A LOW-COMPLEXITY ITERATIVE MIMO SPHERE DECODING ALGORITHM

Mansour Rachid, Babak Daneshrad

Electrical Engr. Dept., University of California – Los Angeles
Los Angeles, CA 90095, USA
phone: + (1) 310-752-3759, email: mrachid@ucla.edu

ABSTRACT

In this work we present an iterative multiple-input multiple-output (MIMO) sphere decoding algorithm based on a proposed Constrained Metric-first search. The search strategy minimizes the number of required iterations as well as the variation in the number of iterations while overcoming the conventional metric-first memory requirements. Further complexity reduction is achieved through the use of a simplified distance norm and sorted QR-decomposition. The proposed algorithm is shown to be better suited for early termination schemes employed to guarantee high throughput as compared to traditional sequential sphere decoding. The decoder is synthesized to a standard TSMC 65nm CMOS process and shown to guarantee 750 Mbps throughput for a 4x4 16-QAM setup with close-to ML (Maximum Likelihood) performance and lower complexity than published decoders.

1. INTRODUCTION

Sphere decoding (SD) [1]–[2] emerged as a promising method to find the optimum ML solution [3] for the MIMO decoding problem [4] by reformulating the impractical exhaustive search over all possible transmitted vectors into an efficient depth-first tree search. However, the complexity of the depth-first search employed by the SD algorithm varies with channel and noise conditions and can reach the complexity of an exhaustive search in the extreme [5]. This variation naturally appears in the throughput of the iterative decoder rendering it impractical.

In [6], early termination techniques were applied to the original SD algorithm to guarantee upper and lower bounds on complexity and throughput respectively. In [7], the SD algorithm with early termination was applied to the simplified tree of the Fixed-throughput Sphere Decoder (FSD) [8] in what is referred to as the sequential COSIC (Conditioned Ordered Successive Interference Cancellation) decoder. The sequential COSIC exhibits, to our knowledge, the minimum published complexity among MIMO sphere decoding algorithms. We measure complexity through the throughput normalized to gate count and chip-clock frequency (bps/Hz/G) metric.

We show in this work that under realistic channel conditions, depth-first based iterative decoders suffer severe performance degradation when early termination schemes are applied to guarantee minimum throughput. This motivated proposing an alternative MIMO sphere decoding algorithm.

The proposed algorithm involves the introduction of the Constrained Metric-first Search (CMS), a variant of Dijkstra's Search [9], which significantly reduces the maximum memory requirement that has prevented hardware implementation of Dijkstra's search for sphere decoding. We apply the proposed search to the FSD [8] tree using the ℓ^∞ norm. The latter configuration allows for significant reduction in the complexity of the search and the enumeration overhead. We show that the proposed search is very well suited for early termination schemes used to guarantee throughput. The algorithm is coded in HDL and synthesized to a TSMC 65nm CMOS process showing about 2x improvement in normalized throughput over the next best approach.

Section 2 explains the MIMO sphere decoding problem. Section 3 describes our proposed search strategy and chosen distance norm. Section 4 describes the simplified tree structure and enumeration scheme. In section 5 we discuss early termination schemes and the performance of the proposed algorithm applied to the case of a 4x4 64-carrier OFDM system. Section 6 presents the low hardware complexity of the proposed decoder when synthesized and compared to published results. We conclude in section 7.

2. MIMO DECODING

Consider a M_TxM_R MIMO system with spatial multiplexing. Let $\mathbf{s}$ be the transmitted vector of M_T independent symbols chosen from a given constellation Ω. The received vector $\mathbf{y}$ of length M_R is given by:

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

where $\mathbf{H}$ is the M_RxM_T channel matrix and $\mathbf{n}$ is the length-M_R noise vector at the receiver. MIMO decoding attempts to find the M_T transmitted symbols in $\mathbf{s}$ given $\mathbf{y}$ and $\mathbf{H}$. Without loss of generality, we assume that $M_R = M_T = M$.

The Maximum Likelihood (ML) solution [2] is the optimum estimate $\hat{\mathbf{s}}$ out of Ω_M where:

$$\hat{\mathbf{s}} = \arg\min d(\mathbf{s}) \quad (1)$$
$$\text{and } d(\mathbf{s}) = \left\|\mathbf{y} - \mathbf{H}\mathbf{s}\right\|^2$$

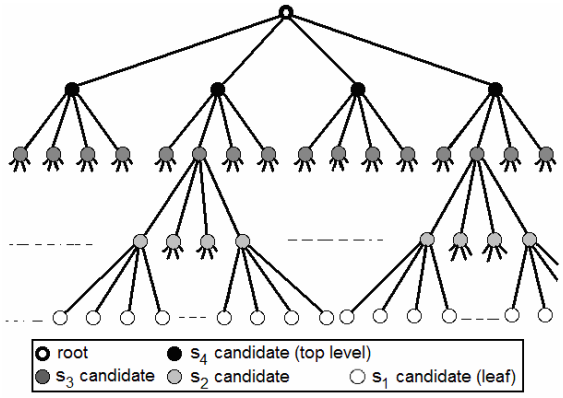

Figure 1 –Sample sphere decoder tree for a 4x4 QPSK system. Note that only part of the tree is shown.

Finding the ML solution requires a search over all possible combinations of $[s_M, s_{M-1}, ..., s_1]$. The number of candidate vectors grows exponentially with the number of antennas (64^4 for a 4x4 64-QAM system) and an exhaustive search becomes impractical.

The search, however, can be formulated into a tree search by performing QR-decomposition on the channel matrix **H**:

$$\mathbf{H} = \mathbf{QR}$$

where Q is a unitary matrix and R is an upper triangular matrix. The distance d(**s**) in (1) becomes:

$$d(\mathbf{s}) = \left\| \tilde{\mathbf{y}} - \mathbf{Rs} \right\|^2$$

where $\tilde{\mathbf{y}} = \mathbf{Q}^H \mathbf{y}$.

Assuming $\mathbf{s} = [s_M, s_{M-1}, ..., s_1]$ and $\tilde{\mathbf{y}} = [\tilde{y}_M, \tilde{y}_{M-1}, ..., \tilde{y}_1]$, we obtain an M –level tree with partial distances:

$$\begin{aligned} d(s_M) &= \left\| \tilde{y}_M - R_{MM} s_M \right\|^2 \\ d(s_{M-1}) &= \left\| \tilde{y}_{M-1} - R_{(M-1)M} s_M - R_{(M-1)(M-1)} s_{M-1} \right\|^2 \\ &... \\ d(s_1) &= \left\| \tilde{y}_1 - \sum_{K=2}^{M} R_{1K} s_K - R_{11} s_1 \right\|^2 \end{aligned} \quad (2)$$

where the total distance $d(\mathbf{s}) = \left\| \tilde{\mathbf{y}} - \mathbf{Rs} \right\|^2 = \sum_i d(s_i)$.

Finding the ML solution thus translates to finding the path in the tree with minimum distance $d(s_1)$ at the leaf. The search for the path utilizes the partial distance equations (2). Figure 1 shows the tree structure for a 4x4 QPSK system.

Sphere decoding [1] performs a *depth-first* search [10] over the tree by visiting child nodes before sibling nodes. Any node whose distance exceeds a certain radius *d* falls outside the sphere and is automatically pruned along with its children and siblings (if the latter are enumerated [11]). If a leaf inside the sphere of radius *d* is reached, the radius is updated as the distance to that leaf. When no unvisited nodes are inside the sphere, the path to the latest visited leaf is the solution. Accordingly, depth-first search might traverse paths whose distance exceeds the distance to the solution before the radius *d* reaches its final value.

3. CONSTRAINED METRIC-FIRST SEARCH

We propose an alternative search strategy that minimizes the number of iterations during the decoding of a received vector **y**. The search is based on Dijkstra's search algorithm [9]. Dijkstra's search finds the shortest path between two vertices on a graph of positive edges using a metric-first strategy. By treating all the leaves as a single node, the algorithm can find the ML solution as the shortest path from the root to a leaf as described in table 1.

Table 1 –Dijkstra's search in a sphere decoding scenario

Initialize	Boundary database D_B = {all top-level nodes}.
	Path database D_P = {root}.
Step 1	Find node X in D_B with the least distance.
	If X is a leaf then
	This leaf is the closest leaf. The solution is the path from the root to this leaf. The path is found in D_P. *Search ends.*
	Else
	Step 2 Add X to D_P.
	Step 3 Expand X, add its children to D_B.
	Go to step 1.

As can be seen from table 1, the algorithm visits the closest node on the boundary at every iteration. This is equivalent to visiting all nodes in the tree within a sphere of radius *d* equal to the distance to the leaf that is the solution. This particular value of *d* is the final and minimum sphere radius in a depth-first search (discussed in section 2). Thus, as is shown in [12], Dijkstra's algorithm minimizes the number of visited nodes required to find the ML solution.

Dijkstra's search, however, has been hitherto avoided in hardware implementation for MIMO sphere decoding due to the large memory size that databases D_B and D_P might require. In the worst case, the search might visit all the nodes in the tree except those on the lowest level. Thus, as the number of antennas increases, the maximum size of D_P and D_B grows exponentially and the complexity of sorting D_B grows linearly too. Furthermore, the number of children to be expanded per iteration grows linearly with constellation size.

We propose the *Constrained Metric-first Search* (CMS) that enforces two constraints on Dijkstra's search:

a. Limit the maximum size of D_B and accordingly D_P to some value N where boundary nodes that are not among the current best N are disposed of along with their paths.

b. When expanding a node only add its best child and best sibling to D_B.

Note that constraint (b) has no effect on the performance of the search. It allows however for significant complexity reduction by avoiding unneeded expansions and supports the feasibility of constraint (a).

We investigate the performance of the proposed search strategy as compared to the depth-first SD under the *ℓ^∞ norm* described in table 2. The ℓ^∞ norm in particular is the most suitable distance norm for iterative sphere decoding:

a. Partial distance calculation and accumulation in the ℓ^∞ norm require a comparison each as compared to multiplications and/or additions in the cases of the Euclidean (ℓ^2) and the ℓ^1 norms (or hybrid norms).
b. The number of visited nodes during a search is reduced by the ℓ^∞ norm because of the low cumulative distance (maximum instead of sum).
c. The SNR penalty incurred due to the ℓ^∞ norm approximation is small, e.g. ~ 1dB at a BER of 10^{-3} (Figure 2).

Figure 2 shows the BER curves obtained for the proposed search strategy for a 4x4 16-QAM setup. Also shown are the ML solution (ℓ^2 norm) and the unconstrained ℓ^∞ norm solution (depth or metric-first).

The results show that with N = 8 (only 8-paths in memory), the performance of the proposed Constrained Metric-first search is indistinguishable from the unconstrained depth or metric-first search under the ℓ^∞ norm. Note that unconstrained metric-first search can hold up to 16^3 paths in memory for this particular setup.

Figure 3 shows a results sample for a 16-QAM 4x4 setup at 20dB SNR, we note 3 advantages of CMS over depth-first:

a. Lower average number of visited nodes as expected.
b. Particularly fewer visits of nodes that are the closest among their siblings. This is important because visiting the first node in a set of siblings is more costly as can be seen from the partial distance equations (2) or their ℓ^∞ norm equivalents (3). This result is also expected since the CMS provides a trade-off between depth and breadth-first directions.

$$d(s_M) = \max\{\mathrm{Re}(\hat{y}_M - R_{MM}s_M), \mathrm{Im}(\hat{y}_M - R_{MM}s_M)\}$$
$$d(s_i) = \max\left\{\mathrm{Re}\left(\hat{y}_i - \sum_{K=i+1}^{M} R_{iK}s_K - R_{ii}s_i\right), \mathrm{Im}\left(\hat{y}_i - \sum_{K=i+1}^{M} R_{iK}s_K - R_{ii}s_i\right)\right\} \quad (3)$$

c. A significantly smaller standard deviation in the number of visits. As will be seen in section 5, this allows CMS to provide ~2x gain in throughput under early termination for practical use.

Table 2 – The ℓ^∞ norm

Distance Norm	Distance D to (x,y)	Cumulative Distance
Euclidean (ℓ^2) norm	$\sqrt{x^2+y^2}$	D(parent) + D(child)
ℓ^∞ norm	$\max(\lvert x\rvert, \lvert y\rvert)$	max(D(parent), D(child))

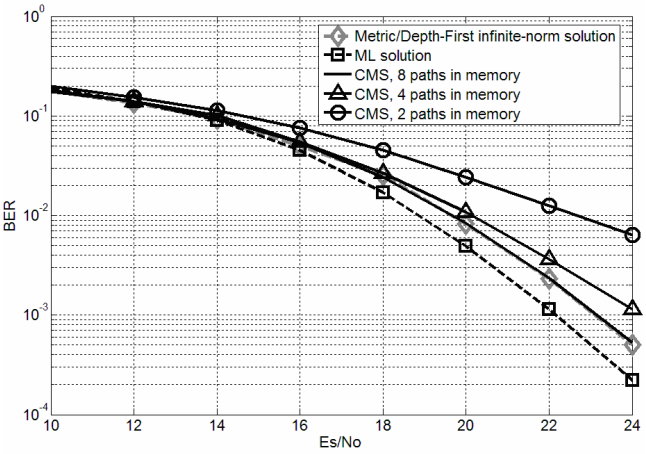

Figure 2 – Performance of the proposed CMS search strategy for a 4x4 16-QAM setup.

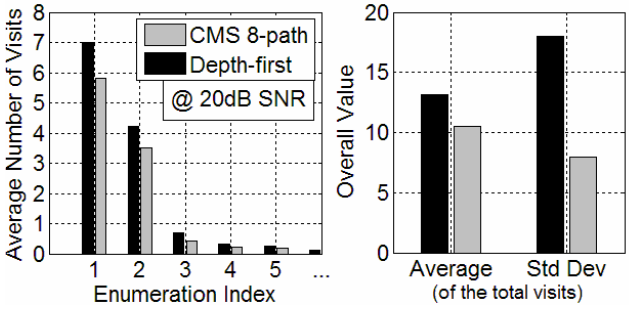

Figure 3 –Statistics of the number of visits for a 4x4 16-QAM setup at *20dB SNR*. Average (per sibling, overall) and standard deviation.

4. CMS ON THE FSD TREE

In addition to choosing a search strategy that minimizes the number of iterations for a given tree structure, further reduction in complexity can be achieved through using an inherently pruned tree structure presented in [8] for the fixed-throughput sphere decoder (FSD). Figure 4 shows a sample.

Recall from (3) that the distance to a node on layer *i* in the tree is scaled by the value of R_{ii}. Accordingly, ordering the spatial streams (and thus the layers in the tree) moves critical branch decisions between the different layers. By a particular ordering, a tree composed of only M paths can be used for a 4x4 M-QAM system (traditionally M^4 paths in the tree) with almost no loss in performance [8]. The sorting can be done during the QR- decomposition prior to the search stage [13].

In addition to the reduction in the average required iterations, applying the CMS to the FSD tree allows for reduction in the enumeration overhead of the search.

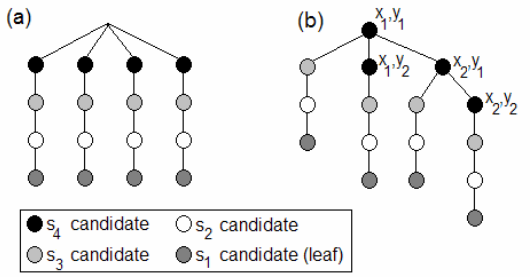

Figure 4 – (a) 4x4 QPSK FSD tree (b) Tree as traversed by the CMS algorithm with simplified enumeration. Indices x_1, x_2, y_1, and y_2 correspond to the order of the constellation point in the real and imaginary dimensions.

Enumeration corresponds to the sorting of each set of siblings. In a depth-first search, enumeration is required for efficient pruning of the tree [11] whereas in the proposed CMS it is required for enforcing constraint (b). In both cases, enumeration can either be performed when the parent node is expanded or distributed over the visits to the sibling nodes.

In the case of the FSD tree, as shown in figure 4, enumeration is only required on the top-level of the tree whereas only the closest child is visited on lower levels. Also, since the proposed CMS handles a sorted set of boundary nodes at all time then when a top-level node is visited, both its imaginary and real neighbouring sibling nodes can be added to the boundary (figure 4). This reduces the enumeration space from M to $2\sqrt{M}$ points for an M-QAM constellation.

5. THROUGHPUT GUARANTEE

Like depth-first, CMS suffers from variability in the number of iterations required to find the best solution. However, as shown in section 3, the variability of CMS is significantly smaller and we thus expect better performance when the search is terminated prematurely to guarantee throughput.

Early termination in the context of iterative sphere decoding corresponds to limiting the number of cycles or iterations where each cycle refers to the expansion of one node. Naturally, limiting the number of cycles independently for each vector is not efficient. A block early termination (BET) scheme was accordingly proposed in [6] for depth-first search and adopted later in [7] for depth-first search on the FSD tree (sequential COSIC). The scheme works by allowing vector k of a block of n vectors

$$N_k = (n \times N_{av}) - N_{used} - \left[(n-k) \times M\right] \text{ cycles}$$

where N_{av} is the average cycles/vector over the block, N_{used} is the number of already used cycles in the block, and M is the number of antennas.

This scheme guarantees each vector a minimum of M cycles allowing a depth-first search to reach the first leaf.

To accommodate the minimum number of cycles M for finding a leaf, we modify the CMS such that if the remaining allowed cycles is less than L, where L < M, the search expands the best node such that it is not above level L.

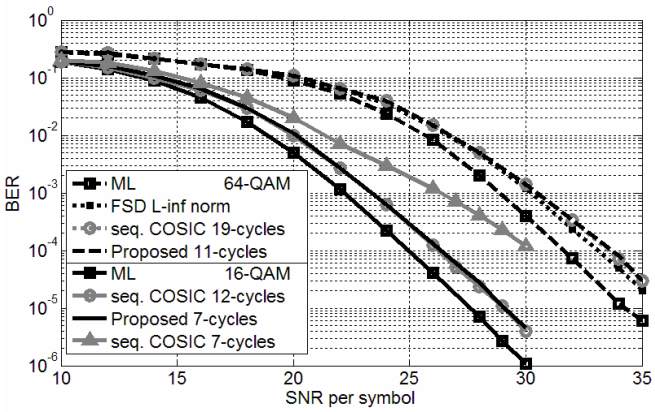

Figure 5 – BER performance of the proposed algorithm as compared to ML, FSD [8], and sequential COSIC [7] under a channel with 15 ns of r.m.s. delay spread for both 16 and 64-QAM.

Performance and Analysis

In a scenario where the vectors in a block correspond to independent and identically distributed channel instances, a block with sufficiently large size n would exhibit global performance. N_{av} can then be equal to the global average number of required cycles. This is the case that is assumed in [6] and [7] where n = 64 and the vectors are suggested to be obtained form a 64-subcarrier OFDM system.

In reality, sub-carriers in an OFDM system exhibit high correlation between their channels (e.g. 802.11n or WiMax). This is also true if the vectors of a block are obtained sequentially from a non-OFDM system.

We present simulation results of our proposed algorithm as compared to the depth-first based alternative (sequential COSIC [7]) for realistic channel conditions. We assume each block of vectors is obtained from a 64-subcarrier OFDM system with 20MHz of bandwidth.

Figure 5 shows the case of a 15ns r.m.s. delay spread channel. This scenario is at the highest-correlation end of the typical indoor r.m.s. delay spread range (15 to 50ns) [14]. Notice that the proposed decoding algorithm requires only ~58% as many cycles per vector as the depth-first based alternative.

The difference in the required number of cycles per vector is due to the fact that when vector k uses $N_k > N_{av}$ cycles, vector k+1 corresponding to a correlated sub-carrier would probably require $N_{k+1} > N_{av}$ cycles as well. Thus, a block of vectors will not sample the global population. The required average cycles over that block might thus be greater than N_{av}. The advantage of the proposed algorithm then is the low variability in the number of required visits due to the more intelligent choice of visits.

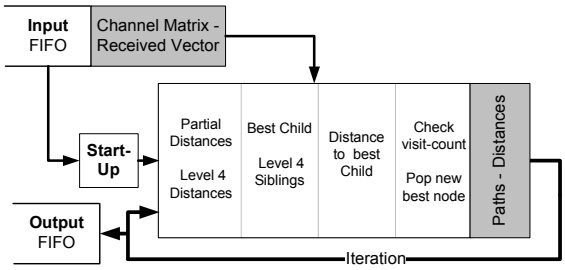

Figure 6 – Block diagram of proposed decoder architecture.

Table 3 – Synthesis Results.

	Proposed	Sequential COSIC [7]	Parallel COSIC [7]	K-Best [6]
Norm	ℓ^∞	ℓ^∞	ℓ^1	ℓ^1
Algorithm	CMS	Depth-first	Parallel FSD	K-Best K=10
Mod. Scheme	16-QAM	16-QAM	16-QAM	16-QAM
Clk. freq.	331 MHz	492 MHz	769 MHz	200 MHz
Area	20 kG	25 kG	83 kG	135 kG
Throughput	757 Mbps	656 Mbps[1]	4.1 Gbps	319 Mbps
Loss[2] vs. ML	1.3 dB	1.3 dB	0.75 dB	~1 dB

[1]Throughput is calculated according to the 12 cycles/vector requirement.
[2]15ns r.m.s. delay spread channel, 20MHz of bandwidth.

6. HARDWARE RESULTS

The proposed decoder was implemented as a single stage data-path that performs one node expansion per iteration as shown in figure 6. The design was coded in HDL and synthesized to a TSMC 65nm standard cell library with 13-bit precision.

Table 3 shows the synthesis results as compared to other published results scaled to the 65nm technology node. The SNR penalty shown is under the 15ns r.m.s delay spread channel; throughput numbers for the sequential COSIC [7] are recalculated accordingly. Area is calculated as the number of equivalent 2-input NAND gates.

The results show ~2x increase in throughput compared to the sequential COSIC [7] for normalized clock frequency and circuit area. On the other hand, compared to non-iterative decoding, the proposed decoder requires 24% of the area of the parallel COSIC [7] (the most efficient parallel search of the FSD to our knowledge).

7. CONCLUSION

In this work we proposed an alternative search strategy for iterative sphere decoding. We showed that a decoder based on the proposed strategy performs satisfactorily well under realistic channel conditions with high-throughput guarantee and ~2x reduction in complexity (increase in normalized throughput) compared to the next best approach.

REFERENCES

[1] E. Viterbo and J. Boutros, “A universal lattice code decoder for fading channels,” IEEE Trans. Inform. Theory, vol. 45, no. 5, pp. 1639–1642, July 1999.

[2] M. O. Damen, H. E. Gamal, and G. Caire, “On maximum likelihood detection and the search for the closest lattice point,” IEEE Trans. Inform. Theory, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

[3] A. Paulraj, R. Nabar, and D. Gore, Introduction to Space-Time Wireless Communications, 1st ed. Cambridge University Press, 2003.

[4] G. J. Foschini, “Layered space-time architecture for wireless communication in a fading environment when using multielement antennas,” Bell Labs Technical Journal, pp. 41–59, Oct. 1996.

[5] B. Hassibi and H. Vikalo, “On the expected complexity of sphere decoding,” in Proc. 35th Asilomar Conference on Signals, Systems and Computers, vol. 2, Monterey, CA, pp. 1051–1055, Nov. 2001.

[6] A. Burg, M. Borgmann, M. Wenk, C. Studer, and H. B¨olcskei, "Advanced receiver algorithms for MIMO wireless communications," in Proc. of the Design Automation and Test Europe Conf., vol. 1, Mar. 2006, pp. 593–598.

[7] C. Hess, M. Wenk, A. Burg, P. Luethi, C. Studer, N. Felber, and W. Fichtner, "Reduced Complexity MIMO Detector with Close-to ML Error Rate Performance", GLSVLSI’07, March 2007, pp. 200-203.

[8] L. G. Barbero and J. S. Thompson, “A fixed-complexity MIMO detector based on the complex sphere decoder,” IEEE 7th Workshop on Signal Processing Advances for Wireless Communications, July 2006, pp.1-5.

[9] E. W. Dijkstra, “A note on two problems in connexion with graphs,” in Numerische Mathematik. Mathematisch Centrum, Amsterdam, Netherlands, 1959, vol. 1, pp. 269–271.

[10] M. Pohst, “On the computation of lattice vectors of minimal length, successive minima and reduced basis with applications,” ACM SIGSAM Bull., vol. 15, 1981, pp. 37–44.

[11] C. P. Schnorr and M. Euchner, “Lattice basis reduction: Improved practical algorithms and solving subset sum problems,” Mathematical Programming, vol. 66, 1994, pp. 181–199.

[12] K. Su, “Efficient maximum likelihood detection for communication over multiple input multiple output channels,” Ph.D. dissertation, Dept. Eng., Univ. of Cambridge, 2005.

[13] A. Wiesel, X. Mestre, A. Page, and J. Fonollosa, “Efficient implementation of sphere demodulation,” in Proc. IEEE Work-shop Signal Process. Advances Wireless Communications, June 2003, pp. 36–40.

[14] Erceg, V., et al., ‘Indoor MIMO WLAN channel models’, IEEE Work Group Document, IEEE 802.11-03/161r1, July 2003.