Iterative Soft-In Soft-Out Sphere Detection for MIMO Systems

Björn Mennenga and Richard Fritzsche and Gerhard Fettweis
Vodafone Chair Mobile Communication Systems
Technische Universität Dresden
Dresden, Germany
Email: mennenga,richard.fritzsche,fettweis@ifn.et.tu-dresden.de

***Abstract*—Soft-Input Soft-Output (SISO) tree search algorithms provide a promising approach for complexity reduced iterative MIMO detection. Realizations based on depth-first search enable near max-log optimal detection at reduced but still high complexity. In this paper we introduce how SISO sphere detectors based on enhancements of single tree search and tuple search algorithms can be efficiently used in iterative detection, outclassing previously proposed decoders or list based iterations. The complexity of the proposed algorithms can be significantly reduced by MMSE preprocessing in combination with a novel unbiased and separated candidate assimilation. Internal clipping of search paths enables further complexity reduction as well as alignment of the tree searches leading to efficient realizations.**

I. Introduction

Future mobile communication systems will make use of multiple-input multiple-output (MIMO) techniques to enhance spectral efficiency. The performance of detecting the spatial multiplexed data streams can be significantly improved by iterative detection-decoding. However, complexity of the SISO search limits the overall throughput. Complexity reduced iterative detection, e.g. relying on the application of list sphere detector [1], M-algorithms [2] or modifications [3] of them, has shown to be of high accuracy but also of high complexity, resulting from the complexity of the underlying tree searches. In order to reduce this complexity, [1] proposed to accomplish the iterations on a candidate list generated in the first iterations, leading to a performance loss and high memory requirements.

In this paper we extent two of the most promising soft-out detectors, relying on Single Tree Search (STS) [4] and Tuple Search (TS) [5] algorithm, to SISO processing. Based on this, we reduce the complexity of these algorithms. Building up on clipping used to limit the extrinsic information [1], we introduce an internal data path clipping for the TS for complexity reduction, suitable for detection algorithms with a-priori knowledge. The definition of a comparable path clipping to the STS enables likewise a significant complexity reduction as well as adjustability, as shown for the non iterative detection in [6]. Additionally, we propose the inclusion of MMSE preprocessing for both algorithms to further reduce the complexity. To avoid performance degradations, we introduce a novel separated candidate processing, enabling the determination of unbiased MMSE metrics for the soft-output calculation without affecting the tree searches. Even if the TS is also applicable to list based iteration, we demonstrate that an iterative sphere search, using the proposed detectors, is preferable for efficient realizations.

II. System model

Throughout this paper, we consider a $N_\mathrm{T} \times N_\mathrm{R}$ MIMO system based on a bit-interleaved coded modulation (BICM) transmission strategy with N_T transmit and N_R receive antennas, as depicted in Fig. 1. A vector $\mathbf{u}$ of i.i.d. information bits is encoded by the outer channel code with rate R. The resulting stream of vectors $\mathbf{c}'$ is bit-interleaved and portioned into blocks $\mathbf{c}$ of $N_\mathrm{T} \cdot L$ bits, where L denotes the number of bits per transmit symbol. For the transmission,

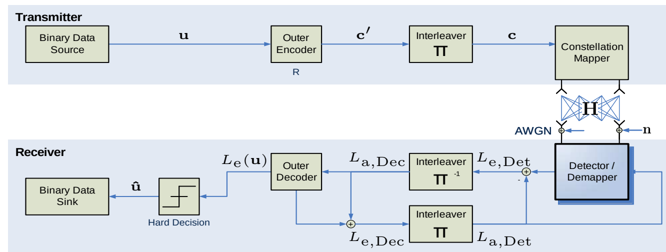


Fig. 1. System model with BICM transmitter and iterative receiver.

the corresponding bits $\mathbf{c} \in \mathcal{C}$, covered in the set of permitted bit vectors, are mapped (e.g. gray mapping) onto complex constellation symbols $\mathbf{x}(\mathbf{c}) = [x_0, ...x_{N_\mathrm{T}-1}]^T = map(\mathbf{c}) \in \mathcal{X}$, the set of valid transmit symbols with cardinality $\#\mathcal{X} = \#\mathcal{C} = 2^L$. We normalize the transmit energy such that $\mathcal{E}\{\mathbf{x}\mathbf{x}^\mathrm{H}\} = E_\mathrm{S}/N_\mathrm{T}\mathbf{I}$. On behalf of the transmission, we consider a flat fading channel and an additive noise vector $\mathbf{n} \in \mathbb{C}^{\mathbf{N_\mathrm{R}} \times \mathbf{1}}$ at the receiver with complex components of zero mean i.i.d. gaussian random variables of variance $N_0/2$ per real dimension ($\mathcal{E}\{\mathbf{n}\mathbf{n}^\mathrm{H}\} = N_0\mathbf{I}$). The considered passive channel is represented by $\mathbf{H} \in \mathbb{C}^{N_\mathrm{T} \times N_\mathrm{R}}$ with entries of a zero mean i.i.d. gaussian random process of variance 1 and is assumed to be perfectly known at the receiver. The received signal $\mathbf{y}$ is therefore given by:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$$

and the signal-to-noise-ratio ($SNR = E_s/N_0$) at the receiver applied to the energy of one information bit can be stated as: $E_\mathrm{b}/N_0 = E_s N_\mathrm{R}/N_0 N_\mathrm{T} L R$.

In order to ensure comparability of the results, we use a simulation setup equivalent to the one used in e.g. [1], [7]. The simulations are carried out for a rate $1/2$ PCCC with $(7_R, 5)$ convolutional codes, an information block size of 9216 bits (including tail bits), gray mapping, a 4×4 MIMO channel and spatial and temporal fading. The detection of the transmitted bits is carried out by a complex-valued soft-in soft-out (SISO) sphere detector in conjunction with a BCJR based decoder with 8 internal iterations.

III. MIMO Detection based on Tree Search

A. Fundamentals

The task of the focused detector is the determination of the bits c most likely sent and the calculation of reliability information for these bits. On behalf of the described system, this can be accomplished by calculating the corresponding log-likelihood ratios (L-values):

$$L\left(c_{m,l}|\mathbf{y}\right) = \ln\left(\frac{P\left(c_{m,l}=+1|\mathbf{y}\right)}{P\left(c_{m,l}=-1|\mathbf{y}\right)}\right) \approx -\frac{1}{N_0}\min_{\mathbf{c}|c_{m,l}=+1}\{\lambda_0\} + \frac{1}{N_0}\min_{\mathbf{c}|c_{m,l}=-1}\{\lambda_0\}, \quad (1)$$

where (1) results from application of the max-log approximation. $c_{m,l}$ represents the l-th bit of a symbol sent by the m-th antenna.

$$\lambda_0\left(\mathbf{y}, \mathbf{c}, \mathbf{L_a}\right) = \|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}(\mathbf{c})\|^2 - \frac{N_0}{2}\sum_{i=0}^{N_T-1}\sum_{j=1}^{L} c_{i,j} L_a(c_{i,j}) \quad (2)$$

represents the distance metric for a set of received symbols $\mathbf{y}$, a given $\mathbf{c}$ and the a-priori knowledge $\mathbf{L_a}$. $\hat{\mathbf{x}}$ corresponds to a possible transmission symbol. As consequence, the detector has to determine beside the most properly sent symbol $\underset{\hat{\mathbf{x}}(\mathbf{c})|\mathbf{c}\in\mathcal{C}}{\arg\min}\{\lambda_0\}$ - the detection hypothesis - and its corresponding metric $\lambda_0(\mathbf{c}^{\mathrm{ML}})$, also the counter-hypotheses $\underset{\hat{\mathbf{x}}(\mathbf{c})|\mathbf{c}\in\mathcal{C},c_{m,l}\neq c^{\mathrm{ML}}_{m,l}}{\arg\min}\{\lambda_0\}$ with their metrics for each bit.

B. Tree Search Basics

Since brute force (maxlogAPP) detection of (1) is known to be of exponential growing complexity with the number of transmit antennas, several close to optimal detection strategies have been lately proposed to find relevant $\arg\min\{\lambda_0\}$. Some of the most promising are based on tree search techniques. As depicted in detail in [1], transforming the detection problem is permitted by QR-decomposition (QRD) of $\mathbf{H} = \mathbf{QR}$, where $\mathbf{Q}$ is unitary and $\mathbf{R}$ an upper triangular matrix. With the modified receive symbols $\mathbf{y}' = \mathbf{Q}^{\mathrm{H}}\mathbf{y}$ and the potential sent symbols , the euclidian distance in the detection

$$\left\|\mathbf{y}' - \mathbf{R}\hat{\mathbf{x}}(\mathbf{c})\right\|^2 \qquad (3)$$

can be interpreted as tree search. The search tree and the relevant notations are drafted in Fig. 2 for a 2 QAM and $N_T = 4$ transmit antennas. The root node of the tree is defined as layer $i = N_T$. In each of the layers i, $i = (N_T - 1)...0$, 2^L possible transmission symbols $\hat{x}_i$ are existing for one parent node. Each of these symbols is represented by a tree node and connected to the parent via branches. Layer $i = (N_T) - 1$, corresponding to the lowest row of (3) and a path from $i = (N_T) - 1$ to $i = 0$ represents a complete set of sent symbols $\hat{\mathbf{x}}$, mapped to the leaves of the tree. Resulting from this, λ_0 (2) can be recursively calculated by the layered branch metric

$$\lambda_i = \underbrace{\lambda_{i+1}}_{\substack{\text{metric from}\\ \text{already estimated}\\ \text{symbols}}} + \Big|\underbrace{y_i''}_{\substack{\text{interference reduced}\\ \text{symbol}}} - r_{ii}\hat{x}_i\Big|^2 - \underbrace{\frac{N_0}{2}\sum_{j=1}^{L} c_{i,j}L_a(c_{i,j})}_{\text{a-priori information}}, \qquad (4)$$

$$y_i'' = y_i' - \sum_{j=i+1}^{N_T-1} r_{ij}\hat{x}_j,$$

out of the squared distance between the nodes and an interference reduced symbol, the a-priori information and the corresponding parent node metric, whereas the root metric is defined to $\lambda_{N_T} = 0$.

The aim of the detection algorithm is the selection and analysis of nodes relevant for (1). Complexity reduction of the search is achieved by eliminating unfavorable paths from the search. Within this paper, we focus on variations of sphere detection algorithm [1] [8], a so called depth-first search, which can be briefly summarized as follows: The sphere search is started in layer $i = N_T$ with an unlimited sphere. At each level i the algorithm analyzes the children nodes with their λ_i (4) and selects one of the nodes within the search sphere, which wasn't extended so far. The selected node is extended by analyzing its children nodes in the layer $i - 1$. Whenever a leaf is reached ($i = 0$), the search radius is adapted. Whenever all children nodes from a parent node within the sphere were extended, the search level is increased and the search continued. As soon the root is reached again, the search is completed. The operations of a sphere detection are drafted exemplarily in Fig. 2 with an adaptation of the radius in operations ④, ⑨ and a ML symbol found in operation ⑨. Paths excluded from the search are illustrated with thin branches, paths examined with thick ones.

One prerequisite for sphere detection is a monotonously increased metric. If the layered distance metric would be not monotonous (at

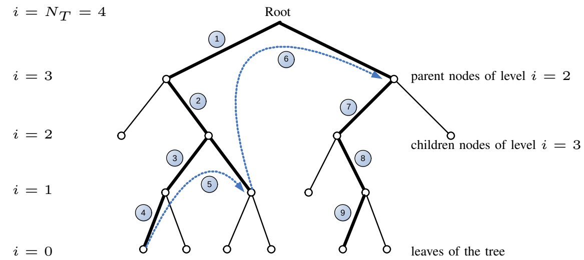


Fig. 2. Example sphere based tree search

least one $\lambda_{i-1} < \lambda_i$), erroneous exclusions might occur: A path with a λ_{i-1} lying inside the sphere (probably leading to a new candidate) will be excluded from the search if its corresponding λ_i is outside the sphere. To avoid negative metric increments by the iterative detection, under supposing $\min|y_i'' - r_{ii}\hat{x}_i|^2 = 0$, equation (4) has hence to be extended by the absolute value of the a-priori information:

$$\begin{aligned}\lambda_i \quad &\mapsto \quad \lambda_i + \frac{N_0}{2}\sum_{j=1}^{L}\left|L_a\left(c_{i.j}\right)\right| \\ &= \quad \lambda_{i+1} + \left|y_i'' - r_{ii}\hat{x}_i\right|^2 \\ &\quad + \frac{N_0}{2}\sum_{j=1}^{L}\left(\left|L_a\left(c_{i.j}\right)\right| - c_{i,j}L_a(c_{i,j})\right), \qquad (5)\end{aligned}$$

leading to a similar result as [9]. Note that due to the identical extension of hypothesis and counter-hypotheses the modified distance metric has no affect on the L-value calculation in (1).

Assuming an implementation with one extended node per cycle, as described in [10], the decisive factor for the throughput is the number of extended nodes. Throughout this paper, hence the number of extended nodes is chosen as complexity measurement.

C. Methods for Complexity Reduction

Efficient application of the sphere detection is permitted by inclusion of methods allowing a reduction of the average number of nodes extended by the search algorithms.

Crucial for reducing the number of examined nodes is the search sphere size. Since the sphere is reduced by means of determined leaves, it is essential to find proper leaves as soon as possible. A solution to enable this is the so called Schnorr-Euchner enumeration strategy [11]. Analyzed child nodes are extended in ascending order of their λ_i (4). This increases the probability of finding nodes with small λ_0 early and reduces hence the average complexity.

A second method to reduce complexity is provided by layer ordering. Choosing a wrong node in high layers causes unnecessary analysis and extensions of child nodes. Wrong selections in lower layers hence have less negative effect on complexity. Detection of most reliable symbols first minimizes the probability of wrong decision in high layers. A common approach to apply layer ordering is the inclusion of sorted QRD (SQRD) as proposed in [12].

Additionally to these methods, MMSE preprocessing [13] might be used to further reduce complexity. This can be done by utilizing an extended channel matrix for the (S)QRD. Following this procedure, the MMSE preprocessing incurs a data dependant bias $-\sigma^2\left\|\hat{\mathbf{x}}\left(\mathbf{c}\right)\right\|^2$ on the metrics, which jams the L-value calculation. Hence, it should be removed to avoid performance degradations [7]:

$$\lambda_{0,\mathrm{ub}}\left(\mathbf{y},\mathbf{c},\mathbf{L_a}\right) = \lambda_0\left(\mathbf{y},\mathbf{c},\mathbf{L_a}\right) - \sigma^2\left\|\hat{\mathbf{x}}\left(\mathbf{c}\right)\right\|^2. \qquad (6)$$

D. Tuple Search Detector

Computing the L-values in (1) requires determination of a detection hypothesis and all counter-hypotheses as depicted in III-A. Since explicit search for all needed minimums leads to insufficient high complexity [6], the key to efficient detection is the adequate usage of information gathered during the search in combination with an adapted search strategy. One possibility is the application of search tuples to the List Sphere Detector (LSD) [1] as described in detail in [5] and briefly depicted as follows: Instead of searching all possible minima, the Tuple Search (TS) algorithm searches the T most likely leaves. As the measurement of the nodes' reliability is given by their distance metrics (5), only nodes with a metric smaller than the T best leaf metrics have to be examined. For this, the metrics λ_0 of the T most reliable determined leaves are stored in a search tuple $\mathcal{T} := \{\lambda_0(\mathbf{c}_1), \lambda_0(\mathbf{c}_2), \ldots, \lambda_0(\mathbf{c}_T), \}, T = \#\mathcal{T}$. The sphere radius is defined to the maximum tuple metric:

$$R = \max_{\mathbf{c}_t | \mathbf{c}_t \in \mathcal{T}} \{\lambda_{0,t}\}.$$

For initialization, the tuple elements are set to $\lambda_{0,t} = \infty$. Resulting from its definition, the sphere search finds all leaves within the current sphere. Each leaf within the current sphere, identified by the search, hence represents a new element of the tuple, which replaces the worst element of the tuple:

$$\lambda_0(\mathbf{c}) \mapsto \max_{\mathbf{c}_t | \mathbf{c}_t \in \mathcal{T}} \{\lambda_{0,t}\}.$$

Status information for the L-value calculation (1) has to be stored separately, since the search criterium (5) might differ from the minima in (1). In case of the proposed MMSE preprocessing, the best leaves with respect to $f(\lambda_0(\mathbf{c}), \sigma^2, \mathbf{c}) = \lambda_0 - \sigma^2 \|\mathbf{x}\|^2 = \lambda_{0,\mathrm{ub}}$ (6) have to be stored for (1). This can be accomplished e.g. via a separate candidate list [5]. For the sake of simplicity we will use in the subsequent $f(\mathbf{c})$ in order to refer to $f(\lambda_0(\mathbf{c}), \sigma^2, \mathbf{c})$.

A candidate list $\mathcal{K} := \{f(\mathbf{c}_1), ..., f(\mathbf{c}_K)\}$ is able to store the $K = \#\mathcal{K}$ best leaves $\mathbf{c}$ visited during the search and their corresponding metrics $f(\mathbf{c})$. For this, the list is initialized with elements of infinitesimal reliability ($f(\cdot) = \infty$) and updated whenever a leaf is reached:

$$\mathbf{c} \mapsto \arg\max_{\mathbf{c}_t | \mathbf{c}_t \in \mathcal{K}} \{f(\mathbf{c}_t)\}, \quad f(\mathbf{c}) \mapsto \max_{\mathbf{c}_t | \mathbf{c}_t \in \mathcal{K}} \{f(\mathbf{c}_t)\},$$

Subsequent to the search, the calculation of the L-values in (1) is carried out over the hypothesis and the counter-hypotheses stored in $\mathcal{K}$, whereas the min-searches are reduced to:

$$\min_{\mathbf{c} | c_{m,l} = \pm 1} \{f(\mathbf{c})\} \mapsto \min_{\mathbf{c} | \mathbf{c} \in \mathcal{K}, c_{m,l} = \pm 1} \{f(\mathbf{c})\}, \quad \mathbf{c} \in \mathcal{K} \subseteq \mathcal{C}.$$

Since the L-value calculation is accomplished over leaves examined during the tree search, probably not all relevant counter-hypotheses were found. Moreover, relevant determined leaves might be dropped if the number of visited leaves exceeds K. Consequently, extrinsic L-values have to be clipped: $|L_e| \leq L_{\max}$ and defining a convenient clipping level $L_{\max}$ is crucial for good performance [3]. Hence, for our simulations the clipping level is chosen such that the average mutual information at the detector output is maximized [14].

Clipping status information enables a novel possibility for complexity reduction for SISO detectors with ZF and approximatively also with MMSE preprocessing: The elimination of counter-hypotheses paths with metrics which will be clipped. Assume $\min_{\mathbf{c} | c_{m,l} = +1} \{f(\mathbf{c})\} - \min_{\mathbf{c} | c_{m,l} = -1} \{f(\mathbf{c})\} \cong \min_{\mathbf{c} | c_{m,l} = +1} \{\lambda_0\} - \min_{\mathbf{c} | c_{m,l} = -1} \{\lambda_0\}$, as for the ZF preprocessing, and extrinsic information $L_e(c_{m,l}) = \frac{c_{m,l}^{\mathrm{ML}}}{N_0} \left(\min_{c_{m,l} \neq c_{m,l}^{\mathrm{ML}}} \{\lambda_0\} - \lambda_0(\mathbf{c}^{\mathrm{ML}}) \right) - L_a(c_{m,l})$ is clipped with L_{max}. All paths for partial symbols $\mathbf{c_i}$ (defined symbols down to layer i) with $\lambda(\mathbf{c_i}) \geq \lambda(\mathbf{c}^{\mathrm{ML}})$ and

$$\lambda(\mathbf{c_i}) \geq N_0 \left(L_{\max} + c_{m,l}^{\mathrm{ML}} L_a(c_{m,l}) \right) + \lambda_0(\mathbf{c}^{\mathrm{ML}}), \tag{7}$$

$$\forall\ c_{m,l} \in \mathbf{c_i} | c_{m,l} \neq c_{m,l}^{\mathrm{ML}} \text{ and } \forall\ c_{m,l} \neq c_{m,l}^{\mathrm{ML}} | m < i,$$

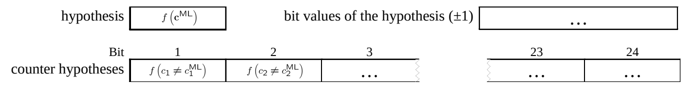


Fig. 3. Bit-specific storage for a 64 QAM and $N_{\mathrm{T}} = 4$

can be eliminated. As result, a clipping value has to be calculated for every $\mathbf{c_i}$ separately, causing overhead to the detection. Instead of this, we propose the inclusion of a reduced but fixed internal clipping value L_{Clip} for the LSD or TS by choosing maximum clipping value:

$$L_{Clip} = \max\left\{0, N_0 \left(L_{\max} + \max_{m,l} \{|L_a(c_{m,l})|\} \right) \right\} + \lambda_0(\mathbf{c}^{\mathrm{ML}}).$$

The search sphere is hence limited by L_{Clip} and the search tuple:

$$R_{Clipped} = \min\left\{ \max_{\mathbf{c}_t | \mathbf{c}_t \in \mathcal{T}} \{\lambda_{\min,t}\}, L_{Clip} \right\}, \tag{8}$$

whereas $R_{Clipped}$ only has to be calculated once a leaf within the sphere is found.

E. Single Tree Search

Another solution for a tree search adapted to the L-value calculation is the Single Tree Search (STS) introduced by [6] and carried out in [4]. As described in detail in [4], the STS searches for all bit values, $c_{m,l} = \pm 1$, the leaves with the lowest metric $\min_{\mathbf{c} | \mathbf{c} \in \mathcal{C}, c_{m,l} = \pm 1} \{\lambda_0\}$. Involved by the bit-specific search, the search sphere varies depending on the examined nodes and has to be established for each node separately. The sphere radius of nodes representing partially symbols $\mathbf{c_i}$ is dependant on the best metrics $\lambda_{c_{m,l}}$ already found for $c_{m,l}$ and the bit values contained in $\mathbf{c_i}$:

$$R(\mathbf{c_i}) = \max\{ \begin{array}{lll} \lambda_{c_{m,l}} & \forall c_{m,l} \in \mathbf{c_i} | c_{m,l} \neq c_{m,l}^{\mathrm{ML}}, & m \geq i, \\ \lambda_{c_{m,l}} & \forall c_{m,l}, & m < i \ \}. \end{array}$$

As for the TS, the L-values might be clipped for complexity reduction. For the iterative detection we propose a modified STS, based on enhanced clipping analog to the TS clipping proposed in section III-D. Since STS's search radii are already dependant on partial symbol vectors $\mathbf{c_i}$, a clipping relying on (7) with

$$L_{Clip}(\mathbf{c_i}) = \max_{m,l} \left\{0, N_0 \left(L_{\max} + c_{m,l}^{\mathrm{ML}} L_a(c_{m,l}) \right) \right\} + \lambda_0(\mathbf{c}^{\mathrm{ML}})$$

is reasonable. Note that the enclosed maximum search only has to be carried out for bits where the path of $\mathbf{c_i}$ might lead to a new hypothesis or counter-hypotheses and hence $\forall\ m, l | m < i$ and $\forall\ m, l | c_{m,l} \in \mathbf{c_i}$; $c_{m,l} \neq c_{m,l}^{\mathrm{ML}}$. Resulting from the definition of the internal clipping value, the search sphere of the STS is given by:

$$R_{Clipped}(\mathbf{c_i}) = \min\{R(\mathbf{c_i}), L_{Clip}(\mathbf{c_i})\}.$$

Besides the additional complexity needed for this sphere determination, the STS incurs an integrated search for each bit. Please note that the proposed clipping leads to a different result than the LLR clipping of [15] and that it can be calculated without case differentiations in parallel to the metric calculations, in opposite to [15].

In order to further reduce the complexity of the STS we propose an introduction of MMSE preprocessing. Unlike the LSD or the TS, the STS does not enable a bias reduction subsequent to the search. As consequence, the MMSE preprocessing leads to significant performance degradations, making an application of MMSE to the STS unattractive. To avoid this limitation, we extend the conventional STS by an additional bit-specific storage (Fig. 3), equal to the one we proposed for the TS in [5]. Based on the MMSE preprocessing (section III-C), the STS itself is carried out, as described above, for the ZF preprocessing. The L-value calculation requires storage of the lowest hypothesis $\mathbf{c}^{\mathrm{ML}}$, the corresponding metric $f(\mathbf{c}^{\mathrm{ML}})$ and the best metric $f(c_i)$ for each counter-hypothesis $\mathbf{c} | c_i \neq c_i^{\mathrm{ML}}$, where c_i represents the i-th bit of $\mathbf{c}$. As for a candidate list, the metrics are initialized with ∞. Whenever a leaf is reached, the current metric is compared with the stored values $\mathbf{c}^{\mathrm{ML}}, f\left(\mathbf{c}^{\mathrm{ML}}\right)$:

$$\begin{aligned} \arg\min\left\{f\left(\mathbf{c}\right), f\left(\mathbf{c}^{\mathrm{ML}}\right)\right\} &\mapsto \mathbf{c}^{\mathrm{ML}}, \\ \min\left\{f\left(\mathbf{c}\right), f\left(\mathbf{c}^{\mathrm{ML}}\right)\right\} &\mapsto f\left(\mathbf{c}^{\mathrm{ML}}\right), \end{aligned}$$

and subsequent to this with $f\left(c_i\right)$ corresponding to the current bit values of $\mathbf{c}$:

$$\min\left\{f\left(\mathbf{c}\right), f\left(c_i\right)\right\} \mapsto f\left(c_i\right), \ \ \forall \ \ c_i \in \mathbf{c}|c_i \neq c_i^{\mathrm{ML}},$$

whereas the best are stored. The calculation of L-values following the search is carried out over the stored $f\left(\mathbf{c}^{\mathrm{ML}}\right)$ and $f\left(c_i\right)$. Note that the extended MMSE STS will not necessarily find the best $\mathbf{c}$ with respect to $f\left(\mathbf{c}\right)$. This is caused since the MMSE STS searches the lowest λ_0 instead of $f\left(\mathbf{c}\right)$. Nevertheless, the proposed search will lead to a better result than the biased MMSE STS and for close to ML detections the amount of leaves with low λ_0 (also low $f\left(\mathbf{c}\right)$) is suitable large to determine valuable leaves.

F. List based Iterations

An approach for complexity reduced iterative detection-decoding, initially proposed in [1], is the reuse of results gathered during the first detection in order to skip the remaining tree searches. A candidate list gathered during the first detection can be used as principal component for the next iterations. The basic idea is to find without a-priori knowledge a big amount of low valued leaves with respect to λ_0 or $f\left(\mathbf{c}\right)$. Available a-priori information will mainly change the sequence of favorable nodes since it increases the metric of disadvantageous paths (5). If the candidate list is big enough, the new maximum a posteriori (MAP) leaf, or at least a leaf close to it, will be included in the list as well as suitable counter-hypotheses. A new tree search is hence not necessary. The adapted detection process is illustrated in Fig. 4. In order to compare list based iterations (LBIs)

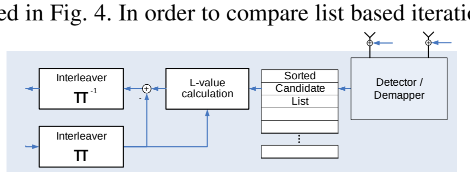


Fig. 4. Adapted receiver model for list based iterations

with the proposed detectors, we use a TS as basic detector for the list creation. Complexity comparison is performed by mapping the computational complexity of LBIs to the complexity of one node extension of the TS:

We define the complexity of one addition, substraction or comparison to 1 $\mathcal{O}$ and the complexity of a multiplication to 8 $\mathcal{O}$. We consider the described $N_T = 4, L = 6$ scenario with a detector performing 1.) all 64 or 2.) average 2 [1] node calculations per node extension. We further assume, a TS needs average 60 cycles respectively node extensions for one detection process. The influence of complexity resulting from the L-value calculation and the preprocessing is neglected.

The complexity of LBIs (list size K) is given by the complexity of metric updates and min-searches to $49K - 49$ $\mathcal{O}$.

The complexity of a TS, $T = 16, K = 64$, is given by the amount and complexity of metric calculations, given by (3), the complexity of resulting sortations/min-searches and a potential bias reduction to ca. 135000 $\mathcal{O}$ for 1.) or respectively ca. 7500 $\mathcal{O}$ for 2.), whereas this complexity might be further reduced by proper implementations.

The number of node extensions corresponding to the complexity of LBIs is given for particular list sizes in table I.

[1]Currently extended node + possible sibling or next parent node

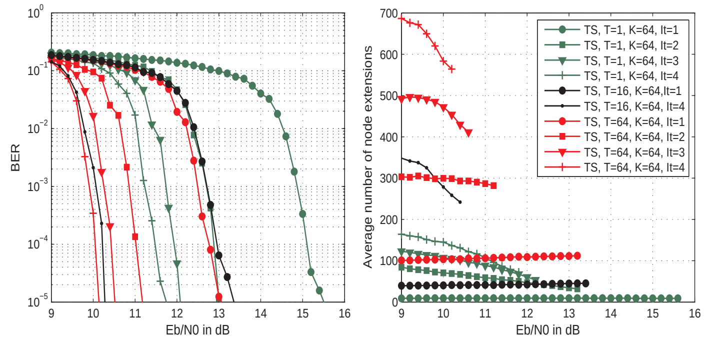
TABLE I

NUMBER OF NODE EXTENSIONS (NE) INVOLVING A COMPARABLE COMPLEXITY AS THE PARTICULAR LIST BASED ITERATION

Iteration over a List K =	64	128	256
Assumption 1.)	1.5 NE	3 NE	6 NE
Assumption 2.)	25 NE	50 NE	100 NE

IV. SIMULATION AND RESULTS

Figure 5 shows the complexity and performance of iterative detection LSD/TS for 1 to 4 iterations. The detector-decoder iterations



(a) BER performance (b) Complexity

Fig. 5. Comparison of iterative detection for a TS, 4×4 MIMO, 64QAM, SQRD and MMSE preprocessing and different number of iterations (It).

lead to a clear improvement of the performance with a saturation at the 3rd respectively 4th iterations. Due to the complexity of one tree search, also the overall detection complexity increases. A detection with an accuracy close to hard output ($T = 1, K = 64$) provides the lowest detection complexity but at poor BER-performance. Detection with maximum accuracy ($T = 64, K = 64$), comparable to a LSD enhanced by the proposed separated MMSE processing, incurs a much higher complexity but obtains a far better performance. The proposed tuple search with suitable tuple and candidate list sizes (e.g. 16/64) enables a detection performance close to optimum at significantly reduced complexity. As illustrated in Figure 6 the complexity of the tree search is heavily varying over the iterations. Depending on the accuracy of the first detection the complexity of the second iteration is enhanced. For a detector T=1, K=64 the second iteration is more than 4 times as complex as the first one. For more accurate detections the enhancement is reduced below a duplication of the initial complexity. This is caused by the fact, that the a-priori information enhances the sphere in case of "false" detection and an enhancement of an already large sphere has less affect on the complexity. The more the search result is inline with a-priori information reaching from the decoder (because of a better channel or a subsequent iteration) the less complex is the tree search.

Figure 7 provides a comparison of the proposed iterative TS with MMSE preprocessing and the proposed iterative STSs for 4 detection-decoding iterations. While the unbounded ZF STS provides the best

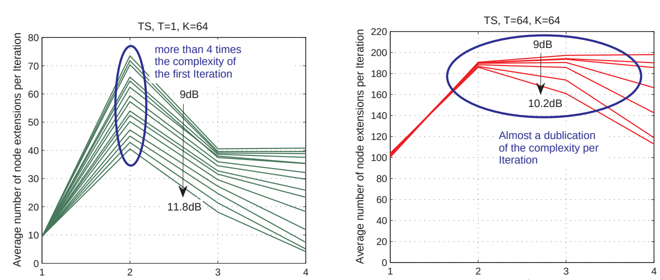
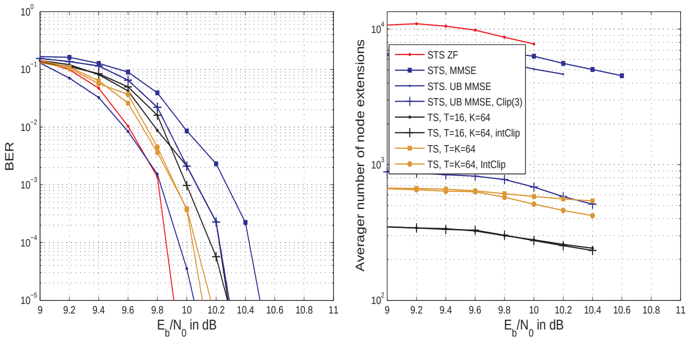


Fig. 6. Complexity distribution of the iterative detection

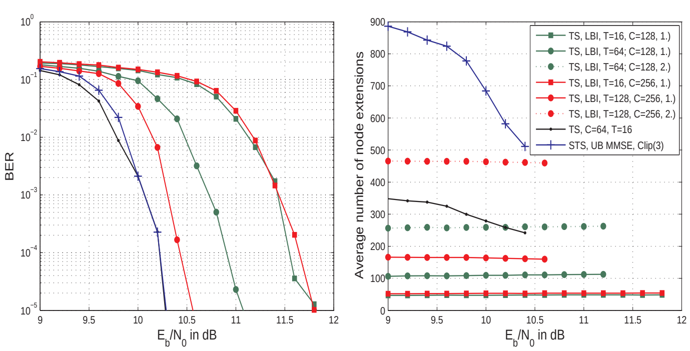


(a) BER performance (b) Complexity

Fig. 7. Performance and complexity of SISO STS and TS (4 det.-dec. It.)

performance its complexity is unfeasible high. MMSE processing enables a clear complexity reduction, but it also incurs a clear performance loss. By application of the proposed unbiased MMSE detection, separated to the underlying STS, this loss is minimized. The application of the proposed internal clipping to the STS allows a tradeoff between the performance and the complexity of the detection. This enables a detection close to optimum (<0.4dB loss) at significantly reduced complexity (almost factor 10). The usage of the TS with maximum accuracy (T=64, K=64) enables a detection at comparable complexity but higher accuracy. The proposed TS with adapted list sizes (e.g. 16/64, $\Rightarrow L_{\max} = 3.3$) leads to further complexity reduction, enabling a detection at the same performance but half the complexity as the clipped MMSE STS. Resulting from its definition, the internal clipping of a TS incurs no performance loss though a complexity reduction. This gain depends on the relation between accuracy of determined leaves to accuracy usable by the list size. If the accuracy of determined leaves is higher than usable by the list (caused by big tuples or low clipping values e.g. for simple detectors), useful information is dropped during the search. Consequently, internal clipping leads to a big complexity reduction. In case of TS adapted to the system, dropping useful information is almost avoided [5] and hence the clipping gain is small.

Finally, Fig. 8 compares the proposed iterative sphere searches with LBIs for 4 detector-decoder iterations, whereas the complexity of the later are given based on the estimations provided for 1.) and 2.) in table I. While the performance of the proposed algorithms is close to optimum, the achievable performance of the LBIs depends on the amount and quality of the determined candidates. As result, an accurate detection requires large lists in combination with a comprehensive tree search in the first iteration. In case of complex tree searches, case 1.) (solid lines), the LBIs lead to a reduced complexity. For more sophisticated tree searches, like case 2.) (dotted lines) or even more improved implementations, the proposed TS clearly outperforms the LBI's in performance as well as complexity.



(a) BER performance (b) Complexity

Fig. 8. Performance and complexity of list based iterative (LBI) detection

V. Conclusion

Iterative detection enables a great possibility for high accurate detection. Within this paper we have shown how SISO extension of the STS or the TS sphere detectors can be efficiently used for this purpose, outperforming the previously proposed list based iterations. Key therefore are searches based on sophisticated pruning strategies, like the proposed internal clipping and/or tailored search tuples, leading to a significant complexity reduction. Further complexity reduction is enabled by the proposed separated processing of unbiased MMSE candidates. Although the complexity of the resulting STS is reduced to only about 7% of the initial complexity, the proposed TS requires only half the complexity at comparable BER performance.

References

[1] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communications*, vol. 51, pp. 389–399, Mar. 2003.

[2] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 491–503, 2006.

[3] Y. de Jong and T. Willink, "Iterative tree search detection for MIMO wireless systems," *Communications, IEEE Transactions on*, vol. 53, no. 6, pp. 930–935, 2005.

[4] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: Algorithms and VLSI implementation," *submitted to IEEE Journal on Selected Areas in Communications*, Apr. 2007.

[5] B. Mennenga, A. von Borany, and G. Fettweis, "Complexity reduced Soft-In Soft-Out Sphere Detection based on Search Tuples," in *IEEE International Conference on Communications (ICC'09)*, Dresden, Germany, 2009.

[6] J. Jaldén and B. Ottersten, "Parallel Implementation of a Soft Output Sphere Decoder," in *Proceedings of Asilomar Conference on Signals, Systems, and Computers*, 2005.

[7] E. Zimmermann and G. Fettweis, "Unbiased MMSE Tree Search Detection for Multiple Antenna Systems," in *Proceedings of the International Symposium on Wireless Personal Multimedia Communications (WPMC'06)*, San Diego, USA, Sep. 2006.

[8] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Transactions on Information Theory*, vol. 48, pp. 2201–2214, Aug. 2002.

[9] S. Bäro, J. Hagenauer, and M. Witzke, "Iterative detection of MIMO transmission using a list-sequential (LISS) detector," *Proceedings of the IEEE International Conference on Communications, 2003. ICC'03.*, vol. 4, 2003.

[10] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal of Solid-State Circuits*, vol. 40, pp. 1566–1577, Jul. 2005.

[11] C. Schnorr and M. Euchner, "Lattice basis reduction: Improving practical lattice basis reduction and solving subset sum problems," *Mathematical Programming*, vol. 66, pp. 181–199, Aug. 1997.

[12] D. Wübben, J. Rinas, R. Böhnke, V. Kühn, and K. Kammeyer, "Efficient Algorithm for Detecting Layered Space-Time Codes," in *4th International ITG Conference on Source and Channel Coding (ITG SCC'02)*, Berlin, Germany, Jan. 2002.

[13] D. Wübben, R. Böhnke, V. Kühn, and K.-D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in *Proceedings of the IEEE 58th Vehicular Technology Conference, 2003 (VTC'03-Fall)*, vol. 1, 6.-9. Oct. 2003, pp. 508–512.

[14] J. Hagenauer, "The Turbo Principal in Mobile Communications," in *Proceedings of the International Symposium on Information Theory and its Applications (ISITA'02)*, Xi'an, China, 7.-11. Oct. 2002.

[15] C. Studer and H. Bolcskei, "Soft-input soft-output sphere decoding," *IEEE International Symposium on Information Theory, 2008 (ISIT'08)*, pp. 2007–2011, 2008.