Soft-Input Soft-Output Single Tree-Search Sphere Decoding

Christoph Studer, *Student Member, IEEE* and Helmut Bölcskei, *Fellow, IEEE*

***Abstract*—Soft-input soft-output (SISO) detection algorithms form the basis for iterative decoding. The computational complexity of SISO detection often poses significant challenges for practical receiver implementations, in particular in the context of multiple-input multiple-output (MIMO) wireless communication systems. In this paper, we present a low-complexity SISO sphere-decoding algorithm, based on the single tree-search paradigm proposed originally for soft-output MIMO detection in Studer *et al.*, IEEE J-SAC, 2008. The new algorithm incorporates clipping of the extrinsic log-likelihood ratios (LLRs) into the tree-search, which results in significant complexity savings and allows to cover a large performance/complexity tradeoff region by adjusting a single parameter. Furthermore, we propose a new method for correcting approximate LLRs —resulting from sub-optimal detectors— which (often significantly) improves detection performance at low additional computational complexity.**

***Index Terms*—Multiple-input multiple-output (MIMO) communication, soft-input soft-output detection, sphere decoding, iterative MIMO decoding**

I. Introduction

SOFT-input soft-output (SISO) detection constitutes the basis for iterative decoding in multiple-input multiple-output (MIMO) systems, which, in general, achieves significantly better (error-rate) performance than decoding based on hard-output or soft-output-only detection algorithms [1]. Unfortunately, this performance gain comes at the cost of a significant (often prohibitive in terms of practical implementation) increase in computational complexity.

Various SISO detection algorithms for MIMO systems offering different performance/complexity tradeoffs have been proposed in the literature, see e.g., [1]–[6]. However, implementing different algorithms, each optimized for a maximum allowed detection effort or for a particular system configuration, would entail considerable circuit complexity. A practical SISO detector for MIMO systems should therefore cover a wide range of performance/complexity tradeoffs and be easily adjustable through a *single* tunable detection algorithm.

Soft-output single tree-search (STS) sphere decoding (SD) in combination with log-likelihood ratio (LLR) clipping [7] has been demonstrated to be suitable for VLSI implementation and allows to conveniently tune detection performance between maximum-likelihood (ML) a posteriori probability (APP) soft-output detection and (low-complexity) hard-output detection. The STS-SD concept is therefore a promising basis for efficient SISO detection in MIMO systems.

Contributions: We describe a SISO STS-SD algorithm that is tunable between max-log optimal SISO and hard-output maximum a posteriori (MAP) detection performance. To this end, we extend the soft-output STS-SD algorithm introduced in [7], [8] by incorporating a priori information. Specifically, this will be done through a method that significantly reduces the tree-search complexity —relative to existing approaches in the literature (e.g., [5], [6])— and avoids the computation of transcendental functions, while maintaining (max-log) optimality. The basic idea underlying this complexity reduction and the tunability of the algorithm is to incorporate clipping of the extrinsic LLRs into the tree search. This requires that the list administration concept and the tree-pruning criterion proposed for soft-output STS-SD in [7] be suitably modified. We furthermore describe an approach for the compensation of self-interference in the LLRs —caused by channel-matrix regularization— directly in the tree search. In addition, we describe a new method for correcting approximate LLRs — resulting from sub-optimal detectors— which (often significantly) improves detection performance at low additional computational complexity. Simulation results show that the resulting SISO STS-SD operates between 1.5 dB and 5.3 dB away from outage capacity at remarkably low computational complexity. In addition, the algorithm offers a significantly larger performance/complexity tradeoff region than the soft-output STS-SD algorithm proposed in [7].

Notation: Matrices are set in boldface capital letters, vectors in boldface lowercase letters. The superscripts ^{T} and ^{H} stand for transpose and conjugate transpose, respectively. We write $A_{i,j}$ for the entry in the ith row and jth column of the matrix $\mathbf{A}$ and b_i for the ith entry of the vector $\mathbf{b} = [\, b_1 \ \cdots \ b_N \,]^T$. The ℓ^2-norm of the vector $\mathbf{b}$ is denoted by $\|\mathbf{b}\|$. $\mathbf{I}_N$ and $\mathbf{0}_{M\times N}$ refer to the $N \times N$ identity matrix and the $M \times N$ all-zero matrix, respectively. Slightly abusing common terminology, we call an $M \times N$ matrix $\mathbf{A}$, where $M \geq N$, satisfying $\mathbf{A}^H\mathbf{A} = \mathbf{I}_N$, unitary. $|\mathcal{O}|$ denotes the cardinality of the set $\mathcal{O}$. The probability of an event $\mathcal{Z}$ is referred to as $\mathrm{P}[\mathcal{Z}]$, the probability density function of a continuous random variable (RV) z is denoted by $\mathrm{p}(z)$ and $\mathbb{E}[Z]$ stands for the expectation of the RV Z. $\overline{x}$ is the binary complement of $x \in \{+1, -1\}$, i.e., $\overline{x} = -x$.

This paper was presented in part at the IEEE International Symposium on Information Theory (ISIT), Toronto, Ontario, Canada, July 2008. This work was partially supported by the STREP project No. IST-026905 (MASCOT) within the Sixth Framework Programme (FP6) of the European Commission and by the Swiss Innovation Promotion Agency (KTI/CTI) project No. 9268.1 PFNM-NM.

The authors are with the Department of Information Technology and Electrical Engineering, ETH Zurich, CH-8092 Zurich, Switzerland (e-mail: {studerc,boelcskei}@nari.ee.ethz.ch).

MATLAB code of the SISO STS-SD algorithm is available for download at `http://www.nari.ee.ethz.ch/commth/research/`

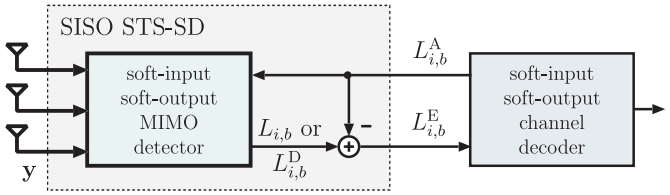


Fig. 1. Iterative MIMO decoder. SISO STS-SD (corresponding to the dashed box) directly computes extrinsic LLRs.

Outline: The remainder of this paper is organized as follows. Section II reviews the transformation of soft-input soft-output MIMO detection into a tree-search problem and presents new methods for tightening of the tree-pruning criterion and for incorporating a priori information into the tree search. Section III describes the new SISO STS-SD algorithm. In Section IV, we propose a method for compensating the impact of channel-matrix regularization on LLRs directly in the tree search. A new technique for computationally efficient correction of approximate LLRs —resulting from, e.g., the max-log approximation, channel-matrix regularization, and/or early termination [7], [9]— is presented in Section V. Simulation results are provided in Section VI. We conclude in Section VII.

II. Soft-Input Soft-Output Sphere Decoding

Consider a MIMO system with M_{T} transmit and $M_{\mathrm{R}} \geq M_{\mathrm{T}}$ receive antennas. The coded bit-stream to be transmitted is mapped to (a sequence of) M_{T}-dimensional transmit symbol vectors $\mathbf{s} \in \mathcal{O}^{M_{\mathrm{T}}}$, where $\mathcal{O}$ stands for the underlying complex scalar constellation[1] and $|\mathcal{O}| = 2^Q$. Each symbol vector $\mathbf{s}$ is associated with a label vector $\mathbf{x}$ containing $M_{\mathrm{T}}Q$ binary values chosen from the set $\{+1, -1\}$ where the null element (0 in binary logic) of GF(2) corresponds to $+1$. The corresponding bits are denoted by $x_{i,b}$, where the indices i and b refer to the bth bit in the binary label of the ith entry of the symbol vector $\mathbf{s} = [\, s_1 \ \cdots \ s_{M_{\mathrm{T}}} \,]^T$. The associated complex baseband input-output relation is given by

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

where $\mathbf{H}$ stands for the $M_{\mathrm{R}} \times M_{\mathrm{T}}$ channel matrix, $\mathbf{y}$ is the M_{R}-dimensional received signal vector, and $\mathbf{n}$ is an i.i.d. circularly symmetric complex Gaussian distributed M_{R}-dimensional noise vector with variance N_0 per complex entry. Different transmit powers on the individual transmit antennas are assumed to be absorbed in the channel matrix $\mathbf{H}$, which —including the corresponding scaling factors— will be referred to as the physical MIMO channel. Throughout the paper, we consider coherent detection, i.e., we assume that the receiver knows the realization of the channel matrix $\mathbf{H}$ perfectly.

A. Max-Log LLR Computation as a Tree Search

In the following, we consider an iterative MIMO decoder as depicted in Fig. 1. The soft-input soft-output MIMO detector computes *intrinsic* LLRs according to [1]

$$L_{i,b} \triangleq \log\left(\frac{\mathrm{P}[x_{i,b} = +1 \,|\, \mathbf{y}, \mathbf{H}]}{\mathrm{P}[x_{i,b} = -1 \,|\, \mathbf{y}, \mathbf{H}]}\right) \tag{2}$$

for all bits $i = 1, \ldots, M_{\mathrm{T}}$, $b = 1, \ldots, Q$, in the label $\mathbf{x}$. Bayes's theorem applied to (2) leads to the equivalent formulation

$$L_{i,b} = \log\left(\sum_{\mathbf{s} \in \mathcal{X}_{i,b}^{(+1)}} \mathrm{p}(\mathbf{y} \,|\, \mathbf{s}, \mathbf{H})\, \mathrm{P}[\mathbf{s}]\right) - \log\left(\sum_{\mathbf{s} \in \mathcal{X}_{i,b}^{(-1)}} \mathrm{p}(\mathbf{y} \,|\, \mathbf{s}, \mathbf{H})\, \mathrm{P}[\mathbf{s}]\right) \tag{3}$$

where $\mathcal{X}_{i,b}^{(+1)}$ and $\mathcal{X}_{i,b}^{(-1)}$ are the sets of symbol vectors that have the bit corresponding to the indices i and b equal to $+1$ and -1, respectively. The probability density function $\mathrm{p}(\mathbf{y} \,|\, \mathbf{s}, \mathbf{H})$ in (3) is given by

$$\mathrm{p}(\mathbf{y} \,|\, \mathbf{s}, \mathbf{H}) = \frac{1}{(\pi N_0)^{M_{\mathrm{R}}}} \exp\left(-\frac{\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2}{N_0}\right)$$

and the prior $\mathrm{P}[\mathbf{s}]$ is, e.g., delivered by an outer channel decoder in the form of *a priori* LLRs

$$L_{i,b}^{\mathrm{A}} \triangleq \log\left(\frac{\mathrm{P}[x_{i,b} = +1]}{\mathrm{P}[x_{i,b} = -1]}\right), \quad \forall i, b.$$

Based on the intrinsic LLRs in (3), the MIMO detector computes the *extrinsic* LLRs

$$L_{i,b}^{\mathrm{E}} \triangleq L_{i,b} - L_{i,b}^{\mathrm{A}}, \quad \forall i, b \tag{4}$$

that are passed to a subsequent SISO channel decoder. Straightforward evaluation of (3) requires the computation of $|\mathcal{O}|^{M_{\mathrm{T}}}$ Euclidean distances per LLR value, which, in general, leads to prohibitive computational complexity. We therefore apply the standard max-log approximation[2] to (3), which enables us to reformulate the LLR computation problem as a weighted tree-search problem that can be solved efficiently using the SD algorithm [7], [8], [11]–[18].

We start by QR-decomposing the channel matrix $\mathbf{H}$ according to $\mathbf{H} = \mathbf{Q}\mathbf{R}$, where the $M_{\mathrm{R}} \times M_{\mathrm{T}}$ matrix $\mathbf{Q}$ is unitary and the $M_{\mathrm{T}} \times M_{\mathrm{T}}$ upper-triangular matrix $\mathbf{R}$ has real-valued positive entries on its main diagonal. Left-multiplying (1) by $\mathbf{Q}^H$ leads to the modified input-output relation

$$\tilde{\mathbf{y}} = \mathbf{R}\mathbf{s} + \mathbf{Q}^H\mathbf{n} \tag{5}$$

where $\tilde{\mathbf{y}} \triangleq \mathbf{Q}^H\mathbf{y}$ and $\mathbf{Q}^H\mathbf{n}$ is also i.i.d. circularly symmetric complex Gaussian with variance N_0 per complex entry. Based on (3), the soft-input soft-output MIMO detector then computes intrinsic max-log LLRs according to [1]

$$L_{i,b}^{\mathrm{D}} \triangleq \min_{\mathbf{s} \in \mathcal{X}_{i,b}^{(-1)}} \left\{\frac{1}{N_0}\left\|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}\right\|^2 - \log \mathrm{P}[\mathbf{s}]\right\} - \min_{\mathbf{s} \in \mathcal{X}_{i,b}^{(+1)}} \left\{\frac{1}{N_0}\|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2 - \log \mathrm{P}[\mathbf{s}]\right\} \tag{6}$$

[1]The algorithm developed in this paper can also be formulated for the case where different constellations are used on different transmit antennas. However, for the sake of simplicity of exposition, we restrict ourselves to employing the same constellation on all transmit antennas.

[2]The max-log approximation corresponds to $\log\left(\sum_k \exp(a_k)\right) \approx \max_k\{a_k\}$ and entails a performance loss compared to using the exact LLRs in (3). As shown in [10], this loss is small, in general.

followed by the computation of extrinsic max-log LLRs, obtained by replacing $L_{i,b}$ in (4) by $L_{i,b}^{\mathrm{D}}$. In the remainder of the paper, whenever we speak of extrinsic LLRs and intrinsic LLRs, we mean the extrinsic max-log LLRs $L_{i,b}^{\mathrm{E}} = L_{i,b}^{\mathrm{D}} - L_{i,b}^{\mathrm{A}}$ and the intrinsic max-log LLRs in (6), respectively. Note that in each of the two minima in (6) we neglected the additive constant that results from the part of the noise $\mathbf{n}$ that is orthogonal to the range space of $\mathbf{H}$. This is possible as the constant in question is independent of $\mathbf{s}$ and, hence, cancels out upon taking the difference in (6).

For each bit, one of the two minima in (6) corresponds to

$$\lambda^{\mathrm{MAP}} \triangleq \frac{1}{N_0}\left\|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}^{\mathrm{MAP}}\right\|^2 - \log \mathrm{P}\left[\mathbf{s}^{\mathrm{MAP}}\right] \tag{7}$$

associated with the MAP solution of the MIMO detection problem

$$\mathbf{s}^{\mathrm{MAP}} = \underset{\mathbf{s}\in\mathcal{O}^{M_{\mathrm{T}}}}{\arg\min}\left\{\frac{1}{N_0}\left\|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}\right\|^2 - \log \mathrm{P}[\mathbf{s}]\right\}. \tag{8}$$

In the remainder of the paper, $\mathbf{x}^{\mathrm{MAP}}$ stands for the (binary-valued) label vector associated with the MAP solution $\mathbf{s}^{\mathrm{MAP}}$. The other minimum in (6) can be written as

$$\lambda_{i,b}^{\overline{\mathrm{MAP}}} \triangleq \min_{\mathbf{s}\in\mathcal{X}_{i,b}^{\overline{\mathrm{MAP}}}}\left\{\frac{1}{N_0}\left\|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}\right\|^2 - \log \mathrm{P}[\mathbf{s}]\right\} \tag{9}$$

where $\mathcal{X}_{i,b}^{\overline{\mathrm{MAP}}} \triangleq \mathcal{X}_{i,b}^{\left(\overline{x_{i,b}^{\mathrm{MAP}}}\right)}$ and $\overline{x_{i,b}^{\mathrm{MAP}}}$ denotes the (bit-wise) *counter-hypothesis* to the MAP hypothesis. With the definitions (7) and (9), the intrinsic LLRs in (6) can be written in compact form as

$$L_{i,b}^{\mathrm{D}} = \begin{cases} \lambda_{i,b}^{\overline{\mathrm{MAP}}} - \lambda^{\mathrm{MAP}}, & x_{i,b}^{\mathrm{MAP}} = +1 \\ \lambda^{\mathrm{MAP}} - \lambda_{i,b}^{\overline{\mathrm{MAP}}}, & x_{i,b}^{\mathrm{MAP}} = -1. \end{cases} \tag{10}$$

We can therefore conclude that efficient max-log-optimal soft-input soft-output MIMO detection reduces to efficiently identifying $\mathbf{s}^{\mathrm{MAP}}$, λ^{MAP}, and $\lambda_{i,b}^{\overline{\mathrm{MAP}}}$, $\forall i, b$.

We next define the partial symbol vectors (PSVs) $\mathbf{s}^{(i)} \triangleq [\, s_i \ \cdots \ s_{M_{\mathrm{T}}} \,]^T$ and note that they can be arranged in a tree that has its root just above level $i = M_{\mathrm{T}}$ and leaves, on level $i = 1$, which correspond to symbol vectors $\mathbf{s}$. The binary-valued label vector associated with $\mathbf{s}^{(i)}$ will be denoted by $\mathbf{x}^{(i)}$. The distances[3]

$$d(\mathbf{s}) \triangleq \frac{1}{N_0}\left\|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}\right\|^2 - \log \mathrm{P}[\mathbf{s}] \tag{11}$$

in (7) and (9) can be computed recursively by employing the following factorization

$$\mathrm{P}[\mathbf{s}] = \prod_{i=1}^{M_{\mathrm{T}}} \mathrm{P}\left[s_i \,|\, \mathbf{s}^{(i+1)}\right] \tag{12}$$

with the definition $\mathrm{P}\left[s_{M_{\mathrm{T}}} \,|\, \mathbf{s}^{(M_{\mathrm{T}}+1)}\right] \triangleq \mathrm{P}[s_{M_{\mathrm{T}}}]$. For the sake of simplicity of exposition, we set $\mathrm{P_c}\left[\mathbf{s}^{(i)}\right] \triangleq \mathrm{P}\left[s_i \,|\, \mathbf{s}^{(i+1)}\right]$.

[3]Note that we are abusing common terminology by calling $d(\mathbf{s})$ a "distance", although it does not satisfy the defining properties of a distance function.

We can now rewrite (11) as

$$d(\mathbf{s}) = \sum_{i=1}^{M_{\mathrm{T}}}\left(\frac{1}{N_0}\left|\tilde{y}_i - \sum_{j=i}^{M_{\mathrm{T}}} R_{i,j}s_j\right|^2 - \log \mathrm{P_c}\left[\mathbf{s}^{(i)}\right]\right)$$

which can be evaluated recursively as $d(\mathbf{s}) = d_1$, with the partial distances (PDs)

$$d_i = d_{i+1} + e_i, \quad i = M_{\mathrm{T}}, \ldots, 1, \tag{13}$$

the initialization $d_{M_{\mathrm{T}}+1} = 0$, and the distance increments (DIs)

$$e_i \triangleq \frac{1}{N_0}\left|\tilde{y}_i - \sum_{j=i}^{M_{\mathrm{T}}} R_{i,j}s_j\right|^2 - \log \mathrm{P_c}\left[\mathbf{s}^{(i)}\right]. \tag{14}$$

Note that the DIs are non-negative since the prior terms satisfy $-\log \mathrm{P_c}\left[\mathbf{s}^{(i)}\right] \geq 0$. From (12) and the upper-triangularity of $\mathbf{R}$ it follows that the dependence of the PD d_i on the symbol vector $\mathbf{s}$ is only through the PSV $\mathbf{s}^{(i)}$. Thus, the MAP detection problem and the computation of the intrinsic LLRs $L_{i,b}^{\mathrm{D}}$ have been transformed into a tree-search problem: PSVs and PDs are associated with nodes, branches correspond to DIs. For brevity, we shall often say "the node $\mathbf{s}^{(i)}$" to refer to the node corresponding to the PSV $\mathbf{s}^{(i)}$ on level i. We shall furthermore use $d\big(\mathbf{s}^{(i)}\big)$ and $d\big(\mathbf{x}^{(i)}\big)$ interchangeably to denote d_i. Each path from the root node down to a leaf node corresponds to a symbol vector $\mathbf{s} \in \mathcal{O}^{M_{\mathrm{T}}}$. The quantities λ^{MAP} and $\lambda_{i,b}^{\overline{\mathrm{MAP}}}$ correspond to the leaves associated with the smallest metric in $\mathcal{O}^{M_{\mathrm{T}}}$ and $\mathcal{X}_{i,b}^{\overline{\mathrm{MAP}}}$, respectively. The SISO STS-SD algorithm uses elements of Schnorr-Euchner SD (SESD) [13], [19], briefly summarized as follows: The search in the weighted tree is constrained to nodes that lie within a radius[4] r around $\tilde{\mathbf{y}}$ and tree traversal is performed depth-first, visiting the children of a given node in ascending order of their PDs. A node $\mathbf{s}^{(i)}$ with PD d_i can be pruned (along with the entire subtree originating from this node) whenever the tree-pruning criterion

$$d_i \geq r^2 \tag{15}$$

is satisfied. In the remainder of the paper, (15) is referred to as the "standard pruning criterion."

The radius r has to be chosen sufficiently large for the SD algorithm to find the MAP solution and—in the soft-output case—the QM_{T} counter-hypotheses $\lambda_{i,b}^{\overline{\mathrm{MAP}}}$ in (9) that are required to compute the intrinsic LLRs in (10). On the other hand, choosing r too large, leads to high complexity as a large number of nodes needs to be visited. In order to avoid the problem of choosing a suitable radius r altogether, we employ a technique known as radius reduction [19], which consists of initializing the algorithm with $r = \infty$, and performing the update $r^2 \leftarrow d(\mathbf{s})$ whenever a valid leaf node $\mathbf{s}$ has been found.

The tree-search complexity measure used in the remainder of the paper is the total number of nodes visited by the detector, including the leaf nodes, but excluding the root node and is simply termed as "complexity" from now on. This complexity measure was shown in [20] and [7] to be representative of the hardware complexity of VLSI implementations for hard-output

[4]Note that r corresponds to the radius of a hypersphere if $\mathrm{P}[\mathbf{s}] = 0$.

and soft-output sphere decoding, respectively. Additionally, we will frequently use the term "computational complexity" with the exact meaning made clear in the specific context.

B. Tightening of the Tree-Pruning Criterion

Tightening of the tree-pruning criterion (15), i.e., a reduction of the right-hand side (RHS) of (15), without sacrificing (max-log) optimality is highly desirable as it reduces complexity. Tightening can be accomplished, for example, through techniques based on semi-definite relaxation and H^∞-estimation theory as proposed in [21]. Unfortunately, these approaches entail, in general, a high computational complexity and are, hence, not well-suited for practical (VLSI) implementation.

In the following, we propose an alternative approach which relies on the observation that the DIs (14) contain a —generally non-zero— bias given by

$$b_i \triangleq \min_{\mathbf{s}^{(i)} \in \mathcal{O}^{M_\mathrm{T}+1-i}} e_i, \quad i = 1, \ldots, M_\mathrm{T} \tag{16}$$

where $b_i \geq 0$, by definition. Consider the case where the detector stands at node $\mathbf{s}^{(i)}$ with corresponding PD d_i. The PDs of all leaves that can be reached from the node $\mathbf{s}^{(i)}$ satisfy

$$d_1 \geq d_i + \sum_{j=1}^{i-1} b_j. \tag{17}$$

At level i, we can therefore prune every node that satisfies a tightened version of the tree-pruning criterion in (15), namely

$$d_i \geq r^2 - \sum_{j=1}^{i-1} b_j. \tag{18}$$

We emphasize that tightening of the tree-pruning criterion is either achieved by using the criterion (18) or by replacing e_i in (13) by the "tightened DIs" $e_i - b_i$ and employing the standard tree-pruning criterion (15). Since the tightened DIs are non-negative, by definition, one can show that for initial radius $r = \infty$, the two approaches visit exactly the same set of nodes (in the same order) during the tree search. In the remainder of the paper we focus on the latter approach.

Computation of the bias term (16) requires enumeration of the DIs e_i over all $\mathbf{s}^{(i)} \in \mathcal{O}^{M_\mathrm{T}+1-i}$, which, in general, leads to prohibitive computational complexity. The major portion of this computational complexity is caused by the computation of the Euclidean distance-term $\frac{1}{N_0}\big|\tilde{y}_i - \sum_{j=i}^{M_\mathrm{T}} R_{i,j} s_j\big|^2$ in (14), whose contribution to the bias (16), as it turns out (corresponding numerical results are shown in Section VI-A1), is negligible. Hence, we only consider the contribution to b_i caused by the prior term $-\log \mathrm{P_c}\big[\mathbf{s}^{(i)}\big]$ and we define accordingly the non-negative quantities

$$p_i \triangleq \min_{\mathbf{s}^{(i)} \in \mathcal{O}^{M_\mathrm{T}+1-i}} \big\{ -\log \mathrm{P_c}\Big[\mathbf{s}^{(i)}\Big] \big\}. \tag{19}$$

The corresponding *tightened DIs* are then given by

$$\tilde{e}_i \triangleq e_i - p_i \tag{20}$$

which are non-negative by definition.

1) Statistically independent symbols: In practice, the symbols s_i ($i = 1, \ldots, M_\mathrm{T}$) are often statistically independent, which implies $\mathrm{P}[\mathbf{s}] = \prod_{i=1}^{M_\mathrm{T}} \mathrm{P}[s_i]$. As an important consequence, the computational complexity incurred by the evaluation of the RHS of (19) is drastically reduced (compared to the case of general $\mathrm{P}[\mathbf{s}]$) as we have to carry out one-dimensional searches according to $p_i = \min_{s_i \in \mathcal{O}} \big\{ -\log \mathrm{P}[s_i] \big\}$ only.

We emphasize that using the tightened DIs $\tilde{e}_i$ in (20) preserves max-log optimality and leads, in general, to significant complexity savings, when compared to the standard DIs given in (14), which are widely used in the literature, see, e.g., [5], [6]. To see this, consider the case where all constellation points are equally likely, i.e., $\mathrm{P}[s_i] = |\mathcal{O}|^{-1}$ for all $s_i \in \mathcal{O}$ and $i = 1, \ldots, M_\mathrm{T}$. The corresponding total bias from level i down to the leaf level is given by $\sum_{j=1}^{i-1} p_j = (i-1)\log|\mathcal{O}|$, which can be large, especially for nodes close to the root. Since pruning at and close to the root level, has, in general, significant impact on the number of nodes visited in the tree search, using the tightened DIs in (20) can lead to a major complexity reduction. Corresponding simulation results are provided in Section VI-A2.

2) Statistically independent bits: We have seen above that statistical independence among individual symbols enables us to tighten the tree-pruning criterion at low additional computational complexity. For bit-interleaved coded modulation [22], we not only have independence on the symbol-level, but also the bits $x_{i,b}$ ($i = 1, \ldots, M_\mathrm{T}$, $b = 1, \ldots, Q$) are statistically independent. As shown next, this independence on the bit-level can be exploited to get further reductions in computational complexity. To see this, consider the case where the MIMO detector obtains a priori LLRs $L_{i,b}^\mathrm{A}$ from an external device, e.g., a SISO channel decoder as depicted in Fig. 1. We then have [23]

$$\mathrm{P}[s_i] = \prod_{b:x_{i,b}=+1} \frac{\exp\big(L_{i,b}^\mathrm{A}\big)}{1+\exp\big(L_{i,b}^\mathrm{A}\big)} \prod_{b:x_{i,b}=-1} \frac{1}{1+\exp\big(L_{i,b}^\mathrm{A}\big)}$$

which can be reformulated in more compact form as

$$\mathrm{P}[s_i] = \prod_{b=1}^{Q} \frac{\exp\Big(\frac{1}{2}\big(1+x_{i,b}\big)L_{i,b}^\mathrm{A}\Big)}{1+\exp\big(L_{i,b}^\mathrm{A}\big)}. \tag{21}$$

The contribution of the a priori LLRs to the prior term in the DIs in (14) can then be obtained from (21) as

$$-\log \mathrm{P_c}\Big[\mathbf{s}^{(i)}\Big] = -\log \mathrm{P}[s_i] = \tilde{K}_i - \sum_{b=1}^{Q} \frac{1}{2} x_{i,b} L_{i,b}^\mathrm{A} \tag{22}$$

where the constants

$$\tilde{K}_i \triangleq \sum_{b=1}^{Q} \left(\frac{1}{2}\big|L_{i,b}^\mathrm{A}\big| + \log\Big(1+\exp\big(-\big|L_{i,b}^\mathrm{A}\big|\big)\Big)\right) \tag{23}$$

are independent of the binary-valued variables $x_{i,b}$ and $\tilde{K}_i > 0$ for $i = 1, \ldots, M_\mathrm{T}$.

It then follows from (22) and (23) that

$$p_i = \sum_{b=1}^{Q} \log\Big(1+\exp\big(-\big|L_{i,b}^\mathrm{A}\big|\big)\Big) \tag{24}$$

with the corresponding tightened DIs $\tilde{e}_i = e_i - p_i$ given by

$$\tilde{e}_i \triangleq \frac{1}{N_0}\left|\tilde{y}_i - \sum_{j=i}^{M_\mathrm{T}} R_{i,j}s_j\right|^2 + K_i - \sum_{b=1}^{Q}\frac{1}{2}x_{i,b}L_{i,b}^{\mathrm{A}} \qquad (25)$$

where $K_i \triangleq \sum_{b=1}^{Q}\frac{1}{2}\left|L_{i,b}^{\mathrm{A}}\right|$. We emphasize that using the tightened DIs in (25) avoids computation of transcendental functions such as terms of the form $\log\big(1+\exp(x)\big)$. This is relevant in terms of computational complexity and arithmetic precision in practical (e.g., VLSI) implementations.

Note that in [5, Eq. 9], the prior term (22) was approximated as

$$-\log \mathrm{P}[s_i] \approx \sum_{b=1}^{Q}\frac{1}{2}\Big(\left|L_{i,b}^{\mathrm{A}}\right| - x_{i,b}L_{i,b}^{\mathrm{A}}\Big)$$

for $\left|L_{i,b}^{\mathrm{A}}\right| > 2$ ($b = 1,\ldots,Q$) which results in exactly the same DIs we arrived at in (25). It is important, though, to realize that using the tightened DIs (25) does *not* lead to an approximation of (10), as the neglected $\log(\cdot)$-term does not depend on $x_{i,b}$ and, hence, cancels out in the computation of the intrinsic LLRs $L_{i,b}^{\mathrm{D}}$ according to (10).

III. Extrinsic LLR Computation in a Single Tree Search

Computing the intrinsic LLRs in (10) requires to determine λ^{MAP} and the metrics $\lambda_{i,b}^{\overline{\mathrm{MAP}}}$ associated with the counter-hypotheses. For given i and b, $\lambda_{i,b}^{\overline{\mathrm{MAP}}}$, is obtained by traversing only those parts of the search tree that have leaves in $\mathcal{X}_{i,b}^{\overline{\mathrm{MAP}}}$. The quantities λ^{MAP} and $\lambda_{i,b}^{\overline{\mathrm{MAP}}}$ can, in principle, be computed using the sphere decoder based on the repeated tree-search (RTS) approach described in [17]. The RTS strategy results, however, in redundant computations as (often significant) parts of the search tree are revisited during the RTS steps required to determine $\lambda_{i,b}^{\overline{\mathrm{MAP}}}$ for all i,b. Following the STS paradigm described for soft-output SD in [7], we note that *efficient* computation of the LLRs $L_{i,b}^{\mathrm{D}}$ requires that every node in the tree be visited *at most* once. This can be achieved by searching for the MAP solution and computing the metrics $\lambda_{i,b}^{\overline{\mathrm{MAP}}}$ concurrently while ensuring that the subtree originating from a given node in the tree is pruned if searching that subtree can not lead to an update of either λ^{MAP} or at least one of the $\lambda_{i,b}^{\overline{\mathrm{MAP}}}$. Besides extending the ideas in [7] to take into account a priori information, the main idea underlying SISO STS-SD presented in this paper is to *directly* compute the extrinsic LLRs $L_{i,b}^{\mathrm{E}}$ through a tree search, rather than computing $L_{i,b}^{\mathrm{D}}$ first and then evaluating $L_{i,b}^{\mathrm{E}} = L_{i,b}^{\mathrm{D}} - L_{i,b}^{\mathrm{A}}$.

Due to the large dynamic range of LLRs, fixed-point detector implementations need to constrain the magnitude of the LLR values. Evidently, clipping of the LLR magnitude leads to a performance degradation. It was demonstrated in [7] that LLR clipping (when built into the tree search) allows to tune the soft-output STS-SD algorithm in terms of complexity versus performance by adjusting the clipping parameter. In the SISO case, we are ultimately interested in the *extrinsic* LLRs $L_{i,b}^{\mathrm{E}}$ and clipping should therefore ensure that $\left|L_{i,b}^{\mathrm{E}}\right| \leq L_{\max}$, $\forall i,b$, where $L_{\max}$ is the LLR clipping parameter. It is therefore sensible to ask whether clipping of the *extrinsic* LLRs can be built directly into the tree search. The answer is in the affirmative and a corresponding solution is described below. We start by writing the extrinsic LLRs as

$$L_{i,b}^{\mathrm{E}} = \begin{cases} \Lambda_{i,b}^{\overline{\mathrm{MAP}}} - \lambda^{\mathrm{MAP}}, & x_{i,b}^{\mathrm{MAP}} = +1 \\ \lambda^{\mathrm{MAP}} - \Lambda_{i,b}^{\overline{\mathrm{MAP}}}, & x_{i,b}^{\mathrm{MAP}} = -1 \end{cases} \qquad (26)$$

where the quantities

$$\Lambda_{i,b}^{\overline{\mathrm{MAP}}} \triangleq \begin{cases} \lambda_{i,b}^{\overline{\mathrm{MAP}}} - L_{i,b}^{\mathrm{A}}, & x_{i,b}^{\mathrm{MAP}} = +1 \\ \lambda_{i,b}^{\overline{\mathrm{MAP}}} + L_{i,b}^{\mathrm{A}}, & x_{i,b}^{\mathrm{MAP}} = -1 \end{cases} \qquad (27)$$

will be referred to as the *extrinsic metrics*. For the following developments it will be convenient to define the function $f(\cdot)$ that transforms an intrinsic metric λ with associated a priori LLR L^{A} and binary label x to an extrinsic metric Λ according to

$$\Lambda = f\big(\lambda, L^{\mathrm{A}}, x\big) \triangleq \begin{cases} \lambda - L^{\mathrm{A}}, & x = +1 \\ \lambda + L^{\mathrm{A}}, & x = -1. \end{cases} \qquad (28)$$

With this notation, we can rewrite (27) more compactly as $\Lambda_{i,b}^{\overline{\mathrm{MAP}}} = f\Big(\lambda_{i,b}^{\overline{\mathrm{MAP}}}, L_{i,b}^{\mathrm{A}}, x_{i,b}^{\mathrm{MAP}}\Big)$. The inverse function of (28) transforms an extrinsic metric Λ to an intrinsic metric λ and is given by

$$\lambda = f^{-1}\big(\Lambda, L^{\mathrm{A}}, x\big) \triangleq \begin{cases} \Lambda + L^{\mathrm{A}}, & x = +1 \\ \Lambda - L^{\mathrm{A}}, & x = -1. \end{cases} \qquad (29)$$

We emphasize that the tree-search algorithm described in the following produces the extrinsic LLRs $L_{i,b}^{\mathrm{E}}$ in (26) rather than the intrinsic ones in (10). Since the soft-output STS-SD algorithm described in [7] delivers $L_{i,b}^{\mathrm{D}}$ and we have $L_{i,b}^{\mathrm{E}} = L_{i,b}^{\mathrm{D}}$ only if $L_{i,b}^{\mathrm{A}} = 0$, i.e., for uniform priors, careful modifications of the list administration steps, the tree-pruning criterion, and the LLR clipping rules are needed.

A. List Administration

The main idea of the SISO STS paradigm is to search the subtree originating from a given node only if the result can lead to an update of either λ^{MAP} or of at least one of the $\Lambda_{i,b}^{\overline{\mathrm{MAP}}}$. To this end, the algorithm needs to maintain a list containing the current MAP hypothesis $\mathbf{x}^{\mathrm{MAP}}$, the corresponding metric λ^{MAP}, and all QM_T extrinsic metrics $\Lambda_{i,b}^{\overline{\mathrm{MAP}}}$. This list is initialized with[5] $\lambda^{\mathrm{MAP}} = \Lambda_{i,b}^{\overline{\mathrm{MAP}}} = \infty$, $\forall i,b$, and $x_{i,b}^{\mathrm{MAP}} = 1$, $\forall i,b$. Whenever a leaf node with corresponding label $\mathbf{x}$ has been reached, the detector distinguishes between two cases:

i) MAP hypothesis update: If $d(\mathbf{x}) < \lambda^{\mathrm{MAP}}$, a new MAP hypothesis has been found. First, all extrinsic metrics $\Lambda_{i,b}^{\overline{\mathrm{MAP}}}$ for which $x_{i,b} = x_{i,b}^{\overline{\mathrm{MAP}}}$ are updated according to

$$\Lambda_{i,b}^{\overline{\mathrm{MAP}}} \leftarrow f\Big(\lambda^{\mathrm{MAP}}, L_{i,b}^{\mathrm{A}}, x_{i,b}^{\overline{\mathrm{MAP}}}\Big)$$

followed by the updates $\lambda^{\mathrm{MAP}} \leftarrow d(\mathbf{x})$ and $\mathbf{x}^{\mathrm{MAP}} \leftarrow \mathbf{x}$. In other words, for each bit in the MAP hypothesis that is changed in the update process, the metric associated with the *former* MAP hypothesis becomes the extrinsic metric of the *new* counter-hypothesis.

[5] In practical implementations, λ^{MAP} and the $\Lambda_{i,b}^{\overline{\mathrm{MAP}}}$ are initialized with the largest quantity that can be represented in the given number format.

ii) Extrinsic metric update: In the case where $d(\mathbf{x}) > \lambda^{\mathrm{MAP}}$, only extrinsic metrics corresponding to counter-hypotheses might be updated. For each pair (i,b) with $x_{i,b} = \overline{x_{i,b}^{\mathrm{MAP}}}$ and $f\Big(d(\mathbf{x}), L_{i,b}^{\mathrm{A}}, x_{i,b}^{\mathrm{MAP}}\Big) < \Lambda_{i,b}^{\overline{\mathrm{MAP}}}$, the SISO STS-SD algorithm performs the update

$$\Lambda_{i,b}^{\overline{\mathrm{MAP}}} \leftarrow f\big(d(\mathbf{x}), L_{i,b}^{\mathrm{A}}, x_{i,b}^{\mathrm{MAP}}\big)\,. \tag{30}$$

The condition $f\Big(d(\mathbf{x}), L_{i,b}^{\mathrm{A}}, x_{i,b}^{\mathrm{MAP}}\Big) < \Lambda_{i,b}^{\overline{\mathrm{MAP}}}$ ensures that the value of the extrinsic metric $\Lambda_{i,b}^{\overline{\mathrm{MAP}}}$ is, indeed, reduced in the update process.

B. Extrinsic LLR Clipping

In order to ensure that the extrinsic LLRs delivered by the algorithm indeed satisfy $\big|L_{i,b}^{E}\big| \le L_{\max}$, $\forall i,b$, the following update rule

$$\Lambda_{i,b}^{\overline{\mathrm{MAP}}} \leftarrow \max\Bigg\{\lambda^{\mathrm{MAP}} - L_{\max}, \min\Big\{\Lambda_{i,b}^{\overline{\mathrm{MAP}}}, \lambda^{\mathrm{MAP}} + L_{\max}\Big\}\Bigg\},\ \forall i,b \tag{31}$$

has to be applied after carrying out the steps in Case i) of the list administration procedure described in Section III-A. Note that for $L_{\max} = \infty$ the detector attains max-log optimal SISO performance, whereas for $L_{\max} = 0$, the LLRs satisfy $L_{i,b}^{\mathrm{E}} = 0$ and the hard-output MAP solution (8) is obtained. In Section VI-B1, it will be demonstrated (numerically) that LLR clipping enables efficient tuning of the performance and complexity of the SISO STS-SD algorithm.

We note that for $L_{i,b}^{\mathrm{A}} = 0$, $\forall i,b$, the update rule (31) reduces to the corresponding update rule [7, Eq. 13] for the soft-output STS-SD algorithm given by

$$\Lambda_{i,b}^{\overline{\mathrm{MAP}}} \leftarrow \min\Big\{\Lambda_{i,b}^{\overline{\mathrm{MAP}}}, \lambda^{\mathrm{MAP}} + L_{\max}\Big\}, \quad \forall i,b. \tag{32}$$

This can be seen by noting that $L_{i,b}^{\mathrm{A}} = 0$, $\forall i,b$, implies that $\Lambda_{i,b}^{\overline{\mathrm{MAP}}} = \lambda_{i,b}^{\overline{\mathrm{MAP}}} \ge \lambda^{\mathrm{MAP}}$, $\forall i,b$, and hence, we have

$$\lambda^{\mathrm{MAP}} - L_{\max} \le \min\Big\{\Lambda_{i,b}^{\overline{\mathrm{MAP}}}, \lambda^{\mathrm{MAP}} + L_{\max}\Big\}, \quad \forall i,b.$$

Note that the computational complexity associated with the evaluation of the RHS of (32) is lower than that incurred by the evaluation of the RHS of (31). Specifically, half of the required compare operations are saved. Numerical results have shown that using the simpler LLR clipping rule (32) in the *soft-input* case entails virtually no difference in terms of performance and complexity. We stress, however, that clipping according to (32) in the soft-input case no longer guarantees that $\big|L_{i,b}^{E}\big| \le L_{\max}$, $\forall i,b$, so that an additional clipping step after the computation of (26) is required. In the presence of run-time constraints this additional clipping step is required anyways (see Section V-C). In the remainder of the paper, we exclusively use the LLR clipping rule (32) and refer to it as "the extrinsic LLR clipping rule."

C. The Tree-Pruning Criterion

Consider the node $\mathbf{s}^{(i)}$ corresponding to the label bits $x_{j,b}$ ($j = i,\dots,M_{\mathrm{T}}$, $b = 1,\dots,Q$). Assume that the subtree originating from this node and corresponding to the label bits $x_{j,b}$ ($j = 1,\dots,i-1$, $b = 1,\dots,Q$) has not been expanded yet. The tree-pruning criterion for the node $\mathbf{s}^{(i)}$ along with its subtree is compiled from two sets, defined as follows:

1) The bits in the partial label $\mathbf{x}^{(i)}$ corresponding to the node $\mathbf{s}^{(i)}$ are compared to the corresponding bits in the label of the current MAP hypothesis. All extrinsic metrics $\Lambda_{i,b}^{\overline{\mathrm{MAP}}}$, with $x_{i,b} = \overline{x_{i,b}^{\mathrm{MAP}}}$, found in this comparison may be affected when searching the subtree originating from $\mathbf{s}^{(i)}$. As the PD $d\big(\mathbf{x}^{(i)}\big)$ is an intrinsic metric, the extrinsic metrics $\Lambda_{i,b}^{\overline{\mathrm{MAP}}}$ need to be transformed to intrinsic metrics according to (29). The resulting set of *intrinsic* metrics, which may be affected by an update, is given by

$$\mathcal{A}_1\Big(\mathbf{x}^{(i)}\Big) \triangleq \Big\{ f^{-1}\Big(\Lambda_{j,b}^{\overline{\mathrm{MAP}}}, L_{j,b}^{\mathrm{A}}, x_{j,b}^{\mathrm{MAP}}\Big) \,\Big|\, \big(j \ge i, \forall b\big) \wedge \Big(x_{j,b} = \overline{x_{j,b}^{\mathrm{MAP}}}\Big)\Big\}.$$

2) The extrinsic metrics $\Lambda_{j,b}^{\overline{\mathrm{MAP}}}$ for $j = 1,\dots,i-1$, $b = 1,\dots,Q$ corresponding to the counter-hypotheses in the subtree originating from $\mathbf{s}^{(i)}$ may be affected as well. Correspondingly, we define

$$\mathcal{A}_2\Big(\mathbf{x}^{(i)}\Big) \triangleq \Big\{ f^{-1}\Big(\Lambda_{j,b}^{\overline{\mathrm{MAP}}}, L_{j,b}^{\mathrm{A}}, x_{j,b}^{\mathrm{MAP}}\Big) \,\Big|\, j < i, \forall b\Big\}.$$

The intrinsic metrics which may be affected during the search in the subtree originating from $\mathbf{s}^{(i)}$ are now given by $\mathcal{A}\big(\mathbf{x}^{(i)}\big) \triangleq \mathcal{A}_1\big(\mathbf{x}^{(i)}\big) \cup \mathcal{A}_2\big(\mathbf{x}^{(i)}\big)$. The node $\mathbf{s}^{(i)}$ along with its subtree is pruned if the corresponding PD $d\big(\mathbf{x}^{(i)}\big)$ satisfies the tree-pruning criterion

$$d\big(\mathbf{x}^{(i)}\big) > \max_{a \in \mathcal{A}\left(\mathbf{x}^{(i)}\right)} a.$$

This tree-pruning criterion ensures that the subtree originating from a given node is explored *only* if this could lead to an update of either λ^{MAP} or of at least one of the extrinsic metrics $\Lambda_{i,b}^{\overline{\mathrm{MAP}}}$. Note that λ^{MAP} does not appear in the set $\mathcal{A}\big(\mathbf{x}^{(i)}\big)$, as the update rules *i)* and *ii)* specified in Section III-A ensure that λ^{MAP} is always smaller than or equal to all intrinsic metrics associated with the counter-hypotheses.

IV. Channel-Matrix Preprocessing

In this section, we describe how performing the QR-decomposition (QRD) on a column-sorted and regularized version of the channel matrix $\mathbf{H}$ in combination with compensation of self-interference in the LLRs—caused by channel-matrix regularization—carried out directly in the tree search can result in a significant complexity reduction at negligible performance loss. The use of column-sorting and regularization for soft-output SD was discussed in detail in [7]. We shall therefore keep the discussion of the general aspects short and put emphasis on self-interference compensation.

A. Column-Sorting and Regularization of the Channel Matrix

Methods for column-sorting and regularization of the channel matrix $\mathbf{H}$ performed on the basis of the received symbol vector $\mathbf{y}$ were discussed, e.g., in [24], [25]. Unfortunately, these techniques require QRD at symbol-vector rate, which leads to a significant computational burden. In contrast, column-sorting and regularization based solely on the channel matrix $\mathbf{H}$ (and possibly on the noise variance) require QRDs only when the channel state changes, which entails a significantly smaller computational burden. In the following, we restrict ourselves to methods of the latter type.

1) Column-sorting: The complexity of SD can be reduced (often significantly) by performing the QRD on a column-sorted version of $\mathbf{H}$ rather than on $\mathbf{H}$ directly, i.e., by computing $\mathbf{HP} = \mathbf{QR}$, where $\mathbf{P}$ is an $M_{\mathrm{T}} \times M_{\mathrm{T}}$ permutation matrix. Reduction in terms of complexity is obtained if levels closer to the root correspond to main-diagonal entries of $\mathbf{R}$ with larger magnitude, or equivalently, to spatial streams with higher effective SNR. A corresponding computationally efficient heuristic for obtaining $\mathbf{P}$ was proposed in [26] and is referred to as sorted QRD (SQRD) in the following. We emphasize that column-sorting does *not* entail a performance degradation and the additional computational complexity required by SQRD (as compared to QRD) is negligible [27].

2) Regularization: A further reduction in terms of complexity —at the cost of slightly reduced performance— can be obtained by performing the tree search on a Tikhonov-regularized (and column-sorted) version of $\mathbf{H}$ according to [28]

$$\underbrace{\begin{bmatrix} \mathbf{H} \\ \alpha \mathbf{I}_{M_{\mathrm{T}}} \end{bmatrix}}_{\overline{\mathbf{H}}} \mathbf{P} = \underbrace{\begin{bmatrix} \mathbf{Q}_a & \mathbf{Q}_c \\ \mathbf{Q}_b & \mathbf{Q}_d \end{bmatrix}}_{\overline{\mathbf{Q}}} \underbrace{\begin{bmatrix} \widetilde{\mathbf{R}} \\ \mathbf{0}_{M_{\mathrm{R}} \times M_{\mathrm{T}}} \end{bmatrix}}_{\overline{\mathbf{R}}} \tag{33}$$

where $\alpha \in \mathbb{R}$ is a suitably chosen regularization parameter. Here, the $M_{\mathrm{T}} \times M_{\mathrm{T}}$ matrix $\widetilde{\mathbf{R}}$ is upper-triangular, $\overline{\mathbf{Q}}$ is unitary, and $\mathbf{Q}_a$, $\mathbf{Q}_b$, $\mathbf{Q}_c$, and $\mathbf{Q}_d$ are of dimension $M_{\mathrm{R}} \times M_{\mathrm{T}}$, $M_{\mathrm{T}} \times M_{\mathrm{T}}$, $M_{\mathrm{R}} \times M_{\mathrm{R}}$, and $M_{\mathrm{T}} \times M_{\mathrm{R}}$, respectively. Note that the computational complexity for regularized SQRD is approximately 50% higher than that for non-regularized SQRD [27], [29]. However, the QRD needs to be performed only if the channel matrix $\mathbf{H}$ changes, as opposed to the tree-search itself, which needs to be carried out at symbol-vector rate. Therefore, the computational complexity increase incurred by channel-matrix regularization is —in most practical applications— not critical.

For regularized SQRD, the input-output relation in (5) is replaced by

$$\hat{\mathbf{y}} = \widetilde{\mathbf{R}}\tilde{\mathbf{s}} + \tilde{\mathbf{n}} \tag{34}$$

where $\hat{\mathbf{y}} \triangleq \mathbf{Q}_a^H \mathbf{y}$, $\tilde{\mathbf{s}} \triangleq \mathbf{P}^T \mathbf{s}$, and $\tilde{\mathbf{n}} \triangleq -\alpha \mathbf{Q}_b^H \mathbf{s} + \mathbf{Q}_a^H \mathbf{n}$. The corresponding intrinsic (max-log) LLRs in (6) are obtained by pretending that the resulting noise $\tilde{\mathbf{n}}$ has the same statistics as $\mathbf{n}$, which leads to

$$\tilde{L}_{i,b}^{\mathrm{D}} \triangleq \min_{\tilde{\mathbf{s}} \in \mathcal{X}_{i,b}^{(-1)}} \left\{ \frac{1}{N_0} \left\| \hat{\mathbf{y}} - \widetilde{\mathbf{R}}\tilde{\mathbf{s}} \right\|^2 - \log \mathrm{P}[\tilde{\mathbf{s}}] \right\} - \min_{\tilde{\mathbf{s}} \in \mathcal{X}_{i,b}^{(+1)}} \left\{ \frac{1}{N_0} \| \hat{\mathbf{y}} - \widetilde{\mathbf{R}}\tilde{\mathbf{s}} \|^2 - \log \mathrm{P}[\tilde{\mathbf{s}}] \right\}. \tag{35}$$

The intrinsic LLRs $\tilde{L}_{i,b}^{\mathrm{D}}$ in (35) will, in general, only be approximations to the true intrinsic LLRs $L_{i,b}^{\mathrm{D}}$ in (6). This is a consequence of $\tilde{\mathbf{n}}$ not being i.i.d. circularly symmetric complex Gaussian distributed with variance N_0 per complex entry, as was assumed to arrive at (35). Specifically, $\tilde{\mathbf{n}}$ contains self-interference (i.e., it depends on $\mathbf{s}$) and $\mathbf{Q}_a$ is, in general, not unitary. Setting $\alpha \triangleq \sqrt{N_0/\mathbb{E}[|s|^2]}$ leads to the so-called minimum mean-square error[6] (MMSE) SQRD [30], which was shown in [7] to result in a good performance/complexity tradeoff for soft-output STS-SD. In the remainder of this paper, regularization will always refer to using MMSE-SQRD. Finally, we note that the LLRs in (35) need to be reordered after the detection stage to account for the permutation induced by $\mathbf{P}$.

B. Compensation of Self-Interference

Note that for $\alpha \neq 0$, the (max-log) LLRs in (35) approximate the intrinsic (max-log) LLRs in (6), resulting in a performance degradation. In order to recover (part of) this performance loss, a method for the compensation of self-interference was developed in [31] for list-based MIMO detectors. The approach described in [31] can not be applied directly to SISO STS-SD. It turns out, however, that compensation of self-interference can be incorporated into the tree-search. This leads to a noticeable performance improvement compared to using (35) directly, while the corresponding increase in complexity is negligible (corresponding numerical results are shown in Section VI-B3).

1) Compensation of self-interference: As shown in [31], the squared Euclidean distance $\left\| \overline{\mathbf{y}} - \overline{\mathbf{H}}\mathbf{s} \right\|^2$ with the vector $\overline{\mathbf{y}} \triangleq \begin{bmatrix} \mathbf{y}^T & \mathbf{0}_{1 \times M_{\mathrm{T}}} \end{bmatrix}^T$ can be expanded in two different ways according to

$$\left\| \overline{\mathbf{y}} - \overline{\mathbf{H}}\mathbf{s} \right\|^2 = \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 + \alpha^2 \|\mathbf{s}\|^2 \tag{36}$$

and

$$\begin{aligned} \left\| \overline{\mathbf{y}} - \overline{\mathbf{H}}\mathbf{s} \right\|^2 &= \left\| \overline{\mathbf{Q}}^H \overline{\mathbf{y}} - \overline{\mathbf{R}}\mathbf{P}^T \mathbf{s} \right\|^2 \\ &= \left\| \hat{\mathbf{y}} - \widetilde{\mathbf{R}}\tilde{\mathbf{s}} \right\|^2 + \left\| \mathbf{Q}_c^H \mathbf{y} \right\|^2 \end{aligned} \tag{37}$$

where (37) is obtained by using (33). Equating the RHS terms of (36) and (37) and using $\|\mathbf{s}\|^2 = \|\tilde{\mathbf{s}}\|^2$ yields

$$\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 = \left\| \hat{\mathbf{y}} - \widetilde{\mathbf{R}}\tilde{\mathbf{s}} \right\|^2 + \left\| \mathbf{Q}_c^H \mathbf{y} \right\|^2 - \alpha^2 \|\tilde{\mathbf{s}}\|^2 \tag{38}$$

which allows us to conclude that the metric $\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2$ contains a contribution that is independent of the symbol vectors, namely $\|\mathbf{Q}_c^H \mathbf{y}\|^2$, and a term caused by self-interference given by $-\alpha^2 \|\tilde{\mathbf{s}}\|^2$. Since we use $\left\| \hat{\mathbf{y}} - \widetilde{\mathbf{R}}\tilde{\mathbf{s}} \right\|^2$ (instead of the left-hand side of (38)) in the LLR computation (35), the two remaining RHS-terms in (38) must be compensated for. As already observed in Section II-B2, constant terms (i.e., terms that are independent of $\mathbf{s}$) cancel out in the LLR computation (10) so that the term $\left\| \mathbf{Q}_c^H \mathbf{y} \right\|^2$ in (38) can be neglected without affecting the resulting LLRs. However, the term $-\alpha^2 \|\tilde{\mathbf{s}}\|^2$ does depend on $\mathbf{s}$ and therefore needs to be compensated for. This

[6] Strictly speaking, the name MMSE-SQRD is justified only in the case of $\mathbb{E}[|s_i|^2] = \mathbb{E}[|s|^2], \forall i$, as otherwise this approach will not deliver the MMSE estimate of $\mathbf{s}$ based on the observation $\mathbf{y}$.

is accomplished by computing the *self-interference free* (SIF) intrinsic max-log LLRs according to [31]

$$\bar{L}^{\mathrm{D}}_{i,b} \triangleq \min_{\tilde{\mathbf{s}}\in\mathcal{X}^{(-1)}_{i,b}} \left\{ \frac{1}{N_0}\big\|\hat{\mathbf{y}} - \widetilde{\mathbf{R}}\tilde{\mathbf{s}}\big\|^2 - \frac{\alpha^2}{N_0}\|\tilde{\mathbf{s}}\|^2 - \log \mathrm{P}[\tilde{\mathbf{s}}] \right\} - \min_{\tilde{\mathbf{s}}\in\mathcal{X}^{(+1)}_{i,b}} \left\{ \frac{1}{N_0}\big\|\hat{\mathbf{y}} - \widetilde{\mathbf{R}}\tilde{\mathbf{s}}\big\|^2 - \frac{\alpha^2}{N_0}\|\tilde{\mathbf{s}}\|^2 - \log \mathrm{P}[\tilde{\mathbf{s}}] \right\}. \tag{39}$$

We emphasize, however, that (39) remains an approximation to (6) as the noise term $\tilde{\mathbf{n}}$ in (34) is *not* i.i.d. circularly symmetric Gaussian distributed with variance N_0 per complex entry, as was assumed to arrive at (35).

2) Compensation of self-interference in the SISO STS-SD algorithm: In [31] it was suggested to compensate self-interference *after* the tree-search has been carried out. In the present case, this would require explicit knowledge of the symbol vectors $\tilde{\mathbf{s}}$ that attain the minima in (39). These symbol vectors are, however, not required to be kept track of in the SISO STS-SD algorithm, as the extrinsic LLRs are computed on the basis of the MAP hypothesis $\mathbf{x}^{\mathrm{MAP}}$, its metric λ^{MAP}, and the extrinsic metrics $\Lambda^{\overline{\mathrm{MAP}}}_{i,b}$ only (see (26)). Inspection of (39) suggests, however, that self-interference compensation may be incorporated into the tree-search procedure, which would avoid keeping track of the minimizing symbol vectors $\tilde{\mathbf{s}}$ in (39). Straightforward modification of the DIs in (25) to accomplish this would lead to the modified DIs $e_i - \frac{\alpha^2}{N_0}|\tilde{s}_i|^2$ which are, however, not guaranteed to be non-negative. As in the tightening of the tree-pruning criterion described in Section II-B, we recognize that symbol-vector-independent terms can be added to the DIs without loss of (max-log) optimality. Therefore, setting the DIs to

$$\bar{e}_i \triangleq e_i + m(\tilde{s}_i) \tag{40}$$

with the non-negative term

$$m(\tilde{s}_i) \triangleq \frac{\alpha^2}{N_0}\left(\max_{s\in\mathcal{O}} |s|^2 - |\tilde{s}_i|^2 \right) \tag{41}$$

leads to the smallest possible non-negative DIs that compensate self-interference directly in the tree search. Note that adding non-negative terms to the DIs as done in (40), in general, increases the (tree-search) complexity. On the other hand, channel-matrix regularization almost always significantly reduces (tree-search) complexity [7] and the increase (caused by adding the term $m(\tilde{s}_i)$), is marginal, as shown numerically in Section VI-B3. In addition, it turns out that self-interference compensation recovers the performance loss due to channel-matrix regularization almost entirely so that near-max-log optimal performance is achieved (see the numerical results in Section VI-B3). In the case of constant-modulus symbol alphabets (e.g., BPSK or 4-QAM) we have $m(\tilde{s}_i) = 0$, $i = 1, \ldots, M_{\mathrm{T}}$, and hence compensation of self-interference in the tree-search (as described above) is not necessary. We conclude by noting that the quantities $\max_{s\in\mathcal{O}}|s|^2$ can be pre-computed so that the additional computational complexity required to incorporate compensation of self-interference into the tree-search procedure is small.

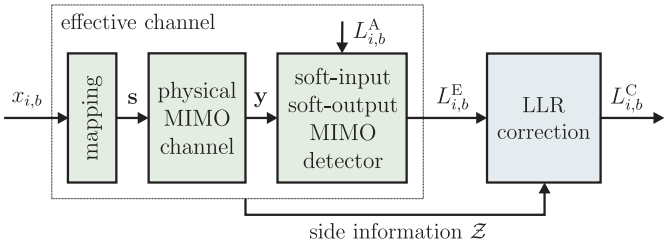


Fig. 2. LLR correction post-processes the LLRs resulting from the effective channel using side information $\mathcal{Z}$.

V. LLR Correction

The max-log approximation, channel-matrix regularization, and other complexity-reducing mechanisms, such as early termination of the tree-search [7], lead to LLRs that are approximations to the true LLRs in (2). However, channel decoders (see Fig. 1) rely on exact LLRs to achieve optimum performance. In the following, we present a post-processing method for correcting approximate LLRs resulting from suboptimal detectors. This method is based on ideas developed in [32] and [33] and is able to (often significantly) improve the performance of (iterative) MIMO decoding while requiring low additional computational complexity.

A. The Basic Idea

We start by defining (or recalling the definitions of) the following objects (see Fig. 2):

- the *effective channel* with the binary-valued inputs $x_{i,b}$ and outputs given by the (possibly approximated) extrinsic LLRs $L^{\mathrm{E}}_{i,b}$.
- the *physical MIMO channel* with input $\mathbf{s}$ and output $\mathbf{y}$.
- the *soft-input soft-output MIMO detector* with inputs $\mathbf{y}$ and $L^{\mathrm{A}}_{i,b}$ and outputs $L^{\mathrm{E}}_{i,b}$.
- the *LLR correction unit* (see Fig. 2) computes corrected extrinsic LLRs $L^{\mathrm{C}}_{i,b}$ based on (approximated) extrinsic LLRs $L^{\mathrm{E}}_{i,b}$ and on side information $\mathcal{Z}$, by applying an LLR correction function

$$L^{\mathrm{C}}_{i,b} = g\Big(L^{\mathrm{E}}_{i,b}, \mathcal{Z}\Big). \tag{42}$$

- the *side information* $\mathcal{Z}$ is, for example, obtained from the (instantaneous) receive SNR, the singular values of the channel matrix $\mathbf{H}$, and/or from knowledge of whether the soft-input soft-output MIMO detector was terminated prematurely [7].

The choice for the LLR correction function in (42) is motivated by drawing the following parallel. The SISO detector considered in Section II computes (intrinsic) LLRs for all transmitted bits $x_{i,b}$ according to (2), based on the observation $\mathbf{y}$ and the side information $\mathbf{H}$. The LLR correction function (42) should implement a soft-output detector for the effective channel in Fig. 2. Hence, LLR correction amounts to computing the log-likelihood function for the bits $x_{i,b}$, based on the observation $L^{\mathrm{E}}_{i,b}$ (instead of $\mathbf{y}$) and the side information

$\mathcal{Z}$ (instead of $\mathbf{H}$) according to

$$g\Big(L_{i,b}^{\mathrm{E}}, \mathcal{Z}\Big) \triangleq \log\left(\frac{\mathrm{P}\Big[x_{i,b}=+1 \,|\, L_{i,b}^{\mathrm{E}}, \mathcal{Z}\Big]}{\mathrm{P}\Big[x_{i,b}=-1 \,|\, L_{i,b}^{\mathrm{E}}, \mathcal{Z}\Big]}\right). \tag{43}$$

Note that this choice for the correction function also ensures that the resulting (corrected) LLRs $L_{i,b}^{\mathrm{C}}$ are, indeed, valid LLRs. The formulation (42) together with (43) entails that for a given pair (i,b) $L_{i,b}^{\mathrm{C}}$ depends only on $L_{i,b}^{\mathrm{E}}$ and $\mathcal{Z}$. Making the correction function $g(\cdot)$ depend on other LLR values, besides the one to be corrected, would certainly improve the correction performance, but at the same time also dramatically increase the computational effort for LLR correction (for computation of the LLR correction function as well as for carrying out the LLR correction itself). This will become clear when the numerical procedure for LLR correction is described below.

The main idea is now, depending on the mechanisms used to approximate the extrinsic LLRs (e.g., the max-log approximation, channel-matrix regularization, early termination of the tree-search), to extract suitable side information $\mathcal{Z}$. To see that this is non-trivial and the problem is multi-faceted, simply note that the set of all possible channel matrices $\mathbf{H}$ is a continuum of $M_{\mathrm{R}} \times M_{\mathrm{T}}$ complex-valued matrices. This continuum will be absorbed in $\mathcal{Z}$ through, e.g., the singular values or the rank of $\mathbf{H}$. We emphasize that we will require the set $\mathcal{Z}$ to be finite. In addition, we will need the individual entries of $\mathcal{Z}$ to have finite cardinality. Hence, continuous-valued quantities, such as, e.g., the SNR or singular values, must be suitably quantized. The total number of different instances of the side information $\mathcal{Z}$ is denoted by Z in the following.

B. Computation of the LLR Correction Function

Once we have chosen $\mathcal{Z}$, the LLR correction function is seen in (43) to be obtained from the conditional probabilities $\mathrm{P}\Big[x_{i,b}=\pm 1 \,|\, L_{i,b}^{\mathrm{E}}, \mathcal{Z}\Big]$. Analytical expressions for correction functions seem very hard to obtain (even for simple examples such as for Hagenauer's approximation to the box function [32]). We next propose an approach for numerically computing (approximations to) the LLR correction function in (43).

First, the range of the LLRs to be corrected needs to be constrained, say to $L_{i,b}^{\mathrm{E}} \in [-L_{\max}, +L_{\max}]$ (e.g., by performing LLR clipping in the detector). This interval is then divided into K equally-sized bins such that the kth bin corresponds to

$$\mathcal{B}_k = \left[-L_{\max} + k\frac{2L_{\max}}{K}, -L_{\max} + (k+1)\frac{2L_{\max}}{K}\right)$$

for $k = 0, \ldots, K-1$. Then, the histogram

$$p_k(\mathcal{Z}) \triangleq \mathrm{P}\big[x_{i,b}=+1 \,|\, L_{i,b}^{\mathrm{E}} \in \mathcal{B}_k, \mathcal{Z}\big], \quad \forall k \tag{44}$$

is computed by performing Monte-Carlo simulations (averaged over noise and channel realizations) with randomly generated bits $x_{i,b}$. For each $L_{i,b}^{\mathrm{E}}$ and given instance of $\mathcal{Z}$, the (approximated) LLR correction function is obtained by linear interpolation between the base-points

$$\left(-L_{\max} + \left(k+\frac{1}{2}\right)\frac{2L_{\max}}{K}\,,\, \log\left(\frac{p_k(\mathcal{Z})}{1-p_k(\mathcal{Z})}\right)\right) \tag{45}$$

where (x,y) denotes the x- and y-coordinates of the base-points. We emphasize that for each instance of $\mathcal{Z}$, in general, a different LLR correction function is obtained. Note that the LLRs resulting from (45) can have a magnitude that is larger than $L_{\max}$ (see Section VI-D) so that an additional LLR-clipping step to limit the dynamic range (to a clipping level possibly larger than $L_{\max}$) may be necessary.

The computational complexity incurred by evaluating the histogram (44) and the corresponding storage requirements depend critically on the number of bins K and on Z. In particular, ZK histogram values need to be stored and hence, it is important to keep ZK small. Application of the LLR correction function itself amounts to simple table look-up operations followed by linear interpolation, which can be performed at very low computational complexity.

C. An Example

We next provide an example that illustrates the potential impact of LLR correction. The complexity of the SISO STS-SD algorithm depends critically on the noise realization $\mathbf{n}$, the channel-matrix realization $\mathbf{H}$, the transmit-vector $\mathbf{s}$, and the a priori LLRs $L_{i,b}^{\mathrm{A}}$. The often prohibitively high worst-case complexity of SD (see, e.g., [34, Section III-C]), constitutes a problem in many practical application scenarios, as it inhibits meeting the throughput requirements of many of the available communication standards. A promising approach to limiting the worst-case complexity of SD, while keeping the resulting performance degradation small, was proposed in [35], [7]. The basic idea is to impose an aggregate complexity constraint of ND_{avg} visited nodes for a block of N symbol vectors by using maximum-first (MF) scheduling. This scheduling strategy allocates the overall complexity budget (in terms of the number of visited nodes in the tree search) according to

$$D_{\max}[j] \triangleq ND_{\mathrm{avg}} - \sum_{\ell=1}^{j-1} D[\ell] - (N-j)M \tag{46}$$

for $j = 1, \ldots, N$, where M is a parameter to be specified below, $D[\ell]$ is the actual number of nodes visited in the detection of the ℓth symbol vector, and $D_{\max}[j]$ is a constraint on the maximum complexity for the detection of the jth symbol vector. The detector is terminated if $D_{\max}[j]$ nodes have been visited and the LLRs are obtained from the current MAP hypothesis, the associated metric λ^{MAP}, and the current extrinsic metrics $\Lambda_{i,b}^{\overline{\mathrm{MAP}}}$. The main idea realized by the policy (46) is that detection of the jth symbol vector is allowed to use up all of the remaining complexity budget (within the block of N symbol vectors) up to $(N-j)M$ nodes, i.e., the parameter M determines that in decoding the remaining $N-j$ symbol vectors, we can afford a budget of at least M nodes per symbol vector. Setting $M = M_{\mathrm{T}}$ and choosing $D_{\mathrm{avg}} \geq M_{\mathrm{T}}$, ensures that for each of the remaining $N-j$ symbol vectors at least the hard-output successive interference cancellation (SIC) solution

TABLE I
REDUCTION OF AVERAGE COMPLEXITY OBTAINED BY TIGHTENING OF THE TREE-PRUNING CRITERION BASED ON THE EUCLIDEAN-DISTANCE TERM ONLY

SNR	$L_{\max}$	std. [nodes]	tight [nodes]	reduction
10 dB	0.0125	34.9	34.4	1.4%
	∞	328.3	327.8	0.2%
20 dB	0.0125	11.0	10.8	1.8%
	∞	227.2	227.0	0.1%

is found [7]. For details on early termination and scheduling, we refer to [7], [10], [35]. For the remainder of the paper, we set $M = M_{\mathrm{T}}$ and we ensure that $D_{\mathrm{avg}} \geq M_{\mathrm{T}}$.

If early termination happens before the extrinsic metric $\Lambda_{i,b}^{\overline{\mathrm{MAP}}}$ was updated from its initial value ∞, the corresponding LLR satisfies $\left|L_{i,b}^{\mathrm{E}}\right| \geq L_{\max}$. Hence, early termination may result in LLRs with a higher reliability (i.e., $\left|L_{i,b}^{\mathrm{E}}\right| > L_{\max}$) than what would be obtained if no complexity constraints had been imposed. This calls for LLR correction with the goal of adjusting the magnitude of such LLRs. Consequently, the side information set $\mathcal{Z}$ should contain a binary-valued state variable, which indicates whether early termination during detection of a symbol vector occurred or not. Corresponding numerical results are provided in Section VI-D1.

VI. SIMULATION RESULTS

Unless explicitly stated otherwise, all simulation results are for a convolutionally encoded (rate $R = 1/2$, generator polynomials $[133_o\ 171_o]$, and constraint length 7) MIMO-OFDM system with $M_{\mathrm{T}} = M_{\mathrm{R}} = 4$, 16-QAM constellation $\mathcal{O}$ with Gray labeling, 64 OFDM tones, TGn type C channel model [36], and a BCJR channel decoder [37] based on the min-sum algorithm and employing the max-log approximation for LLR computation. One frame consists of 1024 randomly interleaved (across space and frequency) bits corresponding to one (spatial) OFDM symbol and we assume that the bits $x_{i,b}$ are statistically independent. A frame is said to be in error if at least one of the bits in the frame is decoded in error. The SNR is per receive antenna and the SNR values specified in the figures are in decibels (dBs). The number of iterations I is the number of times the soft-input soft-output MIMO detector and the SISO channel decoder are used, i.e., $I = 1$ corresponds to soft-output SD according to [7] followed by one application of the BCJR algorithm. The LLR clipping parameters shown in the simulation results correspond to *normalized* LLR clipping parameters according to $L_{\max}/N_0$.

A. Tightening of the Tree-Pruning Criterion

We first numerically characterize the impact on complexity caused by tightening of the tree-pruning criterion.

1) Impact of the Euclidean-distance term: The goal of the simulation results shown in Table I is to quantify the impact of the Euclidean distance term $\frac{1}{N_0}\left|\tilde{y}_i - \sum_{j=i}^{M_{\mathrm{T}}} R_{i,j} s_j\right|^2$ in the bias (16) on the complexity reduction that can be obtained by tightening the tree-pruning criterion according to (18). To this end, we omit the prior term by setting $\log \mathrm{P_c}\left[\mathbf{s}^{(i)}\right] = 0$ in (14) and compare the complexity (averaged over independent channel, noise, and data realizations) resulting from the tightened tree-pruning criterion according to (18) to that of the standard tree-pruning criterion (denoted by "std." in Table I) specified in (15). We observe that the complexity reduction obtained by tightening of the tree-pruning criterion based on the bias caused by the Euclidean distance-term only, is marginal, in particular in the light of the prohibitive effort required to compute (16).

TABLE II
REDUCTION OF AVERAGE COMPLEXITY OBTAINED BY TIGHTENING OF THE TREE-PRUNING CRITERION BASED ON THE PRIOR TERM ONLY

SNR	I	$L_{\max}$	std. [nodes]	tight [nodes]	reduction
10 dB	1	0.0125	1890.4	34.9	98.2%
		∞	2440.2	328.3	86.5%
	2	0.0125	1630.6	43.4	97.3%
		∞	2148.4	406.6	81.1%
20 dB	1	0.0125	1914.7	11.0	99.4%
		∞	2397.0	227.2	90.5%
	2	0.0125	1228.7	6.2	99.5%
		∞	361.9	132.4	65.9%

2) Impact of the prior term: Next, we start with uniform priors, i.e., $L_{i,b}^{\mathrm{A}} = 0$ $(\forall i, b)$ for the first iteration, and employ the tightened DIs in (25). Table II shows that removing the bias p_i in (19) leads to a dramatic reduction in terms of complexity, ranging from 65.9% to 99.5%. Furthermore, we can see that the complexity reduction due to tightening of the tree-pruning criterion is less pronounced in the second iteration (denoted by $I = 2$), but still significant. This can be explained by noting that in the first iteration, the priors satisfy $L_{i,b}^{A} = 0$, which leads to the largest possible values for p_i, $i = 1, \ldots, M_{\mathrm{T}}$ (see Section II-B1). In general, the complexity reduction due to tightening of the tree-pruning criterion decreases with increasing iteration number.

We can now conclude that tightening of the tree-pruning criterion, based on removing the bias caused by the prior term in (14) only, leads to a significant complexity reduction. In the remainder of the paper, we employ tightened DIs according to (25) in combination with the standard pruning criterion (15).

B. Performance/Complexity Tradeoffs

The performance/complexity tradeoffs discussed next and quantified in Figs. 3 – 5, 7, and 8 refer to the *cumulative* (tree-search) complexity in terms of the total number of nodes visited (averaged over independent channel, noise, and data realizations) for SISO detection over I iterations, designated as "average complexity" from now on. The computational complexity incurred by channel decoding is not accounted for in the following. The minimum SNR required to achieve a given frame error rate (FER) is referred to as the "SNR operating point" for that FER.

1) Impact of LLR clipping: From Fig. 3, we can conclude that LLR clipping allows for a smooth performance/complexity tradeoff, adjustable through a single parameter, namely the LLR clipping parameter $L_{\max}$. For a fixed SNR operating

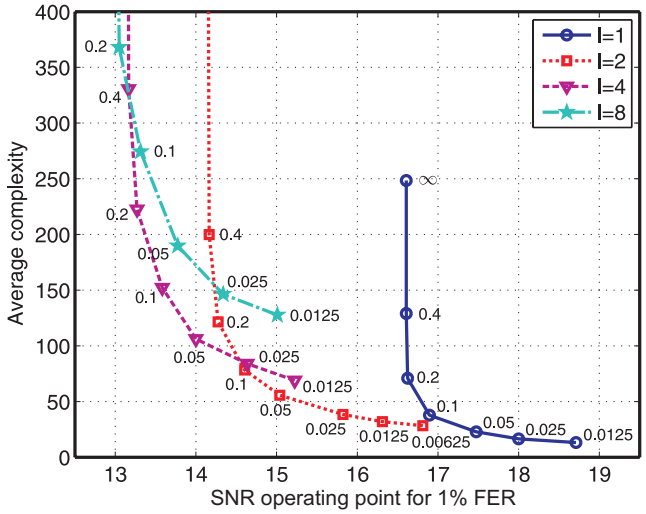


Fig. 3. Performance/complexity tradeoff of SISO STS-SD with SQRD. The numbers next to the curves correspond to normalized LLR clipping parameters.

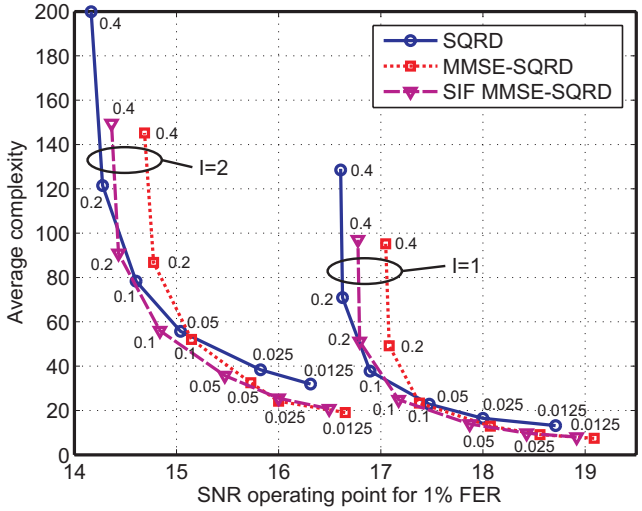


Fig. 4. Performance/complexity tradeoff of SISO STS-SD with SQRD, MMSE-SQRD, and SIF MMSE-SQRD. The numbers next to the curves correspond to normalized LLR clipping parameters.

point, the minimum complexity is not necessarily achieved by maximizing the number of iterations. We conclude that incorporating LLR clipping into the tree search is of paramount importance as it reduces the complexity substantially and renders the detector easily adjustable in terms of performance versus complexity.

2) Column-sorting and regularization: We next examine the impact of column-sorting and regularization of the channel matrix on the performance/complexity tradeoff of the SISO STS-SD algorithm. It can be seen, in Fig. 4, that in the low-complexity regime, the Pareto-optimal tradeoff curve is achieved by MMSE-SQRD. In the high-complexity regime, the performance loss incurred by regularization renders MMSE-SQRD inferior to un-regularized SQRD. This observation has already been made for the soft-output-only case in [7], but is also valid for $I > 1$ under the SISO STS-SD.

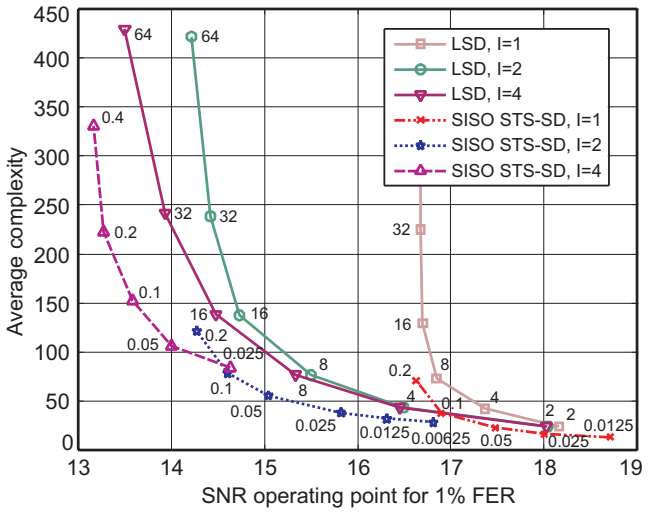


Fig. 5. Performance/complexity tradeoff of LSD [1] and SISO STS-SD, both using SQRD. The numbers next to the curves correspond to the list size for LSD and to normalized LLR clipping parameters for SISO STS-SD.

3) Self-interference free LLRs: Fig. 4 also quantifies the impact of compensating self-interference —according to the procedure described in Section IV-B2— on the performance/complexity tradeoff of the SISO STS-SD algorithm. We observe that compensation of self-interference results in a performance improvement in terms of the SNR operating point of 0.3 dB to 0.5 dB in almost all regions of the performance/complexity curve compared to MMSE-SQRD, but yields no improvement over MMSE-SQRD in the low-complexity regime. In the high-complexity regime, un-regularized SQRD has an SNR-operating point that is 0.15 dB below that obtained in the case of SIF MMSE-SQRD.

C. Comparison with List Sphere Decoding

Fig. 5 compares the performance/complexity tradeoff achieved by list sphere decoding (LSD) as proposed in [1] to that obtained through SISO STS-SD. For the LSD algorithm, we take the complexity to equal the number of nodes visited when building the initial candidate list. The (often significant) computational burden incurred by list administration in LSD is neglected, leading to a complexity measure that favors the LSD algorithm. We can draw the following conclusions from Fig. 5:

i) SISO STS-SD outperforms LSD for all SNR operating points.

ii) LSD requires relatively large list sizes and hence a large amount of memory to come close to (max-log) optimal SISO performance.[7] The underlying reason is that the extrinsic LLRs are obtained from a candidate list that has been computed around the maximum-likelihood solution and hence a priori information is ignored. The SISO STS-SD exhibits significantly smaller memory requirements as it needs memory mainly to store the extrinsic metrics.

Besides LSD, various other SISO detection algorithms for MIMO systems have been developed, see e.g., [2]–[6], [39].

[7] In addition to the memory requirements, the search-and-replace operations required in the LSD algorithm's list administration, quickly lead to prohibitively high VLSI implementation complexity when the list size grows [38].

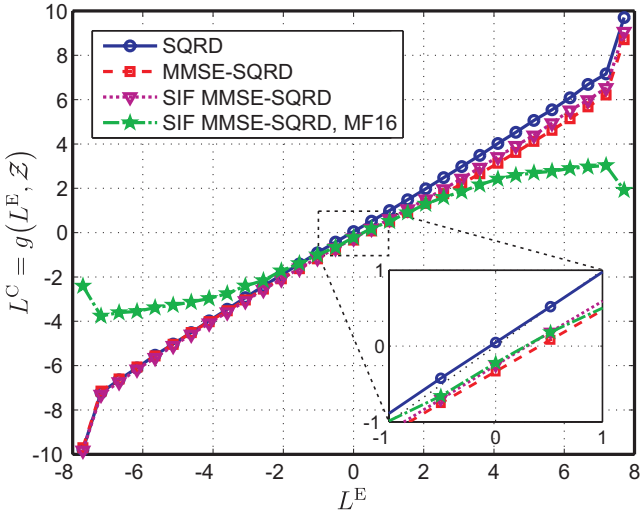


Fig. 6. Different LLR correction functions for $I = 1$ at $\mathsf{SNR} = 16\,\mathrm{dB}$ with $\mathcal{Z} = \{L_{\max}, D_{\mathrm{avg}}, \mathsf{SNR}, T\}$.

The algorithms described in [3] and [6] are related to LSD but require rebuilding the candidate list in each iteration; this can lead to a substantial complexity increase compared to LSD. For [2], [4] issues indicating potentially high computational complexity include the requirement for multiple matrix inversions for each symbol vector in each iteration.[8] In contrast, the QRD required for SD has to be computed only when the channel state changes. The computational complexity of the list-sequential (LISS) algorithm in [5], [39] seems difficult to relate to the complexity measure employed in this paper. However, due to the need for sorting of candidate vectors and the structural similarity of the LISS algorithm to LSD, we expect the performance/complexity tradeoff realized by the LISS algorithm to be comparable to that of the LSD algorithm.

D. Impact of LLR Correction

Fig. 6 shows examples of LLR correction functions (for SISO STS-SD) obtained by linear interpolation using $K = 31$ bins and side information given by

$$\mathcal{Z} = \{L_{\max}, D_{\mathrm{avg}}, \mathsf{SNR}, T\} \tag{47}$$

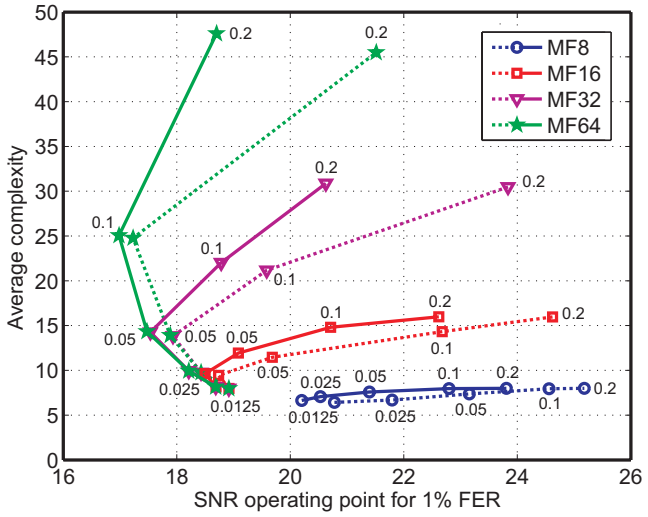
where $L_{\max} = 0.2$, $D_{\mathrm{avg}} \in \{16, \infty\}$, and $T \in \{0, 1\}$ indicates whether early termination occurred ($T = 1$) or not ($T = 0$). Here, the number of instances of $\mathcal{Z}$ is given by $Z = 4$. Note that depending on T, different LLR correction functions need to be applied to the extrinsic LLRs $L^{\mathrm{E}}_{i,b}$. In the following, we analyze the LLR correction functions obtained for column-sorting (SQRD), for regularization with column-sorting (i.e., MMSE-SQRD), and for compensation of self-interference in combination with MMSE-SQRD, all having unconstrained maximum complexity (i.e., $D_{\mathrm{avg}} = \infty$ and, hence, $T = 0$). Fig. 6 shows the corresponding correction functions along with the correction functions for SIF compensation under MMSE-SQRD in combination with MF scheduling for $D_{\mathrm{avg}} = 16$ (denoted by "MF16" in Fig. 6) and $T = 1$. The following observations can be made:

- For unconstrained complexity, i.e., $D_{\mathrm{avg}} = \infty$, LLRs corresponding to $\pm L_{\max}$ are corrected to LLRs with larger magnitude; this is a result of clipping LLRs with magnitude larger than $L_{\max}$ to $\pm L_{\max}$. We note that since the LLR correction functions are obtained by binning and linear interpolation, LLR-values that have slightly smaller (mandated by the bin-width) magnitude than $L_{\max}$ are also corrected to values larger than $L_{\max}$.
- For early termination with MF-scheduling (i.e., $D_{\mathrm{avg}} = 16$ and $T = 1$), LLRs with magnitude close to $L_{\max}$ are corrected to LLRs with smaller magnitude (i.e., their reliability is reduced). In the presence of early termination, LLRs with $\left|L^{\mathrm{E}}_{i,b}\right| \geq L_{\max}$ are, as already mentioned in Section V-C, often a consequence of having terminated the tree search prematurely and hence are corrected to less reliable LLR-values.
- The LLR correction function associated with column-sorting (SQRD) only is almost a linear function with slope one, i.e., $L^{\mathrm{C}}_{i,b} = L^{\mathrm{E}}_{i,b}$, which indicates that little correction is performed. The reason for this behavior is that column-sorting maintains (max-log) optimality and the impact of the max-log approximation itself on performance is small, in general [10].
- The correction functions associated with channel-matrix regularization show a stronger deviation from $L^{\mathrm{C}}_{i,b} = L^{\mathrm{E}}_{i,b}$ (cf. the zoom in Fig. 6), thus reflecting the error in the (max-log) LLRs incurred by regularization (see Section IV-A2).

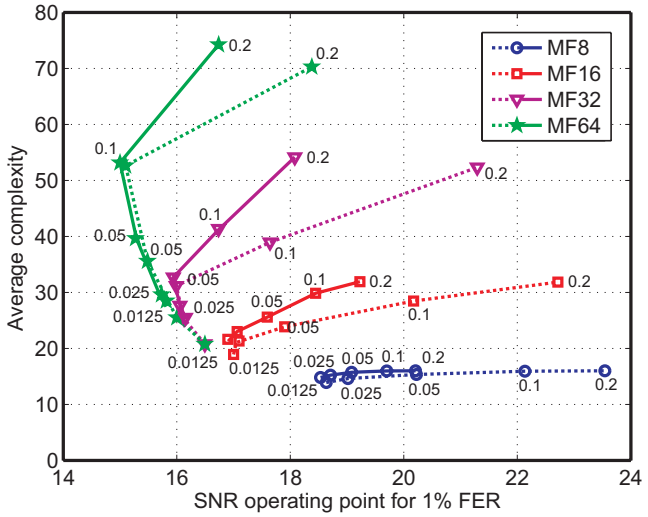
1) Impact of LLR correction on performance/complexity tradeoff for SISO STS-SD with early termination: Fig. 7 shows the performance/complexity tradeoff for early-termination based on MF scheduling with and without LLR correction. The side information was chosen according to (47) and the LLR correction function was computed based on $K = 31$ bins with linear interpolation. As observed for the soft-output STS-SD in [7], for large values of $L_{\max}$, the run-time constrained detector will be unable to compute accurate LLR values since the corresponding average complexity is high; this, in turn, results in poor performance. For small $L_{\max}$, performance is dominated by the impact of LLR clipping rather than early termination. Depending on the constraint on the average run-time, LLR correction can reduce (i.e., improve) the SNR operating point by up to 3 dB. The performance gains resulting from LLR correction are more pronounced for larger clipping parameters as in these cases performance is dominated by the run-time constraint and early termination happens more often. Note that LLR correction also yields slight performance gains for small LLR clipping levels, where the run-time constraints do not affect performance. This indicates that LLR correction can also correct —at least partly— the errors induced by LLR clipping and by channel-matrix regularization.

2) Impact of LLR correction on performance/complexity tradeoff for turbo codes: The next simulation result is aimed at understanding which of the conclusions drawn above change in the presence of more sophisticated channel codes. To this

[8] A detailed complexity analysis of the algorithm described in [2] based on VLSI implementation results can be found in [10].



(a) $I = 1$



(b) $I = 2$

Fig. 7. Impact of LLR correction. The solid lines correspond to the performance obtained with LLR correction, whereas the dotted lines pertain to uncorrected LLRs. Both variants employ early termination with MF scheduling and MMSE-SQRD accompanied by compensation of self-interference in the LLRs.

end, we evaluated the performance/complexity tradeoff for a parallel-concatenated turbo code (PCTC) of rate 1/2 (punctured, memory 2, and generator polynomial $[7_o\ 5_o]$, where 7_o pertains to the feedback path) with eight iterations within the turbo decoder. We use the interleaver specified in the 3GPP standard [40] with 508 information bits. One code-block corresponds to 1024 coded bits including two times four bits for termination of the trellises. We observed (in simulation results) that BCJR decoding in its original form [37] requires very accurate input LLRs. Since we employ the BCJR algorithm in exactly the form as described in [37] for decoding of the PCTC, LLR correction as described in Section V is prudent and is used here with $\mathcal{Z} = \{L_{\max}, \mathsf{SNR}, I\}$.

The results in Fig. 8 indicate that the performance/complexity tradeoff achieved by the PCTC in the first iteration is significantly better than that obtained for the convolutional code (CC) used in the previous simulations.

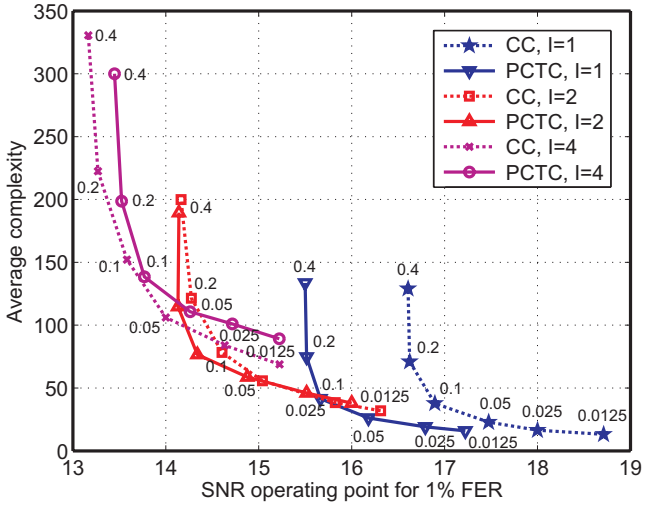


Fig. 8. Performance/complexity tradeoff of SISO STS-SD with SQRD (without regularization) and LLR correction. Comparison between parallel-concatenated turbo codes (PCTCs) and convolutional codes (CCs).

In the second iteration, the performance/complexity tradeoff is almost identical for the two different codes. For $I > 2$, the CC slightly outperforms the PCTC, which could be due to the fact that we use a turbo code with very short block length and a channel model that exhibits correlation across frequency and space (see, e.g., [41]).

E. Information Transfer Characteristics

In order to benchmark the performance of soft-input soft-output MIMO detectors, we compare information transfer characteristics (ICTs) as described in [42, Chapter 16] using an i.i.d. (across space and OFDM tones) Rayleigh multi-path fading channel model. Specifically, the ITCs are obtained as follows. The bits $x_{i,b}$, $\forall i, b$, are assumed to be uniformly distributed and the a priori LLRs are taken to be Gaussian distributed according to [43]

$$L_{i,b}^{\mathrm{A}} = \frac{2}{\sigma^2}\left(x_{i,b} + n\right)$$

where n is a real-valued Gaussian RV with zero mean and variance σ^2. We then define the a priori information content[9] as $I_{\mathrm{A}} = I\left(x_{i,b}; L_{i,b}^{\mathrm{A}}\right)$ and the extrinsic information content at the output of the MIMO detector (averaged over all transmit antennas and bits) as

$$I_{\mathrm{E}} \triangleq \frac{1}{M_{\mathrm{T}} Q} \sum_{i=1}^{M_{\mathrm{T}}} \sum_{b=1}^{Q} I\left(x_{i,b}; L_{i,b}^{\mathrm{E}}\right)$$

in bits per binary symbol where $0 \leq I_{\mathrm{E}} \leq 1$. The ITC is then given by the function $I_{\mathrm{E}} = g(I_{\mathrm{A}})$. Here, we evaluate I_{E} through Monte-Carlo simulations. Note that $L_{i,b}^{\mathrm{A}} = 0$ implies $I_{\mathrm{A}} = 0$ and corresponds to soft-output-only MIMO detection.

1) Impact of LLR clipping: Fig. 9 shows that a normalized LLR clipping parameter of $L_{\max} = 0.4$ achieves almost the

[9]Note that $0 \leq I_{\mathrm{A}} \leq 1$ and I_{A} is determined by σ^2 and corresponds to the mutual information of an additive white Gaussian-noise channel with uniform inputs chosen from a BPSK alphabet.

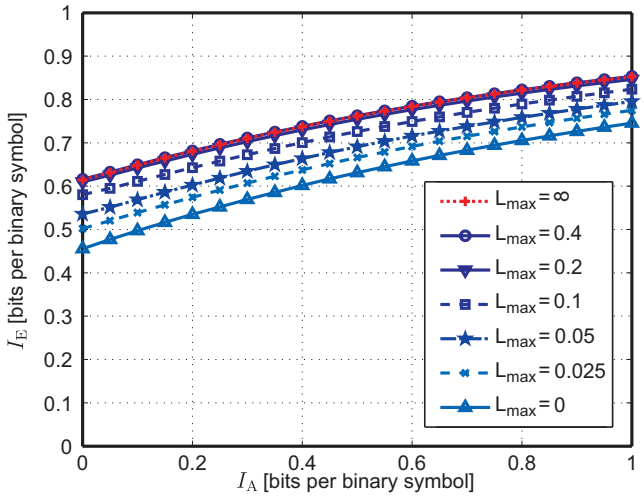


Fig. 9. ITC of SISO STS-SD at SNR $= 12$ dB for different (normalized) LLR clipping parameters.

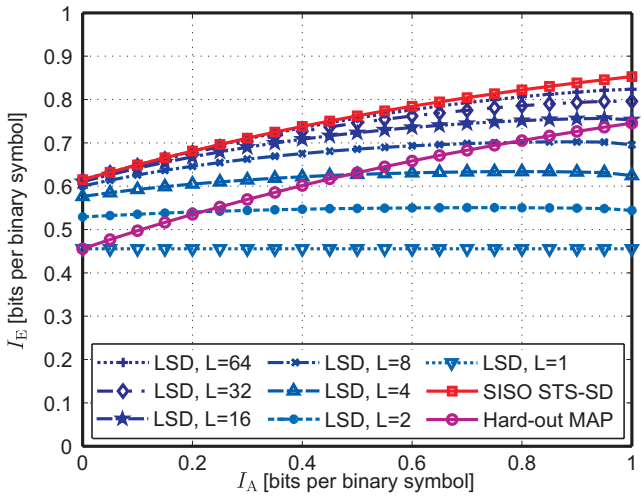


Fig. 10. ITCs of LSD, hard-output MAP (corresponding to $L_{\max} = 0$), and max-log optimal SISO STS-SD (corresponding to $L_{\max} = \infty$) at SNR $= 12$ dB.

same ITC as max-log optimal SISO STS-SD with $L_{\max} = \infty$ (all ITCs are computed under the assumption of no channel-matrix regularization). We note that the same observation was made in the performance/complexity tradeoff simulations in Fig. 3.

2) Performance comparison with LSD: Fig. 10 compares the ITC of SISO STS-SD to that of LSD [1]. For I_{A} close to 1, LSD requires list-sizes in excess of $L = 64$ to yield performance close to that of the max-log-optimal SISO STS-SD algorithm. For I_{A} close to zero, the ITC of the LSD comes close to that of the SISO STS-SD for a list size of $L = 32$. Note that even hard-output MAP detection (which corresponds to SISO STS-SD with $L_{\max} = 0$) can outperform LSD —in terms of ITCs— if I_{A} is close to 1 and the list-size is small. We can conclude that SISO STS-SD exhibits advantages over LSD in terms of complexity and memory requirements, independently of the underlying channel code. This is in agreement with the observations made in Section VI-C.

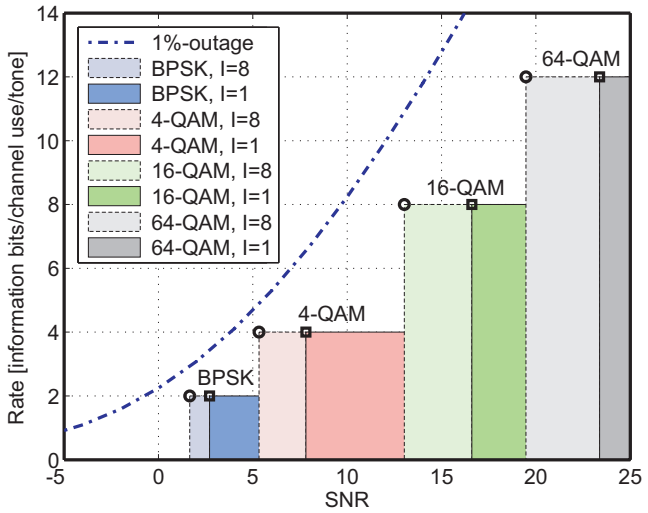


Fig. 11. 1%-outage-capacity compared to the SNR operating points of SISO STS-SD for 1% FER. Squares and circles correspond to SNR operating points realized by $I = 1$ and $I = 8$, respectively.

F. Approaching Outage-Capacity with SISO STS-SD

We finally compare the performance obtained with SISO STS-SD to outage capacity using the TGn type C channel model [36]. To this end, we define the ε-outage capacity $C_{\mathrm{out},\epsilon}$ as [44], [45]

$$\mathrm{P}[I(\mathsf{SNR},\mathcal{H}) < C_{\mathrm{out},\varepsilon}] = \varepsilon \tag{48}$$

where $\mathcal{H} \triangleq \{\mathbf{H}[1],\ldots,\mathbf{H}[N]\}$ contains the $M_{\mathrm{R}} \times M_{\mathrm{T}}$ channel matrices for the $N = 64$ OFDM tones and [46]

$$I(\mathsf{SNR},\mathcal{H}) \triangleq \frac{1}{N}\sum_{\ell=1}^{N} \log_2 \det\left(\mathbf{I}_{M_{\mathrm{R}}} + \frac{\mathsf{SNR}}{M_{\mathrm{T}}}\mathbf{H}^H[\ell]\mathbf{H}[\ell]\right).$$

The FER is lower-bounded by the outage probability (48) according to [47]

$$\mathrm{P}[I\,(\mathsf{SNR},\mathcal{H}) < RM_{\mathrm{T}}Q] \leq \mathrm{FER}(\mathsf{SNR})$$

where the information rate per OFDM tone is given by $RM_{\mathrm{T}}Q$ and $R = 1/2$ denotes the code-rate. The performance comparison we conducted consists of setting the outage probability and the FER to 1% and identifying the gap in the resulting SNR operating points. Fig. 11 shows the corresponding results for different modulation schemes and for $I = 1$ and $I = 8$. Note that the LLR clipping parameters are chosen so as to minimize complexity while retaining near-max-log optimal performance at 1% FER (i.e., we used $L_{\max} = 0.1$, $L_{\max} = 0.4$, $L_{\max} = 2.0$, and $L_{\max} = 6.0$ for 64-QAM, 16-QAM, QPSK, and BPSK, respectively). We can see that SISO STS-SD operates between 1.5 dB (for 4-QAM) and 5.3 dB SNR (for 64-QAM) away from outage capacity.

VII. Conclusions

We proposed a soft-input soft-output MIMO detector based on single tree-search sphere decoding (STS-SD) as introduced in [7], [8]. Besides adapting the single-tree search paradigm to account for soft-inputs, key to obtaining low complexity are tightening of the tree-pruning criterion, clipping of

the extrinsic LLRs built into the tree search, and a novel method for incorporating compensation of self-interference in LLRs —caused by channel-matrix regularization— into the tree search. Finally, we proposed an LLR correction method, which was demonstrated to achieve substantial performance improvements at low additional computational complexity. Our simulation results show that the SISO STS-SD algorithm offers a wide range of performance/complexity tradeoffs and clearly outperforms state-of-the-art SISO detectors for MIMO systems. Finally, we note that a VLSI implementation of the algorithm described in this paper was recently reported in [48].

ACKNOWLEDGMENTS

The authors would like to thank A. Burg for his contributions at early stages of this work and the anonymous reviewers for their valuable comments which helped to improve the exposition.

REFERENCES

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

[2] M. Tüchler, A. C. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Trans. Sig. Proc.*, vol. 50, no. 3, pp. 673–683, Mar. 2002.

[3] J. Boutros, N. Gresset, L. Brunel, and M. Fossorier, "Soft-input soft-output lattice sphere decoder for linear channels," in *Proc. IEEE GLOBECOM*, vol. 3, San Francisco, CA, USA, Dec. 2003, pp. 1583–1587.

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

[5] S. Bäro, J. Hagenauer, and M. Witzke, "Iterative detection of MIMO transmission using a list-sequential (LISS) detector," in *Proc. IEEE ICC*, vol. 4, Anchorage, AK, USA, May 2003, pp. 2653–2657.

[6] H. Vikalo, B. Hassibi, and T. Kailath, "Iterative decoding for MIMO channels via modified sphere decoder," *IEEE Trans. Wireless Comm.*, vol. 3, no. 6, pp. 2299–2311, Nov. 2004.

[7] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: Algorithms and VLSI implementation," *IEEE J. Sel. Areas Comm.*, vol. 26, no. 2, pp. 290–300, Feb. 2008.

[8] J. Jaldén and B. Ottersten, "Parallel implementation of a soft output sphere decoder," in *Proc. Asilomar Conf. on Signals, Systems, and Computers*, Monterey, CA, USA, Nov. 2005, pp. 581–585.

[9] A. Burg, "VLSI circuits for MIMO communication systems," Ph.D. dissertation, ETH Zürich, Switzerland, Series in Microelectronics, vol. 169, Hartung-Gorre Verlag Konstanz, 2006.

[10] C. Studer, "Iterative MIMO decoding: Algorithms and VLSI implementation aspects," Ph.D. dissertation, ETH Zürich, Switzerland, Series in Microelectronics, vol. 202, Hartung-Gorre Verlag Konstanz, 2009.

[11] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. of Computation*, vol. 44, no. 170, pp. 463–471, Apr. 1985.

[12] W. H. Mow, "Maximum likelihood sequence estimation from the lattice viewpoint," Master's thesis, Chinese University of Hong Kong, Department of Information Engineering, 1991.

[13] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, no. 2, pp. 181–191, Sept. 1994.

[14] E. Viterbo and E. Biglieri, "A universal decoding algorithm for lattice codes," in *14ème colloque GRETSI*, Juan-les-Pins, France, Sept. 1993, pp. 611–614.

[15] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inf. Th.*, vol. 45, no. 5, pp. 1639–1642, July 1999.

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

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

[18] J. Jaldén and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Sig. Proc.*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.

[19] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Th.*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[20] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid State Circuits*, vol. 40, no. 7, pp. 1566–1577, July 2005.

[21] M. Stojnic, H. Vikalo, and B. Hassibi, "Speeding up the sphere decoder with H^∞ and SDP inspired lower bounds," *IEEE Trans. Sig. Proc.*, vol. 56, no. 2, pp. 712–726, Feb. 2008.

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

[23] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inf. Th.*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[24] D. Seethaler, H. Artés, and F. Hlawatsch, "Dynamic nulling-and-canceling for efficient near-ML decoding of MIMO systems," *IEEE Trans. Sig. Proc.*, vol. 54, no. 12, pp. 4741–4752, Dec. 2006.

[25] S. W. Kim and K. P. Kim, "Log-likelihood-ratio-based detection ordering in V-BLAST," *IEEE Trans. Comm.*, vol. 54, no. 2, pp. 302–307, Feb. 2006.

[26] D. Wübben, R. Böhnke, J. Rinas, V. Kühn, and K.-D. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *IEE Electronics Letters*, vol. 37, no. 22, pp. 1348–1350, Oct. 2001.

[27] P. J. Lüthi, "VLSI circuits for MIMO preprocessing," Ph.D. dissertation, ETH Zürich, Series in Microelectronics, vol. 203, Hartung-Gorre Verlag Konstanz, 2010.

[28] M. O. Damen, H. El Gamal, and G. Caire, "On maximum likelihood detection and the search for the closest lattice point," *IEEE Trans. Inf. Th.*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

[29] P. Luethi, A. Burg, S. Haene, D. Perels, N. Felber, and W. Fichtner, "VLSI implementation of a high-speed iterative sorted MMSE QR decomposition," in *Proc. IEEE ISCAS*, New Orleans, LA, USA, May 2007, pp. 1421–1424.

[30] D. Wübben, R. Böhnke, V. Kühn, and K.-D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in *Proc. IEEE VTC-Fall*, vol. 1, Orlando, FL, USA, Oct. 2003, pp. 508–512.

[31] E. Zimmermann and G. Fettweis, "Unbiased MMSE tree search detection for multiple antenna systems," in *Proc. WPMC*, San Diego, CA, USA, Sept. 2006.

[32] M. van Dijk, A. J. E. M. Janssen, and A. G. C. Koppelaar, "Correcting systematic mismatches in computed log-likelihood ratios," *Europ. Trans. Telecomm.*, vol. 14, no. 3, pp. 227–244, July 2003.

[33] A. Burg, M. Wenk, and W. Fichtner, "VLSI implementation of pipelined sphere decoding with early termination," in *Proc. EUSIPCO*, Florence, Italy, Sept. 2006.

[34] C. Studer, D. Seethaler, and H. Bölcskei, "Finite lattice-size effects in MIMO detection," in *Proc. 42nd Asilomar Conf. on Signals, Systems, and Computers*, Oct. 2008, pp. 2032–2037.

[35] A. Burg, M. Borgmann, M. Wenk, C. Studer, and H. Bölcskei, "Advanced receiver algorithms for MIMO wireless communications," in *Proc. DATE*, vol. 1, Munich, Germany, Mar. 2006, pp. 593–598.

[36] V. Erceg *et al.*, *TGn channel models*, May 2004, IEEE 802.11 document 03/940r4.

[37] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Th.*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[38] M. Wenk, A. Burg, M. Zellweger, C. Studer, and W. Fichtner, "VLSI implementation of the list sphere algorithm," in *Proc. 24th NORCHIP Conf.*, Linköping, Sweden, Nov. 2006, pp. 107–110.

[39] J. Hagenauer and C. Kuhn, "The list-sequential (LISS) algorithm and its application," *IEEE Trans. Comm.*, vol. 55, no. 5, pp. 918–928, May 2007.

[40] *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Multiplexing and Channel Coding (FDD)*, 3GPP Organizational Partners TS 25.212, Rev. 5.10.0, June 2005.

[41] J. Boutros, F. Boixadera, and C. Lamy, "Bit-interleaved coded modulations for multiple-input multiple-output channels," in *Proc. IEEE ISSSTA*, vol. 1, Sept. 2000, pp. 123–126.

[42] H. Bölcskei, D. Gesbert, C. Papadias, and A. J. van der Veen, Eds., *Space-Time Wireless Systems: From Array Processing to MIMO Communications*. Cambridge Univ. Press, 2006.
[43] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Comm.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.
[44] İ. E. Telatar, "Capacity of multi-antenna Gaussian channels," *Europ. Trans. Telecomm.*, vol. 10, no. 6, pp. 585–596, 1999.
[45] E. Biglieri, J. Proakis, and S. Shamai, "Fading channels: Information-theoretic and communications aspects," *IEEE Trans. Inf. Th.*, vol. 44, no. 6, pp. 2619–2692, Oct. 1998.
[46] H. Bölcskei, D. Gesbert, and A. J. Paulraj, "On the capacity of OFDM-based spatial multiplexing systems," *IEEE Trans. Comm.*, vol. 50, no. 2, pp. 225–234, Feb. 2002.
[47] L. Zheng and D. N. C. Tse, "Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels," *IEEE Trans. Inf. Th.*, vol. 5, no. 49, pp. 1073–1096, May 2003.
[48] E. M. Witte, F. Borlenghi, G. Ascheid, R. Leupers, and H. Meyr, "A scalable VLSI architecture for soft-input soft-output depth-first sphere decoding," *submitted to IEEE Trans. Circuits and Systems II*, Apr. 2010. [Online]. Available: http://arxiv.org/abs/0910.3427