bounded integers. The assumption is that the signal can be considered as sparse in the polynomial Fourier transform (PFT) domain. Hence, the discrete PFT form is given by

$$X\left(k_{1i}, \ldots, k_{pi}\right)$$
$$= \sum_{n=0}^{N-1} \sum_{i=1}^{K} r_i e^{j(2\pi/N)(na_{1i}+n^2 a_{2i}/2+\cdots+n^p a_{pi}/p!)} \quad (50)$$
$$\times e^{-j(2\pi/N)(n^2 k_{2i}/2+\cdots+n^p k_{pi}/p!)} e^{-j(2\pi/N)nk_1}.$$

If we choose a set of parameters $(k_{2i}, k_{3i}, \ldots, k_{pi})$ that match the polynomial phase coefficients $(a_{2i}, a_{3i}, \ldots, a_{pi})$

$$\left(k_{2i}, k_{3i}, \ldots, k_{pi}\right) = \left(a_{2i}, a_{3i}, \ldots, a_{pi}\right), \quad (51)$$

then the ith signal component is demodulated and becomes a sinusoid: $r_i e^{j(2\pi/N)(na_{1i})}$ which is dominant in the PFT. In other words, the spectrum is highly concentrated at $k = a_{1i}$. Therefore, we might say that if (51) is satisfied then the PFT is compressible with the dominant ith component. Note that the sparsity (compressibility) in the PFT domain is observed with respect to the single demodulated component.

In order to define the CS problem in the PFT domain, instead of the signal **s** itself, we will consider a modified signal form obtained after the multiplication with the exponential term:

$$\xi(n) = e^{-j(2\pi/N)(n^2 k_{2i}/2+\cdots+n^p k_{pi}/p!)}, \quad (52)$$

given in the PFT definition (50). The new signal form is obtained as

$$z(n) = s(n)\xi(n) = \sum_{i=1}^{K} r_i$$
$$\cdot e^{j(2\pi/N)(na_{1i}+n^2 a_{2i}/2+\cdots+n^p a_{pi}/p!)} e^{-j(2\pi/N)(n^2 k_{2i}/2+\cdots+n^p k_{pi}/p!)} \quad (53)$$

or in the vector form:

$$\mathbf{z} = \mathbf{s}\boldsymbol{\xi}. \quad (54)$$

Then, the PFT definition given by (50) can be rewritten in the vector form as follows:

$$\mathbf{X} = \mathscr{F}\mathbf{z}, \quad (55)$$

where $\mathscr{F}$ is the discrete Fourier transform matrix ($N \times N$).

For a chosen set of parameters $(k_{2i}, k_{3i}, \ldots, k_{pi})$ in $\boldsymbol{\xi}$ that is equal to the set $(a_{2i}, a_{3i}, \ldots, a_{ni})$ in **s**, $\mathbf{X} = \mathbf{X}_i$ is characterized by one dominant sinusoidal component at the frequency a_{1i}.

4.2.1. CS Scenario. Now, assume that **z** is incomplete in the compressive sensing sense, and instead of **z** we are dealing with M available measurements defined by vector $\mathbf{y}_z$, and

$$\mathbf{y}_z = \mathbf{A}_z \mathbf{X}, \quad (56)$$

where $\mathbf{A}_z$ is the partial random Fourier matrix obtained by omitting rows from $\mathscr{F}$ that correspond to the unavailable samples. When $(k_{2i}, k_{3i}, \ldots, k_{pi}) = (a_{2i}, a_{3i}, \ldots, a_{pi})$, then **X** can be observed as a demodulated version of the ith signal component $\mathbf{X}_i$, having the dominant ith component in the spectrum with the support k_{1i}. The rest of the components in spectrum are much lower than $\mathbf{X}_i$ and could be observed as noise. Hence, we may write the minimization problem in the form

$$\begin{aligned} \min \quad & \|\mathbf{X}_i\|_1 \\ \text{s.t.} \quad & \|\mathbf{y}_z - \mathbf{A}_z \mathbf{X}_i\|_2 < T, \end{aligned} \quad (57)$$

where T is certain threshold. The dominant components can be detected using the threshold based algorithm (Algorithm 5) described in the previous section. Using an iterative procedure, one may change the values of parameters $k_{2i}, \ldots, k_{pi}$ between $k_{\min}$ and $k_{\max}$, $\forall i$. The algorithm will detect the support k_{1i} when the set $(k_{2i}, k_{3i}, \ldots, k_{pi})$ matches the set $(a_{2i}, a_{3i}, \ldots, a_{pi})$; otherwise, no component support is detected. Hence, as a result of this phase, we have identified the sets of signal phase parameters: $\mathbf{k}_i = (k_{1i}, k_{2i}, \ldots, k_{ni}) = (a_{1i}, a_{2i}, \ldots, a_{ni})$.

4.2.2. Reconstruction of Exact Components Amplitudes. The next phase is reconstruction of components amplitudes. Denote the set of measurements positions by $(n_1, n_2, \ldots, n_M)$, such that $[\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(M)] = [\mathbf{s}(n_1), \mathbf{s}(n_2), \ldots, \mathbf{s}(n_M)]$, while the detected sets of signal phase parameters are $\mathbf{k}_i = (k_{1i}, k_{2i}, \ldots, k_{ni}) = (a_{1i}, a_{2i}, \ldots, a_{ni})$.

In order to calculate the exact amplitudes $r_1, r_2, \ldots, r_K$ of signal components, we observe the set of equations in the form

$$\begin{bmatrix} s(n_1) \\ \vdots \\ s(n_M) \end{bmatrix}$$
$$= \begin{bmatrix} e^{j(2\pi/N)(n_1 k_{11}+\cdots+n_1^p k_{p1})} & \cdots & e^{j(2\pi/N)(n_1 k_{1K}+\cdots+n_1^p k_{pK})} \\ e^{j(2\pi/N)(n_2 k_{11}+\cdots+n_2^p k_{p1})} & \cdots & e^{j(2\pi/N)(n_2 k_{1K}+\cdots+n_2^p k_{pK})} \\ \vdots & & \vdots \\ e^{j(2\pi/N)(n_N k_{11}+\cdots+n_N^p k_{p1})} & \cdots & e^{j(2\pi/N)(n_M k_{1K}+\cdots+n_M^p k_{pK})} \end{bmatrix} \begin{bmatrix} r_1 \\ \vdots \\ r_K \end{bmatrix} \quad (58)$$

or in other words we have another system of equations given by

$$\mathbf{y} = \mathbf{A}\mathbf{R}, \quad (59)$$

Input:
▸ Transform matrix $\boldsymbol{\Psi}$, Measurement matrix $\boldsymbol{\Phi}$
▸ $\mathbf{N}_M = \{n_1, n_2, \ldots, n_M\}$, M –number of available samples, N-original signal length
▸ Matrix $\mathbf{A}$: $\mathbf{A} = \boldsymbol{\Phi}\boldsymbol{\Psi}$ ($\mathbf{A}$ is obtained as a partial random Fourier matrix: $\mathbf{A} = \boldsymbol{\Phi}\boldsymbol{\Psi} = \boldsymbol{\Psi}\{\vartheta\}$,
  $\vartheta = \{\psi_{n_1}, \psi_{n_2}, \ldots, \psi_{n_M}\}$
▸ Measurement vector $\mathbf{y} = \boldsymbol{\Phi}\mathbf{f}$
Output:
(1) $i = 1$, $\mathbf{N}_M = \{n_1, n_2, \ldots, n_M\}$
(2) $\mathbf{p} = \varnothing$.
(3) $P \leftarrow 0.99$ (*Set desired probability P*)
(4) $\sigma_{MS}^2 \leftarrow M((N-M)/(N-1))\sum_{i=1}^{M}(|\mathbf{y}(i)|^2/M)$ (*Calculate variance*)
(5) $T^i \leftarrow \sqrt{-\sigma_{MS}^2 \log(1 - P(T)^{1/N})}$ (*Calculate threshold*)
(6) $\mathbf{X}^i \leftarrow \mathbf{A}^{-1}\mathbf{y}$ (*Calculate initial* DFT *of* $\mathbf{y}$)
(7) $\mathbf{k}^i \leftarrow \arg\{|\mathbf{X}^i| > T^i/N\}$ (*Find comp. in* $\mathbf{X}^i$ *above the* T^i)
(8) **Update** $\mathbf{p} \leftarrow \mathbf{p} \cup \mathbf{k}_i$
(9) $\mathbf{A}_{cs}{}^i \leftarrow \mathbf{A}(\mathbf{k}^i)$ (*CS matrix*)
(10) $\widetilde{\mathbf{X}}^i \leftarrow (\mathbf{A}_{cs}{}^{iT}\mathbf{A}_{cs}{}^i)^{-1}\mathbf{A}_{cs}{}^{iT}\mathbf{y}$
(11) **for** $\forall p \in \mathbf{p}$:
(12) $\mathbf{y} \leftarrow \mathbf{y} - (M/N)\mathbf{X}^i(p)\exp(j2\pi p\mathbf{N}_M/N)$;   (*Update* $\mathbf{y}$)
(13) **Update** $\sigma_{MS}^2 \leftarrow (M(N-M)/(N-1))\sum|\mathbf{y}|^2/M$
(14) **If** $\sum|\mathbf{y}|^2/M < \delta$ **break;   Else**
(15) **Set** $i = i + 1$, and go to (5).
(16) **return** $\widetilde{\mathbf{X}}^i$, $\mathbf{k}$

ALGORITHM 5: Automated threshold based iterative algorithm.

where $\mathbf{R} = [r_1, \ldots, r_K]^T$ contains the desired K signal amplitudes. The matrix $\mathbf{A}$ of size $N \times K$ is based on the PFT:

$$\mathbf{A}$$
$$= \begin{bmatrix} e^{j(2\pi/N)(n_1 k_{11} + \cdots + n_1^p k_{p1})} & \cdots & e^{j(2\pi/N)(n_1 k_{1K} + \cdots + n_1^p k_{pK})} \\ e^{j(2\pi/N)(n_2 k_{11} + \cdots + n_2^p k_{p1})} & \cdots & e^{j(2\pi/N)(n_2 k_{1K} + \cdots + n_2^p k_{pK})} \\ \vdots & & \vdots \\ e^{j(2\pi/N)(n_M k_{11} + \cdots + n_M^p k_{p1})} & \cdots & e^{j(2\pi/N)(n_M k_{1K} + \cdots + n_M^p k_{pK})} \end{bmatrix}. \quad (60)$$

The rows of $\mathbf{A}$ correspond to positions of measurements $(n_1, n_2, \ldots, n_M)$, and columns correspond to phase parameters $\mathbf{k}_i = (k_{1i}, k_{2i}, \ldots, k_{ni}) = (a_{1i}, a_{2i}, \ldots, a_{ni})$, for $i = 1, \ldots, K$. The solution of the observed problem can be obtained in the least squares sense as follows:

$$\mathbf{X} = (\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*\mathbf{y}. \quad (61)$$

The resulting reconstructed signal is obtained as

$$s(n) = r_1 e^{-j(2\pi/N)(na_{11} + \cdots + n^p a_{p1}/p!)} + \cdots \\ + r_K e^{-j(2\pi/N)(na_{1K} + \cdots + n^p a_{pK}/p!)}. \quad (62)$$

4.3. CS in the Hermite Transform Domain. The Hermite expansion using N Hermite functions can be written in terms of the Hermite transform matrix. First, let us define the Hermite transform matrix $\mathbf{H}$ (of size $N \times N$) [29, 30]:

$$\mathbf{H} = \frac{1}{N}$$
$$\cdot \begin{bmatrix} \dfrac{\psi_0(1)}{(\psi_{N-1}(1))^2} & \dfrac{\psi_0(2)}{(\psi_{N-1}(2))^2} & \cdots & \dfrac{\psi_0(N)}{(\psi_{N-1}(N))^2} \\ \dfrac{\psi_1(1)}{(\psi_{N-1}(1))^2} & \dfrac{\psi_1(2)}{(\psi_{N-1}(2))^2} & \cdots & \dfrac{\psi_1(N)}{(\psi_{N-1}(N))^2} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\psi_{N-1}(1)}{(\psi_{N-1}(1))^2} & \dfrac{\psi_{N-1}(2)}{(\psi_{N-1}(2))^2} & \cdots & \dfrac{\psi_{N-1}(N)}{(\psi_{N-1}(N))^2} \end{bmatrix}. \quad (63)$$

Here, the pth order Hermite basis function is defined using the pth order Hermite polynomial as follows:

$$\psi_p(n) = (\sigma 2^p p! \sqrt{\pi})^{-1/2} e^{-n^2/2} H_p\left(\frac{n}{\sigma}\right), \quad (64)$$

where σ is the scaling factor used to "stretch" or "compress" Hermite functions, in order to match the signal.

The Hermite functions are usually calculated using the following fast recursive realization:

$$\psi_0(n) = \frac{1}{\sqrt[4]{\pi}} e^{-n^2/2\sigma^2},$$

$$\psi_1(n) = \frac{\sqrt{2}n}{\sqrt[4]{\pi}} e^{-n^2/2\sigma^2}, \quad (65)$$

$$\psi_i(n) = n\sqrt{\frac{2}{i}}\Psi_{i-1}(n) - \sqrt{\frac{i-1}{i}}\Psi_{i-2}(n), \quad \forall i \geq 2.$$

The Hermite transform coefficients can be calculated in the matrix form as follows:

$$\mathbf{c} = \mathbf{Hs}, \tag{66}$$

where the vector of Hermite transform coefficients is $\mathbf{c} = [c_0, c_1, \ldots, c_{N-1}]^T$, while the vector of signal samples is $\mathbf{s} = [s(1), s(2), \ldots, s(N)]^T$.

Following the Gauss-Hermite approximation, the inverse matrix $\mathbf{\Psi} = \mathbf{H}^{-1}$ contains N Hermite functions and it is given by

$$\mathbf{\Psi} = \begin{bmatrix} \psi_1(1) & \psi_2(1) & \cdots & \psi_N(1) \\ \psi_1(2) & \psi_2(2) & \cdots & \psi_N(2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(N) & \psi_2(N) & \cdots & \psi_N(N) \end{bmatrix}. \tag{67}$$

Now, in the context of CS, let us assume that a signal $\mathbf{s}$ is K-sparse in the Hermite transform domain, meaning that it can be represented by a small set of nonzero Hermite coefficients c:

$$s(n) = \sum_{p=k_1}^{k_K} c_p \psi_p(n), \tag{68}$$

where c_p is the pth element from the vector $\mathbf{c}$ of Hermite expansion coefficients. In the matrix form we may write

$$\mathbf{s} = \mathbf{\Psi c}, \tag{69}$$

such that most of the coefficients in $\mathbf{c}$ are zero values. Furthermore, assume that only the compressive sensing is done using random selection of M signal values $[n_1, n_2, \ldots, n_M]$. Then, the CS matrix $\mathbf{A}$ can be defined as random partial inverse Hermite transform matrix:

$$\mathbf{A} = \begin{bmatrix} \psi_0(n_1) & \psi_1(n_1) & \cdots & \psi_{N-1}(n_1) \\ \psi_0(n_2) & \psi_1(n_2) & \cdots & \psi_{N-1}(n_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_0(n_M) & \psi_1(n_M) & \cdots & \psi_{N-1}(n_M) \end{bmatrix}. \tag{70}$$

For a vector of available measurements vector $\mathbf{y} = (\mathbf{s}(n_1), \mathbf{s}(n_2), \ldots, \mathbf{s}(n_M))$, the linear system of equations (undetermined system of M linear equations and N unknowns) can be written as

$$\mathbf{y} = \mathbf{Ac}. \tag{71}$$

Since $\mathbf{c}$ has only K nonzero components, the reconstruction problem can be defined as

$$\begin{aligned} \min \quad & \|\mathbf{c}\|_1 \\ \text{s.t.} \quad & \mathbf{y} = \mathbf{Ac}. \end{aligned} \tag{72}$$

If we identify the signal support in the Hermite transform domain by the set of indices $[k_1, k_2, \ldots, k_K]$, then the problem can be solved in the least squares sense as

$$\widehat{\mathbf{c}} = \left(\mathbf{\Theta}_{cs}^* \mathbf{\Theta}_{cs}\right)^{-1} \mathbf{\Theta}_{cs}^* \mathbf{y}, \tag{73}$$

where $(*)$ denotes the conjugate transpose operation, while

$$\mathbf{\Theta}_{cs} = \begin{bmatrix} \psi_{k_1}(n_1) & \psi_{k_1}(n_2) & \cdots & \psi_{k_1}(n_M) \\ \psi_{k_2}(n_1) & \psi_{k_2}(n_2) & \cdots & \psi_{k_2}(n_M) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{k_K}(n_1) & \psi_{k_K}(n_2) & \cdots & \psi_{k_K}(n_M) \end{bmatrix}, \tag{74}$$

and $\widehat{\mathbf{c}} = [c_{k_1}, c_{k_2}, \ldots, c_{k_K}]^T$. The estimated signal $\mathbf{s}$ can be now obtained according to (68).

4.4. CS in the Time-Frequency Domain. Starting from the definition of the short-time Fourier transform

$$\text{STFT}(n, k) = \sum_{m=0}^{M-1} x(n+m) e^{-j2\pi mk/M}, \tag{75}$$

we can define the following STFT vector and signal vector:

$$\begin{aligned} \mathbf{STFT}_M(n) &= [\text{STFT}(n, 0), \ldots, \text{STFT}(n, M-1)]^T, \\ \mathbf{x}(n) &= [x(n), x(n+1), \ldots, x(n+M-1)]^T, \end{aligned} \tag{76}$$

respectively. The STFT at an instant n can be now rewritten in the matrix form:

$$\mathbf{STFT}_M(n) = \mathscr{F}_M \mathbf{x}(n), \tag{77}$$

where $\mathscr{F}_M$ is the standard Fourier transform matrix of size $M \times M$ with elements

$$\mathscr{F}_M(m, k) = e^{-j2\pi km/M}. \tag{78}$$

If we consider the nonoverlapping signal segments for the STFT calculation, then n takes the values from the set $\{0, M, 2M, \ldots, N-M\}$, where M is the window length while N is the total signal length. The STFT vector, for $n \in \{0, M, 2M, \ldots, N-M\}$, is given by

$$\mathbf{STFT} = \begin{bmatrix} \mathbf{STFT}_M(0) \\ \mathbf{STFT}_M(M) \\ \vdots \\ \mathbf{STFT}_M(N-M) \end{bmatrix}. \tag{79}$$

Consequently, (77) can be extended to the entire STFT for all considered values of $n \in \{0, M, 2M, \ldots, N-M\}$ as follows [31, 32]:

$$\begin{bmatrix} \mathbf{STFT}_M(0) \\ \mathbf{STFT}_M(M) \\ \vdots \\ \mathbf{STFT}_M(N-M) \end{bmatrix}$$

$$= \begin{bmatrix} \mathscr{F}_M & 0_M & \cdots & 0_M \\ 0_M & \mathscr{F}_M & \cdots & 0_M \\ \cdots & \cdots & \cdots & \cdots \\ 0_M & 0_M & \cdots & \mathscr{F}_M \end{bmatrix} \begin{bmatrix} \mathbf{x}(0) \\ \mathbf{x}(M) \\ \vdots \\ \mathbf{x}(N-M) \end{bmatrix}, \tag{80}$$
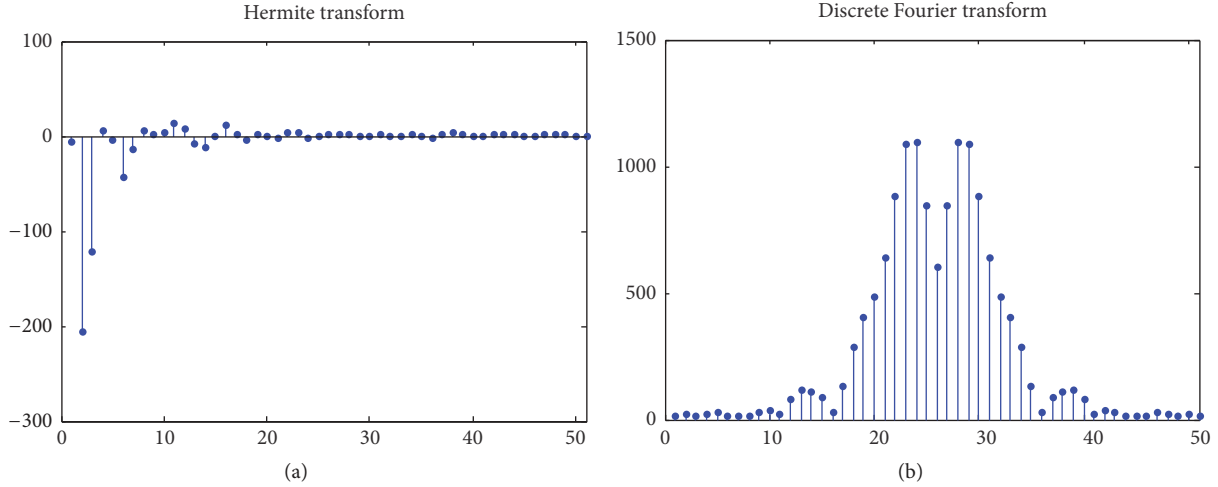
Figure 1: (a) Hermite transform coefficients of the selected QRS complex ($\sigma = 6.8038$) and (b) Fourier transform coefficients (absolute values).

or in the compact form

$$\mathbf{STFT} = \mathscr{F}\mathbf{x}. \tag{81}$$

In the case of time-varying windows with lengths $\{M_1, M_2, \ldots, M_Q\}$, the previous system of equations becomes

$$
\begin{bmatrix}
\mathbf{STFT}_{M_0}(0) \\
\mathbf{STFT}_{M_1}(M_0) \\
\vdots \\
\mathbf{STFT}_{M_Q}(N - M_Q)
\end{bmatrix}
\tag{82}
$$

$$
=
\begin{bmatrix}
\mathscr{F}_{M_0} & 0 & \cdots & 0 \\
0_M & \mathscr{F}_{M_1} & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots \\
0 & 0 & \cdots & \mathscr{F}_{M_Q}
\end{bmatrix}
\begin{bmatrix}
\mathbf{x}(0) \\
\mathbf{x}(M) \\
\vdots \\
\mathbf{x}(N - M)
\end{bmatrix}.
$$

Note that the signal vector is $\mathbf{x} = [\mathbf{x}(0)^T, \mathbf{x}(M)^T, \ldots, \mathbf{x}(N - M)^T]^T = [x(0), x(1), \ldots, x(N-1)]^T$ and can be expressed using the N-point inverse Fourier transform as follows:

$$\mathbf{x} = \mathscr{F}_N^{-1}\mathbf{X}, \tag{83}$$

where $\mathbf{X}$ is a sparse vector of DFT coefficients. By combining (81) and (83), we obtain [31]

$$\mathbf{STFT} = \mathscr{F}\mathscr{F}_N^{-1}\mathbf{X}. \tag{84}$$

Furthermore, if we assume that $\mathbf{STFT}$ vector is not completely available but has only a small percent of available samples, then we are dealing with the CS problem in the form

$$\mathbf{y}_{\mathrm{STFT}} = \mathbf{AX}, \tag{85}$$

where $\mathbf{y}_{\mathrm{STFT}}$ is a vector of available measurements from $\mathbf{STFT}$, while $\mathbf{A}$ is the CS matrix obtained as partial combined

transform matrix $\mathbf{\Psi} = \mathscr{F}\mathscr{F}_N^{-1}$ with rows corresponding to the available samples $\mathbf{y}_{\mathrm{STFT}}$. In order to reconstruct the entire $\mathbf{STFT}$, we can define the minimization problem in the form

$$
\begin{aligned}
\min \quad & \|\mathbf{X}\|_1 \\
\text{s.t.} \quad & \mathbf{y}_{\mathrm{STFT}} = \mathbf{AX},
\end{aligned}
\tag{86}
$$

which can be solved using any of the algorithms provided in the previous section.

5. Experimental Evaluation

Example 1. Let us observe the QRS complex extracted from the real ECG signal. It has been known that these types of signals are sparse in the Hermite transform (HT) domain [33, 34]. As an illustration, we present the Hermite transform and discrete Fourier transform (DFT) in Figure 1, from which it can be observed that HT provides sparse representation while DFT is dense. The analyzed signal is obtained from the MIT-ECG Compression Test Database [35], originally sampled uniformly with $T = 1/250$ [s] being the sampling period, with amplitude gain 1/400. Extracted QRS is of length $N = 51$, centered at the R peak. Note that, in order to provide the sparsest HT representation, the scaling factor σ should be set to the optimal value [33, 34], which is 6.8038 in this example, that is, $\sigma \Delta t = 0.027$ in seconds, proportional to the scaling factor 0.017 presented in [34] for signal with 27 samples, when it is scaled with the signal lengths ratio. Also, the QRS signal is resampled at the zeros of Hermite polynomial using sinc interpolation functions as described in [33].

The QRS signal is subject to compressed sensing approach, meaning that it is represented by 50% of available measurements. The available measurements and the corresponding HT (which is not sparse in the CS conditions) are shown in Figure 2.

In order to reconstruct the entire QRS signal from available measurements, two of the presented algorithms
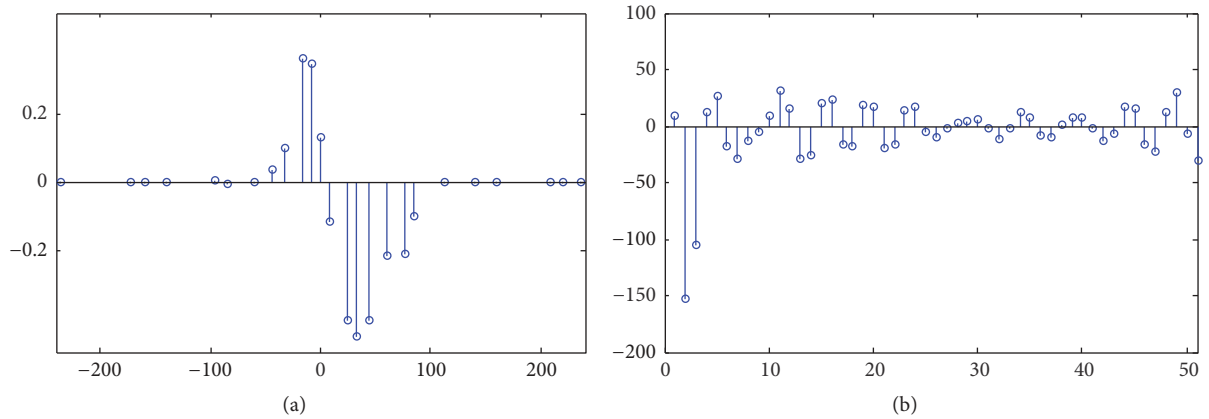
FIGURE 2: (a) Measurements of the selected QRS complex (50% of the total samples number) and (b) Hermite transform of measurements vector.
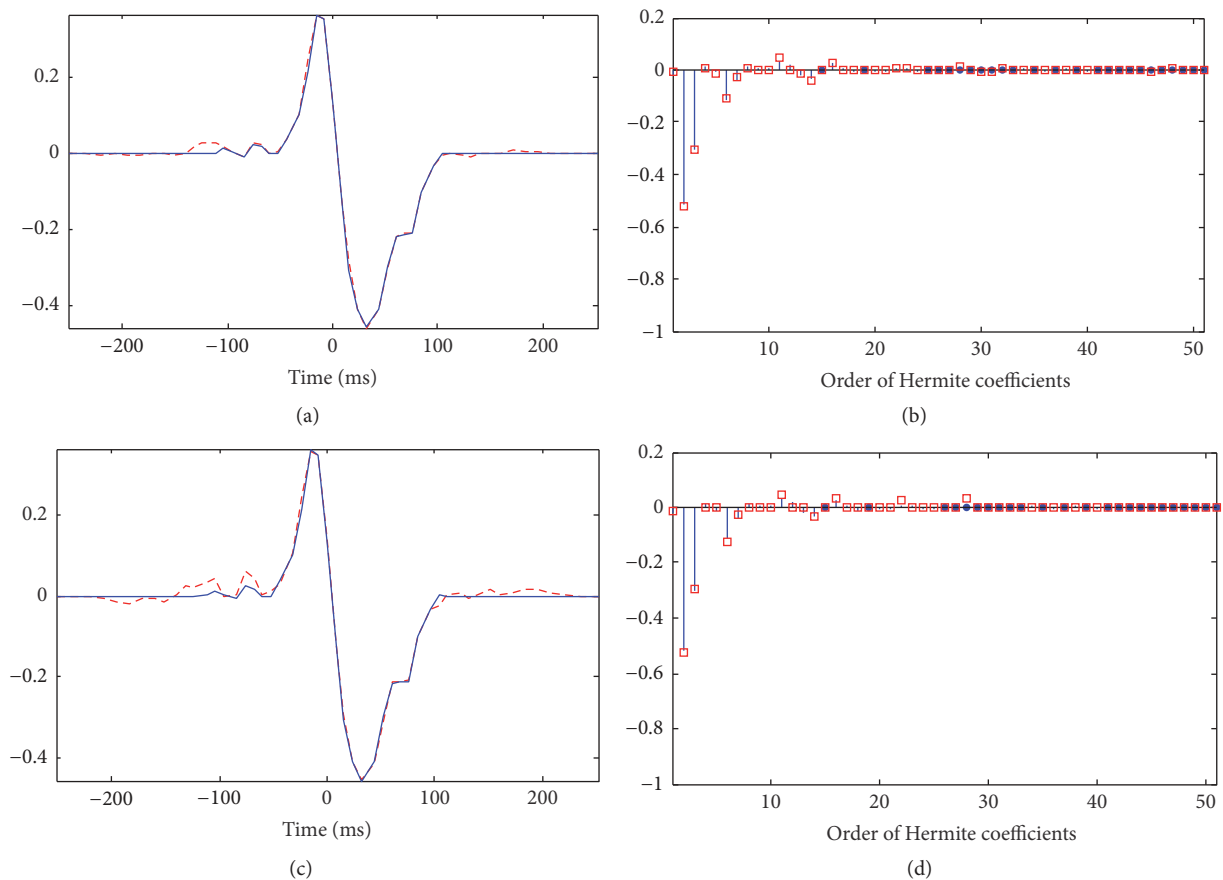

FIGURE 3: Results of signal reconstruction using gradient-based algorithms: (a) original signal (solid line) and reconstructed signal (dash line) and (b) Hermite coefficients of the original signal (solid line) and the reconstructed signal (squares). Reconstruction results using OMP algorithm: (c) original signal (solid) and reconstructed signal (dash) in time domain and (d) Hermite coefficients of original signal (solid line) and reconstructed signal (squares).

are employed, namely, the OMP and gradient-based reconstruction algorithm.

The reconstruction results are shown in Figure 3 in both time and transform domain. It can be observed that both algorithms provide quite successful reconstruction

performance, which is certainly better in the case of gradient-based approach. In order to measure and compare the reconstruction performance of these two algorithms, the mean square error (MSE) is calculated, showing that the gradient-based algorithm in the considered case provides

Input:
▶ Available samples position $\mathbf{N}_M = \{n_1, n_2, \ldots, n_M\}$
▶ Missing samples position $\mathbf{N} \setminus \mathbf{N}_M$, where $\mathbf{N} = \{1, 2, \ldots, N\}$
▶ Available samples $\mathbf{f}(n)$, $n \in \mathbf{N}_M$
▶ Measurement vector: $\mathbf{y}(n) = \{\mathbf{f}(n)$, for $n \in \mathbf{N}_M$; 0, for $n \in \mathbf{N} \setminus \mathbf{N}_M\}$
Output:
(1) Set $i = 0$, $\mathbf{y}^{(0)}(n) \leftarrow \mathbf{y}(n)$, $\Delta \leftarrow \max |\mathbf{y}^{(0)}(n)|$
(2) **repeat**
(3)   Set $\mathbf{y}_p(n) = \mathbf{y}^{(i)}(n)$
(4)   **repeat**
(5) $i \leftarrow i + 1$
(6)   **for** $n_s \leftarrow 0$ to $N - 1$ **do**
(7)   **if** $n_s \in \mathbf{N} \setminus \mathbf{N}_M$
(8) $\mathbf{x}^+(k) \leftarrow \mathfrak{I}\{\mathbf{y}^{(i)}(n) + \Delta\delta(n - n_s)\}$
(9) $\mathbf{x}^-(k) \leftarrow \mathfrak{I}\{\mathbf{y}^{(i)}(n) - \Delta\delta(n - n_s)\}$
(10) $\mathbf{G}^{(i)}(n_s) \leftarrow (1/N)(\|\mathbf{x}^+\|_1 - \|\mathbf{x}^-\|_1)$
(11)   **else** $\mathbf{G}^{(i)}(n_s) \leftarrow 0$
(12)   **end if**
(13) $\mathbf{y}^{(i+1)}(n_s) \leftarrow \mathbf{y}^{(i)}(n_s) - \mathbf{G}^{(i)}(n_s)$
(14)   **end for**
(15) $\beta_i = \arccos(\langle \mathbf{G}^{i-1} \mathbf{G}^i \rangle / \|\mathbf{G}^{i-1}\|_2^2 \|\mathbf{G}^i\|_2^2)$
(16)   **until** $\beta_i < 170°$
(17) $\Delta \leftarrow \Delta / \sqrt{10}$;
(18) $\Xi = 10\log_{10}(\sum_{n \in \mathbf{N}\setminus\mathbf{N}_M} |\mathbf{y}_p(n) - \mathbf{y}^{(i)}(n)|^2 / \sum_{n \in \mathbf{N}\setminus\mathbf{N}_M} |\mathbf{y}^{(i)}(n)|^2)$
(19) **until** $\Xi < \Xi_{\max}$
(20) **return** $\mathbf{y}^{(i)}(n)$
Reconstructed signal $\mathbf{x}_R(n) = \mathbf{y}^{(i)}(n)$

ALGORITHM 6: Adaptive gradient-based algorithm.

MSE = 13.85, much lower than MSE = 31.64 for the OMP algorithm.

Example 2. In this example, the reconstruction of images using adaptive gradient-based algorithm (Algorithm 6) is considered. The images are generally not strictly sparse in any domain and as such could be quite demanding for reconstruction. For that reason, as stated in Section 3.7, the adaptive gradient-based algorithm is employed, since it does not require the strict sparsity condition. As a suitable transform domain representation for natural image, the two-dimensional DCT ($\mathfrak{I} \rightarrow$ 2DDCT) is used. We might say that the image can be observed as approximately sparse in the 2DDCT domain.

The image with missing pixels is divided into 16×16 blocks, where 40% of the pixels are missing in each block (missing pixels are denoted in white in Figures 4(a), 4(c), and 4(e)). Hence, the algorithm is applied on a block-by-block basis where the measurement vector $\mathbf{y}$ is created for each observed image block using the available block pixels. The value of gradient step $\Delta = 128$ is used. The resulting reconstructed images are shown in Figures 4(b), 4(d), and 4(f).

The quality of the reconstructed image is measured using the structural similarity index (SSIM) [36] between the original image (with full set of pixels) and reconstructed image. The SSIM ranges between 0 and 1, 1 being identical and

0 being not similar. It is designed to improve the traditional metrics such as peak signal-to-noise ratio (PSNR) in order to be more consistent with the perception of the human visual system. The achieved values of SSIM for the test images in Figure 4 are SSIM = 0.9, SSIM = 0.94, and SSIM = 0.95.

Example 3. Let us assume that a radar signal consists of stationary (sinusoidal) components belonging to the rigid body and nonstationary components belonging to the disturbances caused by rotating target parts and noise. The Fourier transform and the STFT of the considered radar signal type are given in Figure 5.

The total signal length is 4096 samples. The STFT is calculated for the window width $w = 64$ samples (STFT is of size 64×64). The aim of this experiment is to extract the rigid body components which are presented as a set of stationary sinusoids.

The STFT values of the original full signal along columns and then the 75% of the largest values (possibly belonging to rotating components and noise) are set to zero or in other words discarded from the STFT plane. Also, a small percentage of the smallest values along the sorted columns are also set to zero with the aim of discarding small values belonging to noise. The remaining STFT values are unsorted and placed to their original positions in the STFT plane. These values represent a small amount of available measurements in the STFT domain and we are faced with the CS problem. In
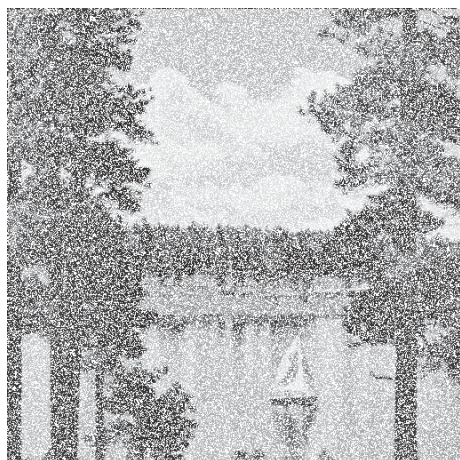
(a)
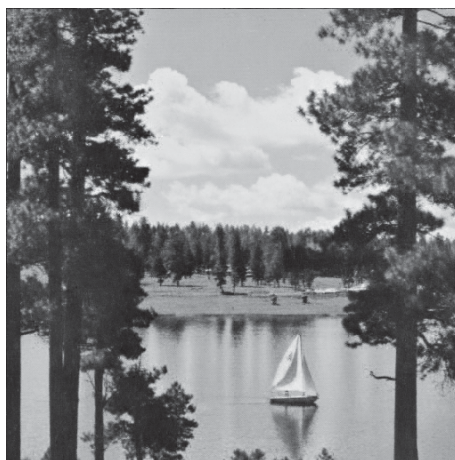
(b)

(c)

(d)

(e)

(f)

Figure 4: ((a), (c), and (e)) Available measurements/pixels (missing values are denoted by white pixels). ((b), (d), and (f)) Reconstructed images.
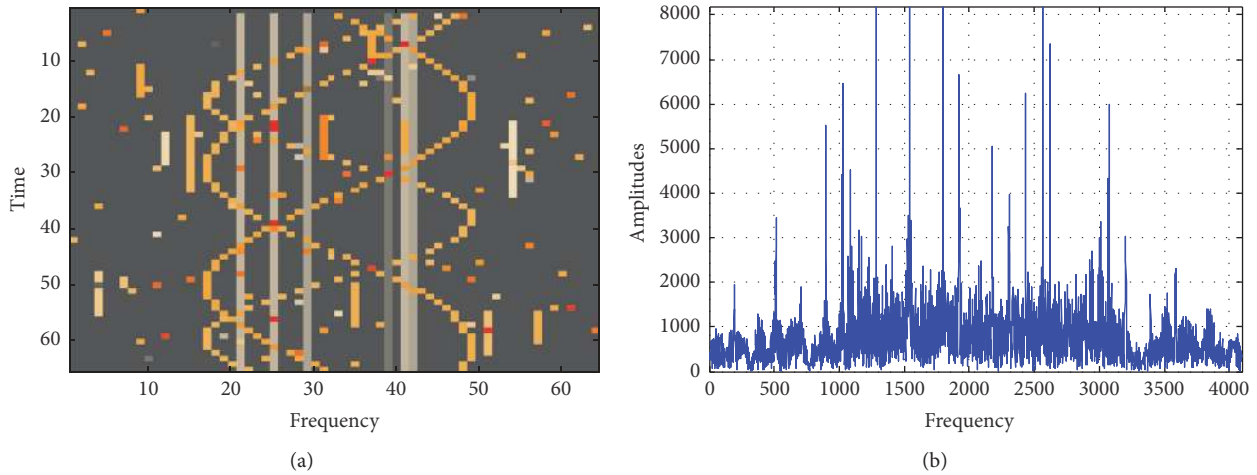
(a)

(b)

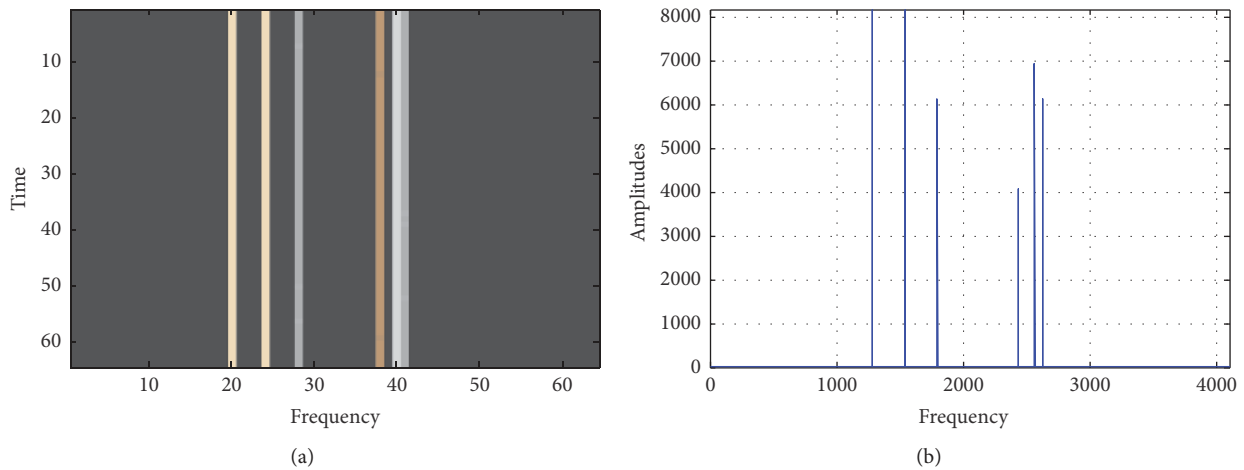FIGURE 5: (a) STFT of the observed signal; (b) DFT of the observed signal.

(a)

(b)

FIGURE 6: (a) STFT of the observed signal; (b) DFT of the observed signal.

Figure 6, we show the available measurements from the STFT after discarding all unwanted components.

In order to reconstruct the stationary rigid body components from the available measurements, the CS matrix is defined as partial combined transform matrix with rows corresponding to the available samples in the STFT (according to the procedure described in Section 4.4). The STFT of the reconstructed stationary components is given in Figure 6(a), while the corresponding DFT is given in Figure 6(b).

6. Conclusion

The paper reviews the fundamental concepts of compressive sensing theory, comprising the main requirements, conditions, and common optimization problem formulations. Several algorithms for signal reconstruction from incomplete measurements are summarized, allowing an insight into the diversity of approaches, the related complexity, assumptions, and efficiency. As a special contribution, the paper presents some interesting compressive sensing formulations for different types of signals and sparsity domains, such as the

Fourier transform domain, polynomial Fourier transform domain, Hermite transform, and time-frequency domain. Particularly, this part aims to reveal the possibilities and perspectives of using compressive sensing in different signal processing scenarios.

Competing Interests

The authors declare that they have no competing interests.
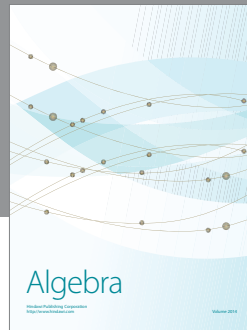
Acknowledgments

References

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

[2] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[3] Y. Eldar and G. Kutyniok, *Compressive Sensing Theory and Applications*, Cambridge University Press, 2012.

[4] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, Springer, New York, NY, USA, 2013.

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

[6] L. Stankovic, M. Dakovic, S. Stankovic, and I. Orovic, "Sparse signal processing," in *Digital Signal Processing*, L. Stankovic, Ed., CreateSpace, 2015.

[7] S. Stanković, I. Orović, and E. Sejdić, *Multimedia Signals and Systems: Basic and Advanced Algorithms for Signal Processing*, Springer, New York, NY, USA, 2016.

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

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

[10] G. Pope, *Compressive Sensing: a Summary of Reconstruction Algorithms*, Eidgenössische Technische Hochschule, Zürich, Switzerland, 2008.

[11] L. Stanković, M. Daković, and S. Vujović, "Adaptive variable step algorithm for missing samples recovery in sparse signals," *IET Signal Processing*, vol. 8, no. 3, pp. 246–256, 2014.

[12] S. Stanković, I. Orović, and L. Stanković, "An automated signal reconstruction method based on analysis of compressive sensed signals in noisy environment," *Signal Processing*, vol. 104, pp. 43–50, 2014.

[13] L. Stankovic, S. Stankovic, and M. Amin, "Missing samples analysis in signals for applications to L-estimation and compressive sensing," *Signal Processing*, vol. 94, no. 1, pp. 401–408, 2014.

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

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

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

[17] 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 Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[18] Y. Liu, I. Gligorijevic, V. Matic, M. De Vos, and S. Van Huffel, "Multi-sparse signal recovery for compressive sensing," in *Proceedings of the IEEE International Conference on Engineering in Medicine and Biology Society (EMBC '12)*, pp. 1053–1056, San Diego, Calif, USA, August-September 2012.

[19] Y. C. Eldar, *Sampling Theory: Beyond Bandlimited Systems*, Cambridge University Press, New York, NY, USA, 2015.

[20] P. Mächler, *VLSI Architectures for Compressive Sensing and Sparse Signal Recovery*, Hartung Gorre, Konstanz, Germany, 2013.

[21] M. F. Duarte and Y. C. Eldar, "Structured compressed sensing: from theory to applications," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4053–4085, 2011.

[22] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, Pa, USA, 1997.

[23] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: algorithms and applications," *IEEE Access*, vol. 3, pp. 490–530, 2015.

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

[25] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[26] M. A. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak, "Majorization-minimization algorithms for wavelet-based image restoration," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2980–2991, 2007.

[27] I. Selesnick, "Sparse signal restoration," *Connexions*, 2009, http://cnx.org/content/m32168/.

[28] S. Stanković, I. Orović, and L. Stanković, "Polynomial Fourier domain as a domain of signal sparsity," *Signal Processing*, vol. 130, pp. 243–253, 2017.

[29] S. Stanković, L. Stanković, and I. Orović, "Compressive sensing approach in the Hermite transform domain," *Mathematical Problems in Engineering*, vol. 2015, Article ID 286590, 9 pages, 2015.

[30] M. Brajović, I. Orović, M. Daković, and S. Stanković, "Gradient-based signal reconstruction algorithm in Hermite transform domain," *Electronics Letters*, vol. 52, no. 1, pp. 41–43, 2016.

[31] L. Stankoví, S. Stankoví, T. Thayaparan, M. Dakoví, and I. Oroví, "Separation and reconstruction of the rigid body and micro-Doppler signal in ISAR part II-statistical analysis," *IET Radar, Sonar and Navigation*, vol. 9, no. 9, pp. 1155–1161, 2015.

[32] L. Stanković, S. Stanković, T. Thayaparan, M. Daković, and I. Orović, "Separation and reconstruction of the rigid body and micro-Doppler signal in ISAR part I—theory," *IET Radar, Sonar & Navigation*, vol. 9, no. 9, pp. 1147–1154, 2015.

[33] A. Sandryhaila, S. Saba, M. Puschel, and J. Kovacevic, "Efficient compression of QRS complexes using Hermite expansion," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 947–955, 2012.

[34] A. Sandryhaila, J. Kovačević, and M. Püschel, "Compression of QRS complexes using Hermite expansion," in *Proceedings of the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '11)*, pp. 581–584, IEEE, Prague, Czech Republic, May 2011.

[35] MIT-BIH ECG Compression Test Database, http://www.physionet.org/physiobank/database/cdb.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.