A Dominance-Based Soft-Input Soft-Output MIMO Detector With Near-Optimal Performance

Giuseppe Papa, *Student Member, IEEE*, Domenico Ciuonzo, *Student Member, IEEE*, Gianmarco Romano, *Member, IEEE*, and Pierluigi Salvo Rossi, *Senior Member, IEEE*

***Abstract*—Iterative detection-and-decoding for multi-input multi-output (MIMO) communication systems require a soft-input soft-output (SISO) detection algorithm, which, in the optimal formulation, is well known to be exponentially complex in the number of transmitting antennas. This paper presents a novel SISO detector for MIMO systems, named SISO king decoder. It is a tree-search branch-and-bound algorithm, which exploits the properties of the channel matrix and the a-priori information on the transmitted bits to reduce the overall computational complexity. The proposed algorithm is compared with the SISO single tree-search sphere decoder [1]. Simulation results are provided to show the complexity reduction without relevant performance loss in terms of bit-error rate.**

***Index Terms*—Iterative detection-and-decoding system (IDDS), max-log, multi-input multi-output (MIMO), soft-input soft-output (SISO) detection, tree-search.**

I. Introduction

MULTI-INPUT multi-output (MIMO) communication enables systems with high data-rate requirements. It is known that their capacity increases with the minimum between the number of transmit antennas and the number of receive antennas [2]. However, a strong limitation to MIMO systems implementation is given by the computational complexity of the optimal detection (which is known to be exponential with the number of bits per transmission) [3]. Therefore, several low-complexity suboptimal alternatives have been proposed in the literature, such as branch-and-bound algorithms [4], lattice-based approaches [5] and other tree-search algorithms as the A^* algorithm [6].

A mature approach for MIMO communication systems is the development of iterative receivers [7], [8]. The turbo concept, originally proposed for capacity-achieving error correcting codes [9], is extended to iterative receivers [10], which are used in coded MIMO. The iterative process involves the soft-input soft-output (SISO) detector and the SISO channel decoder, which exchange soft information to improve performance at each iteration at the expense of a longer processing time. In SISO detection for MIMO systems, *a good balance between performance and complexity is crucial*.

Several algorithms with reduced complexity have been proposed in the literature. A sub-optimal solution limited to the case of quadrature phase-shift keying (QPSK) signaling and based on the semidefinite relaxation of the detection problem is presented in [11], which provides a rigorous analysis of the computational complexity. An intermediate approach between optimal SISO detection and Max-Log approximated detection [3], named partial marginalization (PM), is proposed in [12]. In the PM algorithm the tradeoff between performance and runtime, which is constant and fully predictable, is set through a tuning parameter. Practical implementations of soft-output detectors are the low-cost implementation for NVIDIA compute unified device architecture (CUDA) graphics processing unit (GPU) in [13] and the VLSI architecture and application-specific integrated circuit (ASIC) synthesis in [14]. An ASIC implementation of a SISO detector based on the minimum mean square error parallel interference cancellation (MMSE-PIC) is proposed in [15].

In the literature, the term *sphere decoder* (SD) refers to a collection of extremely efficient tree-search algorithms, providing optimal or suboptimal solutions with reduced computational complexity with respect to (w.r.t.) the exhaustive search of maximum likelihood (ML) detection. Inspired from the work on vector search in lattices [16], [17], several low-complexity algorithms based on SD have been proposed, e.g. ML decoding for channels with memory [18] and for multidimensional modulations in fading channels [19]. In the context of multi-antenna systems, SD has been extended to both uncoded and space-time coded transmissions [20]. Description and performance comparison of different methods for SD-based ML detection are found in [21], [22]. SD can be applied to underdetermined systems as well, i.e., when the number of transmit antennas is greater than the number of receive antennas. However in the latter case no pruning can be performed at the first levels of the tree. To avoid this problem in [23], [24] specific versions of SD have been proposed. Other SD algorithms, approaching near-ML performance and suitable for implementation with very-large-scale integration (VLSI) architectures, have been proposed in [25].

Manuscript received October 26, 2013; revised February 28, 2014, June 24, 2014, and September 18, 2014; accepted October 27, 2014. Date of publication November 4, 2014; date of current version December 15, 2014. This work was partially supported by POR Campania FSE 2007-2013 "Embedded Systems." The associate editor coordinating the review of this paper and approving it for publication was M. Matthaiou.

G. Papa and G. Romano are with the Department of Industrial and Information Engineering, Second University of Naples, 81031 Aversa (CE), Italy (e-mail: giuseppe.papa@unina2.it; gianmarco.romano@unina2.it).

D. Ciuonzo is with the Department of Electrical Engineering and Information Technologies (DIETI), University of Naples "Federico II", 80125 Naples, Italy (e-mail: domenico.ciuonzo@ieee.org).

P. Salvo Rossi is with the Department of Industrial and Information Engineering, Second University of Naples, 81031 Aversa (CE), Italy. and also with the Department of Electronics and Telecommunications, Norwegian University of Science and Technology, 7491 Trondheim, Norway (e-mail: salvorossi@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCOMM.2014.2367013

The SISO version of the SD proposed in [1] is a single tree-search algorithm able to get the same performance as the Max-Log approximated detector [3] at significantly reduced computational complexity. It is based on the same single tree-search mechanism proposed in [26] and presents the additional remarkable feature of tuning the tradeoff between performance and complexity reduction by adjusting only one parameter (see Section VI and [1] for more details about the "clipping" technique). A formal definition of optimality and optimal efficiency (in terms of complexity) is given in [27] for the class of soft-output detection algorithm based on the SD, and a soft-output SD, being optimal under the aforementioned properties, is presented. A soft-output extension of the hard-output fixed-complexity SD (FSD) [28] is presented in [29]. Another important point in practical MIMO receivers is the worst-case computational complexity, which influences directly the system data-rate. A soft-output detector with a fixed upper limit in decoding complexity is proposed in [30]. The list sphere decoder (LSD) is another suboptimal algorithm originally proposed in [3]. It is a tree-search algorithm and reduces the computational complexity pruning those nodes with a metric larger than a given radius threshold. A critical point in the LSD is the choice of the pruning radius, which trades performance and complexity. A novel version of the LSD is proposed in [31] and is based on a probabilistic method to select the pruning radius. In [32] the authors propose an exact Max-Log detector based on multiple hard SD steps (the number of needed steps is equal to the number of transmit antennas plus one) and a suboptimal solution, which needs only one hard SD step, leading to a remarkable complexity saving. Hardware implementations of SISO detectors based on SD have been proposed, such as the field-programmable gate array (FPGA) prototype and the ASIC synthesis in [33]. Furthermore, in [34] the authors extend the FSD [28] to the SISO case and propose a VLSI implementation.

Contributions: In this paper we present an SISO detection algorithm for MIMO systems, named SISO king decoder (KD), which has been preliminarily investigated in [35]. The proposed algorithm generalizes to the SISO case the KD developed for the hard-input hard-ouput case in [36] for M-ary quadrature amplitude modulation (M-QAM) constellations and in [37] for phase-shift keying (PSK) constellations. The SISO KD is a tree-search algorithm, which reduces the complexity through *dominance conditions*. The latter exploit the properties of the channel matrix and the a-priori information on the trasmitted bits. The basic idea is to strongly reduce the set of candidates to the computation of the log-likelihood ratios (LLRs) via a pruning mechanism based on the definition of local minimum of the metric, which arises from the Max-Log approximation of the LLR. The advantages of the SISO KD are: (i) *does not need any decomposition or inversion of the channel matrix* and (ii) *can prune already at the first levels of the tree in the case of underdetermined systems, without any change to the formulation presented in this work*, as opposed to most algorithms based on the SD. As stated in [36], the dominance conditions for the hard case have already been derived in [38], where they are used in a multi-user detection algorithm based on the Hopfield neural networks. The dominance conditions have also been employed in [39] for maximum-likelihood sequence detection and in [40] for a multistage detection algorithm. Furthermore, in [41] the authors show how the prior information can be incorporated in the dominance conditions. The proposed algorithm makes a different use of the dominance conditions, i.e., as pruning criteria in the tree-search mechanism. It does not compute the exact Max-Log solution of the SISO detection problem, but simulation results show that it achieves performance very close to the Max-Log detector with significantly reduced computational complexity. This result allows us to compare it with the SISO single-tree search SD (STSSD) [1], which represents an efficient (in terms of complexity) exact and approximated implementation of the Max-Log solution. Here, w.r.t. our earlier conference paper [35], the algorithm is extended to the case of generic M-QAM constellation, a computationally efficient way to compute the required metrics is presented and a detailed analysis of the computational complexity (including a worst-case numerical analysis) of KD is performed.

Outline: The remainder of this paper is organized as follows. In Section II the complex-valued and the equivalent real-valued models of a narrowband MIMO system are described. In Section III we briefly recall the SISO detection problem and the approximated formulation via Max-Log. In Section IV we present the dominance conditions and derive the SISO KD. In Section V the complexity analysis of the proposed algorithm is provided. In Section VI we demonstrate via simulation results that the SISO KD is able to achieve near-optimal performance with reduced computational complexity w.r.t. the alternatives proposed in the literature. Finally, in Section VII we draw some concluding remarks; proofs and derivations are confined to the Appendix.

Notation: Lower-case bold letters denote column vectors, with a_i denoting the ith element of the vector $\boldsymbol{a}$; upper-case bold letters denote matrices, with A_{ij} and $\boldsymbol{a}_i$ denoting the (i, j)th element and the ith column of the matrix $\boldsymbol{A}$, respectively; $\boldsymbol{I}_n$ is the $n \times n$ identity matrix; $\boldsymbol{0}_{n\times m}$ is a $n \times m$ matrix of zeros; $\text{diag}(\boldsymbol{A})$ is a diagonal matrix with the same size and the same diagonal as the square matrix $\boldsymbol{A}$; $|\cdot|$, $\|\cdot\|$, $\Pr\{\cdot\}$, $(\cdot)^T$, $\Re(\cdot)$, and $\Im(\cdot)$ denote the absolute value, the L_2 norm operator, the probability of events, the transpose operator and the real and imaginary part, respectively; $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ $(\mathcal{CN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}))$ is the (circularly complex) normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$; finally, $\sim$ means "distributed as."

II. System Model

We consider a narrowband MIMO system with $\tilde{K}$ transmit antennas and $\tilde{N}$ receive antennas. The corresponding discrete-time model is

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{s}} + \tilde{\boldsymbol{n}}, \tag{1}$$

where $\tilde{\boldsymbol{s}} \in \mathbb{C}^{\tilde{K}\times 1}$ is the transmitted symbols vector, $\tilde{\boldsymbol{H}} \in \mathbb{C}^{\tilde{N}\times\tilde{K}}$ is the fading channel matrix, $\tilde{\boldsymbol{n}} \in \mathbb{C}^{\tilde{N}\times 1}$, with $\tilde{\boldsymbol{n}} \sim \mathcal{CN}(\boldsymbol{0}_{\tilde{N}\times 1}, \sigma^2\boldsymbol{I}_{\tilde{N}})$, is the noise vector and $\tilde{\boldsymbol{y}} \in \mathbb{C}^{\tilde{N}\times 1}$ is the received vector. We assume perfect channel state information at the receiver, i.e., $\tilde{\boldsymbol{H}}$ is known.

The system employs bit-interleaved coded modulation [42]. The block diagram of the transmitter is shown in Fig. 1(a). The

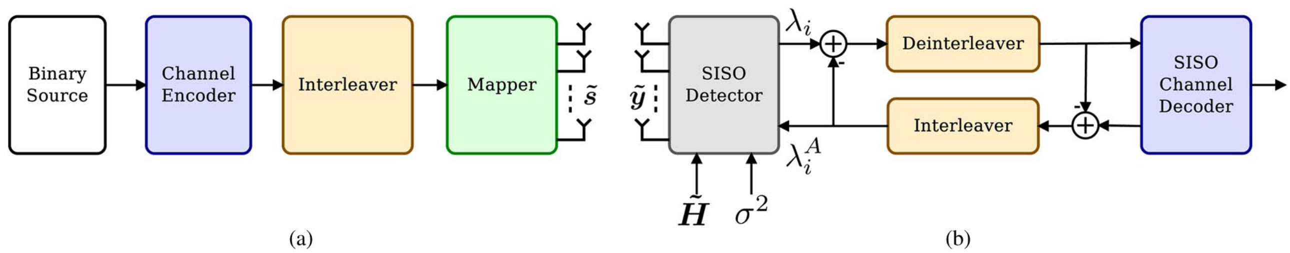


Fig. 1. MIMO system: (a) block diagram of the transmitter, (b) block diagram of the IDDS.

binary stream is coded and interleaved to enforce independence of information among the transmitted bits. The bits are then mapped to symbols using a generic M-QAM constellation, including binary phase-shift keying (BPSK) and QPSK as special cases. The block diagram of the typical iterative detection-and-decoding system (IDDS) [3] is shown in Fig. 1(b). The receiver processes a block of $\frac{w}{\tilde{K}b}$ received vectors $\tilde{\boldsymbol{y}}$ in each snapshot,[1] where w is the codeword length and $b \triangleq \log_2 M$. The SISO detector relies on the perfect channel state information, namely $\tilde{\boldsymbol{H}}$ and σ^2. During each iteration the SISO detector processes these vectors and the a-priori soft informations and outputs a w-dimensional vector of a-posteriori soft informations, one for each coded bit, to which the a-priori soft informations are subtracted to obtain the extrinsic soft informations. These ones are deinterleaved and decoded. The SISO channel decoder based on a-posteriori probability (APP) has been formalized in [43]. The channel decoder provides two outputs: (i) the update of soft informations on uncoded bits, which can be used to make the decisions for the current iteration; (ii) the update of soft informations on coded bits, which are converted to extrinsic soft informations, re-interleaved and used as *a-priori* soft informations for the SISO detector in the next iteration. At the first iteration the a-priori soft informations are set to zero. The iterative process stops after a preset number of iterations and the receiver moves to the next snapshot.

A. Equivalent Real-Valued Model

For our purposes, it is convenient to refer to an equivalent real-valued model with transmitted symbols drawn from a BPSK constellation. This can be obtained in two steps. Since the considered constellations are separable, we can equivalently rewrite the system model in (1) as

$$\boldsymbol{y} = \bar{\boldsymbol{H}}\boldsymbol{s} + \boldsymbol{n}, \tag{2}$$

where $\boldsymbol{y} \triangleq \begin{pmatrix}\Re(\tilde{\boldsymbol{y}})\\ \Im(\tilde{\boldsymbol{y}})\end{pmatrix} \in \mathbb{R}^{N\times 1}$, $\boldsymbol{s} \triangleq \begin{pmatrix}\Re(\tilde{\boldsymbol{s}})\\ \Im(\tilde{\boldsymbol{s}})\end{pmatrix} \in \mathbb{R}^{2\tilde{K}\times 1}$, $\boldsymbol{n} \triangleq \begin{pmatrix}\Re(\tilde{\boldsymbol{n}})\\ \Im(\tilde{\boldsymbol{n}})\end{pmatrix} \in \mathbb{R}^{N\times 1}$, $N \triangleq 2\tilde{N}$ and

$$\bar{\boldsymbol{H}} \triangleq \begin{pmatrix}\tilde{\boldsymbol{H}}_R & -\tilde{\boldsymbol{H}}_I\\ \tilde{\boldsymbol{H}}_I & \tilde{\boldsymbol{H}}_R\end{pmatrix}, \tag{3}$$

where $\tilde{\boldsymbol{H}}_R \triangleq \Re(\tilde{\boldsymbol{H}})$, $\tilde{\boldsymbol{H}}_I \triangleq \Im(\tilde{\boldsymbol{H}})$; also it can be readily verified that $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}_{N\times 1}, \frac{\sigma^2}{2}\boldsymbol{I}_N)$. Secondly, we can express each real-valued constellation in terms of a vector of b antipodal signals, that is $\boldsymbol{s} = \boldsymbol{P}x$, where $\boldsymbol{P} \in \mathbb{Z}^{2\tilde{K}\times K}$ and $\boldsymbol{x} \in \{-1, +1\}^K$, $K \triangleq \tilde{K}b$. The matrix $\boldsymbol{P}$ determines the mapping from binary to real-valued symbols and its expression can be derived from the formulation in [23], [44]. Substituting such expression in (2) and defining $\boldsymbol{H} \triangleq \bar{\boldsymbol{H}}\boldsymbol{P}$ leads to

$$\boldsymbol{y} = \boldsymbol{H}x + \boldsymbol{n}. \tag{4}$$

It needs to note that the matrix $\boldsymbol{P}$ prevents the use of Gray mapping with higher order constellations $(M > 4)$, because of the presence of the a-priori information in the SISO detector formulation (see Section III).

[1]A snapshot consists of the transmission of a single codeword.

III. SISO Detection

The SISO detector computes the *a-posteriori* LLRs $\forall\, i = 1, \ldots, K$ [3]

$$\begin{aligned}\lambda_i &\triangleq \ln \frac{\Pr\{x_i = +1|\boldsymbol{y}\}}{\Pr\{x_i = -1|\boldsymbol{y}\}}\\ &= \ln \frac{\sum\limits_{\boldsymbol{x}\in\mathcal{X}:x_i=+1} \exp\left(-\frac{\|\boldsymbol{y}-\boldsymbol{H}x\|^2}{\sigma^2} + \frac{1}{2}\sum\limits_{j=1}^{K} x_j\lambda_j^A\right)}{\sum\limits_{\boldsymbol{x}\in\mathcal{X}:x_i=-1} \exp\left(-\frac{\|\boldsymbol{y}-\boldsymbol{H}x\|^2}{\sigma^2} + \frac{1}{2}\sum\limits_{j=1}^{K} x_j\lambda_j^A\right)},\end{aligned} \tag{5}$$

where $\mathcal{X} \triangleq \{-1, +1\}^K$ and $\lambda_j^A \triangleq \ln \frac{\Pr\{x_j=+1\}}{\Pr\{x_j=-1\}}, j = 1, \ldots, K$, are the *a-priori* LLRs. From inspection of (5), it is apparent that the computation of λ_i, $\forall\, i = 1, \ldots, K$, requires the following 2^K terms

$$\exp\left(-\frac{\|\boldsymbol{y}-\boldsymbol{H}x\|^2}{\sigma^2} + \frac{1}{2}\sum_{j=1}^{K} x_j\lambda_j^A\right), \qquad \forall\, \boldsymbol{x} \in \mathcal{X}; \tag{6}$$

therefore it is exponentially complex with K, thus leading to prohibitive complexity as the number of bits per transmitting vector grows. By applying the Max-Log approximation [3]

$$\ln \sum_i \exp(a_i) \approx \max_i\ a_i, \tag{7}$$

the ith a-posteriori LLR can be approximated as

$$\lambda_i^M \triangleq \min_{\boldsymbol{x}\in\mathcal{X}:x_i=-1} m(\boldsymbol{x}) - \min_{\boldsymbol{x}\in\mathcal{X}:x_i=+1} m(\boldsymbol{x}), \tag{8}$$

where

$$m(\boldsymbol{x}) \triangleq \frac{\|\boldsymbol{y}-\boldsymbol{H}x\|^2}{\sigma^2} - \frac{1}{2}\sum_{j=1}^{K} x_j\lambda_j^A. \tag{9}$$

The loss in performance of (8) w.r.t. the exact formulation in (5) has been shown to be negligible[2] [45].

Equation (8) can also be conveniently expressed as [32], [41]

$$\lambda_i^M = \begin{cases} m\left(\boldsymbol{x}^{(i)}\right) - m(\boldsymbol{x}^{MAP}), & \boldsymbol{x}_i^{MAP} = +1 \\ m(\boldsymbol{x}^{MAP}) - m\left(\boldsymbol{x}^{(i)}\right), & \boldsymbol{x}_i^{MAP} = -1, \end{cases} \tag{10}$$

where

$$\boldsymbol{x}^{MAP} \triangleq \arg\max_{\boldsymbol{x}\in\mathcal{X}} \Pr\{\boldsymbol{x}|\boldsymbol{y}\} \tag{11}$$

is the maximum *a-posteriori* (MAP) solution[3] and

$$\boldsymbol{x}^{(i)} \triangleq \arg\min_{\boldsymbol{x}\in\mathcal{X}:x_i=-x_i^{MAP}} m(\boldsymbol{x}) \tag{12}$$

is the ith *counter-hypothesis* [1].

Remarks: The Max-Log approximation itself does not lead to a complexity reduction, because (8) still requires the computation of 2^K metrics; indeed, it was originally introduced to remedy to the numerical instability, which affects (5) [45]. Nonetheless (8) can be exploited to design low-complexity algorithms [1], [32], [41].

IV. SISO King Decoder

In this section we describe the main contribution of this paper, the SISO KD algorithm which is derived in terms of a tree-search where the cost function at each node is based on the generalization of the dominance condition (see [38], [39], [41]). Furthermore we provide a computationally efficient method to compute the metric defined in (9).

A. Dominance Conditions

We reformulate the dominance conditions in the general case of SISO detection by taking into account the *a-priori* information. Let $\alpha_i(\boldsymbol{x})$ be a vector which differs from $\boldsymbol{x}$ only in the ith element, i.e., $\alpha_i(\boldsymbol{x}) \triangleq [x_1 \dots -x_i \dots x_K]^T$. Such a vector will be referred to as the ith adjacent of $\boldsymbol{x}$. A vector $\boldsymbol{x}$ that satisfies the set of conditions

$$m(\boldsymbol{x}) < m(\alpha_i(\boldsymbol{x})), \qquad \forall\, i = 1,\dots,K, \tag{13}$$

is a local minimum of $m(\boldsymbol{x})$. The ith condition in (13) is equivalently written as (see the Appendix)

$$x_i \left[\frac{4}{\sigma^2}\left(\boldsymbol{h}_i^T\boldsymbol{y} - \sum_{\substack{j=1\\j\neq i}}^{K} G_{ij}x_j\right) + \lambda_i^A\right] > 0, \tag{14}$$

where $\boldsymbol{G} \triangleq \boldsymbol{H}^T\boldsymbol{H}$. Eq. (14) does not allow an efficient search of the local minima of $m(\boldsymbol{x})$, because it depends on all the elements of $\boldsymbol{x}$. We observe that verifing (14) is a matter of determining the sign of the left-hand side (l.h.s.) or, equivalently, the sign of the expression in the brackets

$$\frac{4}{\sigma^2}\left(\boldsymbol{h}_i^T\boldsymbol{y} - \sum_{\substack{j=1\\j\neq i}}^{K} G_{ij}x_j\right) + \lambda_i^A, \tag{15}$$

which on occasion can be determined even without knowledge on $\boldsymbol{x}$. We consider the condition

$$\left|\frac{4}{\sigma^2}\boldsymbol{h}_i^T\boldsymbol{y} + \lambda_i^A\right| > \frac{4}{\sigma^2}\sum_{\substack{j=1\\j\neq i}}^{K} |G_{ij}|, \tag{16}$$

which does not depend on $\boldsymbol{x}$. Noticing that the right-hand side (r.h.s.) of (16) is the maximum possible value of $-\frac{4}{\sigma^2}\sum_{j=1}^{K} G_{ij}x_j$ as $\boldsymbol{x}$ changes, it is apparent that, if (16) is satisfied, (15) has the same sign as the argument of the absolute value in the l.h.s. of (16), regardless of the elements in $\boldsymbol{x}$. Therefore, if (16) is satisfied, we can conclude that all the vectors $\boldsymbol{x}$ such that

$$x_i \neq \text{sign}\left(\frac{4}{\sigma^2}\boldsymbol{h}_i^T\boldsymbol{y} + \lambda_i^A\right) \tag{17}$$

cannot be local minima, because they cannot satisfy (14), namely the ith condition in (13), and can be excluded from the search.

Furthermore, if a partial vector $\rho_{i-1}(\boldsymbol{x}) \triangleq [x_1 \dots x_{i-1}]^T$ is available, a condition more general than (16) can be derived. Noticing that (15) can be rewritten as

$$\frac{4}{\sigma^2}\left(\boldsymbol{h}_i^T\boldsymbol{y} - \sum_{j=1}^{i-1} G_{ij}x_j\right) + \lambda_i^A - \frac{4}{\sigma^2}\sum_{j=i+1}^{K} G_{ij}x_j, \tag{18}$$

we consider the less restrictive condition

$$\left|\frac{4}{\sigma^2}\left(\boldsymbol{h}_i^T\boldsymbol{y} - \sum_{j=1}^{i-1} G_{ij}x_j\right) + \lambda_i^A\right| > \frac{4}{\sigma^2}\sum_{j=i+1}^{K} |G_{ij}|, \tag{19}$$

which depends only on $\rho_{i-1}(\boldsymbol{x})$. Since the r.h.s. of (19) is the maximum possible value of $-\frac{4}{\sigma^2}\sum_{j=i+1}^{K} G_{ij}x_j$ as $[x_{i+1} \dots x_K]$ changes, one can notice that, if (19) is satisfied, (15) has the same sign as the argument of the absolute value in the l.h.s. of (19)

$$q_i\left(\rho_{i-1}(\boldsymbol{x})\right) \triangleq \frac{4}{\sigma^2}\left(\boldsymbol{h}_i^T\boldsymbol{y} - \sum_{j=1}^{i-1} G_{ij}x_j\right) + \lambda_i^A, \tag{20}$$

regardless of the remaining elements $[x_{i+1} \dots x_K]$. Therefore, if (19) is satisfied for a given $\rho_{i-1}(\boldsymbol{x})$, we can conclude that all the vectors $\boldsymbol{x}$ such that

$$x_i \neq \text{sign}\left(q_i\left(\rho_{i-1}(\boldsymbol{x})\right)\right) \tag{21}$$

cannot be local minima, because they cannot satisfy the ith condition in (13), and can be excluded from the search. Equation (19) reduces to (16) for $i = 1$ and can be thought as a generalization of (16) conditioned to the knowledge of $\rho_{i-1}(\boldsymbol{x})$.

[2] This is not always true for underdetermined scenarios, as it will be shown in Section VI.

[3] In fact, it can be readily verified that $\boldsymbol{x}^{MAP} = \arg\max_{\boldsymbol{x}\in\mathcal{X}} m(\boldsymbol{x})$.

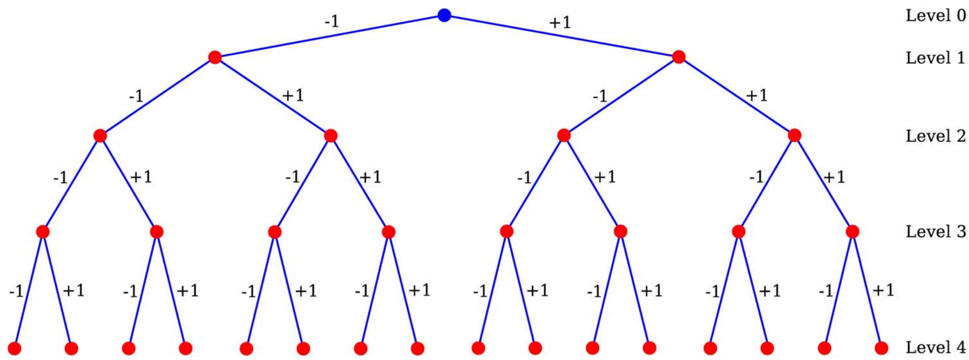


Fig. 2. Binary tree representation of MIMO detection problem for the case $K = 4$.

For a given i, (19) is named *dominance condition* [37], since it assesses whether if the projection of the $\boldsymbol{y}$ on the ith column of $\boldsymbol{H}$ (once suppressed the partial interference given by $\rho_{i-1}(\boldsymbol{x})$) and the *a-priori* information are strong enough to dominate over the worst-case residual interference, which is given by the r.h.s. When (19) is satisfied, such a condition allows to decide on x_i, without making any assumption on $[x_{i+1} \ldots x_K]$. It is worth noticing that, when $i = K$, (19) is always satisfied, because the r.h.s. is null. Nevertheless (20) needs to be computed, to decide on x_K.

B. Algorithm Formulation

The computation of the K LLRs in (8) requires the metric of the MAP solution and of the K counter-hypotheses, which can be computed through multiple searches [32]. Differently, in this paper, we propose a single tree-search where we apply the dominance conditions as pruning criterion at each node. The result is a reduced set of possible solutions, that always contains the MAP solution, but not the counter-hypotheses. These are extracted from the reduced set of candidates without the need of repeated searches.

1) Local Minima Tree-Search: The local minima search can be set up as a search on a binary tree with $K+1$ levels numbered from 0 (the root level) to K (the leaves level). Fig. 2 shows the binary tree when $K = 4$, which arises, for instance, from a system with $\tilde{K} = 2$ employing QPSK modulation or $\tilde{K} = 4$ employing BPSK modulation. Each of the 2^K leaves is associated with one possible vector $\boldsymbol{x}$ and each non-leaf node corresponds to a partial vector according to its level and the labels of the branches in the path from the root to the node itself.[4] The proposed local minima search belongs to the general class of *branch-and-bound* algorithms [46] and is formalized in Table I with a pseudo-code. It uses the dominance condition as a criterion to prune portions of the tree: when the current node n satisfies (19), one of its children is pruned according to (21) and, consequently, all the vectors associated with the leaves belonging to the corresponding sub-tree will not be included in the reduced set. The algorithm uses several sub-procedures, which have the following (intuitive) meaning:

- `next_node`(T): return the next node of the tree T to be visited or return NULL if there are no more nodes;
- `set_visited`(T, n): mark the node n as visited in the tree T;
- `vector`(n): return the partial vector associated to the node n;
- `level`(n): return the level number of the node n;
- `compute_q`(i, v): compute $q_i(v)$ according to (20);
- `bounding`(i): return the r.h.s. of the ith dominance condition;
- `prune_left`(T, n): prune the left child (and the related sub-tree) of the node n from the tree T;
- `prune_right`(T, n): prune the right child (and the related sub-tree) of the node n from the tree T.

TABLE I
LOCAL MINIMA TREE-SEARCH

```
Create the binary tree T with K + 1 levels and mark all nodes as not visited
n ← next_node(T)
while n ≠ NULL do
    set_visited(T, n)
    l ← level(n)
    if l < K then {n is not a leaf}
        ρ ← vector(n)
        q ← compute_q(l + 1, ρ)
        b ← bounding(l + 1)
        if |q| > b then {dominance condition is satisfied}
            if q > 0 then
                prune_left(T, n)
            else
                prune_right(T, n)
            end if
        end if
    end if
    n ← next_node(T)
end while
```

The procedure `next_node`(T) can be designed to select the next node according to different strategies. The top-down[5] approach is the most convenient, because it allows to prune the leaves that cannot be local minima as soon as possible. Employing a different strategy can lead to visit a node that will be anyway pruned during the visit to one of its ancestors, thus wasting computational resources. The two well-known top-down visit fashions are *depth-first* and *breadth-first* [46], which represent the two most common visiting strategies for tree-search algorithms. The SISO KD can employ both of them (and any other top-down visiting strategy) without affecting

[4] Without loss of generality, we assume that the left branch departing from a non-leaf node is labeled with -1 and the right one with $+1$.

[5] The tree is visited starting from the root node towards the leaves; a node can be visited only if all its ancestors have been visited.

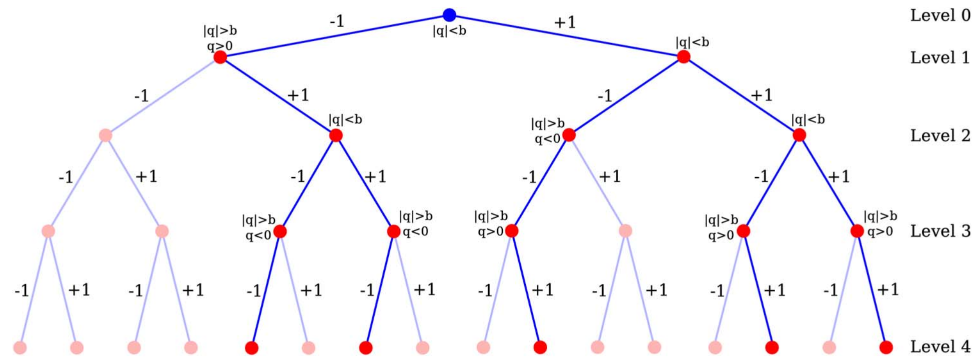


Fig. 3. An example of binary tree pruned by the local minima tree-search algorithm in Table I for the case $K = 4$.

its performance and computational complexity, because the pruning criterion depends only on the partial vector associated to the node under analysis and not on the previous history of the execution. This is not true for some of the algorithms based on the SD. The STSSD is a depth-first-only tree-search and starts with an infinite initial pruning radius, thus, its pruning capabilities are strongly conditioned by the first visited path from the root to the first visited leaf and also by the successive visiting order. The LSD is a breadth-first tree-search and its performance and pruning capabilities are strongly affected by the choice of the pruning radius.[6]

An example of the final state of the binary tree for the case $K = 4$ after the execution of the pruning process is shown in Fig. 3, where shaded branches and nodes have been cut.

2) LLR Approximation: The procedure terminates with a set of vectors associated to the survived leaves, denoted with $\mathcal{S}$, which contains the local minima and possibly other vectors not excluded by the dominance conditions (cf. (19)). The latter is analogous to the *candidate list* of the LSD in [3], although the SISO KD uses a different pruning criterion. While $\boldsymbol{x}^{MAP} \in \mathcal{S}$, since a global minimum is also a local minimum, there is no guarantee that $\boldsymbol{x}^{(i)} \in \mathcal{S}$, $\forall\, i = 1, \ldots, K$, because $\boldsymbol{x}^{(i)}$ is not necessarily a local minimum. The exact $\boldsymbol{x}^{(i)}$ could be found executing another tree-search, not checking the dominance conditions on the nodes until the level $i - 1$ and pruning in advance all the branches at level i (and the corresponding sub-trees) labeled with $\boldsymbol{x}_i^{MAP}$. This approach would need additional K searches on the tree. With the aim of reducing the complexity, we avoid these multiple searches by approximating the LLRs using only the vectors found in $\mathcal{S}$. Thus, we define the set

$$\mathcal{X}^{(i)} \triangleq \left\{\boldsymbol{x} \in \mathcal{S} : x_i = -x_i^{MAP}\right\} \cup \left\{\alpha_i(\boldsymbol{x}^{MAP})\right\}. \quad (22)$$

and

$$\boldsymbol{x}^{(i)} \triangleq \underset{\boldsymbol{x} \in \mathcal{X}(i)}{\arg\min}\, m(\boldsymbol{x}). \quad (23)$$

The LLRs are then approximated as

$$\lambda_i^K \triangleq \begin{cases} m\left(\boldsymbol{x}^{(i)}\right) - m(\boldsymbol{x}^{MAP}), & x_i^{MAP} = +1 \\ m(\boldsymbol{x}^{MAP}) - m\left(\boldsymbol{x}^{(i)}\right), & x_i^{MAP} = -1. \end{cases} \quad (24)$$

The reason for adding $\alpha_i(\boldsymbol{x}^{MAP})$ in (22) is because when $\boldsymbol{x}^{(i)}$ is adjacent to $\boldsymbol{x}^{MAP}$, it may be pruned during the tree-search, since it cannot be a local minimum by definition. Thus, taking into account in every case $\alpha_i(\boldsymbol{x}^{MAP}), \forall i = 1, \ldots, K$, represents another chance to find $\boldsymbol{x}^{(i)}$. Furthermore, this choice always allows LLR (approximate) evaluation, even in the case $\{\boldsymbol{x} \in \mathcal{S} : x_i = -\boldsymbol{x}_i^{MAP}\} = \emptyset$, since the pruning mechanism does not ensure that the final set $\mathcal{S}$ contains at least a vector such that $x_i = -\boldsymbol{x}_i^{MAP}$, $\forall\, i = 1, \ldots, K$. This enrichment of the final set $\mathcal{S}$ with the adjacents to the MAP solution is often named *bit-flipping* and appears in different versions in prior works, such as [32]. We will show in Section VI that the choice in (22) leads to near-optimal performances.

C. Cumulative Metric

Even though the reduction of possible candidates reduces the computational complexity w.r.t. the exhaustive search, further savings can be obtained by efficiently computing the metric $m(\boldsymbol{x})$. Equation (9) can be written as cumulative sum of partial contributions, each one depending on a portion of the vector $\boldsymbol{x}$. In addition, these contributions re-use portions of the already computed dominance conditions, leading to a further complexity reduction.

We start by showing that (9) can be expanded as

$$m(\boldsymbol{x}) = \frac{\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{y} + \boldsymbol{x}^T\boldsymbol{G}x}{\sigma^2} - \frac{1}{2}\sum_{j=1}^{K} x_j\lambda_j^A. \quad (25)$$

Since (24) is a difference between two metrics, all terms not depending on $\boldsymbol{x}$ vanish. In view of the aforementioned consideration, we define the equivalent metric

$$\bar{m}(\boldsymbol{x}) \triangleq \frac{\boldsymbol{x}^T\boldsymbol{G}x - \boldsymbol{x}^T\boldsymbol{E}x - 2\boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{y}}{\sigma^2} - \frac{1}{2}\sum_{j=1}^{K} x_j\lambda_j^A, \quad (26)$$

where $\boldsymbol{E} \triangleq \operatorname{diag}(\boldsymbol{G})$ and $\boldsymbol{x}^T\boldsymbol{E}x$ does not depend on $\boldsymbol{x}$, because $x_i^2 = 1, \forall\, i = 1, \ldots, K$. Since $\boldsymbol{G} - \boldsymbol{E}$ is symmetric with zero diagonal, denoting with $\boldsymbol{G}_L$ and $\boldsymbol{G}_U$ its lower and upper triangular part, respectively, we have

$$\begin{aligned} \boldsymbol{x}^T(\boldsymbol{G} - \boldsymbol{E})\boldsymbol{x} &= \boldsymbol{x}^T(\boldsymbol{G}_L + \boldsymbol{G}_U)\boldsymbol{x} = \\ &= \boldsymbol{x}^T\left(\boldsymbol{G}_L + \boldsymbol{G}_L^T\right)\boldsymbol{x} = 2\boldsymbol{x}^T\boldsymbol{G}_L\boldsymbol{x} \end{aligned} \quad (27)$$

[6]It can be implemented in depth-first mode with complexity depending also on the visiting order.

and

$$\bar{m}(\boldsymbol{x}) = \frac{2\boldsymbol{x}^T\left(\boldsymbol{G}_L\boldsymbol{x} - \boldsymbol{H}^T\boldsymbol{y}\right)}{\sigma^2} - \frac{1}{2}\sum_{j=1}^{K} x_j\lambda_j^A, \tag{28}$$

that is rewritten as

$$\bar{m}(\boldsymbol{x}) = \sum_{i=1}^{K} x_i\left[\frac{2}{\sigma^2}\left(\sum_{j=1}^{i-1} G_{ij}x_j - \boldsymbol{h}_i^T\boldsymbol{y}\right) - \frac{\lambda_i^A}{2}\right]. \tag{29}$$

Finally, the equivalent metric can be computed as

$$\bar{m}(\boldsymbol{x}) = \sum_{i=1}^{K} \bar{m}_i\left(\rho_i(\boldsymbol{x})\right), \tag{30}$$

where

$$\bar{m}_i\left(\rho_i(\boldsymbol{x})\right) \triangleq x_i\left[\frac{2}{\sigma^2}\left(\sum_{j=1}^{i-1} G_{ij}x_j - \boldsymbol{h}_i^T\boldsymbol{y}\right) - \frac{\lambda_i^A}{2}\right], \tag{31}$$

$\forall\, i = 1,\ldots,K$, are the metric increments and, since $\bar{m}_i(\rho_i(\boldsymbol{x}))$ depends only on x_j, $j = 1,\ldots,i$, each of the K increments can be computed when visiting the corresponding node at level i during the tree-search.

It is worth noticing that the formulation in (30) is equivalent in the MIMO context to the Ungerboeck recursion used in the equalization of inter-symbol-interference channels [47], [48].

Furthermore, it can be noticed that

$$\bar{m}_i\left(\rho_i(\boldsymbol{x})\right) = -\frac{x_i}{2}q_i\left(\rho_{i-1}(\boldsymbol{x})\right). \tag{32}$$

Hence a further complexity saving is achieved: at a given level i of the tree, since $q_i(\rho_{i-1}(\boldsymbol{x}))$ is already available from the computation of the dominance condition, the corresponding increment to the metric $\bar{m}_i(\rho_i(\boldsymbol{x}))$ is computed simply multiplying it by $-\frac{x_i}{2}$.

The algorithm in Table I is modified as in Table II to include the metrics computation. Four procedures are added:

- `get_metric`(n): get the metric of the node n;
- `set_metric`(n, m): set the metric m for the node n;
- `right_child`(n): return the right child of the node n;
- `left_child`(n): return the left child of the node n;

V. Computational Complexity

A detailed complexity analysis of the proposed algorithm is reported in this section. A common complexity metric for the broad class of tree-search algorithms is the average number of visited nodes [1]. However, two algorithms which use different trees,[7] cannot be compared on the basis of this metric, because of the different number of nodes.

On the basis of the aforementioned reason, we choose the number of elementary operations performed by the algorithm as an adequate metric for complexity analysis. Also, we prefer to count separately sums and products, because each of them affects the execution time of the algorithm in a different way (in fact, a product is usually more complex than a sum).

[7]In our case, a binary tree for the SISO KD and a M-ary tree for the STSSD.

TABLE II
Local Minima Tree-Search With Metrics Computation

```
Create the binary tree T with K + 1 levels and mark all nodes as not visited
n ← next_node(T)
while n ≠ NULL do
    set_visited(T, n)
    l ← level(n)
    if l < K then {n is not a leaf}
        ρ ← vector(n)
        q ← compute_q(l + 1, ρ)
        b ← bounding(l + 1)
        m ← get_metric(n)
        if |q| > b then {dominance condition is satisfied}
            if q > 0 then
                prune_left(T, n)
                set_metric(right_child(n), m − q/2)
            else
                prune_right(T, n)
                set_metric(left_child(n), m + q/2)
            end if
        else {no pruning}
            set_metric(right_child(n), m − q/2)
            set_metric(left_child(n), m + q/2)
        end if
    end if
    n ← next_node(T)
end while
```

A. Pre-Processing

The main contribution to the complexity comes from dominance conditions and metrics computation. By inspection of (19), we notice that only the term at the l.h.s.

$$\frac{4}{\sigma^2}\sum_{j=1}^{i-1} G_{ij}x_j \tag{33}$$

needs to be computed at each visited node. The remaining terms in (19) depend only on $\boldsymbol{H}$, $\boldsymbol{y}$ and $\boldsymbol{\lambda}^A$ (the vector of the *a-priori* LLRs) and not on $\rho_{i-1}(\boldsymbol{x})$. Thus they can be computed together with $\boldsymbol{G}$ before the tree-search is started. The evaluation of the aforementioned terms forms the pre-processing step of the algorithm, whose complexity is fixed by $\tilde{K}$, $\tilde{N}$, and b and is summarized in the first five rows of Table III. Even though $\frac{4}{\sigma^2}\boldsymbol{G}$ could be computed directly, we prefer to go through several steps, which exploit the structure of $\bar{\boldsymbol{H}}$ and $\boldsymbol{P}$ and produce intermediate results useful in the following.[8] We start by observing that

$$\bar{\boldsymbol{G}} \triangleq \bar{\boldsymbol{H}}^T\bar{\boldsymbol{H}} = \begin{pmatrix} \tilde{\boldsymbol{H}}_R^T\tilde{\boldsymbol{H}}_R + \tilde{\boldsymbol{H}}_I^T\tilde{\boldsymbol{H}}_I & \tilde{\boldsymbol{H}}_I^T\tilde{\boldsymbol{H}}_R - \tilde{\boldsymbol{H}}_R^T\tilde{\boldsymbol{H}}_I \\ \tilde{\boldsymbol{H}}_R^T\tilde{\boldsymbol{H}}_I - \tilde{\boldsymbol{H}}_I^T\tilde{\boldsymbol{H}}_R & \tilde{\boldsymbol{H}}_R^T\tilde{\boldsymbol{H}}_R + \tilde{\boldsymbol{H}}_I^T\tilde{\boldsymbol{H}}_I \end{pmatrix} \tag{34}$$

has only $n_{\bar{\boldsymbol{G}}} \triangleq \tilde{K}(\tilde{K}+1)$ independent elements (since the other elements are given by symmetry). Hence the evaluation

[8]The operations counts described here fit the case of an even b and $b > 2$ (square M-QAM with $M > 4$); the cases of an odd b and $b = 2$ are straightforward and are omitted for the sake of brevity.

TABLE III
SISO KING DECODER OPERATIONS COUNTS

		Description	Sums	Products
Pre-processing	1	$\frac{4}{\sigma^2}\bar{\boldsymbol{G}}$	$(N-1)n_{\bar{\boldsymbol{G}}}$	$(N+1)n_{\bar{\boldsymbol{G}}}$
	2	$\frac{4}{\sigma^2}\hat{\boldsymbol{G}}$	0	$(\frac{b}{2}-1)n_{\bar{\boldsymbol{G}}}$
	3	$\frac{4}{\sigma^2}\boldsymbol{G}$	0	$(\frac{b}{2}-1)\frac{b}{2}n_{\bar{\boldsymbol{G}}}$
	4	$\frac{4}{\sigma^2}\boldsymbol{H}^T\boldsymbol{y}+\boldsymbol{\lambda}^A$	$2\tilde{K}(N-1)+K$	$2\tilde{K}(N+\frac{b}{2})$
	5	$\frac{4}{\sigma^2}\sum_{j=i+1}^{K}\lvert G_{ij}\rvert$	$\frac{1}{2}(K^2-3K+2)$	0
Tree-search	6	Pruning	$2\tilde{K}\sum_{i=2\tilde{K}+1}^{K-1}s_i+\sum_{i=1}^{2\tilde{K}}is_i$	$2\tilde{K}\sum_{i=2\tilde{K}+1}^{K-1}s_i+\sum_{i=1}^{2\tilde{K}}is_i$
	7	Metric	$\sum_{i=2}^{K}s_i$	0
Post-processing	8	MAP adjacents	$\mathcal{O}(b^2\tilde{K}^3)$	$\mathcal{O}(b^2\tilde{K}^3)$
	9	Extrinsic LLRs	$2K$	$2K$

of $\frac{4}{\sigma^2}\bar{\boldsymbol{G}}$ requires $(N-1)n_{\bar{\boldsymbol{G}}}$ sums and $(N+1)n_{\bar{\boldsymbol{G}}}$ products, as reported in the first row in Table III. Exploiting the particular structure of $\boldsymbol{P}$ (since it can be formulated as a row-concatenation of scaled identity matrices with the last block being $\boldsymbol{I}_{2\tilde{K}}$), $\frac{4}{\sigma^2}\hat{\boldsymbol{G}} \triangleq \frac{4}{\sigma^2}\boldsymbol{P}^T\bar{\boldsymbol{G}}$ requires $(\frac{b}{2}-1)n_{\bar{\boldsymbol{G}}}$ products (assuming $\frac{4}{\sigma^2}\bar{\boldsymbol{G}}$ has been already computed) and $\frac{4}{\sigma^2}\boldsymbol{G} = \frac{4}{\sigma^2}\hat{\boldsymbol{G}}P$ requires $(\frac{b}{2}-1)\frac{b}{2}n_{\bar{\boldsymbol{G}}}$ products (assuming $\frac{4}{\sigma^2}\hat{\boldsymbol{G}}$ has been already computed), as reported in the second and third row in Table III, respectively. The fourth row of Table III shows the complexity of $\frac{4}{\sigma^2}\bar{\boldsymbol{H}}^T\boldsymbol{y}+\boldsymbol{\lambda}^A$, which can be rewritten as $\frac{4}{\sigma^2}\boldsymbol{P}^T\bar{\boldsymbol{H}}^T\boldsymbol{y}+\boldsymbol{\lambda}^A$. Observing that $\frac{4}{\sigma^2}\boldsymbol{P}^T\bar{\boldsymbol{H}}^T\boldsymbol{y}$ is a scaled repetition of $\frac{4}{\sigma^2}\bar{\boldsymbol{H}}^T\boldsymbol{y}$, only $2\tilde{K}(N-1)+K$ sums and $2\tilde{K}(N+\frac{b}{2})$ products are required, rather than KN sums and $K(N+1)$ products.

The number of sums for the r.h.s. of the dominance conditions is the expansion of

$$\sum_{i=1}^{K-2}(K-i-1), \tag{35}$$

which comes from the observation that for the ith dominance condition it needs to execute $K-i-1$ sums, except for $i=K-1$, for which the r.h.s. is equal to $\frac{4}{\sigma^2}\lvert G_{K-1K}\rvert$, and $i=K$, for which the r.h.s. is zero. The expansion is shown in the fifth row of Table III.

B. Tree-Search

The sixth row of Table III refers to the tree-pruning process, whose complexity is not deterministic. Indeed, at level $i=1,\ldots,K$, we need to compute the l.h.s. of a dominance condition for each survived node[9] of the level $i-1$, namely compute (33), and subtract it from the remainder of the l.h.s., already computed in the pre-processing step. Also, (33) can be rewritten as

$$\check{\boldsymbol{g}}_i^T\sum_{j=1}^{i-1}\boldsymbol{p}_jx_j = \check{\boldsymbol{g}}_i^T(p_1\ \ldots\ \boldsymbol{p}_{i-1})\rho_{i-1}(\boldsymbol{x}), \tag{36}$$

where $\check{\boldsymbol{G}} \triangleq \hat{\boldsymbol{G}}^T$. Since the term $(\boldsymbol{p}_1\ \ldots\ \boldsymbol{p}_{i-1})\rho_{i-1}(\boldsymbol{x})$ does not depend on $\boldsymbol{H}$, $\boldsymbol{y}$ or $\boldsymbol{\lambda}^A$, but only on the particular node being visited, it can be precomputed and "stored" in the node.[10] In this way the computational cost of the l.h.s. of the dominance condition is fixed to $2\tilde{K}$ sums and $2\tilde{K}$ products, regardless of the level number of the node being visited. The direct calculation of (33) requires $i-1$ sums and $i-1$ products, thus the alternative formulation in (36) leads to a lower complexity $\forall\, i=2\tilde{K}+2,\ldots,K$. Furthermore, $\forall\, i=1,\ldots,2\tilde{K}+1$, it is possible to get the same complexity as the direct calculation, neglecting the last $2\tilde{K}-i+1$ rows of the matrix $(\boldsymbol{p}_1\ \ldots\ \boldsymbol{p}_{i-1})$, which are all zeros, and, consequently, the last $2\tilde{K}-i+1$ elements of $\check{\boldsymbol{g}}_i$. Finally, the computational cost of the dominance conditions for both sums and products is

$$\sum_{i=1}^{2\tilde{K}+1}(i-1)s_{i-1}+2\tilde{K}\sum_{i=2\tilde{K}+2}^{K}s_{i-1}$$

$$=\sum_{i=1}^{2\tilde{K}}is_i+2\tilde{K}\sum_{i=2\tilde{K}+1}^{K-1}s_i. \tag{37}$$

The seventh row of Table III reports the contribution of the computation of the metric increments. Exploiting the cumulative formulation (see Section IV-C) and postponing the division by 2 in (32) after the final LLRs computation in (24),[11] each metric increment requires only one sum, thus at level $i>1$ s_i sums are needed.

C. Post-Processing

The complexity of the last two steps is fixed by b and $\tilde{K}$. The cumulative formulation (see Section IV-C) is used also for computation of $m(\alpha_i(\boldsymbol{x}^{MAP}))$, $\forall\, i=1,\ldots,K$. By ordering these adjacents according to the index of the element different from the MAP solution, namely from $\alpha_1(\boldsymbol{x}^{MAP})$ to $\alpha_K(\boldsymbol{x}^{MAP})$, and processing the set formed by the ith element of every vector, we observe that $\forall\, i=1,\ldots,K$ only $i+1$ increments have to be computed, because the other $K-i-1$ are equal to the $(i+1)$th (they share the same partial vector). On the basis of the aforementioned reason and applying again

[9]The number of survived nodes at level i is denoted with s_i.

[10]It is worth noticing that, though the memory required for this "storing" grows exponentially with K, the complexity analysis is focused on execution time, thus memory is not considered a severe constraint.

[11]This change does not affect the result and leads to a further complexity saving, replacing the execution of one division for each metric increment with only K divisions by 2 on the results of (24).

the alternative calculation in (36), it can be shown that the computation of the metrics of the K adjacents to the MAP solution has a computational cost that grows with $\tilde{K}^3$ and b^2 for both sums and products,[12] as shown in the eighth row of Table III.

Finally, noticing that (24) can be rewritten as

$$\lambda_i^K = x_i^{MAP} \left(\bar{m}\left(\boldsymbol{x}^{(i)}\right) - \bar{m}(\boldsymbol{x}^{MAP}) \right), \qquad (38)$$

each LLR requires only a sum and a product.[13] This complexity is reported in the last row of Table III, where we also count K division by 2, as products, which compensate the divisions not performed for each metric increment as explained in Section V-B, and K additional sums that are needed to compute the extrinsic LLRs by subtracting λ_i^A from $\lambda_i^K, \forall\, i = 1, \ldots, K$.

VI. Simulation Results

System performance in terms of bit-error rate (BER) and average computational complexity, both vs signal-to-noise ratio (SNR) per receive antenna, have been obtained via Monte Carlo simulations. The SNR is defined as the ratio between the average energy per receive antenna and the noise variance σ^2. The simulated system employs a recursive convolutional encoder with rate $\frac{1}{2}$, constraint length 3, generators $(7,5)_8$ and feedback polynomial 7_8; the length of the codewords is 2000 bits. The permutation pattern of the interleaver has been chosen randomly for each snapshot. The channel coefficients of $\tilde{\boldsymbol{H}}$ have been generated according to $\mathcal{CN}(0,1)$ for each MIMO transmission. The simulations have been averaged over 10^5 snapshots. We have considered four system setups, which differ in the number of antennas and in the employed constellation: $\tilde{K} = \tilde{N} = 2$ with 16-QAM, $\tilde{K} = \tilde{N} = 4$ with 16-QAM, $\tilde{K} = \tilde{N} = 2$ with 64-QAM and $\tilde{K} = \tilde{N} = 4$ with 64-QAM. We have chosen all setups as *fully-loaded*, i.e., $\tilde{K} = \tilde{N}$, because our element of comparison, the STSSD, does not support the *underdetermined* scenarios, i.e., $\tilde{K} > \tilde{N}$. As already pointed out in Section I, the STSSD allows to tune the tradeoff between performance and complexity reduction by adjusting a parameter, which is the maximum permitted value of the normalized[14] LLRs, named $L_{\max}$; this technique is referred to as *clipping*. We study optimal SISO detection, the SISO KD, the STSSD without clipping ($L_{\max} = +\infty$, then attaining exact Max-Log performance [1]), the STSSD with $L_{\max} = 0.4$ and $L_{\max} = 0.1$, and the MMSE-PIC, as formalized in [15].

Fig. 4 shows the BER as a function of SNR for iterations 1, 2, and 6 (the successive iterations do not improve the performance significantly) for the case $\tilde{K} = \tilde{N} = 2$ with 16-QAM. For each iteration, simulations demonstrate that the STSSD without clipping and the SISO KD achieve roughly the same performance, which is very close to the optimum. The STSSD with $L_{\max} = 0.4$ exhibits a relatively small performance loss, that is less than 0.5 dB of SNR when BER $= 10^{-3}$ in the sixth iteration. Conversely, a lower value of $L_{\max}$ produces a considerable loss in performance, which increases with the iteration index. In the case $L_{\max} = 0.1$, the STSSD achieves BER equal to 10^{-3} with almost 1, 2 and more than 2 dB of SNR more w.r.t. the optimum in the first, the second and sixth iteration, respectively. The MMSE-PIC performance at BER $= 10^{-3}$ is similar to the STSSD with $L_{\max} = 0.1$ and is even worse for lower values of BER.

Figs. 5 and 6 report the average number of sums and products as a function of the SNR executed by the MMSE-PIC, the SISO KD and the STSSD (in the three cases $L_{\max} = 0.1, 0.4, +\infty$) in the first iteration and in the first six iterations of the IDDS, respectively. The MMSE-PIC's and STSSD's computational complexity has been evaluated by counting the operations

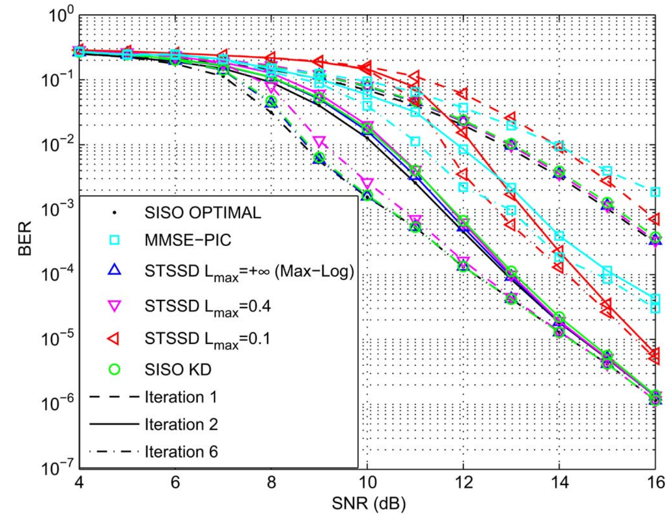


Fig. 4. BER vs SNR (dB) for an MIMO system with the IDDS, 16-QAM and $\tilde{K} = \tilde{N} = 2$. Comparison of the optimal SISO detector, MMSE-PIC, STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

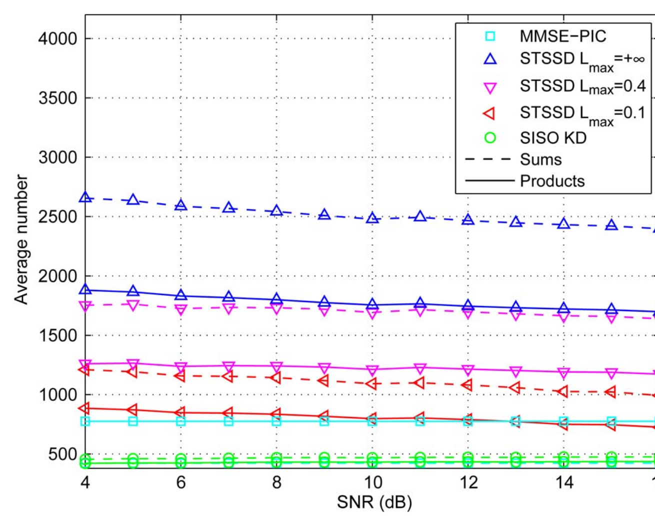


Fig. 5. Average number of elementary operations (sums and products) needed for algorithm implementation vs SNR (dB); first iteration of the IDDS for an MIMO system with 16-QAM and $\tilde{K} = \tilde{N} = 2$. Comparison of MMSE-PIC, STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

[12] The exact expressions are omitted for the sake of brevity.

[13] This product is actually a sign change, which is tipically much faster than a standard product. However, to keep simplicity in the complexity evaluation, we have considered the sign changes as they were standard products, which leads to a pessimistic analysis of the complexity of the proposed approach and one can expect to gain further complexity reduction in a practical implementation.

[14] In the context of the STSSD, the normalized LLR is the LLR divided by σ^2.

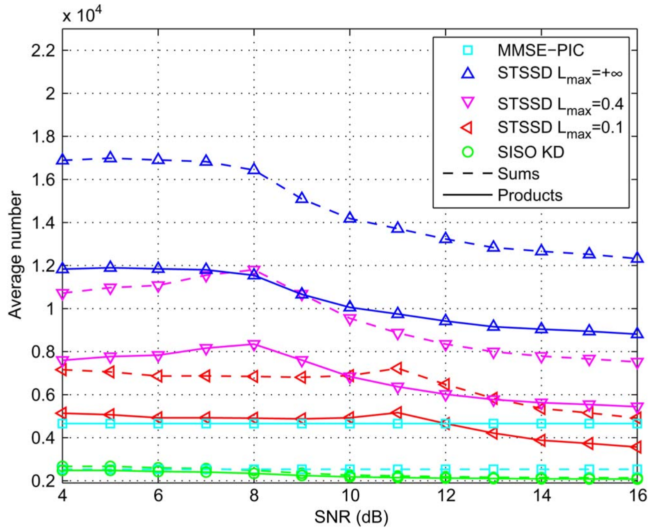


Fig. 6. Average number of elementary operations (sums and products) needed for algorithm implementation vs SNR (dB); accumulation of the first six iterations of the IDDS for an MIMO system with 16-QAM and $\tilde{K} = \tilde{N} = 2$. Comparison of MMSE-PIC, STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

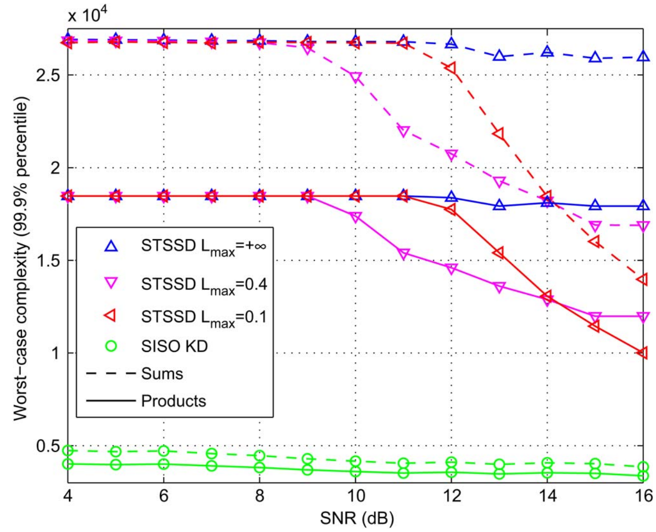


Fig. 7. Worst-case complexity as the 99.9% percentile of the number of elementary operations (sums and products) needed for algorithm implementation vs SNR (dB); accumulation of the first six iterations of the IDDS for an MIMO system with 16-QAM and $\tilde{K} = \tilde{N} = 2$. Comparison of STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

executed in the implementation made available by the authors of [1]. Since the STSSD uses the complex-valued system model in (1), it executes operations on complex-valued numbers. With the intent of a fair comparison, we count two real-valued sums for each complex-valued sum and two real-valued sums and four real-valued products for each complex-valued product. The complexity of the channel matrix decomposition has been included in the STSSD's operations counts. It is worth noticing that in a quasi-static scenario, namely when the channel matrix is slowly varying, the decomposition may be executed less frequently than detection, thus leading to a complexity saving, which, anyway, is small, because the number of required operations is negligible w.r.t. the overall STSSD complexity. Nonetheless, it can be noticed that a practical implementation of the STSSD requires, in any case, a module for the channel matrix decomposition, which contributes to the overall silicon area required. As expected, the STSSD reduces its complexity as the SNR grows. Furthermore, it brings down the complexity as $L_{\max}$ decreases. Nevertheless, it is manifest that the SISO KD outperforms the STSSD for both $L_{\max} = +\infty$ and $L_{\max} = 0.4$ and for both sums and products. The STSSD with $L_{\max} = 0.1$ is closer to the SISO KD, particularly for the average number of products, but it exhibits substantially worse performance, as already observed. The constant complexity of the MMSE-PIC is between the SISO KD and the STSSD with $L_{\max} = 0.1$ for the products and very close to the SISO KD for the sums. On the other hand, we remind that its performance loss is considerable.

An additional favourable feature of the SISO KD is shown in Fig. 7, which reports the worst-case complexity of the considered algorithms as the 99.9% percentile of the number of sums and products executed in the first six iterations of the IDDS vs SNR. It is clear that the SISO KD presents a considerably lower worst-case complexity w.r.t. the STSSD (in the three cases $L_{\max} = 0.1, 0.4, +\infty$) for all the considered values of SNR. This reflects in *a lower worst-case time of execution*.

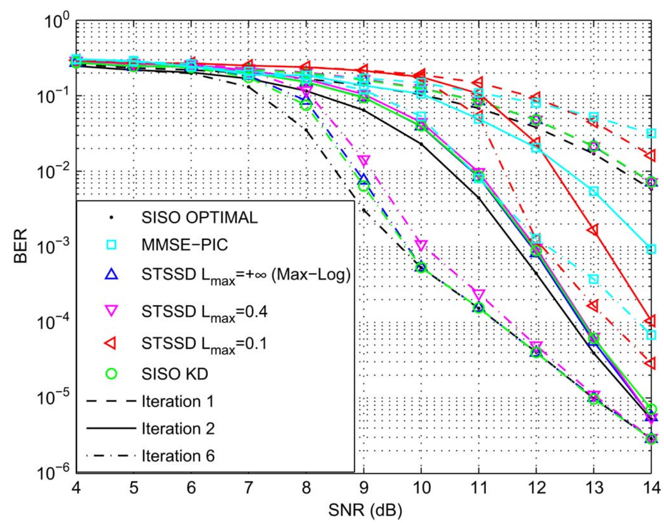


Fig. 8. BER vs SNR (dB) for an MIMO system with the IDDS, 16-QAM and $\tilde{K} = \tilde{N} = 4$. Comparison of the optimal SISO detector, MMSE-PIC, STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

Fig. 8 shows the performance for the case $\tilde{K} = \tilde{N} = 4$ with 16-QAM for iterations 1, 2, and 6 (also in this case the successive iterations do not improve the performance significantly). With this setup, both the STSSD without clipping (Max-Log performance) and the SISO KD exhibit a slight performance loss w.r.t. the optimum, quantifiable in about 0.5 dB of SNR when the target BER is 10^{-3} in the second iteration. The performance loss of the STSSD with $L_{\max} = 0.4$ w.r.t. the optimum is similar in the first two iterations and slightly higher in the sixth one (less than 0.5 dB of additional SNR to achieve BER $= 10^{-3}$). Differently, in the case $L_{\max} = 0.1$, the performance loss w.r.t. the optimum is apparent: 1.5 dB and 2.5 dB of additional SNR to achieve the BER of 10^{-3} in second and sixth iteration, respectively. The MMSE-PIC loses 1 dB more

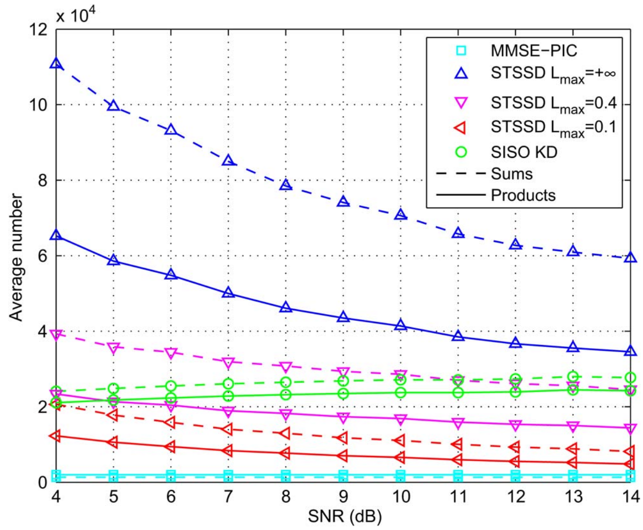


Fig. 9. Average number of elementary operations (sums and products) needed for algorithm implementation vs SNR (dB); first iteration of the IDDS for an MIMO system with 16-QAM and $\tilde{K} = \tilde{N} = 4$. Comparison of MMSE-PIC, STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

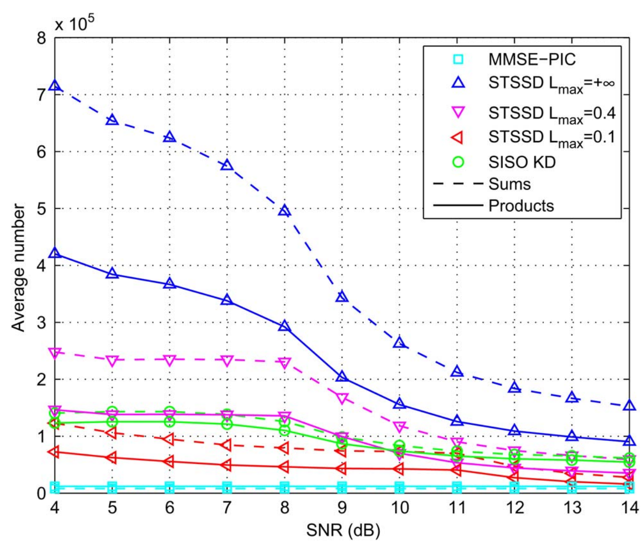


Fig. 10. Average number of elementary operations (sums and products) needed for algorithm implementation vs SNR (dB); accumulation of the first six iterations of the IDDS for an MIMO system with 16-QAM and $\tilde{K} = \tilde{N} = 4$. Comparison of MMSE-PIC, STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

in the second iteration at $\text{BER} = 10^{-3}$ w.r.t. the STSSD with $L_{\max} = 0.1$ and is comparable to it in the sixth one.

Figs. 9 and 10 show the results related to the computational complexity for the case $\tilde{K} = \tilde{N} = 4$ with 16-QAM. For this setup the complexity of the MMSE-PIC is always the lowest one for both sums and products. In the first iteration (Fig. 9) the SISO KD exhibits a lower complexity w.r.t. the STSSD without clipping for all values of SNR, while it is comparable to the STSSD with $L_{\max} = 0.4$. The complexity of the SISO KD is almost always greater than the one of the STSSD with $L_{\max} = 0.1$, which, for the previous discussion, cannot be anyway considered a good solution to the SISO detection problem in terms of BER. It is worth noticing that the complexity of

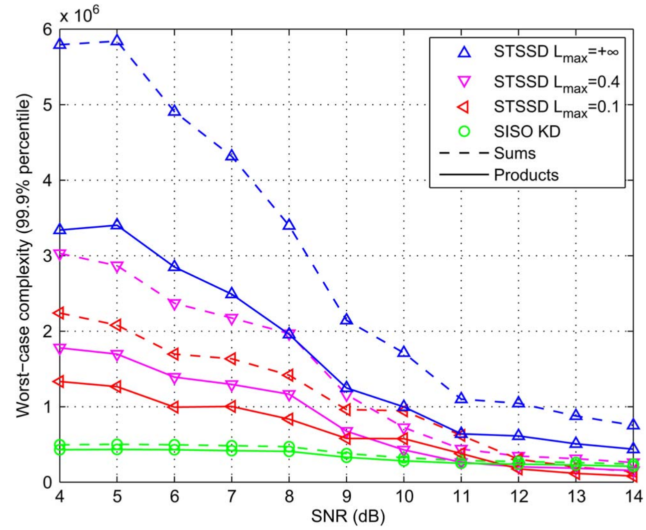


Fig. 11. Worst-case complexity as the 99.9% percentile of the number of elementary operations (sums and products) needed for algorithm implementation vs SNR (dB); accumulation of the first six iterations of the IDDS for an MIMO system with 16-QAM and $\tilde{K} = \tilde{N} = 4$. Comparison of STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

the SISO KD grows with the SNR, which can be explained rewriting (19) as[15]

$$\left| \boldsymbol{h}_i^T \boldsymbol{H} \bar{\boldsymbol{x}} + \boldsymbol{h}_i^T \boldsymbol{n} - \sum_{j=1}^{i-1} G_{ij} x_j \right| > \sum_{j=i+1}^{K} |G_{ij}|, \qquad (39)$$

where we used (4) and $\bar{\boldsymbol{x}}$ is the actual transmitted vector. Given a particular system setup, the only term in (39) affected by the value of the SNR is $\boldsymbol{h}_i^T \boldsymbol{n}$, whose variance decreases as the SNR grows. This means that the dominance conditions are less likely satisfied as the SNR grows, which leads to a lower pruning capability of the algorithm and, thus, to a higher complexity. With regard to the cumulated analysis of the first six iterations (Fig. 10), we observe that the computational complexity of the SISO KD is always lower than the one of the STSSD without clipping and comparable to the one of the STSSD with $L_{\max} = 0.4$ for higher values of SNR. The STSSD with $L_{\max} = 0.1$ achieves lower complexity w.r.t. the SISO KD, but, as already pointed out, it exhibits worse performance in terms of BER. Furthermore, we observe that the complexity of the SISO KD grows slightly until 6 dB of SNR and then starts decreasing. This can be explained rewriting (19) for the iterations after the first one as

$$\left| \frac{4}{\sigma^2} \left(\boldsymbol{h}_i^T \boldsymbol{H} \bar{\boldsymbol{x}} + \boldsymbol{h}_i^T \boldsymbol{n} - \sum_{j=1}^{i-1} G_{ij} x_j \right) + \lambda_i^A \right| > \frac{4}{\sigma^2} \sum_{j=i+1}^{K} |G_{ij}|. \qquad (40)$$

At low SNR, the absolute values of λ_i^A, $\forall i = 1, \ldots, k$, are likely to be low, because of the uncertainty originating from the

[15] As already noticed in Section II, in the first iteration there is no *a-priori* information for the SISO detection, namely $\lambda_i^A = 0, \forall i = 1, \ldots, K$.

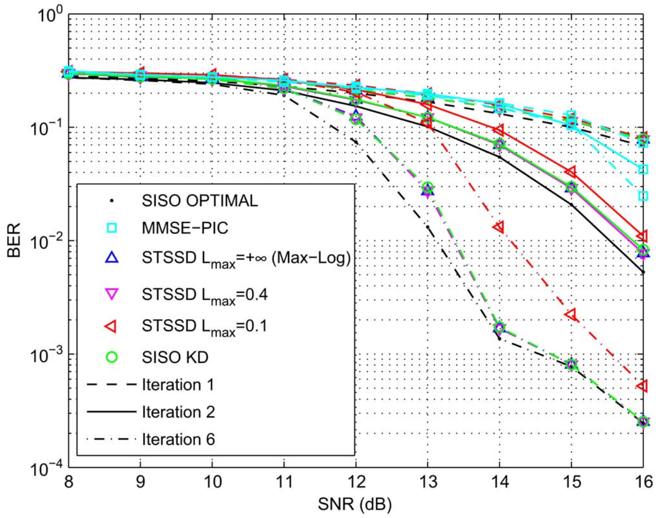


Fig. 12. BER vs SNR (dB) for an MIMO system with the IDDS, 64-QAM and $\tilde{K} = \tilde{N} = 2$. Comparison of the optimal SISO detector, MMSE-PIC, STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD

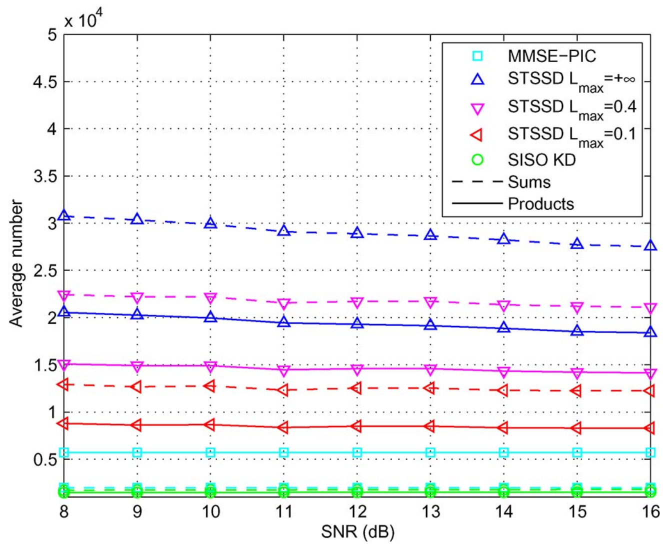


Fig. 13. Average number of elementary operations (sums and products) needed for algorithm implementation vs SNR (dB); first iteration of the IDDS for an MIMO system with 64-QAM and $\tilde{K} = \tilde{N} = 2$. Comparison of MMSE-PIC, STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

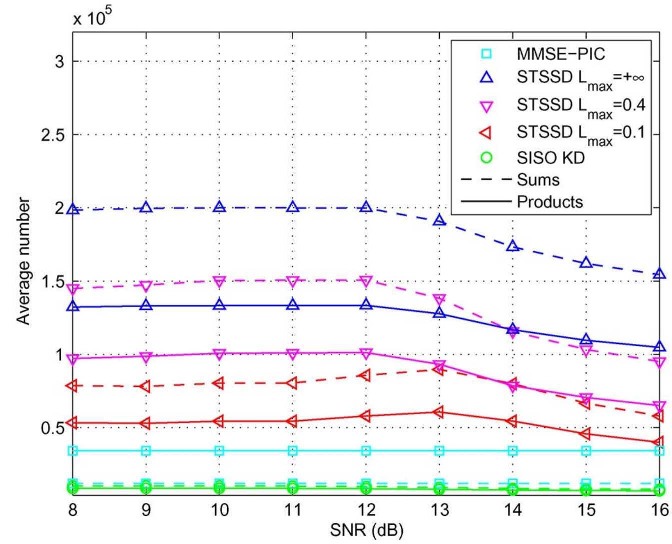


Fig. 14. Average number of elementary operations (sums and products) needed for algorithm implementation vs SNR (dB); accumulation of the first six iterations of the IDDS for an MIMO system with 64-QAM and $\tilde{K} = \tilde{N} = 2$. Comparison of MMSE-PIC, STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

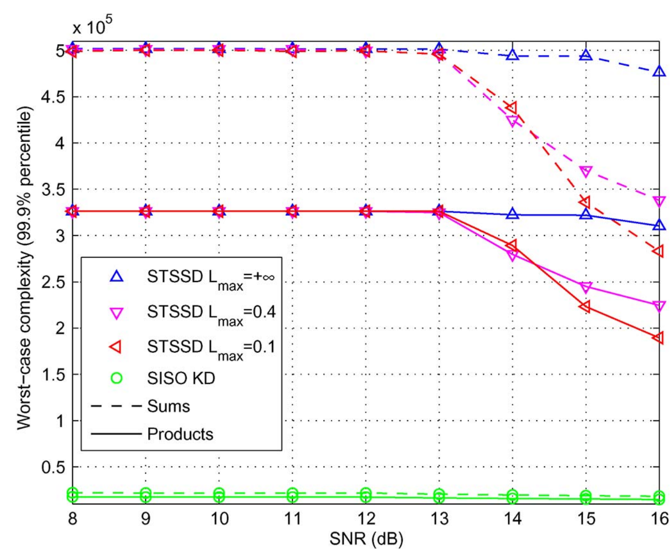


Fig. 15. Worst-case complexity as the 99.9% percentile of the number of elementary operations (sums and products) needed for algorithm implementation vs SNR (dB); accumulation of the first six iterations of the IDDS for an MIMO system with 64-QAM and $\tilde{K} = \tilde{N} = 2$. Comparison of STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

high noise on the channel, thus (40) is well approximated by (39) and we get the growing behaviour of the complexity as in the first iteration. At high SNR, the $\lambda_i^A, \forall\, i = 1, \ldots, k$, cannot be neglected anymore, because their absolute values are higher, and become the predominant term in (40), making it more likely to be satisfied.[16]

Fig. 11 shows the worst-case complexity of the considered algorithms as the 99.9% percentile of the number of sums and products executed in the first six iterations of the IDDS vs SNR for the case $\tilde{K} = \tilde{N} = 4$. The SISO KD exhibits a worst-case complexity substantially lower than the one of the STSSD without clipping over the considered range of SNR and lower than the one of the STSSD with $L_{\max} = 0.4$ for low values of SNR, while they are comparable for higher values of SNR.

Fig. 12 shows the performance for iterations 1, 2 and 6 for the case $\tilde{K} = \tilde{N} = 2$ with 64-QAM. For each iteration, the STSSD without clipping, the STSSD with $L_{\max} = 0.4$ and the SISO KD achieve roughly the same performance, which is very close to the optimum in the sixth iteration at $\text{BER} = 10^{-3}$. Conversely, the STSSD with $L_{\max} = 0.1$ achieves BER equal to 10^{-3} with 1 dB of SNR more w.r.t. the optimum in the sixth iteration. The MMSE-PIC is even worse in the second and sixth iteration.

[16]A similar profile (but less pronounced) of the complexity curves for both the first iteration (Fig. 5) and the first six iterations (Fig. 6) can be noticed also for the case $\tilde{K} = \tilde{N} = 2$.

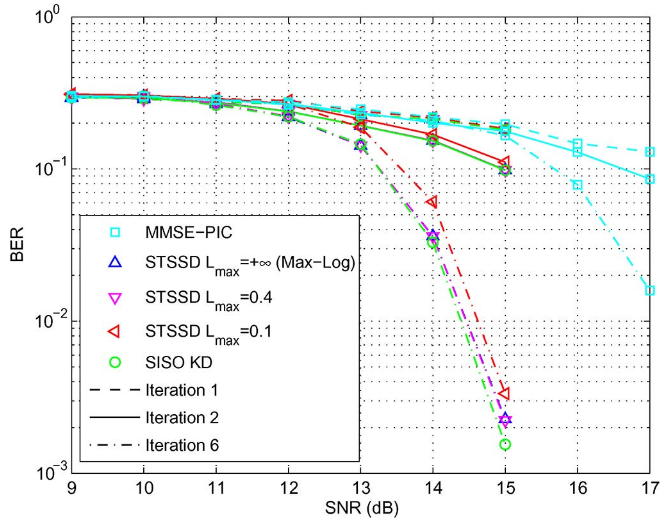


Fig. 16. BER vs SNR (dB) for an MIMO system with the IDDS, 64-QAM and $\tilde{K} = \tilde{N} = 4$. Comparison of MMSE-PIC, STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

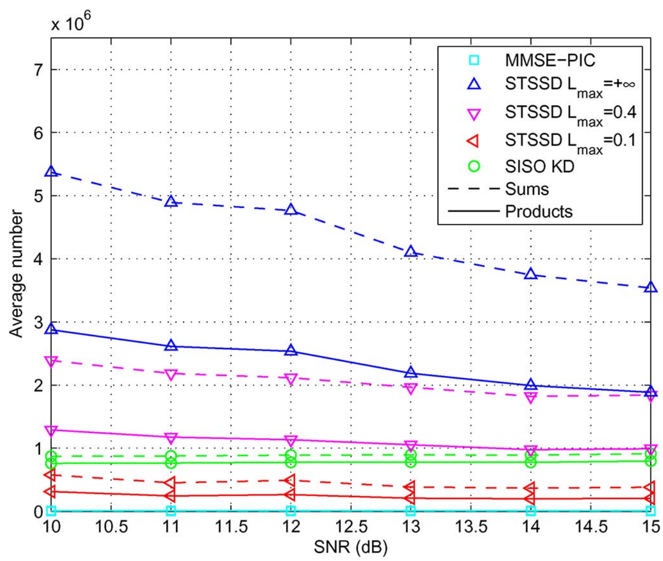


Fig. 17. Average number of elementary operations (sums and products) needed for algorithm implementation vs SNR (dB); first iteration of the IDDS for an MIMO system with 64-QAM and $\tilde{K} = \tilde{N} = 4$. Comparison of MMSE-PIC, STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

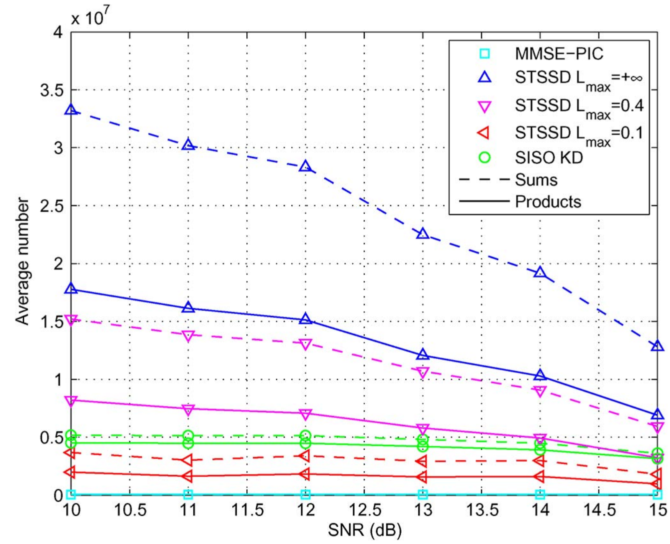


Fig. 18. Average number of elementary operations (sums and products) needed for algorithm implementation vs SNR (dB); accumulation of the first six iterations of the IDDS for an MIMO system with 64-QAM and $\tilde{K} = \tilde{N} = 4$. Comparison of MMSE-PIC, STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

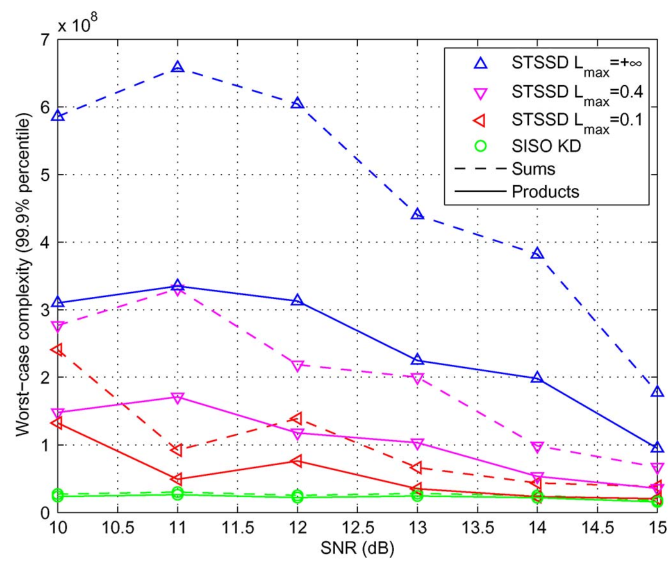


Fig. 19. Worst-case complexity as the 99.9% percentile of the number of elementary operations (sums and products) needed for algorithm implementation vs SNR (dB); accumulation of the first six iterations of the IDDS for an MIMO system with 64-QAM and $\tilde{K} = \tilde{N} = 4$. Comparison of STSSD ($L_{\max} = 0.1, 0.4, +\infty$) and SISO KD.

The complexity analysis is very similar to the case $\tilde{K} = \tilde{N} = 2$ with 16-QAM. Indeed, the number of products and the number of sums executed by the MMSE-PIC are between the SISO KD and the STSSD with $L_{\max} = 0.1$ and very close to the SISO KD, respectively. Furthermore, in the first iteration (Fig. 13) and the first six iterations (Fig. 14) it can be noticed the STSSD brings down the complexity as $L_{\max}$ decreases, but the SISO KD outperforms the STSSD for all the considered values of $L_{\max}$ for both sums and products. This is also confirmed by the worst-case complexity, as shown in Fig. 15.

Also for the case $\tilde{K} = \tilde{N} = 4$ with 64-QAM the performance of the SISO KD is practically equal to the Max-Log one, as shown in Fig. 16, and the complexity reduction w.r.t. the STSSD without clipping and the STSSD with $L_{\max} = 0.4$ is achieved, as shown in Figs. 17 and 18 for the average number of operations and in Fig. 19 for the worst-case complexity.

Finally, a simulation has been conducted to evaluate the capability of the SISO KD in an *underdetermined* setup with $\tilde{K} = 4$ and $\tilde{N} = 2$ (all the other parameters are the same as in the previous setups). The results are shown in Fig. 20, where the first three iterations of the IDDS with optimal SISO detection, Max-Log detection and the SISO KD are compared. It is apparent that in the underdetermined case the Max-Log approximation entails a non-negligible performance gap w.r.t. optimal detection, clearly evident in the third iteration. However, as can be readily noted, the SISO KD does not further reduce the performance.

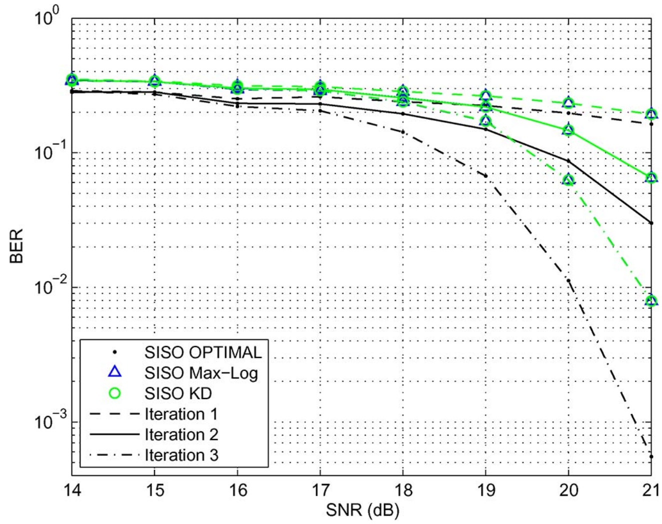


Fig. 20. BER vs SNR (dB) for an MIMO system with the IDDS, 16-QAM, $\tilde{K} = 4$ and $\tilde{N} = 2$. Comparison of the optimal SISO detector, Max-Log approximated SISO detector and SISO KD.

VII. Conclusion

We proposed an SISO MIMO detector, named SISO KD, which is a single tree-search branch-and-bound algorithm. It achieves significant complexity reduction through the dominance conditions, which exploit the interference properties of the channel matrix and the a-priori information on the transmitted bits. We presented an analysis of the computational complexity of the proposed algorithm, based on the number of elementary operations required for its implementation. We demonstrated the effectiveness of the SISO KD through simulations, which show performance very close to the optimal solution and a remarkable complexity reduction w.r.t. the STSSD.

Appendix
Local Minima Conditions

Here we derive the conditions for a generic $\boldsymbol{x}$ to be a local minimum of (9). Starting from (13), we get

$$\frac{\|\boldsymbol{y} - \boldsymbol{H}x\|^2}{\sigma^2} - \frac{1}{2}\sum_{j=1}^{K} x_j \lambda_j^A < \frac{\|\boldsymbol{y} - \boldsymbol{H}\alpha_i(\boldsymbol{x})\|^2}{\sigma^2} - \frac{1}{2}\sum_{j=1}^{K} \alpha_{ij}(\boldsymbol{x})\lambda_j^A, \quad (41)$$

where $\alpha_{ij}(\boldsymbol{x})$ is the jth element of $\alpha_i(\boldsymbol{x})$,

$$\frac{\|\boldsymbol{y} - \boldsymbol{H}\alpha_i(\boldsymbol{x})\|^2 - \|\boldsymbol{y} - \boldsymbol{H}x\|^2}{\sigma^2} + x_i\lambda_i^A > 0, \quad (42)$$

$$\frac{\alpha_i(\boldsymbol{x})^T\boldsymbol{H}^T\boldsymbol{H}\alpha_i(\boldsymbol{x}) - \boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{H}x - 2\alpha_i(\boldsymbol{x})^T\boldsymbol{H}^T\boldsymbol{y}}{\sigma^2} + \frac{2\boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{y}}{\sigma^2} + x_i\lambda_i^A > 0. \quad (43)$$

Adding and subtracting $\boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{H}\alpha_i(\boldsymbol{x})$ at the numerator of the fraction and defining $\triangle\boldsymbol{x} \triangleq \boldsymbol{x} - \alpha_i(\boldsymbol{x}) = [\boldsymbol{0}_{1\times i-1}\ 2x_i\ \boldsymbol{0}_{1\times K-i}]^T$, we get

$$\frac{2(\triangle\boldsymbol{x})^T\boldsymbol{H}T\boldsymbol{y} - (\triangle\boldsymbol{x})^T\boldsymbol{H}^T\boldsymbol{H}\alpha_i(\boldsymbol{x}) - \boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{H}\triangle\boldsymbol{x}}{\sigma^2} + x_i\lambda_i^A > 0. \quad (44)$$

Adding and subtracting $(\triangle\boldsymbol{x})^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x}$, we obtain

$$\frac{2(\triangle\boldsymbol{x})^T\boldsymbol{H}^T\boldsymbol{y} + (\triangle\boldsymbol{x})^T\boldsymbol{H}^T\boldsymbol{H}\triangle\boldsymbol{x} - 2(\triangle\boldsymbol{x})^T\boldsymbol{H}^T\boldsymbol{H}x}{\sigma^2} + x_i\lambda_i^A > 0, \quad (45)$$

$$\frac{4x_i\boldsymbol{h}_i^T\boldsymbol{y} + 4\boldsymbol{h}_i^T\boldsymbol{h}_i - 4x_i\boldsymbol{h}_i^T\boldsymbol{H}x}{\sigma^2} + x_i\lambda_i^A > 0, \quad (46)$$

which provides (14).

Acknowledgment

The authors would like to express their sincere gratitude to the Associate Editor and the anonymous reviewers for taking their time into reviewing this manuscript and providing comments that contributed to improve the quality and the readability of the manuscript.

References

[1] C. Studer and H. Bölcskei, "Soft-input soft-output single tree-search sphere decoding," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4827–4842, Oct. 2010.

[2] E. Biglieri *et al.*, *MIMO Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[3] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.

[4] J. Luo, K. R. Pattipati, P. K. Willett, and G. M. Levchuk, "Fast optimal and suboptimal any-time algorithms for CDMA multiuser detection based on branch and bound," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 632–642, Apr. 2004.

[5] W. H. Mow, "Universal lattice decoding: Principle and recent advances," *Wireless Commun. Mobile Comput.*, vol. 3, no. 5, pp. 553–569, Aug. 2003.

[6] L. Ekroot and S. Dolinar, "A* decoding of block codes," *IEEE Trans. Commun.*, vol. 44, no. 9, pp. 1052–1056, Sep. 1996.

[7] P. Salvo Rossi and R. R. Müller, "Joint twofold-iterative channel estimation and multiuser detection for MIMO-OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 7, no. 11, pp. 4719–4729, Nov. 2008.

[8] G. M. Kraidy and P. Salvo Rossi, "Achieving full diversity over the MIMO fading channel with space-time precoders and iterative linear receivers," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2407–2411, Aug. 2011.

[9] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE ICC*, May 1993, vol. 2, pp. 1064–1070.

[10] M. Sellathurai and S. Haykin, "TURBO-BLAST for wireless communications: Theory and experiments," *IEEE Trans. Signal Process.*, vol. 50, no. 10, pp. 2538–2546, Oct. 2002.

[11] M. Nekuii, M. Kisialiou, T. N. Davidson, and Z.-Q. Luo, "Efficient soft-output demodulation of MIMO QPSK via semidefinite relaxation," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 8, pp. 1426–1437, Dec. 2011.

[12] D. Persson and E. G. Larsson, "Partial marginalization soft MIMO detection with high order constellations," *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 453–458, Jan. 2011.

[13] M. Wu, Y. Sun, S. Gupta, and J. R. Cavallaro, "Implementation of a high throughput soft MIMO detector on GPU," *J. Signal Process. Syst.*, vol. 64, no. 1, pp. 123–136, Jul. 2011.

[14] Y. Sun and J. R. Cavallaro, "High-throughput soft-output MIMO detector based on path-preserving trellis-search algorithm," *IEEE Trans. VLSI Syst.*, vol. 20, no. 7, pp. 1235–1247, Jul. 2012.

[15] C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation," *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1754–1765, Jul. 2011.

[16] U. Fincke 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.

[17] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programm.*, vol. 66, no. 1–3, pp. 181–199, Aug. 1994.

[18] W. H. Mow, "Maximum likelihood sequence estimation from the lattice viewpoint," *IEEE Trans. Inf. Theory*, vol. 40, no. 5, pp. 1591–1600, Sep. 1994.

[19] 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.

[20] O. Damen, A. Chkeif, and J.-C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Commun. Lett.*, vol. 4, no. 5, pp. 161–163, May 2000.

[21] 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.

[22] 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.

[23] T. Cui and C. Tellambura, "An efficient generalized sphere decoder for rank-deficient MIMO systems," *IEEE Commun. Lett.*, vol. 9, no. 5, pp. 423–425, May 2005.

[24] X.-W. Chang and X. Yang, "An efficient tree search decoder with column reordering for underdetermined MIMO systems," in *Proc. IEEE GLOBECOM*, Nov. 2007, pp. 4375–4379.

[25] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 491–503, Mar. 2006.

[26] J. Jaldén and B. Ottersten, "Parallel implementation of a soft output sphere decoder," in *Conf. Rec. 39th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2005, pp. 581–585.

[27] N. Prasad, K. Y. Kalbat, and X. Wang, "Optimally efficient Max-Log APP demodulation in MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 8, pp. 1400–1414, Dec. 2011.

[28] L. G. Barbero and J. S. Thompson, "Fixing the complexity of the sphere decoder for MIMO detection," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2131–2142, Jun. 2008.

[29] L. Liu, J. Lofgren, and P. Nilsson, "Low-complexity likelihood information generation for spatial-multiplexing MIMO signal detection," *IEEE Trans. Veh. Technol.*, vol. 61, no. 2, pp. 607–617, Feb. 2012.

[30] S.-L. Shieh, R.-D. Chiu, S.-L. Feng, and P.-N. Chen, "Low-complexity soft-output sphere decoding with modified repeated tree search strategy," *IEEE Commun. Lett.*, vol. 17, no. 1, pp. 51–54, Jan. 2013.

[31] J. Lee, B. Shim, and I. Kang, "Soft-input soft-output list sphere detection with a probabilistic radius tightening," *IEEE Trans. Wireless Commun.*, vol. 11, no. 8, pp. 2848–2857, Aug. 2012.

[32] R. Wang and G. B. Giannakis, "Approaching MIMO channel capacity with soft detection based on hard sphere decoding," *IEEE Trans. Commun.*, vol. 54, no. 4, pp. 587–590, Apr. 2006.

[33] P. Radosavljevic, Y. Guo, and J. R. Cavallaro, "Probabilistically bounded soft sphere detection for MIMO-OFDM receivers: Algorithm and system architecture," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 8, pp. 1318–1330, Oct. 2009.

[34] X. Chen, G. He, and J. Ma, "VLSI implementation of a high-throughput iterative fixed-complexity sphere decoder," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 5, pp. 272–276, May 2013.

[35] G. Papa, D. Ciuonzo, G. Romano, and P. Salvo Rossi, "Soft-input soft-output king decoder for coded MIMO wireless communications," in *Proc. 8th ISWCS*, Nov. 2011, pp. 760–763.

[36] G. Romano, D. Ciuonzo, P. Salvo Rossi, and F. Palmieri, "Low-complexity dominance-based sphere decoder for MIMO systems," *J. Signal Process.*, vol. 93, no. 9, pp. 2500–2509, Sep. 2013.

[37] G. Romano, D. Ciuonzo, P. Salvo Rossi, and F. Palmieri, "Tree-search ML detection for underdetermined MIMO systems with M-PSK constellations," in *Proc. 7th ISWCS*, Sep. 2010, pp. 102–106.

[38] G. Kechriotis and E. S. Manolakos, "A hybrid digital signal processing-neural network CDMA multiuser detection scheme," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 2, pp. 96–104, Feb. 1996.

[39] P. Ödling, H. B. Eriksson, and P. O. Dörjesson, "Making MLSD decisions by thresholding the matched filter output," *IEEE Trans. Commun.*, vol. 48, no. 2, pp. 324–332, Feb. 2000.

[40] P. Švač, F. Meyer, E. Riegler, and F. Hlawatsch, "Soft-heuristic detector for large MIMO systems," *IEEE Trans. Signal Process.*, vol. 61, no. 18, pp. 4573–4586, Sep. 2013.

[41] B. Steingrimsson, Z.-Q. Luo, and K. M. Wong, "Soft quasi-maximum-likelihood detection for multiple-antenna wireless channels," *IEEE Trans. Signal Process.*, vol. 51, no. 11, pp. 2710–2719, Nov. 2003.

[42] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.

[43] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Commun. Lett.*, vol. 1, no. 1, pp. 22–24, Jan. 1997.

[44] Z. Mao, X. Wang, and X. Wang, "Semidefinite programming relaxation approach for multiuser detection of QAM signals," *IEEE Trans. Wireless Commun.*, vol. 6, no. 12, pp. 4275–4279, Dec. 2007.

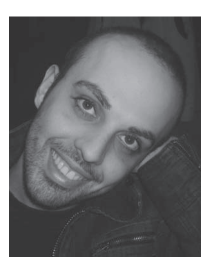
[45] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE ICC*, Jun. 1995, vol. 2, pp. 1009–1013.

[46] 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.

[47] G. Colavolpe and A. Barbieri, "On MAP symbol detection for ISI channels using the Ungerboeck observation model," *IEEE Commun. Lett.*, vol. 9, no. 8, pp. 720–722, Aug. 2005.

[48] M. Lončar and F. Rusek, "On reduced-complexity equalization based on Ungerboeck and Forney observation models," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3784–3789, Aug. 2008.

Giuseppe Papa (S'13) was born in Caserta, Italy, on May 17th, 1985. He received the B.Sc. (summa cum laude) and the M.Sc. (summa cum laude) degrees in computer engineering, in 2007 and 2011, respectively, both from Second University of Naples, Aversa (CE), Italy. In 2011–2012 and 2014 he was involved in the Visiting Researcher Programme (VRP) at Centre for Maritime Research and Experimentation (CMRE), La Spezia, Italy, under the Maritime Situational Awareness (MSA) project. He is currently a Ph.D. student in electronics and computer engineering at the Department of Industrial and Information Engineering, Second University of Naples, Aversa (CE), Italy.



Domenico Ciuonzo (S'11) was born in Aversa, Italy, on June 29th, 1985. He received the B.Sc. (summa cum laude) and the M.Sc. (summa cum laude) degrees in computer engineering and the Ph.D. degree in electronic engineering from the Second University of Naples (SUN), Aversa, Italy, in 2007, 2009, and 2013, respectively. In 2011 he was involved in the Visiting Researcher Programme of Centre for Maritime Research and Experimentation, La Spezia, Italy; he worked in the "Maritime Situation Awareness" project. In 2012 he was a visiting scholar at the Electrical and Computer Engineering Department of University of Connecticut (UConn), Storrs, US. In 2013–2014 he was a PostDoc at Dept. of Industrial and Information Engineering of SUN, under the project "MICENEA," funded by MIUR within the program Futuro in Ricerca (FIRB) 2013. He is currently a PostDoc researcher at DIETI, University of Naples, "Federico II," Italy. His research interests are mainly in the areas of Data Fusion, Statistical Signal Processing, Target Tracking and Wireless Sensor Networks. The quality of his reviewing activity was recognized by the Editorial Board of IEEE COMMUNICATIONS LETTERS, which nominated him "Exemplary Reviewer" for the year 2013. Dr. Ciuonzo is a reviewer for several IEEE, Elsevier and Wiley journals in the areas of communications, defense and signal processing. He has also served as reviewer and TPC member for several IEEE conferences.

Gianmarco Romano (M'11) received the "Laurea" degree in electronic engineering from the University of Naples "Federico II" and the Ph.D. degree from the Second University of Naples, in 2000 and 2004, respectively. From 2000 to 2002 he has been Researcher at the National Laboratory for Multimedia Communications (C.N.I.T.) in Naples, Italy. In 2003 he was Visiting Scholar at the Department of Electrical and Electronic Engineering, University of Connecticut, Storrs, USA. Since 2005 he has been with the Department of Information Engineering, Second University of Naples and in 2006 has been appointed Assistant Professor. His research interests fall within the areas of communications and signal processing.

Pierluigi Salvo Rossi (SM'11) was born in Naples, Italy, on April 26, 1977. He received the "Laurea" degree in telecommunications engineering (*summa cum laude*) and the Ph.D. degree in computer engineering, in 2002 and 2005, respectively, both from the University of Naples "Federico II," Naples, Italy.

During his Ph.D. studies, he has been a researcher at the Inter-departmental Research Center for Signals Analysis and Synthesis (CIRASS), University of Naples "Federico II," Naples, Italy; at the Department of Information Engineering, Second University of Naples, Aversa (CE), Italy; at the the Communications and Signal Processing Laboratory (CSPL), Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA, US; and an adjunct professor at the Faculty of Engineering, Second University of Naples, Aversa (CE), Italy. From 2005 to 2008 he worked as a postdoc at the Department of Computer Science and Systems, University of Naples "Federico II," Naples, Italy; at the Department of Information Engineering, Second University of Naples, Aversa (CE), Italy; at the Department of Electronics and Telecommunications, Norwegian University of Science and Technology, Trondheim, Norway; and visited the Department of Electrical and Information Technology, Lund University, Lund, Sweden. Since 2008, he is an assistant professor (tenured in 2011) in telecommunications at the Department of Industrial and Information Engineering, Second University of Naples, Aversa (CE), Italy; and has been a guest researcher and professor at the Department of Electronics and Telecommunications, Norwegian University of Science and Technology, Trondheim, Norway.

He is an Editor for the IEEE COMMUNICATIONS LETTERS. His research interests fall within the areas of communications and signal processing.