Research Article

A System View on Iterative MIMO Detection: Dynamic Sphere Detection versus Fixed Effort List Detection

Christina Gimmler-Dumont,[1] Frank Kienle,[1] Bin Wu,[2] and Guido Masera[2]

[1] Microelectronic Systems Design Research Group, University of Kaiserslautern, Erwin-Schroedinger-Straße,
   67663 Kaiserslautern, Germany
[2] Dipartimento di Elettronica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

Correspondence should be addressed to Christina Gimmler-Dumont, gimmler@eit.uni-kl.de

Multiple-antenna systems are a promising approach to increase the data rate of wireless communication systems. One efficient possibility is spatial multiplexing of the transmitted symbols over several antennas. Many different MIMO detector algorithms exist for this spatial multiplexing. The major difference between different MIMO detectors is the resulting communications performance and implementation complexity, respectively. Particularly closed-loop MIMO systems have attained a lot of attention in the last years. In a closed-loop system, reliability information is fed back from the channel decoder to the MIMO detector. In this paper, we derive a basic framework to compare different soft-input soft-output MIMO detectors in open- and closed-loop systems. Within this framework, we analyze a depth-first sphere detector and a breadth-first fixed effort detector for different application scenarios and their effects on area and energy efficiency on the whole system. We present all system components under open- and closed-loop system aspects and determine the overall implementation cost for changing an open-loop system in a closed-loop system.

1. Introduction

Multiple-antenna (MIMO) systems are a promising approach to increase the data rate of wireless communication systems in rich-scattering environments. Spatial multiplexing is a spectrally efficient way to exploit the diversity of the MIMO channel while an outer error correction code ensures the desired quality of service for a given data rate. This setting is called a Bit Interleaved Coded Modulation (BICM) system (see Section 3). Particularly iterative MIMO detection attained a high attention in the last years. In an iterative receiver, reliability information is fed back from the outer channel decoder to the MIMO detector and vice versa. The resulting communications performance is improved by 3 dB and more compared to open-loop decoding [1, 2].

This improvement is gained at the cost of a highly complex signal detection (Section 4). Optimal detection by exhaustive search is infeasible for realistic scenarios (4×4 antennas, 16- or 64-QAM). Finding the right trade-off between communications performance and implementation complexity and understanding the implications on the whole receiver is one of the major challenges in the design of iterative MIMO receivers. MIMO detection algorithms and their implementations have been extensively studied in the literature (Section 2). They can be divided into classes with similar characteristics, for example, linear filters or breadth-first tree search algorithms.

The fixed effort list detector (breadth-first search, Section 4.2) and the sphere detector (depth-first search, Section 4.1) are among the most promising approaches to obtain a good communications performance in iterative systems at reasonable implementation complexity. The fixed effort detector processes the MIMO vectors at a constant throughput whereas the sphere detector has a dynamic throughput due to the nature of the depth-first search. However, the sphere detector is able to approach the optimum detection while the communications performance of the fixed effort detector is restricted by the storage requirements of the generated lists.

In this paper, we explore the design space for iterative MIMO detection from a system perspective comparing fixed effort and sphere detection. We start with an investigation of the system communications performance for both algorithms (Section 5) and continue with an architectural analysis of the complete receiver system. Not only the implementation of the MIMO detectors but also of the other building blocks in the iterative receiver (channel preprocessing and channel decoding) needs to be studied to analyze the whole system (Section 6). Therefore, it is mandatory to fix some shared design constraints. We introduce a generic architecture framework which connects the building blocks by system memories in order to be able to exchange individual blocks easily (Section 7.1). Characteristics of each block are analyzed in a system context (Section 7.2); for example, the channel decoder can employ different algorithms for open loop decoding and closed loop decoding.

A fair comparison of different MIMO detectors is only possible as a part of an iterative receiver. Different architectures have advantages for different system constraints, thus we compare fixed effort and sphere detector in several throughput centric and communication centric scenarios (Section 7.3). Eventually, we investigate the system cost in terms of throughput, area, and power when moving from an open-loop to a closed-loop system (Section 7.4). The corresponding area and energy efficiency numbers drop by more than a factor of 2 for closed-loop decoding with one iteration.

2. Review of State-of-the-Art Detection Algorithms and Their Implementations

Multiple-antenna systems employing spatial multiplexing increase the spectral efficiency. However, this improvement comes at the cost of an increased receiver complexity. Finding the right trade-off between communications performance and implementation complexity in MIMO detection is one of the key challenges in the receiver design.

In order to optimally solve the MIMO detection problem, an exhaustive search for the best solutions can be done over all signal constellations. The number of possible signal constellations increases exponentially with the number of antennas and the number of bits per modulation symbol. For a 4×4 antenna system employing 16-QAM, more than 65000 constellations exist. For 64-QAM, this number rises to more than 16000000. This makes an exhaustive search infeasible for a hardware implementation [9].

As the optimal exhaustive search is far too complex for hardware implementations, many suboptimal detection algorithms exist with a big range in communications performance and complexity. They can be divided into the following classes.

2.1. Linear MIMO Detection. Zero-Forcing (ZF) and minimum mean square error (MMSE) filters apply an inverse of the channel to the received signal in order to restore the transmitted signal [10]. These linear filters can be implemented at a low complexity; however, their communications performance is very low as well. The MMSE filter considers the noise power in the interference cancellation and therefore shows a slightly better performance.

2.2. Successive Interference Cancellation. The successive interference cancellation (SIC) technique was initially adopted by the vertical Bell Laboratories layered space-time (V-BLAST) system [11]. In contrast to the basic ZF and MMSE filters, SIC detects the transmitted streams sequentially. It chooses the substream with largest signal-to-noise ratio and removes the interference of each detected stream before continuing the detection process. The performance of the SIC algorithm is generally better than ZF and MMSE filters.

2.3. Breadth-First Tree Search Algorithms. For further improvement of the communications performance, the MIMO detection problem can be mapped on a tree search. The tree search algorithms can be divided into breadth-first and depth-first search algorithms.

Breadth-first algorithms offer a constant throughput with a small loss in communications performance compared to an optimal detection. Among the best known techniques are the K-best algorithm [12, 13] and the fixed-complexity detector [14]. While traversing the tree, the K-best detector keeps the K best nodes in each level. This requires sorting operations which result in a high implementation cost. The fixed-effort detector follows a regular tree traversal path which is determined at design time. This regularization enables the design of highly-efficient parallel architectures [14], however, at slightly lower communications performance than the K-best algorithm. In general, the communications performance of breadth-first algorithms depends on the number of nodes visited in each layer of the tree.

2.4. Depth-First Tree Search Algorithms. Depth-first detectors apply pruning criteria to remove parts of the tree in the search to reduce the computational complexity [15]. They approach the ML solution for hard output and the MAP solution for soft output. Sphere detectors achieve the best communications performance among the different detection techniques, but due to the nature of the depth-first search, their throughput is variable. The sequential tree search order makes it difficult to parallelize the detection. There exist many sub-optimal variants regarding enumeration technique, pruning criterion, or simplified metric calculations, for example, [3, 16].

The hardware implementation of sphere detection has been extensively explored for hard- and soft-output versions, for example, [17, 18]. Different forms of pipelining have been proposed to increase the architecture parallelism [3, 19].

2.5. Iterative MIMO Detection and Channel Decoding. In this paper, we investigate iterative receivers where MIMO detector and channel decoder exchange reliability information to increase the communications performance. Therefore, the aforementioned algorithms have to be adjusted to utilize the given soft-input information. Studer et al. implemented a soft-input soft-output extension of the linear MMSE filter (called MMSE-PIC) in [8]. Breadth-first algorithms have

been extended to list detectors. Thereby, the breadth-first algorithm generates a number of candidate vectors which are stored in a list. The iterative detection process is only based on the available vectors in the list. In contrast to breadth-first algorithms, soft information can be directly included in depth-first sphere detection algorithms, for example, [1, 2]. Witte et al. presented the first implementation of such a soft-input soft-output sphere detector in [5] based on the single-tree-search algorithm (STS) of [20].

2.6. State-of-the-Art MIMO Detection Architectures. Architectures for MIMO detection have been extensively studied in the literature for all kind of algorithms. Several silicon implementation results of the proposed MIMO detection architectures are listed in Table 1.

A fundamental one-node-per-cycle hardware architecture for the hard-output depth-first sphere decoder is introduced in [3] together with the l^∞-norm approximation for complexity reduction. This architecture has been firstly extended to a soft-output version in [4] by applying techniques including single-tree-search, sorted QR decomposition and LLR clipping, and further enhanced to be soft-input soft-output in [5], to perform iterative MIMO decoding. Other architectural improvements, such as the modified best first with fast descent (MBF-FD) MIMO detection [6], and the parallel and scalable architecture for modified metric first (MMF) list sphere detection (LSD), have been proposed to enhance detection efficiency and performance. The basic architectural considerations for implementing the depth-first sphere decoders are generalized in [21], from high-level architecture and enumeration strategy to approximations and pipeline interleaving.

The architecture for K-best algorithm is modified in [22] by applying bidirectional partial tree search and hybrid two-step scheme to reduce complexity. Another similar approach, namely, the early pruned technique, is applied to reduce the complexity of the K-best algorithm [7].

Besides the sphere decoders, several other MIMO detection algorithms have been investigated. In [23], the Markov chain Monte Carlo (MCMC) simulation techniques are reported to achieve comparable performance to LSD. The MMSE-SIC algorithm has also been improved to be soft-input soft-output and achieve very high throughput by applying parallel architecture [8].

3. System Model

In this paper, we focus on a bit interleaved coded modulation (BICM) scheme like that shown in Figure 1. The source generates a random infoword $\mathbf{u}$ of length K_c which is encoded by the channel encoder. The interleaved codeword $\mathbf{X}^N$ consists of N_c bits which are linearly grouped into N subblocks $\mathbf{x}_n$:

$$\mathbf{X}^N = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N). \tag{1}$$

Each subblock $\mathbf{x}_n$ consists of Q coded bits:

$$\mathbf{x}_n = \left(x_{1,n}, x_{2,n}, \ldots, x_{q,n}, \ldots, x_{Q,n} \right), \quad x_{q,n} \in \{-1, +1\}. \tag{2}$$

Each $\mathbf{x}_n$ is mapped directly to a complex symbol $s = \mathrm{map}(\mathbf{x}_n)$ chosen from a 2^Q-ary QAM modulation scheme. M_T symbols are combined in one transmission vector $\mathbf{s}_t$. M_T is the number of transmit antennas:

$$\mathbf{s}_t = (s_{1,t}, s_{2,t}, \ldots, s_{m,t}, \ldots, s_{M_T,t}). \tag{3}$$

The whole modulated sequence is represented by

$$\mathbf{S}^T = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_t, \ldots, \mathbf{s}_T). \tag{4}$$

T time slots are needed to transmit all symbols of one codeword. The transmission of vector $\mathbf{s}_t$ in time step t is modeled by

$$\mathbf{y}_t = \mathbf{H}_t \cdot \mathbf{s}_t + \mathbf{n}_t \tag{5}$$

with $\mathbf{H}_t$ the channel matrix of dimension $M_T \times M_R$ and $\mathbf{n}_t$ the noise vector of dimension M_R whose entries are zero-mean and unit variance Gaussian variables. The elements of $\mathbf{H}_t$ are modeled as independent, complex, zero-mean, Gaussian random variables. Real and imaginary part are independent variables each with variance $\sigma^2 = N_0/2$. It is assumed that $\mathbf{H}_t$ is ergodic, that is, its entries change independently after each channel use. Furthermore, $\mathbf{H}_t$ is perfectly known by the MIMO detector and all employed antenna constellations are symmetric with $M_T = M_R = M$. The received vectors $\mathbf{y}_t$ are gathered in the matrix $\mathbf{Y}^T$

$$\mathbf{Y}^T = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t, \ldots, \mathbf{y}_T) \tag{6}$$

with

$$\mathbf{y}_t = (y_{1,t}, y_{2,t}, \ldots, y_{m,t}, \ldots, y_{M_R,t}). \tag{7}$$

Before the decoding starts, the channel preprocessing applies a QR decomposition on $\mathbf{Y}^T$ and $\mathbf{H}_t$ (for details see Section 4). This results in the transformed received vectors $\hat{\mathbf{Y}}^T$ and updated channel matrices $\mathbf{R}_t$. The decoding process is iterative between MIMO detector and channel decoder. They exchange probability information on the codeword. The soft-in-soft-out MIMO detector determines the likelihood of the bits for each received vector $\hat{\mathbf{y}}_t$ using the a priori information $\mathbf{L}_t^a$ from the channel decoder. Only the extrinsic information $\boldsymbol{\lambda}^e = \boldsymbol{\lambda} - \mathbf{L}^a$ is passed on to the channel decoder.

The channel decoder processes the whole codeword at a time. It uses the interleaved a priori information $\boldsymbol{\lambda}^a$ from the MIMO detector for the calculation of the estimated information bit sequence $\hat{\mathbf{u}}$ and the a posteriori logarithmic likelihood ratios (LLRs) $\boldsymbol{\Lambda}$ of the codeword. The extrinsic information $\mathbf{L}^e = \boldsymbol{\Lambda} - \boldsymbol{\lambda}^a$ is returned to the MIMO detector thus closing the iterative loop.

4. MIMO Detection

A received symbol vector $\mathbf{y}_t$ can be seen as a weighted superposition of the entries of $\mathbf{s}_t$ disturbed by Gaussian noise. The task of the MIMO detector is the equalization and separation of the originally sent symbols $\mathbf{s}_t$. The MIMO detector works on one received vector $\mathbf{y}_t$ at a time.

TABLE 1: ASIC implementations of recently reported MIMO detectors.

| Publication	[3]	[4]	[5]	[6]	[7]	[8]
Algorithm	Hard-output sphere decoder	Soft-output STS-SD	SISO STS-SD	Soft-output MBF-FD	Hard-output K-best	SISO MMSE-PIC
Antenna	4×4	4×4	4×4	8×8	4×4	4×4
Modulation	16-QAM	16-QAM	16-QAM	64-QAM	64-QAM	64-QAM
Iterative decoding	no	no	yes	no	no	yes
Constant throughput	no	no	no	no	yes	yes
Technology	250 nm	250 nm	90 nm	130 nm	130 nm	90 nm
Clock frequency	71 MHz	71 MHz	250 MHz	198 MHz	138 MHz	568 MHz
Core area	50 KG	57 KG	96 KG	350 KG	491 KG	410 KG
Max. throughput	169 Mbit/s	70 Mbit/s	72 Mbit/s	429 Mbit/s	1200 Mbit/s	757 Mbit/s
Power consumption	—	—	—	58.2 mW	185 mW	189.1 mW
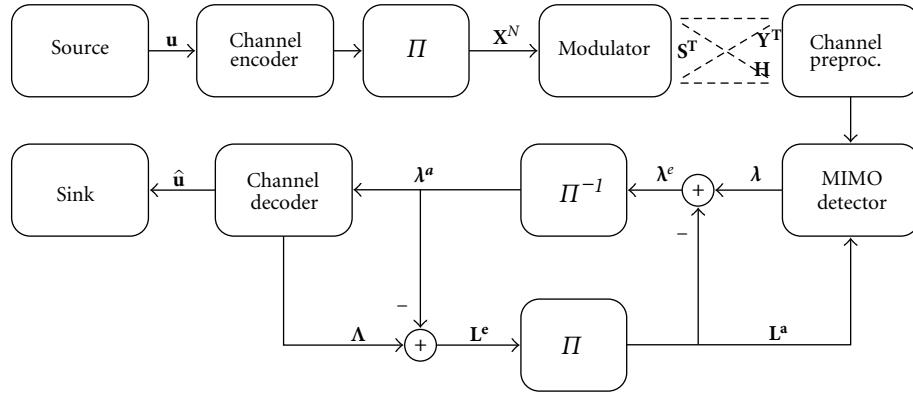
FIGURE 1: System model of bit interleaved coded modulation scheme with iterative MIMO detection and channel decoding in the receiver.

For all detection-related explanations, the time indices of y, H, and s are dropped for ease of notation. Even if not mentioned specifically for each equation, the vectors s and x are always connected via $s = \text{map}(x)$. $x_{q,m}$ denotes the qth bit of the mth symbol in s.

For iterative detection and decoding the MIMO detector computes logarithmic likelihood values (LLRs) on each bit

$$\lambda\left(x_{q,m}\right) = \ln \frac{P\left(x_{q,m} = +1 \mid \mathbf{y}\right)}{P\left(x_{q,m} = -1 \mid \mathbf{y}\right)}. \tag{8}$$

For independent $x_{q,m}$, the probability $P(x_{q,m} = +1 \mid \mathbf{y})$ is obtained by summing up the probabilities of all possible symbol vectors s which contain $x_{q,m} = +1$:

$$P\left(x_{q,m} = +1 \mid \mathbf{y}\right) = \sum_{\forall \mathbf{s} \mid x_{q,m} = +1} P(\mathbf{s} \mid \mathbf{y}). \tag{9}$$

Using *Bayes theorem*, $P(\mathbf{s} \mid \mathbf{y})$ can be expressed as

$$P(\mathbf{s} \mid \mathbf{y}) = \frac{P(\mathbf{s}) \cdot P(\mathbf{y} \mid \mathbf{s})}{P(\mathbf{y})}. \tag{10}$$

We can observe that the analyzed probability consists of three parts. $P(\mathbf{s})$ takes into account that not every s is equally likely given the a priori information $\mathbf{L}^a$ from the channel decoder. As the codeword is interleaved before the QAM mapping the bits $x_{q,m}$ are assumed independent from each other. Therefore, $P(\mathbf{s})$ is the product of its bits' probabilities:

$$P(\mathbf{s}) = \prod_i P\left(x_{q,m}\right). \tag{11}$$

The conditional probability $P(\mathbf{y} \mid \mathbf{s})$ illustrates how likely it is to receive the signal $\mathbf{y}$ when s has been sent. It equals the probability of the noise needed to receive $\mathbf{y}$ when s is sent over the channel $\mathbf{H}$. As the noise $\mathbf{n}$ is additive white Gaussian with variance N_0, $P(\mathbf{y} \mid \mathbf{s})$ can be written as

$$P(\mathbf{y} \mid \mathbf{s}) = P(\mathbf{n} = \mathbf{y} - H\mathbf{s}) = \frac{1}{\sqrt{2\pi}} e^{-\|\mathbf{y} - H\mathbf{s}\|^2 / N_0}. \tag{12}$$

The third part $P(\mathbf{y})$ is constant during the detection of $\mathbf{y}$ and is cancelled out when applying (10) and (12) to (8):

$$\lambda\left(x_{q,m}\right) = \ln \frac{\sum_{\forall \mathbf{s} \mid x_{q,m} = -1} P(\mathbf{s}) \cdot e^{-\|\mathbf{y} - H\mathbf{s}\|^2 / N_0}}{\sum_{\forall \mathbf{s} \mid x_{q,m} = +1} P(\mathbf{s}) \cdot e^{-\|\mathbf{y} - H\mathbf{s}\|^2 / N_0}}. \tag{13}$$

The large number of multiplications and the exponential function involved in the computation of (13) make it less attractive for implementation. Therefore, it is transformed

into the logarithmic domain where the exponential function disappears and the multiplications become additions. Hereby, a problem is posed by the additions. The *Jacobian logarithm* is used to formulate them as

$$\ln(e^x + e^y) = \max{}^*(x, y), \quad (14)$$

with

$$\max{}^*(x, y) = \max(x, y) + \ln\left(1 + e^{-|x-y|}\right). \quad (15)$$

The max*-operation can be approximated by the normal max-operation. This leads to the Max-Log-Map approximation [1]:

$$\lambda\left(x_{q,m}\right) \approx \max_{\forall \mathbf{s}|x_{q,m}=-1}\left\{\ln P(\mathbf{y} \mid \mathbf{s}) + \ln P(\mathbf{s})\right\}$$
$$- \max_{\forall \mathbf{s}|x_{q,m}=+1}\left\{\ln P(\mathbf{y} \mid \mathbf{s}) + \ln P(\mathbf{s})\right\},$$

$$\lambda\left(x_{q,m}\right) \approx \max_{\forall \mathbf{s}|x_{q,m}=-1}\left\{-\frac{||\mathbf{y} - H\mathbf{s}||^2}{N_0} + \sum_{\forall q,m}\ln P\left(x_{q,m}\right)\right\}$$
$$- \max_{\forall \mathbf{s}|x_{q,m}=+1}\left\{-\frac{||\mathbf{y} - H\mathbf{s}||^2}{N_0} + \sum_{\forall q,m}\ln P\left(x_{q,m}\right)\right\}. \quad (16)$$

Exchanging maximum by minimum operations the next equation is obtained:

$$\lambda\left(x_{q,m}\right) \approx \min_{\forall \mathbf{s}|x_{q,m}=+1}\left\{||\mathbf{y} - H\mathbf{s}||^2 - N_0 \sum_{\forall q,m}\ln P\left(x_{q,m}\right)\right\}$$
$$- \min_{\forall \mathbf{s}|x_{q,m}=-1}\left\{||\mathbf{y} - H\mathbf{s}||^2 - N_0 \sum_{\forall q,m}\ln P\left(x_{q,m}\right)\right\}. \quad (17)$$

An interpretation for (17) is that we derive the LLR value $\lambda(x_{q,m})$ from the most likely symbol vectors $\mathbf{s}$ with $x_{q,m}$ being $+1$ or -1, respectively. The metric $d(\mathbf{s})$ measures the likelihood that a specific vector $\mathbf{s}$ has been sent:

$$d(\mathbf{s}) = ||\mathbf{y} - H\mathbf{s}||^2 - N_0 \sum_{\forall q,m}\ln P\left(x_{q,m}\right). \quad (18)$$

Small metrics $d(\mathbf{s})$ relate to a high probability of $\mathbf{s}$ having been sent.

Calculating all possible $d(\mathbf{s})$ to determine (17) becomes quickly infeasible for higher antenna constellations and/or higher-order modulations as the complexity grows with 2^{QM}. Therefore, many sub-optimal algorithms with lower complexity exist. Most of them are based on a tree search. In order to map the metric calculations (18) on a tree, the channel matrix $\mathbf{H}$ is decomposed into a unitary matrix $\mathbf{Q}$ and an upper-triangular matrix $\mathbf{R}$. The Euclidean distance is rewritten as

$$||\mathbf{y} - H\mathbf{s}||^2 = ||\mathbf{y}' - \mathbf{R}\mathbf{s}||^2 \quad (19)$$

with $\mathbf{y}' = \mathbf{Q}^H\mathbf{y}$. Equation (18) is replaced by the equivalent metric

$$d(\mathbf{s}) = ||\mathbf{y}' - \mathbf{R}\mathbf{s}||^2 - N_0 \sum_{\forall q,m}\ln P\left(x_{q,m}\right). \quad (20)$$

The triangular structure of $\mathbf{R}$ allows the recursive calculation of $d(\mathbf{s})$

$$d_m = d_{m+1} + \gamma_m\left(s^{(m)}\right) \quad (21)$$

with the starting point $d_{M+1} = 0$ and $d(\mathbf{s}) = d_1$. The metric update $\gamma_m(s^{(m)})$ depends on the partial symbol vector $s^{(m)} = (s_m, s_{m+1}, \ldots, s_M)$:

$$\gamma_m\left(s^{(m)}\right) = \left|y'_m - \sum_{j=m}^{M} R_{m,j}s_j\right|^2 - N_0 \sum_{q=1}^{Q}\ln P\left(x_{q,m}\right). \quad (22)$$

This recursive structure can be represented by a tree with $M+1$ levels as shown in Figure 2 for the modulation alphabet $\{-1, +1\}$. The root node corresponds to d_{M+1} and each leaf node corresponds to the metric $d(\mathbf{s})$ of one possible vector $\mathbf{s}$. Each level corresponds to the detection of one symbol s_m. Branches are labeled with an element of the modulation alphabet. When advancing from a parent to a child node, the metric of the child node d_m is calculated from the metric of the parent node d_{m+1} and the branch metric γ_m.

Based on this tree search, many different MIMO detection algorithms exist. The main differences between the algorithms can be described by how they traverse the tree, for example, breadth-first, depth-first, or metric-first, and how branches of the tree are excluded. In general, those algorithms result in different communications performance and implementation complexities. In the next sections, we will present two different algorithms and show the trade-offs between them.

4.1. Sphere Detector. The sphere detector is a depth-first search which considers all symbol vectors $\mathbf{s}$ in the computation of (17) which lie inside a sphere of radius r around the received vector $\mathbf{y}$, that is, for which $d(\mathbf{s}) < r$. The radius r is determined before the search starts. The choice of the radius offers a trade-off between very good communications performance and throughput. For a high radius, many nodes are visited and the resulting communications performance is close to the optimum. For a low radius, the search is very fast but the communications performance is degraded.

During the search, the sphere detector may visit many leaf nodes but only stores the data relevant for the computation of the LLR values (17). Furthermore, sorted QR decomposition [24] and MMSE preprocessing [10] are used as additional techniques for complexity reduction.

4.2. Fixed Effort List Detector. A fixed effort detector [25] generates a list $\mathbf{L}$ of leaf nodes and their according Euclidean distances. It is based on a breadth-first search in which the number of child nodes is predetermined for every layer of the tree. Thus, the number of visited nodes is constant for one so-called *node distribution*. Typically, in the beginning
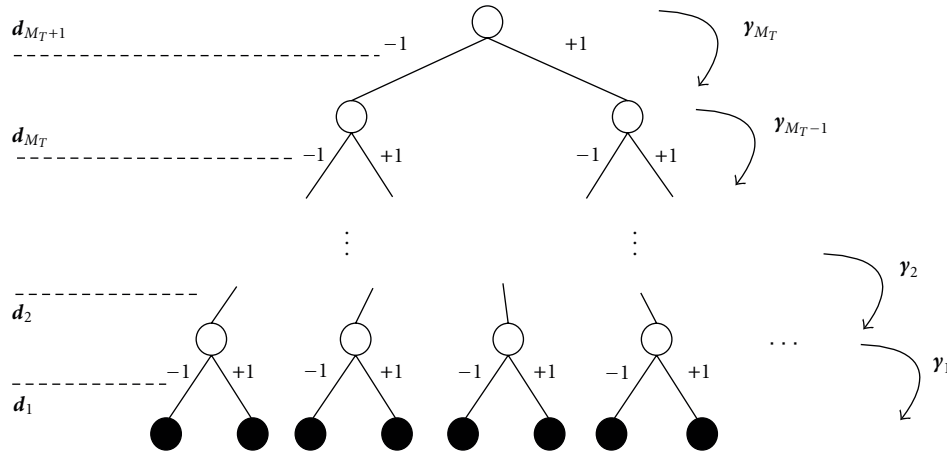
FIGURE 2: Detection problem represented by a tree for the modulation alphabet $\{-1, +1\}$ (BPSK) and M_T antennas.

of the tree search, many children nodes are visited while, in lower layers, only one or two nodes are expanded. Therefore, the use of a sorted QR decomposition which moves the unreliable layers to the top of the tree is mandatory [14, 24]. Each candidate in the list consists of a bit vector **x** and the corresponding Euclidean distance d_E.

In order to obtain soft-output LLRs and to be able to process a priori information, the fixed effort MIMO detector has to be followed by an LLR generator. In the LLR generator, the a posteriori LLRs are approximated by (17) but the minimum search only runs over those vectors **s** which have been stored in the list **L**. Also, the Euclidean distance has been stored in **L** and does not have to be recalculated.

5. Results Communications Performance

The design space for iterative MIMO detection and channel decoding is enormous considering all the possibilities for sub-optimal algorithms, the choice of the channel code, scheduling between detector and decoder, channel and modulation parameters, and so forth. Covering all these possibilities is out of scope of this paper. Therefore, we introduce the following restrictions on the design space. As channel code we employ a WiFi compliant 64-state non-systematic, nonrecursive convolutional code. The decoding of convolutional codes is noniterative thus removing the scheduling problem between inner and outer iterations. We use code rate 1/2 and code words of 2304 bits. This code length has been chosen to allow a comparison with existing LDPC codes of the WiMax, WiFi standards [26]. This in-depth comparison will be done in a future publication. The channel is modeled as Rayleigh fading with 4 transmit and 4 receive antennas.

As a first step of the design space exploration we compare the communications performance for two different MIMO detection algorithms, namely, the sphere detector and the fixed effort detector from Section 4. The two algorithms offer a trade-off between hardware efficiency and communications efficiency. Two modulation schemes are compared—16-QAM and 64-QAM—which pose different requirements to the MIMO detector in terms of complexity.

Figures 3 and 4 show the communications performance results for the two algorithms for 4×4 antennas, 16-QAM and 64-QAM, respectively. The frame error rate is measured after the convolutional decoder. The red curves show the results of the close-to-optimum sphere detector. The green and blue curves stem from the fixed effort detector with different list sizes L. We limited the number of outer iterations to 3. Currently, this is the highest number of iterations we assume in a hardware realization since the throughput will linearly decrease with the iterations. Anyway, additional iterations will not result in a further significant gain in communications performance [1, 2].

In both figures, we observe a similar behaviour of the different algorithms. Both, the sphere detector and the fixed effort detector have their largest gain within the first iteration (up to 4 dB for the sphere detector and around 3 dB for the fixed effort detector). Furthermore, the communications performance of the fixed effort detector depends significantly on the list size L. Particularly for small list sizes (green curves), more than one iteration does not significantly improve the performance anymore. Whereas the difference between small (green) and big list sizes (blue curves) is small in iteration 0, it is well known that, for the larger list sizes, the communications performance is better in successive iterations. When an extremely large list is adopted (e.g., 1024 for 16-QAM and 4096 for 64-QAM), the performance of the fixed effort list detector approaches the soft-output depth-first sphere detector.

Recapitulatory, the most important observations are listed in the following. After iteration 0, fixed effort and sphere detector based MIMO detection obtain a similar communications performance. Both achieve the biggest gain within the first iteration. The communications performance of the fixed effort detector depends heavily on the list size. For small list sizes, no more than one iteration is beneficial as the decoding process "gets stuck," that is, does not further
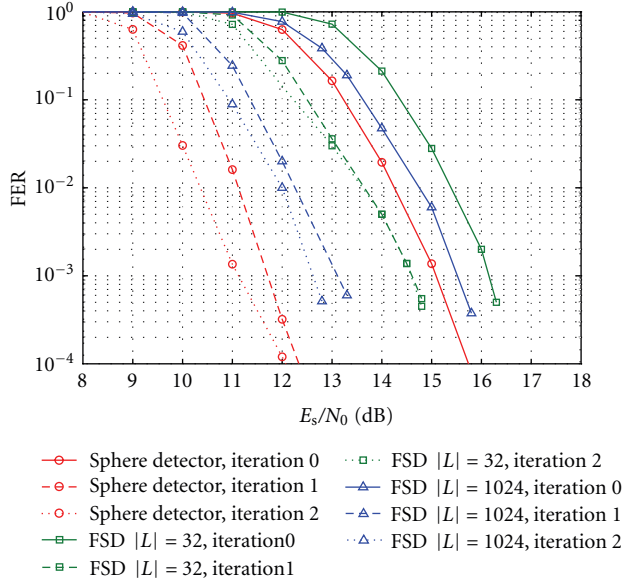
FIGURE 3: Communications performance of 4×4 antennas system, 16-QAM modulation for different MIMO detection algorithms.
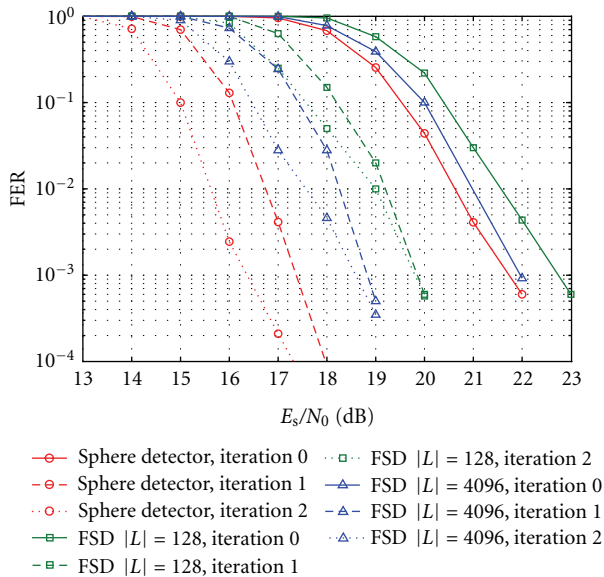

FIGURE 4: Communications performance of 4×4 antennas system, 64-QAM modulation for different MIMO detection algorithms.

improve. The best communications performance is achieved by sphere detection with several outer iterations.

6. Results VLSI Components

In this section, we will present the architectures and implementation results of the different VLSI components which will be combined and analyzed as an iterative receiver in Section 7.

All designs were synthesized in a 65 nm low-power bulk CMOS standard cell library. Target frequency after place & route is 300 MHz which is typically used for industrial designs. In order to ensure 300 MHz after place & route, synthesis was done with a target frequency of 360 MHz. We considered the following PVT parameters: Worst Case (WC, 1.1 V, 125°C), Nominal Case (NOM, 1.2 V, 25°C) and Best Case (BC, 1.3 V, −40°C). Synthesis was performed with Synopsis Design Compiler in topographical mode, place & route (P&R) with Synopsys IC Compiler. Synthesis as well as P&R were performed with Worst Case PVT settings of the 65 nm library.

6.1. QR Decomposition. From the bunch of existing algorithms, we chose the modified Gram-Schmidt process [27] to compute the QR decomposition due to its simplicity and stability when working with finite precision values. Input and output matrices are quantized with 12 bits for real and imaginary values, respectively. It has been shown that this quantization yields only a minor degradation in system communications performance [28]. The resulting architecture runs a sorted QR decomposition with MMSE preprocessing for a 4×4 channel matrix in 167 clock cycles. After P&R it has an area of $0.14 \, \mathrm{mm}^2$ and consumes a power of 12.0 mW in nominal case when running at 300 MHz.

6.2. Convolutional Decoder. In open-loop systems, convolutional codes can be decoded with the Viterbi algorithm [29] which provides the ML solution, that is, a sequence of hard output bits. In closed-loop MIMO systems, however, soft-output LLR values of the whole codeword are needed for the outer iterations. Thus, the BCJR algorithm [30] has to be applied to obtain the soft-output MAP solutions. Input and output LLR values are quantized with 6 bits each.

State-of-the art convolutional decoders process 1 bit per clock cycle. Consequently, they obtain a throughput of 300 Mbit/s at a clock frequency of 300 MHz. In [31], a 65 nm technology Viterbi decoder design has been presented which is able to run at a clock frequency of more than 300 MHz. It consumes an area of $0.11 \, \mathrm{mm}^2$ and has a power consumption of approximately 40 mW.

Implementations of the BCJR-algorithm for 64-state convolutional codes are not widely available in the literature. Therefore, we chose the 180 nm technology decoder design from [32]. We scaled the original implementation data down to 65 nm technology yielding an area of $0.31 \, \mathrm{mm}^2$ and a power consumption of approximately 240 mW (area scaling factor: $65^2/180^2$, power scaling factor: $65^{1.5}/180^{1.5}$).

6.3. Sphere Detector. The tree search for sphere detection can be separated into five basic operations: computing the interference reduced symbol, enumerating the most promising children nodes, computing the metrics, processing the results of the leaf nodes and storing intermediate results and choosing the next node. In the presented sphere decoder architecture, each of these operations has been implemented in a separate block, see Figure 5. The enumeration unit performs the enumeration of children nodes either based
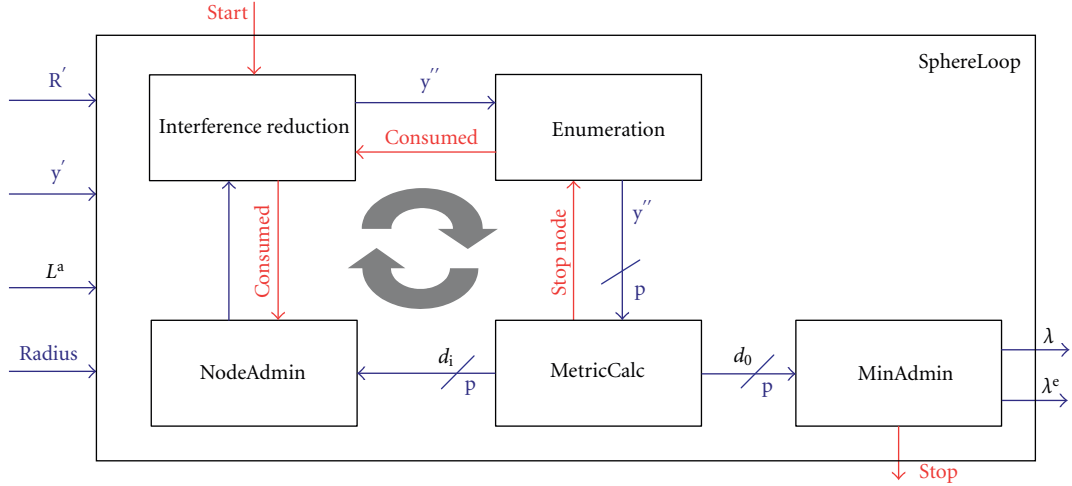
Figure 5: Sphere detector architecture.

Table 2: Implementation results of the sphere detector architecture after place & route for a clock frequency of 300 MHz.

At 300 MHz	Sphere Detector
Modulation	up to 64-QAM
Antennas	up to 4×4
Area	$0.26\,\text{mm}^2$
Throughput	38–58 Mbit/s
Power consumption	15 mW

Table 3: Implementation results for the components of the fixed effort list detector architecture after place & route for a clock frequency of 300 MHz.

At 300 MHz	Fixed effort detector		LLR-unit	
Modulation	16-QAM	64-QAM	16-QAM	64-QAM
List size	32	128	32	128
Area	$0.36\,\text{mm}^2$		$0.14\,\text{mm}^2$	
Throughput	267 Mbit/s	109 Mbit/s	141 Mbit/s	55 Mbit/s
Power	103 mW	118 mW	21 mW	31 mW

on the interference reduced symbol or based on the a priori information.

The presented architecture computes two nodes per cycle in contrast to other depth-first sphere decoders (e.g., [3, 5]) which employ a one-node-per-cycle architecture. This is a new approach which doubles the throughput compared to state-of-the-art implementations. Its detailed architecture will be presented in a future publication since this paper deals with system analysis and the trade-off between communications performance versus implementation performance. The sphere detector works with antenna systems up to 4×4 antennas and QAM modulation schemes up to 64-QAM. During run-time, throughput can be traded off against communications performance by adjusting the radius. However, due to the nature of the depth-first search, the throughput is dynamic and varies with the channel conditions and the outer iterations. After place & route, the design has an area of $0.26\,\text{mm}^2$ and a power consumption of only 15 mW. The implementation data is summarized in Table 2.

6.4. Fixed Effort List Detector. The architecture of the fixed effort list detector supports 16-QAM and 64-QAM modulation. The list size is configurable to be 32 and 128 for 16-QAM and 64-QAM, respectively. It consists of a list generator (employing the fixed effort detection algorithm) and an

individual LLR generator to generate soft-outputs, as shown in Figure 6.

The list generator is implemented by an eight-nodes-per-cycle parallel architecture, which processes 8 nodes in each clock cycle concurrently as a group, with the breadth-first tree search order. Eight identical units are employed for each of the main arithmetic tasks, such as enumeration and metric calculation. After the tree search, a candidate list is sent to the LLR generator, which receives the a priori data from channel decoder and computes the extrinsic data. The LLR generator is also implemented with highly parallel architecture. The throughput of both, the list generator and the LLR generator, depends highly on the list size. Implementation results after place & route are summarized in Table 3.

7. System Analysis

In this section we will investigate the cost for practical applications with respect to throughput, area, and power. Therefore, we first introduce a generic architecture framework supporting different MIMO detectors and channel decoders. After presenting each building block individually in the last section, we will analyze different aspects of the components regarding the complete iterative system. The major problem of MIMO iterative systems with the overall design decisions is the dynamic constraints for throughput
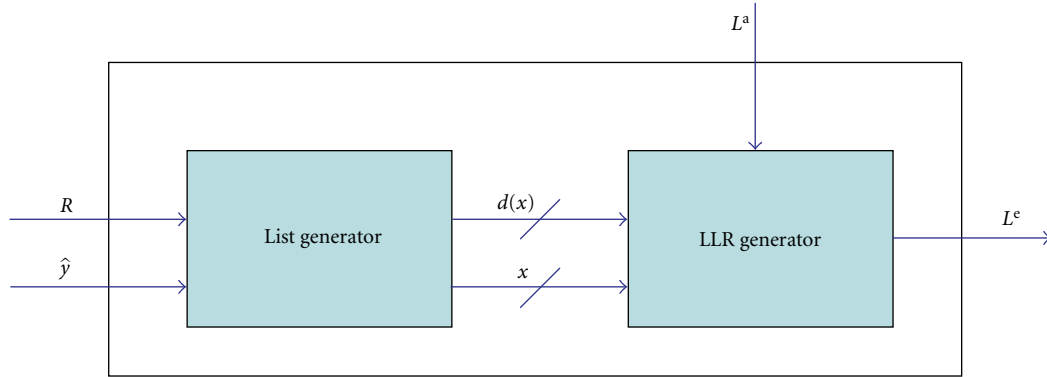
FIGURE 6: Fixed Effort list detector architecture.

and communications performance in different application scenarios. Thus, we will compare the sphere detector and fixed effort detector for different scenarios and SNR ranges. Eventually, we analyze the difference in implementation costs for open- and closed-loop systems.

7.1. Architecture Framework. We have mapped the iterative receiver structure from Figure 1 onto a general architecture framework which allows to plug in different MIMO detectors and channel decoders. The framework—shown in Figure 7—connects the main building blocks via several system memories. The area for each memory is shown in Figure 7. The total area of all system memories is 0.271 mm^2.

The iterative receiver structure from Figure 1 is mapped onto this generic framework. During the inner iterations of the channel decoder the values in DEC_IN might be updated. Thus, the original information is not on hand after decoding. The a posteriori LLR values λ have to be stored in DET_OUT in order to be able to extract the extrinsic information $\mathbf{L}^a$ for the next iteration of the detector. Interleaver and Deinterleaver tables are stored in INT and DE_INT and are read by interleaver unit Π and deinterleaver unit Π^{-1}, respectively. We assume that all complex values require 12 bits for real and imaginary part, respectively, and that all LLR values are quantized with 6 bits.

In the further analysis, we distinguish between the open-loop system without feedback between channel decoder and MIMO detector and the closed-loop with feedback. In closed-loop systems, all memories are mandatorily required. When the MIMO detector is processing a codeword, the decoder has to wait until it is finished and vice versa. Thus, MIMO detector and channel decoder are never active at the same time.

For an open-loop system, the architecture framework can be simplified. First of all, the memories related to the feedback loop—DET_OUT, DET_IN, and INT—are obviously not needed. But in addition, the QR decomposition can provide the data as needed for the MIMO detector so the memories Y_HAT and MAT_R are not required. While the channel decoder is working on one codeword the MIMO detector can already start the next one. In this way, MIMO detector and channel decoder can both be active at all times.

The only additional requirement to enable full activity is the doubling of DEC_IN. In summary, in open-loop systems we need an area of 0.123 mm^2 for system memories and in closed-loop systems we need 0.271 mm^2.

7.2. Components in the System. In Section 6 the VLSI building blocks were introduced without any system considerations. In the following paragraphs, we will look at the dependencies between throughput, communications performance, and different system parameters for each component and what are requirements on the components in open-loop and closed-loop receivers. The observations from the next paragraphs are also summarized in Table 4. The units are shown in columns next to each other giving a good overview of individual design problems, throughputs, and constraints.

QR Decomposition. The presented design for QR decomposition processes matrices for 2×2 or 4×4 antennas including the sorting of layers and MMSE preprocessing. For 4×4 matrices, the unit processes $1.8 \cdot 10^6$ matrices per second consuming 6.68 nJ per matrix. Under the assumption of a truly ergodic channel, that is, the channel changes independently after each use, this relates to 28.8 Mbit/s for 16-QAM, or 43.2 Mbit/s for 64-QAM. In contrast to the MIMO detector, a higher constellation size is beneficial for the bit throughput of the QR decomposition because the processing time depends only on the size of the matrix. In a realistic channel, it is expected that the channel will stay constant for several channel uses. In this case, the QR decomposition only has to be done once for several MIMO vectors and the bit throughput increases. For the QR decomposition there is no difference between open-loop and closed-loop systems as the channel preprocessing is only done once for every channel matrix.

Sphere Detector. The sphere detector architecture detects MIMO vectors for systems with up to 4×4 antennas and QAM modulation schemes up to 64-QAM. Throughput and communications performance depend mainly on the number of visited nodes during the tree search. The sphere radius offers a good trade-off parameter which regulates the number of nodes which can be visited. For a low radius, a
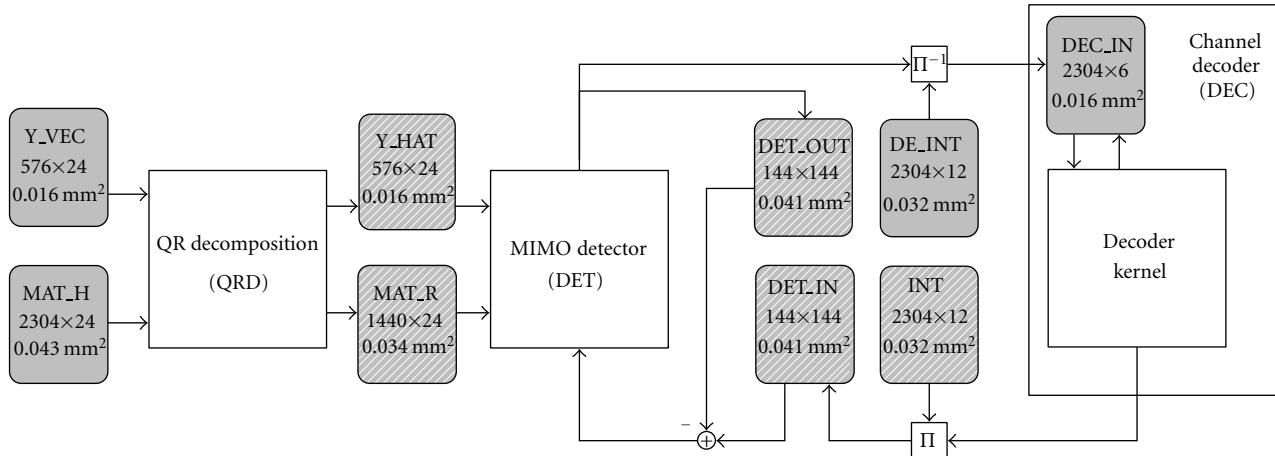
Figure 7: Generic architecture framework including main building blocks and system memories. In open-loop systems, the diagonally hatched memories are not needed while DEC_IN has to be doubled.

Table 4: Design overview for individual components in open-loop and closed-loop systems. Showing them in columns next to each other gives a good overview of individual design problems, throughputs, and constraints even if they are not put in a system yet.

Component	QR decomposition	MIMO sphere	MIMO fixed effort	Convolutional decoder
Flexibility	2 × 2 or 4 × 4 matrices	up to 4 × 4 antennas, up to 64-QAM	4 × 4 antennas, 16-QAM or 64-QAM	code rates 0.5–1
Throughput depends on	Number of antennas, sorting, MMSE or zero-forcing	Modulation, number of antennas, radius	Modulation, number of antennas, list size	Constant
Communications performance depends on	MMSE/zero-forcing, sorted/unsorted	radius	list size	—
Throughput range (4 × 4, 64-QAM)	≥43 Mbit/s (ergodic)	38–58 Mbit/s	109 Mbit/s (fixed effort det.), 55 Mbit/s (LLR)	300 Mbit/s
Open loop		Best communications performance	Good communications performance	Low complexity Viterbi algorithm with hard output
		Dynamic throughput over SNR	List storage not required	
Area	0.14 mm^2 (P&R)	0.26 mm^2 (P&R)	0.36 + 0.14 mm^2 (P&R)	0.11 mm^2 [31]
	+0.06 mm^2 memories	+0.032 mm^2 memories	+0.032 mm^2 memories	+0.032 mm^2 memories
Closed loop	No further processing necessary	Best communications performance	Gets stuck after 2nd iteration	BCJR algorithm with soft-output of the parity information
		Dynamic throughput over iterations	For one feedback loop good throughput	
Area	0.14 mm^2 (P&R)	0.26 mm^2 (P&R)	0.36 + 0.14 mm^2 (P&R)	0.31 mm^2 [32]
	+0.11 mm^2 memories	+0.146 mm^2 memories	+0.146 mm^2 memories	+0.016 mm^2 memories
			+0.32 mm^2 list storage	

high throughput is obtained at the cost of a reduced communications performance and vice versa for a high radius.

Particularly for iterative receivers, the sphere detector offers the best communications performance possible. Due to the depth-first search strategy, the processing time for one MIMO vector is not constant. In fact, it depends on the SNR of the current channel realization. So even for one SNR value, the throughput varies for different MIMO vectors. Generally, the number of nodes will decrease for higher SNR values.

The throughput also changes over the outer iterations. This is problematic when a worst case throughput has to be ensured. Otherwise, there are no changes within the architecture for open- or closed-loop systems.

Fixed Effort List Detector. The fixed effort detector architecture is optimized for 4 × 4 antenna systems with two node distributions for 16-QAM and 64-QAM, respectively. This results in list sizes of 32 or 128 entries. The following LLR

generator is able to work with list sizes up to 128 entries. The node distributions determine the number of nodes which will be visited for one MIMO vector. The choice of the node distribution, however, varies according to the number of antennas, the modulation scheme, and the required list size. The communications performance of the fixed effort detector is directly influenced by the list size. For small list sizes, iterative detection and decoding obtain no more gain after the first iteration. Furthermore, it is mandatory to use a sorted QR decomposition which moves the least reliable layers to the top of the tree. Otherwise, the communications performance drops by several dB. In open-loop processing, the list which is generated in the FSD can be directly used as input for the LLR generator. List storage is not required. Like for the sphere detector, the memories DET_OUT, DET_IN, and INT are not needed. When moving to closed-loop receivers, the lists of all MIMO vectors have to be stored to be reused in the next iterations. The required memory is determined by the 64-QAM case with a list size of 128. For the whole block consisting of 2304 bits, 12288 list entries with 36 bits are needed. The resulting memory consumes approximately 0.32 mm^2. This shows already why bigger lists will not be feasible because already for a list size of 128 the list storage consumes almost the same area as the fixed effort detector core itself.

Convolutional Decoder. The chosen architecture for convolutional decoding processes all code rates ≥ 0.5. The throughput is fixed to 300 Mbit/s by the choice of the architecture independent of the code rate. In the open-loop system, no feedback information is required, thus hard-output bits of the information word are sufficient. In this case, the low-complexity Viterbi algorithm can be chosen which finds the optimal maximum likelihood (ML) solution. In the closed loop, however, soft-output LLR values of the complete codeword are needed as feedback for the MIMO detector. This requires an extended version of the BCJR algorithm which also produces LLR values of the parity information. The introduction of the BCJR algorithm increases the decoder area from approximately 0.11 mm^2 to 0.31 mm^2.

7.3. Scenario Analysis. In most publications, MIMO detectors are analyzed as an individual building block. However, the major problem of iterative MIMO systems are the dynamics of different system scenarios, for example, different throughput and communications performance requirements. The argumentation for one specific architecture is often misleading if it is only based on one specific scenario. Depending on quality of service or throughput requirements, different detection strategies will have advantages. Arguments for a specific realization can be reversed when changing the required flexibility or the multiplexing scheme.

In this section, we will analyze and compare sphere detector and fixed effort list detector in different scenarios. One part of the scenarios will be communication centric, that is, what is the cost to reach a certain frame error rate at a certain signal-to-noise ratio. Other scenarios concentrate on throughput exploring hardware units and

power consumption in order to reach a certain throughput. The scenarios combined with the summarized result data are shown in Table 5. Typically, worst case constraints in systems are for the highest antenna/modulation system. Thus only in the 4×4 antennas, 64-QAM case is shown within the presented system examples. For the fixed effort list detector architecture two LLR units are employed to balance the throughput between list generation and LLR generation.

For all scenarios, it is assumed that the channel decoder processes one bit per clock cycle resulting in a throughput of 300 Mbit/s. This is a typical assumption for state-of-the-art convolutional decoder architectures. While the throughput of the channel decoder is fixed, the throughputs of the MIMO detectors vary depending on the chosen scenario leading to an unbalanced processing time for MIMO detection and channel decoding. In open-loop systems, MIMO detector and channel decoder work as a pipeline. The system throughput is determined by the component with the lowest throughput only, typically the MIMO detector.

For closed-loop systems, there are two alternatives. Either two code words are processed in parallel—one in the MIMO detector and one in the channel decoder—or only one codeword is processed at a time. Working on the same codeword in parallel is prevented by the channel interleaver because detector and decoder always have to wait until the other one has finished processing the whole codeword. In the first case all system memories have to be doubled to store the data of the two code words. Furthermore, for unbalanced processing the throughput is still determined by the slower component whereas the faster component is idle for the rest of the time. On the other hand, if only one code word is handled by the iterative receiver, every component has to wait until the other one has finished the current code word but the memories are not impacted. The system throughput T_{sys} in this case depends on the throughputs of MIMO detector T_{mimo} and channel decoder T_{dec} and the number of outer iterations iter (starting at 0) in the following nonlinear way:

$$\frac{1}{T_{\text{sys}} \cdot (\text{iter} + 1)} = \frac{1}{T_{\text{mimo}}} + \frac{1}{T_{\text{dec}}}. \qquad (23)$$

The system throughput decreases linearly with the number of iterations. As the throughputs of the MIMO detectors largely vary for the different scenarios we chose the second case for our analysis; that is, only one codeword is processed at a time.

The scenarios in Table 5 either target a system frame-error rate of 10^{-3} at different signal-to-noise ratios or specific system throughputs ranging from 30 Mbit/s up to 300 Mbit/s. In the *communication centric scenarios*, the current architecture of the fixed effort list detector is able to achieve the target frame-error rate for the two highest SNR values at a good system throughput of 110 Mbit/s for open loop and 40 Mbit/s for closed loop. The average power consumption decreases for closed-loop systems because the list generator only runs in iteration 0. In the following iterations, only list storage and LLR unit are active. Theoretically, the fixed effort list detector can reach the frame-error rate of 10^{-3} at 18 dB with a list of size 4096 as shown in Figure 4. In that case, the list storage would increase by a factor of 128 to approximately 10.2 mm^2. The processing units would scale

Table 5: System perspective constraints for different scenarios for 4×4 antenna, 64-QAM systems. The resulting throughput, area, power, and communications performance are very dynamic. Two different types of scenarios are analyzed: communications centric and throughput centric. The fixed effort list detector consists of one fixed effort detector and two LLR units.

Scenario description	MIMO detector components	Detector throughput	Detector area	Power consumption detector	System throughput
Fixed Effort List Detector—Communication Centric					
FER = 10^{-3} at SNR = 22 dB	$1 \times$ Fixed Effort List Det., 0 iterations	110 Mbit/s	0.64 mm^2	180 mW	110 Mbit/s
FER = 10^{-3} at SNR = 20 dB	$1 \times$ Fixed Effort List Det. + list storage, 1 iteration	110 Mbit/s	0.96 mm^2	153 mW	40 Mbit/s
FER = 10^{-3} at SNR = 18 dB	theoretically with list size 4096, _not adequate_				
FER = 10^{-3} at SNR = 16 dB	not possible	—	—	—	—
Sphere Detector—Communication Centric					
FER = 10^{-3} at SNR = 22 dB	$1 \times$ Sphere Detector, 0 iterations	38 Mbit/s	0.26 mm^2	15 mW	38 Mbit/s
FER = 10^{-3} at SNR = 20 dB	$1 \times$ Sphere Detector, 1 iteration	58 Mbit/s	0.26 mm^2	15 mW	24 Mbit/s
FER = 10^{-3} at SNR = 18 dB	$1 \times$ Sphere Detector, 1 iteration		0.26 mm^2	15 mW	
FER = 10^{-3} at SNR = 16 dB	$1 \times$ Sphere Detector, 2 iterations	_Achievable but T very low (not adequate)_			
Fixed Effort List Detector—Throughput Centric					
T = 300 Mbit/s, 0 iterations	$3 \times$ Fixed Effort List Det.	330 Mbit/s	1.92 mm^2	540 mW	300 Mbit/s
T = 100 Mbit/s, 0 iterations	$1 \times$ Fixed Effort List Det.	110 Mbit/s	0.64 mm^2	180 mW	110 Mbit/s
T = 100 Mbit/s, 1 iteration	$6 \times$ Fixed Effort List Det. + list storage	660 Mbit/s	4.16 mm^2	918 mW	103 Mbit/s
T = 30 Mbit/s, 1 iteration	$1 \times$ Fixed Effort List Det. + list storage	110 Mbit/s	0.96 mm^2	153 mW	40 Mbit/s
Sphere Detector—Throughput Centric					
T = 300 Mbit/s, 0 iterations	$8 \times$ Sphere Detector	304 Mbit/s	2.08 mm^2	120 mW	300 Mbit/s
T = 100 Mbit/s, 0 iterations	$3 \times$ Sphere Detector	114 Mbit/s	0.78 mm^2	45 mW	114 Mbit/s
T = 30 Mbit/s, 0 iterations	$1 \times$ Sphere Detector	38 Mbit/s	0.26 mm^2	15 mW	38 Mbit/s
T = 30 Mbit/s, 1 iteration	$2 \times$ Sphere Detector	76 Mbit/s	0.52 mm^2	30 mW	30 Mbit/s

by a similar factor depending on the targeted throughput. Therefore, a list size of 4096 is not feasible.

The sphere detector is able to reach the target communications performance for all given signal-to-noise ratios with up to two iterations. However, the throughput is much lower than for the fixed effort detector. At 20 dB the radius can be lowered to increase the throughput as a frame-error rate of 10^{-3} is achieved easily. At 16 dB, 2 outer iterations are necessary heavily reducing the throughput to where it is not adequate anymore.

In the _throughput centric_ scenarios, we analyze which parallelism is needed for the MIMO detector to reach a certain system throughput. For open-loop systems, the system throughput linearly increases with the number of detector instantiations. For an open-loop throughput of 300 Mbit/s, three fixed effort list detector instances or eight sphere detector instances are needed. Even though the MIMO detectors have a throughput higher than 300 Mbit/s, the system throughput is in this case limited by the channel decoder running at a constant throughput of 300 Mbit/s. For most throughput centric scenarios, the resulting area for both detectors are similar. The power consumption, however, for the sphere detector is much lower. This can be explained by the additional power needed for the list storage and the LLR units on one hand. Furthermore, the fixed

effort detector architecture processes eight different nodes in parallel whereas the sphere detector is only working on two nodes in parallel which are siblings in the tree.

In summary, the fixed effort list detector is advantageous if a high throughput has to be guaranteed at a reasonable communications performance. However, best communications performance cannot be achieved because the required higher list sizes would imply infeasibly huge list storage memories. The depth-first sphere detector achieves best communications performance. With multiple instances, the sphere detector achieves a high throughput at a decent area and very good energy efficiency.

7.4. Open-Loop versus Closed-Loop Considerations. After comparing sphere detector and fixed effort detector for different application scenarios, we will now look at the effect on the whole system when moving from an open-loop implementation to a closed-loop implementation. For this analysis, we set the detector throughput to 300 Mbit/s balancing the throughput between MIMO detector and channel decoder.

The power consumption of the system memories does not depend on a specific detector architecture but only on the MIMO detector throughput. Based on the number of accesses (e.g., 4 read accesses on Y_HAT per detection),

TABLE 6: Difference in implementation cost between an open-loop and a closed-loop system. Area and energy efficiency drop by more than a factor of 2 for the iterative system.

| Employed MIMO detector	Open-loop system, 0 iterations		Closed-loop system, 1 iteration	
	Sphere detector	Fixed effort detector	Sphere detector	Fixed effort detector
System throughput	300 Mbit/s	300 Mbit/s	75 Mbit/s	75 Mbit/s
Total system area	$2.5\,\text{mm}^2$	$2.3\,\text{mm}^2$	$2.8\,\text{mm}^2$	$3.6\,\text{mm}^2$
Total system power	180 mW	520 mW	195 mW	365 mW
System area efficiency	$120\,\text{(Mbit/s)/mm}^2$	$130\,\text{(Mbit/s)/mm}^2$	$27\,\text{(Mbit/s)/mm}^2$	$21\,\text{(Mbit/s)/mm}^2$
System energy efficiency	1.7 bit/nJ	0.6 bit/nJ	0.4 bit/nJ	0.2 bit/nJ
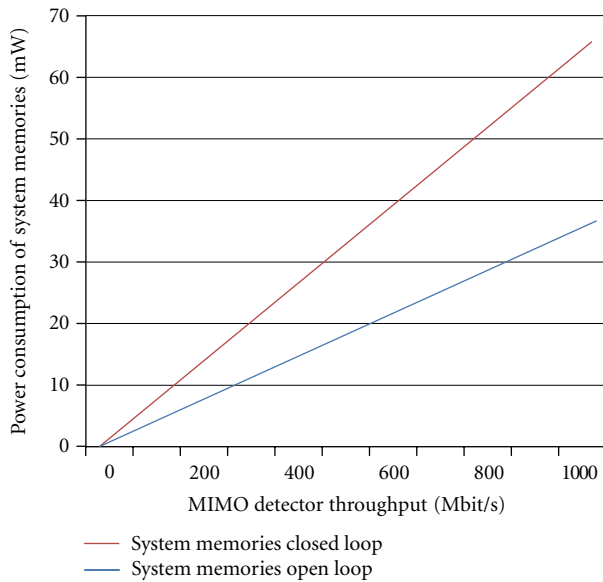
FIGURE 8: Power consumption of system memories depending on the MIMO detector throughput.

we determined the average power for each memory (see Figure 8). The power consumption of the memories for closed-loop decoding is approximately twice as high as in open-loop decoding. This stems from the fact that certain system memories are not needed in open-loop decoding (see Section 7.1). The implementation data of channel preprocessing and channel decoder have been summarized in Table 4.

Table 6 shows the main characteristics of the resulting open- and closed-loop *systems* employing sphere detector or fixed effort detector, respectively. We determine area and energy efficiency according to [31]. Higher numbers represent a higher efficiency. The throughput of the closed-loop system drops by a factor of 4 because only one codeword is processed at a time. This scheduling has a positive effect on the power consumption as each component is only active 50% of the time. The gain in communications performance by the outer iteration is between 3 and 4 dB. However, it can be observed that area and energy efficiency do not decrease by a factor of 2 as might be expected. In fact, the efficiency of the closed loop-system drops by factors between 3 and 6 compared to the open-loop system.

8. Conclusions

Multiple-antenna systems offer an increased bandwidth efficiency compared to single-antenna systems. Iterative receivers which exchange reliability information between MIMO detector and channel decoder will become mandatory in the near future. Choosing the MIMO detector algorithm and architecture from one of the various existing approaches has big effects on the complete system. In this paper, we have compared the depth-first variable throughput sphere detector to the breadth-first fixed effort detector in communication centric and throughput centric application scenarios. The fixed effort detector is advantageous if a high throughput has to be ensured at moderate communications performance. However, it has been observed that the sphere detector shows excellent behaviour for one outer iteration. Even with multiple instances, it obtains a decent area and very good energy efficiency.

Furthermore, we have presented an analysis of all components of the iterative receiver including channel preprocessing, MIMO detection, channel decoding, and system memories. We have shown that area and power efficiency decrease by more than a factor of 2 when changing from an open-loop decoder implementation to a closed-loop decoder employing 1 iteration independent of the choice of the MIMO detector.

Acknowledgment

References

[1] B. M. Hochwald and S. Ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389–399, 2003.

[2] H. Vikalo, B. Hassibi, and T. Kailath, "Iterative decoding for MIMO channels via modified sphere decoding," *IEEE Transactions on Wireless Communications*, vol. 3, no. 6, pp. 2299–2311, 2004.

[3] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI Implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, 2005.

[4] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: algorithms and VLSI implementation," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 2, pp. 290–300, 2008.

[5] E. M. Witte, F. Borlenghi, G. Ascheid, R. Leupers, and H. Meyr, "A scalable VLSI architecture for soft-input soft-output single tree-search sphere decoding," *IEEE Transactions on Circuits and Systems II: express Briefs*, vol. 57, no. 9, Article ID 5570931, pp. 706–710, 2010.

[6] C.-H. Liao, T.-P. Wang, and T.-D. Chiueh, "A 74.8 mW soft-output detector IC for 8×8 spatial-multiplexing MIMO communications," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 2, Article ID 5405138, pp. 411–421, 2010.

[7] L. Liu, F. Ye, X. Ma, T. Zhang, and J. Ren, "A 1.1-Gb/s 115-pJ/bit configurable MIMO detector using 0.13-muhboxm CMOS technology," *IEEE Transactions on Circuits and Systems II*, vol. 57, no. 9, pp. 701–705, 2010.

[8] C. Studer, S. Fateh, and D. Seethaler, "A 757 Mb/s 1.5 mm^2 90 nm CMOS soft-input soft-output MIMO detector for IEEE 802.11n," in *Proceedings of the 36th European Solid State Circuits Conference (ESSCIRC'10)*, pp. 530–533, Seville, Spain, September 2010.

[9] D. Garrett, L. Davis, S. Ten Brink, B. Hochwald, and G. Knagge, "Silicon complexity for maximum likelihood MIMO detection using spherical decoding," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, pp. 1544–1552, 2004.

[10] D. Wubben, R. Bohnke, V. Kuhn, and K.-D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition,," in *Proceedings of the IEEE 58th Vehicular Technology Conference (VTC'03)*, vol. 1, pp. 508–512, October 2003.

[11] G. J. Foschini, G. D. Golden, R. A. Valenzuela, and P. W. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 11, pp. 1841–1852, 1999.

[12] 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.

[13] K. W. Wong, C. Y. Tsui, R. S. K. Cheng, and W. H. Mow, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'02)*, vol. 3, pp. III-273–III-276, May 2002.

[14] L. G. Barbero and J. S. Thompson, "Fixing the complexity of the sphere decoder for MIMO detection," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, Article ID 4543065, pp. 2131–2142, 2008.

[15] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2201–2214, 2002.

[16] B. Mennenga and G. Fettweis, "Search sequence determination for tree search based detection algorithms," in *Proceedings of the IEEE Sarnoff Symposium (SARNOFF'09)*, pp. 1–6, April 2009.

[17] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI Implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1576, 2005.

[18] M. Wenk, A. Burg, M. Zellweger, C. Studer, and W. Fichtner, "VLSI implementation of the list sphere algorithm," in *Proceedings of the 24th Norchip Conference*, pp. 107–110, Linkoping, Sweden, November 2006.

[19] M. T. Gamba and G. Masera, "Look-ahead sphere decoding: algorithm and VLSI architecture," *IET Communications*, vol. 5, no. 9, pp. 1275–1285, 2011.

[20] C. Studer and H. Bölcskei, "Soft-input soft-output sphere decoding," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT'08)*, pp. 2007–2011, Toronto, Canada, July 2008.

[21] M. Wenk, L. Bruderer, A. Burg, and C. Studer, "Area- and throughput-optimized VLSI architecture of sphere decoding," in *Proceedings of the 18th IEEE/IFIP International Conference on VLSI and System-on-Chip (VLSI-SoC'10)*, pp. 189–194, Madrid, Spain, September 2010.

[22] S. Chen and T. Zhang, "Low power soft-output signal detector design for wireless MIMO communication systems," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED'07)*, pp. 232–237, 2007.

[23] S. A. Laraway and B. Farhang-Boroujeny, "Implementation of a Markov chain Monte Carlo based multiuser/MIMO detector," *IEEE Transactions on Circuits and Systems I*, vol. 56, no. 1, pp. 246–255, 2009.

[24] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K. D. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *Electronics Letters*, vol. 37, no. 22, pp. 1348–1350, 2001.

[25] L. G. Barbero and J. S. Thompson, "Extending a fixed-complexity sphere decoder to obtain likelihood information for turbo-MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 5, pp. 2804–2814, 2008.

[26] I. 802.16, Local and metropolitan area networks; Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems; Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands.

[27] G. H. Golub and C. F. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, London, UK, 3rd edition, 1996.

[28] G. L. Nazar, C. Gimmler, and N. Wehn, "Implementation comparisons of the QR decomposition for MIMO detection," in *Proceedings of the 23rd Symposium on Integrated Circuits and Systems Design (SBCCI'10)*, pp. 210–214, September 2010.

[29] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[30] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. IT-20, no. 2, pp. 284–287, 1974.

[31] F. Kienle, N. Wehn, and H. Meyr, "On complexity, energy and implementation-efficiency of channel decoders," *IEEE Transactions on Communications*, vol. 59, no. 12, pp. 3301–3310, 2011.

[32] C. Studer, *Iterative MIMO decoding: algorithms and VLSI implementation aspects*, Ph.D. dissertation, ETH Zürich, Zurich, Switzerland, 2009.