Soft-Output Sphere Decoding: Single Tree Search vs. Improved K-Best

Martin Mayer, Michal Šimko and Markus Rupp
Institute of Telecommunications
Vienna University of Technology, Austria
Gusshausstrasse 25/389, A-1040 Vienna, Austria
Email: { mmayer, msimko, mrupp }@nt.tuwien.ac.at

***Abstract*—Multiple-Input Multiple-Output systems provide high multiplexing gain for digital transmissions. However, this is only achievable if an expedient detection method is used. A common method is Maximum Likelihood (ML) detection which enables soft decisions for each received bit along with good error performance. The drawback of this method is its demanding algorithm. In order to meet real-time constraints, the ML detection can be approximated. In this paper, we compare three different implementations of the soft sphere decoder: the single tree search which achieves true ML performance, a conventional k-best algorithm that delivers approximated ML detection, and a novel improved k-best algorithm with better ML approximation at cost of slightly increased execution time. We examine different performance aspects of these sphere decoder implementations and give a recommended complexity-border which indicates where the usage of an ML approximation becomes appropriate.**

***Index Terms*—LTE, MIMO, ML detection, sphere decoding, k-best, tree search, log likelihood ratio.**

I. INTRODUCTION

Modern wireless standards like UMTS Long Term Evolution (LTE) rely on Multiple-Input Multiple-Output (MIMO) systems due to their gain in spectral efficiency. Transmitter and receiver of these systems are equipped with multiple antennas which allow the transmission of multiple data streams concurrently in the same frequency band. However, the decoding effort heavily depends on the system complexity (e.g. number of antennas, modulation scheme) and on the channel realization. A trade-off between performance and computational complexity has to be found.

Maximum Likelihood (ML) detection achieves minimum error probability and is crucial for a solid MIMO detector implementation. Based on the ML solution, soft decisions for each received bit are possible, enabling improved error performance. The Sphere Decoding Algorithm (SDA) is a common tool to facilitate ML detection, but results in an exhaustive search in worst case. Its throughput is therefore not fixed as the computation time varies between executions. Given run time constraints, the true ML detection is rendered impractical above a certain level of system complexity. The k-best sphere decoder using ML approximation can be introduced as remedy, but a higher Bit Error Ratio (BER) has to be taken in account which is examined in this paper.

In [1], performance and implementation aspects of soft-output sphere decoding are discussed. We compare their proposed Single Tree Search (STS) [1, 7] with our implementations of a conventional k-best algorithm (KBA) and a novel improved k-best algorithm (KBI) with better ML approximation that reduces BER.

The remainder of this paper is organized as follows. In Section II, we explain how to transform ML detection and Log Likelihood Ratio (LLR) computation into a tree search problem. Furthermore, we briefly describe the STS algorithm and its benefits. In Section III, we illustrate our improved k-best algorithm. Simulation results of the three implementations are presented in Section IV, and we conclude in Section V.

II. SOFT-OUTPUT SPHERE DECODING

Consider a MIMO system with N_T transmit antennas and N_R receive antennas ($N_R \geq N_T$). The mathematical model can be written as

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

where $\mathbf{y}$ denotes the $N_R \times 1$ received symbol vector, $\mathbf{x}$ the $N_T \times 1$ transmitted symbol vector, $\mathbf{H}$ the estimated [2] $N_R \times N_T$ MIMO channel matrix and $\mathbf{n}$ the $N_R \times 1$ Additive White Gaussian Noise (AWGN) vector. Elements of the transmitted symbol vector $\mathbf{x}$ depend on the underlying symbol alphabet. The number of bits per symbol at an antenna is denoted by Q, the number of possible symbols by 2^Q. *In this paper, we assume that all antennas employ the same modulation scheme and that* $N_T = N_R$. The number of possible transmit symbol vectors is then determined by

$$N_S = 2^{Q \cdot N_T} \tag{2}$$

and increases exponentially with increasing bit number Q and increasing antenna number N_T.

The ML detector for the transmitted symbol vector $\mathbf{x}$ is defined as

$$\mathbf{x}^{\mathrm{ML}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2. \tag{3}$$

It detects the most probable transmitted symbol vector $\mathbf{x}^{\mathrm{ML}}$ by finding the smallest distance metric between received symbol vector $\mathbf{y}$ and each possible symbol vector $\mathbf{x}$ transmitted over estimated channel $\mathbf{H}$. Therefore, all N_S distance metrics have to be calculated. The Sphere Decoding Algorithm (SDA) can be introduced in order to improve detection speed. It transforms the ML detection into a tree search that constrains the detection by a search radius. How this transformation works is shown in [1]. Channel matrix $\mathbf{H}$ has to be QR-decomposed according to $\mathbf{H} = \mathbf{QR}$ where $\mathbf{Q}$ is unitary $N_R \times N_T$ and $\mathbf{R}$ is real-valued upper-triangular $N_T \times N_T$. Left-multiplying (1) by $\mathbf{Q}^H$ leads to the modified input-output relation

$$\widetilde{\mathbf{y}} = \mathbf{R}\mathbf{x} + \mathbf{Q}^H\mathbf{n} \quad \text{with} \quad \widetilde{\mathbf{y}} = \mathbf{Q}^H\mathbf{y}. \tag{4}$$

We obtain the new ML detector

$$\mathbf{x}^{\text{ML}} = \arg\min_{\mathbf{x}} \|\widetilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\|_2^2. \tag{5}$$

In terms of (5), if we start the metric calculations at the bottom row $i = N_T$ of matrix $\mathbf{R}$, we have to execute metric calculations only for the symbol alphabet of antenna $i = N_T$ in symbol vector $\mathbf{x}$ because $\mathbf{R}$ is upper-triangular and only contains one non-zero element $R_{N_T N_T} \neq 0$ in that row. Consider this the top layer calculation of a tree with N_T layers. The root node of the tree lies above $i = N_T$, and its distance is initialized to $d_{N_T+1} = 0$. The distance metrics (Euclidean distances) $d(\mathbf{x}) = \|\widetilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\|_2^2$ can be computed recursively with Partial Euclidean Distances (PEDs)

$$d_i = d_{i+1} + |e_i|^2 \quad , \quad i = N_T, N_T - 1, ..., 1 \tag{6}$$

and the Distance Increments (DIs)

$$|e_i|^2 = \left| \widetilde{y}_i - \sum_{j=i}^{N_T} R_{i,j} x_j \right|^2. \tag{7}$$

The DIs correspond to branches of the tree whereas the PEDs are associated with nodes. Each leaf node of the tree corresponds to a specific transmit symbol vector since it stores a unique set of symbols (determined by the paths taken) and contains its distance metric. According to (2), $2^{Q \cdot N_T}$ symbol vectors have to be calculated by a brute force solution. Fortunately, the tree search can be improved by SDAs like the Schnorr-Euchner Sphere Decoder (SESD) [5, 6]. It employs a depth first search with radius reduction, meaning everytime a leaf node is reached, the sphere-radius (initialized with ∞) is reduced to its distance metric in order to prune nodes and subtrees with higher PEDs in the next depth first search. It runs until only the ML leaf node is left in the sphere. Hence, execution time varies from only one depth first search in best case to exhaustive search in worst case.

Soft output MIMO detection requires the computation of LLRs for all coded bits. To reduce complexity, we apply *max-log approximation* [3]

$$L(b_n) = \min_{\mathbf{x}|b_n=0} \|\widetilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\| - \min_{\mathbf{x}|b_n=1} \|\widetilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\|, \tag{8}$$

where b_n denotes the n-th bit of the transmitted symbol vector. The total bit number of one symbol vector is obtained by $N_B = QN_T$. For each bit, one of the two minima in (8) is the *maximum likelihood distance* λ^{ML} we obtain by e.g. an SDA. Given λ^{ML}, the other minimum has to be calculated for all N_B bits, again by an SDA. We call these other distance minima the *counter hypotheses distances* λ_n^{CH} [1]. The max-log LLRs are now determined by

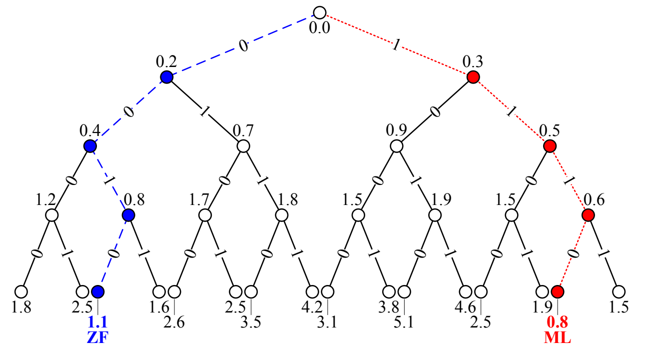

Fig. 1. Tree example for $N_T = N_R = 4$, $Q = 1$. The dashed line indicates branches to the Zero Forcing (ZF) leaf node, the dotted line to the Maximum Likelihood (ML) solution. The numbers next to the nodes denote the PEDs, the branch numbers the bit. By following the ZF path, we obtain $\mathbf{x}^{\text{ZF}} = [0\ 0\ 1\ 0]^T$ as symbol vector, for the ML path $\mathbf{x}^{\text{ML}} = [1\ 1\ 1\ 0]^T$ assuming that $b_n = 0$ corresponds to *symbol* 0, $b_n = 1$ to *symbol* 1.

$$L(b_n) = \begin{cases} \lambda^{\text{ML}} - \lambda_n^{\text{CH}}, & b_n^{\text{ML}} = 0 \\ \lambda_n^{\text{CH}} - \lambda^{\text{ML}}, & b_n^{\text{ML}} = 1 \end{cases}. \tag{9}$$

Summed up, we have to execute $QN_T{+}1$ sphere decoders in order to obtain the LLRs for one symbol vector by a repeated tree search [8]. More complex modulation schemes (increased Q) result in more necessary sphere decoder executions for the counter hypotheses.

The more efficient STS is introduced in [1, 7]. It ensures that every node in the tree is visited at most once by searching the ML solution *and* all counter hypotheses concurrently. The ammount of calculated nodes of the STS compared to the repeated tree search is significantly lower which leads to lower execution times. We compare this SDA implementation with our k-best implementations.

III. Improved K-Best Algorithm

The KBA employs breadth-first search instead of depth-first search like the STS. In terms of sphere decoding, this means that the sphere-radius is fixed for one execution, because we only reach leaf nodes in the end. Hence, we have to find an appropriate initial radius for our search. Since the KBA retains only the best nodes (k smallest PEDs) of each layer anyway, the search radius may not be important to obtain the (approximated) ML solution, but it helps reducing the number of calculated nodes in each layer which results in fewer calculations and faster sorting. Especially in the low Signal to Noise Ratio (SNR) region, the distance metric of the *Zero Forcing (ZF) solution* $\mathbf{x}^{\text{ZF}}$ shows good pruning results. We use it as *initial sphere radius* for *all* our sphere decoder implementations:

$$r_{ZF} = \sum_{i=N_T}^{1} \left| \widetilde{y}_i - \sum_{j=i}^{N_T} R_{i,j} x_j^{\text{ZF}} \right|^2 \tag{10}$$

In terms of a tree search, ZF solution means that we follow the branches with the smallest DIs from top layer ($i = N_T$)

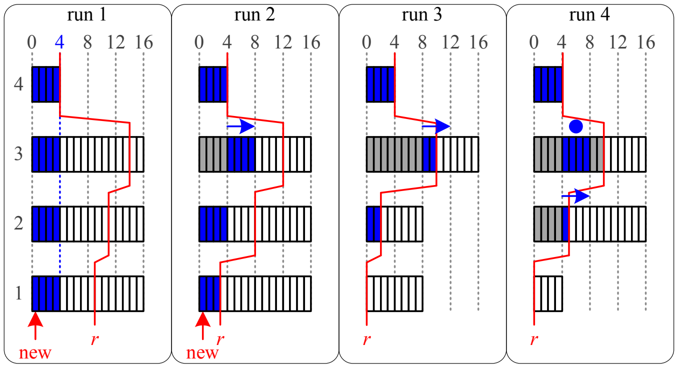

Fig. 2. Principal scheme of KBI for $N_T = N_R = 4$, $Q = 2$, $k = 4$-best. In *run 1*, the search radius is initialized to $r = r_{ZF}$ (10). Since we use 4-best, we have to calculate 16 nodes in each layer if no node is pruned by the radius (each node has $2^Q = 4$ children). Nodes are *sorted with ascending PEDs* in each layer. The conventional KBA would stop after the first run, the KBI reduces the search radius to the newly found distance and checks the next set of nodes in layer 3 in *run 2* (notice reduced radius). It turns out that nodes 5...8 in layer 3 contain an even better distance in the end, the radius is again decreased. Now we have to look at the two remaining (unpruned) nodes of layer 3 in *run 3*. They do not contain a smaller distance, we keep the node offset of *run 2* (which contained the best result) and advance to the next deeper layer in *run 4*. We look at one remaining node in layer 2 and do not find a smaller distance. This makes the solution found in *run 2* the best ML approximation.

to leaf node ($i = 1$). This is equivalent to a 1-best algorithm, since we retain only one (best) node in each layer. It is possible that the PED of a node in a higher layer is larger than others in the same layer, but by following a specific path the DIs become small in deeper layers which leads to the best PED (smallest distance metric) at the leaf node. This is shown in Fig. 1 and it is the challenge of the sphere decoder. Increased number of antennas and thus more tree layers make it harder to find the proper ML solution given a considerable noisy channel. This impairs the error performance of the k-best decoder because even if we employ relatively high k-values of retained nodes in each layer, it is possible that we miss out these with the smallest distance path in the end. To minimize the ammount of falsely discarded nodes, proposed KBI (shown in Fig. 2) also checks nodes in each layer that are outside the k-range but inside the pruning radius. The necessary runs are smartly reduced by effective pruning, because good ML approximation and hence sufficient pruning radius is usually found after the first run.

This is however still not the true ML solution, but approximates it much better than the conventional KBA. To obtain the true ML solution, we have to check all combinations of node packets within the radius which leads to high computing overhead again. Further improvements can be achieved by applying a sorted QR-decomposition [4].

IV. Simulations

In our KBI implementation, we utilize the improvement only to find λ^{ML}. For the counter hypotheses distances λ_n^{CH}, we employ the conventional KBA. Applying KBI on counter hypotheses would considerably increase execution time and does not improve BER significantly, so it turned out.

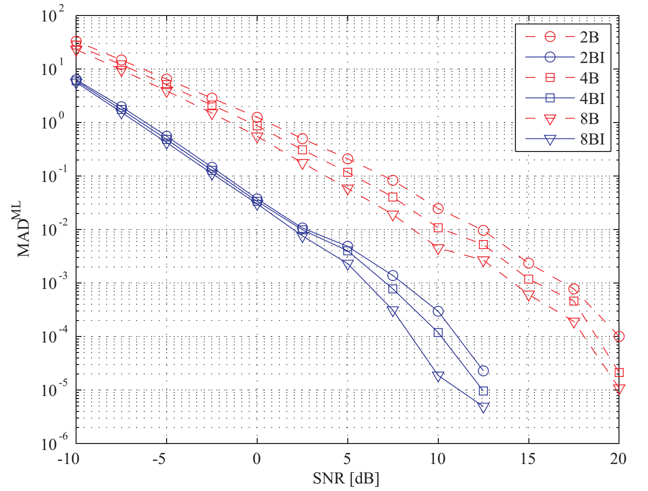

Fig. 3. Mean Absolute Deviation (11) between true ML distances (by STS) and approximated ML distances (by KB, KBI) with different k-values of maximum survivor nodes. 64-QAM system with $N_T = N_R = 4$.

Our simulations are averaged over a certain number of random channel realizations C with fixed noise variance. The simulation results depend on four major parameters: channel noise variance σ_n^2 (determines SNR), antenna number $N_T = N_R$, modulation scheme (number of bits Q) and maximum number of survivor nodes k. At first, we simulate a 64-QAM system ($Q = 6$) with Gray mapping and $N_T = N_R = 4$.

Fig. 3 shows the mean absolute deviation

$$\left|\text{MAD}^{\text{ML}}\right| = \frac{1}{C}\sum_{i=1}^{C}\left|\lambda_i^{\text{ML}} - \lambda_i^{\text{ML,approx}}\right| \qquad (11)$$

of true ML distances found with STS compared to the approximated ones. The deviation of KBI distances is considerably smaller than of k-best (KB), which confirms our improvements.

Fig. 4 shows the BER of the detectors. Based on the LLRs (9), bits are set according to $b_n = 1$ if $L(b_n) \geq 0$ and $b_n = 0$ if $L(b_n) < 0$. The obtained symbol vectors rely on these bit decissions. Due to better ML approximation, KBI outperforms KB as expected.

Fig. 5 shows the number of total calculated nodes during execution, which is proportional to the execution time. In the low SNR region, KBI has to calculate noticeable more nodes than KB, because distortions are higher which leads to more runs (worse pruning radius, see Fig. 2). In the high SNR region, KBI and KB are close together. The major ammount of calculated nodes is contributed by the λ_n^{CH} calculations where we employ the faster (non-improved) KBA. Proposed KBI improvement is confined to the λ^{ML} calculation, which is fast in the high SNR region. Although STS calculates λ^{ML} and λ_n^{CH} concurrently, the number of calculated nodes is higher than for KB/KBI, because it grants to find the true ML distance which is accomplished by an extensive search.

Fig. 6 shows how the number of calculated nodes depends on the number of antennas ($N_T = N_R$) and modulation

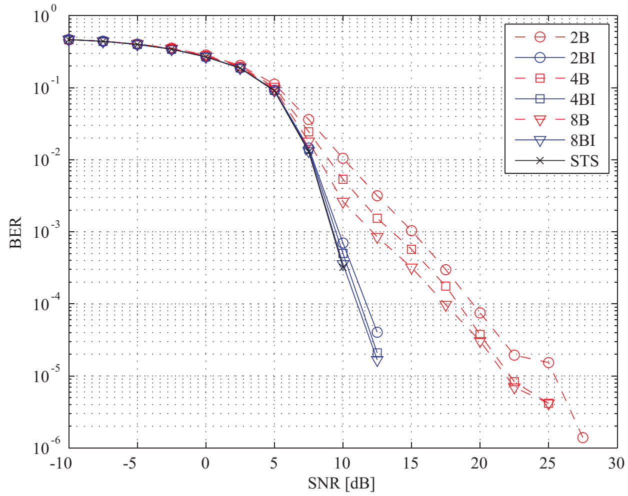

Fig. 4. Bit Error Ratio (BER) of 64-QAM system with $N_T = N_R = 4$. Single Tree Search (STS) is compared to k-best (KB) and improved k-best algorithm (KBI) using different k-values of maximum survivor nodes.

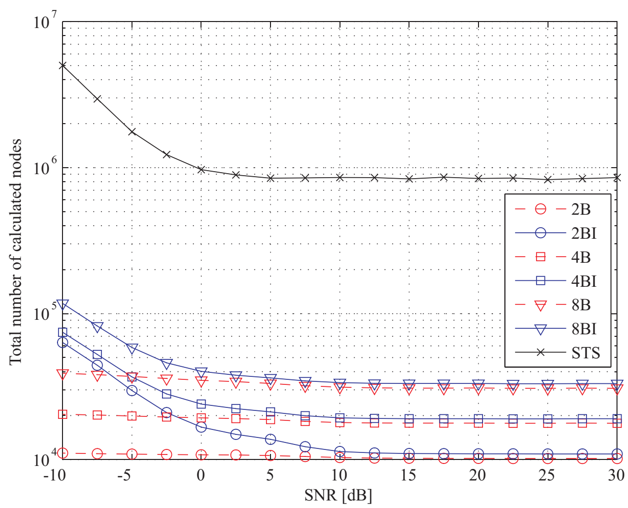

Fig. 5. Total number of calculated nodes using 64-QAM system with $N_T = N_R = 4$. Single Tree Search (STS) is compared to k-best (KB) and improved k-best algorithm (KBI) using different k-values of maximum survivor nodes.

scheme (bits per symbol Q). Ceteris paribus increments of $N_T = N_R$ result in more node calculations (higher execution time) since the number of possible symbol vectors (2) increases exponentially with the number of antennas. Assuming SNR $=$ 10dB and $k = 4$ best, the border where KBI needs fewer calculations than STS is found at the intersection of the graphs. Given a 16-QAM system ($Q = 4$), the usage of KBI is inappropriate below three antennas.

V. Conclusions

The sphere decoder is a versatile tool to implement ML detection in MIMO systems. However, its decoding throughput heavily depends on system complexity and channel realization. The KBA can be used to approximate ML detection with fixed throughput. We introduced an improved KBA with better ML approximation and only small overhead in complexity and execution time. At a certain point of system complexity,

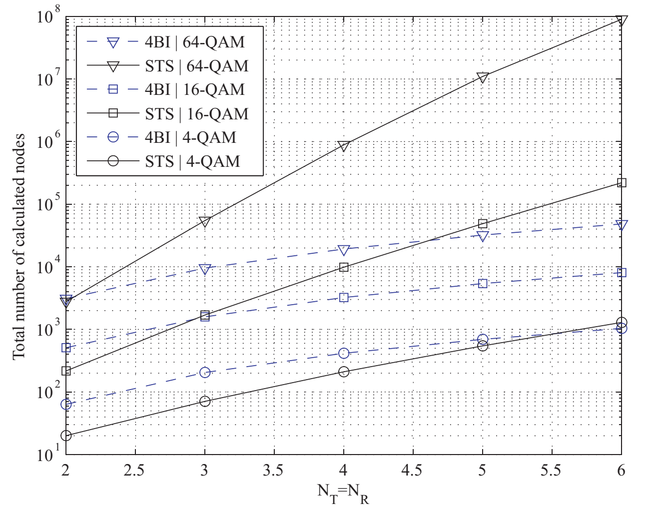

Fig. 6. Total number of calculated nodes depending on antenna number and modulation scheme. Maximum survivor node number k=4, SNR=10dB. Single Tree Search (STS) is compared to improved k-best algorithm (KBI).

STS can be exchanged with KBI in order to achieve higher decoding throughput at insignificant BER increment.

Acknowledgments

The authors would like to thank the LTE research group for continuous support and lively discussions. This work has been funded by the Christian Doppler Laboratory for Wireless Technologies for Sustainable Mobility, KATHREIN-Werke KG, and A1 Telekom Austria AG. The financial support by the Federal Ministry of Economy, Family and Youth and the National Foundation for Research, Technology and Development is gratefully acknowledged.

References

[1] C. Studer, M. Wenk, A. Burg and H. Bölcskei, “Soft-Output Sphere Decoding: Performance and Implementation Aspects”, in *Fortieth Asilomar Conference on Signals, Systems and Computers*, pp. 2071-2076, Nov. 2006.

[2] Michal Šimko, Di Wu, Christian Mehlführer, Johan Eilert and Dake Liu, “Implementation Aspects of Channel Estimation for 3GPP LTE Terminals”, in *Proc. 17th European Wireless Conference (EW 2011)*, Apr. 2011.

[3] B. M. Hochwald and S. ten Brink, “Achieving near-capacity on a multiple-antenna channel”, in *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389-399, Mar. 2003.

[4] D. Wübben, R. Böhnke, J. Rinas, V. Kühn and K. Kammeyer, “Efficient algorithm for decoding layered space-time codes”, in *IEEE electronic letters*, vol. 37, no. 22, pp. 1348-1350, Oct. 2001.

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

[6] E. Agrell, T. Eriksson, A. Vardy and K. Zeger, “Closest point search in lattices”, in *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 1620-1625.

[7] J. Jaldén and B. Ottersten, “Parallel implementation of a sphere decoder”, in *Proceedings Asilomar Conference on Signals, Systems and Computers*, pp. 581-585, Nov. 2005.

[8] R. Wang and G. Giannakis, “Approaching near-capacity with reduced-complexity soft sphere decoding”, in *Proc. of IEEE Wireless Communications and Newtorking Conf. (WCNC)*, vol. 3, pp. 1620-1625, Mar. 2004.