Fixed-Point Aspects of MIMO OFDM Detection on SDR Platforms

D. Guenther, T. Kempf, G. Ascheid
Institute for Communication Technologies and Embedded Systems, RWTH Aachen University, Germany
guenther@ice.rwth-aachen.de

Abstract - In IEEE 802.11n compliant receivers, MIMO detection causes a major part of the computational complexity. Various publications exist on the issue of ASIC design for MIMO detection. However, the increasing variety of mobile communication standards calls for more flexible platforms, implementing the different standards in software, hence called Software Defined Radios (SDRs). This work focuses on achieving quality of service close to floating point performance while using the limitted fixed-point precision typically available on SDR platforms. A suitable algorithm for QR decomposition of the channel matrices for real time processing is derived, and the consecutive spatial equalizing as well as the SINR calculation are presented. A software implementation on the maturing P2012 platform by ST Microelectronics is benchmarked with respect to timing- and error correction performance.

I. Introduction

With the increasing popularity of mobile communication, the amount of mobile communication standards is increasing equally. Furthermore, modern portable communication devices are supposed to support cellular networks like LTE [1] and wireless LANs, e.g. IEEE 802.11n [2]. Instead of creating an application-specific integrated circuit (ASIC) for each communication standard, the Software Defined Radio (SDR) approach implements each standard in software on a flexible, programmable platform instead. Simultaneously, the demand for higher data rates is answered by multiple-input and multiple-output (MIMO) systems with more than one antenna at the transmitter- and receiver side to benefit from spacial diversity. In LTE as well as in IEEE 802.11n, MIMO comes along with orthogonal frequency-division multiplexing (OFDM) as modulation technique, which minimizes the impact of frequency selective fading.

As discussed in [3], MIMO processing requires vector analysis. Hence single instruction multiple data (SIMD) capabilities are of central importance for an SDR platform. However, programmable platforms come along with fixed bitwidths for integer datatypes (typically 8, 16 and 32 bit), while in an ASIC solution, the internal bitwidth can be chosen freely. In [4], it has been shown how the QR decomposition of the MIMO channel matrices can be numerically stabilized for a limitted bitwidth. An approach which will be adopted and extended here.

In this work, the maturing P2012 platform [5] by ST Microelectronics is used as a reference. The P2012 is composed of several clusters of RISC cores. Each cluster comprises a maximum of 16 cores which can be tailored by application specific extensions like the VECx extension, offering single-instruction multiple-data (SIMD) instructions for vector analysis. Figure 1 gives an overview of the P2012 platform. The proposed detector along with all other parts of the inner modem is implemented on this platform.

The remainder of this paper is organized as follows: Section II introduces the system model and the structure of an adequate MIMO OFDM receiver. Section III presents efficient implementations of the QR decomposition of the channel matrices, focussing on how to achieve good fixed point stability with the limited available bitwidth. Moreover, the calculation of the signal to interference and noise ratio (SINR), which operates on the same data, is explained. Section IV discusses the execution time of different algorithms, while Section V elaborates on the resulting error correction performance. Section VI concludes the paper.

II. System Model

OFDM divides the available bandwidth into a number of equally spaced subchannels (subcarriers). Hence, one may treat each subchannel separately as a frequency flat channel. For a system with N_t transmitter antennas and N_r receiver antennas, each subchannel is modeled by the transmission equation:

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

Whereat $\mathbf{x}$ is the transmitted symbol vector of dimension N_t, $\mathbf{y}$ is the received symbol vector of dimension N_r, $\mathbf{H}$ is the channel matrix for the current subchannel of dimension $N_r \times N_t$ and $\mathbf{n}$ is the noise vector of dimension N_r.

The basic structure of a MIMO OFDM receiver is shown in Figure 2. The receiver is divided into an inner modem and an outer modem. The inner modem works on the complex baseband representation of the received data, while the outer modem gets a bitwise representation as input. Within the inner modem, OFDM processing comprises several steps: Firstly removing the cyclic prefix, which is inserted upfront every OFDM symbol to protect against inter-symbol-interference (ISI) and inter-carrier-interference (ICI). Secondly and mainly,

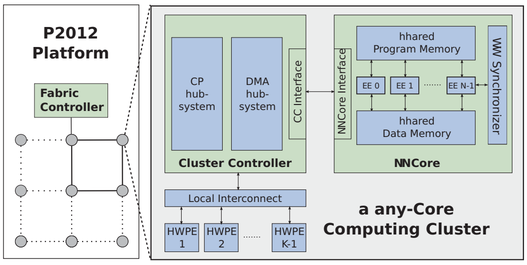

Fig. 1. P2012 platform [5]

This work has been supported by the UMIC Research Centre, RWTH Aachen University and by the EC under grant 2PARMA FP7-248716.

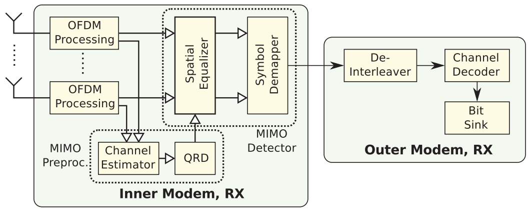

Fig. 2. MIMO OFDM receiver overview

it performs an OFDM demodulation, which is an FFT, to separate the OFDM subcarriers. Thirdly, it conducts a subcarrier demapping, since certain guard carriers at the channel borders are left empty to mitigate inter-channel interference. The channel estimator derives an estimate of the channel matrix $\mathbf{H}$, to which a QR decomposition (QRD) is applied. The combination of channel estimation and QRD is also referred to as MIMO preprocessing. The QRD of $\mathbf{H}$ is used by the spatial equalizer to mitigate the impact of the channel on the payload data. Soft symbol demapping finally converts the complex baseband representation of the received data back to a bitwise representation. Equalizing and soft demapping together are also referred to as MIMO detection, whose output is then fed to the deinterleaver and the channel decoder.

MIMO equalizing, which is the main focus of this work, derives an estimate $\hat{\mathbf{x}}$ for the originally transmitted symbol vector $\mathbf{x}$ by using $\mathbf{y}$ and the estimated channel matrix $\hat{\mathbf{H}}$. Two common approaches, which will be discussed in the following are **linear minimum mean square error** (LMMSE) equalizing and **successive interference cancellation** (SIC).

A. LMMSE Equalizing

LMMSE minimizes the expected value of the square error of the estimated and transmitted symbol vector by multiplying $\mathbf{y}$ by an equalizer matrix $\mathbf{G}$ of dimension $N_t \times N_r$.

$$\begin{aligned} \hat{\mathbf{x}} &= \mathbf{G}\mathbf{y} \\ \arg\min_{\mathbf{G}} \mathrm{E}\left\{ \left|\mathbf{x} - \hat{\mathbf{x}}\right|^2 \right\} &= \arg\min_{\mathbf{G}} \mathrm{E}\left\{ \left|\mathbf{x} - \mathbf{G}\mathbf{y}\right|^2 \right\} \end{aligned} \quad (2)$$

Assuming uncorrelated additive white Gaussian noise (AWGN) results in:

$$\mathrm{E}\left\{\mathbf{n}^H \mathbf{n}\right\} = N_0 \cdot \mathbf{I} \quad (3)$$

With spectral noise density N_0 and $\mathbf{I}$ representing an identity matrix. Using this relationship in (2) delivers the LMMSE equalizer matrix:

$$\mathbf{G} = \left(\hat{\mathbf{H}}^H \hat{\mathbf{H}} + N_0 \mathbf{I}\right)^{-1} \hat{\mathbf{H}}^H \quad (4)$$

Using the regularized channel matrix $\bar{\mathbf{H}}$ of dimension $(N_r + N_t) \times N_t$,

$$\bar{\mathbf{H}} = \begin{pmatrix} \hat{\mathbf{H}} \\ \sqrt{N_0}\mathbf{I} \end{pmatrix} \quad (5)$$

(3) can be rewritten as:

$$\mathbf{G} = \left(\bar{\mathbf{H}}^H \bar{\mathbf{H}}\right)^{-1} \hat{\mathbf{H}}^H \quad (6)$$

However, (6) still contains a matrix inversion, which is not suitable for implementation on platforms with limited fixed point precision. Therefore, $\bar{\mathbf{H}}$ is decomposed into the product of $\mathbf{Q}$ and $\mathbf{R}$, so that:

$$\mathbf{Q}\mathbf{R} = \begin{pmatrix} \mathbf{Q_a} \\ \mathbf{Q_b} \end{pmatrix} \mathbf{R} = \bar{\mathbf{H}} \quad (7)$$

where the matrix $\mathbf{Q_a}$ is of dimension $N_r \times N_t$ and $\mathbf{Q_b}$ is of dimension $N_t \times N_t$.

$$\mathbf{Q}^H \mathbf{Q} = \mathbf{I} \quad \wedge \quad \mathbf{Q_b} = \sqrt{N_0}\mathbf{R}^{-1} \quad (8)$$

Now, (6) can be rewritten as an inversion-free equation:

$$\mathbf{G} = \mathbf{R}^{-1} \mathbf{Q_a}^H = \frac{\mathbf{Q_b}}{\sqrt{N_0}} \mathbf{Q_a}^H \quad (9)$$

B. MMSE-SIC Equalizing

In contrast to linear equalizing, as presented above, which derives all elements of the symbol vector estimate at once, MMSE-SIC equalizing calculates the estimate componentwise. For that purpose, (2) is reformulated:

$$\mathbf{Q_a}^H \mathbf{y} = \mathbf{R}\hat{\mathbf{x}} \quad (10)$$

Due to the upper triangular structure of $\mathbf{R}$, the equation can be successively solved for the elements of $\hat{\mathbf{x}}$, starting from the element with the highest index. Before re-using an already determined element to derive another, it is quantized (sliced) to the closest constellation symbol. This technique improves detection quality, given the fact that elements are sliced to the correct constellation symbol. Otherwise, an error propagation is caused.

$$\hat{x}_i = \frac{\tilde{y}_i - \sum_{j=i+1}^{N_t} r_{ij}\,\mathrm{Q}\left[\hat{x}_j\right]}{r_{ii}} \qquad i = N_r \;..\; 1 \quad (11)$$

To circumvent error propagation, an ordered SIC may be used. Ordering is performed in a way that the symbol with the highest SINR is detected first. In [6], it is shown that this corresponds to reordering the QRD by a multiplication by a permutation matrix $\mathbf{P}$ so that $r_{ii} < r_{jj}$ for $i < j$.

$$\bar{\mathbf{H}} = \mathbf{Q}\mathbf{R}\mathbf{P}^T \quad (12)$$

This means instead of (10), one has to successively solve:

$$\mathbf{Q_a}^H \mathbf{y} = \mathbf{R}\mathbf{P}^T \hat{\mathbf{x}} \quad (13)$$

Based on the above given equations, LMMSE and MMSE-SIC equalizing are divided into a **preprocessing** and an **actual equalizing** phase. Preprocessing comprises the QRD and for LMMSE also the calculation of $\mathbf{G}$. Note that preprocessing is performed for every subcarrier and is independent of the received data payload. Actual equalizing then uses the results from preprocessing to calculate $\hat{\mathbf{x}}$. For LMMSE, this phase solely consists of a number of matrix vector multiplications, while for SIC-MMSE, a pre-multiplication of the received symbol vectors by $\mathbf{Q_a}^H$ has to be performed. Then, finally, the estimated symbol vectors are demapped to a soft, bitwise representation.

III. QRD Investigation

Different approaches exist for performing a QR decomposition. Two methods suitable for a hardware friendly implementation are **Modified Gram-Schmidt** (MGS) and **Givens Rotation** (GR).

A. *Modified Gram-Schmidt*

The MGS algorithm operates column-wise on the regularized channel matrix $\bar{\mathbf{H}}$. It starts from the leftmost column which is first normalized and then projected on the right-hand columns. The result of the projection is subtracted from these columns. Then, the process starts again from the next, righthand column vector. Thereby, linear dependencies are removed. However, the repeated subtraction of projections may cause the righter column vectors to become too big or small for an accurate fixed-point implementation. For that reason, a technique called dynamic scaling (DS) is introduced. During the QR decomposition, DS performs bitwise left- and right shifts on the column vectors to keep them within fixed-point range. Similar to [4], this work uses a MGS with DS, which is described by the pseudo-code given in Algorithm 1.

DS is performed in lines 3 to 9, while the remainder of the algorithm is the ordinary MGS algorithm. Note that the algorithm does not deliver $\mathbf{R}$. Instead, one gets $\mathbf{Q_b} = \sqrt{N_0}\mathbf{R}^{-1}$ which can be used in (9) to derive $\mathbf{G}$ directly.

For high SNR regions, like the SNR-target-region for 64-QAM modulation, this algorithm may become critical, though. Due to limited fixed-point precisions, the lower scaled identity matrix of $\bar{\mathbf{H}}$ becomes too small for an accurate fixed-point representation. Thus, $\mathbf{Q_b}$ and the equalizer matrix $\mathbf{G}$ will degrade equally, compromising the entire detection. For this reason, this work uses the unity-regularized channel matrix (URCM) $\bar{\mathbf{H}}_u$ instead of $\bar{\mathbf{H}}$.

$$\bar{\mathbf{H}}_u = \begin{pmatrix} \mathbf{H} \\ \mathbf{I} \end{pmatrix} \tag{14}$$

Naturally, this affects the normalization and the projection in lines 10 and 12. However, the square norms of the initial $\bar{\mathbf{H}}$ matrix can be calculated from $\mathbf{H}$ and N_0. Subsequently these norms can be updated as presented in [7]. For the projection, though, the last N_t entries of the column vectors $\mathbf{v_j}$ have to be scaled down by a factor of $\sqrt{N_0}$. These considerations lead to Algorithm 2.

Algorithm 1 MMSE MGS-QRD with DS

```
1:  $\mathbf{V} \leftarrow \bar{\mathbf{H}}$
2:  for $i = 1$ to $N_t$ do
3:    for $j = i$ to $N_t$ do
4:      if $\max\{|\Re\{v_{j,1}\}|, |\Im\{v_{j,1}\}|, ..\} < B_l$ then
5:        $\mathbf{v_j} \leftarrow 2\mathbf{v_j}$
6:      else if $\max\{|\Re\{v_{j,1}\}|, |\Im\{v_{j,1}\}|, ..\} > B_h$ then
7:        $\mathbf{v_j} \leftarrow \mathbf{v_j}/2$
8:      end if
9:    end for
10:   $\mathbf{v_i} \leftarrow \mathbf{v_i}/\|\mathbf{v_i}\|$
11:   for $j = i+1$ to $N_t$ do
12:     $\mathbf{v_j} \leftarrow \mathbf{v_j} - \left(\mathbf{v_i}^H \mathbf{v_j}\right)\mathbf{v_i}$
13:   end for
14: end for
15: $\mathbf{Q} \leftarrow [\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_{N_t}}]$
```

Algorithm 2 MMSE MGS-QRD with DS and URCM

```
$\mathbf{V} \leftarrow \bar{\mathbf{H}}_\mathbf{u}$
for $i = 1$ to $N_t$ do
  $\xi_i = \left(\mathbf{H}^H\mathbf{H}\right)_{i,i} + N_0$
end for
for $i = 1$ to $N_t$ do
  for $j = i$ to $N_t$ do
    if $\max\{|\Re\{v_{j,1}\}|, |\Im\{v_{j,1}\}|, ..\} < B_l$ then
      $\mathbf{v_j} \leftarrow 2\mathbf{v_j}$
      $\xi_j \leftarrow 4 \cdot \xi_j$
    else if $\max\{|\Re\{v_{j,1}\}|, |\Im\{v_{j,1}\}|, ..\} > B_h$ then
      $\mathbf{v_j} \leftarrow \mathbf{v_j}/2$
      $\xi_j \leftarrow 1/4 \cdot \xi_j$
    end if
  end for
  $\mathbf{v_i} \leftarrow \mathbf{v_i}/|\sqrt{\xi_i}|$
  for $j = i+1$ to $N_t$ do
    $s = \left(\mathbf{v_i}^H \odot \mathbf{a}^T\right)\left(\mathbf{v_j} \odot \mathbf{a}\right)$
    $\mathbf{v_j} \leftarrow \mathbf{v_j} - s\mathbf{v_i}$
    $\xi_j \leftarrow \xi_j - |s|^2$
  end for
end for
$\mathbf{Q} \leftarrow [\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_{N_t}}]$
```

Here, ξ_i is the vector norm of column vector $\mathbf{v}_i$ and $\mathbf{a}$ is a real valued scaling vector that scales the last N_t entries down by a factor of $\sqrt{N_0}$ while leaving the prior ones unchanged. This component wise multiplication is denoted by the $\odot$-operator.

While the $\mathbf{R}$-matrix is not required for linear equalizing, it is necessary for non-linear algorithms like SIC. As mentioned in [4], the matrix is lost when applying DS, but it can be restored by keeping track of the bitshifts e_i formerly performed on the column vector i of $\bar{\mathbf{H}}_\mathbf{u}$. Combined with column sorting as proposed in [8], this leads to Algorithm 3. While the sorted MGS algorithm proposed in [7] uses the column vector norm to derive the column scaling factors, the algorithm presented here uses the absolute values in combination with DS. Hence, the resulting algorithm is computationally less complex for an SDR application.

B. *Inverse Square Root Calculation*

Normalization of the column vectors as required by the MGS algorithm is a computationally complex tasks, since it inhibits a square root calculation and a division. Observing Algorithms 2 and 3, one sees that but the inverse of r_{ii} is used in the QRD as well as during spatial equalizing. For that reason, it makes sense to store $(1/r_{ii})$ instead of r_{ii} in the diagonal entries of the $\mathbf{R}$-matrix.

As proposed in [9], the inverse square root calculation can be efficiently approximated by Newton's Method (NM). The

Algorithm 3 MMSE MGS-SQRD with DS and URCM

```
$\mathbf{V} \leftarrow \bar{\mathbf{H}}_\mathbf{u}$
$\mathbf{P} \leftarrow \mathbf{I}_{N_t}$
for $i = 1$ to $N_t$ do
    $\xi_i = \left(\mathbf{H}^H\mathbf{H}\right)_{i,i} + N_0$
    $e_i = 0$
end for
for $i = 1$ to $N_t$ do
    $k = \mathrm{argmin}_{j=i,..,N_t}\left(\xi_j\right)$
    exchange columns $i$ and $k$ in $\mathbf{V}$, $\mathbf{R}$ and $\mathbf{P}$
    exchange elements $i$ and $k$ in $\xi$ and $\mathbf{e}$
    for $j = i$ to $N_t$ do
        if $\max\{|\Re\{v_{j,1}\}|, |\Im\{v_{j,1}\}|, ..\} < B_l$ then
            $\mathbf{v_j} \leftarrow 2\mathbf{v_j}$
            $e_j \leftarrow e_j + 1$
            $\xi_j \leftarrow 4 \cdot \xi_j$
        else if $\max\{|\Re\{v_{j,1}\}|, |\Im\{v_{j,1}\}|, ..\} > B_h$ then
            $\mathbf{v_j} \leftarrow \mathbf{v_j}/2$
            $e_j \leftarrow e_j - 1$
            $\xi_j \leftarrow 1/4 \cdot \xi_j$
        end if
    end for
    $r_{ii} = 1/\sqrt{\xi_i} \cdot 2^{+e_i}$
    $\mathbf{v_i} \leftarrow \mathbf{v_i}/\sqrt{\xi_i}$
    for $j = i + 1$ to $N_t$ do
        $s = \left(\mathbf{v_i}^H \odot \mathbf{a}^T\right)\left(\mathbf{v_j} \odot \mathbf{a}\right)$
        $r_{ij} = s \cdot 2^{-e_j}$
        $\mathbf{v_j} \leftarrow \mathbf{v_j} - s\mathbf{v_i}$
        $\xi_j \leftarrow \xi_j - |s|^2$
    end for
end for
$\mathbf{Q} \leftarrow [\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_{N_t}}]$
```

general form of NM for iteratively deriving the zeros of a function f is given by:

$$y_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)} \tag{15}$$

To solve $y = 1/\sqrt{x}$, the function whose zeros one has to derive is $f(y) = 1/y^2 - x$, so the iteration is given by:

$$y_{n+1} = y_n \frac{3 - xy_n^2}{2} \tag{16}$$

Requiring no divisions, this term is very suitable for an SDR implementation. Additionally, the application of DS is limitting the dynamic range of the input values, so in combination with a start value selection from a few predefined values, good results can be achieved with less than 5 iterations.

C. Givens Rotation

In contrast to MGS, which is based on vector-vector projections and subtractions, the GR algorithm rotates the row vectors of the channel matrix to derive a set of orthogonal basis vectors. Consequently, the vector norms remain unchanged, so numerical stabilization like DS is not required here. The starting point for the MMSE approach is a composite matrix:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{H} & \mathbf{I}_{N_r} \\ \sqrt{\frac{N_0}{E_s}}\mathbf{I}_{N_t} & 0 \end{bmatrix} \tag{17}$$

The matrix is processed from the bottom upwards, combining two adjacent rows in a rotation operation. Such a rotation is performed in two steps. First, real and imaginary part of the upper row vector are turned so that the leftmost, non-zero element of the vector becomes real. Then, lower and upper vectors are rotated so that the leftmost, non-zero element of the lower vector becomes zero. These rotation are repeated from top to bottom until the submatrix $\mathbf{H}$ has been turned into a triangular matrix. After N processing steps, the matrix $\mathbf{Z}^{(N)}$ now has the following shape:

$$\mathbf{Z}^{(N)} = \begin{bmatrix} \mathbf{R} & \mathbf{Q}_a^H \\ 0 & \mathbf{Q}_c^H \end{bmatrix} \tag{18}$$

The channel matrix can be written as the product:

$$\mathbf{H} = \mathbf{Q}_a\mathbf{R} \tag{19}$$

If sorting as described in Alg.3 is used, the equation includes a the permutation matrix $\mathbf{P}$ and changes according to:

$$\mathbf{H} = \mathbf{Q}_a\mathbf{R}\mathbf{P}^T \tag{20}$$

The algorithm for the sorted variant as also described in [7] is given in Algorithm 4. Each Givens rotation is represented by a multiplication with a rotation matrix $\mathbf{\Theta}$. A matrix $\mathbf{\Theta}_{(p,q,\theta)}$, which rotates rows p and q by an angle of θ contains trigonometric functions. These functions are often not available on SDR platforms, however they can be approximated by several iteration of the so called CORDIC algorithm. A description

Algorithm 4 MMSE GR-SQRD

```
$\mathbf{Z} \leftarrow \mathbf{Z}^{(0)}$
$\mathbf{P} \leftarrow \mathbf{I}_{N_t}$
for $i = 1$ to $N_t$ do
    $\xi_i = \left(\mathbf{H}^H\mathbf{H}\right)_{i,i} + N_0$
end for
for $i = 1$ to $N_t$ do
    $k = \mathrm{argmin}_{j=i,..,N_t}\left(\xi_j\right)$
    exchange columns $i$ and $k$ in $\mathbf{P}$ and the first $N_r + i - 1$
    rows of $\mathbf{Z}$
    compute a series of Givens rotations $\Theta_u$ so that the
    elements $\mathbf{Z}_{(i+1,i)}$ until $\mathbf{Z}_{(i+N_r,i)}$ become zero.
    $\mathbf{Z} \leftarrow \left(\prod_{u=(i-1)N_r}^{iN_r} \mathbf{\Theta}_u\right)\mathbf{Z}$
    for $j = i + 1$ to $N_t$ do
        $\xi_j \leftarrow \xi_j - |z_{ij}|^2$
    end for
end for
$\mathbf{R} = \mathbf{Z}_{(1..N_t,1..N_t)}$
$\mathbf{Q}_a^H = \mathbf{Z}_{(1..N_t,N_t..N_t+N_r)}$
```

of this algorithm is beyond the scope of this work, but the interested reader is referred to [7].

D. SINR Calculation

Soft symbol demapping of the equalized symbol vectors is commonly performed using the max-log approximation according to the below given equation. The index k denotes the stream, from which a symbol is to be demapped. $L(b_{k,i})$ is the LLR value for the bit at position i within the constellation symbol from stream k, and A_i^0 and A_i^1 are the subsets of constellation symbols with a zero or one bit at position i respectively.

$$L(b_{k,i}) \approx \rho_k \left(\min_{s \in A_i^0} |z_k - s|^2 - \min_{s \in A_i^1} |z_k - s|^2 \right) \tag{21}$$

The signal to interference and noise ratio (SINR) of stream k is given by ρ_k [10] and can be calculated according to:

$$\rho_k \approx \frac{1}{\frac{\sigma_n^2}{E_s} \left[\left(\mathbf{H}^H \mathbf{H} + \frac{\sigma_n^2}{E_s} \mathbf{I}_{M_r} \right)^{-1} \right]_{k,k}} \tag{22}$$

Using the results of the regularized QR decomposition, the above equation can be reformulated to:

$$\rho_k \approx \frac{1}{\frac{\sigma_n^2}{E_s} \left[\mathbf{R}^{-1} \left(\mathbf{R}^{-1} \right)^H \right]_{k,k}} \tag{23}$$

This means, even though strictly speaking, demapping is not a part of spatial equalizing, it uses the results of the latter. Also note that in combination with regularized MGS-QRD and URCM, $\mathbf{R}^{-1}$ is directly available from the lower part of the $\mathbf{Q}$ matrix. As a consequence, it makes sense to include SINR calculation into MIMO preprocessing, as it will be done in the following.

IV. Execution Time

A fast execution time is crucial to keep the real time constraints of IEEE 802.11n. Even though more advanced detection algorithms like Sphere Decoding [11] are available, the limits of current SDR platforms require the application of more straight forward detectors. Table I lists the execution time of several suitable MIMO preprocessing and spatial equalizing algorithms for a 2×2 and 4×4 antenna configuration on a single xp70 core equipped with a VECx SIMD extension. Times for preprocessing are accumulated for the entire frame. Times for spatial equalizing contain the processing of one OFDM slot of $4 \mu s$ duration.

Also note, that for linear equalizing, the QR decomposition has to be followed by a matrix-matrix multiplication, to derive the equalizer matrix $\mathbf{G}$. Naturally, this step is not required for SIC preprocessing, but in the non-linear equalizing, every received symbol vector has to be pre-multiplied by $\mathbf{Q}_a^H$ before starting the back substitution. For linear equalizing, on the other hand, the actual equalizing solely consists of a matrix vector multiplication.

Regarding preprocessing, this table shows that dynamic scaling as well as sorted QR decomposition come at an additional cost in terms of execution time, which has to be justified by a superior algorithmic performance. One also sees a major advantage of the regularized MGS QR decomposition over the GR variant, since $\mathbf{R}^{-1}$ is directly available from the decomposition in the first case, while the triangular matrix $\mathbf{R}$ has to be inverted first in the latter case. Thus, SINR calculation can be performed faster, if MGS-QRD is used.

System	**2x2**		**4x4**	
Operation	cycles	T (μs)	cycles	T (μs)
MIMO Preprocessing (per frame)				
mgs-mmse-qrd	22,848	38.08	55,536	92.56
mgs-mmse-ds-qrd (UMCR)	35,424	59.04	66,624	111.04
mgs-mmse-ds-sqrd (UMCR)	43,248	72.08	85,392	142.32
gr-mmse-sqrd	-	-	112,032	186.72
matrix-matrix mul.	2,496	4.16	11,472	19.12
sinr-calc-r	9,456	15.76	25,824	43.04
sinr-calc-r-inv	7,248	12.08	15,600	26.00
Spatial Equalizing (per OFDM slot)				
back substitution	1,188	1.98	2,736	4.56
matrix-vector mul.	1,968	3.28	3,312	5.52

TABLE I. Single core execution time

Apart from that, the table shows that the CORDIC based GR is significantly slower than the MGS variant. Even though [12] indicates that GR is more suitable for a high throughput ASIC solution, the regular data accesses of MGS are more suitable for a SIMD implementation. Moreover the VECx vector unit is not specifically tailored to the needs of baseband processing, so it contains no special instructions to speed up the CORDIC algorithm.

Even though the multi-core aspects of the P2012 SDR application are covered in [13], two important aspects of the application, related to the above presented execution times, shall be mentioned here. Firstly one sees that the execution time for preprocessing exceeds the real time assigned to the 2×2 or 4×4 preamble. However, it is not an OFDM slot that has to be processed in real time but the entire MIMO OFDM frame. For this reason, the additional latency introduced by preprocessing can be compensated by a faster spatial equalizing. Secondly, the results in Table I show that also the actual equalizing does not achieve real time execution with a single xp70 core. For this reason, the application inherent data level parallelism must be used, as presented in [13], to distribute each task along a parallelizable dimension to enable real time execution.

V. Algorithmic Performance

To achieve real time execution, the target application uses a 16 bit fixed point format. This format causes precision problems for $\mathbf{Q_a}$ and $\mathbf{Q_b}$ in (7), which were addressed by dynamic scaling and URCM respectively. To evaluate the effect of these measures, the error correction capabilities of the resulting implementation in terms of bit-error-rate (BER) are presented in this section. For that purpose, a channel simulation featuring AWGN and i.i.d. Rayleigh Fading was set up. As in the frequently used TGn-C, a 150ns power delay spread is assumed. The power delay profile is modelled as an exponential 20dB drop. The error correction of the fixed

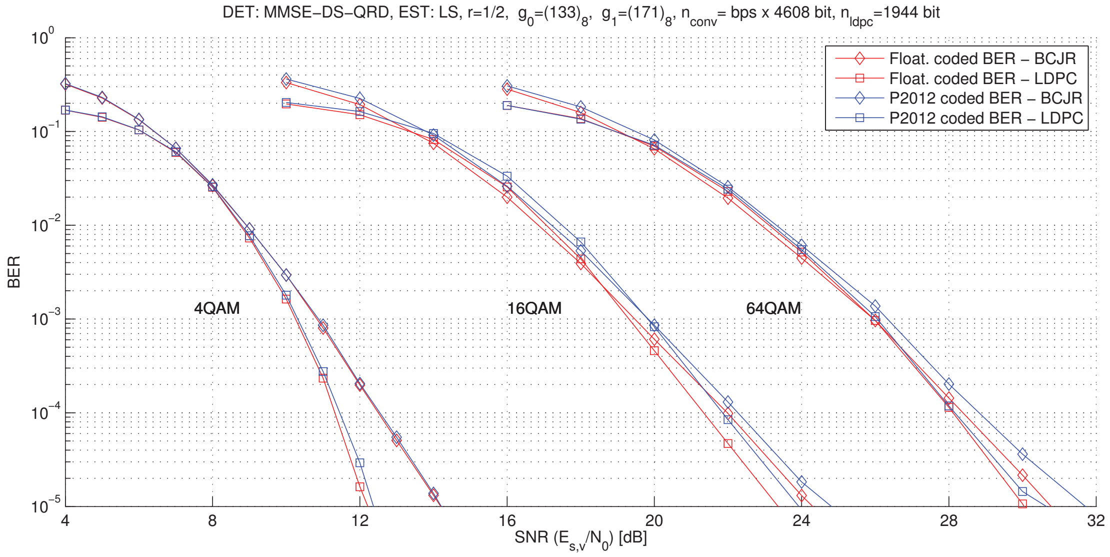

Fig. 3. Coded BER comparison of 4x4 MIMO OFDM use cases

point P2012 application is then compared to a floating point reference implementation. Note that even though this paper focuses on detection aspects, the fixed point implementation, which was benchmarked, comprises the **entire** inner modem application on the transmitter and receiver sides. Hence the resulting BERs are a realistic estimate for the actually achievable performance. Figure 3 shows the coded BER curves of the P2012 fixed-point inner modem compared to the floating point reference (UPEG), where the signal power of the SNR is given as the power per symbol of each receiver antenna. To show the capabilities of the two inner modems, they are combined with a soft-input BCJR and LDPC channel decoder. As one can see from the figure, the MGS QR decomposition extended by dynamic scaling and URCM delivers close to floating point algorithmic performance. Similar to [14], it has been observed in the scope of this work that SIC equalizing (not shown) does not outperform linear MMSE equalizing in terms of coded BER, while its computational complexity is significantly higher.

VI. Conclusion

In this paper, SDR specific aspects of non-iterative MIMO detection were discussed. The main focus was on achieving close to floating point algorithmic performance, using the limitted 16 bit precision typically available on DSP platforms. It has been seen that linear MMSE equalizing based on the Modified Gram-Schmidt QR decomposition offers the desired algorithmic performance at a reasonably low computational complexity. While the focus of this paper was on an SDR platform equipped with SIMD cores, current and future work is also investigating implementations on other types of cores like VLIW. Apart from that, it will be investigated how gains in algorithmic performance can be achieved by more advanced algorithms, while keeping the computational complexity in a region feasible for SDR solutions.

References

[1] Jim Zyren. Overview of the 3GPP Long Term Evolution Physical Layer. jul 2007.

[2] IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11, Amendment 5. *IEEE Std 802.11n-2009*, pages c1 –502, oct. 2009.

[3] Kees van Berkel, Frank Heinle, Patrick P. E. Meuwissen, Kees Moerman, and Matthias Weiss. Vector Processing as an Enabler for Software-Defined Radio in Handheld Devices. *EURASIP J. Appl. Signal Process.*, 2005:2613–2625, January 2005.

[4] Hun Seok et al. Kim. A practical, hardware friendly MMSE detector for MIMO-OFDM-based systems. *EURASIP J. Adv. Signal Process*, 2008:94:1–94:14, January 2008.

[5] D. Fuin L. Benini, E. Flamand and D. Melpignano. P2012: Building an ecosystem for a scalable, modular and high-efficiency embedded computing accelerator. In *DATE'12*, 2012.

[6] D. Wuebben, R. Bohnke, V. Kuhn, and K.-D. Kammeyer. Mmse extension of v-blast based on sorted qr decomposition. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 1, pages 508 – 512 Vol.1, oct. 2003.

[7] Peter Jan Luethi. *VLSI circuits for MIMO preprocessing*. ETH, 2009.

[8] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K.D. Kammeyer. Efficient algorithm for decoding layered space-time codes. *Electronics Letters*, 37(22):1348 –1350, oct 2001.

[9] Fast Inverse Square Root. Chris lomont. 2003.

[10] I.B. Collings, M.R.G. Butler, and M. McKay. Low Complexity Receiver Design for MIMO Bit-Interleaved Coded Modulation. In *International Symposium on Spread Spectrum Techniques and Applications, 2004.*

[11] 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. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 57(9):706 –710, sept. 2010.

[12] P. Luethi, C. Studer, S. Duetsch, E. Zgraggen, H. Kaeslin, N. Felber, and W. Fichtner. Gram-schmidt-based qr decomposition for mimo detection: Vlsi implementation and comparison. In *Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on*, pages 830 –833, 30 2008-dec. 3 2008.

[13] Torsten Kempf, Daniel Guenther, Aamir Ishaque, and Gerd Ascheid. Mimo ofdm transceiver for a many-core computing fabric - a nucleus based implementation. In *SDR'11 - The Wireless Innovation Forum Conference on Communications Technologies and Software Defined Radio*, Washington D.C., USA, dec 2011.

[14] E. Zimmermann and G. Fettweis. Adaptive vs. Hybrid Iterative MIMO Receivers Based on MMSE Linear and Soft-SIC Detection. In *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, pages 1 –5, sept. 2006.