# Sparse channel estimation algorithms for OTFS system

Rabah Ouchikh, Abdeldjalil Aïssa-El-Bey, Thierry Chonavel, Mustapha Djeddou

Received: 22 April 2022 | Revised: 13 July 2022 | Accepted: 13 July 2022

DOI: 10.1049/cmu2.12469

IET Communications

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

# Sparse channel estimation algorithms for OTFS system

Rabah Ouchikh[1] | Abdeldjalil Aïssa-El-Bey[2] | Thierry Chonavel[2] | Mustapha Djeddou[1]

[1]Laboratoire Télécommunications, Ecole Militaire Polytechnique, Bordj El-Bahri, Algeria

[2]IMT Atlantique, Lab-STICC, UMR CNRS 6285, Brest F-29238, France

**Correspondence**
Abdeldjalil Aïssa-El-Bey, Mathematical and Electrical Engineering, IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238, Brest, France.
Email: abdeldjalil.aissaelbey@imt-atlantique.fr

**Abstract**

Orthogonal time-frequency space (OTFS) modulation, which has recently been proposed in the literature, is one of the promising techniques designed in the 2D Delay-Doppler domain adapted to combat high Doppler fading channels. However, channel estimation in high Doppler scenarios in advanced mobile-communication systems is still a challenging task. In this paper, the problem of channel estimation in the Delay-Doppler domain of the OTFS is focused on. First, a simple adaptation of the generalized orthogonal matching pursuit procedure, which will serve as a baseline method in this work, is proposed. Then, iterative algorithms are derived beneficiating from the sparsity of the channel. The unknown channel vector is separated into an unknown sparse support vector corresponding to the delay and Doppler taps, and an unknown vector of channel gains. These algorithms involve $\ell_1$-norm minimization and a two-stage iterative procedure to recover alternatively the channel support and its coefficients. The estimation problem is also addressed from a Bayesian point of view. The sparse representation is reformulated as a specific marginalization of the maximum a posteriori problem on the support of the channel. To deal with the intractability of this problem, two existing techniques are adapted to this context, namely: The Monte Carlo Markov chain with the Gibbs sampler and variational mean-field approximation with the variational Bayesian expectation-maximization procedure. Finally, to assess the performance of the proposed algorithms, their complexity and performance are compared against existing methods. Experimental tests, conducted in high-mobility scenarios and low-latency applications, show that the proposed schemes are slightly more expensive in terms of complexity load but perform significantly better in terms of normalized mean square error and bit error rate.

## 1 | INTRODUCTION

Orthogonal frequency division multiplexing (OFDM) is the most popular modulation technique deployed in 4G long-term evolution and 5G mobile-communication systems [1]. It is known for its robustness and high spectral efficiency in time-invariant frequency selective channels. However, its performance deteriorates when the Doppler effect is high (in high-mobility environments, such as high-speed trains); hence, the need for other modulation techniques to deal with this problem [2].

Orthogonal time frequency space (OTFS) modulation, recently proposed in [3, 4], is one of the promising techniques for advanced mobile-communication systems (5G and beyond) thanks to the increased performance that it provides compared to OFDM modulation. With OTFS, each data symbol is multiplexed onto 2D orthogonal basis functions specifically developed to deal with time-varying multipath channels dynamics at high speeds.

To detect data with minimal errors, the channel's impulse response must be known on the receiver side. Several channel estimation (CE) schemes for OTFS systems have been proposed in the literature. Let's briefly review some of them.

In [5], a 2D turbo compressed sensing (CS) algorithm for CE is featured based on the previous work on turbo CS for fast time-varying CE suggested in [6]. The support matrix of the Delay-Doppler channel is modeled by using a Markov random field and the Bernoulli Gaussian distribution.

The authors in [7] suggested a one uplink-aided high-mobility downlink CE scheme for the massive multiple-input multiple-output (MIMO) OTFS networks. The MIMO-OTFS signal model along the uplink and the reciprocity between the uplink and the downlink are formulated to recover the parameters of the channel including the delay, the angle, the Doppler frequency, and the channel gain for each physical scattering path.

A 3D-structured orthogonal matching pursuit algorithm-based CE technique is proposed in [8], which exploits the 3D-structured channel sparsity in the Delay-Doppler-angle domain.

In [9], a CE scheme is established using pulses in the Delay-Doppler domain as pilots adapted for MIMO-OTFS. The equivalent MIMO-OTFS channel matrix is obtained using a single MIMO-OTFS frame thanks to the sufficient spacing between pilots and data in the Delay-Doppler domain.

In another work [2], an embedded pilot-aided CE scheme for OTFS is highlighted. In each OTFS frame, pilot, guard, and data symbols are arranged appropriately in the Delay-Doppler domain to avoid interference between the data symbols and the pilot at the receiver side. The used CE is founded on a threshold method. The estimated channel is fed to data detection through the message-passing technique proposed in [10].

The authors in [11] proposed a pilot design and optimization for the OTFS system to improve the accuracy of CE. The classical orthogonal matching pursuit (OMP) technique is adopted for CE.

In [12], a Bayesian learning-aided sparse CE scheme for OTFS modulated systems are featured. The pilots are directly transmitted over the time-frequency domain grid for estimating the Delay-Doppler domain channel state information (CSI). Within the same context in [13], an off-grid CE scheme for OTFS systems is proposed based on the sparse Bayesian learning framework.

Another pilot pattern, namely: a superimposed scheme, has been recently proposed. In this scheme, pilots and data symbols are spread in the Delay-Doppler domain. Several algorithms for CE and data detection using this scheme are available [14–16]. In [14], a data-aided CE algorithm for a superimposed pilot and data transmission scheme is suggested to improve spectral efficiency. To accurately estimate the channel and detect the data symbols, the channel is coarsely estimated based on the pilot symbol, followed by an iterative process that detects the data symbols and refines the channel estimate. In [15], pilots and data symbols are superimposed in the Delay-Doppler grid. In [16], an iterative algorithm for CE and data detection is proposed. The CE step is formulated as a sparse recovery problem via the variational Bayesian expectation-maximization (VB-EM) procedure.

In the present paper, several CE algorithms for OTFS systems are proposed and compared against two state-of-the-art methods recently suggested in [2] and [11].

First, this work starts with a simple adaptation of a method that belongs to the greedy family, which is the generalized orthogonal matching pursuit (*GOMP*) [17, 18]. This adapted technique will serve as a baseline method in the developed solutions.

Next, two algorithms involving $\ell_1$-minimization that take into account the sparse nature of this problem are proposed. The main idea is the separation of the unknown channel vector into an unknown sparse support vector, which corresponds to the locations of delay and Doppler taps, and an unknown vector of channel gains. These two $\ell_1$-minimization algorithms use an iterative two-stage procedure that consists on optimizing the recovery problem alternatively concerning support and gain vectors. The first method resorts to the penalization of the mean square error with the $\ell_1$-norm of the gain vector. The second uses the reweighted $\ell_1$-norm, which further improves the recovery performance.

Alternatively, the sparse representation problem is reformulated as a specific marginalization of the maximum a posteriori (MAP) problem on the sparse support-vector. And to deal with the intractability of this problem, two Bayesian-based algorithms are proposed. Whereas, the first relies on the Monte Carlo Markov chain (*MCMC*) with the Gibbs sampler, the second uses a particular variational mean-field approximation and the VB-EM procedure.

In summary, the main contributions of this work are:

1. The proposal of two algorithms for sparse CE that involve $\ell_1$-minimization to optimize the recovery issue about gain and support vectors.
2. The proposal of two Bayesian-based algorithms for sparse CE.
3. The complexity analysis of the suggested algorithms and the evaluation of their performance in terms of NMSE and BER.

Simulations will show that, compared to state-of-the-art methods, most of the proposed algorithms perform significantly better in terms of normalized mean square error (NMSE) and bit error rate (BER) but are slightly more expensive. Experimental tests will also show that the proposed algorithms remain applicable in high-mobility scenarios and low-latency applications.

The rest of the paper is structured as follows. In Section 2, the basic concepts of OTFS modulation are introduced and the CE problem is formulated. In Section 3, the proposed CE algorithms are described and their complexity is evaluated. In Section 4, the two selected state-of-the-art methods are described and the performance of the proposed algorithms are compared against these methods.

**Notations**: symbols $a$, $\boldsymbol{a}$, and $\boldsymbol{A}$, stand for scalar, vector, and matrix, respectively. $(.)^H$ denotes Hermitian transposition. Operators $\otimes$ and $\odot$ denote Kronecker product and Hadamard product, respectively. vec(.) denotes the column vectorization of an $M \times N$ matrix into an $MN$ column vector. $\boldsymbol{F}_n$ and $\boldsymbol{F}_n^H$ denote, respectively, the $n$-point direct and inverse discrete Fourier transform matrices. $\boldsymbol{I}_M$ is the $M \times M$ identity matrix.

## 2 | GENERAL CONCEPTS OF OTFS AND PROBLEM FORMULATION

In this section, first, the basic concepts of OTFS modulation are detailed. Then, the input-output relation of the

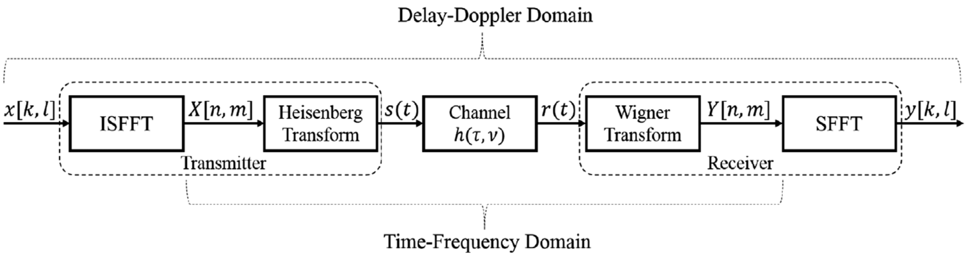

**FIGURE 1** OTFS modulation block diagram: transmitter, channel and receiver

**TABLE 1** OTFS system parameters

| Notation | Physical signification |
|---|---|
| $N, M$ | Number of time slots, number of subcarriers |
| $\Delta t$ $(s)$, $\Delta f$ (Hz) | Slot duration, subcarrier spacing |
| $\frac{1}{N\Delta t}$ | Quantization step along the Doppler frequency axis |
| $\frac{1}{M\Delta f}$ | Quantization step along the delay axis |
| $k, l$ | Doppler index, delay index |
| $p_{tx}, p_{rx}$ | Transmit pulse, receive pulse |
| $T = N\Delta t, B = M\Delta f$ | OTFS frame duration, OTFS frame bandwidth |
| $\tau_{max}, \nu_{max}$ | Maximum delay shift, maximum Doppler shift |

OTFS-modulator, OTFS-demodulator, and channel in the Delay-Doppler domain is derived. Finally, the CE problem treated in this work is formulated.

## 2.1 | General concepts of OTFS

Details about the basic concepts of OTFS modulation can be found in [2], [4], [10], [19], [20]. Table 1 summarizes the notations and their physical meaning.

The time-frequency grid $G_{TF}$ and the Delay-Doppler grid $G_{DD}$ are expressed as follows:

$$G_{TF} = \{(n\Delta t, m\Delta f)\},$$
$$G_{DD} = \left\{\left(\frac{k}{N\Delta t}, \frac{l}{M\Delta f}\right)\right\}, \tag{1}$$

for $(n, k) \in \{0, \dots, N-1\}^2$ and $(m, l) \in \{0, \dots, M-1\}^2$.

The cross-ambiguity function between $p_{tx}(t)$ and $p_{rx}(t)$ is:

$$A_{p_{tx},p_{rx}}(t, f) \triangleq \int p_{tx}^*(u-t)p_{rx}(u)e^{-j2\pi f(u-t)}\mathrm{d}u. \tag{2}$$

We say that the *bi-orthonormal property* condition between $p_{tx}$ and $p_{rx}$ is satisfied if

$$\int p_{tx}^*(t)p_{rx}(t-n\Delta t)e^{j2\pi m\Delta f(t-n\Delta t)}\mathrm{d}t = \delta(n)\delta(m). \tag{3}$$

Here, a single-input single-output OTFS system whose block diagram is given in Figure 1 is considered. This diagram is divided into three parts: transmitter, Delay-Doppler channel, receiver.

On the transmitter side, the 2D information symbols $\boldsymbol{X}_{DD} = \{x[k,l]\}_{k=0,l=0}^{N-1,M-1}$ (e.g. QAM or M-PSK symbols), which are arranged on the Delay-Doppler grid $G_{DD}$, are mapped on the time-frequency grid $G_{TF}$ ($n = 0 : N-1$ and $m = 0 : M-1$) via the inverse symplectic fast Fourier transform as follows:

$$\boldsymbol{X}[n,m] = \frac{1}{\sqrt{NM}}\sum_{k=0}^{N-1}\sum_{l=0}^{M-1} x[k,l]e^{j2\pi(\frac{nk}{N}-\frac{ml}{M})}. \tag{4}$$

Then, the Heisenberg transform modulator, which is equivalent to an $M$-point inverse fast Fourier transform with a pulse-shaping $p_{tx}(t)$, will be applied on $\boldsymbol{X}$ to generate the transmitter time-domain signal $s(t)$ as follows:

$$s(t) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} \boldsymbol{X}[n,m]p_{tx}(t-n\Delta t)e^{j2\pi m\Delta f(t-n\Delta t)}. \tag{5}$$

The signal $r(t)$, received through a linear time-variant channel, can be expressed as [4]:

$$r(t) = \iint h(\tau,\nu)s(t-\tau)e^{j2\pi\nu(t-\tau)}d\tau d\nu + w(t), \tag{6}$$

where $w(t) \sim \mathcal{CN}(w(t); 0, \sigma^2)$ is the additive noise.

The Delay-Doppler channel is characterized by a reduced number of reflectors, each defined by its delay, Doppler shift and amplitude. The channel is sparse and its response $h(\tau,\nu)$ can be expressed in the following form:

$$h(\tau,\nu) = \sum_{i=1}^{P} h_i\delta(\tau-\tau_i)\delta(\nu-\nu_i), \tag{7}$$

where $\delta(.)$ is the Dirac delta function; $P$ is the number of paths; $h_i \in \mathbb{C}$ is the complex gain. $\tau_i \in [0, \tau_{max}]$ and $\nu_i \in [-\nu_{max}, \nu_{max}]$ are, respectively, the delay and Doppler shift associated with the $i^{th}$ path.

$l_i$ and $k_i$ designate the delay and Doppler taps for the $i^{th}$ path:

$$\{l_i, k_i\} = \{\tau_i M\Delta f, \nu_i N\Delta t\}. \tag{8}$$

On the receiver side, the signal $r(t)$ first passes through a matched filter bank, which supplies the cross-ambiguity function between $p_{rx}(t)$ and $r(t)$. Then, the output $Y(t, f)$ of the matched filter is sampled, leading to $\boldsymbol{Y}[n, m] = Y(t, f)|_{t=n\Delta t, f=m\Delta f}$. This procedure is referred to as the Wigner transform.

Next, the relationship between time-frequency output samples $\boldsymbol{Y}[n, m]$ and input samples $\boldsymbol{X}[n, m]$ is:

$$\boldsymbol{Y}[n, m] = \sum_{p=0}^{N-1} \sum_{q=0}^{M-1} \boldsymbol{H}_{n,m}[p, q] \boldsymbol{X}[p, q], \tag{9}$$

where

$$\begin{aligned} \boldsymbol{H}_{n,m}[p, q] = &\int \int h(\tau, \nu) \\ &\times A_{p_{tx}, p_{rx}}((n-p)\Delta t - \tau, (m-q)\Delta f - \nu) \\ &\times e^{j2\pi(\nu + q\Delta f)((n-p)\Delta t - \tau)} e^{j2\pi\nu p\Delta t} \mathrm{d}\tau \mathrm{d}\nu. \end{aligned} \tag{10}$$

Then, applying the symplectic fast Fourier transform yields

$$y[k, l] = \frac{1}{\sqrt{NM}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \boldsymbol{Y}[n, m] e^{-j2\pi(\frac{nk}{N} - \frac{ml}{M})}. \tag{11}$$

The connection between $y[k, l]$ and the transmitted symbols can be expressed as a 2D circular convolution [10] as follows:

$$\begin{aligned} y[k, l] = &\sum_{k'=-k_\nu}^{k_\nu} \sum_{l'=0}^{l_\tau} b_{k',l'} h_{k',l'} \beta_{k,l} x[[k-k']_N, [l-l']_M] \\ &+ v[k, l], \end{aligned} \tag{12}$$

where $b_{k',l'} \in \{0, 1\}$ is the path indicator and the term $v[k, l] \sim \mathcal{CN}(v[k, l]; 0, \sigma^2)$ is an additive circular white noise with variance $\sigma^2$. $\beta_{k,l}$ is a known phase shift caused by imperfect bi-orthogonality of the rectangular waveform [19], given by

$$\beta_{k,l} = \begin{cases} e^{j2\pi\left(\frac{l-l'}{M}\right)\frac{k'}{N}} & \text{if } l' \leq l < M, \\ e^{j2\pi\left(\frac{l-l'}{M}\right)\frac{k'}{N}} e^{-j2\pi\frac{k}{N}} & \text{if } 0 \leq l < l'. \end{cases} \tag{13}$$

## 2.2 | Problem formulation

In this work, a non-blind estimation approach for CE is considered, that is, to estimate the channel accurately, pilots in the Delay-Doppler map and an efficient pilot pattern are needed. Benefiting from the fact that the channel is sparse in the Delay-Doppler domain, CE can be seen as a sparse recovery problem.

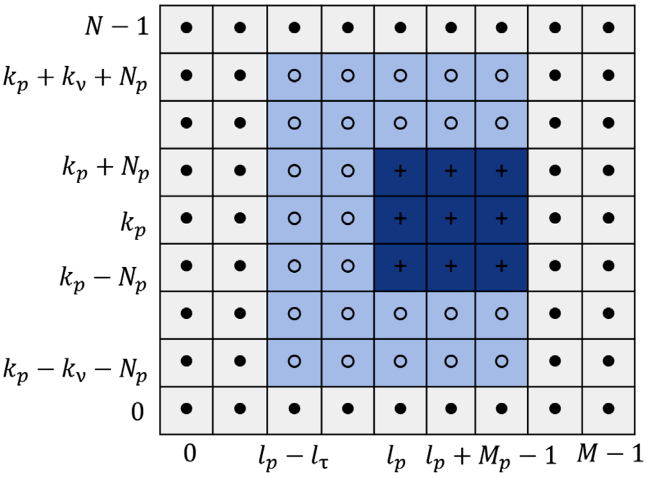

**FIGURE 2** Symbol arrangement in the Delay-Doppler grid (+: pilot; •: data symbols; ○: guard symbols)

The arrangement of pilots, guards, and data symbols proposed in [11] is used, where the proportion of guard symbols is reduced compared to the pattern of pilots proposed in [2]. In addition, the distribution of pilots over several Delay-Doppler positions in the grid helps limit the peak-to-average power ratio. On the receiver side, a simple, yet effective, interference cancellation scheme is adopted in [11] to address pilot contamination in the pilot pattern. Figure 2 illustrates this arrangement in the Delay-Doppler domain grid for OTFS frame transmission.

For each OTFS frame transmission, there are $L_p = (2N_p + 1)M_p$ pilots, $L_g = (2(k_\nu + N_p) + 1)(l_\tau + M_p) - L_p$ guard symbols and $L_d = MN - L_p - L_g$ data symbols in the Delay-Doppler grid $G_{DD}$. Their locations are given as follows:

$$x[k, l] = \begin{cases} x_p[k, l] & \text{if}\ , k \in [k_p - N_p, k_p + N_p] \\ & \text{and } l \in [l_p, l_p + M_p - 1], \\ x_d[k, l] & \text{if } k \notin [k_p - k_\nu - N_p, k_p + k_\nu + N_p], \\ & \text{and } l \notin [l_p - l_\tau, l_p + M_p - 1], \\ 0 & \text{otherwise (guard symbols),} \end{cases} \tag{14}$$

where $[k_p, l_p]$ denotes the reference position of the pilot symbols such that $0 \leq k_p \leq N - 1$ and $0 \leq l_p \leq M - 1$. Here, $x_d[k, l]$ represent the data symbols located at the grid location $[k, l]$.

Guard symbols ensure that received symbols for CE do not interfere with data symbols. We note that in this pattern there are no guard symbols on one side of the pilot. This is due to the channel causality in the delay dimension. Therefore, the data symbols on the right side will not corrupt the pilots. Unlike Doppler taps which can be negative or positive, it is necessary to place the guard intervals only on the left side of the pilots since delay taps cannot be negative.

In the Delay-Doppler domain, due to the transmit symbol arrangement in (14), the received pilot symbols can be expressed in the same way as (12), where $k \in [k_p - N_p, k_p + N_p]$ and $l \in [l_p, l_p + M_p - 1]$. This can be written in a matrix form as:

$$\boldsymbol{y}_p = (\boldsymbol{X}_p \odot \boldsymbol{\Psi})\boldsymbol{h} + \boldsymbol{v} = \boldsymbol{A}\boldsymbol{h} + \boldsymbol{v}. \tag{15}$$

Let $L = (2k_\nu + 1)(l_\tau + 1)$. Then,$\boldsymbol{y}_p \in \mathbb{C}^{L_p}$ is the received vector for pilots, which is expressed as $[\boldsymbol{y}_p]_{((k+N_p)M_p+l-l_p)} = y[k, l]$, and $\boldsymbol{h} \in \mathbb{C}^L$ is the channel vector, which is expressed as $[\boldsymbol{h}]_{(l'(2k_{nu}+1)+k'+k_\nu)} = h_{k',l'}$. Here, $\boldsymbol{X}_p \in \mathbb{C}^{L_p \times L}$ represents the pilots' matrix, with $[\boldsymbol{X}_p]_{(l-l_p+(k+N_p)M_p, l'(2k_\nu+1)+k'+k_\nu)} = x[k - k', l - l']$, while, $\boldsymbol{\Psi} \in \mathbb{C}^{L_p \times L}$ is an additional phase shift matrix given by $[\boldsymbol{\Psi}]_{(l-l_p+(k+N_p)M_p, l'(2k_\nu+1)+k'+k_\nu)} = e^{-j2\pi k'(l-l')/NM}$.

It is worth noting that the vector $\boldsymbol{h}$ contains only $P$ non-zero elements, where $P$ is the number of paths. The CE in this context amounts to estimating the channel support $\{l_i, k_i\}_{i=1}^P$ and corresponding path parameters $\{h_i, \tau_i, \nu_i\}_{i=1}^P$. Note that from $\{l_i, k_i\}_{i=1}^P$ one can calculate $\{\tau_i, \nu_i\}_{i=1}^P$ using (8).

Finally, CE quantifies the sparse vector $\boldsymbol{h}$ from (15). In the next section, the proposed algorithms to solve this problem will be presented.

# 3 | PROPOSED ALGORITHMS

In this section, various algorithms for sparse CE in OTFS systems are presented. A straightforward way to solve problem (15) with the pilot pattern adopted in this work is to use a greedy approach. Thus, the first algorithm is a mere adaptation of the *GOMP* method [17, 18]. This adapted algorithm will be considered here as a baseline method for sparse CE. The remaining algorithms are divided into two categories. The first one includes two iterative algorithms involving $\ell_1$-minimization. The second one includes two algorithms based on Bayesian approaches. This section concludes with the complexity analysis of all the suggested algorithms.

## 3.1 | Adapted *GOMP* for sparse CE

*GOMP* is the generalization of the OMP [21, 22] in the sense that multiple indices are identified per iteration. Therefore, *GOMP* ends the process of recovery with a much smaller number of iterations than the OMP. It is seen in [17] that *GOMP* has a high-recovery performance (compared to $\ell_1$-minimization techniques) with fast processing speed and competitive computational complexity. In the present work, *GOMP* is adapted to solve approximately the following problem:

$$\hat{\boldsymbol{h}} = \arg\min_{\boldsymbol{h}} \|\boldsymbol{y}_p - \boldsymbol{A}\boldsymbol{h}\|_2 \quad \text{subject to} \quad \|\boldsymbol{h}\|_0 \leq \delta, \tag{16}$$

where $\|\boldsymbol{h}\|_0$ denotes the $\ell_0$-pseudo-norm (i.e. the number of non-zero entries of $\boldsymbol{h}$).

*GOMP* is divided into four steps, namely: *Identification*, *Augmentation*, *Estimation*, and *Residual Update*. These steps are detailed in [17]. *GOMP* is one of the greedy algorithms which are the least complex CS-theory detectors, but they suffer from error propagation. A non-zero false estimate at a given iteration can cause the non-zero estimate of all elements of the vector to be incorrect. Moreover, *GOMP* ensures convergence towards a local optimum, but there is no guarantee of convergence towards a global optimum. This is a known weakness of greedy algorithms.

## 3.2 | Iterative two-stage (*I2S*) algorithm

To get around the drawbacks of *GOMP*, a criterion and an algorithm that exploits the sparse structure of the channel $\boldsymbol{h}$ are defined. To this end, one can write $h_k = b_k g_k$, that is, $\boldsymbol{h} = \boldsymbol{b} \odot \boldsymbol{g}$ and look for

$$\begin{aligned} \underset{\boldsymbol{b},\boldsymbol{g}}{\text{minimize}} \quad & \|\boldsymbol{y}_p - \boldsymbol{A}(\boldsymbol{b} \odot \boldsymbol{g})\|_2, \\ \text{subject to} \quad & \boldsymbol{b} \in \{0, 1\}^L, \\ & \boldsymbol{g} \in \mathbb{C}^L. \end{aligned} \tag{17}$$

Solving (17) is computationally very demanding. To ease this problem, the usual approach is the relaxation of the constraints. Here, the constraints involving $\boldsymbol{b}$ are relaxed by using the $\ell_1$-norm with lower and upper bounds. Then, (17) is relaxed as follows:

$$\begin{aligned} \underset{\boldsymbol{b},\boldsymbol{g}}{\text{minimize}} \quad & \|\boldsymbol{y}_p - \boldsymbol{A}(\boldsymbol{b} \odot \boldsymbol{g})\|_2, \\ \text{subject to} \quad & \|\boldsymbol{b}\|_1 \leq \delta, \\ & 0 \leq \boldsymbol{b} \leq 1, \end{aligned} \tag{18}$$

where $\delta$ is introduced to force the sparsity of $\boldsymbol{b}$.

A two-stage algorithm is proposed to solve (18). This algorithm estimates the support $\boldsymbol{b}$ given $\boldsymbol{g}$ in the first stage. Then, it estimates the channel gain vector $\boldsymbol{g}$ given $\boldsymbol{b}$ in the second stage. The estimate of $\boldsymbol{g}$ is obtained via a minimum mean square error estimator which is given by

$$\hat{\boldsymbol{g}} = \boldsymbol{A}_{\hat{\boldsymbol{b}}}^H (\boldsymbol{A}_{\hat{\boldsymbol{b}}}^H \boldsymbol{A}_{\hat{\boldsymbol{b}}} + \gamma \boldsymbol{I}_L)^{-1} \boldsymbol{y}_p, \tag{19}$$

where $\boldsymbol{A}_{\hat{\boldsymbol{b}}} = \boldsymbol{A}\boldsymbol{D}_{\hat{\boldsymbol{b}}}$, $\boldsymbol{D}_{\hat{\boldsymbol{b}}}$ is an $(L \times L)$ matrix given by $\boldsymbol{D}_{\hat{\boldsymbol{b}}} = \text{diag}(\hat{\boldsymbol{b}})$ and $\gamma = \sigma^2/\sigma_g^2$, with $\sigma_g^2$ is the power of the $P$ channel coefficients.

To solve (18), the vector $\boldsymbol{g}_{LS}$ of channel gains using the least-squares solution for (15) is considered. Then, the initial vector $\boldsymbol{g}^{(0)}$ is defined as being the $L$ vector containing the $P$ largest values of $\boldsymbol{g}_{LS}$ and zeros elsewhere: $\boldsymbol{g}^{(0)} = \max_P(\boldsymbol{g}_{LS})$. The proposed iterative algorithm will be referred to as *iterative two-stage* (I2S) and is detailed in Algorithm 1.

At the output of Algorithm 1, the entries of the support vector $\hat{\boldsymbol{b}}$ belong to $[0, 1]$. To reconstruct $\boldsymbol{h}$, a binary support vector is needed. There are two options to estimate it from $\hat{\boldsymbol{b}}$. The first, without any prior, is to apply on $\hat{\boldsymbol{b}}$ a fixed threshold of $\rho = 0.5$. The second is to use an optimal hard threshold [23].

**ALGORITHM 1** Iterative two-stage (*I2S*) algorithm

```
Input: measurements $\boldsymbol{y}_p \in \mathbb{C}^{L_p}$, sensing matrix $\boldsymbol{A} \in \mathbb{C}^{L_p \times L}$,
  Initialize $\boldsymbol{g}_{LS} = (\boldsymbol{A}^H \boldsymbol{A})^{-1} \boldsymbol{A}^H \boldsymbol{y}_p$,
             $\boldsymbol{g}^{(0)} = \max_P(\boldsymbol{g}_{LS})$,
  for $i = 1 : N_{iter}$ do
    stage (1) $\begin{cases} \hat{\boldsymbol{b}}^{(i)} = \arg\min_{\boldsymbol{b}} \|\boldsymbol{y}_p - \boldsymbol{A}(\boldsymbol{b} \odot \boldsymbol{g}^{(i-1)})\|_2, \\ \text{s.t.} \begin{cases} \|\boldsymbol{b}\|_1 \leq P, \\ 0 \leq \boldsymbol{b} \leq 1, \end{cases} \end{cases}$
    stage (2) $\left\{ \boldsymbol{g}^{(i)} = \boldsymbol{A}^H_{\hat{\boldsymbol{b}}^{(i)}} (\boldsymbol{A}^H_{\hat{\boldsymbol{b}}^{(i)}} \boldsymbol{A}_{\hat{\boldsymbol{b}}^{(i)}} + \gamma \boldsymbol{I}_L)^{-1} \boldsymbol{y}_p, \right.$
  end for
Output: $\hat{\boldsymbol{b}}, \hat{\boldsymbol{g}}$.
```

**ALGORITHM 2** Reweighted iterative two-stage (*RI2S*) algorithm

```
Input: measurements $\boldsymbol{y}_p \in \mathbb{C}^{L_p}$, sensing matrix $\boldsymbol{A} \in \mathbb{C}^{L_p \times L}$,
  Initialize $\hat{\boldsymbol{g}}_{LS} = (\boldsymbol{A}^H \boldsymbol{A})^{-1} \boldsymbol{A}^H \boldsymbol{y}_p$,
             $\boldsymbol{g}^{(0)} = \max_P(\hat{\boldsymbol{g}}_{LS})$,
             $\boldsymbol{w}^{(1)} = \boldsymbol{1}_L$,
  for $i = 1 : N_{iter}$ do
    stage (1) $\begin{cases} \hat{\boldsymbol{b}}^{(i)} = \arg\min_{\boldsymbol{b}} \|\boldsymbol{y}_p - \boldsymbol{A}(\boldsymbol{b} \odot \boldsymbol{g}^{(i-1)})\|_2, \\ \text{s.t.} \begin{cases} \|\boldsymbol{w}^{(i)} \odot \boldsymbol{b}\|_1 = \sum_{k=1}^{L} w_k^{(i)} |b_k| \leq P, \\ 0 \leq \boldsymbol{b} \leq 1, \end{cases} \end{cases}$
    stage (2) $\left\{ \boldsymbol{g}^{(i)} = \boldsymbol{A}^H_{\hat{\boldsymbol{b}}^{(i)}} (\boldsymbol{A}^H_{\hat{\boldsymbol{b}}^{(i)}} \boldsymbol{A}_{\hat{\boldsymbol{b}}^{(i)}} + \gamma \boldsymbol{I}_L)^{-1} \boldsymbol{y}_p, \right.$
  end for
  Update: $w_k^{(i+1)} = \frac{1}{|b_k^{(i)}| + \epsilon}$,
Output: $\hat{\boldsymbol{b}}, \hat{\boldsymbol{g}}$.
```

## 3.3 | Reweighted iterative two-stage (*RI2S*) algorithm

To improve sparse signal recovery, an interesting solution has been proposed in [24], named "reweighted $\ell_1$-minimization". This solution can be seen as an intermediate between the $\ell_0$-minimization and the $\ell_1$-minimization because it approximates the former (which is not convex) while preserving its convexity at each iteration, thanks to a criterion based on the $\ell_1$-norm. Therefore, based on this approach, a CE algorithm for OTFS modulation, named *Reweighted iterative two-stage* (*RI2S*) algorithm is proposed. The difference between *RI2S* and *I2S* lies in the first stage, where the estimate of the channel support $\boldsymbol{b}$ (given $\boldsymbol{g}$) is provided by applying reweighted $\ell_1$-minimization. This is detailed in Algorithm 2, where $\boldsymbol{W} = \text{diag}(w_1, w_2, \dots, w_L)$. We note that the initial weights are set to $\boldsymbol{w}^{(1)} = [w_1, w_2, \dots, w_L]^T = \boldsymbol{1}_L$.

Here, the weights $w_k^{(i+1)}$ are updated as follows: $w_k^{(i+1)} = \frac{1}{|b_k^{(i)}| + \epsilon}$. Large/small weights could be used to discourage/encourage non-zero entries in the recovered signal. Parameter $\epsilon > 0$ is introduced to avoid the norm divergence due to high weight values caused by extremely low entries of $\boldsymbol{b}$ (provides stability) and to ensure that a zero-valued component in $b_k^{(i)}$ does not strictly prohibit a non-zero estimate at the next step. A possible adaptive choice of $\epsilon$ is given in [24] as $\epsilon = \max\{|\boldsymbol{b}^{(i-1)}|_{(P+1)}, 10^{-3}\}$, where $|\boldsymbol{b}^{(i-1)}|_{(P+1)}$ is the $(P+1)^{th}$ largest elements of $|\boldsymbol{b}^{(i-1)}|$.

## 3.4 | Bayesian approaches

The unstructured sparse representation problem can be reformulated as a particular marginalized MAP problem on the support $\boldsymbol{b}$ of the sparse vector $\boldsymbol{h}$. To deal with the intractability of direct optimization of the distributions of interest, two possible approaches are considered. The first is based on *MCMC* simulation and involves the Gibbs sampler. The second is based on a particular variational mean-field approximation and VB-EM procedure [25]. Before the description of the two proposed Bayesian-based CE algorithms, it is needed to define the probabilistic prior models used to derive these techniques.

### 3.4.1 | Probabilistic model

The observation model (15) can be also expressed as follows:

$$\boldsymbol{y}_p = \sum_{k=1}^{L} b_k g_k \boldsymbol{A}_k + \boldsymbol{v}, \tag{20}$$

where $\boldsymbol{A}_k$ is the $k^{th}$ column of $\boldsymbol{A}$ and $\boldsymbol{v} = [v_1, \dots, v_L]^T$ is the noise vector which is a white circular Gaussian, with $p(v_k) = \frac{1}{\pi\sigma^2} e^{-|v_k|^2/2\sigma^2}$. Therefore, $p(\boldsymbol{y}_p|\boldsymbol{g}, \boldsymbol{b}) = \mathcal{CN}(\boldsymbol{A}_b \boldsymbol{g}_b, \sigma^2 \boldsymbol{I}_L)$, where $\boldsymbol{A}_b \in \mathbb{C}^{L_p \times P}$ and $\boldsymbol{g}_b \in \mathbb{C}^P$ are made up of $\boldsymbol{A}$ and $\boldsymbol{g}$ considering indices $k$ where $b_k \neq 0$.

The sparse nature of $\boldsymbol{h}$ lies on the fact that most of its entries are close to 0 and a few of them are not negligible. To account for this prior information, the entries of $\boldsymbol{h}$ can be described as a mixture of two zero-mean Gaussian distributions, with respective variances $\sigma_0^2$ and $\sigma_1^2$, such that $\sigma_1^2 \gg \sigma_0^2$: $p(\boldsymbol{h}) = \prod_{k=1}^{L} p(h_k) = \prod_{k=1}^{L} p_k \mathcal{CN}(h_k; 0, \sigma_1^2) + (1 - p_k)\mathcal{CN}(h_k; 0, \sigma_0^2)$, where $p_k = p(b_k = 1) = 1 - p(b_k = 0)$. Therefore, $\boldsymbol{g}$ obeys the following probabilistic model:

$$p(\boldsymbol{g}) = \prod_{k=1}^{L} p(g_k) \quad \text{where} \quad g_k \sim \mathcal{CN}\left(g_k; 0, \sigma_{g_k}^2\right). \tag{21}$$

Note that the variances $\sigma_{g_k}^2$ of Gaussian distributions realizations are independent of the support $\boldsymbol{b}$. To detect the location of spikes, note that the variables $b_k$ associated with the corresponding entries of $\boldsymbol{h}$ are assumed to be independent Bernoulli random variables, such that $b_k = 1$ if a spike is present at $h_k$ and $b_k = 0$ otherwise. Thus, $h_k|b_k \sim \mathcal{CN}(h_k; 0, (\sigma_1^2)^{b_k}(\sigma_0^2)^{1-b_k})$.

Unstructured sparsity can be modeled by a standard choice based on a product of the Bernoulli distributions, as follows:

$$p(\boldsymbol{b}) = \prod_{k=1}^{L} p(b_k) \quad \text{where} \quad b_k \sim \mathrm{Ber}(p_k). \tag{22}$$

Alternatively, instead of seeking to approximate the means of the posterior distributions via *MCMC* simulation, recent approaches aim to iteratively calculate exactly a variational approximation of the target posterior distribution.

### 3.4.2 | Channel estimation in a Bayesian framework

At this point, MAP estimation of the sparse channel parameters, which corresponds to the optimal Bayesian estimator [26] is addressed. In this case, the estimation problem of support and channel gains can take the following form:

$$(\hat{\boldsymbol{g}}, \hat{\boldsymbol{b}}) = \underset{\boldsymbol{g},\boldsymbol{b}}{\arg\max} \log(p(\boldsymbol{g}, \boldsymbol{b}|\boldsymbol{y}_p)). \tag{23}$$

First, the channel support vector $\boldsymbol{b}$ is determined. To achieve this, the probability of a false decision on $\boldsymbol{b}$ is minimized. The decision that leads to this minimization is given as follows:

$$\hat{\boldsymbol{b}} = \underset{\boldsymbol{b}\in\{0,1\}^L}{\arg\max} \log(p(\boldsymbol{b}|\boldsymbol{y}_p)). \tag{24}$$

The maximization of $p(\boldsymbol{b}|\boldsymbol{y}_p)$ is too costly. This is because solving (24) requires the evaluation of the function $\log(p(\boldsymbol{b}|\boldsymbol{y}_p)$ for all sequences of $\boldsymbol{b} \in \{0,1\}^L$ ($2^L$ evaluations). Individual decisions for the entries of the support $\boldsymbol{b}$ can be made from a marginalized MAP, leading to:

$$\hat{b}_k = \underset{b_k\in\{0,1\}}{\arg\max} \log(p(b_k|\boldsymbol{y}_p)). \tag{25}$$

In the same way, the evaluation of $p(b_k|\boldsymbol{y}_p)$ is intractable due to the costly marginalization of $p(\boldsymbol{b}|\boldsymbol{y}_p)$ over the $b_l$'s, for $l \neq k$. To circumvent this difficulty, *MCMC* with Gibbs sampler and mean-field variational approximations compute a tractable surrogate $q(b_k)$ of $q(b_k|\boldsymbol{y}_p)$ [25],[27]. In this case, (25) will be approximated by the following easier problem:

$$\hat{b}_k = \underset{b_k\in\{0,1\}}{\arg\max} \log(q(b_k)), \tag{26}$$

Once the support $\boldsymbol{b}$ is estimated, channel coefficients $\boldsymbol{g}$ can be estimated as $\hat{\boldsymbol{g}} = \underset{\boldsymbol{g}}{\arg\max} \log(p(\boldsymbol{g}|\hat{\boldsymbol{b}}, \boldsymbol{y}_p))$:

$$\hat{\boldsymbol{g}}_{\hat{\boldsymbol{b}}} = (\boldsymbol{A}_{\hat{\boldsymbol{b}}}^T \boldsymbol{A}_{\hat{\boldsymbol{b}}} + \boldsymbol{\Delta})^{-1} \boldsymbol{A}_{\hat{\boldsymbol{b}}}^T \boldsymbol{y}_p, \tag{27}$$

where $\hat{\boldsymbol{g}}_{\hat{\boldsymbol{b}}}$ denotes entries of $\boldsymbol{g}$ restricted to the estimated support, $\boldsymbol{A}_{\hat{\boldsymbol{b}}}$ is the corresponding columns of $\boldsymbol{A}$ and $\boldsymbol{\Delta} = \mathrm{diag}(\sigma^2/\sigma_{g_1}^2, \sigma^2/\sigma_{g_2}^2, \dots, \sigma^2/\sigma_{g_L}^2)$. If $\sigma^2 \lll \sigma_{g_k}^2$ $\forall k$, then, this solution reduces to the least-square solution.

### 3.4.3 | *MCMC* algorithm

The idea is to generate realizations of a given process and use them to compute an estimator of the parameters. Thus, to decide $\boldsymbol{b}$, the marginal posterior probability $p(\theta|\boldsymbol{y}_p)$ of $\boldsymbol{\theta} = (\boldsymbol{g}, \boldsymbol{b})$ is simulated using a Gibbs algorithm [28]. The Gibbs sampler simulates realizations of samples $\theta_k$ according to the a posteriori marginals $p(\theta_k|\boldsymbol{y}_p, \theta_{-k})$. Denoting by $\boldsymbol{g}_{-k}$ the vector $\boldsymbol{g}$ where entry $g_k$ has been removed and defining $\boldsymbol{b}_{-k}$ in the same way, each loop of the Gibbs algorithm leads to the iterative simulation of variables $g_k \sim p(g_k|\boldsymbol{g}_{-k}, \boldsymbol{b}, \boldsymbol{y}_p)$ and $b_k \sim p(b_k|\boldsymbol{b}_{-k}, \boldsymbol{g}, \boldsymbol{y}_p)$. Implementation of the Gibbs sampler is summarized in Appendix A.1. The Gibbs sampler is given by the following distributions:

$$\begin{aligned} q(g_k) &= p(g_k|\boldsymbol{g}_{-k}, \boldsymbol{b}, \boldsymbol{y}_p) \\ &= \mathcal{CN}\left(g_k; \frac{\sigma_k^2}{\sigma^2} \boldsymbol{A}_k^T \left(\boldsymbol{y}_p - \boldsymbol{A}_{-k} \boldsymbol{g}_{-k}\right), \sigma_k^2\right), \\ q(b_k) &= p(b_k|\boldsymbol{b}_{-k}, \boldsymbol{g}, \boldsymbol{y}_p) \\ &= \frac{\alpha^{b_k}(1-\alpha)^{1-b_k} \mathcal{CN}(g_k; 0, \sigma_{b_k}^2)}{\alpha \mathcal{CN}(g_k; 0, \sigma_1^2) + (1-\alpha)\mathcal{CN}(g_k; 0, \sigma_0^2)}, \end{aligned} \tag{28}$$

where $\boldsymbol{A}_k = \boldsymbol{A}(:, k)$, $\boldsymbol{A}_{-k} = (\boldsymbol{A}_1, \dots, \boldsymbol{A}_{k-1}, \boldsymbol{A}_{k+1}, \dots, \boldsymbol{A}_L)$, $\sigma_k^2 = (\|\boldsymbol{A}_k\|^2/\sigma^2 + 1/\sigma_{b_k}^2)^{-1}$ and $\alpha = p(b_k = 1)$.

$(\boldsymbol{b}^{(i)}, \boldsymbol{g}^{(i)})_{i=1:K}$ denote the vectors supplied by operating the Gibbs sampler $K$ times. After $K_0$ iterations, which represent a learning period, the samples are (roughly) sampled according to the target distribution. Then, comparing averaged simulations for $b_k$ to a threshold $\rho = 0.5$, minimizes the Bayes risk for the corresponding decision $\hat{b}_k \in \{0, 1\}$:

$$\hat{b}_k = \begin{cases} 1 & \text{if } \frac{1}{K-K_0} \sum_{i=K_0+1}^{K} b_k^{(i)} > \rho, \\ 0 & \text{otherwise,} \end{cases} \tag{29}$$

Then, for $\hat{\boldsymbol{g}}$, the corresponding maximum likelihood estimator (27) can be chosen.

Details of the implementation of the proposed *MCMC* algorithm for CE are summarized in Algorithm 3.

### 3.4.4 | *SoBaP* algorithm

The methodology adopted here is to compute an approximation $q(b_k)$ of the posterior probability $p(b_k|\boldsymbol{y}_p)$ named the mean-field approximation. It is detailed in [25]. The mean-field

**ALGORITHM 3** *MCMC* algorithm

```
Input: measurements $\boldsymbol{y}_p \in \mathbb{C}^{L_p}$, sensing matrix $\boldsymbol{A} \in \mathbb{C}^{L_p \times L}$, Bernoulli parameter $\alpha$,
  Initialization: $\boldsymbol{g}$ and $\boldsymbol{b}$,
          $\hat{\boldsymbol{g}}$: $\hat{\boldsymbol{g}}^{(0)} \sim (\mathcal{U}_{[0,1]})_{1:L}$,
          $\hat{\boldsymbol{b}}$: $\hat{\boldsymbol{b}}^{(0)} \sim (\mathcal{B}_{\alpha})_{1:L}$,
  while $k \leq K$ and $|q(b_i^{(k)}) - q(b_i^{(k-1)})| < \epsilon \; \forall i \in \{1, \dots, L\}$ do
    for $l = 1 : L$
    do
      simulate $q(g_l^{(k)}) = p(g_l^{(k)} | \boldsymbol{g}_{-l}, \boldsymbol{b}, \boldsymbol{y}_p)$
      $= \mathcal{CN}(g_l^{(k)}; \frac{\sigma_k^2}{\sigma^2} \boldsymbol{A}_l^T (\boldsymbol{y}_p - \boldsymbol{A}_{-l} \boldsymbol{g}_{-l}), \sigma_k^2)$,
      simulate $q(b_l^{(k)}) = p(b_l^{(k)} | \boldsymbol{b}_{-l}, \boldsymbol{g}, \boldsymbol{y}_p)$
      $= \frac{\alpha^{b_l^{(k)}} (1-\alpha)^{1-b_l^{(k)}} \mathcal{CN}(g_l; 0, \sigma_{b_l}^2)}{\alpha \mathcal{CN}(g_l; 0, \sigma_1^2) + (1-\alpha) \mathcal{CN}(g_l; 0, \sigma_0^2)}$,
    end for
  end while
  compute estimated $\boldsymbol{b}$: $\hat{b}_k = \mathbf{1}_{[\rho, \infty[}(\frac{1}{K-K_0} \sum_{i=K_0+1}^{K} b_k^{(i)})$,
  compute the ML estimator of $\boldsymbol{g}$ using (27),
Output: $\hat{\boldsymbol{h}} = \hat{\boldsymbol{b}} \odot \hat{\boldsymbol{g}}$.
```

approximation [29] of $p(\boldsymbol{\theta} | \boldsymbol{y}_p)$ is the surrogate distribution $q^{\star}(\boldsymbol{\theta})$ satisfying

$$\underset{q^{\star}(\boldsymbol{\theta})}{\text{minimize}} \int_{\boldsymbol{\theta}} q(\boldsymbol{\theta}) \log \left( \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} | \boldsymbol{y}_p)} \right) \mathrm{d}\boldsymbol{\theta},$$

$$\text{subjectto} q(\boldsymbol{\theta}) = \prod_{k=1}^{K} q(\boldsymbol{\theta}_k),$$

$$\int_{\boldsymbol{\theta}_k} q(\boldsymbol{\theta}_k) \mathrm{d}\boldsymbol{\theta}_k = 1, \quad \forall k \in [1, K], \tag{30}$$

where $\boldsymbol{\theta} = (\boldsymbol{b}, \boldsymbol{g})$ and $p(\boldsymbol{\theta} | \boldsymbol{y}_p)$ is its posterior distribution.

The problem (30) can be solved by successive minimizations of the Kullback-Leibler divergence [30] with respect to factors $q(\boldsymbol{\theta}_i)$. Then, the VB-EM algorithm [31–33] used in this work is given in [25]. It will almost surely converge towards a saddle point or a local/global maximum of the problem (30) under mild conditions [25]. The relation between this procedure and the known expectation-maximization algorithm [34, 35] results from the addition of the constraint $q(\boldsymbol{\theta}_i) = \delta(\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i)$ on some $q(\boldsymbol{\theta}_i)'s$.

The mean-field approximation offers a good framework for approximating the marginals $p(\boldsymbol{\theta}_i | \boldsymbol{y}_p)$. Indeed,

$$\begin{aligned} p(\boldsymbol{\theta}_i | \boldsymbol{y}_p) &= \int_{\boldsymbol{\theta}_{-i}} p(\boldsymbol{\theta} | \boldsymbol{y}_p) \mathrm{d}\boldsymbol{\theta}_{-i}, \\ &\simeq \int_{\boldsymbol{\theta}_{-i}} q(\boldsymbol{\theta} | \boldsymbol{y}_p) \mathrm{d}\boldsymbol{\theta}_{-i}, \\ &\simeq q(\boldsymbol{\theta}_i | \boldsymbol{y}_p), \end{aligned} \tag{31}$$

where the last equality stems from the constraint of (30).

**ALGORITHM 4** *SoBaP* algorithm

```
Input: measurements $\boldsymbol{y}_p \in \mathbb{C}^{L_p}$, sensing matrix $\boldsymbol{A} \in \mathbb{C}^{L_p \times L}$,
  Initialization: $p(\boldsymbol{b}) = \prod_k p(b_k)$,
          prior mean for $\boldsymbol{g}$: $\boldsymbol{m} = \mathbf{0}_L$,
          probability $\boldsymbol{q}$: $\boldsymbol{q}^{(0)} \sim (\mathcal{U}_{[0,1]})_{1:L}$,
          $\boldsymbol{r}$: $\boldsymbol{r}^{(0)} = \boldsymbol{y}_p - \boldsymbol{A}(\boldsymbol{b} \odot \boldsymbol{m})$,
  while $k \leq K$ and $\forall i \in \{1 : L\} \, |q(b_i^{(k)}) - q(b_i^{(k-1)})| < \epsilon$ do
    Estimation of $\sigma^2$ (see Appendix A.3).
    for $l = 1, \dots, L$ do
      compute $\Sigma(b_k = 1)$ (see Appendix A.2).
      compute $m(b_l)^{(k)}$ and $q_l^{(k)}$ (see appendix A.2),
      update $\boldsymbol{r}^{(k)}$: $\boldsymbol{r}^{(k)} = \boldsymbol{r}^{(k)} - \boldsymbol{A}_l (b_l^{(k)} m(b_l)^{(k)})$,
    end for
  end while
  estimate support $\boldsymbol{b}$: $\hat{\boldsymbol{b}} = (\boldsymbol{q} > 0.5)$,
  estimate $\boldsymbol{g}$ conditional to $\boldsymbol{q}$ using (27),
Output: $\hat{\boldsymbol{b}}, \hat{\boldsymbol{g}}$.
```

The soft Bayesian pursuit (*SoBaP*) algorithm used in this section considers the particular case where the mean-field approximation $q(\boldsymbol{g}, \boldsymbol{b})$ of $p(\boldsymbol{g}, \boldsymbol{b} | \boldsymbol{y}_p)$ simply writes $q(\boldsymbol{g}, \boldsymbol{b}) = \prod_k q(g_k, b_k)$. Together with the model (21), the corresponding VB-EM update is given in Appendix A.2. From (31), an approximation of $p(b_k | \boldsymbol{y}_p)$ therefore follows simply from the relations $p(b_k | \boldsymbol{y}_p) \simeq \int q(g_k, b_k) \mathrm{d}g_k = q(b_k)$.

By using this approximation, (25) can be solved easily by a simple threshold operation, that is, $\hat{b}_k = 1$ if $q(b_k = 1) > \rho$ and $\hat{b}_k = 0$ otherwise, with $\rho = 0.5$ as for the *MCMC* algorithm. A crucial parameter that can influence the performance of *SoBaP* and that deserves to be estimated adaptively (at each iteration) is the noise variance $\sigma^2$. A good estimator of $\sigma^2$ is crucial for the convergence of the algorithm. Its estimation requires its introduction as a new variable in $\boldsymbol{\theta}$, that is, $\boldsymbol{\theta} = [\boldsymbol{b}, \boldsymbol{g}, \sigma^2]$. Then, $q(\boldsymbol{g}, \boldsymbol{b}, \sigma^2) = q(\sigma^2) \prod_k q(g_k, b_k)$. Letting $q(\sigma^2) = \delta(\sigma^2 - \hat{\sigma}^2)$, the estimation of $\sigma^2$ is given in Appendix A.3. The steps of *SoBaP* are detailed in Algorithm 4.

### 3.5 | Complexity analysis of the proposed algorithms

In the case of *GOMP*, the complexity of the $k^{th}$ iteration in terms of multiplications is approximatively $2L_pL + (4K^2 + 2K)kL_p$ [17]. Thus, the overall complexity is dominated by the identification step. For $N_g$ iterations, it is $\mathcal{O}(N_gL^2)$.

In the case of *I2S* and *RI2S*, solving the convex optimization problem considered in the first stage is relatively fast. The solution for this linear programming problem is obtained via the Matlab CVX toolbox [36, 37], a package for solving convex problems based on the interior-point method. The complexity is dominated by $\mathcal{O}(L^3)$. Whereas, the second stage, which involves the calculation of a minimum mean square error, is evaluated around $\mathcal{O}(L^3)$. Therefore, the overall complexity of these two algorithms is around $\mathcal{O}(N_t L^3)$ for $N_t$ iterations.

In the case of *MCMC*, the complexity requires $\mathcal{O}(L^2)$ operations for each iteration. Thus, the overall complexity of this algorithm is upper-bounded by $\mathcal{O}(N_m L^2)$ for $N_m$ iterations.

In the case of *SoBaP*, the most expensive operation is the update (equation (41) in [25]) which is $\mathcal{O}(L^2)$ at each iteration. Thus, the overall complexity of *SoBaP* is around $\mathcal{O}(N_s L^2)$ for $N_s$ iterations.

## 4 | RESULTS AND DISCUSSION

In this section, the proposed algorithms are tested and compared with two state-of-the-art methods from [2] and [11]. The description of these methods is given in Section 4.2. The comparative tests are presented in Section 4.3 in terms of complexity, NMSE, and BER in real scenarios.

### 4.1 | Simulation setup

All the simulations are performed on a regular desktop PC using MATLAB 2018a. Information bits are mapped to *QPSK* before being transmitted through the OTFS transmitter.

For the channel delay model, the Extended Vehicular A model [38] of the 3GPP, the standardization body for 5G cellular communications is adopted. Note that each delay tap has a different single Doppler shift in the form $\nu_i = \nu_{max}\cos(\theta_i)$, where $\theta_i \sim \mathcal{U}_{[0,\pi]}$ and $\nu_{max}$ denotes the maximum Doppler shift determined by the user's equipment speed. The maximum delay tap is set to $l_\tau = 10$ and the maximum Doppler tap to $k_\nu = 4$, which corresponds to a high-speed scenario with a maximum user's equipment speed $v = 120\ km/h$. The channel coherence time $T_c = 1/\nu_{max} = 2.25\ ms$ is longer than the OTFS symbol duration $T_s = T/(M\Delta f) = 0.13\ \mu s$, which means that the channel variation is slow in this case.

To achieve good CE, the size of pilot signals should cover the maximum delay and Doppler spread, that is, $M_p \geq l_\tau$ and $2N_p + 1 \geq k_\nu$. Thus, to balance the estimation accuracy and spectral efficiency, $M_p$ and $N_p$ are set to 16 and 4, respectively.

Pilots are chosen to optimize the average mutual incoherence property of $\boldsymbol{A}$ which is expressed as [11]

$$\mu_{avg} = \frac{2}{L(L-1)} \sum_{1\leq i<j\leq L} \frac{|\boldsymbol{a}_i^H \boldsymbol{a}_j|}{||\boldsymbol{a}_i||_2.||\boldsymbol{a}_j||_2}, \tag{32}$$

where $\boldsymbol{a}_i$ denotes the $i^{th}$ column of $\boldsymbol{A}$. Note that the coherence reflects the correlation between the columns of $\boldsymbol{A}$; a higher correlation will degrade the performance. Therefore, pilots will be chosen so as to minimize $\mu_{avg}$. Four types of pilot sequences are tested: *constant amplitude zero auto-correlation* sequences (*CAZAC*), *Gaussian* sequences, *Hadamard* sequences, and *particle swarm optimization (PSO)* sequences [11]. Table 2 shows that the best value of $\mu_{avg}$ is the one corresponding to the *PSO* sequences, which justifies choice of this type of pilots. Other system parameters retained for simulations are listed in Table 3.

**TABLE 2** Average mutual incoherence property ($\mu_{avg}$)

| Pilot sequence | $\mu_{avg}$ |
|---|---|
| Gaussian | 0.082 |
| Hadamard | 0.059 |
| CAZAC | 0.047 |
| PSO | 0.024 |

**TABLE 3** Simulation parameters

| Parameter | Value |
|---|---|
| The carrier frequency ($f_c$) | 4 GHz |
| Spacing between the sub-carriers ($\Delta f$) | 15 kHz |
| OTFS frame size ($N, M$) | (128, 512) |
| Modulation scheme | *QPSK* |
| Bandwidth ($B$) | 7.68 MHz |

Performance will be calculated in terms of BER and NMSE. The NMSE is a normalized mean square error with values in [0, 1], defined by

$$\text{NMSE} = 1 - \left(\frac{|\boldsymbol{h}^H \hat{\boldsymbol{h}}|}{||\boldsymbol{h}||_2 ||\hat{\boldsymbol{h}}||_2}\right)^2. \tag{33}$$

### 4.2 | Description of the selected state-of-the-art methods

The comparison of the developed techniques is done against the two state-of-the-art methods proposed in [2] and [11]. The first one is well known and widely used. The second one is a recent on-grid channel estimation method that uses a pilots scheme and guard intervals in the Delay-Doppler domain.

In [2], a simple CE method based on thresholding is proposed to solve (15) with respect to $\boldsymbol{h}$ using a pilot pattern with only one pilot symbol $x_p$ transmitted per frame. It operates as follows: for $1 \leq k \leq L_p$, entries $k$ such that $|\boldsymbol{y}_p(k)| \geq \lambda$ (real positive detection threshold), determine the support vector $\boldsymbol{b}$ of $\boldsymbol{h}$ where we set (above the threshold) $\boldsymbol{b}(k) = 1$ and $\boldsymbol{h}(k) = \boldsymbol{y}_p(k)/x_p$. Otherwise, $\boldsymbol{b}(k) = 0$ and $\boldsymbol{h}(k) = 0$. This method relies on the fact that the existence of a path implies that the received symbol is the weighted pilot signal with additive white Gaussian noise. Otherwise, it is only noise.

This method (Threshold method) is simple and has a low complexity. However, miss-detection and false-alarm probabilities can be altered by varying the threshold $\lambda$ on path detection. To guarantee good performance, the $SNR_p$ (pilot signal-to-noise ratio) must be high (the minimum value of $SNR_p$ used in [2] is 30 dB). If we assume that $SNR_p = SNR_d$ (the signal-to-noise ratio of data symbols), then this method may not detect some paths. In particular, for $SNR_p = SNR_d < 20$ dB, this method generally detects $n < P$ paths. An optimal threshold can be used to counter this problem [23]. At this point, as this approach will serve as a benchmark, it is slightly modified as follows. It is assumed that $P$ is known and that the estimation of channel coefficients are given as $\boldsymbol{h}(k) = \boldsymbol{y}_p(k)/x_p$, where $k$'s are the indices of the $P$ values with largest amplitudes in $\boldsymbol{y}_p$. This modification does not penalize at all the method. In fact, it will enable the method to reach its best performance so as to provide a good testing ground for the proposed algorithms.

As second method, the OMP procedure proposed in [11] is used. It uses a pilot pattern, where the proportion of guard symbols is reduced. To achieve accurate CE, based on the MIP criterion, an OTFS pilot sequence optimization problem is formulated and a PSO-based procedure is developed to solve it. The classical OMP algorithm is adopted as the CE algorithm.

## 4.3 | Comparison with the state-of-the-art methods

Here, the proposed CE algorithms are compared with the two selected state-of-the-art methods in terms of complexity, NMSE, and BER.

### 4.3.1 | Complexity

To refine the complexity analysis given in Section 3.5, the order of the number of iterations $N_g$, $N_t$, $N_m$, and $N_s$ must be known.

It is shown in [17] that $N_g \le \min(P, L/N)$, where $N$ is a small constant and $P$ is the number of channel taps. In practice, $N_g$ and $P$ have the same order. Therefore, the overall complexity of *GOMP* is $n_g\mathcal{O}(L^2)$.

The parameters $N_t$, $N_m$, and $N_s$ are determined experimentally. The average number of iterations needed for convergence as a function of the dimension $L$ for each of the proposed algorithms is plotted. Figure 3 shows the average number of iterations allowing *I2S*, *RI2S*, *MCMC* and *SoBaP* to converge, as a function of $L$ for a constant maximum delay tap ($l_\tau = 10$). Note that $L = (l_\tau + 1)(2k_\nu + 1)$, therefore, varying $k_\nu$ in the interval $[0, 16]$ amounts to varying $L$ in $[11, 363]$.

Figure 3 shows that the average number of iterations for *I2S* and *RI2S* remains constant (about 10) independently of the dimension $L$. For *MCMC* and *SoBaP*, this average number grows according $L$. However, in the case of *SoBaP*, this number variates slowly. The dotted lines in Figure 3 correspond to linear approximations of the *MCMC* and *SoBaP* curves. These approximations remain valid for the values of $k_\nu$ used in the present work. In summary, the computational cost of all the proposed algorithms is $\mathcal{O}(L^3)$; except for the baseline method *GOMP* which is $\mathcal{O}(L^2)$.

Another metric to qualify the complexity of these algorithms is the runtime. Figure 4 shows the runtime in seconds of the proposed algorithms as well as that of the two state-of-the-art methods as a function of $k_\nu$.

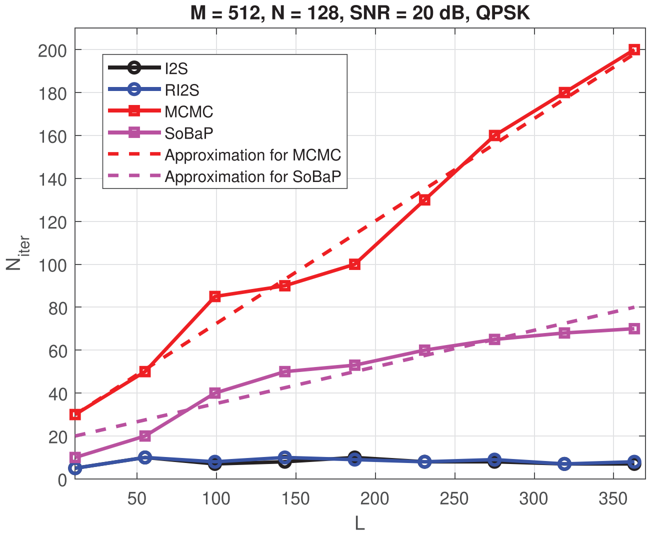

**FIGURE 3** Number of iterations $N_{iter}$ versus $L$, for $M = 512$, $N = 128$, $M_p = 16$, $N_p = 4$, $l_\tau = 10$, $SNR = 20$ dB and $QPSK$ modulation

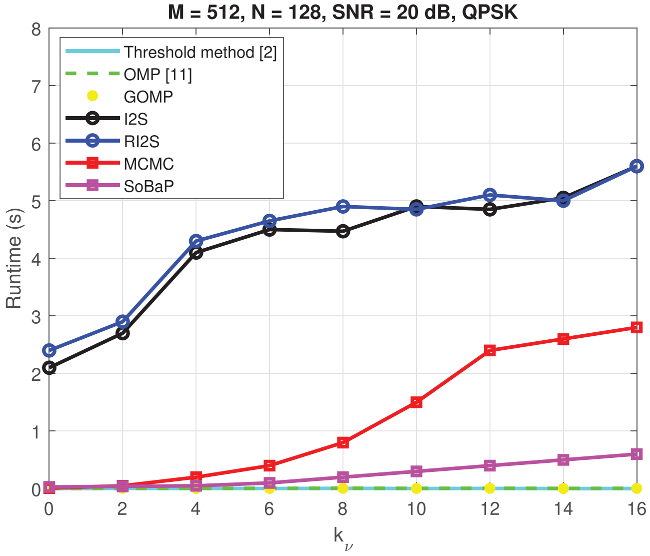

**FIGURE 4** Runtime versus maximum Doppler tap $k_\nu$, for $M = 512$, $N = 128$, $M_p = 16$, $N_p = 4$, $l_\tau = 10$, $SNR = 20$ dB and $QPSK$ modulation

Figure 4 confirms the complexity analysis of Section 3.5. The runtime of *I2S*, *RI2S*, *MCMC* and *SoBaP* are of the same order. The runtime of the baseline *GOMP* is smaller and is comparable to that of the state-of-the-art methods in [2] and [11].

### 4.3.2 | Normalized mean square error (NMSE)

The variation of NMSE as a function of SNR is now investigated. We have set $SNR = SNR_d = SNR_p$. For a fair comparison with the one-pilot scheme, the total power of each OTFS frame is the same for all schemes.

Figure 5 illustrates that *GOMP* and the methods in [2] and [11] show similar performance. All the other algorithms perform better, *SoBaP* being the best overall.

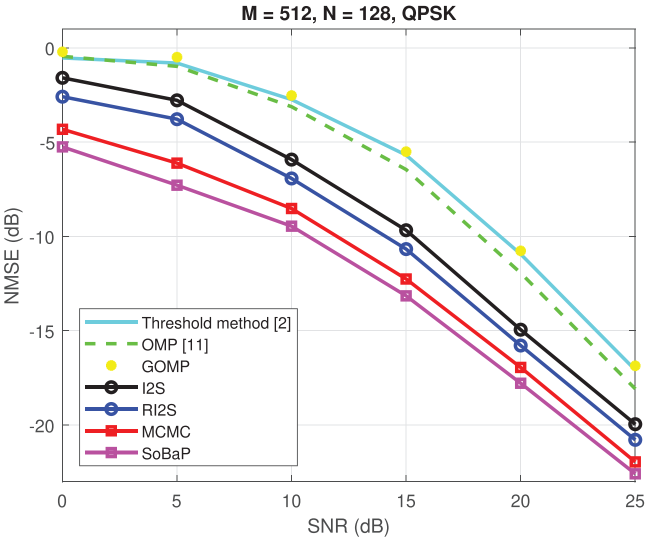

**FIGURE 5** NMSE versus SNR : maximum Doppler speed 120 km/h, for $M = 512$, $N = 128$, $M_p = 16$, $N_p = 4$ $l_\tau = 10$, $k_\nu = 4$ and *QPSK* modulation

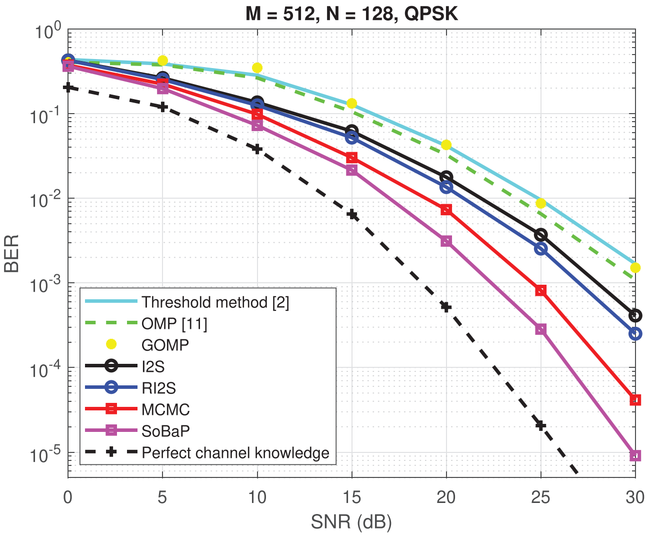

**FIGURE 6** BER versus SNR : maximum Doppler speed 120 km/h, for $M = 512$, $N = 128$, $M_p = 16$, $N_p = 4$, $l_\tau = 10$, $k_\nu = 4$ and *QPSK* modulation

### 4.3.3 | Bit error rate (BER)

The variation of BER as a function of SNR is now investigated. Here, the message-passing procedure [10] is used to detect the data symbols by solving a set of *MN* linear equations after the removal of pilots contribution.

Figure 6 shows the variation of BER versus SNR. Note that state-of-the-art methods in [2] and [11] and *GOMP* perform similarly. However, the other algorithms perform better. Here again, it is observed that *SoBaP* is the best overall and achieves a BER close to that of the oracle.

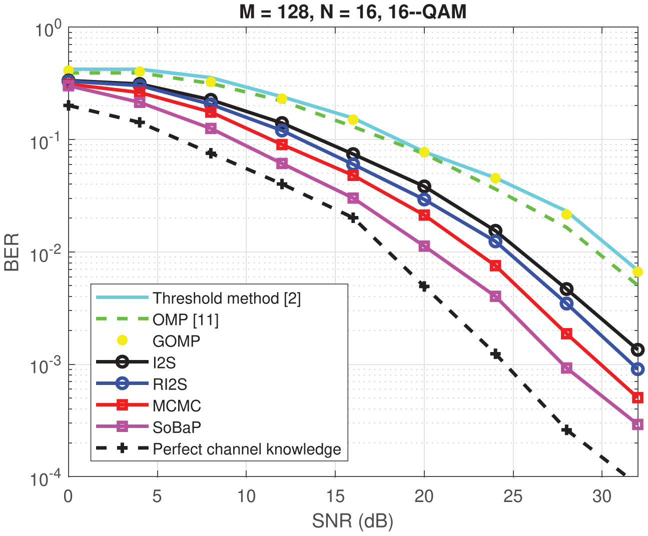

**FIGURE 7** BER versus SNR: low latency communication, for $M = 128$, $N = 16$, $M_p = 16$, $N_p = 4$, $l_\tau = 10$, $k_\nu = 4$ and $16 - QAM$ modulation

### 4.3.4 | BER under low-latency communications

New generations of advanced mobile communications primarily require low-latency communications [39], that is, small $N$. Figure 7 shows the variation of BER versus SNR in the case of a low-latency application. Here, the number of time slots and the number of subcarriers used are $N = 16$ and $M = 128$, which corresponds to a frame duration of 1.1 ms.

It is worth noting that there is a downgrade of the performance of all methods (which is expected in low-latency applications). However, the downgrade for state-of-the-art methods and *GOMP* is more pronounced than that of other algorithms. Here again, *SoBaP* is still the most efficient algorithm.

## 4.4 | Discussion

Several different algorithms for sparse CE in OTFS systems (*GOMP*, *I2S*, *RI2S*, *MCMC* and *SoBaP*) are suggested and compared with the state-of-the-art methods in [2] and [11]. Comparison have been made in terms of complexity, NMSE and BER.

Simulations have shown that *GOMP* and the methods in [2] and [11] have similar lowest complexity. *SoBaP* costs slightly more but is still less expensive than the other algorithms.

Regarding the performance in terms of NMSE and BER, simulations have shown that *GOMP* performs similarly to methods in [2] and [11]. All the other algorithms perform better. Those based on Bayesian approaches (*MCMC* and *SoBaP*) dominate those based on $\ell_1$-minimization (*I2S* and *RI2S*). In particular, *SoBaP* is the best algorithm. It exceeds the two state-of-the-art methods proposed in [2] and [11] by about 8 dB at BER $= 10^{-3}$ and by about 6 dB at NMSE $= -15$ dB.

Consequently, it appears that *SoBaP* offers a good compromise between performance and complexity.

It is worth noting that the proposed algorithms remain interesting even in high-speed scenarios. This is due to the fact that the channel coherence time is higher than the duration of an OTFS symbol for high speeds. Therefore, the channel variations remain slow in Delay-Doppler domain and the interference between adjacent subcarriers has similar properties. For example, with the simulation parameters used in this work and for a relative speed of 500 km/h, the channel coherence time is equal to 0.54 ms. This value is still greater than the duration of an OTFS symbol $T_s = 0.13\ \mu$s.

In future work, it will be worthwhile to test the performance of these algorithms in OTFS systems using Turbo coding [40–42]. Moreover, it will be interesting to adapt these algorithms (in particular *SoBaP*) to the case of superimposed schemes [14–16].

## 5 | CONCLUSION

In this paper, the design of CE algorithms in the Delay-Doppler domain for OTFS systems has been addressed. First, iterative algorithms involving unknown sparse support vector, which corresponds to the delay and Doppler taps, and an unknown vector of channel values have been developed. The two proposed algorithms use an iterative two-stage procedure consisting in optimizing the support and channel gains alternatively. The first resorts to the $\ell_1$-minimization of the support vector. The second uses a reweighted $\ell_1$-minimization to improve the recovery performance. The estimation problem is also addressed from a Bayesian point of view. The structured sparse representation is reformulated as a particular marginalized MAP problem on the support vector. In order to deal with the intractability of this problem, two existing techniques are adapted to this context. The first is based on the *MCMC* with the Gibbs sampler and the second on a specific variational mean-field approximation, as well as the VB-EM procedure. The proposed schemes have been compared to a recognized CE method and to a recent on-grid channel estimation method. Experimental results have shown that the proposed schemes are computationally slightly more expensive but perform significantly better in terms of NMSE and BER and remain applicable in high-mobility scenarios and low-latency applications.

## CONFLICT OF INTEREST

The authors have declared no conflict of interest.

## DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analysed during the current study

## ORCID

*Abdeldjalil Aïssa-El-Bey* https://orcid.org/0000-0002-6267-3118

## REFERENCES

1. Melki, R., Noura, H.N., Mansour, M.M., Chehab, A.: A survey on OFDM physical layer security. Phys. Commun. 32, 1–30 (2019)
2. Raviteja, P., Phan, K.T., Hong, Y.: Embedded pilot-aided channel estimation for OTFS in delay–doppler channels. IEEE Trans. Veh. Technol. 68(5), 4906–4917 (2019)
3. Monk, A., Hadani, R., Tsatsanis, M., Rakib, S.: OTFS-orthogonal time frequency space: A novel modulation technique meeting 5G high mobility and massive mimo challenges. Technical Report. https://arxivorg/ftp/arxiv/papers/1608/160802993pdf
4. Hadani, R., Rakib, S., Tsatsanis, M., Monk, A., Goldsmith, A.J., Molisch, A.F., et al.: Orthogonal time frequency space modulation. In: IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6. IEEE, Piscataway (2017)
5. Zhang, M., Wang, F., Yuan, X., Chen, L.: 2D structured turbo compressed sensing for channel estimation in OTFS systems. In: IEEE International Conference on Communication Systems (ICCS), pp. 45–49. IEEE, Piscataway (2018)
6. Ma, J., Yuan, X., Ping, L.: Turbo compressed sensing with partial DFT sensing matrix. IEEE Signal Process Lett. 22(2), 158–161 (2014)
7. Liu, Y., Zhang, S., Gao, F., Ma, J., Wang, X.: Uplink-aided high mobility downlink channel estimation over massive MIMO-OTFS system. IEEE J. Sel. Areas Commun. 38(9), 1994–2009 (2020)
8. Shen, W., Dai, L., An, J., Fan, P., Heath, R.W.: Channel estimation for orthogonal time frequency space (OTFS) massive MIMO. IEEE Trans. Signal Process. 67(16), 4204–4217 (2019)
9. Ramachandran, M.K., Chockalingam, A.: MIMO-OTFS in high-doppler fading channels: Signal detection and channel estimation. In: IEEE Global Communications Conference (GLOBECOM), pp. 206–212. IEEE, Piscataway (2018)
10. Raviteja, P., Phan, K.T., Hong, Y., Viterbo, E.: Interference cancellation and iterative detection for orthogonal time frequency space modulation. IEEE Trans. Wireless Commun. 17(10), 6501–6515 (2018)
11. Wang, S., Guo, J., Wang, X., Yuan, W., Fei, Z.: Pilot design and optimization for OTFS modulation. IEEE Wireless Commun. Lett. 10(8), 1742–1746 (2021)
12. Srivastava, S., Singh, R.K., Jagannatham, A.K., Hanzo, L.: Bayesian learning aided sparse channel estimation for orthogonal time frequency space modulated systems. IEEE Trans. Veh. Technol. 70(8), 8343–8348 (2021)
13. Wei, Z., Yuan, W., Li, S., Yuan, J., Ng, D.W.K.: Off-grid channel estimation with sparse bayesian learning for OTFS systems. IEEE Trans. Wireless Commun. (2022)
14. Yuan, W., Li, S., Wei, Z., Yuan, J., Ng, D.W.K.: Data-aided channel estimation for OTFS systems with a superimposed pilot and data transmission scheme. IEEE Wireless Commun. Lett. 10(9), 1954–1958 (2021)
15. Mishra, H.B., Singh, P., Prasad, A.K., Budhiraja, R.: OTFS channel estimation and data detection designs with superimposed pilots. IEEE Trans. Wireless Commun. (2021)
16. Ouchikh, R., Aïssa El Bey, A., Chonavel, T., Djeddou, M.: Iterative channel estimation and data detection algorithm for OTFS modulation. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5263–5267. IEEE, Piscataway (2022)
17. Wang, J., Kwon, S., Shim, B.: Generalized orthogonal matching pursuit. IEEE Trans. Signal Process. 60(12), 6202–6216 (2012)
18. Salahdine, F., Kaabouch, N., El Ghazi, H.: A survey on compressive sensing techniques for cognitive radio networks. Phys. Commun. 20, 61–73 (2016)
19. Raviteja, P., Hong, Y., Viterbo, E., Biglieri, E.: Practical pulse-shaping waveforms for reduced-cyclic-prefix OTFS. IEEE Trans. Veh. Technol. 68(1), 957–961 (2018)
20. Raviteja, P., Viterbo, E., Hong, Y.: OTFS performance on static multipath channels. IEEE Wireless Commun. Lett. 8(3), 745–748 (2019)
21. Cai, T.T., Wang, L.: Orthogonal matching pursuit for sparse signal recovery with noise. IEEE Trans. Inf. Theory 57(7), 4680–4688 (2011)
22. Ma, Z., Peng, Q., Liu, H., Yan, Z.: An improved channel estimation method using doubly reduced matching pursuit over fractional delay multipath channel. Phys. Commun. 29, 86–94 (2018)
23. Gavish, M., Donoho, D.L.: The optimal hard threshold for singular values is $4/\sqrt{3}$. IEEE Trans. Inf. Theory 60(8), 5040–5053 (2014)
24. Candes, E.J., Wakin, M.B., Boyd, S.P.: Enhancing sparsity by reweighted $\ell_1$ minimization. J. Fourier Anal. Appl. 14(5-6), 877–905 (2008)

25. Drémeau, A., Herzet, C., Daudet, L.: Boltzmann machine and mean-field approximation for structured sparse decompositions. IEEE Trans. Signal Process. 60(7), 3425–3438 (2012)
26. Levy, B.C.: Principles of signal detection and parameter estimation. Springer Science & Business Media, New York (2008)
27. Wainwright, M.J., Jordan, M.I.: Variational inference in graphical models: The view from the marginal polytope. In: Proceedings of the Annual Allerton Conference on Communication Control and Computing, vol. 41, pp. 961–971. IEEE, Piscataway (2003)
28. Rabaste, O., Chonavel, T.: Estimation of multipath channels with long impulse response at low SNR via an MCMC method. IEEE Trans. Signal Process. 55(4), 1312–1325 (2007)
29. Beal, M.J.: Variational algorithms for approximate Bayesian inference. UCL University College, London (2003)
30. Van Erven, T., Harremos, P.: Rényi divergence and kullback-leibler divergence. IEEE Trans. Inf. Theory 60(7), 3797–3820 (2014)
31. Minka, T.P.: A family of algorithms for approximate Bayesian inference. Massachusetts Institute of Technology, Cambridge (2001)
32. Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., et al.: The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures. Bayesian Stat. 7(453-464), 210 (2003)
33. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
34. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Royal Stat. Soc.: Series B (Methodolog.) 39(1), 1–22 (1977)
35. Neal, R.M., Hinton, G.E.: A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Learning in Graphical Models, pp. 355–368. Springer, Dordrecht (1998)
36. Grant, M., Boyd, S.: Cvx: Matlab software for disciplined convex programming, version 2.1 (2014)
37. Grant, M.C., Boyd, S.P.: Graph implementations for nonsmooth convex programs. In: Recent Advances in Learning and Control, pp. 95–110. Springer, London (2008)
38. U.E.U. conformance specification Radio, 3rd generation partnership project; technical specification group radio access network; evolved universal terrestrial radio access (E-UTRA); user equipment (UE) conformance specification radio transmission and reception (2011)
39. Dey, P., Masal, A., Kaimalettu, S., Milleth, J.K., Ramamurthi, B.: Reference signal design to mitigate co-channel interference in 5G OFDM systems with multiple numerologies. Phys. Commun. 53, 101653 (2022)
40. Vasudevan, K.: Coherent detection of turbo-coded OFDM signals transmitted through frequency selective rayleigh fading channels with receiver diversity and increased throughput. Wirel. Personal Commun. 82(3), 1623–1642 (2015)
41. Vasudevan, K.: Near capacity signaling over fading channels using coherent turbo coded OFDM and massive MIMO. arXiv preprint arXiv:171110104 (2017)
42. Li, S., Yuan, J., Yuan, W., Wei, Z., Bai, B., Ng, D.W.K.: Performance analysis of coded OTFS systems over high-mobility channels. IEEE Trans. Wireless Commun. 20(9), 6033–6048 (2021)

**How to cite this article:** Ouchikh, R., Aïssa-El-Bey, A., Chonavel, T., Djeddou, M.: Sparse channel estimation algorithms for OTFS system. IET Commun. 1–13 (2022). https://doi.org/10.1049/cmu2.12469

# APPENDICES

## A.1 | Gibbs sampler

The Gibbs sampler is implemented as follows:

- Initialization: $\boldsymbol{b} = \boldsymbol{b}^{(0)}$ and $\boldsymbol{g} = \boldsymbol{g}^{(0)}$,
- For $i \geq K$ and for $k$ randomly covering $\{1, \dots, L\}$
  - Detection step :

$$\text{calculate: } d_k = p(b_k = 1 \lfloor \boldsymbol{y}_p, \boldsymbol{\theta}_{-k}^{(i)}),$$
$$\text{draw: } b_K^{(i)} = \mathbf{1}_{d_k > u}, \text{ with } u \sim \mathcal{U}_{[0,1]},$$

  - Estimation step :

$$\text{draw: } g_k^{(i)} \sim \mathcal{CN}(g_k^{(i)}; m_{b_k^{(i)}}, \sigma^2_{b_k^{(i)}}),$$

where $\mathcal{U}_I$ is the uniform distribution of $I$, and

$$\mathbf{1}_I(t) = \begin{cases} 1 & \text{if } t \in I, \\ 0 & \text{otherwise.} \end{cases} \tag{A.1}$$

## A.2 | VB-EM update

Particularized to model (21), the corresponding VB-EM update is given as:

$$q(g_k, b_k) = q(g_k|b_k)q(b_k), \tag{A.2}$$

where

$$q(g_k|b_k) = \mathcal{N}(m(b_k), \Sigma(b_k)) \tag{A.3}$$

$$q(b_k) \propto \sqrt{\Sigma(b_k)} e^{\left(\frac{1}{2}\frac{m(b_k)^2}{\Sigma(b_k)}\right)} p(b_k), \tag{A.4}$$

and

$$\Sigma(b_k) = \frac{\sigma_{g_k}^2 \sigma^2}{\sigma^2 + b_k \sigma_{g_k}^2 \boldsymbol{A}_k^T \boldsymbol{A}_k}, \tag{A.5}$$

$$m(b_k) = b_k \frac{\sigma_{g_k}^2}{\sigma^2 + b_k \sigma_{g_k}^2 \boldsymbol{A}_k^T \boldsymbol{A}_k} \boldsymbol{r}_k^T \boldsymbol{A}_k, \tag{A.6}$$

$$\boldsymbol{r}_k = \boldsymbol{y}_p - \sum_{l \neq k} q(b_l = 1) m(b_l = 1) \boldsymbol{A}_l. \tag{A.7}$$

## A.3 | Estimation of $\sigma^2$

The estimate of $\sigma^2$ is therefore given as follows:

$$\begin{aligned} \hat{\sigma}^2 &= \arg\max_{\sigma^2} \left\{ \sum_{\boldsymbol{b}} \int \prod_k q(g_k, b_k) \log(p(\boldsymbol{g}, \boldsymbol{b}, \boldsymbol{y}_p | \sigma^2)) \right\} \\ &= \frac{1}{L} \langle \| \boldsymbol{y}_p - \sum_k b_k g_k \boldsymbol{A}_k \|^2 \rangle_{\prod_k q(g_k, b_k)} \\ &= \frac{1}{L} (\boldsymbol{y}_p^T \boldsymbol{y}_p - 2 \sum_k q(b_k = 1) m(b_k = 1) \boldsymbol{y}_p^T \boldsymbol{A}_k \\ &+ \sum_k \sum_{l \neq k} q(b_k = 1) q(b_l = 1) m(b_k = 1) m(b_l = 1) \boldsymbol{A}_k^T \boldsymbol{A}_l \\ &+ \sum_k q(b_k = 1)(\Sigma(b_k = 1) + m(b_k = 1)^2) \boldsymbol{A}_k^T \boldsymbol{A}_k). \end{aligned} \tag{A.8}$$