LOW-COMPLEXITY AND HIGH-PERFORMANCE SOFT MIMO DETECTION BASED ON DISTRIBUTED M-ALGORITHM THROUGH TRELLIS-DIAGRAM

Yang Sun and Joseph R. Cavallaro

Department of Electrical and Computer Engineering
Rice University, Houston, TX 77005
Email: {ysun, cavallar}@rice.edu

ABSTRACT

This paper presents a novel low-complexity multiple-input multiple-output (MIMO) detection scheme using a *distributed M-algorithm* (DM) to achieve high performance soft MIMO detection. To reduce the searching complexity, we build a MIMO trellis graph and split the searching operations among different nodes, where each node will apply the M-algorithm. Instead of keeping a global candidate list as the traditional detector does, this algorithm keeps multiple small candidate lists to generate soft information. Since the DM algorithm can achieve good BER performance with a small M, the sorting cost of the DM algorithm is lower than that of the conventional K-best MIMO algorithm. The proposed algorithm is very suitable for high speed parallel processing.

Index Terms— MIMO, M-Algorithm, Soft Detection

1. INTRODUCTION

The main challenge of soft MIMO detection is to efficiently and accurately generate the log-likelihood radios (LLRs) for the outer channel decoder. The optimal soft MAP detection will consume enormous computing power and require tremendous computation resources which makes it impossible to be employed in MIMO systems with higher-order modulation schemes. To reduce the exponentially algorithmic complexity, some close-to-optimal soft detection algorithms have been proposed by researchers [1, 2, 3, 4, 5]. These algorithms can be categorized as either depth-first or breadth-first tree-search algorithms. The depth-first detector has non-deterministic complexity and variable throughput which makes it sensitive to the channel conditions. The breadth-first K-best detector has a high sorting complexity especially when K is large.

In this paper, we propose a new soft MIMO detection method using the M-algorithm [6]. In [7], we introduced a greedy algorithm to achieve high throughput MIMO detection. In this paper, we extend this greedy algorithm and propose a distributed M-algorithm (DM) for soft-output MIMO detection, where no feedback from the outer channel decoder is available. We transform the traditional MIMO detection problem into a shortest path search problem in a MIMO trellis diagram where each trellis node is physically mapped to one transmit symbol in the alphabet. Strictly speaking, although trellis is memoryless, we broadly use it to efficiently represent the search space of the MIMO signal. The trellis-search based DM algorithm has two steps. In the first step, it prunes unlikely trellis paths by only keeping the best M paths, the ones with the least M cumulative weights, among QM candidates at each trellis node. In the second step, the pruned trellis will be augmented to generate candidate lists for LLR calculation. This algorithm significantly reduces the searching cost compared with the conventional tree-search algorithm because searching is done in a distributed way. Based on the simulation results, this algorithm can achieve a BER performance that is very close to the optimal soft MAP detection.

2. SYSTEM MODEL

We consider a coded MIMO system with N_T transmit antennas and N_R receive antennas. The MIMO transmission can be modeled as:

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

where $\mathbf{H}$ is an $N_R \times N_T$ complex matrix, $\mathbf{s} = [s_0\ s_1\ ...\ s_{N_T-1}]^T$ is an $N_T \times 1$ transmit vector, $\mathbf{y}$ is an $N_R \times 1$ received vector, and $\mathbf{n}$ is a vector of independent zero-mean complex Gaussian noise entries with variance σ^2 per real component. Each complex symbol s_k is associated with a real bit-level vector $\mathbf{x_k} = [x_{k,0}\ x_{k,1}\ ...\ x_{k,B-1}]^T$, where the b-th bit of $\mathbf{x_k}$ is denoted as $x_{k,b}$ and B is the number of bits per symbol. Symbol s_k and its associated bit vector $\mathbf{x_k}$ will be used interchangeably through the paper. The soft-output detector is to compute the LLR value for each bit $x_{k,b}$ as follows:

$$L(x_{k,b}|\mathbf{y}) = \ln \frac{P[x_{k,b}=+1|\mathbf{y}]}{P[x_{k,b}=-1|\mathbf{y}]}. \tag{2}$$

Assuming there is no prior knowledge of the transmit signal, using the maxlog approximation [8], (2) can be simplified to

$$L(x_{k,b}|\mathbf{y}) \approx \frac{1}{2\sigma^2}\Big(\min_{\mathbf{s}\in\mathbb{X}_{k,b}^{-1}} d(\mathbf{s},\mathbf{y}) - \min_{\mathbf{s}\in\mathbb{X}_{k,b}^{+1}} d(\mathbf{s},\mathbf{y})\Big), \tag{3}$$

where set $\mathbb{X}_{k,b}^{+1} = \{\mathbf{s}|x_{k,b}=+1\}$ and set $\mathbb{X}_{k,b}^{-1} = \{\mathbf{s}|x_{k,b}=-1\}$. Using QR decomposition according to $\mathbf{H} = \mathbf{QR}$, where $\mathbf{Q}$ and $\mathbf{R}$ refer to an $N_R \times N_T$ unitary matrix and an $N_T \times N_T$ upper triangular matrix, respectively, the Euclidean distance can be calculated as

$$d(\mathbf{s},\mathbf{y}) = \|\mathbf{y} - \mathbf{H}\cdot\mathbf{s}\|^2 = \|\hat{\mathbf{y}} - \mathbf{R}\cdot\mathbf{s}\|^2 + C, \tag{4}$$

where $\hat{\mathbf{y}} = \mathbf{Q}^H\mathbf{y}$, and C is a constant that can be dropped when calculating the LLR in (3).

3. ALGORITHM

3.1. MIMO Trellis Construction

The Euclidean distance can be computed recursively as:

$$d_k = d_{k+1} + \gamma_k, \quad k = N_T-1, N_T-2, ..., 0, \tag{5}$$

where $d_{N_T} = 0$, and $d(\mathbf{s}) = d_0$. The non-negative distance increment γ_k is defined as:

$$\gamma_k = \left| \hat{y}_k - \sum_{j=k}^{N_T-1} R_{k,j} s_j \right|^2. \tag{6}$$

To calculate (5), we create an MIMO trellis diagram. Figure 1 shows an example trellis diagram for a 4×4 QPSK system. In the MIMO trellis diagram, the nodes are ordered into N_T vertical slices or levels. The trellis starts with level $N_T - 1$ and ends with level 0 where level k corresponds to antenna k. In each level, there are $Q = 2^B$ different nodes. Each node maps to a different complex QAM symbol that belongs to a known constellation alphabet. Thus, each transmit *vector* is a *path* through the trellis diagram. In the trellis representation, the number of the nodes grows linearly, instead of exponentially when using the tree structure, with the number of the transmit antennas. The nodes in level k are represented as $v_k(q)$. Each q is associated with a complex symbol s. In this paper, the symbol s and its decimal label q will be used interchangeably. The edge between nodes $v_{k+1}(q')$ and $v_k(q)$ has a weight of $\gamma_k(\mathbf{q}^{(k)})$, where $\mathbf{q}^{(k)}$ is the partial symbol vector $\mathbf{q}^{(k)} = [q_k \; q_{k+1} \; ... \; q_{N_T-1}]^T$. Because of the upper triangular property of $\mathbf{R}$, from (6) we know that the weight function γ_k can be calculated from the partial symbol vector $\mathbf{q}^{(k)}$. It should be noted that this MIMO trellis is not memoryless because the weight function not only depends on the previous state but also depends on all the previous encountered states.
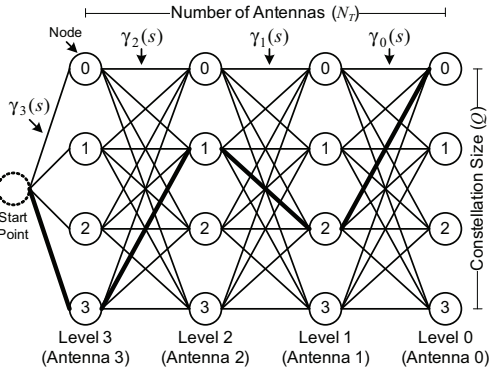

Fig. 1. An example trellis diagram for 4×4 QPSK system.

Given a received MIMO symbol vector, we may associate each edge in the trellis with a weight so that the problem of MIMO detection becomes the problem of finding the minimum-weight path in the trellis. Then the nodes encountered along this path are the detected transmit vector. For example, the shortest path (or the ML path) in Figure 1 is highlighted which corresponds to a (hard) detected vector of $\mathbf{q} = [0 \; 2 \; 1 \; 3]^T$. In the trellis diagram, a path weight d is the sum of the branch weights along the path. The branch metric γ is added each time to the partial path weight:

$$d_k(q) = d_{k+1}(q') + \gamma_k(\mathbf{q}^{(k)}), \tag{7}$$

where q' and q are the nodes at level $k + 1$ and k, respectively.

3.2. DM Algorithm For Soft MIMO Detection

For coded MIMO systems, the search for the shortest paths needs to be carried out for every bit $x_{k,b}$ in order to compute the LLR based on (3). It is obvious that soft detection has a much higher computational complexity than hard detection. Traditionally, soft detection is implemented by keeping a global "list" of the best n candidates in the tree. The list size has to be as large as possible to achieve good performance.

In this section, we propose a search-efficient soft detection algorithm based on the distributed M-algorithm through the trellis diagram. Instead of keeping a global "list", we keep a small local list for every trellis node. Each trellis node represents a hypothesis for the QAM symbol transmitted by antenna k. The goal of the DM algorithm is to find M shortest paths for each hypothesis for symbol s_k. Thus, for each antenna k, QM candidates found by Q different symbol hypotheses form a bigger list for generating the LLRs. The DM algorithm has two steps which are summarized as follows.

i) Step 1 of the DM algorithm is called **Path Reduction** (PR), where it will prune the unlikely paths in the trellis by applying the M-algorithm at each trellis node. Figure 2 illustrates the basic flow of the PR algorithm. Let M be the local list size. The levels of the trellis are labeled in a descending order, starting from $N_T - 1$ and ending with 0. Each node in Figure 2 receives QM incoming path candidates from the previous level of the trellis, and then the M best paths, the ones with the least M cumulative weights, are selected from these QM candidates. Next, these M survivors are fully-extended so that each node will have QM outgoing paths to the next level of the trellis. This PR process repeats until the end of the trellis. The details of the PR algorithm are summarized in Algorithm 1.

Figure 3 shows an example 4×4 QPSK trellis after applying the PR operation, where each node only keeps $M = 2$ best incoming paths, the ones with the least M cumulative path weights. Recall that in the flow of the PR algorithm (c.f. Figure 2), the M best paths, or survivors, are selected from QM incoming paths at each node. These M survivors correspond to the M incoming paths for each node in Figure 3. One important observation is that for each hypothesis q ($0 \leq q \leq Q - 1$) for symbol s_0, the symbol transmitted by antenna 0, has $M = 2$ smallest full-path weights. So altogether QM candidates form a bigger candidate list $\mathcal{L}_0$ for antenna 0. Let the binary representation of q to be $\{b_0 \; b_1 \; ... \; b_{B-1}\}$. Because we have found a list of M candidates for every case of q, the $LLR(b_x)$ can be generated based on list $\mathcal{L}_0$. As two Euclidean distances for $b_x = 0, 1$ always exist in list $\mathcal{L}_0$, we can compute the LLR more accurately (in the traditional MIMO algorithm, LLR clipping is often required to mitigate inaccuracies in LLR computation due to small list size [9]). However, except for antenna 0, not every trellis node has a full-path through the trellis. For example, trellis node $v_1(0)$ and $v_1(2)$ in Figure 3 are dead-ends and they do not have any outgoing paths. This means for hypotheses $q = 0, 2$ for symbol s_1, there are no associated full-path weights. To solve this problem, we can extend the trellis by running a path augmentation operation.

ii) Step 2 of the DM algorithm is called **Path Augmentation** (PA). The PA algorithm is used to extend the trellis so that each trellis node will have a list of M smallest full-paths that pass through this node. As described before, there is no need to do the PA operation for nodes at level 0, i.e. antenna 0. For other antennas $k \neq 0$, each trellis node $v_k(q)$, where q is 0 through $Q - 1$, will be extended for k levels to find the M best full-paths that pass through this node. Figure 4 shows the flow of the PA algorithm for an antenna $k \neq 0$, where $M = 2$. The algorithm starts with retrieving the β-metrics from step 1 of the DM algorithm, that is the β-metrics computed at level k in the PR algorithm. Then these QM β-metrics are sent to a search unit or *searcher* to select the best M *outgoing* paths for node $v_k(q)$. Next, these M survivors are fully extended for the next level of the trellis and similarly these QM newly extended paths are sent

Algorithm 1 Path Reduction (Step1 of the DM Algorithm)

0) **Initialization**: Set level $k = N_T - 1$. In the beginning, there is only one incoming path to each node $v_k(q')$, $0 \leq q' \leq Q-1$. For each node $v_k(q')$, set the m-th surviving path metric $\alpha_k^{(m)}(q')$ as:
$$\alpha_k^{(m)}(q') = \begin{cases} |\hat{y}_k - R_{k,k}s_k|^2, & m = 0 \\ +\infty, & m \neq 0 \end{cases}.$$

1) **Path metrics calculation**: For each node $v_k(q')$, the retained M surviving paths $\alpha_k^{(m)}(q')$ are fully extended for the nodes in the next level of the trellis $v_{k-1}(q)$, $0 \leq q \leq Q-1$, as follows:
  for $(0 \leq q' \leq Q-1)$
    for $(0 \leq m \leq M-1)$
      for $(0 \leq q \leq Q-1)$
        $\beta_{k-1}(q', m, q) = \alpha_k^{(m)}(q') + \gamma_{k-1}^{(m)}(\mathbf{q}^{(k-1)})$,
where $\beta_{k-1}(q', m, q)$ are the extended outgoing path metrics and $\gamma_{k-1}^{(m)}(\mathbf{q}^{(k-1)})$ are the branch metrics as defined in (6).

2) **Path metrics selection**: Because of the trellis structure, $\beta_{k-1}(q', m, q)$ need to be reordered by switching the indices q' and q to create incoming path candidates $\alpha_{k-1}(q, m, q')$ for nodes $v_{k-1}(q)$, $0 \leq q \leq Q-1$, as: $\alpha_{k-1}(q, m, q') = \beta_{k-1}(q', m, q)$. Then for each node $v_{k-1}(q)$, the best M paths are selected from QM candidates, where the m-th survivor is saved as $\alpha_{k-1}^{(m)}(q)$. Next, set $k = k - 1$. If $k = 0$ algorithm stops, else goto 1).
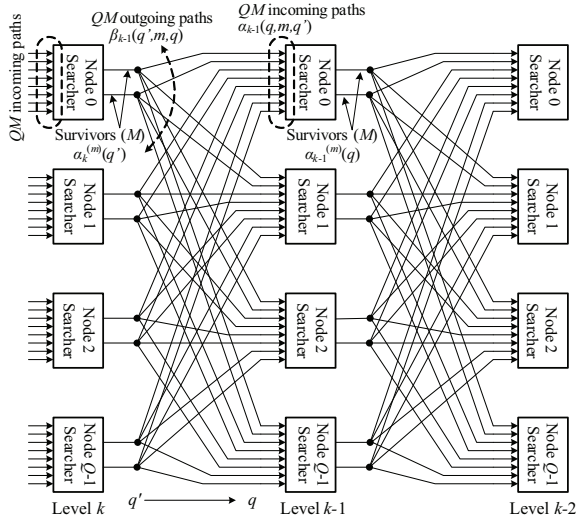

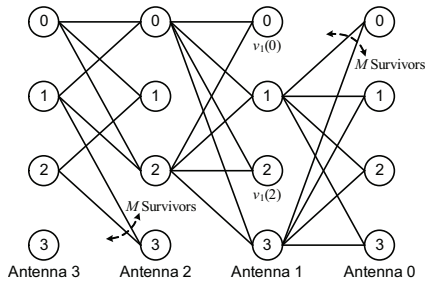**Fig. 2**. Flow of the path reduction algorithm where $M = 2$ survivors are retained at each node.

Fig. 3. An example of path reduction for a 4×4 QPSK trellis, where $M = 2$ incoming paths are selected by each node.

to a searcher to select the best M paths for level $k - 2$. The search operation repeats until it reaches the end of the trellis. As a result of the path augmentation, each node $v_k(q)$, $0 \leq q \leq Q - 1$, has a list of candidates, the ones with the M smallest full-path weights. In other words, for antenna k, each hypothesis for symbol s_k has found the M smallest Euclidean distances. By simply merging these QM candidates, we can create a global candidate list $\mathcal{L}_k$ for computing the LLRs for data bits $x_{k,b}$ using (2).
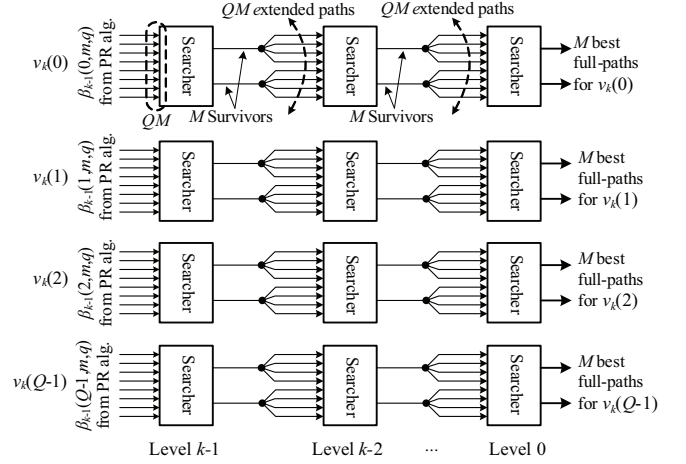

Fig. 4. Flow of the path augmentation algorithm for antenna k.

4. SIMULATION RESULTS

We consider 4×4 16-QAM and 64-QAM MIMO systems where the channel matrices are assumed to have independent random Gaussian distributions. A sorted QR decomposition of the channel matrix is used. The soft-output of the detector is fed to a length 2304, rate 1/2 WiMax layered LDPC decoder [10], which performs up to 15 LDPC iterations. Figures 5 and 6 compare the BER performance with different M values. As a comparison, we also plot the BER curves for the soft MAP detector with exhaustive-search and the soft linear MMSE detector. When $M = 1$, the DM detector shows about 1 dB performance loss compared to the optimal case at BER level of 10^{-5}. When $M = 2, 3$, the DM detector shows a very small performance degradation. When $M = 4$, the DM detector shows almost no performance loss. The simulation result indicates that the DM detector even with a small M can achieve near-ML detection performance.

5. ALGORITHM COMPLEXITY ANALYSIS

Sorting of candidates is often the bottleneck in the traditional K-best detection algorithms. In this section, we compare the sorting cost of the proposed DM algorithm with that of the K-best algorithm. Both DM and K-best algorithms need to carry out a (n, m) sorting operation: find the best m candidates out of n candidates. For large m, which is the case in the K-best algorithms, a full sorting of n candidates is typically needed which significantly limits the system throughput. For example, Kim suggests a large $K = 512$ to achieve good performance for a 4×4 16-QAM system [11].

By contrast, the DM algorithm can achieve good performance with small M, e.g. $M = 2$, so the (n, m) sorting operation does
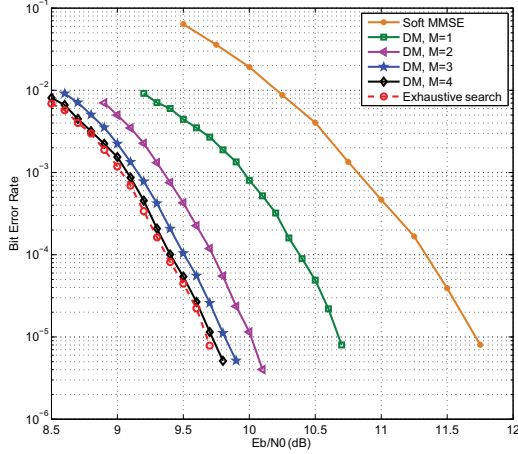
Fig. 5. Simulation results for a LDPC-coded 4×4 16-QAM system.
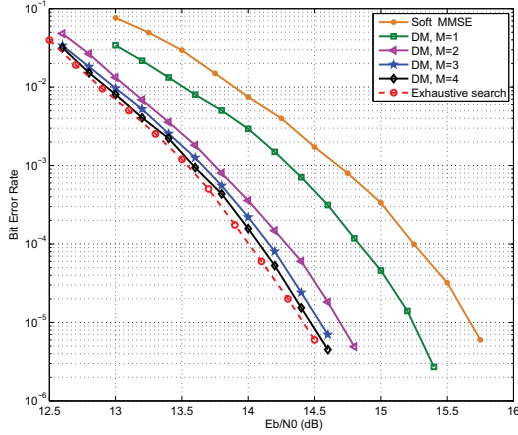

Fig. 6. Simulation results for a LDPC-coded 4×4 64-QAM system.

not necessarily require a full sorting of n candidates, i.e. finding the smallest number or the two smallest numbers from n candidates can be realized very efficiently. Table 1 compares the (n, m) sorting cost of the DM and the K-best algorithms for 4×4 16-QAM systems. The sorting cost is measured by the number of pairwise comparisons. The complexity of (n, m) is assumed to be $n \log(n)$. For the DM algorithm, the sorting complexity is measured by $Q(QM) \log(QM)$, where $Q = 16$. For the K-best algorithm, the sorting complexity is given by $(QK) \log(QK)$. Compared to the traditional K-best algorithm which needs to perform large sorting tasks, the proposed DM algorithm has a much lower sorting complexity which can lead to efficient high-speed hardware implementations. Compared to the depth-first soft sphere detector, e.g. [2], which has variable-complexity, the DM algorithm has a fixed-complexity which is more suitable for hardware implementation.

6. CONCLUSION

We propose a low-complexity, high-performance, soft-output MIMO detection algorithm. The detector significantly reduces the sorting cost by using a distributed M-algorithm through the trellis diagram. This algorithm is very suitable for parallel implementation.

Table 1. Algorithm complexity comparison

Algorithm	Complexity	BER degradation @10^{-5}
DM (M=1)	1024	0.95 dB
DM (M=2)	2560	0.35 dB
DM (M=3)	4289	0.15 dB
DM (M=4)	6144	0.05 dB
K-best (K=64)	10240	0.8 dB
K-best (K=128)	22528	0.3 dB
K-best (K=256)	49152	0.1 dB
K-best (K=512)	106496	0.03 dB

8. REFERENCES

[1] D. Garrett, L. Davis, S. ten Brink, B. Hochwald, and G Knagge, "Silicon Complexity for Maximum Likelihood MIMO Detection Using Spherical Decoding," *IEEE J. Solid-State Circuit*, vol. 39, pp. 1544–1552, Sept. 2004.

[2] C. Studer, A. Burg, and H. Bolcskei, "Soft-output sphere decoding: algorithms and VLSI implementation," *IEEE J. Selected Areas in Comm.*, vol. 26, pp. 290–300, February 2008.

[3] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE J. Selected Areas in Comm.*, vol. 24, pp. 491–503, Mar. 2006.

[4] Q. Li and Z. Wang, "Improved K-best sphere decoding algorithms for MIMO systems," in *IEEE Int. Symp. on Circuits and Syst.*, Nov. 2006, pp. 2190–2194.

[5] S. Chen, T. Zhang, and Y. Xin, "Relaxed K-Best MIMO Signal Detector Design and VLSI Implementation," *IEEE Tran. VLSI*, vol. 15, pp. 328–337, March 2007.

[6] J. Anderson and S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Tran. on Comm.*, vol. 32, pp. 169–176, 1984.

[7] Y. Sun and J. R. Cavallaro, "High Throughput VLSI Architecture for Soft-Output MIMO Detection Based on a Greedy Graph Algorithm," in *ACM Great Lakes Symposium on VLSI Design*, May 2009, pp. 445–450.

[8] B.M. Hochwald and S.T. Brink, "Achieving Near-Capacity on a Multiple-Antenna Channel," *IEEE Tran. Comm.*, vol. 51, pp. 389–399, Mar. 2003.

[9] Y. L. C. de Jong and T. J. Willink, "Iterative tree search detection for mimo wireless systems," *IEEE Tran. on Comm.*, vol. 53, no. 6, pp. 930–935, 2005.

[10] Y. Sun and J. R. Cavallaro, "A low-power 1-Gbps reconfigurable LDPC decoder design for multiple 4G wireless standards," in *IEEE Internationl SOC Conference*, Spet. 2008, pp. 367–370.

[11] H. Kim, D-U Lee, and J. D. Villasenor, "Design Tradeoffs and Hardware Architecture for Real-Time Iterative MIMO Detection Using Sphere Decoding and LDPC Coding," *IEEE J. Selected Areas in Comm.*, vol. 26, pp. 1003–1014, Aug. 2008.