On the Sphere Decoder for Frequency-Selective MIMO channels

Barbero, L., Ratnarajah, T., & Cowan, C. (2008). On the Sphere Decoder for Frequency-Selective MIMO channels. *IEEE Transactions on Signal Processing*, 56(12), 6031-6043. [4627464].
<https://doi.org/10.1109/TSP.2008.2005745>

Published in:
IEEE Transactions on Signal Processing

Queen's University Belfast - Research Portal:
[Link to publication record in Queen's University Belfast Research Portal](#)

General rights

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

On the Complexity of the Sphere Decoder for Frequency-Selective MIMO Channels

Luis G. Barbero, *Member, IEEE*, T. Ratnarajah, *Senior Member, IEEE*, and Colin F. N. Cowan, *Senior Member, IEEE*

Abstract—This paper compares the complexity of the sphere decoder (SD) and a previously proposed detection scheme, denoted here as block SD (BSD), when they are applied to the detection of multiple-input multiple-output (MIMO) systems in frequency-selective channels. The complexity of both algorithms depends on their preprocessing and tree search stages. Although the BSD was proposed as a means of greatly reducing the complexity of the preprocessing stage of the SD, no study was done on how the complexity of the tree search stage could be affected by that reduced preprocessing stage. This paper shows, both analytically and through simulation, that the reduction in preprocessing complexity provided by the BSD has the side effect of increasing the complexity of its tree search stage compared to that of the SD, independent of the signal-to-noise ratio (SNR). In addition, this paper shows how sorting the columns of the frequency-selective channel matrix in the SD does not reduce the complexity of the tree search stage, contrary to what occurs in frequency-flat channels.

Index Terms—Detection, frequency-selective channel, multiple-input multiple-output (MIMO), sphere decoder (SD).

I. INTRODUCTION

THE use of multiple-input multiple-output (MIMO) technology has emerged as one of the most relevant technical breakthroughs in modern wireless communications after theoretical analysis showed that significant capacity increase could be achieved under certain conditions by using multiple antennas at both transmitter and receiver [1]. The design of the detection algorithms at the receiver is one of the major challenges of MIMO systems, especially in high data-rate applications, where the frequency-selective channel induces intersymbol interference (ISI) at the receiver in addition to the multiple antenna interference (MAI) and the noise. Although orthogonal frequency division multiplexing (OFDM) can be used to transform the frequency-selective channel in a banded flat fading channel, that solution does not exploit the maximum diversity of the channel [2]. In addition, OFDM systems are sensitive to the effect of phase noise, intercarrier interference (ICI) and nonlinearities in the transmitter power amplifiers [3]. As an alternative, single carrier systems can be used, requiring equalization techniques to mitigate the ISI.

Manuscript received April 08, 2008; revised August 06, 2008. First published September 19, 2008; current version published November 19, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Geert Leus. This work was supported by the U.K. Engineering and Physical Sciences Research Council under Grant EP/C004132/1.

The authors are with the Institute of Electronics, Communications and Information Technology (ECIT), Queen's University of Belfast, BT3 9DT Belfast, U.K. (e-mail: l.barbero@ecit.qub.ac.uk).

Digital Object Identifier 10.1109/TSP.2008.2005745

In order to achieve the optimal maximum-likelihood (ML) performance, the maximum-likelihood sequence estimation (MLSE) can be used, although its complexity is exponential with the product of the number of antennas and the number of channel taps [4]. Recently, the sphere decoder (SD) has been proposed for frequency-selective MIMO channels, also achieving the optimal performance [5]. The SD was first introduced as a means of obtaining vectors of minimal length in a lattice, with applications to algebraic number theory or cryptography [6]. It was later applied to solve the detection problem in wireless communications through fading channels, achieving the same performance as the optimal maximum-likelihood detector (MLD) [7]. Finally, the SD became extremely popular after it was explicitly applied to MIMO systems [8], [9], sparking a large amount of research both from a theoretical and from an implementation point of view. Although different detection schemes have been proposed to approach the performance of the SD in frequency-selective channels [10]–[12], the SD remains the most promising alternative given its ML performance and its suitability for hardware implementation [13], despite its exponential lower-bound in the complexity for high number of antennas and constellation orders [14].

The complexity of the SD in frequency-selective MIMO channels is determined by the complexity of its preprocessing and its tree search stages. This paper analyzes that complexity for the SD and a variant of the SD, denoted as block SD (BSD) in the sequel, briefly proposed in [5]. In particular, the BSD was proposed as a means of reducing the complexity of the SD by performing a lower-complexity preprocessing stage. Although the BSD greatly reduces the preprocessing complexity of the SD, no attention was paid in [5] to the resulting tree search complexity in the BSD. This paper shows that the reduced preprocessing stage of the BSD has a negative effect on its corresponding tree search stage. The complexity of the tree search stage of the BSD increases beyond that of the SD counterpart, resulting in a larger overall complexity in the low to medium signal-to-noise ratio (SNR) regime. Analytical and simulation results indicate that, among the two algorithms, the SD is, in most cases, the most promising alternative to achieve ML performance. In addition, simulation results show how, contrary to case of frequency-flat channels, sorting the columns of the channel matrix does not reduce the complexity of the SD tree search. To the best of our knowledge, this last result has not been previously reported in the literature.

The rest of the paper is organized as follows. Section II describes the frequency-selective MIMO system model. Section III introduces the SD and the BSD applied to frequency-selective MIMO channels. Section IV concentrates on

the complexity analysis of the SD and the BSD, looking at the preprocessing and the tree search stages. Complexity results for different scenarios are presented in Section V. Finally, conclusions are drawn in Section VI.

II. SYSTEM MODEL

We consider a complex-valued baseband MIMO system with M transmit and N receive antennas, denoted as $M \times N$, in a block-fading frequency-selective propagation environment as proposed in [5] and [10]–[12]. Thus, the MIMO channel is constant over the duration of a transmission block and changes independently between blocks. The channel between transmitter j and receiver i is modelled as a linear FIR filter with L independent symbol-spaced taps, denoted as $\mathbf{h}_{i,j} = [h_{i,j}(0), h_{i,j}(1), \dots, h_{i,j}(L-1)]^T$, where $h_{i,j}(l) \sim \mathcal{CN}(0, \sigma_l^2)$ for $i = 1, \dots, N$, $j = 1, \dots, M$ and σ_l is set according to the power delay profile (PDP) of the channel for $l = 0, \dots, L-1$, such that $\mathbb{E}[\mathbf{h}_{i,j}^H \mathbf{h}_{i,j}] = 1$. For notation purposes, we consider the transmission of MT symbols over a span of T symbol-spaced time instants, resulting in a block size T . The vector of transmitted symbols per block is denoted as $\mathbf{x} = [\bar{\mathbf{x}}^T(0), \bar{\mathbf{x}}^T(1), \dots, \bar{\mathbf{x}}^T(T-1)]^T$, where the vector $\bar{\mathbf{x}}(k) = [\bar{x}_1(k), \bar{x}_2(k), \dots, \bar{x}_M(k)]^T$ contains the symbols transmitted from all M antennas at time instant $k = 0, \dots, T-1$. The symbols are taken independently from an arbitrary constellation $\mathcal{O}$ of P points and the total transmitted energy is independent of the number of transmit antennas, i.e., $\mathbb{E}[\mathbf{x}\mathbf{x}^H] = M^{-1}\mathbf{I}_{MT}$ where $\mathbf{I}_{MT}$ denotes the $MT \times MT$ identity matrix. We further assume a system with no inter-block interference so that $L-1$ zeros are appended per transmit antenna at the end of each block. Correspondingly, the vector of received symbols spans over $K = T + L - 1$ time instants and can be written as $\mathbf{y} = [\bar{\mathbf{y}}^T(0), \bar{\mathbf{y}}^T(1), \dots, \bar{\mathbf{y}}^T(K-1)]^T$ where the vector $\bar{\mathbf{y}}(k) = [\bar{y}_1(k), \bar{y}_2(k), \dots, \bar{y}_N(k)]^T$ contains the symbols at the N receive antennas at time instant $k = 0, \dots, K-1$. Thus, assuming symbol-synchronous sampling at the receiver and ideal timing, the vector of received symbols can be written as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (1)$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}(0) & & & \\ \vdots & \ddots & & \\ \mathbf{H}(L-1) & \cdots & \mathbf{H}(0) & \\ & \ddots & \vdots & \ddots \\ & & \mathbf{H}(L-1) & \cdots & \mathbf{H}(0) \\ & & & \ddots & \vdots \\ & & & & \mathbf{H}(L-1) \end{bmatrix} \quad (2)$$

and

$$\mathbf{H}(l) = \begin{bmatrix} h_{1,1}(l) & \cdots & h_{1,M}(l) \\ \vdots & \ddots & \vdots \\ h_{N,1}(l) & \cdots & h_{N,M}(l) \end{bmatrix}$$

for $l = 0, \dots, L-1$. The vector $\mathbf{v} = [\bar{\mathbf{v}}^T(0), \bar{\mathbf{v}}^T(1), \dots, \bar{\mathbf{v}}^T(K-1)]^T$, where $\bar{\mathbf{v}}(k) = [\bar{v}_1(k), \bar{v}_2(k), \dots, \bar{v}_N(k)]^T$ for $k = 0, \dots, K-1$, contains independent complex Gaussian noise samples, where $\mathbb{E}[\mathbf{v}\mathbf{v}^H] = \sigma^2\mathbf{I}_{NK}$. Finally, we assume that the channel is perfectly known at the receiver and that $N \geq M$.

III. SPHERE DECODER

The optimum detector for the system described in Section II and defined by (1) is the MLD, whose solution is given by

$$\hat{\mathbf{x}}_{\text{ML}} = \arg \min_{\mathbf{x} \in \mathcal{O}^{MT}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2.$$

However, for the system under consideration, the MLD has an exponential complexity with the total number of transmitted symbols $O(P^{MT})$, which makes the MLD prohibitively complex for practical scenarios. Even the Viterbi algorithm, which yields the same performance and makes use of the Markovian properties of the channel [4], also has an exponential complexity with the product of the number of transmit antennas and the number of channel taps $O(P^{ML})$. In order to reduce that complexity, the SD, previously proposed for flat fading MIMO channels [9], has recently been proposed for frequency-selective MIMO channels [5]. The SD reduces the complexity of the MLD although, strictly speaking, it still has an exponential complexity (in the worst case and in the average case) of $O(P^{\gamma MT})$ with $\gamma \in (0, 1)$ [14]. This paper concentrates on the complex-valued version of the SD that requires no real-valued decomposition of the system, can be applied to an arbitrary constellation [15] and results in a more optimized hardware implementation [13].

The main idea behind the SD is to reduce the complexity of the MLD by searching over only the vectors $\mathbf{x}$ that satisfy

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \leq R^2 \quad (3)$$

i.e., searching over the lattice vectors $\mathbf{H}\mathbf{x}$ that lie within a hypersphere of radius R around the received vector $\mathbf{y}$.

Making use of the QR decomposition of $\mathbf{H} = \mathbf{Q}[\mathbf{R}^T \mathbf{0}]^T$ [9], where $\mathbf{Q}$ is an $NK \times NK$ unitary matrix, $\mathbf{R}$ is an $MT \times MT$ upper triangular matrix with entries $r_{i,j}$ for $i, j = 1, \dots, MT$, $\mathbf{0}$ is a 0-matrix of the appropriate size, and partitioning $\mathbf{Q} = [\mathbf{Q}_1 \mathbf{Q}_2]$ into two submatrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$, containing the first MT columns and the last $NK - MT$ columns of $\mathbf{Q}$, respectively, the metric calculation in (3) can be rewritten as

$$\|\mathbf{Q}_1^H \mathbf{y} - \mathbf{R}\mathbf{x}\|^2 + \|\mathbf{Q}_2^H \mathbf{y}\|^2 \leq R^2$$

and can be further simplified as

$$\|\mathbf{w} - \mathbf{R}\mathbf{x}\|^2 \leq R^2 \quad (4)$$

where $\mathbf{w} = \mathbf{Q}_1^H \mathbf{y}$ and the term $\|\mathbf{Q}_2^H \mathbf{y}\|^2$ has been dropped since it is constant for each received vector $\mathbf{y}$ and independent of the transmitted vector $\mathbf{x}$.

The solution to (4) can be obtained recursively taking into account that the SD is equivalent to a constrained *depth-first* tree search through a tree with MT levels where P branches

originate from each node on the tree [16]. Thus, the vectors that satisfy (4) can be obtained recursively starting from $i = MT$ and traversing down the tree until $i = 1$. For each level $i = MT, \dots, 1$, only the branches associated to the constellation points x_i that satisfy

$$|z_i - r_{i,i}x_i|^2 \leq R_i^2 \quad (5)$$

are visited during the search process, noting that

$$z_i = w_i - \sum_{j=i+1}^{MT} r_{i,j}x_j \quad (6)$$

with $z_{MT} = w_{MT}$, and

$$R_i^2 = R^2 - \sum_{j=i+1}^{MT} |z_j - r_{j,j}x_j|^2 \quad (7)$$

with $R_{MT}^2 = R^2$.

If a complete lattice vector (i.e., multidimensional lattice point), at $i = 1$, is found inside the hypersphere, the radius is updated with its associated metric and the algorithm continues the search until no more lattice vectors are found inside the current hypersphere. At that point, the vector $\mathbf{x}$ corresponding to the last lattice vector found inside the hypersphere is the ML solution. The reduction in complexity of the SD compared to an exhaustive search comes from the fact that, at any level i , if $R_i^2 < 0$ or no branches satisfy (5), the subtrees originating from those branches can be discarded.

It should be noted that the order in which the branches are visited during the search process is relevant to the final complexity of the SD. Two possible enumeration techniques exist, Fincke–Pohst (FP) [6] and Schnorr–Euchner (SE) [17], with the latter resulting in a lower complexity [9]. The SE enumeration consists of sorting the branches in increasing order of their associated metrics $|z_i - r_{i,i}x_i|^2$, such that the branches with the lowest metric are visited first. That requires direct calculation of the P $|z_i - r_{i,i}x_i|^2$ values or the application of the method presented in [15]. The SE enumeration also has the additional advantage of reducing the dependability of the complexity of the SD on the initial radius R . In fact, the initial radius R can be set to infinite and be immediately updated with the metric associated to the first lattice vector found, which corresponds to the Babai solution [18].

The SD can also use the QL decomposition of the channel matrix $\mathbf{H}$ to perform the tree search from $i = 1$ and traverse the tree in the opposite direction [19], [20]. Both alternatives would give the same ML performance and have the same complexity if all the elements of the channel matrix $\mathbf{H}$ have exactly the same distribution. This would be the case in flat fading scenarios and frequency-selective scenarios with a uniform PDP. However, in the more realistic case of an exponential PDP, the QL decomposition can yield a slight complexity increase in high-dimensional systems, as will be shown in Section V.

A. Block Sphere Decoder

In [5], the authors briefly proposed a modification of the SD for frequency-selective MIMO channels, denoted BSD in this

paper, to reduce the computational complexity of the QR decomposition. The BSD performs the same metric calculation as in (3) but removes the need for a full QR decomposition of $\mathbf{H}$ by making use of its block Toeplitz structure, with blocks of size $N \times M$ as given by (2). In fact, only a QR decomposition of the block $\mathbf{H}(L-1)$ is required before the sphere decoder algorithm can be applied [5]. However, the algorithm is only mentioned in [5], with no analysis of the claimed improvement in complexity. A more detailed description of the BSD is given in this section in order to analyze its complexity in Section IV.

Consider the partition of the channel matrix $\mathbf{H} = [\mathbf{H}_1^T \mathbf{H}_2^T]^T$ where $\mathbf{H}_1$ is an $N(L-1) \times MT$ matrix

$$\mathbf{H}_1 = \begin{bmatrix} \mathbf{H}(0) & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{H}(L-2) & \cdots & \mathbf{H}(0) & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$$

and $\mathbf{H}_2$ is an $NT \times MT$ block upper triangular matrix with blocks of size $N \times M$

$$\mathbf{H}_2 = \begin{bmatrix} \mathbf{H}(L-1) & \cdots & \mathbf{H}(0) & \mathbf{0} \\ & \ddots & \vdots & \ddots \\ & & \mathbf{H}(L-1) & \cdots & \mathbf{H}(0) \\ & & & \ddots & \vdots \\ \mathbf{0} & & & & \mathbf{H}(L-1) \end{bmatrix}.$$

The (squared) Euclidean distance calculation performed by the BSD can then be written as

$$\underbrace{\|\mathbf{y}_1 - \mathbf{H}_1\mathbf{x}\|^2}_{d_1} + \underbrace{\|\mathbf{y}_2 - \mathbf{H}_2\mathbf{x}\|^2}_{d_2} \leq R^2 \quad (8)$$

where $\mathbf{y}_1 = [\bar{\mathbf{y}}^T(0), \bar{\mathbf{y}}^T(1), \dots, \bar{\mathbf{y}}^T(L-2)]^T$ and $\mathbf{y}_2 = [\bar{\mathbf{y}}^T(L-1), \bar{\mathbf{y}}^T(L), \dots, \bar{\mathbf{y}}^T(K-1)]^T$.

To solve (8), the BSD transforms d_2 to obtain an upper triangular structure, performs a tree search from $i = MT$ to $i = 1$, considering only the term d_2 , and adds the term d_1 at level $i = 1$ to obtain the total metric and check whether the lattice vector lies inside the hypersphere [5]. The BSD reduces the preprocessing complexity of the SD by applying the QR decomposition only to $\mathbf{H}(L-1)$, instead of $\mathbf{H}$ [5]. However, the effect that preprocessing has on the tree search complexity has not been studied.

In particular, making use of the QR decomposition of $\mathbf{H}(L-1) = \tilde{\mathbf{Q}}[\tilde{\mathbf{R}}^T \mathbf{0}]^T$, where $\tilde{\mathbf{Q}}$ is an $N \times N$ unitary matrix, $\tilde{\mathbf{R}}$ is an $M \times M$ upper triangular matrix with entries $\tilde{r}_{i,j}$ for $i, j = 1, \dots, M$, $\mathbf{0}$ is a 0-matrix of the appropriate size, and partitioning $\tilde{\mathbf{Q}} = [\tilde{\mathbf{Q}}_1 \tilde{\mathbf{Q}}_2]$ into two submatrices $\tilde{\mathbf{Q}}_1$ and $\tilde{\mathbf{Q}}_2$, containing the first M columns and the last $N-M$ columns of $\tilde{\mathbf{Q}}$, respectively, the term d_2 in (8) can be rewritten as

$$d_2 = \|\mathbf{Q}'_1 \mathbf{y}' - \mathbf{R}'\mathbf{x}\|^2 + \|\mathbf{Q}'_2 \mathbf{y}'\|^2$$

where $\mathbf{Q}'_1$ and $\mathbf{Q}'_2$ are $NT \times MT$ and $NT \times (N-M)T$ block diagonal matrices with $N \times M$ and $N \times (N-M)$ blocks, respectively

$$\mathbf{Q}'_1 = \begin{bmatrix} \tilde{\mathbf{Q}}_1 & \mathbf{0} \\ & \ddots \\ \mathbf{0} & \tilde{\mathbf{Q}}_1 \end{bmatrix}, \quad \mathbf{Q}'_2 = \begin{bmatrix} \tilde{\mathbf{Q}}_2 & \mathbf{0} \\ & \ddots \\ \mathbf{0} & \tilde{\mathbf{Q}}_2 \end{bmatrix}$$

$\mathbf{R}'$ is an $MT \times MT$ block diagonal matrix with $M \times M$ blocks

$$\mathbf{R}' = \begin{bmatrix} \tilde{\mathbf{R}} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \tilde{\mathbf{R}} \end{bmatrix}$$

with entries $r'_{i,j}$ for $i, j = 1, \dots, MT$, and $\mathbf{y}' = [\bar{\mathbf{y}}'^T(0), \bar{\mathbf{y}}'^T(1), \dots, \bar{\mathbf{y}}'^T(T-1)]^T$, where $\bar{\mathbf{y}}'(k) = [\bar{y}'_1(k), \bar{y}'_2(k), \dots, \bar{y}'_N(k)]^T$ for $k = 1, \dots, T-1$, is obtained as

$$\mathbf{y}' = \mathbf{y}_2 - \mathbf{H}_3 \mathbf{x} \quad (9)$$

with $\mathbf{H}_3$ being an $NT \times MT$ block upper triangular matrix with $N \times M$ 0-matrix blocks at the main diagonal, i.e.,

$$\mathbf{H}_3 = \begin{bmatrix} \mathbf{0} & \mathbf{H}(L-2) & \cdots & \mathbf{H}(0) & \mathbf{0} \\ & \ddots & \ddots & \vdots & \ddots \\ & & \mathbf{0} & \mathbf{H}(L-2) & \cdots & \mathbf{H}(0) \\ & & & \ddots & \ddots & \vdots \\ \mathbf{0} & & & & \mathbf{0} & \mathbf{H}(L-2) \\ & & & & & \mathbf{0} \end{bmatrix}.$$

Thus, considering only d_2 in (8), the tree search is performed following the constraint:

$$\|\mathbf{w}' - \mathbf{R}'\mathbf{x}\|^2 + \|\mathbf{Q}_2'^H \mathbf{y}'\|^2 \leq R^2 \quad (10)$$

where $\mathbf{w}' = \mathbf{Q}_1'^H \mathbf{y}' = [\bar{\mathbf{w}}'^T(0), \bar{\mathbf{w}}'^T(1), \dots, \bar{\mathbf{w}}'^T(T-1)]^T$, with $\bar{\mathbf{w}}'(k) = [\bar{w}'_1(k), \bar{w}'_2(k), \dots, \bar{w}'_M(k)]^T$ for $k = 1, \dots, T-1$. Noting that $\|\mathbf{Q}_2'^H \mathbf{y}'\|^2$ can also be written as

$$\|\mathbf{Q}_2'^H \mathbf{y}'\|^2 = \sum_{k=0}^{T-1} \|\tilde{\mathbf{Q}}_2^H \bar{\mathbf{y}}'(k)\|^2 \quad (11)$$

it can be observed that, from the definition of $\mathbf{y}'$ in (9), only the term $\bar{\mathbf{y}}'(T-1)$ is independent of the transmitted vector $\mathbf{x}$. Thus, contrary to the SD, only the last term of the sum in (11) can be dropped from the metric calculation in (10), yielding

$$\|\mathbf{w}' - \mathbf{R}'\mathbf{x}\|^2 + \sum_{k=0}^{T-2} \|\tilde{\mathbf{Q}}_2^H \bar{\mathbf{y}}'(k)\|^2 \leq R^2. \quad (12)$$

Therefore, during the tree search, for each level $i = MT, \dots, 1$, only the branches associated to the constellation points x_i that satisfy

$$|z'_i - r'_{i,i}x_i|^2 \leq R_i'^2 \quad (13)$$

are visited during the search process, noting that

$$z'_i = w'_i - \sum_{j=i+1}^{MT} r'_{i,j}x_j \quad (14)$$

with $z'_{MT} = w'_{MT}$, and

$$R_i'^2 = R^2 - \sum_{j=i+1}^{MT} |z'_j - r'_{j,j}x_j|^2 - \sum_{k=\lceil i/M \rceil - 1}^{T-2} \|\tilde{\mathbf{Q}}_2^H \bar{\mathbf{y}}'(k)\|^2 \quad (15)$$

with $R_{MT}'^2 = R^2$, resulting in a tree search equivalent to that of the SD. The only difference, lies on the fact that at level $i = 1$, the contribution of the term d_1 needs to be considered to check whether a lattice vector lies inside the hypersphere.

The BSD can also use the QL decomposition of the channel matrix $\mathbf{H}(0)$ to perform the tree search from $i = 1$ and traverse the tree in the opposite direction, as has been stated for the SD in the previous section. Although both alternatives, QR and QL decomposition, would give the same ML performance, the QL decomposition yields a lower complexity in the case of exponential PDP, as will be shown in Section V.

It should be noted that the BSD concept was introduced in [5] in the context of the equivalent real decomposition of the MIMO system. However, the BSD cannot be applied in that case given that the special block-Toeplitz structure of $\mathbf{H}$ is not maintained in the equivalent real channel matrix. This factor, together with the reasons outlined for the SD in Section III have made us concentrate on the complex-valued MIMO system for the complexity analysis of the next section.

IV. COMPLEXITY ANALYSIS

This section analyzes the complexity of the SD and the BSD looking at both the preprocessing stage and the tree search stage. Thus, the overall complexity of both algorithms is studied looking at the effect the reduction in complexity of the preprocessing stage of the BSD has on the complexity of its tree search stage. The total number of real operations has been considered for the complexity analysis.¹ For simplicity, all the operations have the same relevance on the final operation count. Finally, it should be noted that the complexity analysis in this section has been done assuming a QR decomposition, although the analysis can equivalently be applied to the QL decomposition.

A. Preprocessing Stage of the SD

The following operations are performed during preprocessing stage of the SD:

- 1) the QR decomposition of the $NK \times MT$ matrix $\mathbf{H}$ to obtain $\mathbf{Q}_1$ and $\mathbf{R}$;
- 2) the computation of $\mathbf{w} = \mathbf{Q}_1^H \mathbf{y}$.

In order to study the complexity of the QR decomposition, we consider the use of the modified Gram-Schmidt (MGS) method [19], reproduced in Algorithm 1 for completeness, where, in general, $\mathbf{A}(:, i)$ denotes the i th column of $\mathbf{A}$. It should be noted that other QR decomposition methods exist, namely Householder transformation or Givens rotation [19], but the MGS has been considered due to the tractability of its operations from a complexity analysis point of view. Although the *absolute* complexity of the three methods is different, the main motivation of this paper is to assess the *relative* complexity between the SD and the BSD, which we can assume to be similar, independently of the method used. For the analysis,

¹We assume that a complex product requires 4 real products and 2 real additions, a complex addition requires 2 real additions and a complex by real division requires 2 real divisions.

Algorithm 1: $(\mathbf{Q}_1, \mathbf{R}) = \mathbf{QR}(\mathbf{H})$

```

1   $\mathbf{Q}_1 = \mathbf{H}$ 
2   $r_{1,1} = \|\mathbf{Q}_1(:, 1)\|$ 
3   $\mathbf{Q}_1(:, 1) = \mathbf{Q}_1(:, 1)/r_{1,1}$ 
4  for  $i = 1, \dots, MT - 1$ 
5      for  $j = i + 1, \dots, MT$ 
6           $r_{i,j} = \mathbf{Q}_1(:, i)^H \mathbf{Q}_1(:, j)$ 
7           $\mathbf{Q}_1(:, j) = \mathbf{Q}_1(:, j) - r_{i,j} \mathbf{Q}_1(:, i)$ 
8      end
9       $r_{i+1,i+1} = \|\mathbf{Q}_1(:, i+1)\|$ 
10      $\mathbf{Q}_1(:, i+1) = \mathbf{Q}_1(:, i+1)/r_{i+1,i+1}$ 
11 end

```

we divide the operations of the QR decomposition between the following:

- norm operations (NO), which account for the operations of lines 2, 3, 9, and 10 in Algorithm 1;
- inner operations (IO), which account for the operations of lines 6 and 7 in Algorithm 1.

Thus, the number of operations of the QR decomposition is given by $N_{\text{SD-QR}} = N_{\text{SD-NO}} + N_{\text{SD-IO}}$. The number of operations of the preprocessing stage can then be written as $N_{\text{SD-pre}} = N_{\text{SD-QR}} + N_{\text{SD-w}}$, where $N_{\text{SD-w}}$ denotes the number of operations required in the computation of $\mathbf{w}$.

Taking into account the 0-matrix blocks in the block-Toeplitz matrix $\mathbf{H}$ in (2), the number of operations of the preprocessing stage $N_{\text{SD-pre}}$ can be approximated by

$$N_{\text{SD-pre}} \approx 4N(2L-1)(MT)^2 + 8N(L-1)MTML - \frac{8N}{3}(2L-3)(ML)^2 \quad (16)$$

as shown in Appendix A.

Considering only the dominant term, we observe that there is a reduction in complexity by a factor of K/L compared to the QR decomposition of the full $NK \times MT$ matrix $\mathbf{H}$, whose preprocessing complexity is $O(8NK(MT)^2)$. In addition, since $K = T + L - 1$, that complexity reduction factor can be further approximated by T/L if $T \gg L$. Although the result is just an approximation, it gives an important intuitive idea about the preprocessing complexity. That is, the block-Toeplitz structure of $\mathbf{H}$ effectively reduces the preprocessing complexity to be equal to that of an $NL \times MT$ matrix, as opposed to the $NK \times MT$ matrix $\mathbf{H}$. In addition, that reduction in complexity could be expected also from an implementation point of view. Apart from the number of hardware architectures proposed for general QR decomposition [21], parallel algorithms and architectures also exist for block-Toeplitz matrices [22].

B. Preprocessing Stage of the BSD

In this case, the following operations are required during the preprocessing stage:

- 1) the full QR decomposition of the $N \times M$ matrix $\mathbf{H}(L-1)$ to obtain $\tilde{\mathbf{Q}}_1$, $\tilde{\mathbf{Q}}_2$ and $\tilde{\mathbf{R}}$;
- 2) the computation of $\tilde{\mathbf{w}}'(T-1) = \tilde{\mathbf{Q}}_1^H \tilde{\mathbf{y}}'(T-1)$.

Apart from the sizes of the matrices involved, two differences can be observed compared to the SD. First, the matrix $\tilde{\mathbf{Q}}_2$ needs to be computed since it is required during the tree search of the BSD as shown by (12). Second, only $\tilde{\mathbf{w}}'(T-1)$ can be computed during the preprocessing stage as opposed to $\mathbf{w}'$. This is due to the fact that $\tilde{\mathbf{w}}'(k)$ for $k = 0, \dots, T-2$ depends on the transmitted vector $\mathbf{x}$, as shown by (9), and can only be calculated during the tree search.

Initially, we consider the QR decomposition, that involves the following operations.

- NO: This is equivalent to the SD case in the previous section, the number of operations is $N_{\text{BSD-NO}} = 6NM$.
- IO: As in the previous section, the number of operations $N_{\text{BSD-IO}}$ is

$$N_{\text{BSD-IO}} = \sum_{m=1}^{M-1} \sum_{n=m+1}^M (16N-2) = M(M-1)(8N-1).$$

- Orthonormalization operations (OO): In order to obtain $\tilde{\mathbf{Q}}_2$, the Gram-Schmidt algorithm can be used [23]. Thus, the number of operations $N_{\text{BSD-OO}}$ is

$$N_{\text{BSD-OO}} = (N-M)((M+N-1)(8N-1) + 6N).$$

Taking into account that the number of operations of the QR decomposition $N_{\text{BSD-QR}} = N_{\text{BSD-NO}} + N_{\text{BSD-IO}} + N_{\text{BSD-OO}}$ and that the computation of $\tilde{\mathbf{w}}'(T-1)$ involves a matrix-vector product requiring $N_{\text{BSD-w}} = 8NM - 2M$ operations, the total number of operations during the preprocessing stage of the BSD is

$$N_{\text{BSD-pre}} = N_{\text{BSD-QR}} + N_{\text{BSD-w}} = 8N^3 - 3N^2 + (8N-2)M + N. \quad (17)$$

Thus, comparing the result in (17) with (16), the reduction in preprocessing complexity stated in [5] is clear, given the smaller size of $\mathbf{H}(L-1)$ compared to $\mathbf{H}$. However, due to the differences in the tree search between the SD and the BSD, as shown in Section III, it is necessary also to analyze the complexity of the tree search stage to assess whether the BSD results in a reduced complexity algorithm.

C. Tree Search Stage of the SD

The complexity of the tree search of the SD has been shown to be variable depending on the noise level and the channel conditions [9], [24]. Thus, no closed-form expression about the exact complexity can be obtained, although different asymptotical analysis exist [14], [18]. In order to obtain closed-form expressions, our analysis looks at the complexity of following only one path down the tree, which corresponds to the DFE

path given that the SE enumeration is used. Although such analysis does not take into account the effect of the noise level and the channel conditions, it can be considered as a lower bound that represents a good approximation in the high SNR scenario, where following only the DFE path yields the ML solution with high probability [9].

The operations required during the tree search of the SD can be divided between the following:

- 1) the computation of the P different branch metrics (BM) $|z_i - r_{i,i}x_i|^2$ from each node at each level $i = MT, \dots, 1$, as shown by (5);
- 2) the computation of z_i for $i = MT, \dots, 1$, as shown by (6);
- 3) the update of R_i for $i = MT - 1, \dots, 1$, as shown by (7).

The number of operations required to calculate the P BM is $N_{\text{SD-BM}} = 7PMT$. Although calculating all the P BM could seem excessive from a simulation point of view given that only one branch is followed down the tree at any particular time (*depth-first* search), this assumption represents more accurately the actual complexity of a hardware implementation of the SD. It has been previously shown for the SD that calculating all the BM in parallel provides a more optimized architecture with a higher detection speed [13], [25]. Thus, the above assumption can be considered a valid one for our analysis, especially since we are interested in the relative complexity of the two methods, rather than their absolute complexities. For simplification purposes, it has been assumed that all the branches are also calculated in the last level, i.e., $i = 1$.

The computation of z_i for $i = MT, \dots, 1$, taking into account the 0-elements of $\mathbf{R}$ [26], requires the following number of operations:

$$N_{\text{SD-z}} = 4M^2L(2T - L) - 4M^2(T - L) - 4MT \quad (18)$$

as shown in Appendix B.

Thus, taking into account that the number of operations required to update R_i is $N_{\text{SD-R}} = MT - 1$, the lower bound of the number of operations of the tree search of the SD is given by

$$\begin{aligned} N_{\text{SD-tree}} &= N_{\text{SD-BM}} + N_{\text{SD-z}} + N_{\text{SD-R}} \\ &= 4M^2L(2T - L) - 4M^2(T - L) \\ &\quad + (7P - 3)MT - 1. \end{aligned} \quad (19)$$

Again, considering only the dominant term in (19) and assuming that, in the general case, $T > L$, the block-Toeplitz structure of $\mathbf{H}$ reduces the complexity of the tree search compared to the full $NK \times MT$ matrix $\mathbf{H}$ case, which results in a tree search complexity lower bound $O(4(MT)^2)$. Furthermore, if we approximate $4M^2L(2T - L) \approx 8M^2LT$, we can see that the reduction in complexity can be approximated by a factor $T/2L$. Clearly, the larger T becomes compared to L , the larger the complexity reduction becomes given that a higher percentage of 0-matrix blocks can be found in $\mathbf{H}$.

D. Tree Search Stage of the BSD

The analysis in this section also concentrates on a lower bound of the complexity of the tree search, i.e., the complexity of following only one path down the tree. However, as shown in Section III-A, some additional operations are required during

the tree search of the BSD compared to that of the SD. The operations required during the tree search of the BSD can be divided between the following:

- 1) the computation of the P different BM $|z'_i - r'_{i,i}x'_i|^2$ from each node at each level $i = MT, \dots, 1$, as shown by (13);
- 2) the computation of z'_i for $i = MT, \dots, 1$, as shown by (14);
- 3) the update of R'_i for $i = MT - 1, \dots, 1$, as shown by (15);
- 4) the computation of d_1 at level $i = 1$, as shown by (8).

The number of operations required to calculate the P BM is $N_{\text{BSD-BM}} = 7PMT$ as in the SD case. The number of operations required to obtain z_i for $i = MT, \dots, 1$, considering the BSD tree search, is

$$\begin{aligned} N_{\text{BSD-z}} &= 4NML(2T - L) + 2MT(2M - 3) \\ &\quad + 4NM(L - 2) + 2M \end{aligned} \quad (20)$$

as shown in Appendix C.

In order to calculate the number of operations required to update R_i , we can see from (15) that we need to consider the $MT - 1$ subtractions of $|z'_j - r'_{j,j}x'_j|^2$ and the $T - 1$ subtractions of $\|\tilde{\mathbf{Q}}_2^H \tilde{\mathbf{y}}'(k)\|^2$. For the latter, a matrix-vector multiplication and a norm calculation are required in each subtraction ($\tilde{\mathbf{y}}'(k)$ does not need to be calculated given that $\mathbf{y}'$ is indirectly available from the calculation of z_i). Thus, the number of operations required to update R_i is given by

$$N_{\text{BSD-R}} = (T - 1)(8N + 2)(N - M) + MT - 1.$$

The computation of d_1 requires

$$N_{\text{BSD-d}} = 4NML(L - 1) + 4N(L - 1) \quad (21)$$

operations, where, on the right-hand side of (21), the first term corresponds to the computation of $\mathbf{y}_1 - \mathbf{H}_1\mathbf{x}$ (considering the 0-matrix blocks within $\mathbf{H}_1$) and the second term to the norm computation.

Finally, the lower bound of the complexity of the tree search of the BSD can be written as

$$\begin{aligned} N_{\text{BSD-tree}} &= N_{\text{BSD-BM}} + N_{\text{BSD-z}} + N_{\text{BSD-R}} + N_{\text{BSD-d}} \\ &\approx 8NLM T + 8N^2T + 4M^2T + (7P - 8N)MT \end{aligned} \quad (22)$$

where the lower order terms have been discarded.

E. Overall Complexity

Expressions for a lower bound of the overall complexity of the two algorithms can be obtained combining the appropriate results from previous sections, i.e., (16) and (19) for the SD and (17) and (22) for the BSD. This section concentrates on the analysis of those lower bounds, looking at their dominant term, representing their asymptotic behavior.

Table I shows the dominant terms in the complexity of the different stages of the algorithms under study, assuming $T > L$ and $N \geq M$. It can be seen how the preprocessing complexity can be considerably reduced applying the BSD as pointed out in [5]. This reduction comes from the fact that the preprocessing is performed only on $\mathbf{H}(L - 1)$ or $\mathbf{H}(0)$, depending on whether

TABLE I
DOMINANT COMPLEXITY TERMS OF THE PREPROCESSING AND
TREE STAGES OF THE SD AND THE BSD

Algorithms	N_{pre}	N_{tree}
SD	$8NL(MT)^2$	$4(2T - L)M^2L$
BSD	$8N^3$	$8NLM T$

the QR or the QL decomposition is used. Clearly, this effect is especially relevant for large MIMO systems and block sizes, since $N_{\text{BSD-pre}}$ is independent of the number of taps L and the block size T , as shown in (17). Considering the complexity of the tree stage, it can be seen how the lower bound in the BSD *increases* compared to the lower bound in the SD. Thus, although the BSD reduces the complexity of the preprocessing stage as indicated in [5], it also has the side effect of increasing the complexity lower bound of the tree search stage compared to the SD. Furthermore, since this analysis only considers a lower bound, that increase in complexity can be expected to be more relevant when looking at the average or worst-case complexity of MIMO systems, as shown in the next section.

Although the preprocessing stage seems to determine the overall complexity of the SD, this is only valid in the high SNR regime. In a low to medium SNR scenario, more than one path needs to be searched by the SD with high probability, making the complexity of the tree search stage have a more determinant effect on the overall complexity. The same can be said about the BSD, although, in this case, the tree search stage has a dominant effect on the overall complexity also in the high SNR scenario as shown by Table I.

Finally, it should be emphasized that the conclusions drawn in this section are valid only asymptotically, for large MIMO systems and block sizes. For a practical MIMO system, it is necessary to evaluate the complete complexity expressions in order to compare the relative complexity of the two algorithms under study. This last aspect is studied in the next section.

V. SIMULATION RESULTS

This section evaluates the bit error rate (BER) performance and complexity of the SD and the BSD through Monte Carlo simulations and the expressions in Section IV. In all cases, the SE enumeration and the complex-valued version of the algorithms have been used. Given that both the SD and the BSD provide exactly the same BER performance, only one plot is devoted to it. Fig. 1 shows the performance of both algorithms in a 2×2 system with 4-QAM and $T = 8$ as a function of the SNR per bit, $E_b/N_0 = \log_2^{-1}(P)/\sigma^2$. Channels with $L = 2$ and $L = 3$ taps per path following a uniform PDP, i.e., $\sigma_l^2 = 1/L$ for $l = 0, \dots, L - 1$, have been considered. It can be observed the diversity increase in the performance of the SD and the BSD as the number of taps in the channel increases. For comparison purposes, the performance of an uncoded OFDM system based on the IEEE 802.11n standard and using the SD per subcarrier has also been included. It can be seen how the performance of the OFDM system does not benefit from all the degrees of freedom of the frequency-selective channel, which matches the conclusions in [2].

Fig. 1. BER performance of the SD, the BSD and the SD-OFDM as a function of the SNR per bit in a channel with uniform PDP.

Fig. 2(a) shows the number of operations of the two stages of the algorithms as a function of the block size T . A 4×6 system with 16-QAM has been considered where the channel has $L = 3$ taps per path following an arbitrary PDP. The number of operations of the preprocessing stage, N_{pre} , and the lower bound of the number of operations of the tree search stage, N_{tree} , are shown. The curves show the exact number of operations without discarding lower order terms. Dashed curves have been used to show the approximations performed in Section IV, i.e., $N_{\text{SD-pre}}$ in (16) and $N_{\text{BSD-tree}}$ in (22) but they appear exactly superimposed to the solid curves, showing the validity of those approximations. First of all, it can be seen how, as claimed in [5], the complexity of the preprocessing stage is greatly reduced in the BSD, in addition to being independent of the block size T . However, the BSD has a higher tree search complexity than the SD, aspect that was not considered in [5] when proposing the BSD to reduce the complexity of the SD. The same trend can be observed in Fig. 2(b), which shows the number of operations of the two stages as a function of the number of antennas $M = N$. A system with 16-QAM, blocks of size $T = 10$ and a channel with $L = 3$ taps per path have been considered. Noting that the results in Fig. 2 can be used to obtain the overall complexity of the algorithms in the high SNR regime (where the actual complexity of the tree search stage corresponds to the lower bound with high probability), two main conclusions can be drawn from them.

- 1) The BSD greatly reduces the complexity of the preprocessing stage compared to the SD as indicated in [5]. Thus, in the high SNR regime, the overall complexity of the BSD is dominated by N_{tree} , whereas N_{pre} dominates the complexity of the SD. Clearly, by observing the results in Fig. 2, the BSD presents the lowest complexity in the high SNR regime, supporting the claim in [5].
- 2) However, we have seen how the preprocessing stage of the BSD has the side effect of modifying its tree search stage, increasing its complexity compared to that of the SD. Although that increase can be considered negligible at high

Fig. 2. Complexity of the preprocessing stage and minimum complexity of the tree search stage of the SD and the BSD (a) As a function of the block size T and (b) As a function of the number of antennas $M = N$.

SNR, it needs to be studied closely in the low to medium SNR regime, where, with high probability, it is required to follow more than one path down the tree to find the ML solution [9].

Thus, Fig. 3(a) shows the average complexity of the tree search stage of the SD and the BSD as a function of the SNR per bit in a 2×2 system with 4-QAM. Each path in the channel is considered to have $L = 3$ taps following a uniform PDP. Block sizes of $T = 4$ and $T = 8$ time instants have been considered. It can be observed how the complexity of the tree search stages increases with decreasing SNR and with increasing block size. In particular, it can be observed how the BSD presents a considerably higher complexity, especially in the low to medium SNR regime, compared to the SD. Although the asymptotic minimum complexity of the tree search stage has not been plotted for clarity purposes, Table II shows N_{tree} for the different algorithms. It can be seen how, at medium to high SNR, the average complexity of the SD is very close

Fig. 3. Average (a) tree search complexity and (b) overall complexity of the SD and the BSD as a function of the SNR per bit in a channel with uniform PDP.

TABLE II
LOWER BOUND OF THE COMPLEXITY OF THE TREE SEARCH STAGE OF THE SD AND THE BSD FOR THE SCENARIO OF FIG. 3

N_{tree}	$T = 4$	$T = 8$
SD	423	943
BSD	619	1,251

to its lower bound while the average complexity of the BSD, even at high SNR, is still far from it. This is due to the fact that even if the first path followed down the tree in the BSD corresponds to the ML solution, additional operations need to be performed to calculate d_1 for all the other branches on that level to make sure that no other branch in the last level has a lower Euclidean metric. On the other side, the SD only requires a comparison of the Euclidean metric with the other stored metrics to assess whether that first path corresponds to the ML solution. This factor, together with the additional operations required to calculate $\mathbf{z}$ in the BSD, explain the considerable

Fig. 4. Average tree search complexity of the SD with different channel matrix orderings as a function of the SNR per bit in a channel with uniform PDP.

higher average complexity of its tree search stage compared to the SD. The effect of the preprocessing complexity is taken into account in Fig. 3(b), showing the average overall complexity of the algorithms in the same scenario. It can be seen how, looking at the average overall complexity, the BSD can be considered as an alternative to reduce the complexity of the SD in the high SNR regime, although the advantage reduces as the block size T increases.

Since sorting the columns of the channel matrix has been successfully applied to reduce the complexity of the search stage of the SD in frequency-flat channels [9], Fig. 4 looks at the effect that sorting has on the SD in frequency-selective channels. A 2×2 system with 4-/16-QAM, $T = 8$ and $L = 3$ with uniform PDP has been considered. The V-BLAST-ZF ordering proposed in [9] for the SD and a sorting based on the sorted QR decomposition (SQRD) [27] have been applied. It can be seen how, in the frequency-selective case, sorting the columns of the channel matrix does not reduce the complexity of the tree search stage. In fact, the complexity is increased due to the fact that the sorted channel matrix does not necessarily have a block-Toeplitz structure, resulting in an increase in the number of operations as shown in Section IV-C. In order to illustrate that fact, the dashed curve labelled $SD^{(1)}$ shows the complexity of the SD when the full matrix $\mathbf{H}$ and no sorting are assumed. That curve can be used as a reference to see how the complexity does not change significantly by sorting the columns of the channel matrix. This result that, to the best of our knowledge, has not been previously reported in the literature, is of special importance since it indicates that the conclusions drawn for the SD in a frequency-flat environment do not necessarily apply to the frequency-selective case.

Fig. 5 looks at the complexity of the algorithms when a more realistic exponential PDP is considered. In particular, since not all the elements of $\mathbf{H}$ have the same variance, Fig. 5 looks at the differences in complexity if the QR or the QL decomposition are used. In particular, Fig. 5(a) shows the average complexity of the

Fig. 5. Average (a) tree search complexity and (b) overall complexity of the SD and the BSD as a function of the SNR per bit in a channel with exponential PDP.

tree search stage of the SD and the BSD as a function of the SNR per bit in a 2×4 system with 16-QAM. Each path in the channel is considered to have $L = 3$ taps following an exponential PDP, i.e., $\sigma_l^2 = e^{-l}/\sigma_h^2$ for $l = 0, \dots, L-1$ with $\sigma_h^2 = \sum_{l=0}^{L-1} e^{-l}$. Blocks of size $T = 5$ time instants have been considered. First of all, the complexity of the SD is practically independent of whether the QR or the QL decomposition are used in this particular scenario. However, it can be observed how the complexity of the BSD increases dramatically if the more common QR decomposition is used compared to the QL decomposition. This effect is not present if a uniform PDP is considered. In the case of an exponential PDP, due to the different variances of the elements of $\mathbf{H}$, the order in which the matrix decomposition is performed has an effect in the complexity of the algorithm. For the SD algorithms, the effect is negligible given that the full matrix is used for both the QL and the QR decomposition. However, for the BSD, the QR is applied to $\mathbf{H}(L-1)$ while the QL is applied

to $\mathbf{H}(0)$. Clearly, since the first taps of $\mathbf{h}_{i,j}$ contain more energy, it is beneficial, from a tree search perspective, to perform a QL decomposition on $\mathbf{H}(0)$. This implies that the tree search of the BSD is performed on the upper part of $\mathbf{H}$, from $i = 1$ to $i = MT$, which corresponds to the higher energy taps, thus reducing the number of paths that need to be searched to obtain the ML solution compared to the BSD with QR decomposition. The effect of the preprocessing complexity is taken into account in Fig. 5(b), showing the average overall complexity of the algorithms in the same scenario. It can be seen how the BSD with QL decomposition can be considered as an alternative to the SD to reduce the average overall complexity in the medium to high SNR regime.

Although the average overall complexity has been reduced if the BSD with QL decomposition is used in a channel with exponential PDP, the variance of that complexity also needs to be taken into account, especially from an implementation point of view. This is of especial importance given that in an actual communication system data needs to be processed in a fixed number of operations. In the case of the algorithms under study, the variance of the complexity is determined by the variance of the tree search complexity, due to the constant preprocessing complexity for a given system. Fig. 6 shows the CDF of the complexity exponent of the tree search stage for the SD and the BSD, using both QR and QL decomposition, in a 4×4 system with 4-QAM at $E_b/N_0 = 15$ dB. A block size $T = 6$ and the same channel as in Fig. 5 have been considered. It can be observed, looking at the slope of the CDF curves, that the variance of the tree search complexity of the BSD, even when the QL is used, is considerable larger than that of the SD. As stated above, this larger variance can represent a problem if the BSD is to be implemented in real time [28]. Another aspect that could not be observed in Fig. 5 is the fact that the average and the variance of tree search complexities of the SD is slightly larger if the QL decomposition is used instead of the QR decomposition in a channel with exponential PDP. It should be noted that no difference appears in any of the algorithms between the QR and the QL decomposition if a channel with uniform PDP is considered, since all the elements of $\mathbf{H}$ have the exact same probability density function (pdf).

Hence, it has been shown how the main advantage of the BSD lies on its considerably lower preprocessing complexity compared to the SD. However, that lower preprocessing complexity has the side effect of increasing the complexity of the tree search. This represents an important disadvantage from an implementation point of view, given that the tree search has a variable complexity depending on the SNR. Thus, although the BSD can provide a lower overall complexity compared to the SD at very high SNR, that reduction turns into an increase at low to medium SNR or if the dimensionality of the problem (M , N , T , L , or P) increases. In any case, both algorithms have a variable overall complexity whereas actual communication systems require data to be processed at a constant rate. In order to overcome that problem, the approach presented in [26] to reduce the average and the variance of the tree search complexity of the SD could be combined with early-termination techniques [28] to achieve the required fixed complexity, accepting the consequent performance degradation.

Fig. 6. CDF of the complexity of the tree search stage of the SD and the BSD in a channel with exponential PDP.

It should also be noted that the results of the tree search complexity have been obtained for a relatively small block size ($T < 10$) in order to present accurate results for the BSD at low SNRs.² Although the preprocessing complexity of the SD increases with the block size, the average and variance of the tree search complexity of the BSD suffer a comparatively larger increase with the block size compared to those of the SD, making the BSD more difficult to implement in practice. This fact, together with the results presented in this section, indicate that the SD still represents the most promising alternative to achieve ML performance in MIMO systems with frequency-selective channels.

VI. CONCLUSION

This paper has analyzed the complexity of two variants of the SD algorithm applied to the detection of MIMO systems in a frequency-selective environment. The joint detection performed for every transmitted block in a frequency-selective channel causes the preprocessing stage to have an important effect on the overall complexity of the SD given the $NK \times MT$ dimension of the equivalent channel matrix $\mathbf{H}$. Although the BSD was briefly proposed in [5] to greatly reduce the preprocessing complexity, no study was made of the effect that reduction could have on the tree search complexity. Globally, this paper has shown that the preprocessing stage of the BSD has a very negative effect on the complexity of its tree search stage, reducing its relevance as an alternative to the original SD. Through Monte Carlo simulations, this paper has shown how the BSD results in a higher average tree search complexity compared to the SD, independent of the SNR. If we consider the average overall complexity of the algorithms in a channel with uniform PDP, the BSD yields the lowest average complexity only in the high SNR regime. In addition, it has been shown how the sorting of the columns of the channel matrix does not

²As an example of more realistic block sizes, the Global System for Mobile Communications (GSM) standard specifies a block size $T = 58$ [29].

help reduce the complexity of the tree search stage of the SD in frequency-selective channels.

In a more realistic frequency-selective environment where the channel taps follow an exponential PDP, the BSD has a considerably larger average overall complexity, except in the very high SNR regime. That complexity can be greatly reduced if the QL decomposition is applied to the preprocessing stage. Even in this case, the larger variance of the complexity of the tree search stage of the BSD compared to the SD makes the BSD less suitable for hardware implementation where data needs to be processed in a fixed number of operations.

APPENDIX A DERIVATION OF $N_{\text{SD-pre}}$

In order to calculate the number of operations of the preprocessing stage $N_{\text{SD-pre}}$, we first consider the QR decomposition of $\mathbf{H}$ and divide the analysis in the number of NO and IO. For that purpose, we take into account that the number of elements of the i th column of $\mathbf{Q}_1$ always different from 0 is $N(L + \lceil i/M \rceil - 1)$ for $i = 1, \dots, MT$ and that the number of elements of each column of $\mathbf{H}$ always different from 0 is NL . Thus, the number of NO can be written as

$$\begin{aligned} N_{\text{SD-NO}} &= M \sum_{m=1}^T 6N(L + m - 1) \\ &= 3N(2L + T - 1)MT. \end{aligned} \quad (23)$$

Considering now the IO, we divide the number of operations $N_{\text{SD-IO}}$ in two terms, the number of operations related to line 6 of Algorithm 1, $N_{\text{SD-IO-6}}$, and the number of operations related to line 7 of Algorithm 1, $N_{\text{SD-IO-7}}$. Although the number of elements of the i th column of $\mathbf{Q}_1$ always different from 0 is $N(L + \lceil i/M \rceil - 1)$, that value is just NL at the beginning of the algorithm. Thus, assuming $T > L$, $N_{\text{SD-IO-6}}$ can be divided in the following terms:

- 1) the number of operations between pairs of columns (i, j) of $\mathbf{Q}_1$, where $\lceil i/M \rceil = \lceil j/M \rceil$, i.e.,

$$N_{6-1} = \sum_{m=1}^T \sum_{n=1}^M (M - n)(8N(L + m - 1) - 2); \quad (24)$$

- 2) the number of operations between pairs of columns (i, j) of $\mathbf{Q}_1$, where $\lceil i/M \rceil \neq \lceil j/M \rceil$ and $\text{mod}(i, M) \neq 1$, with $\text{mod}(\cdot, \cdot)$ representing the modulo operation. This number of operations N_{6-2} is given by

$$\begin{aligned} N_{6-2} &= \sum_{m=1}^{T-L+1} M(M-1)(L-1)(8N(L+m-1)-2) \\ &+ \sum_{m=T-L+2}^T M(M-1)(T-m)(8N(L+m-1)-2); \end{aligned} \quad (25)$$

- 3) the number of operations between pairs of columns (i, j) of $\mathbf{Q}_1$, where $\lceil i/M \rceil \neq \lceil j/M \rceil$ and $i = 1$, i.e.,

$$N_{6-3} = \sum_{m=1}^{L-1} M(8N(L-m)-2); \quad (26)$$

- 4) finally, the number of operations between pairs of columns (i, j) of $\mathbf{Q}_1$, where $\lceil i/M \rceil \neq \lceil j/M \rceil$, $\text{mod}(i, M) = 1$ and $i \neq 1$. This number of operations N_{6-4} is given by

$$\begin{aligned} N_{6-4} &= \sum_{m=2}^{T-L+1} (M(L-2)(8N(L+m-1)-2) + M(8N-2)) \\ &+ \sum_{m=T-L+2}^T M(T-m)(8N(L+m-1)-2). \end{aligned} \quad (27)$$

Combining (24)–(27), the number of operations $N_{\text{SD-IO-6}}$ can be directly written, due to space constraints, as

$$\begin{aligned} N_{\text{SD-IO-6}} &= N_{6-1} + N_{6-2} + N_{6-3} + N_{6-4} \\ &\approx 2N(2L-1)(MT)^2 \\ &+ 4N(L-1)MTML \\ &- \frac{4N}{3}(2L-3)(ML)^2 \end{aligned} \quad (28)$$

where the lower order terms have been discarded.

Equivalently, also assuming $T > L$, $N_{\text{SD-IO-7}}$ can be divided in the following terms:

- 1) the number of operations between pairs of columns (i, j) of $\mathbf{Q}_1$, where $\lceil i/M \rceil = \lceil j/M \rceil$, which can be written as

$$N_{7-1} = \sum_{m=1}^T \sum_{n=1}^M 8N(M-n)(L+m-1); \quad (29)$$

- 2) the number of operations between pairs of columns (i, j) of $\mathbf{Q}_1$, where $\lceil i/M \rceil \neq \lceil j/M \rceil$, which can be written as

$$\begin{aligned} N_{7-2} &= \sum_{m=1}^{T-L+1} 8NM^2(L-1)(L+m-1) \\ &+ \sum_{m=T-L+2}^T 8NM^2(T-m)(L+m-1). \end{aligned} \quad (30)$$

Combining (29) and (30), the number of operations $N_{\text{SD-IO-7}}$ can be directly written as

$$\begin{aligned} N_{\text{SD-IO-7}} &= N_{7-1} + N_{7-2} \\ &\approx 2N(2L-1)(MT)^2 \\ &+ 4N(L-1)MTML \\ &- \frac{4N}{3}(2L-3)(ML)^2 \end{aligned} \quad (31)$$

where the lower order terms have been discarded.

Combining (28) and (31), we obtain

$$\begin{aligned} N_{\text{SD-IO}} &= N_{\text{SD-IO-6}} + N_{\text{SD-IO-7}} \\ &\approx 4N(2L-1)(MT)^2 \\ &\quad + 8N(L-1)MTML \\ &\quad - \frac{8N}{3}(2L-3)(ML)^2. \end{aligned} \quad (32)$$

Comparing (32) to (23), it can be seen that $N_{\text{SD-NO}}$ can be discarded compared to $N_{\text{SD-IO}}$. Thus, the number of operations of the QR decomposition can be approximated as

$$N_{\text{SD-QR}} \approx N_{\text{SD-IO}}.$$

Considering now the computation of $\mathbf{w}$, and given the 0-matrix blocks within $\mathbf{Q}_1$, the number of operations required is

$$\begin{aligned} N_{\text{SD-w}} &= M \sum_{m=1}^T (8N(L+m-1) - 2) \\ &= 4N(2L+T-1)MT - 2MT. \end{aligned} \quad (33)$$

Again, comparing (33) to (32), it can be seen that $N_{\text{SD-w}}$ can be discarded compared to $N_{\text{SD-QR}}$ which results in the approximation of $N_{\text{SD-pre}}$ indicated by (16).

APPENDIX B DERIVATION OF $N_{\text{SD-z}}$

Considering the block-Toeplitz structure of $\mathbf{H}$, the total number of operations required to obtain all z_i for $i = MT, \dots, 1$, if only one path is followed by the SD, can be written, assuming $T > L$, as

$$\begin{aligned} N_{\text{SD-z}} &= T \sum_{m=1}^M 8(m-1) \\ &\quad + \underbrace{\sum_{n=1}^T \min(8M^2(L-1), 8M^2(n-1))}_{\alpha}. \end{aligned} \quad (34)$$

The term α in (34) can be obtained writing

$$\begin{aligned} \alpha &= \sum_{n=1}^{T-L+1} 8M^2(L-1) + \sum_{n=T-L+2}^T 8M^2(T-n) \\ &= 4(L-1)M^2(2T-L). \end{aligned} \quad (35)$$

Then, replacing α in (34) by (35) and evaluating (34) yields the result in (18).

APPENDIX C DERIVATION OF $N_{\text{BSD-z}}$

The total number of operations required to obtain all z_i for $i = MT, \dots, 1$, if only one path is followed by the BSD, can be obtained considering the following three terms:

- 1) the number of operations required to obtain $\mathbf{y}'$, as shown by (9),

$$\begin{aligned} N_1 &= \sum_{m=1}^T \min(8NM(L-1), 8NM(m-1)) \\ &= 4(L-1)NM(2T-L) \end{aligned} \quad (36)$$

which has been obtained using the approach employed in (35) to calculate α , also assuming $T > L$;

- 2) the number of operations required to obtain $\tilde{\mathbf{w}}'(k)$ for $k = 0, \dots, T-2$ in (10), which corresponds to

$$N_2 = (T-1)N_{\text{BSD-w}} = (T-1)(8NM - 2M) \quad (37)$$

i.e., $T-1$ times the number of operations required to calculate $\tilde{\mathbf{w}}'(T-1)$;

- 3) the number of operations required to obtain z_i once w_i is available, as shown by (14),

$$N_3 = T \sum_{m=1}^M 8(m-1) = 4MT(M-1) \quad (38)$$

which has been obtained taking into account the block-diagonal structure of $\mathbf{R}'$.

Combining (36)–(38), $N_{\text{BSD-z}} = N_1 + N_2 + N_3$ yields the result in (20).

REFERENCES

- [1] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Pers. Commun. Mag.*, vol. 6, no. 3, pp. 311–335, Mar. 1998.
- [2] A. Medles and D. T. M. Slock, "Optimal diversity vs multiplexing tradeoff for frequency selective mimo channels," in *Proc. IEEE Int. Symp. Information Theory (ISIT'05)*, Adelaide, Australia, Sep. 2005, vol. 1, pp. 1813–1817.
- [3] R. van Nee and R. Prasad, *OFDM For Wireless Multimedia Communications*. London, U.K.: Artech House, 2000.
- [4] G. D. Forney, Jr., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inf. Theory*, vol. 18, no. 3, pp. 363–378, May 1972.
- [5] H. Vikalo and B. Hassibi, "Maximum-likelihood sequence detection of multiple antenna systems over dispersive channels via sphere decoding," *EURASIP J. Appl. Signal Process.*, vol. 2002, no. 5, pp. 525–531, May 2002.
- [6] U. Finckel and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comput.*, vol. 44, no. 170, pp. 463–471, Apr. 1985.
- [7] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1639–1642, Jul. 1999.
- [8] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.
- [9] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.
- [10] S. Liu and Z. Tian, "Near-optimum soft decision equalization for frequency selective MIMO channels," *IEEE Trans. Signal Process.*, vol. 52, no. 3, pp. 721–733, Mar. 2004.
- [11] S. Bavarian and J. K. Cavers, "Symbol by symbol soft-input soft-output multiuser detection for frequency selective MIMO channels," in *Proc. IEEE Wireless Communications Networking Conf. (WCNC)*, New Orleans, LA, Mar. 2005, vol. 1, pp. 515–520.
- [12] X. Li, C. Yin, and G. Yue, "Soft decision equalization of multiple antenna systems over dispersive channels via max-log-MAP sphere decoder," in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Communications (PIMRC)*, Helsinki, Finland, Sep. 2006, vol. 1, pp. 1–5.
- [13] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, pp. 1566–1577, Jul. 2005.
- [14] J. Jaldén 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.
- [15] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, pp. 389–399, Mar. 2003.

- [16] A. D. Murugan, H. E. Gamal, M. O. Damen, and G. Caire, "A unified framework for tree search decoding: Rediscovering the sequential decoder," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 933–953, Mar. 2006.
- [17] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Program.*, vol. 66, pp. 181–199, 1994.
- [18] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005.
- [19] G. H. Golub and C. F. V. Loan, *Matrix Computations*. London, U.K.: The Johns Hopkins Univ. Press, 1996.
- [20] V. Kühn, *Wireless Communications Over MIMO Channels*. Chichester, U.K.: Wiley, 2006.
- [21] K. R. Liu and K. Yao, "Multiphase systolic algorithms for spectral decomposition," *IEEE Trans. Signal Process.*, vol. 40, no. 1, pp. 190–201, Jan. 1992.
- [22] P. Alonso, J. M. Badía, and A. M. Vidal, "Solving the block-Toeplitz least-squares problem in parallel," *Concurrency and Computation: Practice and Experience*, vol. 17, no. 1, pp. 49–67, Jan. 2005.
- [23] T. S. Shores, *Applied Linear Algebra and Matrix Analysis*. New York: Springer, 2007.
- [24] L. G. Barbero and J. S. Thompson, "Performance of the complex sphere decoder in spatially correlated MIMO channels," *IET Commun.*, vol. 1, no. 1, pp. 122–130, Feb. 2007.
- [25] L. G. Barbero and J. S. Thompson, "Rapid prototyping of the sphere decoder for MIMO systems," in *Proc. IEEE/EURASIP Conf. DSP Enabled Radio (DSPeR)*, Southampton, U.K., Sep. 2005, vol. 1, pp. 41–47.
- [26] L. G. Barbero, T. Ratnarajah, and C. Cowan, "A multistage sphere decoder for frequency-selective MIMO channels," presented at the IEEE Int. Conf. Personal, Indoor, Mobile Radio Communications (PIMRC), Athens, Greece, Sep. 2007.
- [27] D. Wübben, R. Böhnke, V. Kühn, and K.-D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in *Proc. IEEE Vehicular Technology Conf. (VTC'03-Fall)*, Orlando, FL, Oct. 2003, vol. 1, pp. 508–512.
- [28] A. Burg, M. Borgmann, M. Wenk, C. Studer, and H. Bölcskei, "Advanced receiver algorithms for MIMO wireless communications," presented at the Design, Automation Test in Europe (DATE), Munich, Germany, Mar. 2006.
- [29] A. Mehrotra, *GSM System Engineering*. Norwood, MA: Artech House, 1997.

Luis G. Barbero (M'02–S'04–M'07) was born in Barcelona, Spain, in 1976. He received the M.Sc. degree in telecommunications from the Technical University of Catalonia (UPC), Barcelona, Spain, in 2000 and the Ph.D. degree in electrical engineering from the University of Edinburgh, Edinburgh, U.K., in 2006.

From 1999 to 2000, he was with Vocal Technologies, Ltd., Peralada, Spain, working on the DSP implementation of modem transceivers. In 2000, he joined Massana, Ltd., Dublin, Ireland, working

on the physical layer testing and integration of gigabit Ethernet NIC devices. From 2001 to 2003, he was a Senior Engineer at Systemic AG/Philips Semiconductors, Dresden, Germany, working on the design and implementation of the physical layer of a multimode WLAN system. Since 2007, he has been a Research Fellow at the Institute of Electronics, Communications and Information Technology at Queen's University of Belfast, Belfast, U.K. His research interests include signal processing for wireless communications, MIMO detection algorithms, turbo-MIMO systems and implementation aspects of next-generation wireless systems.

T. Ratnarajah (S'94–A'96–M'05–SM'05) received the B.Eng. (Hons), M.Sc., and Ph.D. degrees.

He is currently a Principal Research Engineer with the Institute for Electronics, Communications and Information Technologies (ECIT), Queen's University Belfast, Belfast, U.K. Since 1993, he has held various research positions with the University of Ottawa, Ottawa, ON, Canada; Nortel Networks, Ottawa, ON, Canada; McMaster University, Hamilton, ON, Canada; and Imperial College, London, U.K.

His research interests include random matrices theory, information theoretic aspects of MIMO channels and ad hoc networks, wireless communications, signal processing for communication, statistical and array signal processing, biomedical signal processing, and quantum information theory. He has published over 80 publications in these areas and holds four U.S. patents.

Dr. Ratnarajah is a member of American Mathematical Society and the Information Theory Society.

Colin F. N. Cowan (M'82–SM'91) received the B.Sc. degree in engineering and the Ph.D. degree, both from the University of Edinburgh, Edinburgh, U.K., in 1977 and 1980, respectively.

From 1980 to 1989, he was a Lecturer and from 1989 to 1991 a Reader in the Electrical Engineering Department at the University of Edinburgh. From 1991 to 1996, he was a Professor of Signal Processing with Loughborough University, Loughborough, U.K., and the head of the Electronic and Electrical Engineering Department from 1994 to

1996. During his time at Loughborough, he was the head of the Communications and Signal Processing Research Group, within which he still holds a visiting chair. He currently holds the Nortel Networks/Royal Academy of Engineering research professorship in Telecommunications with the Queen's University Belfast, Belfast, U.K. He is also the Research Director responsible for Digital Communications with the ECIT research Institute. He has published over 250 papers and articles in the area of adaptive signal processing.

Prof. Cowan has been actively involved in the organization of a number of key international conferences, including chairing the First IEEE International Conference on Artificial Neural Networks in 1989 and the Seventh European Signal Processing Conference (EUSIPCO) in 1994.