

Document downloaded from:

<http://hdl.handle.net/10251/122916>

This paper must be cited as:

Simarro, MA.; García Mollá, VM.; Vidal Maciá, AM.; Martínez Zaldívar, FJ.; Gonzalez, A. (2018). Soft MIMO detection through sphere decoding and box optimization. *Signal Processing*. 145:48-58. <https://doi.org/10.1016/j.sigpro.2017.11.010>



The final publication is available at

<https://doi.org/10.1016/j.sigpro.2017.11.010>

Copyright Elsevier

Additional Information

Soft MIMO Detection Through Sphere Decoding and Box Optimization

M. Ángeles Simarro^{a,*}, Víctor M. Garcia-Mollá^b, Antonio Vidal^b, F.J. Martínez-Zaldívar^a, Alberto Gonzalez^{a,1}

^a*Institute of Telecommunications and Multimedia Applications, Universitat Politècnica de València, Camino de Vera s/n 46022 Valencia, SPAIN.*

^b*Department of Information Systems and Computing, Universitat Politècnica de València, Camino de Vera s/n 46022 Valencia, SPAIN.*

Abstract

Achieving optimal detection performance with low complexity is one of the major challenges, mainly in multiple-input multiple-output (MIMO) detection. This paper presents three low-complexity Soft-Output MIMO detection algorithms that are based mainly on Box Optimization (BO) techniques. The proposed methods provide good performance with low computational cost using continuous constrained optimization techniques. The first proposed algorithm is a non-optimal Soft-Output detector of reduced complexity. This algorithm has been compared with the *Soft-Output Fixed Complexity* (SFSD) algorithm, obtaining lower complexity and similar performance. The two remaining algorithms are employed in a turbo receiver, achieving the max-log Maximum a Posteriori (MAP) performance. The two Soft-Input Soft-Output (SISO) algorithms were proposed in a previous work for soft-output MIMO detection. This work presents its extension for iterative decoding. The SISO algorithms presented are developed and compared with the SISO *Single Tree Search* algorithm (STS), in terms of efficiency and computational cost. The results show that the proposed algorithms are more efficient for high order constellation than the STS

*Corresponding author

Email addresses: mdesiha@iteam.upv.es (M. Ángeles Simarro), vmgarcia@dsic.upv.es (Víctor M. Garcia-Mollá), avidal@dsic.upv.es (Antonio Vidal), fjmartin@dcom.upv.es (F.J. Martínez-Zaldívar), agonzal@dcom.upv.es (Alberto Gonzalez)

¹EURASIP member.

algorithm.

Keywords: MIMO, Soft-Input, Soft-Output, detection, Box Optimization.

1. Introduction

Soft-Output (SO) detection implies significant challenges for practical receiver implementations, particularly in the context of multiple-input multiple-output (MIMO) wireless communication systems. SO detection achieves significantly better performance than Hard-Output (HO) detection and is needed to reach optimum performance of the modern coding systems [1]. On the other hand, an iterative receiver that employs Soft-Input Soft-Output (SISO) detection improves the performance of a Non-Iterative Soft-Output (NISO) detection [1], since the near-optimal performance of turbo receivers is achieved by exchanging soft extrinsic information between a SISO detector and a SISO decoder in an iterative loop.

Throughout the paper we are going to use the following notation: SISO will denote a detector in an iterative receiver (the detector takes the received signal and the soft information provided by the channel decoder as inputs); NISO will denote a non iterative detection (the detector takes the received signal as input but does not process any information provided by the channel decoder, so no iteration is performed between the detector and the decoder); and SO is used to denote both cases. The problems that arise from these technologies (NISO and SISO) for MIMO detection are computationally very complex, furthermore the complexity exponentially increases when antenna number increases. Thus, it is necessary to develop efficient algorithms in both cases if these techniques are to be used in practical applications. Furthermore several strategies such as the max-log approximation [2] ($\log \sum_i \exp(\phi_i) \approx \max \phi_i$) can be employed to reduce the complexity of the exact computation.

There are several SO algorithms for MIMO systems [3]-[8], and they offer different tradeoffs between performance and complexity. The *Repeated Tree Search* (RTS) and the *Single Tree Search* (STS) algorithms provide the max-

log SO solution. Both algorithms use an HO Sphere Decoder (SD) to compute the extrinsic information and have been thoroughly discussed for the NISO case in [8] and for the SISO case in [9], including the application of a clipping boundary to both algorithms. The STS algorithm reduces the computational cost of the solution with respect to the RTS algorithm. Moreover, there are several non-optimal max-log NISO detection algorithms such as *Soft-Output Fixed-complexity SD* (SFSD) [10], the *Smart Ordering and Candidate Adding* (SOCA) [11] or *Soft-Output K-Best* [12]. Among these methods, SFSD reaches almost max-log Maximum Likelihood (ML) performance with low complexity.

The work described in this paper focuses on low-cost SO MIMO detection algorithms. We propose three methods: one for NISO case called *Box Optimization Hard Detector with soft-output* (BOHD-SO), which exhibits a non-optimal max-log ML performance; and two SISO methods, *SISO Box Optimization Repeated Tree Search* (SISO-BORTS) and *SISO Double Tree Search* (SISO-DTS), with max-log Maximum a Posteriori (MAP) performance based on a previous work reported in [13]. These algorithms reduce the complexity based on the use of continuous constrained optimization methods where the bounds are fixed, hence the name of *Box Optimization* (BO). As the target of this work, this auxiliary technique is used to implement different low-complexity SO detection algorithms.

The work described in [13] has as its main goal the improvement in efficiency of NISO max-log detection algorithms, where the iteration at the receiver was not considered. In the present paper we meaningfully extend the work presented in [13] to the Soft-Input case. Thereby in the present work the *a priori* information is incorporated by extending BORTS and DTS presented in [13] to the SISO detection.

Furthermore, a new method called BOHD-SO has been proposed. This algorithm is based on the BO algorithm and the HO ML detector described in [14]. Therefore, the BOHD-SO algorithm has been proposed and evaluated for NISO detection. Since it is a suboptimal algorithm, it will be compared with the SFSD algorithm in terms of efficiency and accuracy. It has been proved that

the use of the BOHD-SO algorithm in a turbo receiver does not improve the performance detection, for this reason it is only presented for the NISO case.

Table 1 summarizes the main features of the proposed and referenced algorithms.

Algorithm	SO	SISO	No clipping	Clipping	Performance	Ref
<i>BOHD</i>	No	No	Yes	No	max-log ML	[14]
<i>SFSD</i>	Yes	No	Yes	No	almost max-log ML	[10]
<i>BOHD-SO</i>	Yes	No	Yes	No	almost max-log ML	this work
<i>RTS</i>	Yes	Yes	Yes	Yes	max-log MAP	[15]
<i>STS</i>	Yes	Yes	Yes	Yes	max-log MAP	[8][9]
<i>BORTS</i>	Yes	No	Yes	Yes	max-log ML	[13]
<i>DTS</i>	Yes	No	No	Yes	max-log ML	[13]
<i>SISO-BORTS</i>	Yes	Yes	Yes	Yes	max-log MAP	this work
<i>SISO-DTS</i>	Yes	Yes	No	Yes	max-log MAP	this work

Table 1: The main features of the algorithms.

2. Model Description

Let us consider a MIMO Bit-Interleaved Coded-Modulation (BICM)[16] system with n_T transmit antennas and n_R receive antennas ($n_R \geq n_T$), as shown Figure 1. In this system, the sequence of information bits is encoded using an error-correcting code and is passed through a bitwise interleaver before being demultiplexed into n_T streams. The bits are mapped into a complex symbol s_i , which is taken from a constellation Ω of size $|\Omega| = M$ and hence carrying $k = \log_2 M$ code bits each, $s_i \leftrightarrow (x_{i,1}, \dots, x_{i,k})$ with $x_{i,b} \in \{0, 1\}$. Thus, the mapper translates $n_T k$ bits to one transmit symbol vector \mathbf{s} . In the following, the index $i \in \{1, \dots, n_T\}$ refers to the antenna index and the index $b \in \{1, \dots, k\}$ refers to the bit index within symbol s_i . The transmit symbol vector is given by $\mathbf{s} = (s_1, \dots, s_{n_T})^T$ and the complex baseband model can be written as

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

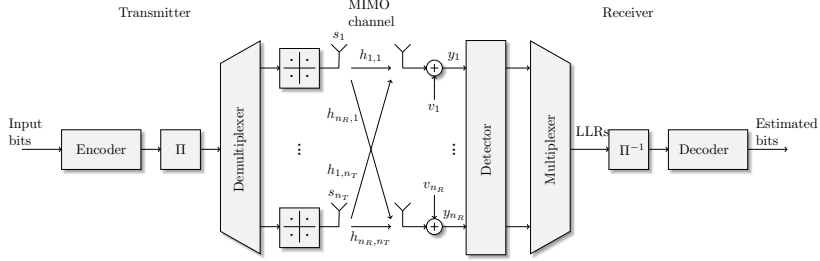


Figure 1: Block diagram of a MIMO-BICM system with n_T transmitting antennas and n_R receiving antennas.

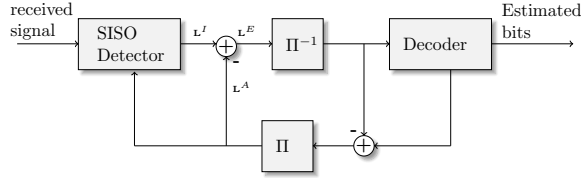


Figure 2: Block diagram of an ID-BICM receiver.

Here, $\mathbf{H} \in \mathbb{C}^{n_R \times n_T}$ is the MIMO channel matrix, which is composed of independent elements $h_{ij} \sim \mathcal{N}(0, 1)$, and \mathbf{v} denotes an additive white Gaussian noise (AWGN) complex vector with elements $v_j \sim \mathcal{N}(0, \sigma_n^2)$ where σ_n^2 represents the noise variance per complex dimension.

A BICM with Iterative Decoding (ID-BICM) can be used on the receiver side. Figure 2 shows the block diagram of an ID-BICM receiver. In this case, a SISO detector is needed. On the other hand, when iterations are not performed, the detector used is a NISO detector, see Figure 1.

First of all, let us to review the formulation of a SISO detection in an ID-BICM receiver since it is the general case. Afterwards, the NISO detection will be presented as a specific case of SISO detection. *Extrinsic* soft information in the form of log-likelihood ratios (LLRs) and denoted by \mathbf{L}^E is exchanged between the detector and the decoder in the ID-BICM scheme, as shown in Figure 2. Each $L_{i,b}^E$ value in the \mathbf{L}^E matrix represents the *extrinsic* soft information that is associated with the $x_{i,b}$ bit. These *extrinsic* LLRs can be easily

computed using the *intrinsic* information (\mathbf{L}^I) calculated by the detector for each of the encoded bits $x_{i,b}$ and the *a priori* information (\mathbf{L}^A) provided by the decoder:

$$\mathbf{L}^E = \mathbf{L}^I - \mathbf{L}^A. \quad (2)$$

Thus, \mathbf{L}^E , \mathbf{L}^I , and \mathbf{L}^A are matrices of $n_T \times k$ size where each (i, b) entry represents the corresponding soft information associated with the $x_{i,b}$ bit.

Based on (1) and using the Bayes' theorem and the max-log approximation [1], the *intrinsic* max-log LLR values can be calculated as:

$$L_{i,b}^I = \frac{1}{\sigma_n^2} \left[\min_{s \in \chi_{i,b}^0} \{\Lambda(\mathbf{s})\} - \min_{s \in \chi_{i,b}^1} \{\Lambda(\mathbf{s})\} \right], \quad (3)$$

where $\chi_{i,b}^u$ denotes the set of possible transmitted vectors for which b bit in s_i is equal to u . The $\Lambda(\mathbf{s})$ metric is derived from

$$\Lambda(\mathbf{s})' = \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 - \log P[\mathbf{s}], \quad (4)$$

where the term $-\log P[\mathbf{s}]$ uses the *a priori* soft reliability information matrix (\mathbf{L}^A) provided by the decoder (see Figure 2) and can be computed as:

$$\log P[\mathbf{s}] = K - \sum_{i=1}^{n_T} \sum_{b=1}^k \frac{1}{2} (x_{i,b} L_{i,b}^A), \quad (5)$$

where K is a constant term independent of the binary-valued variables $x_{i,b}$. Further details about the reliability information provided by the detector can be found in [1].

Let us use the QR decomposition of the channel matrix, $\mathbf{H} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$, where the matrix $\mathbf{Q} = [\mathbf{Q}_0 \quad \mathbf{Q}_1]$ is unitary and \mathbf{R} is upper-triangular. In (4), the metric can be computed as

$$\Lambda(\mathbf{s})' = \|\mathbf{z} - \mathbf{R}\mathbf{s}\|^2 + \|\mathbf{z}_0\|^2 - \log P[\mathbf{s}], \quad (6)$$

where $\tilde{\mathbf{z}} = \mathbf{Q}^H \mathbf{y} = \begin{bmatrix} \mathbf{z} \\ \mathbf{z}_0 \end{bmatrix}$. The term $\|\mathbf{z}_0\|^2$ cancels out in the LLR calculation. Therefore, the metric in (4) will be computed as

$$\Lambda(\mathbf{s}) = \|\mathbf{z} - \mathbf{R}\mathbf{s}\|^2 - \log P[\mathbf{s}], \quad (7)$$

Since \mathbf{Q} is a unitary matrix, \mathbf{y} and $\tilde{\mathbf{z}}$ have the same Euclidean norm. Therefore, due to the upper triangular shape of \mathbf{R} , (7) will be computed as:

$$\Lambda(\mathbf{s}) = \sum_{i=1}^{n_T} \left[\left| z_i - \sum_{j=i}^{n_T} R_{i,j} s_j \right|^2 - \log P[s_i] \right], \quad (8)$$

with $-\log P[\mathbf{s}] = \sum_{i=1}^{n_T} -\log P[s_i]$. Expression (8) can be iteratively calculated and depicted by a tree structure, computing a partial distance at each level i . The partial distances are evaluated as

$$\Lambda_i = \Lambda_{i+1} + e_i, \quad i = n_T, n_T - 1, \dots, 1 \quad (9)$$

with

$$e_i = \frac{1}{\sigma_n^2} \left| z_i - \sum_{j=i}^{n_T} R_{i,j} s_j \right|^2 - \log P[s_i]. \quad (10)$$

Note that the $-\log P[s_i]$ term in (10) can provide negative increments in (9). This situation leads to an unfavorable tree search. In order to avoid this effect, the $-\log P[s_i]$ is redefined as [9]

$$-\log P[s_i] \approx \sum_{b=1}^k \frac{1}{2} (|L_{i,b}^A| - x_{i,b} L_{i,b}^A). \quad (11)$$

Therefore the term $-\log P[\mathbf{s}]$ in (5) is set to:

$$-\log P[\mathbf{s}] \approx \sum_{i=1}^{n_T} \sum_{b=1}^k \frac{1}{2} (|L_{i,b}^A| - x_{i,b} L_{i,b}^A). \quad (12)$$

It is important to note that using the modified $-\log P[s_i]$ in (10) does not lead to an approximation of (3) [9].

One of the two minima in (3) is the metric (d^{MAP}) given by the maximum a posteriori (MAP) transmitted symbol vector, \mathbf{s}^{MAP} :

$$\mathbf{s}^{MAP} = \arg \min_{\mathbf{s} \in \Omega^{n_T}} \{ \|\mathbf{z} - \mathbf{R}\mathbf{s}\|^2 - \log P[\mathbf{s}] \} \quad (13)$$

$$d^{MAP} = \|\mathbf{z} - \mathbf{R}\mathbf{s}^{MAP}\|^2 - \log P[\mathbf{s}^{MAP}]. \quad (14)$$

The other minimum in (3) has to be calculated for every coded bit. This minimum is called a *counter-hypothesis* metric ($\bar{d}_{i,b}$), which is given by the minimum metric associated to

$$\mathbf{s} \in \mathcal{X}_{i,b}^{(\overline{x_{i,b}^{MAP}})} \quad (15)$$

where $\overline{x_{i,b}^{MAP}}$ denotes the complement of bit $x_{i,b}$ in \mathbf{s}^{MAP} . This counter-hypothesis metric can be computed as

$$\bar{d}_{i,b} = \min_{\mathbf{s} \in \mathcal{X}_{i,b}^{(\overline{s_{i,b}^{MAP}})}} \left(\frac{\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2}{\sigma_n^2} - \log P[\mathbf{s}] \right). \quad (16)$$

Finally the max-log MAP *intrinsic* LLRs are calculated through a tree search as

$$L_{i,b}^I = \frac{1}{\sigma_n^2} (d^{MAP} - \bar{d}_{i,b}) (1 - 2x_{i,b}^{MAP}), \quad (17)$$

where the term at the end adjust the sign depending on whether d^{MAP} corresponds to the first or the second minimum in (3).

Instead of computing the *intrinsic* information first as in (17) and then evaluating expression (2), the *extrinsic* information can be obtained through a tree search by a small change. The *counter-hypothesis* distance can be replaced by *extrinsic* metrics, which are computed as:

$$\bar{\delta}_{i,b} = \bar{d}_{i,b} + (1 - 2x_{i,b}^{MAP}) L_{i,b}^A. \quad (18)$$

Then the max-log MAP extrinsic LLRs in a SISO detector can be computed using (17), just by replacing $\bar{d}_{i,b}$ with $\bar{\delta}_{i,b}$:

$$L_{i,b}^E = \frac{1}{\sigma_n^2} (d^{MAP} - \bar{\delta}_{i,b}) (1 - 2x_{i,b}^{MAP}). \quad (19)$$

For NISO detection the *a priori* information \mathbf{L}^A is zero. In this case, there is no difference between *extrinsic* and *intrinsic* information, so the *intrinsic* information computed by (17) directly is fed back to the decoder. To compute (17) for NISO detection, the previous formulation can be used omitting the term $-\log P[\mathbf{s}]$ in (8),(10),(13), and (14). This way, instead of the MAP solution, the

maximum likelihood solution (ML) (\mathbf{s}^{ML} and d^{ML}) is obtained (using (13) and (14) without the term $-\log P[\mathbf{s}]$):

$$\mathbf{s}^{ML} = \arg \min_{\mathbf{s} \in \Omega^{n_T}} \{ \|\mathbf{z} - \mathbf{R}\mathbf{s}\|^2 \}, \quad (20)$$

$$d^{ML} = \|\mathbf{z} - \mathbf{R}\mathbf{s}^{ML}\|^2. \quad (21)$$

Therefore, to compute the max-log LLRs it is necessary to calculate a detection hypothesis (MAP solution for SISO and ML for NISO) and all of the *counter-hypothesis* metrics as this section describes. This involves a high computational cost. Some strategies like the clipping method or non-optimal algorithms can be used to reduce the computational cost, which leads to a reduction in performance.

The clipping method can be easily applied to reduce complexity [17]. Given a clipping parameter L_{clip} which ensures that $|L_{i,b}^E| \leq L_{clip}$, it is assumed that any $\bar{\delta}_{i,b}$ larger than $d^{MAP} + L_{clip}$ can be set to the value $d^{MAP} + L_{clip}$ and does not need to be computed exactly, thereby reducing the search of our detector and thus the complexity. However, the performance is degraded. This method is widely used in MIMO detection and it is important to note that the LLR values obtained by the max-log MAP SISO algorithms, which use the same clipping value L_{clip} , will be exactly the same.

3. Box optimization algorithm

The goal of this work is to use continuous constrained optimization techniques to implement different SO low-complexity algorithms. The BO method can be used to drastically reduce the computation required by MIMO signal detection algorithms in different ways. By solving the continuous unconstrained least squares problem

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s} \in \mathbb{C}^{n_T}} \|\mathbf{z} - \mathbf{R}\mathbf{s}\|^2, \quad (22)$$

by $\hat{\mathbf{s}} = (\mathbf{R}^H \mathbf{R})^{-1} \mathbf{R}^H \mathbf{z}$ and rounding $\hat{\mathbf{s}}$ to the nearest element of the constellation Ω , we obtain the known Zero-Forcing (ZF) estimator ($\hat{\mathbf{s}}_q$). If one or more

components of the $\hat{\mathbf{s}}$ vector are outside of the interval $[\min(\Omega_{\mathbb{R}}), \max(\Omega_{\mathbb{R}})]$ (it is assumed that Ω can be represented as $\Omega_{\mathbb{R}} \times \Omega_{\mathbb{R}}$), it can be said that $\hat{\mathbf{s}}$ is out of the constellation. It was reported in [13] that, in this case, by solving an auxiliary problem described as continuous constrained optimization, a better approximation to the ML solution than $\hat{\mathbf{s}}_q$ is given.

Consequently, the BO method is applied when $\hat{\mathbf{s}}$ is out of the constellation, solving the following auxiliary problem:

$$\begin{aligned} \mathbf{s}^b &= \arg \min_{\mathbf{s} \in \mathbb{C}^{n_T}} \|\mathbf{z} - \mathbf{R}\mathbf{s}\|^2 & (23) \\ \min(\Re(\Omega)) &\leq \Re(s_i) \leq \max(\Re(\Omega)), 1 \leq i \leq n_T \\ \min(\Im(\Omega)) &\leq \Im(s_i) \leq \max(\Im(\Omega)), 1 \leq i \leq n_T \end{aligned}$$

where s_i , $1 \leq i \leq n_T$, are the components of the \mathbf{s} vector in (1). Once (23) is solved, all of the components of \mathbf{s}^b are then rounded to the nearest element of the constellation Ω , obtaining \mathbf{s}_q^b . The solution search is bounded by a zone in the form of a box in (23), hence the name of Box Optimization. The BO algorithm that is used in this work to solve (23) is described in [14]. This BO algorithm is an adapted version of the algorithm proposed in [18] using specific features of the problem and thus notably reducing its computational cost. The BO algorithm described in [14] is applied during the sphere decoder search, to decide whether a partial solution should be (or not) further explored. This can happen in any level of the tree, and the cost of the application of the BO algorithm depends partly on the level of the tree being explored. Furthermore, the cost of the application of the BO algorithm depends on the number of inner iterations of the BO algorithm. This number is usually 1 (as was shown experimentally in [14]), but not always, and cannot be predicted in advance. Considering all these details, it is not possible to give a meaningful bound on the theoretical complexity of the BO algorithm

All of the SO detection algorithms proposed in this paper use the BO technique. The general method to apply the BO method to the SO detection is represented in Algorithm 1.

Algorithm 1 General method for application of the BO algorithm in MIMO detection.

Input: \mathbf{z} , \mathbf{R} , Ω , $\hat{\mathbf{s}}$

Output: \mathbf{s}_q^b

- 1: **if** $\hat{\mathbf{s}}$ is in of Ω^{n_T} box **then**
 - 2: $\hat{\mathbf{s}}$ is rounded to the nearest element of Ω obtaining \mathbf{s}_q^b
 - 3: **else**
 - 4: BO is applied to obtain \mathbf{s}^b as in (23)
 - 5: \mathbf{s}^b is rounded to the nearest element of Ω obtaining \mathbf{s}_q^b
 - 6: **end if**
-

Paper [13] claims that $\hat{\mathbf{s}}$ will be mainly out of the constellation in low SNR scenarios. However, the probability that $\hat{\mathbf{s}}$ was out or in was not provided in [13]. Furthermore, it can be observed that the results show improvements for all the signal-to-noise ratio (SNR) values, not only for low SNR since they also depends on the channel matrix condition number. For this reason, the probability that $\hat{\mathbf{s}}$ was out, P_{out} , has been analyzed. A mathematical derivation of P_{out} can be found in Appendix A. P_{out} has also been evaluated for different numbers of antennas, constellation orders, and SNR values. The results show that, for all the evaluated parameters, P_{out} is really high (greater than 70%).

3.1. Box Optimization with SD detection

The continuous constrained optimization technique has been used to help the SD detection algorithms reduce the computational complexity. One possibility proposed in [19] is the use of \mathbf{s}_q^b , given by Algorithm 1, to select the initial SD radius instead of the ZF estimator ($\hat{\mathbf{s}}_q$). The distance given by \mathbf{s}_q^b is usually closer to the d^{ML} distance than the one given by $\hat{\mathbf{s}}_q$ since the BO method will be applied with high probability. Thus, the number of expanding nodes and the complexity are reduced. Alternatively, a tighter radius estimation before the expansion of each node can be obtained. As explained in [20], the pruning

condition before the expansion of each node can be set to:

$$\tilde{\Lambda}_{i+1} \leq (r^2 - c), \quad (24)$$

where $\tilde{\Lambda}_{i+1}$ refers to Λ_{i+1} in (9) without *a priori* information. Paper [20] discussed several methods to compute the lower bound c , one of which is the use of a BO technique. However, the methods evaluated in [20] require high computational resources. On the other hand, [14] uses a BO method that includes a number of improvements and reduces the complexity. Consequently, an optimal *Box Optimization Hard Detector* (BOHD) was presented in [14]. The BOHD algorithm is much faster than standard ML SD detectors, especially for high order constellations or large numbers of antennas.

4. BO-based Soft MIMO detectors

In this section we investigate several possibilities for the MIMO detection problem. Particularly, in subsection 4.1 we present a new NISO detector called BOHD-SO algorithm. The proposed algorithm uses the BOHD hard output detector and the BO estimator to compute the LLR values, achieving a low complexity detector. Since the proposed BOHD-SO algorithm is suboptimal, it will be compared with the SFSD algorithm in terms of efficiency and accuracy. On the other hand two SISO MIMO detectors are presented in subsections 4.2 and 4.3. These algorithms are an extension of the algorithms proposed in [13]. In [13] the SISO RTS and STS algorithms was improved using the box optimization (BO) techniques, giving place to the BORTS and DTS algorithms. The BORTS and DTS algorithms was proposed in [13] for the NISO case. In these subsections, we extend these algorithms for the SISO detection. In this way the necessary modifications to incorporate the *a priori* information in the SISO-BORTS and SISO-DTS algorithms are presented in this subsection.

4.1. NISO BOHD-SO

The proposed algorithm is based on a repeated strategy to compute the *counter-hypothesis* distances. First, a tree search is run once, using the BOHD

Algorithm 2 BOHD-SO algorithm

Input: $\mathbf{z}, \mathbf{R}, \Omega$ **Output:** \mathbf{L}^I

- 1: \mathbf{s}^{ML} and d^{ML} computed by BOHD
 - 2: $\hat{\mathbf{s}} = (\mathbf{R}^H \mathbf{R})^{-1} \mathbf{R}^H \mathbf{z}$
 - 3: **if** $\hat{\mathbf{s}}$ is in of Ω^{n_T} box **then**
 - 4: $\mathbf{s}_{bo} = \hat{\mathbf{s}}$
 - 5: **else**
 - 6: BO is applied to obtain \mathbf{s}^b as in (23)
 - 7: $\mathbf{s}_{bo} = \mathbf{s}^b$
 - 8: **end if**
 - 9: **for** $i = 1$ to n_T **do**
 - 10: **for** $b = 1$ to k **do**
 - 11: \mathbf{s}_{bo} is rounded to the nearest element of $\tilde{\Omega}_{i,b}$ obtaining $\bar{d}_{i,b}$
 - 12: $L_{i,b}^I$ is computed using (17)
 - 13: **end for**
 - 14: **end for**
-

algorithm proposed in [14]. Thus, the ML solution is computed in an efficient way. Once \mathbf{s}^{ML} and d^{ML} have been calculated, the *counter-hypothesis* distances are obtained using a modified version of Algorithm 1. Algorithm 2 shows how BOHD-SO works.

From lines 3 to 8 of Algorithm 2, the general method presented in Algorithm 1 is applied but performing a slight variation. The output is not rounded to the nearest element of Ω because the rounding operation will be different for each *counter-hypothesis* metric. From lines 3 to 8 of Algorithm 2, the \mathbf{s}_{bo} vector is computed. This vector has already been calculated by the BOHD algorithm in the preprocessing stage (see Algorithm 1 in [14]). Thus, the calculation of \mathbf{s}_{bo} can be omitted in Algorithm 2, reducing the computational cost.

The rounding operation is done in line 11, where $\tilde{\Omega}_{i,b}$ is the “restricted” Ω . By $\tilde{\Omega}_{i,b}$, we mean that the considered constellation points are those that

have the corresponding bit complemented. To compute the *counter-hypothesis* distance associated with the $L_{i,b}^E$ value when the (i, b) bit of the \mathbf{s}^{ML} solution is equal to one, we only consider the constellation points with the (i, b) bit equal to zero. Therefore, we obtain all the *counter-hypothesis* distances, repeating this strategy for each bit.

The BO method could be executed for each *counter-hypothesis* metric by applying the original Algorithm 1 for each distance and considering $\tilde{\Omega}$ instead of Ω as the input argument. However, this means that the BO method has to be applied for each *counter-hypothesis* metric, increasing the computational cost. In this way, Algorithm 2 executes the BO method only once, reducing the complexity.

It was also verified by simulation that the performance in terms of BER of this algorithm is far from the max-log ML performance due to the simple way in which the *counter-hypothesis* distances are obtained. Lattice reduction (LR) techniques is commonly applied to improve the performance of low-complexity detectors for MIMO systems without increasing the complexity dramatically [21][22][23]. In the proposed algorithm, when LR is aided to the proposed algorithm as in [22], the performance results are very close to the max-log performance, as shown in the Results section. To implement the LR algorithm the Lenstra, Lenstra and Lovsz (LLL) algorithm [24] have been considered due to it offers a good trade-off between performance and complexity.

4.2. SISO-BORTS

The RTS algorithm described in [15] starts by solving (13) through a MAP SD algorithm. The Schnorr-Euchner sphere decoder (SESD) [25] is usually selected for this purpose. Once \mathbf{s}^{MAP} and d^{MAP} have been computed, the *counter-hypothesis* distances are obtained by running the SESD again for each bit, as described in [15]. Therefore, the SESD has to be executed $n_T \cdot k + 1$ times. Clearly the main disadvantage of this strategy is the increased complexity, especially for low SNR.

The large reduction in time and in visited nodes shown in [14] for the BOHD

hard detector makes it a perfect candidate to replace the standard SESD detector in the RTS algorithm. In [13], this idea was proposed and evaluated for NISO, showing a large reduction in complexity for the new RTS algorithm. In this work, we propose and evaluate the extension to the Sof-Input case given a SISO-BORTS algorithm. For this purpose, the original BOHD algorithm proposed in [14] has to be slightly modified. The original algorithm computes the ML solution, regardless of the prior information provided by the decoder. The *a priori* information has to be incorporated to compute (9) in each evaluated node. Besides, it is important to note that the pruning criteria evaluated in (24) has to be computed as:

$$\tilde{\Lambda}_{i+1} \leq r^2 - c - \sum_{j=1}^i \min \{-\log P[s_j]\}. \quad (25)$$

Thus, the tightening of the tree-pruning criteria using (25) reduces the number of visited nodes compared to (24). The SISO-BORTS algorithm described above has been implemented for the cases with and without clipping. Furthermore, the resulting algorithm would be trivially parallelizable (like the original RTS).

In [17], it is explained how the clipping technique is easily included in the RTS strategy, leading to a reduction in search complexity. Since the MAP solution is computed previously, when the SESD reruns to compute each extrinsic metric, the initial maximum radius will be upper bounded by $d^{MAP} + L_{clip}$. This strategy considerably reduces the number of nodes explored and the computation cost.

For the proposed algorithm, when clipping is applied and the BOHD algorithm is used again to compute the *counter-hypothesis*, we can use the \mathbf{s}^b estimated vector by box optimization to bound the distance. We know that any possible transmitted vector will give us a greater Euclidean distance than the distance given by \mathbf{s}^b . Therefore, if we satisfy the following equation

$$\left(\|\mathbf{z} - \mathbf{R}\mathbf{s}^b\|^2 + \min_{\mathbf{s} \in \Omega^{n_T}} \{-\log P[\mathbf{s}]\} \right) > (d^{MAP} + L_{clip} + (1 - 2x_{i,b}^{MAP}) L_{i,b}^A) \quad (26)$$

we can set the $\bar{\delta}_{i,b}$ value that we are computing to $d^{MAP} + L_{clip}$ and avoid the

tree search, thereby reducing the complexity. It is important to remember that the indices (i, b) refer to the b bit in the i antenna. Any possible transmitted vector will give us a $\bar{\delta}_{i,b}$ distance larger than $d^{MAP} + L_{clip}$.

4.3. SISO-DTS

The STS algorithm was proven to be more efficient than the RTS algorithm in [9], The STS algorithm has the standard SESD structure and it transverses the tree only once, calculating all the necessary metrics to compute (19) with only one execution. In order to detect all of these metrics simultaneously, the radius must be larger (at least as large as $\max(\bar{\delta}_{i,b})$) and calculated again before computing any node or leaf. The STS algorithm without clipping cannot be easily combined with BO techniques, since the STS radius must be larger and the BO obtains an extremely tight bounds of the radius.

When clipping is used the situation is different as explained in [13]. The work in [9] describes how LLR clipping can be incorporated into the STS algorithm by modifying the list administration when the current MAP hypothesis changes. If clipping is applied to the STS algorithm, all of the nodes with a partial distance contained in the interval $[d^{MAP}, d^{MAP} + L_{clip}]$ are visited. The minimum number of nodes to be visited should be the number of nodes with a partial distance within this interval. However, the STS algorithm proceeds like the SESD detector; it starts with the initial distance as $+\infty$ and updates it whenever STS finds a feasible leaf. These first leaves may have a partial distance that is larger than d^{MAP} . As long as STS does not find the MAP solution, the algorithm expands nodes with partial distances that are larger than $d^{MAP} + L_{clip}$. This means that some (possibly many) extra nodes with a partial distance greater than $d^{MAP} + L_{clip}$ may have to be expanded. Taking advantage of the availability of the fast BOHD algorithm (which has been extended in a previous section to the Soft-Input case) to reduce the number of nodes, we can first compute \mathbf{s}^{MAP} and d^{MAP} . Since the \mathbf{s}^{MAP} has already been computed, the maximum value for the $\bar{\delta}_{i,b}$ can be set to $d^{MAP} + L_{clip}$, decreasing the number of visited nodes. As we stated in the explanation of the STS algorithm, the radius is calculated

again before computing of any node or leaf. It must be noted that the radius recalculation is a process that is quite expensive. In terms of computing time it is beneficial to avoid this recalculation; however, if no recalculation is made, the number of visited nodes can be too large. Taking advantage of the BOHD algorithm by computing the \mathbf{s}^{MAP} solution in a previous step, we can use the following as a pruning condition. Given a partial distance d_p , if d_p is greater than $d^{MAP} + L_{clip}$, this node is pruned; otherwise, the node is expanded. The outcome of this new modification (which is an extension of the DTS algorithm proposed previously in [13] for NISO) should be that the number of visited nodes increases and the average time complexity decreases (compared with the original STS algorithm) because most of radius recalculations are avoided.

5. Results

Numerical simulations have been performed in order to evaluate the proposals. The numerical results have been computed by Monte Carlo simulations with high enough channel realizations to provide statistically reliable results. To this end a number of minimum simulated frames has been defined. A BICM scheme with random interleaved has been considered. In the simulations each channel realization remained constant during a block of 16 transmitted vectors. A convolutional encoder of codeword size 2304 and rate 1/2 was used. The generator polynomials $[133_o, 171_o]$, constraint length 7 and max-log BCJR channel decoder were chosen. The number of iterations in the iterative receiver was set to 4.

First, we present the results obtained through simulation for NISO detection. The SFSD algorithm was chosen for comparison because, it almost achieves the max-log ML performance with low complexity since it is a suboptimal algorithm. The performances of the algorithms were compared in terms of Bit Error Rate (BER) and computational cost (floating point operations (flops)). A 4×4 complex MIMO scenario with 16 and 64 QAM constellations was also chosen.

Figure (3a) shows the BER performance for the proposed BOHD-SO algo-

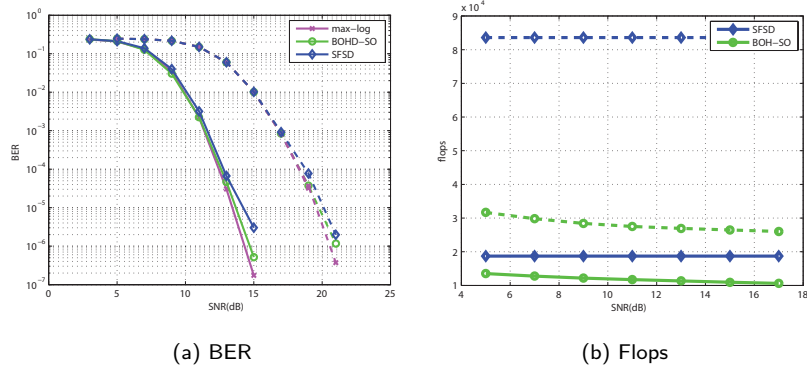


Figure 3: BER and number of flops for the BOHD-SO and SFSD algorithms in a 4×4 MIMO system with a 16-QAM constellation (continuous lines) and a 64-QAM constellation (discontinuous lines).

rithm, the SFSD algorithm, and the max-log cases. The curves show how the BOHD-SO and SFSD algorithms almost achieve max-log performance for both constellation sizes, with BOHD-SO being slightly better than SFSD (especially for 64-QAM constellations). The performance of the BOHD-SO detector in a system with higher code rate has been also analyzed achieving also near max-log performance.

The computational cost in terms of flops is represented in Figure (3b), comparing the efficiency of BOHD-SO and SFSD. The curves show how BOHD-SO needs less flops than SFSD, achieving better accuracy in terms of BER. The max-log performance can also be achieved by the STS or SISO-DTS algorithms, but increasing the cost of the BOHD-SO algorithm.

As mentioned in the introduction section, the performance of two max-log MAP algorithms without clipping is equal. A similar situation occurs when clipping is applied using the same clipping value. This has been verified by the measurement of the BER performance, which is illustrated by Figure 4. This figure represents the BER performance from the first iteration (the detector and decoder have been run once) to the fourth iteration (the detector and decoder have been run four times). The results for STS, SISO-BORTS, and SISO-DTS

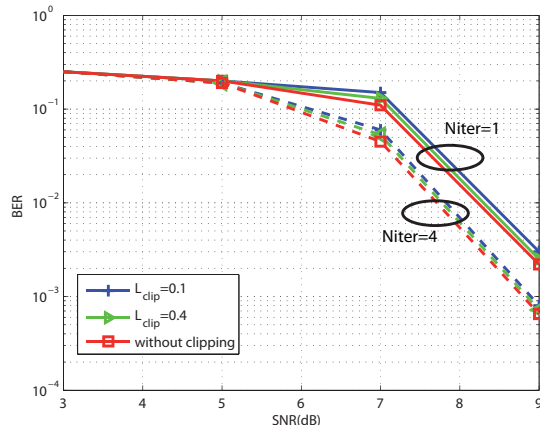


Figure 4: Max-log BER performance for a 4×4 MIMO system with a 16-QAM and $L_{clip} = 0.1$

are exactly the same for the three algorithms, as was expected.

The complexity results obtained for the extensions to the Soft-Input case of the DTS and BORTS algorithms are presented in Figures 5, 6, and 7. The computational complexity of the SISO algorithms is evaluated by the average number of visited nodes, flops, computing times, and number of comparisons. For each SNR, the average of 10000 realizations for each parameter were recorded. For this type of algorithms, the number of visited nodes is chosen as the main metric because it is independent of the computing platform. However, we also record the computing times because we have large variations in the cost of the expansion of a single node. The number of flops is another metric that is often used; however, in this case, it can be somewhat misleading. The reason is that these algorithms perform a large number of comparisons, which in some cases is larger than the number of flops. For this reason, we also record the number of comparisons.

Figure (5) summarizes the numerical results in the case without clipping. The results show that STS is faster for 16-QAM modulation. Furthermore, STS requires less flops than SISO-BORTS. However, the number of expanded nodes is similar for both algorithms and the number of comparisons is reduced by the

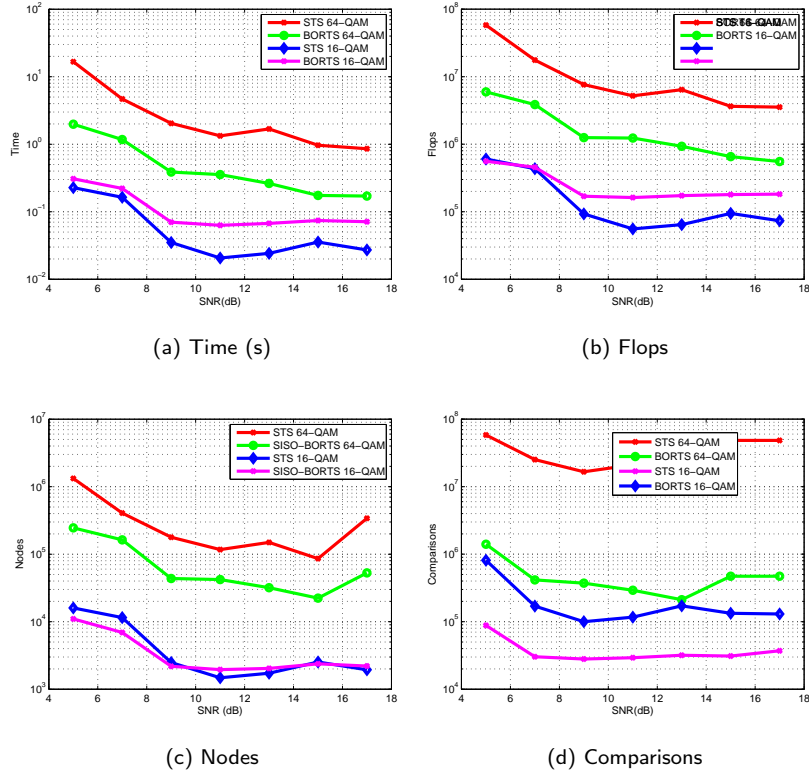


Figure 5: Average computation cost parameters for the SISO STS and SISO-BORTS algorithms for a 4×4 MIMO system, 16-QAM and 64-QAM without clipping and 4 iterations at the iterative receiver.

SISO-BORTS algorithm. The results for the 64-QAM constellations are really different. The SISO-BORTS algorithm outperforms the STS algorithm in all evaluated parameter values. The percentage cost reduction of the computational cost between SISO-BORTS and STS for each evaluated parameter has been represented in Table 2, for a 4×4 MIMO system using a 64-QAM constellation without clipping. A higher percentage indicates a lower computational cost of the SISO-BORTS algorithm with regard to the STS algorithm. Table 2 shows the great advantage of using SISO-BORTS instead of the STS algorithm without clipping when 64-QAM constellation is employed.

Figures (6) and (7) represent the results obtained for the STS, SISO-BORTS,

Without clipping

SNR(dB)	5	7	9	11	13	15	17
Time	88.12%	74.87%	80.99%	73.29%	84.41%	81.95%	87.80%
Flops	89.75%	78.14%	83.64%	76.30%	85.46%	82.04%	90.26%
Nodes	81.44%	59.88%	75.56%	64.08%	78.66%	74.02%	84.43%
Comparisons	98.88%	97.59%	98.34%	97.75%	98.61%	98.2402%	99.03%

Table 2: Percentage reduction of the computational performance parameters between SISO-BORTS and STS for a 4×4 MIMO system and a $64 - QAM$ constellation without clipping. A higher percentage reduction indicates a lower computational cost of the SISO-BORTS algorithm with regard to the STS algorithm.

and SISO-DTS algorithms applying the clipping method. The experiments were done using two different clipping parameter values (0.1 and 0.4). In the $16 - QAM$ case, STS and SISO-DTS exhibit a better performance when compared with SISO-BORTS. If we compare the two best algorithms for this case, it can be observed that SISO-DTS is faster than RTS for both clipping values and all of the evaluated parameters. When the modulation order and the clipping parameter increase, the SISO-BORTS algorithm becomes comparatively more efficient than the STS algorithm.

One phenomenon that requires attention is that SISO-DTS is substantially faster than STS in terms of computing times and flops; however, SISO-DTS expands an equal number or more nodes than STS. Clearly the computing time per node of SISO-DTS is smaller than STS due to the large number of radius recalculations, which has been avoided in STS. This statement is clearly reflected in Table 3, which represents the percentage of reduction using the SISO-DTS algorithm instead of STS for a $64 - QAM$ constellation and $L_{clip} = 0.1$.

6. Conclusion

The proposed detection algorithms exhibit very meaningful properties. The BOHD-SO algorithm is a non-optimal NISO algorithm that almost achieves the max-log performance with very low complexity, obtaining a more accurate response than the SFSD algorithm and less complexity in terms of flops when

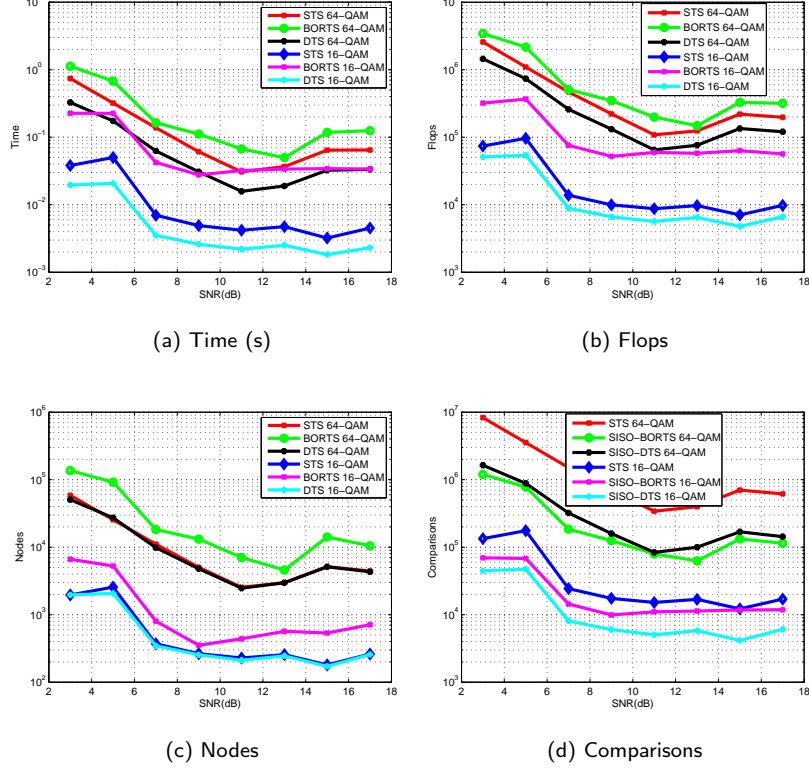


Figure 6: Average computation cost parameters for the SISO STS, SISO-BORTS and SISO-DTS algorithms for a 4×4 MIMO system, 16-QAM and 64-QAM with $L_{clip} = 0.1$ and 4 iterations at the iterative receiver.

Clip 0.1

SNR(dB)	5	7	9	11	13	15	17
Time	55.73%	45.56%	54.76%	49.46%	48.75%	48.43%	49.49%
Flops	43.88%	32.77%	44.79%	40.52%	40.32%	38.67%	38.77%
Nodes	13.57%	-6.62%	11.38%	4.52%	2.81%	0.57%	0.91%
Comparisons	80.26%	75.06%	78.79%	76.87%	75.42%	74.92%	75.88%

Table 3: Percentage reduction of the computational cost parameters between SISO-DTS and STS for a 4×4 MIMO system and a 64-QAM constellation with $L_{clip} = 0.1$. A higher percentage reduction indicates a lower computational cost of the SISO-DTS algorithm with regard to the STS algorithm.

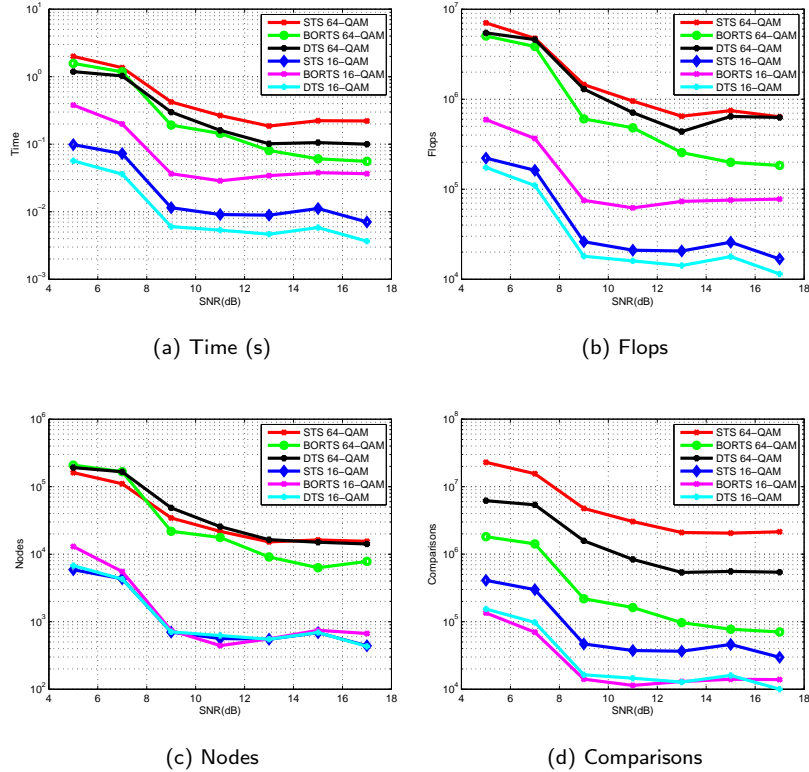


Figure 7: Average computation cost parameters for the SISO STS, SISO-BORTS and SISO-DTS algorithms for a 4×4 MIMO system, 16-QAM and 64-QAM with $L_{clip} = 0.4$ and 4 iterations at the iterative receiver.

LR is applied. Thus, the BOHD-SO algorithm is a good option when a non-iterative receiver is used, providing good performance with low complexity.

On the other hand, the extension of SDTS and BORTS to the Soft-Input case was studied and analyzed. The main feature of these extension is that by achieving the MAP solution, the algorithms obtain better results in terms of computational cost than the STS algorithm when the constellation size increases. The advantages of SISO-BORTS with respect to the STS algorithm are also quite remarkable for large problems; for example, this algorithm reduced the computational time in 64-QAM up to 80% in the case without clipping and up to 40% with a clipping value of 0.4. The SISO-DTS algorithm that is

used with clipping, outperformed the computational cost of the STS algorithm for all of the cases and parameters tested. Therefore, in an iterative receiver and large systems, the SISO-BORTS algorithm is the best option when clipping is not used. When clipping is applied, the SISO-DTS algorithm provides the max-log MAP performance with the least complexity, regardless of the system size.

Appendix A. Proof of Probability of being out of the box

The BO algorithm is used when the ZF estimator is out of the box that is defined by the limits of the constellation. The ZF estimator is calculated in (A.1) as:

$$\hat{\mathbf{s}} = \mathbf{H}^\dagger \mathbf{y} = \mathbf{s} + \mathbf{H}^\dagger \mathbf{v}, \quad (\text{A.1})$$

where $\mathbf{H} \in \mathbb{C}^{n_R \times n_T}$ with $(n_R \geq n_T)$, $\mathbf{y} \in \mathbb{C}^{n_R \times 1}$, $\mathbf{v} \in \mathbb{C}^{n_R \times 1}$ and $\mathbf{s} \in \mathbb{C}^{n_T \times 1}$. The BO algorithm is applied when any component of $\hat{\mathbf{s}}$ is outside of the box that is delimited by the constellation. Therefore, BO execution probability can be computed as the union of n_T compatible events, since it will be the probability that one or more components of $\hat{\mathbf{s}}$ get outside. In a simple case with only two antennas, we would have the probability of the union of two events:

$$\begin{aligned} P_{out} &= P(s_1 \cup s_2)_{out} \\ &= P(s_1)_{out} + P(s_2)_{out} - P(s_1 \cap s_2)_{out} \\ &= P(s_1)_{out} + P(s_2)_{out} - P(s_1)_{out} \cdot P(s_2)_{out}, \end{aligned} \quad (\text{A.2})$$

where $P(s_1)_{out}$ represents the probability of being out the box of the first component in $\hat{\mathbf{s}}$ and $P(s_2)_{out}$ represents the probability of the second component. On the other hand, $P(s_1 \cap s_2)_{out}$ can be calculated as $P(s_1)_{out} \cdot P(s_2)_{out}$ because the events are independent. Extending this result to the n_T transmit antennas

event case, we have:

$$\begin{aligned}
P_{out} = & \sum_{i=1}^{n_T} P(s_i)_{out} - \sum_{i \neq j} P(s_i \cap s_j)_{out} + \sum_{i \neq j \neq k} P(s_i \cap s_j \cap s_k)_{out} - \\
& - \dots + (-1)^{(n_T+1)} P \left(\bigcap_{i=1}^{n_T} s_i \right)_{out}, \tag{A.3}
\end{aligned}$$

where $P(s_i)_{out}$ represents the probability of being out the box of the s_i component. The next step is to compute the $P(s_i)_{out}$ values.

As can be observed in (A.1), the s_i component may or may not be outside, depending on the noise variance and on the channel matrix pseudoinverse. First of all, when the channel is a diagonal matrix with its diagonal elements equal to one, the probability only depends on the noise. Thus, the probability of a component being out ($P(s_i)_{out}$) will be the probability that the real part is out or the probability that the imaginary part is out, or both. In other words,

$$P(s_i)_{out} = 2p(s_i)_{out} - p(s_i)_{out}^2. \tag{A.4}$$

The probability that the real or imaginary part is out is denoted by $p(s_i)_{out}$, so (A.4) can be computed as a union of two independent events.

We will assume that there exists a set $\Omega_{\mathbb{R}}$ such that the constellation Ω can be obtained as $\Omega_{\mathbb{R}} \times \Omega_{\mathbb{R}}$. Therefore, if Ω is an $M - QAM$ constellation, each component (real and imaginary) of the constellation belongs to a L-PAM constellation with L equal to \sqrt{M} . Let the sent symbol be the m -th constellation point and let $p(s_{i,m})_{out}$ be the probability of getting outside, which can be calculated as:

$$p(s_{i,m})_{out} = p(n < [1 - m]d) + p(n > [L - m]d), \quad m = 1, \dots, L. \tag{A.5}$$

where d is the distance that the contiguous elements of the constellation are separated and n is a gaussian noise of zero mean and σ_n^2 variance. Since it can be checked by symmetry that

$$p(s_{i,m})_{out} = p(s_{i,L-m+1})_{out}, \tag{A.6}$$

if the symbols are equiprobable, the average probability will be:

$$p(s_i)_{out} = \frac{\sqrt{M}}{M} \cdot 2 \sum_{m=1}^{L/2} p(s_{i,m})_{out} \quad (\text{A.7})$$

$$= \frac{1}{\sqrt{M}} \cdot 2 \sum_{m=1}^{L/2} \{p(n < [1 - m]d) + p(n > [L - m]d)\}. \quad (\text{A.8})$$

Given the symmetry of $Q(x)$ we have

$$p(n > z) = p(n < -z) = Q\left(\frac{z}{\sigma_n}\right) \quad (\text{A.9})$$

with

$$Q(x) = \frac{1}{2} \operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right). \quad (\text{A.10})$$

Thus:

$$p(n < [1 - m]d) = p(n > [m - 1]d) = Q\left(\frac{[m - 1]d}{\sigma_n}\right) \quad (\text{A.11})$$

$$p(n > [L - m]d) = Q\left(\frac{[L - m]d}{\sigma_n}\right). \quad (\text{A.12})$$

It is necessary to describe the relationship between the quotient d/σ_n and the SNR. If the constellation is polar, then the coordinates of the m -th symbol (i.e., $E_m = (2m - 1)d/2$) will be the opposite of the $M - m + 1$ -th. Thus, their energies will be equal. In this case, the energy average by symbol can be expressed as:

$$E_s = \frac{1}{L} \sum_{m=1}^L E_m = \frac{2}{L} \cdot \sum_{m=1}^{L/2} \left[(2m - 1) \frac{d}{2}\right]^2 = \frac{2}{L} \cdot \sum_{m=1}^{L/2} \left[(2m - 1)^2 \frac{d^2}{4}\right] = \frac{d^2}{12} (L^2 - 1). \quad (\text{A.13})$$

Then

$$\text{SNR} = \frac{E_s \cdot n_T}{\sigma^2} = \frac{\frac{1}{12} d^2 (L^2 - 1)}{\sigma^2} \cdot n_T. \quad (\text{A.14})$$

According to the previous equation

$$\frac{d}{\sigma} = \sqrt{\frac{12}{L^2 - 1} \cdot \frac{\text{SNR}}{n_T}} \quad (\text{A.15})$$

However, a non-ideal channel will affect (A.7). In (A.1), it can be observed how the noise varies depending on the pseudoinverse of the channel, $\tilde{\mathbf{n}} = \mathbf{H}^\dagger \mathbf{n}$.

Thus, the noise in each receiver antenna is computed as

$$\check{n}_i = \mathbf{H}_i^\dagger \mathbf{n}, \quad (\text{A.16})$$

where the subindex \mathbf{H}_i^\dagger denotes the i -th row of \mathbf{H}^\dagger . Therefore, the power of the new noise in each receiver antenna can be obtained as

$$E \left\{ \check{n}_i \check{n}_i^H \right\} = E \left\{ \mathbf{H}_i^\dagger \mathbf{n} \mathbf{n}^H (\mathbf{H}_i^\dagger)^H \right\} = \mathbf{H}_i^\dagger E \left\{ \mathbf{n} \mathbf{n}^H \right\} (\mathbf{H}_i^\dagger)^H \quad (\text{A.17})$$

$$= \mathbf{H}_i^\dagger \sigma_n^2 (\mathbf{H}_i^\dagger)^H \quad (\text{A.18})$$

Then, computing the SVD decomposition of \mathbf{H} as

$$\mathbf{H} = \mathbf{U} \mathbf{S} \mathbf{V}^H = \mathbf{U} \begin{bmatrix} \mathbf{\Sigma} \\ \mathbf{0} \end{bmatrix} \mathbf{V}^H, \quad (\text{A.19})$$

we obtain

$$\mathbf{H}_i^\dagger = \mathbf{V}_i \begin{bmatrix} \mathbf{\Sigma}^{-1} & \mathbf{0} \end{bmatrix} \mathbf{U}^H = \mathbf{V}_i \begin{bmatrix} \mathbf{\Sigma}^{-1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U}_1^H \\ \mathbf{U}_2^H \end{bmatrix} = \mathbf{V}_i \mathbf{\Sigma}^{-1} \mathbf{U}_1^H. \quad (\text{A.20})$$

Therefore

$$\mathbf{H}_i^\dagger (\mathbf{H}_i^\dagger)^H = \mathbf{V}_i \mathbf{\Sigma}^{-1} \mathbf{U}_1^H \mathbf{U}_1 \mathbf{\Sigma}^{-1} \mathbf{V}_i^H \quad (\text{A.21})$$

$$= \mathbf{V}_i \mathbf{\Sigma}^{-2} \mathbf{V}_i^H. \quad (\text{A.22})$$

Thus, the power of the new noise in each receiver antenna is computed as

$$\sigma_{\check{n}_i}^2 = \sigma_n^2 \mathbf{H}_i^\dagger (\mathbf{H}_i^\dagger)^H \quad (\text{A.23})$$

$$= \sigma_n^2 \mathbf{V}_i (\mathbf{\Sigma}^{-2}) \mathbf{V}_i^H. \quad (\text{A.24})$$

Therefore, taking into account the channel matrix. the $p(s_i)_{out}$ probability is calculated as

$$p(s_i)_{out} = \frac{1}{\sqrt{M}} \cdot 2 \sum_{m=1}^{L/2} \left\{ Q \left(\frac{[m-1]d}{\sigma_{\check{n}_i}} \right) + Q \left(\frac{[L-m]d}{\sigma_{\check{n}_i}} \right) \right\} \quad (\text{A.25})$$

Table A.4 represents P_{out} using (A.3) versus the experimental values. These probabilities have been computed as an average of 1000 realizations and have been represented for different numbers of antennas, constellation orders and SNR values.

Table A.4: Experimental/theoretical probabilities of \hat{s} be out of the constellation in %.

SNR(dB)	4-QAM		16-QAM		64-QAM	
	4×4	8×8	4×4	8×8	4×4	8×8
0	99,90\99,92	1\1	98,94\98,99	99,99\99,99	97,62\97,57	99,97\99,97
5	99,77\99,81	1\1	96,54\96,70	99,92\99,93	91,85\91,44	99,60\99,55
10	99,68\99,71	1\1	93,49\93,91	99,66\99,73	83,10\82,45	98,00\97,74
15	99,64\99,65	1\1	91,38\91,84	99,35\99,44	74,61\74,64	94,81\94,63
20	99,60\99,62	1\1	90,46\90,72	99,14\99,21	69,35\69,77	91,45\91,73

References

- [1] B. M. Hochwald, S. Ten Brink, Achieving near-capacity on a multiple-antenna channel, *IEEE transactions on communications* 51 (3) (2003) 389–399.
- [2] P. Robertson, E. Villebrun, P. Hoeher, A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain, in: *Communications, 1995. ICC'95 Seattle, 'Gateway to Globalization', 1995 IEEE International Conference on*, Vol. 2, IEEE, 1995, pp. 1009–1013.
- [3] M. Tüchler, A. C. Singer, R. Koetter, Minimum mean squared error equalization using a priori information, *IEEE Transactions on Signal processing* 50 (2002) 673–683.
- [4] J. Boutros, N. Gresset, L. Brunel, M. Fossorier, Soft-input Soft-output lattice sphere decoder for linear channels, in: *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE, San Francisco, CA, USA, 2002*, pp. 1583–1587.
- [5] B. Steingrimsson, T. Luo, K. M. Wong, Soft quasi-maximum-likelihood detection for multiple-antenna wireless channels, *IEEE Transactions on Signal Processing* 51 (11) (2003) 2710–2719.
- [6] S. Båro, J. Hagenauer, M. Witzke, Iterative detection of MIMO transmis-

- sion using a list-sequential (LISS) detector, in: Communications, 2003. ICC '03. IEEE International Conference on, Vol. 4, 2003, pp. 2653–2657 vol.4.
- [7] H. Vikalo, B. Hassibi, T. Kailath, Iterative decoding for MIMO channels via modified sphere decoder, *IEEE Transactions on Wireless communications* 3 (6) (2004) 2299–2311.
- [8] C. Studer, A. Burg, H. Bölcskei, Soft-output sphere decoding: algorithms and VLSI implementation, *IEEE Journal on Selected Areas in Communications* 26 (2) (2008) 290–300.
- [9] C. Studer, H. Bölcskei, Soft-Input Soft-Output Single Tree-Search Sphere Decoding, *IEEE Transactions on Information Theory* 56 (10) (2010) 4827–4842.
- [10] L. G. Barbero, T. Ratnarajah, C. Cowan, A low-complexity soft-MIMO detector based on the fixed-complexity sphere decoder, in: Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, IEEE, 2008, pp. 2669–2672.
- [11] D. L. Milliner, E. Zimmermann, J. R. Barry, G. Fettweis, A fixed-complexity smart candidate adding algorithm for soft-output MIMO detection, *IEEE Journal of Selected Topics in Signal Processing* 3 (6) (2009) 1016–1025.
- [12] S. Chen, T. Zhang, Y. Xin, Relaxed-best MIMO signal detector design and vlsi implementation, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 15 (3) (2007) 328–337.
- [13] V. Garcia-Molla, M. A. Simarro, F. J. Martínez-Zaldívar, A. Gonzalez, A. M. Vidal, Maximum likelihood soft-output detection through sphere decoding combined with box optimization, *Signal Processing* 125 (2016) 249–260.

- [14] V. Garcia-Molla, A. Vidal, A. Gonzalez, S. Roger, Maximum likelihood detection through sphere decoding combined with box optimization, *Signal Processing* 98 (2014) 287–294.
- [15] R. Wang, G. B. Giannakis, Approaching MIMO channel capacity with soft detection based on hard sphere decoding, *IEEE Transactions on Communications* 54 (2006) 587–590.
- [16] G. Caire, G. Taricco, E. Biglieri, Bit-interleaved coded modulation, *IEEE transactions on information theory* 44 (3) (1998) 927–946.
- [17] M. S. Yee, Acoustics, speech, and signal processing, 2005. proceedings.(icassp'05). iee international conference on, in: *Proc. IEEE ICASSP*, Vol. 3, 2005, pp. 1013–1016.
- [18] A. Björck, *Numerical methods for least squares problems*, Siam, 1996.
- [19] X.-W. Chang, Q. Han, Solving box-constrained integer least squares problems, *IEEE Transactions on wireless communications* 7 (1) (2008) 277–287.
- [20] M. Stojnic, H. Vikalo, B. Hassibi, Speeding up the Sphere Decoder with H^∞ and SDP inspired lower bounds, *IEEE Transactions on Signal Processing* 56 (2) (2008) 712–726.
- [21] C. Windpassinger, R. F. Fischer, Low-complexity near-maximum-likelihood detection and precoding for mimo systems using lattice reduction, in: *Information Theory Workshop, 2003. Proceedings. 2003 IEEE*, IEEE, 2003, pp. 345–348.
- [22] H. Yao, G. W. Wornell, Lattice-reduction-aided detectors for mimo communication systems, in: *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE*, Vol. 1, IEEE, 2002, pp. 424–428.
- [23] X. Ma, W. Zhang, Performance analysis for mimo systems with lattice-reduction aided linear equalization, *IEEE Transactions on Communications* 56 (2).

- [24] A. K. Lenstra, H. W. Lenstra, L. Lovász, Factoring polynomials with rational coefficients, *Mathematische Annalen* 261 (4) (1982) 515–534.
- [25] E. Agrell, T. Eriksson, A. Vardy, K. Zeger, Closest point search in lattices, *IEEE transactions on information theory* 48 (8) (2002) 2201–2214.