Low-Complexity Soft-Output Sphere Decoding with Modified Repeated Tree Search Strategy

Shin-Lin Shieh, *Member, IEEE,* Rong-Dong Chiu, Shih-Lun Feng, and Po-Ning Chen, *Senior Member, IEEE*

***Abstract*—Many solutions for detecting signals transmitted over flat-faded multiple input multiple output (MIMO) channels have been proposed, e.g., the zero-forcing (ZF), minimum mean squared error (MMSE), sphere decoding (SD) algorithms, to name a few. These approaches however suffer from either unsatisfactory performance or high complexity. In this paper, we focus on the soft-output SD algorithm and propose a modification on the repeated tree search (RTS) strategy. It is shown that our modification can maintain a fixed upper limit in decoding complexity and results in a good performance-complexity tradeoff.**

***Index Terms*—MIMO, sphere decoding, repeated tree search.**

I. Introduction

In recent years, wireless transmission with multiple antennas at the transmitter and receiver, also being referred to as the multiple input multiple output (MIMO) system, has attracted enormous interest. It is considered to be the technology that can provide significant capacity improvement over existing communication systems.

In an MIMO fading channel suffering additive white Gaussian noises (AWGN), different data streams are transmitted from different antenna elements via the same channel; so the receiver has to separate these data streams in order to recover them. Some detection algorithms for MIMO systems have thus been proposed and are reviewed in the following.

Linear detection methods, such as zero-forcing (ZF) or minimum mean-squared error (MMSE), estimate the information of channel matrix and then use the estimate to compensate the channel effect. Although having usually low computational complexity, the linear detection methods cannot totally remove the inter-stream interference and may induce noise enhancement; they accordingly could result in significant performance degradation. As a contrary, the brutal-force maximum likelihood (ML) detector is optimal in performance but its computational complexity is high. Being regarded as a balance of the previous two, the sphere decoding (SD) algorithm [1], [2] smartly reduces the number of candidate symbol vectors during the codeword search and can still statistically guarantee the finding of the ML solution with greatly reduced complexity.

Unlike uncoded systems where single hard-decision ML solution is directly outputted, the soft information for each information bit is required for iterative decoding. The soft-output SD algorithm [3], [4], [5] thus draws research attention recently. In comparison with the hard-output SD algorithm, the soft-output SD algorithm generally requires more computational complexity; therefore a particular method to reduce its complexity is necessary [4]. Some known complexity-reduction methods in literature include log-likelihood ratio (LLR) clipping, channel matrix regularization, and run-time constraint (i.e., imposing a constraint on the maximal computational complexity of the decoder). As expected, these methods reduce the decoding complexity at a price of performance degradation. The above complexity-reduction methods still yield a varying complexity; yet a hardware implementation may prefer one with a fixed complexity, which motivated the work of a fixed complexity soft-output SD in [5].

In this paper, we propose a modification on the repeated tree search (RTS) in [3], resulting in less performance degradation and also less complexity than those of the *single tree search* (STS) in [4], and the *smart ordering and candidate adding* (SOCA) algorithm in [5]. Moreover, our modification can maintain a fixed upper limit in decoding complexity and thus meet the requirement of hardware implementation.

The remaining of the paper are organized as follows. Section II introduces the system model and the existing MIMO detection approaches. Section III gives the detail of the soft-output SD algorithm. Section IV presents the idea of our proposed algorithm. Section V summarizes the simulation results, and Section VI concludes the paper.

Throughout the paper, superscripts "T" and "H" are reserved to denote the transpose and Hermitian transpose of a matrix, respectively.

II. System Model and Known MIMO Detectors

Consider an MIMO system with N_{T} transmit antennas and N_{R} receive antennas, where $N_{\mathrm{T}} \leq N_{\mathrm{R}}$. At the transmitter end, Q coded information bits are mapped to a complex constellation $\mathcal{O}$ (e.g., QPSK, 16-QAM, etc); hence, the number of constellation points is $|\mathcal{O}| = 2^Q$. The set of system vectors being transmitted is then given by $\mathcal{O}^{N_{\mathrm{T}}}$. Assume that the channel suffers flat fading. The received symbol vector thus can be written as:

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

where $\mathbf{x} \in \mathcal{O}^{N_{\mathrm{T}}}$ is the transmitted symbol vector with covariance matrix $\mathbb{E}\{\mathbf{x}\mathbf{x}^H\} = \mathbf{I}_{N_{\mathrm{T}}}$; $\mathbf{y} \in \mathbb{C}^{N_{\mathrm{R}}}$ denotes the received symbol vector; $\mathbf{n} \in \mathbb{C}^{N_{\mathrm{R}}}$ is an independent zero-mean Gaussian-distributed complex noise vector with common variance N_0 per entry; and $\mathbf{H} \in \mathbb{C}^{N_{\mathrm{R}} \times N_{\mathrm{T}}}$ denotes the $N_{\mathrm{R}} \times N_{\mathrm{T}}$ channel matrix with each element in $\mathbf{H}$ being a complex Gaussian variable with zero mean and variance $1/N_{\mathrm{T}}$. In the above expression, $\mathbb{C}$ denotes the domain of complex numbers, and each channel matrix realization $\mathbf{H}$ is assumed perfectly

Manuscript received August 2, 2012. The associate editor coordinating the review of this letter and approving it for publication was M. Lentmaier.

S.-L. Shieh is with the Graduate Institute of Comm. Eng., Nat'l Taipei Univ., Taipei, Taiwan 23741, R.O.C. (e-mail: slshieh@mail.ntpu.edu.tw).

R.-D. Chiu, S.-L. Feng, and P.-N. Chen are with Nat'l Chiao-Tung Univ., Taiwan 30010, R.O.C.

Digital Object Identifier 10.1109/LCOMM.2012.112012121728

estimated by the receiver. Note that in our setting, the signal-to-noise ratio (SNR) per receive antenna is equal to $1/N_0$.

The SD algorithm has recently been used to solve the signal detection problem for MIMO systems because it can significantly reduce the computational complexity in comparison with the brutal force ML detector while maintaining the same ML performance. The idea behind the SD algorithm can be described as follows. It first sets a sphere centered at the received symbol vector with a properly chosen radius. Then instead of searching the entire system constellation, it only searches the constellation points inside the sphere.

The SD algorithm can be separated into two steps: 1) preprocessing step and 2) tree search step. The preprocessing step is mainly on the construction of the tree structure. Specifically, the channel matrix $\mathbf{H}$ is $\mathbf{QR}$-decomposed with sorting and regularization [4] as:

$$\begin{bmatrix} \mathbf{H} \\ \alpha \mathbf{I}_{N_\mathrm{T}} \end{bmatrix} \mathbf{P} = \mathbf{QR} \tag{2}$$

where $\mathbf{Q}$ is an $(N_\mathrm{R}+N_\mathrm{T})\times N_\mathrm{T}$ unitary matrix, $\mathbf{R}$ is an $N_\mathrm{T}\times N_\mathrm{T}$ upper triangular matrix with diagonals being real-valued, and the $N_\mathrm{T}\times N_\mathrm{T}$ permutation matrix $\mathbf{P}$ and the real parameter α are carefully chosen to fit the need of sorting and regularization [4], [5]. Partition $\mathbf{Q}$ into $\mathbf{Q}_1$ and $\mathbf{Q}_2$ according to

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1^\mathrm{T} & \mathbf{Q}_2^\mathrm{T} \end{bmatrix}^\mathrm{T}$$

where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are respectively $N_\mathrm{R}\times N_\mathrm{T}$ matrix and $N_\mathrm{T}\times N_\mathrm{T}$ matrix. Then, multiplying (1) by $\mathbf{Q}_1^\mathrm{H}$ leads to a modified input-output relation as

$$\tilde{\mathbf{y}} = \mathbf{Q}_1^\mathrm{H}\mathbf{y} = \mathbf{Q}_1^\mathrm{H}\mathbf{H}\mathbf{x} + \mathbf{Q}_1^\mathrm{H}\mathbf{n} = \mathbf{R}\tilde{\mathbf{x}} + \tilde{\mathbf{n}}$$

where $\tilde{\mathbf{x}} = \mathbf{P}^T\mathbf{x}$ and $\tilde{\mathbf{n}} = \mathbf{Q}_1^\mathrm{H}\mathbf{n}$. In matrix form, the above equation can be written as

$$\begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_{N_\mathrm{T}} \end{bmatrix} = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,N_\mathrm{T}} \\ 0 & r_{2,2} & \cdots & r_{2,N_\mathrm{T}} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{N_\mathrm{T},N_\mathrm{T}} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_{N_\mathrm{T}} \end{bmatrix} + \begin{bmatrix} \tilde{n}_1 \\ \vdots \\ \tilde{n}_{N_\mathrm{T}} \end{bmatrix}.$$

By treating $\tilde{\mathbf{n}}$ to be independent Gaussian distributed, it can be obtained that

$$\begin{aligned} \hat{\mathbf{x}}_\mathrm{ML} &= \arg \min_{\tilde{\mathbf{x}}\in\mathcal{O}^{N_\mathrm{T}}} \| \tilde{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{x}} \|^2 \\ &= \arg \min_{\tilde{\mathbf{x}}\in\mathcal{O}^{N_\mathrm{T}}} \sum_{i=1}^{N_\mathrm{T}} \left| \tilde{y}_i - \sum_{j=i}^{N_\mathrm{T}} r_{i,j}\tilde{x}_j \right|^2 . \end{aligned} \tag{3}$$

The second step then performs tree traversal based on (3). There are three major tree traversal algorithms that have been proposed in the literature, which are depth-first search [4], breadth-first search [6], and best-first search [7] algorithms.

III. Soft-Output Sphere Decoding and Methods for Complexity Reduction

Denote by $\tilde{x}_{j,b}$ the bth bit in the constellation point corresponding to the jth entry of vector $\tilde{\mathbf{x}}$. In order to reduce the computational burden, we approximate the true LLR for bit $\tilde{x}_{j,b}$ by its *max-log approximation* [4]:

$$L(\tilde{x}_{j,b}) = \min_{\tilde{\mathbf{x}}\in\mathcal{X}_{j,b}^{(0)}} \| \tilde{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{x}} \|^2 - \min_{\tilde{\mathbf{x}}\in\mathcal{X}_{j,b}^{(1)}} \| \tilde{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{x}} \|^2, \tag{4}$$

where $\mathcal{X}_{j,b}^{(0)}$ and $\mathcal{X}_{j,b}^{(1)}$ are sets of vectors that have the bth bit in the jth entry equal to 0 and 1, respectively. The computation of Eq. (4) can be done via a tree search process, after which the resulted LLRs should be permuted back to the $\mathbf{x}$-domain using the relation of $\tilde{\mathbf{x}} = \mathbf{P}^T\mathbf{x}$.

Several tree traversal strategies have been proposed for the generation of the LLR values. They are described below.

1) Repeated Tree Search (RTS): The idea behind the RTS strategy [3] is to repeat the tree search to compute the value of (4) for every bit in the symbol vector based on the ML solution located by the hard-output SD algorithm. Its main drawback is that some branch computations may be performed more than once, resulting in significant complexity waste.

2) Single Tree Search (STS): The STS is a more efficient tree search strategy when it is compared with the RTS. It ensures that every node in the tree is visited at most once; hence it reduces considerably the computational complexity. In particular, the STS [4] searches the ML solution as well as its corresponding counter-hypothesis paths concurrently in a depth-first fashion. Whenever a leaf is reached, two situations will be considered: If the leaf updates the temporary ML solution, the metric of the former temporary ML solution can serve as the metric of a counter-hypothesis of the new temporary ML solution, and if however no new temporary ML solution is found, the algorithm checks whether better counter-hypotheses of the current temporary ML solution have been traversed and the LLRs are updated accordingly.

3) Smart Ordering and Candidate Adding (SOCA) Algorithm: In [5], a soft-output SD algorithm named SOCA was proposed. By performing QR decomposition with a smart ordering criterion and also by adding pre-defined numbers of layer-by-layer candidates for its breadth-first search, it achieves a good performance-complexity tradeoff. A striking characteristic of the SOCA is that it has a fixed complexity. This makes it well suited for hardware implementation.

There are many subsequent researches focusing on further complexity reduction of the tree search algorithms mentioned previously. Additional complexity reduction enhancements such as LLR clipping, sorting and regularization are subsequently proposed [3], [4], [5].

IV. Modified RTS Traversal Strategy

In this work, we propose to modify the RTS soft-output SD algorithm such that the computational complexity can be limited by a pre-defined number. It is observed that better performance-complexity tradeoff can be resulted in comparison with the STS and the SOCA.

Similar to the original RTS, the proposed soft-output SD algorithm is separated into two stages. In the first stage, a hard-output solution is found by an SD algorithm. We set a maximum allowable complexity T_1 for the first stage so that only near-ML hard-output is guaranteed. After finding the near-ML hard-output path, we repeat the tree traversal in the second stage to generate soft output with a fixed complexity upper limit T_2. By pre-defining T_1 and T_2, our modified RTS algorithm can guarantee to generate soft output with complexity no more than $(T_1 + T_2)$. Details are given below.

One suitable candidate for the first stage is the Schnorr-Euchner sphere decoder (SESD) with radius reduction [10].

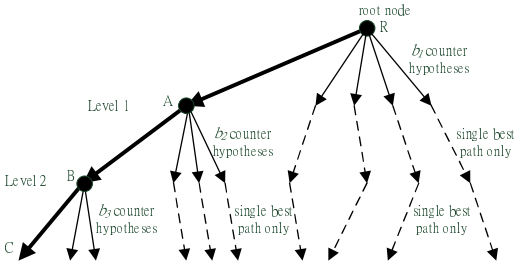

Fig. 1. Illustration of the second stage with $N_T = 3$ and $\mathbf{b} = [b_1\ b_2\ b_3] = [432]$ for the proposed modification. The thick solid line corresponds to the near-ML path obtained from the first stage.

It is known that the SESD has variable and potentially high decoding complexity; hence it is not suitable for hardware implementation. We therefore propose to set an upper limit T_1 such that the first stage is terminated when either the SESD finds the ML solution within complexity T_1 or the maximum allowable complexity T_1 is reached. With a complexity constraint, the hard-decision output no longer guarantees to be ML. Nevertheless, we will later show by simulations that a small to medium value of T_1 is adequate to secure a good performance-complexity tradeoff.

We next describe the second stage. First, a vector $\mathbf{b} = [b_1, \cdots, b_{N_\mathrm{T}}]$ that specifies the number of counter hypothesis paths to be extended at each level is given, where $1 \le b_i \le Q$. Note that at each level, there are Q counter hypothesis paths corresponding to the near-ML path obtained from the first stage; but only the best b_i of them are extended at level i.[1] Further extension along these b_i counter-hypothesis paths only include the best path. For a better understanding, a simple illustration of the second stage is given in Fig. 1.

For all soft-output SD algorithms, clipping the LLR to make it within $\pm L_{\max}$ plays an important role for performance, complexity or both [4], [5]. In this work, the counter hypothesis paths traversed by our modified RTS strategy may not be the ones required by (4) since usually $b_i < Q$. In such case, the max-log approximated LLR will be infinity for these non-traversed paths. The selection of clipping limit $L_{\max}$ thus is essential in our modification. Notably, except for the RTS, the situation that some of the counter hypothesis paths required are not visited could also happen to the STS and the SOCA; hence, these two schemes also need a clipping limit to prevent from overestimation of the LLR values. In all cases, LLR clipping can at the same time help reducing the complexity of the second stage.

The upper complexity limit of the second stage can be computed as follows:

$$T_2 = \sum_{i=1}^{N_\mathrm{T}} b_i \left(N_\mathrm{T} + 1 - i\right). \tag{5}$$

By only expending those nodes with branch metric within $L_{\max}$, the complexity of the second stage can be further reduced and is usually smaller than T_2.

We close this section by stress the differences between our modified RTS and the SOCA. First, our modified RTS finds the near-ML hard-output path and compute the LLR values in different stages, whereas the SOCA obtain both concurrently in one tree search. A second difference is that the SOCA adds Q counter hypothesis paths at all levels in a simple bit-flip fashion as only a portion of the MAP path is known before the selection of these counter hypotheses, whereas our modified RTS extends only b_i counter hypothesis paths at level i with respect to a completely known near-ML path.

V. Simulation Results

In our simulations, we assume that the MIMO channels are Rayleigh faded without spatial or temporal correlation, and all channel matrix realizations can be perfectly estimated by the receiver. Also, $N_\mathrm{T} = 4$ transmit antennas, $N_\mathrm{R} = 4$ receive antennas, and 16-QAM constellation are considered. In addition, the channel realizations do not change during the transmission of an entire codeword in the simulated slow fading scenario, as identically assumed in [5].

The outer codes adopted [8] are 3GPP-specified $(2, 1, 8)$ convolutional code and punctured turbo code of code rate $R = 1/2$. For the convolutional coded simulations, 180 16-QAM symbols are fed into a 15×48 block interleaver before they are sent, while for the turbo coded simulations, 500 16-QAM symbols are transmitted after being passed through a (40×50)-block interleaver. As a result, the codeword lengths for the convolutional and turbo coded systems are respectively $180 \times 4 = 720$ bits and $500 \times 16 = 2000$ bits. At the receiver, the Viterbi decoder and the 8-iteration Max-Log-MAP decoder are used respectively for the decoding of convolutional code and turbo code. Simulation results for convolutional and turbo coded systems are then summarized in Figs. 2 and 3, respectively.

In our simulations, various $L_{\max}$ values are tested for the STS, while $L_{\max} = 0.3$ and $L_{\max} = 0.25$ are chosen as suggested by our trial simulations not shown in this paper for the SOCA and our modified RTS, respectively. The channel regularization algorithm used in our modified RTS is the same as that in [4]. As for the sorting approach in QR decomposition, the SQRD [4] is used for both the STS and our modified RTS, while the SOQR [5] is implemented for the SOCA. The maximum allowable complexity for the first stage is $T_1 = 30$, and various vectors $\mathbf{b}$ required for the second stage are examined, which are $\mathbf{b} = [4444]$, $[4442]$, $[4422]$, $[4222]$, $[2222]$; they respectively result in $T_2 = 40$, 38, 34, 28, 20. Finally, the metric for the system performance after decoding is the SNR required to achieve a block error rate (BLER) of 10^{-2}. The index for computational complexity is the number of visited nodes during the tree search; this complexity index is widely adopted for one-node-per-cycle

[1]Take 16-QAM (hence, $Q = 4$) as an example. If $\hat{x}_i = 0000$ is the i-th symbol of the near-ML path, then the four counter hypothesis paths are specified by $\{1000, 0100, 0010, 0001\}$. Among them, the b_i counter hypothesis paths selected are the ones that are closest in distance to the received complex value $\tilde{y}_i$.

It should be mentioned that the SOCA also pre-defines numbers of layer-by-layer candidates for its breadth-first search; however, by considering the performance-complexity tradeoff, the selective choice for the SOCA is $b_2 = b_3 = \cdots = b_{N_\mathrm{T}} = 1$. That is why only b_1 is identified for the SOCA in Figs. 2 and 3.

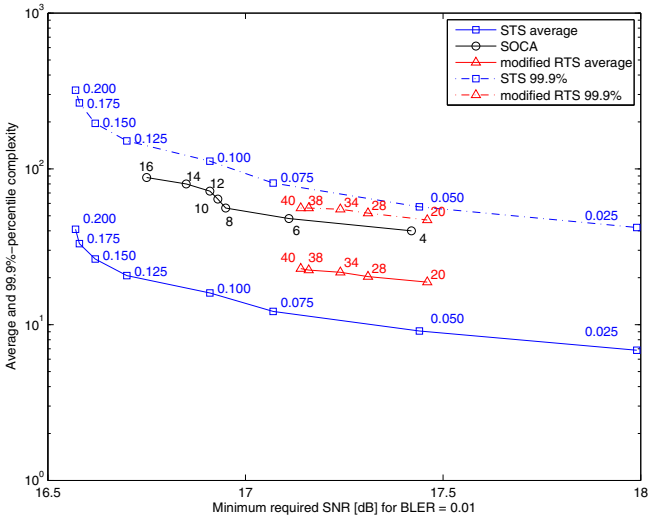

Fig. 2. Performance versus complexity for the STS, the SOCA, and the modified RTS in slow Rayleigh fading channels. The numbers beside the STS marks are the $L_{\max}$ used. The numbers next to the SOCA curve correspond to b_1. The number next to each modified RTS mark is T_2. The channel code adopted is the convolutional code.

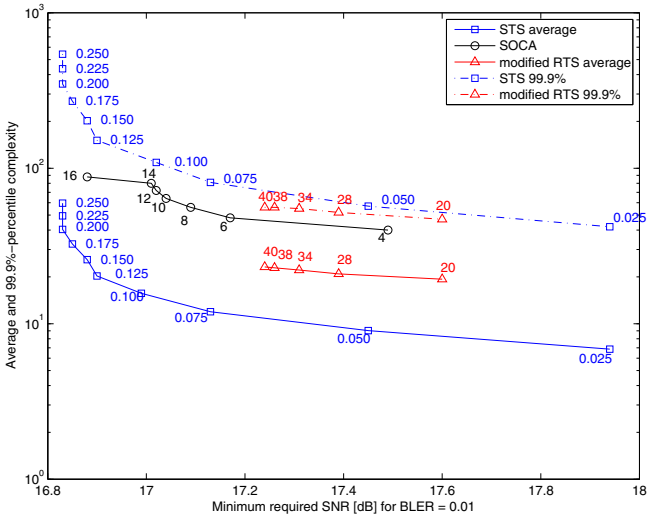

Fig. 3. The same simulations as Fig. 2 except that the channel code adopted is the turbo code.

hardware implementation architecture [9]. Based on the above setting, we are ready to present our simulation results.

As observed from Fig. 2 and Fig. 3, the simulation results for convolutional and turbo coded systems are quite similar. A result not shown in these two figures is that the performance degradation from $T_1 = \infty$ down to $T_1 = 30$ is less than 0.05 dB; this is the basis of our claim that a small to medium T_1 is adequate to secure a good performance for our modified RTS. For a complete comparison, we also show the 99.9th percentile complexities in the two figures. Notably, a highly variant decoding complexity will add difficulties to the hardware implementation of a decoding algorithm. From this regard, the 99.9th percentile complexity can serve as an assessment index for hardware design.

These two figures then show that the STS achieves the best performance-complexity tradeoff in slow fading scenario when only the average complexity is considered; however, its 99.9th percentile complexity is the worst among all simulated schemes. In particular, the 99.9th percentile complexity of the STS is six times more than its average complexity in both figures. Thus, the STS has a large complexity variation. On the contrary, our modified RTS, whose 99.9th percentile complexity almost approaches its strict upper complexity bound (T_1+T_2), turns out to have a much smaller complexity variation than the STS. Note that the SOCA has a fixed decoding complexity so that its 99.9th percentile complexity is identical to its average complexity. Since the curve of the 99th percentile complexity (equivalently, the upper complexity bound) of our modified RTS is only slightly above the curve of the SOCA, and since a hardware designer may use this upper complexity bound as its design criterion, we can conclude that our modified RTS requires a similar hardware complexity to the SOCA. As a result, the SOCA and our modified RTS remain to be more attractive solutions for hardware implementation because of their low complexity variation.

VI. Conclusion

In this paper, we present a modified RTS algorithm for soft detection in an MIMO system as a support to an outer code. A visible performance-complexity tradeoff improvement has been obtained by our proposed modification. The guaranteed decoding complexity upper limit makes this modified RTS algorithm a suitable candidate for hardware implementation.

References

[1] 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, Sep. 1994.

[2] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, vol. 44, pp. 463–471, Apr. 1985.

[3] R. Wang and G. Giannakis, "Approaching MIMO channel capacity with reduced-complexity soft sphere decoding," in *Proc. 2004 IEEE Wireless Communications and Networking Conf.*, vol. 3, pp. 1620–1625.

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

[5] D. L. Milliner, E. Zimmermann, J. R. Barry, and G. Fettweis, "A fixed-complexity smart candidate adding algorithm for soft-output MIMO detection," *IEEE Trans. Signal Process.*, vol. 3, no. 6, pp. 1016–1025, Dec. 2009.

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

[7] M. Myllyla, M. Juntti, and J. R. Cavallaro, "Architecture design and implementation of the increasing radiusXList sphere detector algorithm," in *Proc. 2009 IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 553–556.

[8] 3rd Generation Partnership Project, "Multiplexing and channel coding (FDD)," 3GPP Tech. Spec., TS 25.212 V11.1.0, Mar. 2012.

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

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