IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 24, 2017, accepted September 10, 2017, date of publication September 22, 2017, date of current version October 12, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2754586

Efficient Recovery Algorithm for Discrete Valued Sparse Signals Using an ADMM Approach

NUNO M. B. SOUTO[1,2], (Senior Member, IEEE), AND HUGO ANDRÉ LOPES[1,2,3]
[1]ISCTE-University Institute of Lisbon, 1649-026 Lisboa, Portugal
[2]Instituto de Telecomunicações, 1049-001 Lisboa, Portugal
[3]NOS SGPS, 1600-404 Lisboa, Portugal

Corresponding author: Nuno Souto (nuno.souto@lx.it.pt)

This work was supported in part by FCT - Fundação para a Ciência e Tecnologia and Instituto de Telecomunicações under Project UID/EEA/50008/2013.

ABSTRACT Motivated by applications in wireless communications, in this paper we propose a reconstruction algorithm for sparse signals whose values are taken from a discrete set, using a limited number of noisy observations. Unlike conventional compressed sensing algorithms, the proposed approach incorporates knowledge of the discrete valued nature of the signal in the detection process. This is accomplished through the alternating direction method of the multipliers which is applied as a heuristic to decompose the associated maximum likelihood detection problem in order to find candidate solutions with a low computational complexity order. Numerical results in different scenarios show that the proposed algorithm is capable of achieving very competitive recovery error rates when compared with other existing suboptimal approaches.

INDEX TERMS Sparse signal recovery, discrete signal reconstruction, compressed sensing, generalized spatial modulations (GSM), large scale MIMO (LS-MIMO).

I. INTRODUCTION

In the last decade, the problem of reconstructing sparse signals from a reduced observation set and the associated compressed sensing (CS) paradigm [1] have attracted considerable research interest. Several techniques which take into account the sparsity property have been proposed for signal reconstruction [2], [3]. Amongst the different classes of algorithmic approaches, the two most popular ones are greedy pursuit methods and convex relaxation methods. In greedy pursuit approaches such as the orthogonal matching pursuit (OMP) algorithm [4], the set of nonzero elements of the signal is found iteratively using a greedy strategy. In convex relaxation methods the nonconvex sparsity objective (or constraint) is relaxed into a convex one, often based on the λ_1-norm, and the resulting convex problem is solved using an efficient convex optimization solver. A well-known example of this type of approach is the basis pursuit denoising (BPDN) formulation [5]. Although these different techniques can work well in multiple applications, there are circumstances where additional information about the signal exists, which can be exploited in order to improve the reconstruction method. This is the case of sparse signal vectors whose elements are taken from a discrete set. These signals are common in several wireless communications scenarios such as generalized spatial modulation multiple input multiple output (SM-MIMO) schemes [6], machine-type communications [7] and spectrum sensing for cognitive radios [8].

While one can apply a conventional CS algorithm and project the solution found onto the discrete set, the performance will be far from optimal unless knowledge of the discrete nature of the signal is directly taken into account inside the reconstruction method [9], [10]. A few algorithms that follow this approach have been proposed recently by several authors. In [11] the reconstruction problem is formulated both as a sphere decoding and as a semidefinite relaxation problem for which standard algorithms can be applied. However, depending on the problem size and noise level the decoding complexity can become high and the approach is limited to binary signals, i.e., with symbols taken from {0,1}. Relying on the adaptation of CS algorithms to discrete signals, Sparrer *et al.* [12] proposed the concatenation of a greedy algorithm like OMP to detect the support set and a sphere decoder (SD) to estimate the discrete symbols. In [13], the OMP was modified with the inclusion of an inner step where "soft" estimates for the symbols are computed taking into account the discrete nature and noise levels. However the authors assume the transmission of binary polar signals only. Sparrer and Fischer [14] propose an OMP algorithm modified with the inclusion of an estimation step based on the minimum mean squared error (MMSE) criterion. In [9] the OMP algorithm was modified in order to include a warm starting procedure where the discrete nature of the symbols is taken into account. The authors show that some performance improvements over the OMP algorithm can be achieved, depending on the noise level and number of measurements. In [15], a greedy algorithm named multipath matching pursuit with slicing (sMMP) was proposed where multiple promising candidates for minimizing the residual are evaluated and which includes an inner integer slicing step. The algorithm has a performance that depends on the number of child candidates per node expanded at each level of the tree search procedure and has a worst case complexity that grows exponentially with the sparsity of the signal. In [16], an adapted basis pursuit formulation was studied where box constraints are included in order to relax the nonconvex discrete valued signal restrictions. In [10], mixed integer linear programming solvers were applied for the recovery of integer signals but it was shown that the computational complexity of the approach is limited in practice to relatively small problems, concluding that further heuristics and approximation schemes are required.

In this paper we follow a different approach and propose an algorithm comprising a sequence of three different subproblems with simple solutions, namely an unconstrained quadratic problem, a projection onto a discrete set and a projection over the set of vectors with a predefined cardinality. This allows us to incorporate the knowledge of the discrete valued nature of the signal directly into the detection process. In order to split the main problem we apply the alternating direction method of the multipliers (ADMM) which is a well-known operator splitting method often adopted for convex optimization problems [17], [18]. Despite the lack of theoretical convergence guarantees (with a few exceptions addressed in recent studies [19], [20]), ADMM has also been used as a powerful heuristic for several nonconvex problems often achieving excellent performance [19]–[23]. While the ability of ADMM to decompose complex problems into simpler subproblems makes it particularly appealing for discrete valued sparse signal recovery, it does not seem to have been applied to this context. Due to the nonconvex nature of the problem, ADMM is applied as a heuristic in the sense that it is able to provide a solution faster than an exact method but it is not guaranteed to be an optimal point. As the solutions found may not be the exact ones we also propose two refinement steps for improving the solution candidates produced by the main algorithm. Numerical results in different wireless communications scenarios show that the proposed reconstruction approach can attain very competitive performance against other existing methods. It is important to note that while in this paper we address directly the maximum likelihood detection (MLD) formulation, which results in a splitting where the problem of dealing with a specific discrete set is confined to a projection step, there is a very different approach often employed to cope with discrete variables. It is based on the idea of binarization, i.e., mapping discrete variables to binary vectors and relaxing them to constrained continuous ones, as suggested in [24] and [25] within the scope of ADMM. While binarization allows the application of binary tailored methods to non-binary sets (as done in [26] regarding MIMO detection using semidefinite relaxation) the cost is an often higher decoding complexity due to the larger working matrices and optimization variables which result from the binary mapping. Furthermore binarization also restricts the formulation to finite sets. In this paper we address a more general formulation in the sense that it can be posed over finite or infinite discrete sets, such as an infinite lattice.

The paper is organized as follows. Section II presents the system model and formulates the discrete valued sparse signal reconstruction problem. Section III derives the proposed signal recovery algorithm while section IV discusses efficient implementations of the algorithm which explore special characteristics of the system matrix, focusing in a particular problem of interest in wireless communication. Section V presents simulation results of the proposed approach followed by the conclusions in Section VI.

Notation: Matrices and vectors are denoted by uppercase and lowercase boldface letters, respectively. The superscript $(\cdot)^H$ denotes the conjugate transpose of a matrix/vector, $\|\cdot\|_2$ is the 2-norm of a vector, $\|\cdot\|_0$ is its cardinality, $\sup p(\mathbf{x})$ returns the set of indices of nonzero elements in $\mathbf{x}$ (i.e., the support of $\mathbf{x}$), $\mathrm{diag}(\cdot)$ represents the vector containing the diagonal elements of a matrix and $\mathbf{I}_n$ is the $n \times n$ identity matrix.

II. SYSTEM MODEL AND PROBLEM STATEMENT

Let us consider a noisy observation process modelled as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n} \tag{1}$$

where $\mathbf{y} \in \mathbb{C}^{L\times 1}$ is the L-sized observation (received) vector, $\mathbf{H} \in \mathbb{C}^{L\times N}$ is the system (channel) matrix and $\mathbf{n} \in \mathbb{C}^{L\times 1}$ is the vector containing independent zero-mean circularly symmetric Gaussian noise samples with covariance $2\sigma^2\mathbf{I}_N$. The $N\times 1$ vector $\mathbf{s}$ represents the original sparse signal comprising $K \ll N$ nonzero elements drawn from a complex valued discrete set $\mathcal{A}$, which may be finite or not. Assuming an underdetermined scenario where $K \leq L < N$, the objective is then to reconstruct $\mathbf{s}$ exploiting its sparsity and discrete valued nature. The MLD for the original signal can be formulated as

$$\min_{\mathbf{s}} f(\mathbf{s}) \triangleq \|\mathbf{y} - \mathbf{H}\mathbf{s}\|_2^2 \tag{2}$$

$$\text{subject to } \mathbf{s} \in \mathcal{A}_0^N \tag{3}$$

$$\|\mathbf{s}\|_0 = K. \tag{4}$$

where $\mathcal{A}_0 \overset{def}{=} \mathcal{A}\cup\{0\}$. Note that constraints (3) and (4), which restrict the estimated signal to be discrete and restrict its cardinality, clearly make the optimization problem nonconvex.

III. DISCRETE SPARSE DETECTOR ALGORITHM

A. ALGORITHM DESCRIPTION

In the following we apply ADMM as a heuristic to solve the MLD formulation and estimate the original signal. First we rewrite problem (2)-(4) as

$$\min_{\mathbf{s},\mathbf{x},\mathbf{z}} \ \|\mathbf{y}-\mathbf{H}\mathbf{s}\|_2^2 + I_{\mathcal{C}}(\mathbf{x}) + I_{\mathcal{A}_0^N}(\mathbf{z}) \quad (5)$$

$$\text{subject to } \mathbf{s}=\mathbf{x} \quad (6)$$

$$\mathbf{s}=\mathbf{z}. \quad (7)$$

where $\mathcal{C}=\{\mathbf{x}:\|\mathbf{x}\|_0=K\}$ and $I_{\mathcal{D}}(\mathbf{v})$ is the indicator function which returns 0 if $\mathbf{v}\in\mathcal{D}$ and $+\infty$ otherwise. We can then write the augmented Lagrangian as

$$\begin{aligned} L_{\rho_x,\rho_z}(\mathbf{s},\mathbf{x},\mathbf{z},\boldsymbol{\nu},\boldsymbol{\lambda}) &= \|\mathbf{y}-\mathbf{H}\mathbf{s}\|_2^2 + I_{\mathcal{C}}(\mathbf{x}) + I_{\mathcal{A}_0^N}(\mathbf{z}) \\ &\quad + 2\mathrm{Re}\left\{\boldsymbol{\nu}^H(\mathbf{s}-\mathbf{x}) + \boldsymbol{\lambda}^H(\mathbf{s}-\mathbf{z})\right\} \\ &\quad + \rho_x\|\mathbf{s}-\mathbf{x}\|_2^2 + \rho_z\|\mathbf{s}-\mathbf{z}\|_2^2 \end{aligned} \quad (8)$$

where $\boldsymbol{\nu},\boldsymbol{\lambda}\in\mathbb{C}^{N\times 1}$ are the dual variables and ρ_x, ρ_z are the penalty parameters for constraints (6) and (7). We can use scaled variables, $\mathbf{u}=\boldsymbol{\nu}/\rho_x$ and $\mathbf{w}=\boldsymbol{\lambda}/\rho_z$, and rewrite (8) as

$$\begin{aligned} L_{\rho_x,\rho_z}(\mathbf{s},\mathbf{x},\mathbf{z},\boldsymbol{\nu},\boldsymbol{\lambda}) &= \|\mathbf{y}-\mathbf{H}\mathbf{s}\|_2^2 + I_{\mathcal{C}}(\mathbf{x}) + I_{\mathcal{A}_0^N}(\mathbf{z}) \\ &\quad + \rho_x\left(\|\mathbf{s}+\mathbf{u}-\mathbf{x}\|_2^2 - \|\mathbf{u}\|_2^2\right) \\ &\quad + \rho_z\left(\|\mathbf{s}+\mathbf{w}-\mathbf{z}\|_2^2 - \|\mathbf{w}\|_2^2\right). \end{aligned} \quad (9)$$

The application of gradient ascent to the dual problem results in a series of iterative steps involving the independent minimization of the augmented Lagrangian over primal variables $\mathbf{s}$, $\mathbf{x}$ and $\mathbf{z}$ followed by the updates of the scaled dual variables $\mathbf{u}$ and $\mathbf{w}$:

$$\mathbf{s}^{t+1} = \min_{\mathbf{s}} L_{\rho_x,\rho_z}(\mathbf{s},\mathbf{x}^t,\mathbf{z}^t,\mathbf{u}^t,\mathbf{w}^t) \quad (10)$$

$$\mathbf{x}^{t+1} = \min_{\mathbf{x}} L_{\rho_x,\rho_z}(\mathbf{s}^{t+1},\mathbf{x},\mathbf{z}^t,\mathbf{u}^t,\mathbf{w}^t) \quad (11)$$

$$\mathbf{z}^{t+1} = \min_{\mathbf{z}} L_{\rho_x,\rho_z}(\mathbf{s}^{t+1},\mathbf{x}^{t+1},\mathbf{z},\mathbf{u}^t,\mathbf{w}^t) \quad (12)$$

$$\mathbf{u}^{t+1} = \mathbf{u}^t + \mathbf{s}^{t+1} - \mathbf{x}^{t+1}. \quad (13)$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \mathbf{s}^{t+1} - \mathbf{z}^{t+1}. \quad (14)$$

Therefore, the original problem has been decomposed into a sequence of smaller subproblems which are easier to solve. Concerning step (10), using $\nabla_{\mathbf{s}^H} L_{\rho_x,\rho_z}(\mathbf{s},\mathbf{x}^t,\mathbf{z}^t,\boldsymbol{\nu}^t,\boldsymbol{\lambda}^t)=0$, $\mathbf{s}^{t+1}$ can be found as the solution of

$$\begin{aligned} &\left(\mathbf{H}^H\mathbf{H} + (\rho_x+\rho_z)\,\mathbf{I}_N\right)\mathbf{s}^{t+1} \\ &\quad = \left(\mathbf{H}^H\mathbf{y} + \rho_x\left(\mathbf{x}^t-\mathbf{u}^t\right) + \rho_z\left(\mathbf{z}^t-\mathbf{w}^t\right)\right). \end{aligned} \quad (15)$$

The $\mathbf{x}$ and $\mathbf{z}$ update steps are obtained as ([17], ch. 9)

$$\mathbf{x}^{t+1} = \prod\nolimits_{\mathcal{C}}\left(\mathbf{s}^{t+1}+\mathbf{u}^t\right) \quad (16)$$

$$\mathbf{z}^{t+1} = \prod\nolimits_{\mathcal{A}_0^N}\left(\mathbf{s}^{t+1}+\mathbf{w}^t\right) \quad (17)$$

where $\prod_{\mathcal{D}}(\cdot)$ denotes the projection onto set $\mathcal{D}$. While (17) can be computed by rounding each entry to the closest element in $\mathcal{A}_0$, (16) can be performed by zeroing the $N-K$ smallest magnitude elements. Note that the latter approach does not implement an exact projection over $\mathcal{C}$ but over the set of vectors with bounded cardinality ($\mathcal{C}'=\{\mathbf{x}:\|\mathbf{x}\|_0\leq K\}$). However the difference arises only in the extreme cases where the argument has less than K nonzero elements, which are difficult to handle with an exact projection.

TABLE 1. Discrete valued sparse ADMM algorithm.

Input: $\mathbf{u}^0$, $\mathbf{w}^0$, $\mathbf{x}^0$, $\mathbf{z}^0$, $\mathbf{H}$, $\mathbf{y}$, ρ_x, ρ_z, Q
$f_{best}=\infty$.
for t=0,1,…Q-1 **do**
  Solve (15) for $\mathbf{s}^{t+1}$.
  $\mathbf{x}^{t+1} \leftarrow \prod_{\mathcal{C}}\left(\mathbf{s}^{t+1}+\mathbf{u}^t\right)$.
  $I \leftarrow \mathrm{supp}\left(\mathbf{x}^{t+1}\right)$.
  $\mathbf{z}^{t+1} \leftarrow \prod_{\mathcal{A}_0^N}\left(\mathbf{s}^{t+1}+\mathbf{w}^t\right)$.
  $\hat{\mathbf{s}}_{\bar{I}}^{candidate} \leftarrow 0$, $\hat{\mathbf{s}}_{I}^{candidate} \leftarrow \prod_{\mathcal{A}^K}\left(\mathbf{x}_I^{t+1}\right)$
  if $f\left(\hat{\mathbf{s}}^{candidate}\right) < f_{best}$ **then**
    $\hat{\mathbf{s}} \leftarrow \hat{\mathbf{s}}^{candidate}$.
    $f_{best} = f\left(\hat{\mathbf{s}}^{candidate}\right)$.
  end if
  $\mathbf{u}^{t+1} \leftarrow \mathbf{u}^t + \mathbf{s}^{t+1} - \mathbf{x}^{t+1}$.
  $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t + \mathbf{s}^{t+1} - \mathbf{z}^{t+1}$.
end for
Output: $\hat{\mathbf{s}}$.

Table 1 summarizes the steps of the proposed discrete valued sparse ADMM (DVS-ADMM) based algorithm, where $\hat{\mathbf{s}}$ is the estimate of the transmitted symbols, Q is the number of iterations, $I=\mathrm{supp}\left(\mathbf{x}^{t+1}\right)$, $\bar{I}=\{1,\ldots,N\}\backslash I$, $\mathbf{H}_I$ is the $L\times K$ reduced system matrix containing the columns of $\mathbf{H}$ indexed by I and $\hat{\mathbf{s}}_I^{candidate}$ is the reduced $K\times 1$ vector whose components are the nonzero elements of $\hat{\mathbf{s}}^{candidate}$.

B. SOLUTION IMPROVEMENT

Due to the nonconvex nature of the feasible set, ADMM is applied as an heuristic and, thus, there are no guarantees that the derived algorithm converges to the global minimum. In order to improve the generated solution candidates we propose the use of two additional heuristic solution refinement steps.

First, we can explore points which differ only in two nonzero positions from $\mathbf{x}^{t+1}$ (with all the nonzero elements obtained from $\mathbf{s}^{t+1}+\mathbf{u}^t$), i.e., points belonging to the set

$$\begin{aligned} \mathcal{C}^{neigh}\left(\mathbf{x}^{t+1}\right) = \Big\{\mathbf{y}:\|\mathbf{y}\|_0=K,\ \left\|\mathbf{x}^{t+1}-\mathbf{y}\right\|_0=2, \\ \mathbf{y}_{\widehat{I}}=\mathbf{s}_{\widehat{I}}^{t+1}+\mathbf{u}_{\widehat{I}}^t,\ \widehat{I}=\mathrm{supp}(\mathbf{y})\Big\}, \end{aligned} \quad (18)$$

and check if they result in a smaller objective value (2) after projection over $\mathcal{A}_0$. Points in (18) are close neighbors in the sense that the respective support sets differ in two elements (the minimum possible for a fixed cardinality).

Secondly, we can solve the minimization problem (2) over a restricted convex relaxation where the support set fixed and constraints (3) and (4) removed. Note that since the restricted problem is basically equivalent to a MIMO detection problem, instead of solving the resulting unconstrained quadratic problem (2) we can apply conventional MIMO detection methods such as a projected MMSE estimate which can be written as

$$\hat{\mathbf{s}}_I^{candidate} = \prod_{\mathcal{A}^K}\left(\left(\mathbf{H}_I^H\mathbf{H}_I + 2\sigma^2\mathbf{I}_K\right)^{-1}\mathbf{H}_I^H\mathbf{y}\right). \quad (19)$$

The objective function is then evaluated using the generated solution in order to verify if an improved candidate point was obtained. Both refinement steps can be applied in every iteration or in the last iterations only.

C. INITIALIZATION AND PENALTY PARAMETERS

The operation of the DVS-ADMM algorithm entails the selection of initial points $\mathbf{x}^0$, $\mathbf{z}^0$, $\mathbf{u}^0$ and $\mathbf{w}^0$ as well as of the penalty parameters ρ_x and ρ_z. Due to the nonconvex nature of the problem being solved, both features can have an important impact on the quality of the approximate solution found as there are no guarantees that ADMM converges to the optimal points [17]. These parameters are still active areas of research not only for nonconvex problems [23], [27] but also for convex ones [28]. In the proposed algorithm, ρ_x and ρ_z, which balance the weight between squared error minimization and higher penalization on constraint violations in (8), are used as tuning parameters and can be selected through numerical evaluation for the specific problem at hand.

Concerning the initialization, we propose the use of either a warm or a random start. In the warm initialization procedure the idea is to start with a good candidate vector. We adopt a metric ordering criterion similar to the one proposed in [29] where the received vector is correlated against each measurement matrix column, normalized and sorted according to the resulting squared norm. In our case we only retain the first candidate combination of nonzero positions. The procedure starts with the computation of the support set

$$I = \text{supp}\left(\prod_{C}\left(\mathbf{y}^H\mathbf{H}\left(diag\left(\mathbf{H}^H\mathbf{H}\right)\right)^{-2}\mathbf{H}^H\mathbf{y}\right)\right), \quad (20)$$

followed by $\mathbf{x}^0 = \mathbf{z}^0 = \hat{\mathbf{s}}^{candidate}$, where $\hat{\mathbf{s}}_I^{candidate}$ is obtained using (19) and $\hat{\mathbf{s}}_{\bar{I}}^{candidate} = 0$. The dual variables are initialized as In the case of random start, we simply adopt for the initial vector, $\hat{\mathbf{s}}^{candidate}$, a randomly selected one with elements constrained to the interval defined by the constellation limits, i.e., $\hat{s}_i^{candidate} \in [\min\{\mathcal{A}_0\}, \max\{\mathcal{A}_0\}]$. The initial values of the required variables are then obtained from $\mathbf{x}^0 = \prod_C\left(\hat{\mathbf{s}}^{candidate}\right)$, $\mathbf{z}^0 = \prod_{\mathcal{A}_0^N}\left(\hat{\mathbf{s}}^{candidate}\right)$, $\mathbf{u}^0 = \hat{\mathbf{s}}^{candidate} - \mathbf{x}^0$ and $\mathbf{w}^0 = \hat{\mathbf{s}}^{candidate} - \mathbf{z}^0$. Note that instead of running the algorithm only once with a large number of iterations we can apply it multiple times with different initializations and a lower number of iterations.

The cardinality restriction combined with the discrete-valued nature of the feasible set makes it difficult to find theoretical convergence conditions and guarantees to the global minimum or even to stationary points for DVS-ADMM. Therefore, while a stopping criterion based on the primal and dual residuals (as explained in [17]) could be applied, there are no guarantees that both converge to 0 and, thus, a maximum number of iterations Q has to be adopted to ensure the termination of the algorithm. Still, if stagnation is detected in the progress of the algorithm, such as variables with negligible change after several iterations or arriving at previous values, then this can be used for earlier termination.

D. COMPLEXITY

As stated in the previous subsection, due to the nonconvex nature of the specific problem being addressed there are no theoretical convergence guaranties that the proposed DVS-ADMM algorithm will converge. Therefore, the overall computational complexity depends directly on the maximum number of iterations Q. Regarding the contribution of the individual steps, the $\mathbf{s}$-update (15) is the step with the highest complexity as it involves an $N \times N$ matrix inversion with a cost of $O\left(N^3\right)$. However, since the inverse remains the same after the first iteration, it only needs to be computed once for each matrix $\mathbf{H}$ instance and only vector sums and matrix-vector multiplications with a cost of $O\left(N^2\right)$ have to be performed in the remaining iterations. There is also one matrix-vector multiplication in step (15) (the term $\mathbf{H}^H\mathbf{y}$) which only needs to be computed in the first iteration and has a cost of $O(NL)$. As for the other steps of the algorithm, they consist mostly of vector sums or simple element wise operations over vectors. Only the update of the objective function in step 9 involves a matrix-vector multiplication with a complexity of $O(QLK)$. As such, keeping only the dominant terms, the total complexity order of the algorithm is $O\left(N^3 + NL + QN^2 + QLK\right)$. For comparison, the complexity order of MLD is $O\left(\binom{N}{K}M^K\right)$, where $\binom{N}{K}$ denotes number of combinations of N symbols taken K at a time. OMP/SD [12] is a suboptimal alternative iterative method, which consists in the conventional OMP, where a new element is added to the support set in every iteration, followed by a final application of an SD to estimate the discrete symbols at the positions of the support set. The complexity order is $O\left(NL + K^2L + K^3 + M^{\gamma K}\right)$ where $\gamma \in (0, 1]$ is a factor depending on the signal-to-noise ratio [30]. Q-OMP (from [12]) is an iterative method also based on the conventional OMP but with an additional quantization step in every iteration to select a discrete symbol for the added support position. Its complexity order is $O\left(NL + K^2L + K^3\right)$. sMMP [15] extends the approach of Q-OMP but, in each iteration, explores multiple promising candidate elements for the support set, choosing in the end the best overall candidate in terms of residual minimization. In this case the complexity is $O\left(NLP^{K-1} + \left(K^2L + K^3\right)P^K\right)$, where P is the number of child candidates expanded at each iteration. As another alternative, BPDN [5] solves a convex relaxation of the problem based on λ_1 minimization which can be written as a quadratic program and solved using an iterative interior point method. In the end the solution is projected over the discrete set. Its complexity is $O\left(N^{3.5}\right)$.

IV. EFFICIENT IMPLEMENTATIONS

As stated in section III-D, the step of the proposed algorithm with the highest complexity is the **s**-update (15) and, thus, it is the one whose simplification can have a larger impact on the complexity cost. There are several applications where particular characteristics of matrix **H** can be exploited so as to allow an efficient computation of the inverse required in (15). A discussion about efficient computations of a similar inverse was provided in [31]. In fact, if we rewrite (15) as

$$\mathbf{s}^{t+1} = \left(\mathbf{H}^H\mathbf{H} + (\rho_x + \rho_z)\,\mathbf{I}_N\right)^{-1}\mathbf{r}^t \tag{21}$$

where

$$\mathbf{r}^t = \mathbf{H}^H\mathbf{y} + \rho_x\left(\mathbf{x}^t - \mathbf{u}^t\right) + \rho_z\left(\mathbf{z}^t - \mathbf{w}^t\right), \tag{22}$$

it becomes clear that the **s**-update has the same form of equation (35) from [31]. Therefore all the efficient computation methods described in that paper for solving equation (35) for different matrix types can be directly applied to the s-update step of our algorithm. Those methods can lower the complexity order of the inverse computation from $O\left(N^3\right)$ to $O\left(N\log N\right)$ or even $O\left(N\right)$, depending on the particular structure of **H**.

Besides the cases addressed in [31], there are problems of interest from wireless communications, namely in the context of cyclic prefixed MIMO single carrier transmissions in time dispersive channels [32], whose formulations can fit (2)-(4) and allow a simplification of the proposed algorithm. In these cases, matrix **H** has a block circulant structure with P non circulant blocks (which is different from the cases addressed in [31]) and can be written as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_{P-1} & \cdots & \mathbf{H}_1 \\ \mathbf{H}_1 & \mathbf{H}_0 & \cdots & \mathbf{H}_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{P-1} & \mathbf{H}_{P-2} & \cdots & \mathbf{H}_0 \end{bmatrix}, \tag{23}$$

where $\mathbf{H}_n$, $n = 0, \ldots, P-1$, are $T \times W$ submatrices, with $T = L/P$ and $W = N/P$. Due to its block circulant structure, **H** can be factorized as

$$\mathbf{H} = \left(\mathbf{F}^H \otimes \mathbf{I}_T\right)\boldsymbol{\Lambda}\left(\mathbf{F}\otimes\mathbf{I}_W\right), \tag{24}$$

where **F** is the unitary $N \times N$ discrete Fourier transform (DFT) matrix and $\boldsymbol{\Lambda}$ is block diagonal, i.e., it can be written as

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_0 & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Lambda}_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \boldsymbol{\Lambda}_{P-1} \end{bmatrix} \tag{25}$$

Submatrices $\boldsymbol{\Lambda}_l$, with $l = 0, \ldots, P-1$, are defined as

$$\boldsymbol{\Lambda}_l = \begin{bmatrix} \sum_{n=0}^{P-1} h_{1,1}^n \omega^{ln} & \cdots & \sum_{n=0}^{P-1} h_{1,W}^n \omega^{ln} \\ \vdots & \ddots & \vdots \\ \sum_{n=0}^{P-1} h_{T,1}^n \omega^{ln} & \cdots & \sum_{n=0}^{P-1} h_{T,W}^n \omega^{ln} \end{bmatrix}, \tag{26}$$

where ω is a P^{th} primitive root of unity ($\omega = \exp\left(-j2\pi/P\right)$) and $h_{i,w}^n$ is the element in position (i,w) of submatrix $\mathbf{H}_n$. Note that all the elements of submatrices $\boldsymbol{\Lambda}_l$ are obtained from $T \cdot WP$-sized DFTs applied to the first block column of **H**. Therefore, using a fast Fourier transform (FFT) algorithm, factorization (24) has a complexity cost of $O\left(TWP\log P\right)$. This factorization can be introduced into (21)-(22) leading to the following **s**-update expression

$$\mathbf{s}^{t+1} = \left(\mathbf{F}^H \otimes \mathbf{I}_W\right)\left(\boldsymbol{\Lambda}^H\boldsymbol{\Lambda} + (\rho_x + \rho_z)\,\mathbf{I}_N\right)^{-1}\mathbf{R}^t \tag{27}$$

where

$$\begin{aligned}\mathbf{R}^t = {} & \boldsymbol{\Lambda}^H\left(\mathbf{F}\otimes\mathbf{I}_T\right)\mathbf{y} \\ & + \left(\mathbf{F}\otimes\mathbf{I}_W\right)\left(\rho_x\left(\mathbf{x}^t - \mathbf{u}^t\right) + \rho_z\left(\mathbf{z}^t - \mathbf{w}^t\right)\right).\end{aligned} \tag{28}$$

This form can reduce the complexity cost by taking into account the block diagonal structure of $\boldsymbol{\Lambda}$ which allows the products $\boldsymbol{\Lambda}^H\boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda}^H\mathbf{Y}$ (with $\mathbf{Y} = \left(\mathbf{F}\otimes\mathbf{I}_T\right)\mathbf{y}$) to be computed with a cost of $O\left(NWT\right)$ and $O\left(NT\right)$. The inverse in (27) also benefits from the block diagonal structure and can be obtained using (29), as shown at the bottom of this page. This means that instead of inverting a large $N \times N$ matrix it is only required to perform P smaller $W \times W$ matrix inversions with a total cost of $O\left(PW^3\right)$. Regarding the matrix-vector products involving **F** and $\mathbf{F}^H$, these can be computed with an FFT algorithm with a cost of $O\left(P\log P\right)$. The complexity order of the s-update step can thus be lowered from $O\left(N^3 + QN^2\right)$ to $O\left(N^3/P^2 + QN^2/P + N\left(L/P + Q\right)\log P\right)$.

$$\left(\boldsymbol{\Lambda}^H\boldsymbol{\Lambda} + (\rho_x + \rho_z)\,\mathbf{I}_N\right)^{-1} = \begin{bmatrix} \left(\boldsymbol{\Lambda}_0^H\boldsymbol{\Lambda}_0 + (\rho_x + \rho_z)\,\mathbf{I}_W\right)^{-1} & & 0 \\ & \ddots & \\ 0 & & \left(\boldsymbol{\Lambda}_{P-1}^H\boldsymbol{\Lambda}_{P-1} + (\rho_x + \rho_z)\,\mathbf{I}_W\right)^{-1} \end{bmatrix} \tag{29}$$

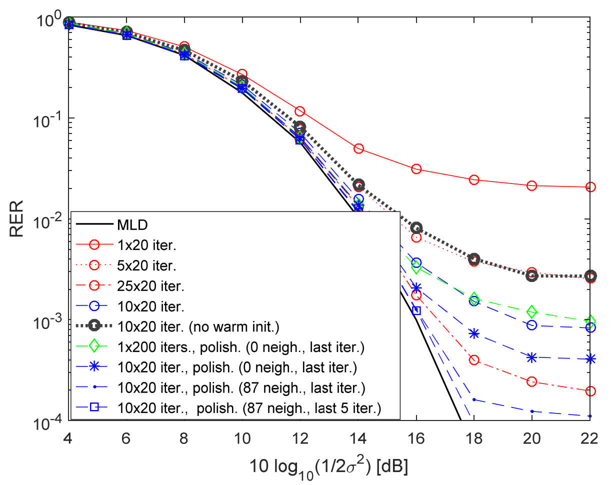


FIGURE 1. DVS-ADMM performance for $N = 32$, $L = 12$, $K = 3$ and 64-QAM.

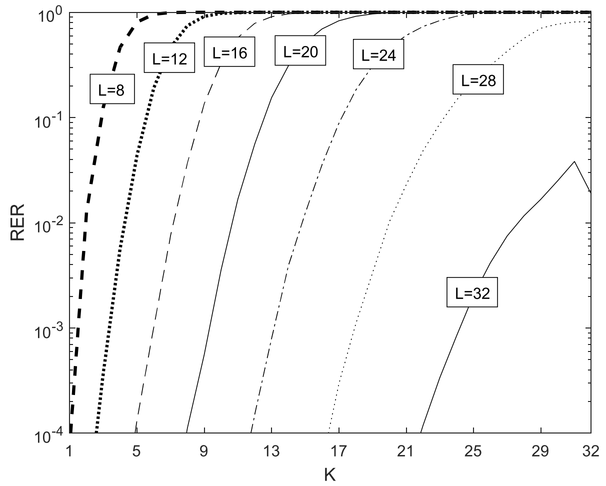


FIGURE 2. RER versus K, for $N = 32$, 64-QAM and $10\log_{10}\left(1/2\sigma^2\right) = 20$ dB.

V. NUMERICAL RESULTS

In this section we evaluate the performance of the proposed DVS-ADMM algorithm in terms of signal vector recovery error rate, defined as RER $= \Pr ob\left\{\hat{\mathbf{s}} \neq \mathbf{s}\right\}$, using Monte Carlo simulations We consider a wireless communications scenario with K random concurrent transmissions, where $\mathbf{H}$ corresponds to the channel matrix whose elements are independently drawn according to a complex Gaussian distribution $\mathcal{CN}(0, 1)$. The nonzero elements in $\mathbf{s}$ are randomly selected from a 64-QAM constellation with $E\left[|s_i|^2\right] = 1$. The simulations were run using Matlab Release 2016a on a 3.60 GHz Intel i7 machine with 8 threads. Each RER point was computed after running 600000 trials (6000 for RERs above 1%).

Fig. 1 shows the RER results for different configurations of DVS-ADMM as a function of the noise level for $N = 32$, $L = 12$ and $K = 3$. The values applied for the penalty parameters were $\rho_x = \rho_z = 2.5$ as these were found (through numerical evaluations) to result in good recovery performances for small problem sizes. In the legend $n_1 \times n_2$ means that n_1 initializations were employed with $Q = n_2$ iterations. As a reference we also include the MLD curve (obtained using a "brute-force" approach where each possible support set for the target sparsity is tested with a SD from [33]). Looking at the results it can be observed that running the algorithm with multiple initializations and a moderate number of iterations instead of a single initialization and a large number of iterations clearly benefits the recovery rate, especially when polishing is also applied. Furthermore, the performance clearly improves if one of the initializations is based on the warm procedure described in section III-C.

Regarding the use of polishing, its impact is more visible in the region of low noise levels where it is able to lower the irreducible RER floor. Even when only one of the polishing steps is applied, namely the restricted MMSE candidate (19) in the last iteration, evident performance improvements can be obtained. These gains become larger when both polishing steps are applied to more than one iteration. It is important to highlight that, even though Fig. 1 corresponds to a difficult scenario where the size of the observation set is considerably smaller than the signal vector (37.5%), the proposed algorithm is still able to achieve RERs below 0.1% and perform close to the MLD for high noise levels.

Considering a noise level of $10\log_{10}\left(1/2\sigma^2\right) = 20$ dB, Fig. 2 illustrates the impact of sparsity on the performance of the DVS-ADMM algorithm in a 10×20 configuration, for different sizes of the observation vector. The only polishing applied relied on the computation and check of the restricted MMSE candidate (19) in the last iteration. It is visible that the detector is able to achieve very low error rates for sparse signals even when the number of available measurements is low. For example, it can reach a RER below 0.01% when $K = 1$ and $L = 8$. We can also observe that the detector is very flexible as it can cope with a wide range of sparsity levels when the number of measurements increases, achieving excellent performances also for non-sparse signals.

Fig. 3 and Fig. 4 present the performance of DVS-ADMM (with 5 initializations, $Q = 20$ and a simple final polishing step relying on the restricted MMSE candidate (19)) in two different scenarios and compare it against other existing algorithms, namely MLD, BPDN (using box constraints as in [16] and with projection of the solution over the discrete set), sMMP with 4 child candidates per node, Q-OMP and OMP/SD (both from [12]). All the tested algorithms (including ours) were directly implemented without any special speedup optimizations. Note that for the larger scenario we applied $\rho_x = \rho_z = 75$ as it was observed that higher penalties allow better results in large problems. No MLD or sMMP results are included in Fig. 4 due to excessive runtimes in large problem settings. In this case, we included an alternative "Oracle SD" curve which corresponds to an SD which knows the support set, thus acting as a lower bound. In both scenarios it can be observed that the proposed DVS-ADMM clearly achieves the best results, performing close to the MLD in

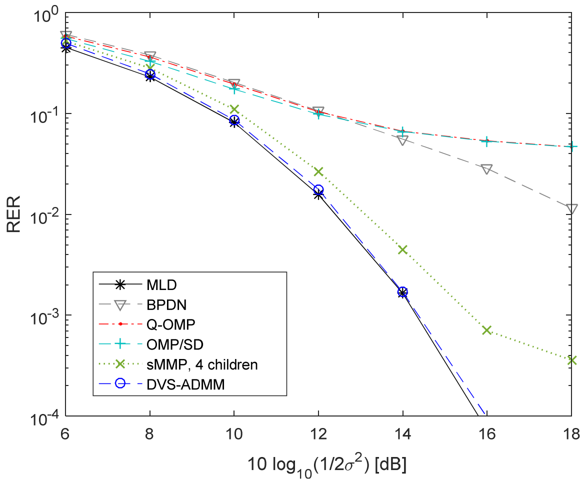


FIGURE 3. RER performance comparison for 64-QAM with $N = 32$, $L = 16$, $K = 3$.

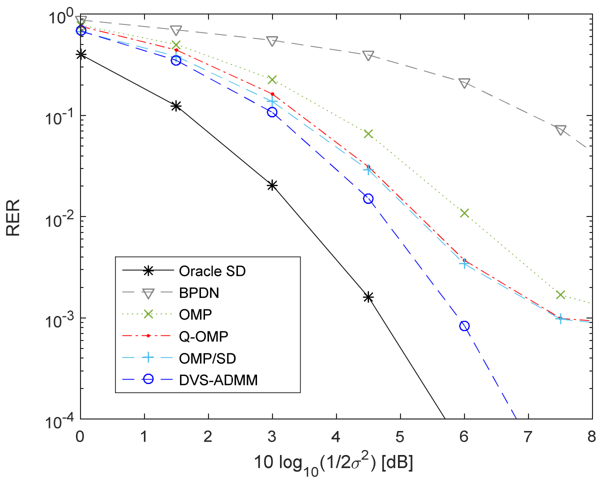


FIGURE 4. RER performance comparison for 64-QAM with $N = 256$, $L = 128$, $K = 20$.

TABLE 2. Average Running times per recovered vector (miliseconds).

Scenario	DVS-ADMM	sMMP	BPDN	Q-OMP	OMP/SD
N=32, L=16, K=3 (Fig. 3)	1.46	3.19	5.6	0.12	0.14
N=256, L=128, K=20 (Fig. 4)	32.7	-	1180	10	10.4

Fig. 3 and showing an almost constant gap around 1.2 dB to the ideal Oracle SD in Fig. 4.

As a final comparison between the different methods, Table 2 presents the program running times per recovered vector measured in the simulations of Fig. 3 and Fig. 4. The times shown in the table include all the steps, iterations and initializations applied in each method. While OMP based algorithms were always the fastest, the performances obtained were clearly worse than DVS-ADMM. The alternative method achieving results closer to DVS-ADMM was sMMP for the small scenario but besides having a greater runtime, in the large problem setting it was excessively slow making it unviable to obtain RER results.

VI. CONCLUSIONS

In this paper a new approach has been described for a common problem in wireless communications which concerns the recovery of discrete valued sparse signals from a limited set of noise corrupted observations. By applying ADMM to the recovery problem, we derived an algorithm where the different tasks of unconstrained squared error minimization problem, cardinality matching and discrete symbol decisions can be independently applied and naturally concatenated. Numerical results in several scenarios showed that the proposed algorithm can outperform other existing suboptimal approaches with a very competitive computational complexity.

REFERENCES

[1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[2] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proc. IEEE*, vol. 98, no. 6, pp. 948–958, Jun. 2010.
[3] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing* (Applied and Numerical Harmonic Analysis). Berlin, Germany: Birkhäuser, 2013.
[4] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Annu. Asilomar Conf. Signals, Syst. Comput.*, Nov. 1993, pp. 40–44.
[5] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.
[6] L. Xiao *et al.*, "Efficient compressive sensing detectors for generalized spatial modulation systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1284–1298, Feb. 2017.
[7] B. Knoop, F. Monsees, C. Bockelmann, D. Peters-Drolshagen, S. Paul, and A. Dekorsy, "Compressed sensing K-best detection for sparse multi-user communications," in *Proc. 22nd Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2014, pp. 1726–1730.
[8] E. Axell, G. Leus, E. G. Larsson, and H. V. Poor, "Spectrum sensing for cognitive radio: State-of-the-art and recent advances," *IEEE Signal Process. Mag.*, vol. 29, no. 3, pp. 101–116, May 2012.
[9] A. Flinth and G. Kutyniok, "Promp: A sparse recovery approach to lattice-valued signals," *Appl. Comput. Harmon. Anal.*, to be published.
[10] J. H. Lange, M. E. Pfetsch, B. M. Seib, and A. M. Tillmann. (Aug. 2016). "Sparse recovery with integrality constraints." [Online]. Available: https://arxiv.org/abs/1608.08678
[11] Z. Tian, G. Leus, and V. Lottici, "Detection of sparse signals under finite-alphabet constraints," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process (ICASSP)*, Apr. 2009, pp. 2349–2352.
[12] S. Sparrer and R. F. H. Fischer, "Adapting compressed sensing algorithms to discrete sparse signals," in *Proc. Workshop Smart Antennas (WSA)*, Erlangen, Germany, Mar. 2014, pp. 1–8.
[13] S. Sparrer and R. F. H. Fischer, "Soft-feedback OMP for the recovery of discrete-valued sparse signals," in *Proc. 23rd Eur. Signal Process. Conf. (EUSIPCO)*, Nice, France, Aug. 2015, pp. 1461–1465.
[14] S. Sparrer and R. F. H. Fischer, "MMSE-based version of OMP for recovery of discrete-valued sparse signals," *Electron. Lett.*, vol. 52, no. 1, pp. 75–77, Jan. 2016.
[15] B. Shim, S. Kwon, and B. Song, "Sparse detection with integer constraint using multipath matching pursuit," *IEEE Commun. Lett.*, vol. 18, no. 10, pp. 1851–1854, Oct. 2014.

[16] S. Keiper, G. Kutyniok, D. G. Lee, and G. E. Pfander. (Sep. 2016). "Compressed sensing for finite-valued signals." [Online]. Available: https://arxiv.org/abs/1609.09450

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[18] B. Luo, L. Cheng, and Y.-C. Wu, "Fully distributed clock synchronization in wireless sensor networks under exponential delays," *Signal Process.*, vol. 125, pp. 261–273, Aug. 2016.

[19] S. Magnússon, P. C. Weeraddana, M. G. Rabbat, and C. Fischione, "On the convergence of alternating direction Lagrangian methods for nonconvex structured optimization problems," *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 3, pp. 296–309, Sep. 2016.

[20] Y. Wang, W. Yin, and J. Zeng. (Nov. 2016). "Global convergence of ADMM in nonconvex nonsmooth optimization." [Online]. Available: https://arxiv.org/abs/1511.06324

[21] T. Erseghe, "A distributed and scalable processing method based upon ADMM," *IEEE Signal Process. Lett.*, vol. 19, no. 9, pp. 563–566, Sep. 2012.

[22] S. Diamond, R. Takapoui, and S. Boyd. (Jan. 2016). "A general system for heuristic solution of convex problems over nonconvex sets." [Online]. Available: https://arxiv.org/abs/1601.07277

[23] K. Huang and N. D. Sidiropoulos, "Consensus-ADMM for general quadratically constrained quadratic programming," *IEEE Trans. Signal Process.*, vol. 64, no. 20, pp. 5297–5310, Oct. 2016.

[24] N. Derbinsky, J. Bento, V. Elser, and J. Yedidia. (May 2013). "An improved three-weight message passing algorithm." [Online]. Available: https://arxiv.org/abs/1305.1961

[25] X. Liu and S. C. Draper, "ADMM LP decoding of non-binary LDPC codes in F_{2^m}," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 2985–3010, Jun. 2016.

[26] X. Fan, J. Song, D. P. Palomar, and O. C. Au, "Universal binary semidefinite relaxation for ML signal detection," *IEEE Trans. Commun.*, vol. 61, no. 11, pp. 4565–4576, Nov. 2013.

[27] M. Hong, Z. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM J. Optim.*, vol. 26, no. 1, pp. 337–364, Jan. 2016.

[28] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 644–658, Mar. 2015.

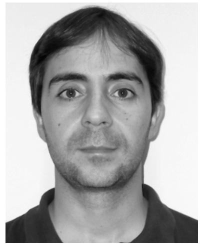
[29] Y. Xiao, Z. Yang, L. Dan, P. Yang, L. Yin, and W. Xiang, "Low-complexity signal detection for generalized spatial modulation," *IEEE Commun. Lett.*, vol. 18, no. 3, pp. 403–406, Mar. 2014.

[30] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.

[31] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 681–695, Mar. 2011.

[32] N. Benvenuto, R. Dinis, D. Falconer, and S. Tomasin, "Single carrier modulation with nonlinear frequency domain equalization: An idea whose time has come–again," *Proc. IEEE*, vol. 98, no. 1, pp. 69–96, Jan. 2010.

[33] A. Ghasemmehdi and E. Agrell, "Faster recursions in sphere decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3530–3536, Jun. 2011.



Nuno M. B. Souto (S'04–M'13–SM'16) received the degree in aerospace engineering-avionics branch, with the Instituto Superior Técnico, Lisbon, Portugal, and the Ph.D. degree in 2006. From 2000 to 2002, he was a Researcher in automatic speech recognition with the Instituto de Engenharia e Sistemas de Computadores, Lisbon, Portugal. He joined as an Assistant Professor with the ISCTE-University Institute of Lisbon in 2006. He has been a Researcher with the Instituto de Telecomunicações, Portugal, since 2002. He has been involved in several international research projects and many national projects. His research interests include wireless networks, signal processing for communications, OFDM, single carrier transmission with frequency domain equalization, channel coding, modulation, channel estimation, synchronization, MIMO schemes, wireless sensor networks, and unmanned aerial vehicles. He is a member of the IEEE Signal Processing Society.

HUGO ANDRÉ LOPES received the B.Sc. and M.Sc. degrees in computer science and telecommunications from the the ISCTE-University Institute of Lisbon in2014 and 2016, respectively. He is currently with NOS SGPS. His research interests include wireless networks, spatial modulations, and MIMO.