FPGA IMPLEMENTATION OF MMSE METRIC BASED EFFICIENT NEAR-ML DETECTION

M. Joham[1], L. G. Barbero[2], T. Lang[1], W. Utschick[1], J. Thompson[3], T. Ratnarajah[2]

1) Associate Institute for Signal Processing, Technische Universität München, 80290 Munich, Germany
Telephone: +49 89 289-28510, Fax: +49 89 289-28504, Email: joham@tum.de
2) Institute of Electronics, Communications and Information Technology,
Queen's University Belfast, Belfast BT3 9DT, UK
3) Institute for Digital Communications, University of Edinburgh, Edinburgh EH9 3JL, UK

ABSTRACT

We consider the problem of detecting a vector signal transmitted over a multiple input-multiple output (MIMO) channel. A number of suboptimal detectors have been proposed to solve that problem, given that maximum likelihood (ML) detection is NP-hard. After reviewing the main concepts of the ML and the minimum mean square error (MMSE) metrics, we introduce an unbiased MMSE metric that can be applied to existing MIMO detectors in order to improve their performance. Applying the biased and unbiased MMSE metrics together with a real-valued representation of the system, the performance and complexity of a number suboptimal MIMO detectors is compared in this paper, showing how the QR decomposition-M (QRD-M) can be used to approximate ML performance with low complexity. In order to further validate those results, the QRD-M algorithm has been implemented on a field-programmable gate array (FPGA) platform, showing an excellent fixed-point performance under real-time conditions. Finally, the resulting real-time detector has been compared to state-of-the-art detectors previously implemented, in terms of complexity, error performance and throughput.

1. INTRODUCTION

For multiple input-multiple output (MIMO) systems without any channel knowledge at the transmitter, e.g., the uplink of a multi-user system [1] or a MIMO system with linear dispersion coding [2], maximum likelihood (ML) detection is an NP-hard problem [3]. Therefore, many suboptimal schemes were proposed with polynomial complexity. Besides linear detection [1], the ordered decision feedback equalizer (DFE) principle of vertical-Bell Labs layered space time (V-BLAST) [4] has a quadratic complexity per received vector and the filters can be computed very efficiently via a symmetrically permuted Cholesky factorization [5]. However, there is a large performance gap between ML and V-BLAST detection which can be reduced by lattice reduction (LR) [6] schemes [7]-[10] for high signal to noise ratio (SNR) [11].

Whereas the detection order of V-BLAST is based only on the statistics of the signals, the dynamic nulling-and-cancelling (DNC) of [12] computes the DFE detection order based on the signal statistics and the received vector itself. The approach to precoding of [13] can also be applied to detection, resulting in a multiple application of DFE with different indices for the data stream detected first. In [14], the underlying lattice of the detection problem is decomposed into cosets with larger Voronoi cells and LR is applied. The schemes of [12]-[14] lead to a cubic complexity per received vector but clearly outperform V-BLAST with and without LR. Semidefinite relaxation was applied to the ML detection problem in [15], leading to a complexity order of 3.5.

The MIMO detection problem can be interpreted as a tree search [16], whose optimal solution is obtained with the algorithms discussed in [17]. A powerful suboptimal breadth-first decoder is the M-algorithm [18]. It is called the QR decomposition-M (QRD-M) algorithm if applied to the tree search problem [19]-[21].

In [17], it was shown that using a minimum mean-square error (MMSE) metric instead of the original ML metric leads to sphere decoders (SDs) with less complexity and the superiority of the MMSE metric compared to the ML metric for suboptimal detectors was highlighted in [16]. Due to this result, we will concentrate on detectors based on the MMSE metric in this paper, using the ML metric only on the SD to ensure optimality.

Over the last years, several application-specific integrated circuit (ASIC) and field-programmable gate array (FPGA) implementations of the SD or close-to-ML detectors were reported in [22]-[27]. Whereas an implementation of the SD based on the l^2-norm and l^1-norm was proposed in [23], most contributions focused on the M-algorithm (or K-best algorithm) [22],[24]-[26]. In [27], a fixed-throughput SD was proposed. Interestingly, all hardware implementation were restricted to a zero forcing (ZF) formulation so far.

We first show how to derive the MMSE metric from the ML metric, where it becomes clear that the MMSE metric is also ML optimal for constant modulus alphabets. Second, we present an unbiased MMSE metric motivated by the un-

biased MMSE DFE presented in [28]. Third, we describe the different suboptimal detectors in a common framework and compare their complexities. Fourth, we report the implementation of the QRD-M algorithm on an FPGA platform based on the MMSE metric and compare it to the state-of-the-art ASIC implementations of close-to-ML detectors.

2. SYSTEM MODEL

To keep the presentation simple, we use the standard MIMO model with an $N \times B$ channel matrix $\boldsymbol{H}$, whose i.i.d. entries are circularly symmetric complex Gaussian with unit variance. The received signal

$$\boldsymbol{x} = \boldsymbol{H}\boldsymbol{s} + \boldsymbol{\eta} \in \mathbb{C}^N \tag{1}$$

is the superposition of the noise $\boldsymbol{\eta} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}_N, \boldsymbol{C}_{\boldsymbol{\eta}})$ and the data signal $\boldsymbol{s} \in \mathbb{A}^B$ transformed by $\boldsymbol{H}$. For quadrature amplitude modulation (QAM) alphabets, i.e. the real and imaginary parts of the elements of the alphabet $\mathbb{A}$ are weighted integers, $\boldsymbol{H}\boldsymbol{s}$ is an element of a lattice. Additionally, we assume that $\mathrm{E}[\boldsymbol{s}] = \mathbf{0}_B$ and $\mathrm{E}[\boldsymbol{s}\boldsymbol{s}^{\mathrm{H}}] = \mathbf{I}_B$. Note that we use a notation with complex vectors and matrices for conciseness, but our simulations are based on a real-valued representation, since the suboptimal detection schemes benefit from it [29].

3. ML AND MMSE METRIC

For ML detection, we assume that $\boldsymbol{s}$ is deterministic but unknown and maximize the likelihood $\mathrm{f}_{\boldsymbol{x}}(\boldsymbol{x}; \boldsymbol{s})$, i.e.,

$$\tilde{\boldsymbol{s}}_{\text{ML}} = \underset{\boldsymbol{s} \in \mathbb{A}^B}{\operatorname{argmax}}\, \mathrm{f}_{\boldsymbol{x}}(\boldsymbol{x}; \boldsymbol{s}).$$

Due to the Gaussianity of $\boldsymbol{x}$ for deterministic $\boldsymbol{s}$, we obtain the rule

$$\tilde{\boldsymbol{s}}_{\text{ML}} = \underset{\boldsymbol{s} \in \mathbb{A}^B}{\operatorname{argmin}}\, \mu_{\text{ML}}(\boldsymbol{s}) \tag{2}$$

with the ML metric

$$\mu_{\text{ML}}(\boldsymbol{s}) = \left(\boldsymbol{G}_{\text{ZF}}\boldsymbol{x} - \boldsymbol{s}\right)^{\mathrm{H}} \boldsymbol{C}_{\boldsymbol{\eta}}^{-1} \boldsymbol{H}^{\mathrm{H}} \boldsymbol{C}_{\boldsymbol{\eta}}^{-1} \boldsymbol{H} \left(\boldsymbol{G}_{\text{ZF}}\boldsymbol{x} - \boldsymbol{s}\right) \tag{3}$$

and $\boldsymbol{G}_{\text{ZF}} = (\boldsymbol{H}^{\mathrm{H}}\boldsymbol{C}_{\boldsymbol{\eta}}^{-1}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{H}}\boldsymbol{C}_{\boldsymbol{\eta}}^{-1}$ is the zero-forcing filter. The corresponding MMSE metric can be written as

$$\begin{aligned}
\mu_{\text{MMSE}}(\boldsymbol{s}) &= \mu_{\text{ML}}(\boldsymbol{s}) + \boldsymbol{s}^{\mathrm{H}}\boldsymbol{s} \\
&\quad - \boldsymbol{x}^{\mathrm{H}}\boldsymbol{G}_{\text{ZF}}^{\mathrm{H}} \left(\mathbf{I}_B + \left(\boldsymbol{H}^{\mathrm{H}}\boldsymbol{C}_{\boldsymbol{\eta}}^{-1}\boldsymbol{H}\right)^{-1}\right)^{-1} \boldsymbol{G}_{\text{ZF}}\boldsymbol{x} \\
&= \left\| \left(\boldsymbol{H}^{\mathrm{H}}\boldsymbol{C}_{\boldsymbol{\eta}}^{-1}\boldsymbol{H} + \mathbf{I}_B\right)^{1/2} \left(\boldsymbol{G}_{\text{MMSE}}\boldsymbol{x} - \boldsymbol{s}\right) \right\|_2^2 \\
&= \left\| \boldsymbol{D}^{-1/2} \left(\boldsymbol{G}_{\text{MMSE-DFE}}\boldsymbol{x} - \boldsymbol{L}^{-1}\boldsymbol{\Pi}\boldsymbol{s}\right) \right\|_2^2
\end{aligned} \tag{4}$$

where $\boldsymbol{G}_{\text{MMSE}} = (\boldsymbol{H}^{\mathrm{H}}\boldsymbol{C}_{\boldsymbol{\eta}}^{-1}\boldsymbol{H} + \mathbf{I}_B)^{-1}\boldsymbol{H}^{\mathrm{H}}\boldsymbol{C}_{\boldsymbol{\eta}}^{-1}$ is the linear MMSE equalizer and $\boldsymbol{G}_{\text{MMSE-DFE}} = \boldsymbol{D}\boldsymbol{L}^{\mathrm{H}}\boldsymbol{\Pi}\boldsymbol{H}^{\mathrm{H}}\boldsymbol{C}_{\boldsymbol{\eta}}^{-1}$ is the MMSE V-BLAST feedforward filter. The diagonal $\boldsymbol{D}$ and the unit lower triangular $\boldsymbol{L}$ result from the symmetrically permuted Cholesky factorization of the MMSE matrix

$$\boldsymbol{\Pi}\left(\boldsymbol{H}^{\mathrm{H}}\boldsymbol{C}_{\boldsymbol{\eta}}^{-1}\boldsymbol{H} + \mathbf{I}_B\right)^{-1}\boldsymbol{\Pi}^{\mathrm{T}} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{H}}$$

and $\boldsymbol{\Pi}$ is a permutation matrix representing the detection order [5]. Clearly, subtracting the term depending on $\boldsymbol{x}$ from $\mu_{\text{ML}}(\boldsymbol{s})$ does not change the minimizer of (2). However, minimizing $\mu_{\text{MMSE}}(\boldsymbol{s})$ is only equivalent to minimizing $\mu_{\text{ML}}(\boldsymbol{s})$ for constant modulus alphabets, i.e. $\boldsymbol{s}^{\mathrm{H}}\boldsymbol{s} = B$ for any $\boldsymbol{s} \in \mathbb{A}^B$. Otherwise, $\boldsymbol{s}^{\mathrm{H}}\boldsymbol{s}$ depends on $\boldsymbol{s}$ and the MMSE rule

$$\tilde{\boldsymbol{s}}_{\text{MMSE}} = \underset{\boldsymbol{s} \in \mathbb{A}^B}{\operatorname{argmin}}\, \mu_{\text{MMSE}}(\boldsymbol{s}) \tag{5}$$

leads to slightly worse results than the ML rule (2), since the MMSE metric $\mu_{\text{MMSE}}(\boldsymbol{s})$ has a bias.

4. UNBIASED MMSE METRIC

The form of the MMSE metric with the MMSE V-BLAST feedforward filter $\boldsymbol{G}_{\text{MMSE-DFE}}$ (cf. [5]) in (4) shows that MMSE V-BLAST is a successive computation of $\boldsymbol{s}$ exploiting the triangular structure of $\boldsymbol{L}^{-1}$. The k-th quantization rule of the MMSE V-BLAST results from (4), when the absolute value of the k-th entry of $\boldsymbol{G}_{\text{MMSE-DFE}}\boldsymbol{x} - \boldsymbol{L}^{-1}\boldsymbol{\Pi}\boldsymbol{s}$ is minimized with respect to the k-th entry of $\boldsymbol{\Pi}\boldsymbol{s}$ instead of the whole norm (4).

According to [28], quantizing the outputs of an unbiased MMSE feedforward filter is superior to quantizing the outputs of $\boldsymbol{G}_{\text{MMSE-DFE}}$. The unbiased MMSE feedforward filter can be found by forcing the weights of the symbols to be detected to one. Thus, we weight the b-th output of the MMSE V-BLAST feedforward filter with a scalar γ_b such that

$$\gamma_b \boldsymbol{e}_b^{\mathrm{T}} \boldsymbol{G}_{\text{MMSE-DFE}} \boldsymbol{H} \boldsymbol{\Pi}^{\mathrm{T}} \boldsymbol{e}_b = 1 \qquad b = 1, \ldots, B.$$

Here, $\boldsymbol{e}_b$ is the b-th column of the identity matrix $\mathbf{I}_B$. Due to the definition of $\boldsymbol{G}_{\text{MMSE-DFE}}$, we have that

$$\begin{aligned}
\boldsymbol{e}_b^{\mathrm{T}} \boldsymbol{G}_{\text{MMSE-DFE}} \boldsymbol{H} \boldsymbol{\Pi}^{\mathrm{T}} \boldsymbol{e}_b &= \boldsymbol{e}_b^{\mathrm{T}} \boldsymbol{D} \boldsymbol{L}^{\mathrm{H}} \boldsymbol{\Pi} \boldsymbol{H}^{\mathrm{H}} \boldsymbol{C}_{\boldsymbol{\eta}}^{-1} \boldsymbol{H} \boldsymbol{\Pi}^{\mathrm{T}} \boldsymbol{e}_b \\
&= \boldsymbol{e}_b^{\mathrm{T}} \boldsymbol{L}^{-1} \boldsymbol{e}_b - \boldsymbol{e}_b^{\mathrm{T}} \boldsymbol{D} \boldsymbol{L}^{\mathrm{H}} \boldsymbol{e}_b \\
&= 1 - \boldsymbol{e}_b^{\mathrm{T}} \boldsymbol{D} \boldsymbol{e}_b
\end{aligned}$$

where we exploited the diagonal structure of $\boldsymbol{D}$ and the property of $\boldsymbol{L}$ to be unit triangular. Therefore, $\boldsymbol{x}$ must be transformed by

$$\boldsymbol{G}_{\text{UB}} = \left(\mathbf{I}_B - \boldsymbol{D}\right)^{-1} \boldsymbol{G}_{\text{MMSE-DFE}} \tag{6}$$

to fulfill the above requirements for unbiasedness. With this result, we get

$$\mu_{\text{MMSE}}(\boldsymbol{s}) = \left\| \boldsymbol{D}_{\text{UB}} \left(\boldsymbol{G}_{\text{UB}}\boldsymbol{x} - \left(\mathbf{I}_B - \boldsymbol{D}\right)^{-1} \boldsymbol{L}^{-1} \boldsymbol{\Pi}\boldsymbol{s} \right) \right\|_2^2$$

with $\boldsymbol{D}_{\text{UB}} = \boldsymbol{D}^{-1/2}(\mathbf{I}_B - \boldsymbol{D})$. Since this expression is equivalent to (4), a successive computation of the entries of

$\boldsymbol{s}$ again leads to MMSE V-BLAST. However, if we set the diagonal entries of $(\mathbf{I}_B - \boldsymbol{D})^{-1}\boldsymbol{L}^{-1}$ to one, i.e., replace it by $(\mathbf{I}_B - \boldsymbol{D})^{-1}\boldsymbol{L}^{-1} - \boldsymbol{D}(\mathbf{I}_B - \boldsymbol{D})^{-1}$, a successive computation of $\boldsymbol{s}$ results in an unbiased MMSE V-BLAST. After replacing $(\mathbf{I}_B - \boldsymbol{D})^{-1}\boldsymbol{L}^{-1}$ by $(\mathbf{I}_B - \boldsymbol{D})^{-1}(\boldsymbol{L}^{-1} - \boldsymbol{D})$, we obtain the unbiased MMSE metric

$$\mu_{\text{UB}}(\boldsymbol{s}) = \|\boldsymbol{D}_{\text{UB}}\left(\boldsymbol{G}_{\text{UB}}\boldsymbol{x} - (\mathbf{I}_B - \boldsymbol{F}_{\text{UB}})\,\boldsymbol{\Pi}\boldsymbol{s}\right)\|_2^2 \qquad (7)$$

where $\boldsymbol{F}_{\text{UB}} = (\mathbf{I}_B - \boldsymbol{D})^{-1}(\mathbf{I}_B - \boldsymbol{L}^{-1})$ which is the strictly lower triangular feedback filter for unbiased MMSE V-BLAST. Since the MMSE metric $\mu_{\text{MMSE}}(\boldsymbol{s})$ is ML-optimal for constant modulus alphabets, e.g. 4-QAM, a minimization of the unbiased MMSE metric deteriorates the result, as the unbiased MMSE metric is not ML optimal. However, suboptimal detection of non-constant modulus alphabets is slightly improved by imposing the constraint of unbiasedness.

5. SUBOPTIMAL DETECTION SCHEMES

The three metrics, viz., the ML, the MMSE, and the unbiased MMSE metric, can be expressed as

$$\mu(\boldsymbol{s}) = \left\|\hat{\boldsymbol{D}}\left(\hat{\boldsymbol{x}} - \hat{\boldsymbol{L}}\boldsymbol{\Pi}\boldsymbol{s}\right)\right\|_2^2.$$

The differences between the metrics lie in the diagonal matrix $\hat{\boldsymbol{D}}$, the unit lower triangular matrix $\hat{\boldsymbol{L}}$ and the permutation matrix $\boldsymbol{\Pi}$, and how the received signal is transformed to get $\hat{\boldsymbol{x}}$. Therefore, we can rewrite any of the three metrics as follows

$$\mu(\boldsymbol{s}) = \sum\nolimits_{i=1}^{B} \hat{d}_{i,i}^2 \left|\hat{x}_i - s_{b_i} - \sum\nolimits_{j=1}^{i-1} \hat{\ell}_{i,j} s_{b_j}\right|^2 \qquad (8)$$

where the i-th element in the j-th column of a matrix $\boldsymbol{A}$ is denoted by $a_{i,j}$, $\hat{x}_i$ is the i-th element of $\hat{\boldsymbol{x}}$, and s_{b_j} is the j-th element of $\boldsymbol{\Pi}\boldsymbol{s}$. The above expression for the metrics helps to understand why the ML problem is a tree search. The first term with $i = 1$ only depends on s_{b_1} and the weights for the $|\mathbb{A}|$ branches leaving the root of the tree result from the different values for s_{b_1}. Accordingly, $|\mathbb{A}|$ branches (corresponding to all possible values of $s_{b_{k+1}}$) leave from the branch belonging to some choice of $s_{b_1}, \ldots, s_{b_k}$. So, we end up with a tree with $|\mathbb{A}|^B$ leaves. ML detection tries to find the leave with the least overall metric. Therefore, the complexity of ML detection is exponential in B.

The following orders of complexity are for complex operations. When the real-valued representation is used, B and $|\mathbb{A}|$ must be replaced by $2B$ and $\sqrt{|\mathbb{A}|}$, respectively.

V-BLAST computes the symbols successively, i.e. $\tilde{s}_{b_k}$ is computed by minimizing the k-th summand of (8), where the already found $\tilde{s}_{b_1}, \ldots, \tilde{s}_{b_{k-1}}$ are fixed. The detection order $b_1, \ldots, b_B$ is computed based on the signals' statistics [5]. The complexity of $\mathrm{O}(3B^2)$ per received vector results from the computation of $\hat{\boldsymbol{x}}$ and the multiplication with $\hat{\boldsymbol{L}}$. The filter computation has cubic complexity [5].

DNC applies the V-BLAST procedure to detect the symbols, but the detection order is recomputed for every received vector [12]. Thus, the filter computation must be repeated for every vector resulting in a complexity of $\mathrm{O}(17B^3/6)$ per received vector. The pre-computation has cubic complexity [12].

The approach of [14] divides the underlying lattice into M cosets and solves the detection problem in every coset, where V-BLAST is used. Consequently, the complexity per received vector is $\mathrm{O}(3MB^2)$. For the filter computation, the Lenstra-Lenstra-Lovász (LLL) algorithm is applied several times and the resulting complexity of the filter computation is quartic up to quintic.

Applying the procedure of [13] to the MIMO detection problem leads to a multiple application of V-BLAST with different values for b_1. Let M be the number of values for b_1. Then, the complexity is $\mathrm{O}(2MB^2 + 2B^2)$ for $M > 2$. For $M = 1$, the system converges to the original V-BLAST. To come close to ML, M should be above $B/2$ and the complexity is about $\mathrm{O}(B^3)$. The filter computation has cubic up to quartic complexity depending on the choice of M [13].

The QRD-M algorithm keeps the M best choices for $s_{b_1}, \ldots, s_{b_k}$ at the k-th stage [19]. When moving to the next stage, the $|\mathbb{A}|$ different values for $s_{b_{k+1}}$ are tested to find the M best choices for $s_{b_1}, \ldots, s_{b_{k+1}}$, where $s_{b_{k+1}}$ is added to the given M choices for $s_{b_1}, \ldots, s_{b_k}$. At the k-th stage, the sum for $i = 1, \ldots, k$ is used as metric instead of the full metric in (8). An efficient implementation of the QRD-M algorithm, taking into account the redundancies present in the computations and the properties of QAM alphabets, leads to a complexity of $\mathrm{O}(M(B^2 + 3B\min(B, |\mathbb{A}|)) + 2B^2)$ per received vector. We observed that $M = \log_2(B)\log_4(|\mathbb{A}|)$ is a good choice for the number of survivors. In addition, the filter computation has a cubic complexity [5].

Finally, the convex relaxation approach of [15] has a complexity of $\mathrm{O}(B^{3.5})$ per received vector, i.e. it is above cubic. Thus, we do not include this approach in our comparison.

6. FPGA IMPLEMENTATION

The QRD-M algorithm has been implemented, using a rapid prototyping methodology, to evaluate its suitability for real-time MIMO detection. A 4×4 system with 16-QAM modulation with $M = 4$ has been considered where the real decomposition of the system has been used for the implementation.

6.1. MIMO Prototyping System

An FPGA-based rapid prototyping system has been used for the implementation of the QRD-4 algorithm, since it provides the flexibility required to move quickly from a computer-based simulation to its hardware implementation. As opposed

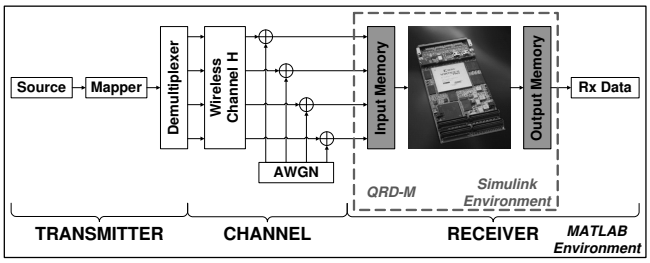

Fig. 1. Hardware-in-the-loop MIMO system diagram.

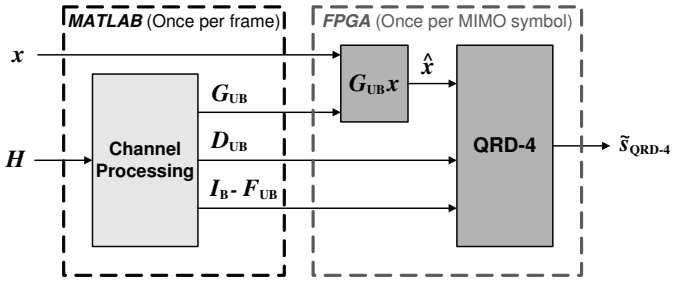

Fig. 2. Partitioning of the QRD-4 between MATLAB and the FPGA.

to previous prototyping approaches that looked at the entire MIMO system, the focus of our approach is on the analysis of the MIMO detector. The FPGA platform has been provided by Alpha Data Ltd and it consists of an ADC-PMC PCI adapter board that hosts an ADM-XP board with a Xilinx Virtex-II-Pro FPGA (XC2VP70).

The rapid prototyping methodology selected is based on The Mathwork's MATLAB and Simulink and Xilinx's DSP System Generator tailored to Alpha Data's FPGA boards. Initially, MATLAB is used to implement a complete MIMO system including transmitter, channel and receiver. The QRD-4 algorithm is then implemented on the FPGA using the DSP System Generator. The development of the FPGA model is embedded in a Simulink testbench that facilitates the debugging of the algorithm in the development stage, with the possibility of monitoring every signal in the FPGA model. The QRD-4 design is then synthesized for the FPGA using Xilinx's synthesis tools. The hardware design and a Simulink-based memory interface are integrated into the MATLAB MIMO system as shown in Fig. 1. This rapid prototyping methodology and system allows us to perform real-time hardware-in-the-loop testing of the algorithm.

6.2. FPGA Architecture

The first step in the implementation of the QRD-4 is the partitioning of the architecture between MATLAB and the FPGA, shown in Fig. 2. Initially, MATLAB performs the sections of the algorithm that are required only once per block, representing a single channel realization, in order to obtain the matrices $\boldsymbol{D}_{\text{UB}}$, $\boldsymbol{G}_{\text{UB}}$ and $\mathbf{I}_B - \boldsymbol{F}_{\text{UB}}$ in (7). The FPGA contains the matrix-vector product to obtain $\hat{\boldsymbol{x}}$ and the QRD-4 algorithm, which are required once per MIMO symbol. A memory interface is used between MATLAB and the FPGA in order to pass the input/output parameters to/from the FPGA.

Given that the computation of $\hat{\boldsymbol{x}}$ corresponds only to a matrix-vector product, we focus on the description of the tasks performed by the QRD-4 block on the FPGA. In the first level, $i = 1$, since we are at the root of the tree and $M = |\mathbb{A}| = 4$, the four constellation points corresponding to the real (or imaginary) components of a 16-QAM constellation are selected as best choices, as described in Section 5. The weights of the branches associated to those constellation points are also computed, i.e., the first term of the sum in (8) for each s_{b_1}, resulting in four child nodes originating from the root node. Although those operations are independent and could be performed in parallel, the weights are computed sequentially in four cycles, in order to reduce the resource use on the FPGA and provide a more balanced trade-off between hardware complexity and throughput performance (i.e., detection speed). In levels $i = 2, \ldots, 7$, the following tasks are performed:

1. Each one of the $M = 4$ nodes from the previous level is extended calculating the weights of the branches associated to the $|\mathbb{A}| = 4$ constellation points, resulting in a total of $|\mathbb{A}| \cdot M = 16$ child nodes. All the child nodes and branch weights originating from the same parent node are computed in parallel. In addition, they are directly sorted in increasing weight order using a look-up-table (LUT) as detailed in [30].[1] Thus, in every one of the four cycles, a set of $|\mathbb{A}|$ child nodes, ordered according to increasing weight, is obtained.

2. Since the QRD-4 needs to select the M child nodes with the smallest associated weights, a sorting unit is required in these levels, to select the best M choices out of $|\mathbb{A}| \cdot M$ candidates. Making use of the fact that the child nodes are grouped in four independently sorted sets, a merge-sort algorithm can be used to obtain the best M choices [31]. That algorithm, briefly described at the end of this section, considerably reduces the complexity of a same size odd-even transposition sorting algorithm. The merge-sort algorithm is implemented in two stages, one that selects the M child nodes out of $2|\mathbb{A}|$ candidates every two cycles, and one that finally selects the best M choices out of $2M$ candidates every four cycles. Those M child nodes are then passed to the following level $i \leftarrow i + 1$.

In the last level, $i = 8$, each one of the M nodes from the previous level is extended selecting only the child node associated to the branch with the smallest weight. Thus, after

[1] Note that a LUT can be used because the real-decomposition of the sys-

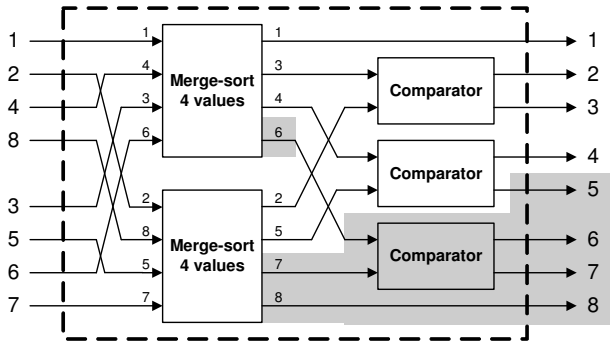

Fig. 3. Merge-sort network of 8 elements.

the four cycles required to go over the M nodes, M full paths are obtained on the tree. The solution of the QRD-4 algorithm is given then by the constellation points associated to the path with the minimum metric as defined in (8).

6.3. Merge-sort algorithm

The merge-sort algorithm takes as an input two sets of $n/2$ independently sorted elements to generate an output set of n sorted elements. The main advantage of this algorithm is that it has a considerably reduced complexity, including the sorting of the two input sets, compared to other parallel sorting algorithms, like the odd-even transposition sorting algorithm [31]. This is of special importance in the implementation of M-algorithm-based MIMO detectors, where the sorting stages represent a considerable percentage of the algorithm complexity [24]. The steps performed by a merge-sort network of n elements are:

1. The two sets are merged using two merge-sort networks of $n/2$ elements.

2. The merged set is finally sorted using a set of $n-1$ comparators.

Therefore, a merge-sort network of n values can be constructed applying the same rule recursively, that is, by using two $n/2$ merge-sort networks and a bank of $n/2-1$ comparators. As stated above, the input sets to the merge-sort network need to be previously sorted, which is guaranteed in the QRD-4 by applying the LUT-based method proposed in [30]. Fig. 3 shows an example of the operation of a merge-sort network of 8 elements once the two input sets of 4 elements have been independently sorted. The shaded area shows the parts of the merge-sort network that are not required in the QRD-4 algorithm, since only the first $M=4$ elements of the sorted set are needed, further reducing the hardware complexity.

tem is used. In the complex case, a sorting procedure would be required as shown in [23].

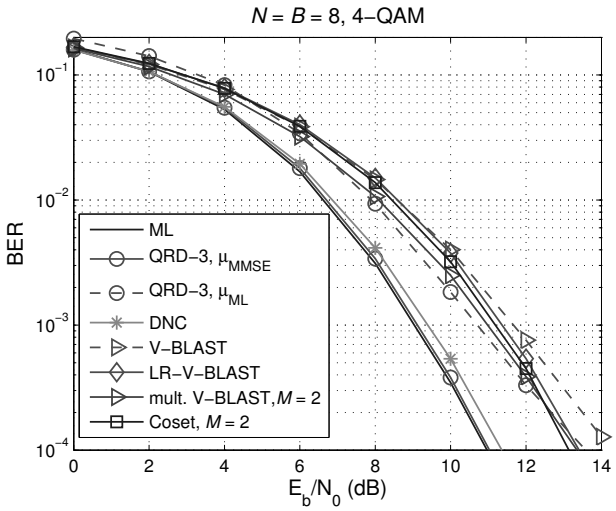

Fig. 4. BER performance of sub-optimal detection schems in an 8×8 system with 4-QAM modulation as a function of the SNR per bit.

7. RESULTS

This section shows the bit error rate (BER) performance of different suboptimal MIMO detection schemes as a function of the SNR per bit, defined as

$$\frac{E_\mathrm{b}}{N_0}=\frac{B}{\log_2(|\mathbb{A}|)\,\mathrm{tr}(\boldsymbol{C_\eta})}$$

where $\mathrm{tr}(\cdot)$ denotes the trace operator. In addition, results of the FPGA implementation of the QRD-4 algorithm are included, comparing them to those of previously implemented tree-search-based MIMO detectors. Unless otherwise stated, all detectors used the real-valued representation of the system.

In Fig. 4, the BER performance of a number of detection schemes is shown in an 8×8 system with 4-QAM modulation. The MMSE metric in (4) has been applied to all the detectors since the MMSE metric is ML-optimal and the unbiased MMSE metric only leads to worse results for 4-QAM. For comparison purposes, the ML metric has also been applied to the QRD-M algorithm. We observe that all schemes outperform the MMSE V-BLAST detector, but only QRD-M, with $M=3$ and MMSE metric, and DNC come close to ML. However, DNC has cubic complexity per received vector, whereas QRD-M only has about twice the complexity of V-BLAST for $M=3$.

When switching to 16-QAM, the differences are more pronounced, as shown in Fig. 5. The unbiased MMSE metric has been used, unless otherwise stated. Following our rule of thumb, we chose $M=6$ for the QRD-M algorithm in this scenario. Again, QRD-M is the best suboptimal detector, with the unbiased MMSE metric being slightly preferable compared to the MMSE metric.

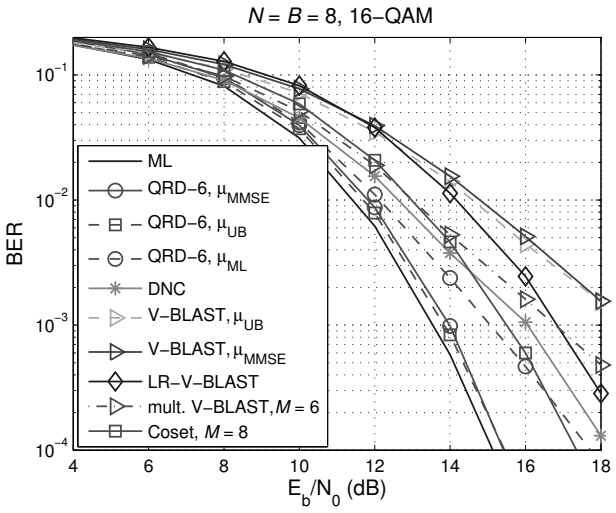

Fig. 5. BER performance of sub-optimal detection schems in an 8×8 system with 16-QAM modulation as a function of the SNR per bit.

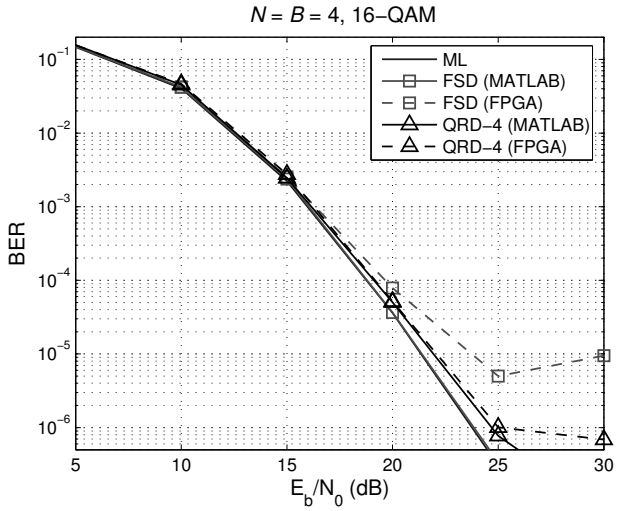

Fig. 6. BER performance of the QRD-4 and the FSD in a 4×4 system with 16-QAM modulation as a function of the SNR per bit.

Xilinx XC2VP70 FPGA	FSD [27]	QRD-4
Slices (33,088)	38% (12,721)	54% (17,917)
Flip-flops (66,176)	23% (15,332)	36% (24,403)
4-input LUTs (66,176)	24% (16,119)	29% (19,560)
Multipliers (328)	48% (160)	29% (96)
RAM blocks (328)	25% (82)	25% (84)

Table 1. FPGA resource use of the QRD-4 algorithm compared to the FSD in a 4×4 system with 16-QAM modulation.

The resource use of the FPGA implementation of the QRD-4 algorithm in a 4×4 system with 16-QAM modulation is summarized in Table 1. The result is compared to that of a previously proposed fixed-complexity SD (FSD) [27] applied to the same system. In order to establish a fair comparison, both algorithms have been implemented so that they provide the same constant throughput of 400Mbps with a clock frequency of 100MHz and a similar BER performance. Initially, it can be observed how the QRD-4 provides a 40% reduction in the number of multipliers, which is the limiting factor in the FSD. This is achieved by making use of the unbiased MMSE metric, which reduces the value of M compared to previous implementations [22], [24] with unnoticeable performance degradation. In addition, the number of multipliers has also been reduced by making use of the real decomposition of the system. Although the number of levels on the tree is effectively doubled, the use of real arithmetic more than halves the number of multipliers, resulting in an overall reduction. On the other hand, the QRD-4 makes a more intensive use of flip-flops and LUTs. This is due mainly to the sorting stages required during the algorithm, shifting the limiting factor from the multipliers to the slices[2]. However, that effect has been reduced by using 16-bit metric values instead of the 32-bit metric values used in the original FSD.

Fig. 6 shows the FPGA BER performance of the QRD-4 algorithm compared to that of the FSD in a 4×4 system with 16-QAM. The input values to both algorithms are quantized using 16 bits per real component. It can be observed for both algorithms how a difference appears between the FPGA and MATLAB BER performances at high E_B/N_0, due to the quantization process. However, although the FSD outperforms the QRD-4 in floating-point arithmetic (MATLAB), the QRD-4 results in a more robust algorithm when fixed-point arithmetic is used (FPGA). This is a consequence of the MMSE metric used in the QRD-4, as opposed to the ML metric used in the FSD. The normalization factor in the MMSE metric reduces the dynamic range of the values of the input matrices to the QRD-M algorithm, making the algorithm more robust against the quantization process.

Table 2 compares the QRD-4 implementation presented in this paper to previous implementations of tree-search-based 4×4 16-QAM MIMO detectors, in particular the K-Best lattice decoder, the SD and the FSD. First of all, the QRD-4 algorithm provides an improved performance compared to previous K-Best ASIC implementations. In order for the K-Best to achieve a similar throughput performance, a small parameter K is required [26], resulting in a non-negligible BER performance degradation. The QRD-4 overcomes that problem by using the MMSE metric presented in this paper, achieving

[2]Each slice on the FPGA contains two flip-flops and two LUTs, thus, a high percentage of the slices are only partially used. Looking at the number of flip-flops or LUTs gives a more accurate idea of the logic complexity of the algorithm given that the number of slices can be affected by the configuration of the synthesis tools.

	K-best 1 [22]	K-best 2 [24]	SD [23]	K-best 3 [26]	FSD-16 [27]	FSD-64 [32]	QRD-4
Hardware platform	ASIC	ASIC	ASIC	ASIC	FPGA	FPGA	FPGA
MIMO system	4×4	4×4	4×4	4×4	4×4	4×4	4×4
Modulation	16-QAM	16-QAM	16-QAM	16-QAM	16-QAM	64-QAM	16-QAM
Metric	l^2-norm	l^2-norm	l^1-norm	l^1-norm	l^2-norm	l^2-norm	l^2-norm
Floating-point BER	quasi-ML	quasi-ML	close to ML	error floor	quasi-ML	quasi-ML	close to ML
Clock freq. (MHz)	100	100	71	132	100 (150)	100 (150)	100 (150)
Throughput (Mbps)	10	53.3	169†	424	400 (600)	300 (450)	400 (600)

Table 2. Comparison of real-time tree-search-based MIMO detectors and the QRD-4 presented in this work. († throughput measured at $E_b/N_0 = 14$dB.)

a quasi-ML BER performance with a small parameter M, as shown in this section. It can also be seen how the QRD-4 achieves the same throughput as an FGPA implementation of the FSD for the same MIMO system. In both cases, the clock frequency and the throughput can be increased by a 50% inserting an additional pipeline stage in the multipliers causing a 10% increase in the number of flip-flops used [32]. The prototyping of the FSD for a 4×4 system using 64-QAM modulation has also been included to show how the FSD concept can be applied to high-dimensional systems, where the K-Best or the SD would have a prohibitive complexity if quasi-ML performance was to be achieved. It can be observed how the higher complexity of the FSD-64 results in a reduced throughput in order to make the implementation fit on the same FPGA board. A similar behaviour should be expected for the QRD-M algorithm, which could also be implemented for the 64-QAM case, since it has been shown to be especially suited for high-dimensional systems. An ASIC implementation of the SD is included for reference purposes [23], showing how it provides a lower throughput with the added disadvantage of being dependent on the noise level and the channel conditions. Finally, given that an FPGA-based rapid prototyping methodology has been used, we believe that the implementation of the QRD-4 on an ASIC could lead to further improvements in its performance.

8. CONCLUSION

An unbiased MMSE metric for MIMO detection has been presented in this paper, leading to a slight BER performance improvement compared to the MMSE metric. We have also identified the QRD-M algorithm as the most promising candidate for suboptimal MIMO detection, since it offers the best trade-off between complexity and performance, if the MMSE metric is used together with a real-valued representation of the system. Additionally, we have presented the FPGA implementation, using a rapid prototyping methodology, of the QRD-M algorithm based on the unbiased MMSE metric with a real-valued system model. The comparison of the QRD-M FPGA implementation to the state-of-the-art implementations of [22],[23],[25]-[27] has shown that the best fixed-point BER performance with 16-bit precision is offered by the QRD-M detector, providing the same throughput as the FSD detector in [27] but with a considerable reduction in the number of multipliers.

9. REFERENCES

[1] S. Verdú, *Multiuser Detection*, Cambridge University Press, 1998.

[2] B. Hassibi and B. M. Hochwald, "High-Rate Codes That Are Linear in Space and Time," *IEEE Transactions on Information Theory*, vol. 48, no. 7, pp. 1804–1824, July 2002.

[3] J. Jaldén and B. Ottersten, "On the Complexity of Sphere Decoding in Digital Communications," *IEEE Transactions on Signal Processing*, vol. 53, no. 4, pp. 1474–1484, April 2005.

[4] C. 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, November 1999.

[5] K. Kusume, M. Joham, W. Utschick, and G. Bauch, "Cholesky Factorization with Symmetric Permutation Applied to Detecting and Precoding Spatially Multiplexed Data Streams," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 3089–3103, June 2007.

[6] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, "Factoring Polynomials with Rational Coefficients," *Mathematische Annalen*, vol. 261, pp. 515–534, 1982.

[7] L. Babai, "On Lovász' Lattice Reduction and the Nearest Lattice Point Problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.

[8] H. Yao and G. W. Wornell, "Lattice-Reduction-Aided Detectors for MIMO Communications Systems," in

Proc. IEEE Globecom'02, November 2002, vol. 1, pp. 424–428.

[9] C. Windpassinger and R. F. H. Fischer, "Low-Complexity Near-Maximum-Likelihood Detection and Precoding for MIMO Systems using Lattice Reduction," in *Proc. IEEE ITW 2003*, March 2003, pp. 345–348.

[10] D. Wübben, R. Böhnke, and K. Kammeyer, "MMSE-Based Lattice-Reduction for Near-ML Detection of MIMO Systems," in *Proc. ITG WSA 2004*, March 2004, pp. 106–113.

[11] M. Taherzadeh, A. Mobasher, and A. K. Khandani, "LLL Lattice-Basis Reduction Achieves the Maximum Diversity in MIMO Systems," in *Proc. IEEE ISIT 2005*, September 2005, pp. 1300–1304.

[12] D. Seethaler, H. Artés, and F. Hlawatsch, "Dynamic Nulling-and-Canceling for Efficient Near-ML Decoding of MIMO Systems," *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4741–4752, December 2006.

[13] M. Joham, D. A. Schmidt, H. Brunner, and W. Utschick, "Symbol-Wise Order Optimization for THP," in *Proc. ITG/IEEE WSA 2007*, February 2007.

[14] K. Su and F. R. Kschischang, "Coset-based lattice detection for MIMO systems," in *Proc. IEEE ISIT 2007*, June 2007.

[15] J. Jaldén and B. Ottersten, "High Diversity Detection Using Semidefinite Relaxation," in *Proc. 40th Asilomar Conference on Signals, Systems and Computers*, October 2006, pp. 2082–2086.

[16] A. D. Murugan, H. El Gamal, M. O. Damen, and G. Caire, "A Unified Framework for Tree Search Decoding: Rediscovering the Sequential Decoder," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 933–953, March 2006.

[17] M. O. Damen, H. El Gamal, and G. Caire, "On Maximum-Likelihood Detection and the Search for the Closest Lattice Point," *IEEE Transactions on Signal Processing*, vol. 49, no. 10, pp. 2389–2402, October 2003.

[18] J. B. Anderson and S. Mohan, "Sequential Coding Algorithms: A Survey and Cost Analysis," *IEEE Transactions on Communications*, vol. 32, no. 2, pp. 169–176, February 1984.

[19] L. Wei and C. Schlegel, "Synchronous DS-SSMA System with Improved Decorrelating Decision-Feedback Multiuser Detection," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 3, pp. 767–772, August 1994.

[20] K. J. Kim and R. A. Iltis, "Joint Detection and Channel Estimation Algorithms for QS-CDMA Signals Over Time-Varying Channels," *IEEE Transactions on Communications*, vol. 50, no. 5, pp. 845–855, May 2002.

[21] Y. Guo and D. McCain, "Reduced QRD-M Detector in MIMO-OFDM Systems With Partial and Embedded Sorting," in *Proc. IEEE Globecom 2005*, December 2005.

[22] K. Wong, C. Tsui, R. S. Cheng, and W. Mow, "A VLSI Architecture of a K-Best Lattice Decoding Algorithm for MIMO Channels," in *Proc. IEEE ISCAS 2002*, May 2002, vol. 3, pp. III–273–III–276.

[23] 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 Sold-State Circuits*, vol. 40, no. 7, pp. 1566–1576, July 2005.

[24] 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, March 2006.

[25] Q. Li and Z. Wang, "Improved K-Best Sphere Decoding Algorithms for MIMO Systems," in *Proc. IEEE ISCAS 2006*, May 2006, pp. 1159–1162.

[26] M. Wenk, M. Zellweger, A. Burg, N. Felber, and W. Fichtner, "K-Best MIMO Detection VLSI Architectures Achieving up to 424 Mbps," in *Proc. IEEE ISCAS 2006*, May 2006, pp. 1151–1154.

[27] L. G. Barbero and J. S. Thompson, "Rapid Prototyping of a Fixed-Throughput Sphere Decoder for MIMO Systems," in *Proc. IEEE ICC 2006*, June 2006, vol. 7, pp. 3082–3087.

[28] J. M. Cioffi, G. P. Dudevoir, M. V. Eyuboglu, and G. D. Forney, "MMSE Decision-Feedback Equalizers and Coding—Part I: Equalization Results," *IEEE Transactions on Communications*, vol. 43, no. 10, pp. 2582–2594, October 1995.

[29] R. F. H. Fischer and C. Windpassinger, "Real versus complex-valued equalisation in V-BLAST systems," *Electronics Letters*, vol. 39, no. 5, pp. 470–471, March 2003.

[30] A. Wiesel, X. Mestre, A. Pagés, and J. R. Fonollosa, "Efficient implementation of sphere demodulation," in *Proc. IEEE SPAWC 2003*, Rome, Italy, June 2003, vol. 1, pp. 36–40.

[31] S. G. Akl, *Parallel Sorting Algorithms*, Academic Press, Inc., 1985.

[32] L. G. Barbero and J. S. Thompson, "FPGA design considerations in the implementation of a fixed-throughput sphere decoder for MIMO systems," in *Proc. IEEE FPL 2006*, Madrid, Spain, Aug. 2006.