VLSI Implementation of a Fixed-Complexity Soft-Output MIMO Detector for High-Speed Wireless

Di Wu, Johan Eilert, Rizwan Asghar and Dake Liu

Original Publication:

Di Wu, Johan Eilert, Rizwan Asghar and Dake Liu, VLSI Implementation of a Fixed-Complexity Soft-Output MIMO Detector for High-Speed Wireless, 2010, EURASIP Journal on Wireless Communications and Networking, (2010), 893184.
http://dx.doi.org/10.1155/2010/893184

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 893184, 13 pages
doi:10.1155/2010/893184

Research Article

VLSI Implementation of a Fixed-Complexity Soft-Output MIMO Detector for High-Speed Wireless

Di Wu (EURASIP Member),[1,2] Johan Eilert,[1,2] Rizwan Asghar,[1] and Dake Liu[1]

[1] *Department of Electrical Engineering, Linköping University, 58183 Linköping, Sweden*
[2] *ST-Ericsson AT AB, Lund, Sweden*

Correspondence should be addressed to Di Wu, diwu@isy.liu.se

Received 30 September 2009; Revised 17 May 2010; Accepted 23 June 2010

Academic Editor: Taşkin Kocak

This paper presents a low-complexity MIMO symbol detector with close-Maximum a posteriori performance for the emerging multiantenna enhanced high-speed wireless communications. The VLSI implementation is based on a novel MIMO detection algorithm called Modified Fixed-Complexity Soft-Output (MFCSO) detection, which achieves a good trade-off between performance and implementation cost compared to the referenced prior art. By including a microcode-controlled channel preprocessing unit and a pipelined detection unit, it is flexible enough to cover several different standards and transmission schemes. The flexibility allows adaptive detection to minimize power consumption without degradation in throughput. The VLSI implementation of the detector is presented to show that real-time MIMO symbol detection of 20 MHz bandwidth 3GPP LTE and 10 MHz WiMAX downlink physical channel is achievable at reasonable silicon cost.

1. Introduction

Multi-antenna or multi-in and multiout (MIMO) technologies have been widely adopted by the latest wireless standards such as 3GPP LTE and WiMAX to enhance the spectrum efficiency. For MIMO systems, a major challenge is the symbol detection at the receiver. In particular, as channel coding (e.g., Turbo) is used, soft output (the log-likelihood ratio, LLR) must be computed as the input to the channel decoder. Consider a MIMO system with n_{TX} transmit antennas and n_{RX} receive antennas. Let s be a transmitted vector of length n_{TX}, obtained by mapping a set of information bits onto an M-QAM constellation $\mathcal{L}$. Then the received vector of length n_{RX} is given by

$$r = \mathbf{H}s + n, \tag{1}$$

where $\mathbf{H}$ is an $n_{\mathrm{RX}} \times n_{\mathrm{TX}}$ complex-valued channel matrix which is assumed to be known. s is the transmitted symbol vector. n is noise vector and r is the received symbol vector. The optimum soft detector is Maximum-A-Posteriori (MAP) detector which computes

$$L(b_i \mid r) = \log\left(\frac{\sum_{s:b_i(s)=1} \exp\left(-1/\sigma^2 \|r - \mathbf{H}s\|^2\right)}{\sum_{s:b_i(s)=0} \exp\left(-1/\sigma^2 \|r - \mathbf{H}s\|^2\right)}\right). \tag{2}$$

Here "$s : b_i(s) = \beta$" means all s for which the ith bit of s is equal to β. Computing (2) requires enumeration of the entire set of possible transmitted vectors. The complexity of doing this is usually not affordable in practice.

As a trade-off between performance and complexity, various MIMO detection methods such as sphere decoding [1, 2], fixed complexity sphere decoding [3, 4], and MFCSO decoding [5] have been proposed to reach near-MAP performance with lower complexity than MAP. In [6], VLSI implementation of a complexity reduced K-best detector for 2×2 MIMO and 16-QAM is presented for WiMAX/WiFi. In [7], VLSI implementation of a soft-output MIMO detector for 2×2 MIMO in WLAN is presented. Without QR decomposition unit being included, it consumes 135 kGate with a reduced candidate list. In [8], a K-best detector for

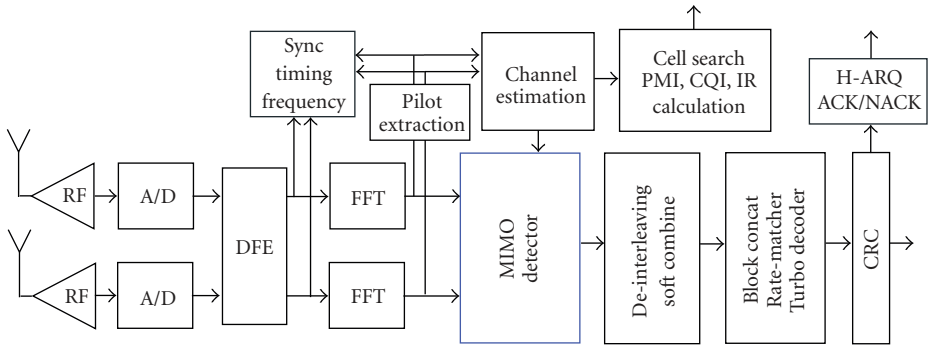
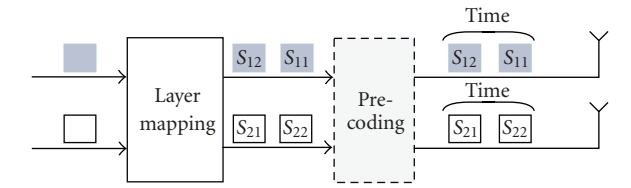
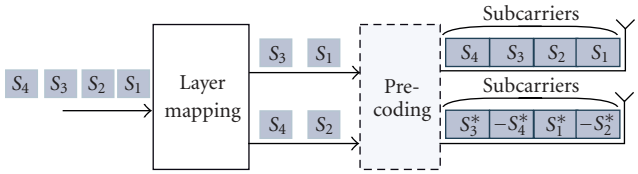


Figure 1: Functional flow of a 3GPP LTE/WiMAX receiver.



(a) Spatial multiplexing (SM), $n_{TX} = 2$.



(b) Space-frequency block coding (SFBC), $n_{TX} = 2$.

Figure 2: Downlink multi-antenna transmission schemes.

4×4 MIMO is implemented in a Xilinx Virtex-5 FPGA. However, the complexity of sphere decoding grows exponentially with the number of transmit antennas and polynomially in the size of the signal constellation. More importantly, the tree search used in sphere decoding is in principle a sequential procedure which is difficult to parallelize. In [3], a fixed-throughput sphere detector is proposed with fixed complexity and parallelism for hard decision. In [5], a low-complexity near-MAP detection method is proposed for high-order modulation (e.g., 64-QAM). The performance loss from MAP due to the suboptimal search introduced in MFCSO is proven by simulation to be small in [5]. However, in [5], the complexity of MFCSO is only presented in number of arithmetic operations without the silicon cost and processing latency being addressed and no comparison with prior art is made. Most importantly, none of these methods proposed have taken the system specific features of LTE (e.g., OFDMA and H-ARQ) into consideration and are mostly based on very simple channel models (e.g., AWGN). In [9], limited evaluation of MFCSO is carried out with a focus on LTE system.

In this paper, with the aid of more realistic LTE and WiMAX simulation chains and different channel models, several MIMO detection algorithms are applied to LTE and WiMAX systems and with their performance quantitatively evaluated. Second, although the MFCSO detection algorithm proposed by the authors in [5] has a very low detection complexity, under random AWGN channels, it requires relatively strong channel coding to maintain a near-MAP performance in frame error ratio [5]. In this paper, its performance with the aid of H-ARQ is investigated. In order to validate MFCSO from VLSI implementation perspectives, both FPGA and ASIC implementation of an MFCSO detector is presented. Note that most commercial terminals are limited by cost and power consumption, especially the power consumption of the analog part of each antenna chain. According to the LTE and WiMAX standards, 4×2 and 2×2 MIMO schemes are included as a good trade-off between performance gain and complexity (or power consumption). Hence, only these schemes are considered in here. The result is compared with a state-of-the-art soft-output sphere decoding (SSD) [1] and the K-best detector presented in [10] from both performance and cost aspects.

The remainder of the paper is organized as follows. In Section 2, the application of MIMO techniques in LTE and WiMAX is presented. Section 3 introduces the linear and MFCSO MIMO detection algorithms. Section 4 addresses the detection flow. The architecture of the detector is addressed in Section 5. The link-level simulation results are presented in Section 6. Section 7 analyzes the implementation complexity, and Section 8 presents the adaptive method used to optimize power efficiency. Section 9 presents both the FPGA-and ASIC-based implementation of the detector. Finally, Section 10 concludes the paper.

2. MultiAntenna in LTE and WiMAX

Wireless standards such as 3GPP LTE and WiMAX have incorporated MIMO transmission schemes to boost the peak data rate. Meanwhile, software-defined radio (SDR) technologies allow both of them to be supported by the same piece of hardware.

3GPP Long-Term Evolution (LTE) is the next generation radio access technology which incorporates Orthogonal

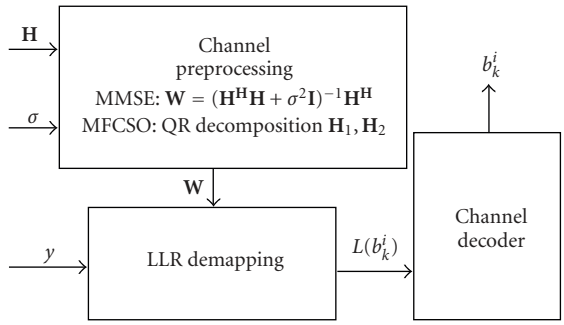


Figure 3: Task flow of soft-output MIMO detection.

Frequency Division Multiple Access (OFDMA) as the multiple access scheme in downlink. MIMO technologies are also mandatory in LTE to achieve the LTE bit-rate targets (e.g 100 Mbit/s peak data rate for downlink). As part of the receiver chain depicted in Figure 1, MIMO symbol detection is a significant challenge for VLSI implementation.

The input to the MIMO detector presented in this paper includes the estimated channel matrix

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}, \tag{3}$$

the received symbol vector r, and the estimated noise variance σ^2. The output of the detector is the LLR values of the demodulated bits.

In both LTE and WiMAX, spatial multiplexing (SM) and transmit diversity have been adopted as the two major MIMO schemes. SM is a MIMO technique aimed at maximizing the data throughput by exploiting the degrees of freedom in MIMO channels. Since the multiplexing gain is only available for high SNR region, spatial multiplexing is usually used when high SNR is available. STBC/SFBC [11] assumes the channel is stationary among adjacent time intervals or subcarriers so that a single codeword is mapped to these adjacent intervals or subcarriers to benefit from either time or frequency diversity in transmission. The most widely used STBC/SFBC scheme is Alamouti scheme in space or frequency domain. Since STBC/SFBC only requires a linear detector to achieve diversity, the detector design is easier. Note that in this paper, only open-loop MIMO is considered without feedback from the terminal.

2.1. Spatial Multiplexing. Spatial multiplexing is a MIMO technique aimed at maximizing the data throughput by exploiting the degrees of freedom in MIMO channels. Since the multiplexing gain is only available in high SNR region, spatial multiplexing is usually used when high SNR is available. As depicted in Figure 2(a), spatial multiplexing usually requires both n_{RX} and n_{TX} to be large. In general, the degree of freedom (multiplexing gain) is determined by $\min(n_{\mathrm{TX}}, n_{\mathrm{RX}})$ which is the rank of the channel matrix $\mathbf{H}$. In case $\mathbf{H}$ is badly conditioned (e.g. when line-of-sight occurs, $\mathbf{H}$ becomes a singular matrix), the pseudoinversion of $\mathbf{H}$ in (15) using linear detection will be very difficult which requires very large dynamic range. In other words, the gain of spatial multiplexing heavily depends on the multipath fading. A dual-stream spatial multiplexing scheme is depicted in Figure 2 (a).

2.2. Transmit Diversity. Transmit diversity schemes that exploit the diversity gain of multi-antenna transmission have also been adopted by LTE and WiMAX. The Space-Time Block Coding (STBC) in WiMAX and Space-Frequency Block Coding (SFBC) in LTE [11] are both transmit diversity schemes to transmit data for guaranteed diversity while requiring only a low-complexity symbol detector on the receiver side. In both cases, the Alamouti matrix [12] is used because it is the only full-rate linear STBC (or SFBC) code with a diversity gain of 2. In other words, the transmit diversity schemes considered in this paper are Alamouti schemes in the space and frequency domains. This assumes the channels of either adjacent symbol intervals or subcarriers are identical, so that either time or frequency diversity will be achieved when a single codeword is mapped to different antennas within two adjacent time or frequency intervals. The basic 4×2 space-frequency channel matrix is defined as

$$\mathbf{H} = \begin{bmatrix} h_{11} & -h_{12} \\ h_{12} & -h_{22} \\ h_{12}^* & h_{11}^* \\ h_{22}^* & h_{12}^* \end{bmatrix}. \tag{4}$$

3. Soft-Output MIMO Detection

The optimum soft-output MIMO detector computes the Log-Likelihood Ratio (LLR) in (2). Commonly the sums in (2) are approximated by their largest terms ("log-max") which requires the solution of problems of the type $\min \|r - \mathbf{H}s\|^2$, subject to $s \in \mathcal{L}$. Since MAP provides the best theoretical performance, it is commonly used as a benchmark when comparing other algorithms

$$L(b_i \mid r) \approx \log\left(\frac{\sum_{s_1 \in \mathcal{L}} \cdots \sum_{s_r \in \mathcal{L}} \max_{s_{r+1},\ldots,s_{n_{\mathrm{TX}}}} \exp\left(-1/\sigma^2 \left\| r - h_1 s_1 - \cdots - h_r s_r - \mathcal{H}(s_{r+1},\ldots,s_{n_{\mathrm{TX}}})^T \right\|^2\right)}{\sum_{s_1 \in \mathcal{L}} \cdots \sum_{s_r \in \mathcal{L}} \max_{s_{r+1},\ldots,s_{n_{\mathrm{TX}}}} \exp\left(-1/\sigma^2 \left\| r - h_1 s_1 - \cdots - h_r s_r - \mathcal{H}(s_{r+1},\ldots,s_{n_{\mathrm{TX}}})^T \right\|^2\right)}\right). \tag{5}$$

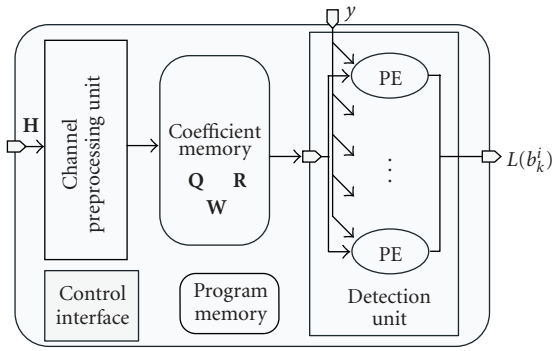


Figure 4: Block diagram of the dual-mode MIMO detector.

3.1. Linear Detection. In linear detection such as Zero-forcing (ZF) and Minimum Mean Squared Error (MMSE), the receiver symbol vector r is multiplied with a linear filter:

$$\text{ZF} : \widetilde{s} = \left(\mathbf{H^H H}\right)^{-1}\mathbf{H^H r} = s + \widetilde{n}_{\text{ZF}}, \tag{6}$$

$$\text{MMSE} : \widetilde{s} = \left(\mathbf{H^H H} + \boldsymbol{\sigma^2 I}\right)^{-1}\mathbf{H^H r} = s + \widetilde{n}_{\text{MMSE}}. \tag{7}$$

The correlation between the elements in the noise vector $\widetilde{n}_{\ldots}$ is neglected and the symbols in s are demodulate individually, treating the output of the model (6) as n_{TX} independent scalar channels. Although linear detectors will incur a severe performance loss in slow fading channels [4], they have very low implementation cost compared to more advanced MIMO detection algorithms which makes them suitable for low-cost real-time implementations. As depicted in Figure 3, the linear detection procedure involves two parts: channel preprocessing and symbol demapping. The channel preprocessing procedure mainly consists of matrix multiplication and inversion as shown in (6) and (7).

3.2. Fixed-Complexity Soft Output (FCSO). The Layered Orthogonal Lattice Detector (LORD) proposed in [13] and the FCSO MIMO detector presented in [4] are similar and use a suboptimal method to reduce the complexity at the cost of negligible performance loss. A general $n_{\text{TX}} \times n_{\text{RX}}$ MIMO system using 64-QAM is taken as a case study. Here each complex-valued symbol is considered to be one layer and only the top layer is exactly marginalized with the remaining three layers approximately marginalized. The channel-rate processing of FCSO involves the QRD of n_{TX} rank-reduced channel matrices

$$\mathcal{H}_{\bar{k}} = [h_1, \ldots, h_{k-1}, h_{k+1}, \ldots, h_{n_{\text{TX}}}], \tag{8}$$

which generates an upper triangular matrix $R_{\bar{k}}$, and a unitary matrix $Q_{\bar{k}}$ so that

$$\mathcal{H}_{\bar{k}} = \mathbf{Q}_{\bar{k}}\mathbf{R}_{\bar{k}}. \tag{9}$$

Here n_{TX} QRD is needed for different $\mathcal{H}_{\bar{k}}$.

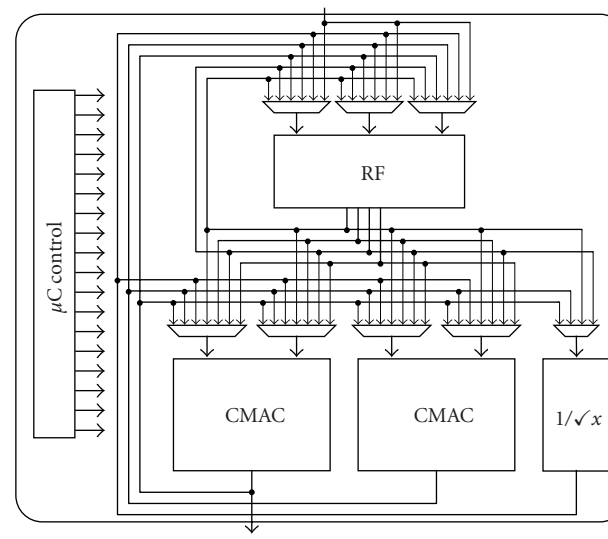


Figure 5: Channel preprocessing unit.

The symbol-rate processing consists of the following steps.

(1) Pick one transmitted symbol $s_i, i \in (1, \ldots, n_{\text{TX}})$ as the top layer. The entire constellation $\mathcal{L}$ is enumerated in the exact marginalization ($\sum$ in (5)) only for s_i. For the kth candidate $\hat{s}_i^k$ in $\mathcal{L}$, by canceling its effect on the received symbol vector r, a new vector

$$\hat{r} = r - \widetilde{h}_i \hat{s}_i^k \tag{10}$$

is computed.

(2) By multiplying $\hat{r}$ with $\mathbf{Q}_{\bar{k}}^H$ from (9), compute

$$\widetilde{r} = \mathbf{Q}_{\bar{k}}^H \hat{r}. \tag{11}$$

(3) Based on $\widetilde{r}$ and R, using DFE, $\hat{s}_b = [\hat{s}_2 \hat{s}_3 \cdots \hat{s}_{n_{\text{TX}}}]^T$ can be estimated using hard decision. From this, compute the Euclidean distance

$$\delta_k = \left|\left|\hat{r} - \mathbf{R}_{\bar{k}}\hat{s}_b\right|\right|^2 \tag{12}$$

and eventually the log-likelihood ratio (LLR). Taking a 64-QAM system as an example, as shown in the following:

$$\mu(b_1, \ldots, b_{24}) = \exp\left(-\frac{1}{\sigma^2}\delta_k\right) \tag{13}$$

the LLR of the six bits that constitute the top-layer symbol can be computed using (12). This involves the computation of 64 different $\delta_k, (k = 1, \ldots, 64)$ as shown in (14)

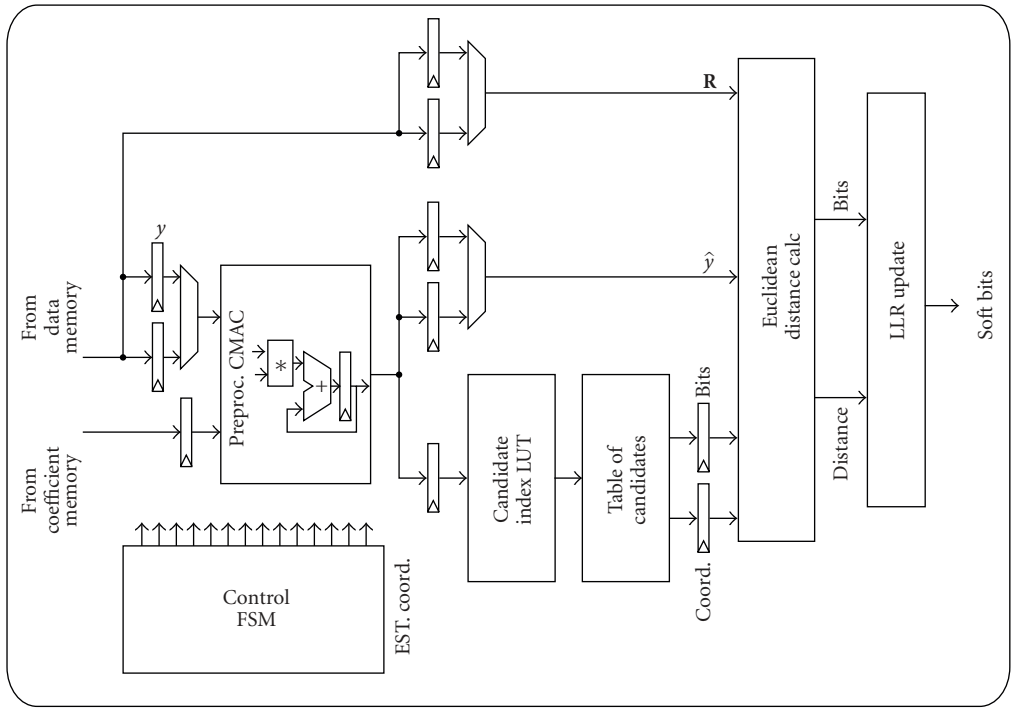


Figure 6: PE in detection unit.

$$L(b_i r) \approx \log\left(\frac{\sum_{b(\hat{s}_1)_1=0}^{1}\cdots\sum_{b(\hat{s}_1)_{i-1}=0}^{1}\ \sum_{b(\hat{s}_1)_{i+1}=0}^{1}\cdots\sum_{b(\hat{s}_1)_6=0}^{1}\left(\max_{b(\hat{s}_2)_1,\cdots,b(\hat{s}_4)_6}\mu(b_1,\cdots,b_{i-1},1,b_{i+1},\ldots,b_{24})\right)}{\sum_{b(\hat{s}_1)_1=0}^{1}\cdots\sum_{b(\hat{s}_1)_{i-1}=0}^{1}\ \sum_{b(\hat{s}_1)_{i+1}=0}^{1}\cdots\sum_{b(\hat{s}_1)_6=0}^{1}\left(\max_{b(\hat{s}_2)_1,\cdots,b(\hat{s}_4)_6}\mu(b_1,\cdots,b_{i-1},0,b_{i+1},\ldots,b_{24})\right)}\right). \quad (14)$$

3.3. Modified FCSO (MFCSO). Although the FCSO detector has substantially reduced the complexity compared to MAP detector, further reduction is still needed for a practical implementation with large signal constellations. In the following, further approximations and improvements to FCSO detection, namely Modified FCSO (MFCSO) detector [5], are elaborated. In [4], the entire constellation $\mathcal{L}$ is enumerated in the exact marginalization ($\sum$ in (5)). In this paper, instead of searching the full constellation $\mathcal{L}$, we propose to sum over only a subset $\mathcal{L}_s \subset \mathcal{L}$ of constellation points around an initial estimate $\hat{s}$. This initial estimate will be obtained by zero-forcing detection. The size of $\mathcal{L}_s$, denoted by N, is chosen to be 16 and 8 in this paper for the complexity and performance comparisons. In effect, the proposed detector is a further approximation of that in [4], which consists of only partially enumerating the symbols selected for exact marginalization (the set $\mathcal{L}$ in (5)).

Similar to FCSO, the channel-rate processing of MFCSO involves computing QRD n_{TX} times, as shown in (9) and (8). As an overhead compared to FCSO, the coefficient matrix

$$\mathbf{W} = \left(\mathbf{H^H H} + \boldsymbol{\sigma}^2\mathbf{I}\right)^{-1}\mathbf{H^H} \quad (15)$$

is needed to perform the ZF/MMSE-based initial estimate of $\hat{s}$ in (16) below. The symbol-rate processing of MFCSO is the following

(1) Linear detection (ZF/MMSE) is carried out to estimate the initial symbol vector

$$\hat{s} = \min_{s_k\in\mathcal{L}}\|\mathbf{H}s - r\|^2. \quad (16)$$

Here s is the transmitted symbol vector, s_k is the kth symbol in it.

(2) For each initially estimated symbol $\hat{s}_k, k \in \{1,\ldots,n_{\mathrm{TX}}\}$, a candidate set $\mathcal{L}_k$ is created. $\mathcal{L}_k$ contains N lattice points close to $\hat{s}_k$.

(3) For each point $l \in \mathcal{L}_k$, approximate marginalization is applied to the rest of the layers either via ZF or ZF-DFE. According to (17), a multiplication of Q_k^H and $\hat{r}$ is needed for each $\hat{r}$ which is updated proportionally to the size of $\mathcal{L}_k$ and the symbol rate. However, note that

$$\tilde{r} = \mathbf{Q}_k^H\hat{r} = \mathbf{Q}_k^H(r - h_k l) = \mathbf{Q}_k^H r - \left(\mathbf{Q}_k^H h_k\right)l, \quad (17)$$

where $\mathbf{Q}_k^H h_k$ is an $n_{\mathrm{TX}} \times 1$ vector, which can be precalculated at channel rate.

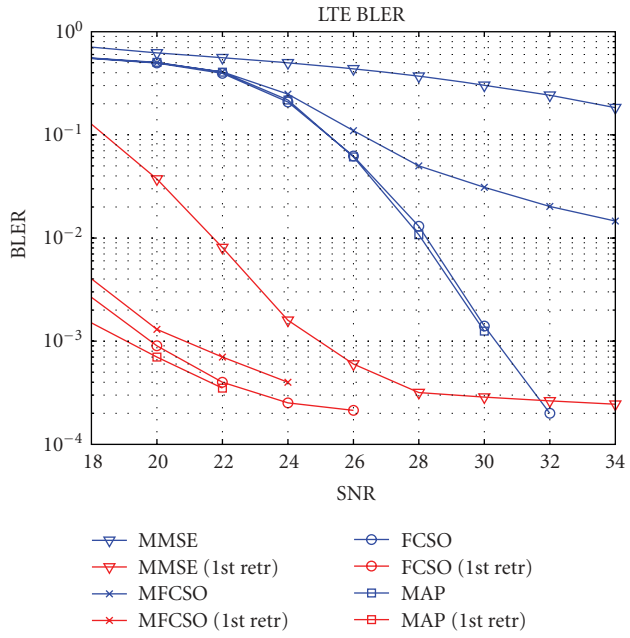


Figure 7: Block error ratio (2×2 SM, CQI=15), red curves are the BLER of the 1st retransmission of H-ARQ.

(4) Using back substitution [14], $\hat{s}_b$ can be estimated from

$$\hat{s}_b = \arg\min_{s_k \in \mathcal{L}} ||\mathbf{R}_{\bar{k}} \hat{s}_b - \tilde{r}||^2. \tag{18}$$

(5) $\hat{s}_b$ together with $\hat{s}_k$ form a complete possible transmitted symbol vector which has an Euclidean distance

$$\delta_l = ||\mathbf{R}_{\bar{k}} \hat{s}_b - \tilde{r}||^2. \tag{19}$$

(6) In total, there will be N different $l \in \mathcal{L}$ values for each layer, and there will be four layers each being the top layer once. Therefore, for a 4×4 system, $4N$ different δ_l values need to be computed. In case $N = 16$, there will be 64 different δ_l values which is 1/4 compared to the FCSO proposed in [4].

(7) For the sake of low complexity, instead of MAP detection, the following approximation can be used, so that

$$L(b_i(s_k)) \approx -\frac{1}{\sigma^2} \left\{ \min_{\mathbf{l} \in \mathcal{L}_k : \mathbf{b_i}(\mathbf{s}_k) = \mathbf{0}} \boldsymbol{\delta}_\mathbf{l} - \min_{\mathbf{l} \in \mathcal{L}_k : \mathbf{b_i}(\mathbf{s}_k) = \mathbf{1}} \boldsymbol{\delta}_\mathbf{l} \right\}. \tag{20}$$

As presented in [5], the performance gap between MAP and MFCSO for 4×4 MIMO using 64-QAM and 3/4 convolutional coding was proven to be small when $N = 16$ (0.5 dB when FER=10^{-2}). The gap increases to 2 dB when $N = 8$. On the other hand, the complexity of the detector when $N = 16$ is already feasible for VLSI implementation.

3.4. MFCSO in LTE and WiMAX. As a simplification of the general MFCSO algorithm presented in Section 3.3, a 2×2 MFCSO method for SM is elaborated in the following. Considering each complex-valued symbol as one layer, only one of them is exactly marginalized and the other is approximately marginalized (using DFE hard decision). The channel rate processing of MFCSO involves the QR decomposition (QRD) of two 2×2 channel matrices which are $\mathbf{H_1} = \mathbf{H}$ in (3) and

$$\mathbf{H_2} = \begin{bmatrix} h_{12} & h_{11} \\ h_{22} & h_{21} \end{bmatrix}. \tag{21}$$

The QRD generates an upper triangular matrix R, and a unitary matrix Q according to (9).

The detection procedure for 2×2 SM described in the following text is slightly different from the MFCSO presented in [5].

(1) Linear detection in (16) is carried out to estimate the 2×1 initial symbol vector

$$\hat{s}_{\text{init}} = \min_{\hat{s}_{\text{init},k} \in \mathcal{L}} ||\mathbf{H_1} s - r||^2. \tag{22}$$

Here s is the transmitted symbol vector, within which, s_k is the kth symbol.

(2) For each initially estimated symbol $\hat{s}_{\text{init},k}$, $k \in \{1, 2\}$, a candidate set $\mathcal{L}_k$ is created. $\mathcal{L}_k$ contains N constellation points close to $\hat{s}_{\text{init},k}$.

(3) First s_2 is chosen as the top-layer symbol. In order to perform DFE,

$$\tilde{r} = \mathbf{Q}_\mathbf{1}^H. \tag{23}$$

needs to be computed. The same operation is needed once again when s_1 is chosen as the top layer later.

(4)For the n^{th} constellation point $\zeta_n \in \mathcal{L}_2$, its effect on $\tilde{r}_1$ will have to be canceled out.

$$\tilde{r}_1' = \tilde{r}_1 - \mathbf{R_1}(1, 2)\zeta_n \tag{24}$$

Based on ζ_n, the partial Euclidean distance

$$\delta_n = ||\mathbf{R_1}(2, 2)\zeta_n - \tilde{r}_2||^2 \tag{25}$$

computed for the top-layer.

(5) DFE is applied to detect the other layer. Using back substitution [14], $\hat{s}_1$ can be estimated from

$$\hat{s}_1 = \arg\min_{\hat{s}_1 \in \mathcal{L}} ||\mathbf{R_1}(1, 1)\hat{s}_1 - \tilde{r}_1'||^2. \tag{26}$$

(6) The estimated $\hat{s}_1$ together with $\hat{s}_2 = \zeta_n$ form a complete possible transmitted symbol vector $\hat{s}$, from which an accumulated full Euclidean distance

$$\delta_n = \delta_n + ||\mathbf{R_1}(1, 1)\hat{s}_1 - \tilde{r}_1'||^2 \tag{27}$$

can be computed.

(7) In total, there will be N different δ_n computed when s_2 is chosen as the top layer. Then s_1 is chosen as the top-layer symbol as well. Based on $\mathbf{Q_2}, \mathbf{R_2}$, and $\hat{s}_{init,1}$, the same procedure needs to be done once again to compute another N different δ_n. Hence, for the 2×2 system, $2N$ different δ_n values need to be computed. They are used to update the LLR values in the end as described in [5].

Table 1: Operations supported by ChPU.

Operation	Description
Cplx squared abs	$c = a.r^2 + a.i^2$
Sum squared abs	$c = a.r^2 + a.i^2 + b.r^2 + b.i^2$
Cplx inner product	$c = \sum(a_i.r^2 + a_i.i^2)$
Cplx multiply	$c.r = a.r * b.r - a.i * b.i$
	$c.i = a.r * b.i + a.i * b.r$
Cplx multiply-add	$c.r = c.r + a.r * b.r - a.i * b.i$
	$c.i = c.i + a.r * b.i + a.i * b.r$
Real-Cplx multiply	$c.r = a.r * b; c.i = a.i * b$
Real Inverse-Sqrt	$b = 1/\sqrt{a}$

4. Flow Analysis of MIMO Detection

Independent of the detection method, the processing flow of MIMO symbol detection can always be partitioned into two parts, namely channel-rate processing and symbol-rate processing as depicted in Figure 3.

4.1. Channel-Rate Preprocessing. The channel preprocessing is about the precalculation of equalization coefficient matrices from the estimated channel matrix **H**. According to (15)), the computation involved in linear detection is mainly matrix manipulation including matrix multiplication and inversion. Here the matrix **H** can be a complex-valued matrix of arbitrary size. As mentioned in [15], in practice, the size of **H** is typically between 2×2 and 4×4. Although larger matrices (e.g., 8×8) can still be managed [15], the cost of real-time implementation will be much higher. For MFCSO, channel-rate processing includes the QR decomposition in (9). For MFCSO, aside from computing **W**, QR decomposition is also needed according to (9).

4.2. Symbol-Rate Processing. The symbol-rate processing in soft-output linear detection [16] is to demap the equalized complex values to soft bits. In case of near-MAP detection methods such as MFCSO, layered processing is involved which requires substantially more computational effort. As described in Section 3.3, the symbol-rate processing in MFCSO involves the multiplication, subtraction, and computing the Euclidean distance based on estimated symbols.

5. Architecture of the MIMO Detector

The block diagram of the MFCSO detector is depicted in Figure 4. The detector contains two major parts, the channel preprocessing unit (ChPU) and the detection unit (DU). As presented in Section 3.3 and [5], it is decided that the candidate set size $N = 16$ for 64-QAM. It allows real-time detection of both 2×2 STBC/SFBC and SM for LTE and WiMAX. Modulation schemes from QPSK to 64-QAM are supported.

5.1. Channel Preprocessing Unit. The ChPU as depicted in Figure 5 handles channel-rate processing tasks such as computation of **W** in (15) and the QR decomposition in (9). These are performed every time the estimated channel is updated. The computed coefficient matrices W will be stored in the coefficient buffer and fed to the LLR demapper as input. As depicted in Figure 5, ChPU contains two Complex-valued Multiply-and-ACcumulate (CMAC), an inverse-square-root unit and a 32-bit register file containing 24 registers. The ChPU is a programmable unit controlled by microcode. The operations supported by the ChPU are listed in Table 1. The method presented in [16] has been used to compute **W**, and the Modified Gram-Schmidt method [14] is used to compute **Q** and **R** matrices in (9).

5.2. Detection Unit. The DU computes the LLR values using the method presented in Section 3 and the Log-Max approximation in (20)

$$L\left(b_k^i\right) = -\frac{1}{\sigma^2}\left\{\min_{\mathbf{l}\in\mathcal{L}_k:\mathbf{b}_k^i=\mathbf{0}} \boldsymbol{\delta} - \min_{\mathbf{l}\in\mathcal{L}_k:\mathbf{b}_k^i=\mathbf{1}} \boldsymbol{\delta}\right\}. \tag{28}$$

The DU consists of a number of processing elements (PE) as illustrated in Figure 6 which can utilize the parallelism in the MFCSO algorithm. The computed LLR values $L(b_k^i)$ can be either directly passed to the channel decoder or combined with previously stored LLR values in the soft-buffer for H-ARQ. Since the processing in DU is at symbol rate which is much higher than the channel-rate processing in ChPU, a fully pipelined architecture is used in DU to allow the computation of 16 different δ_n in (27) to be finished within 16 clock cycles. DU is configured by a control register and can bypass the functions defined in Section 3 to only enable MMSE detection with soft output. The MMSE mode can be used in power saving mode to reduce the power consumption with a loss of detection performance. A 16-bit fixed-point datatype with proper scaling is adopted in DU, the output LLR values are quantized to be 6-bit signed integers. The number of PE in the DU is decided at design time according to the processing load and latency analysis. In this paper, it is chosen to be two based on the latency analysis in Section 9.3.

5.3. Memory Subsystem. The MIMO detector itself does not contain memory except the small program memory. In order to store the temporarily computed **W**, $\mathbf{Q_1}$, $\mathbf{R_1}$, $\mathbf{Q_2}$, and $\mathbf{R_2}$ which are updated by the channel preprocessor at the channel rate, a coefficient buffer as depicted in Figure 4 is needed. The coefficient memory stores the above values for all data subcarriers (up to 20 MHz bandwidth for LTE and 10 MHz to WiMAX). The FIFO that stores the incoming data to the detector from the channel estimator and the subcarrier demapper is not shown in the figure, neither is the FIFO that passes the computed LLR values to the channel decoder hardware. Note that in case STBC is used, the number of data stored in W memory can be reduced almost by half owing to the Alamouti features of **W**, and no **Q** and **R** matrices are needed.

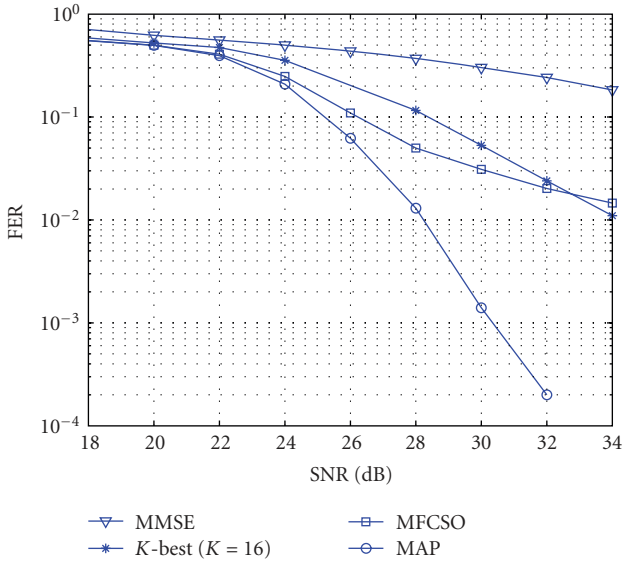


Figure 8: LTE coded frame Error rate (rate 0.926, 64-QAM).

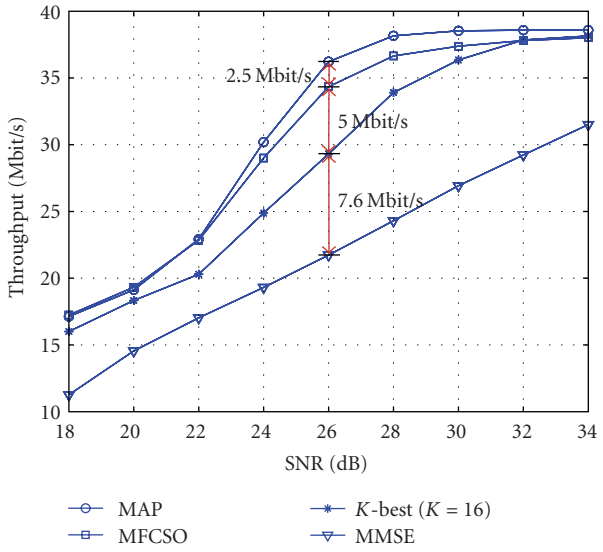


Figure 9: LTE coded throughput (rate 0.926, 64-QAM).

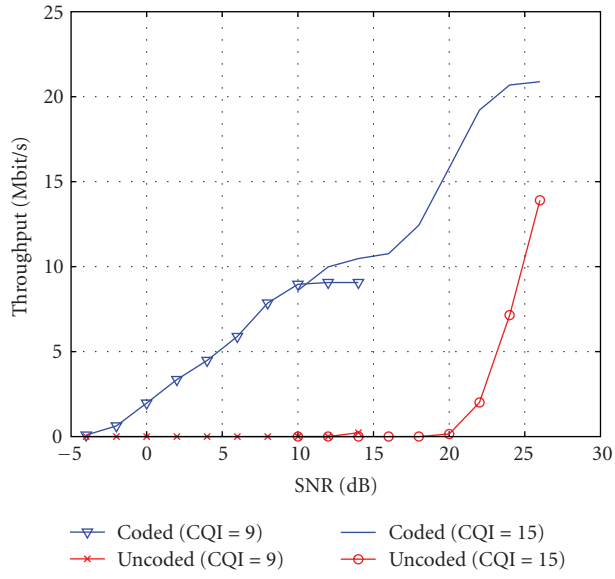


Figure 10: Throughput (2 × 2 SFBC, MMSE).

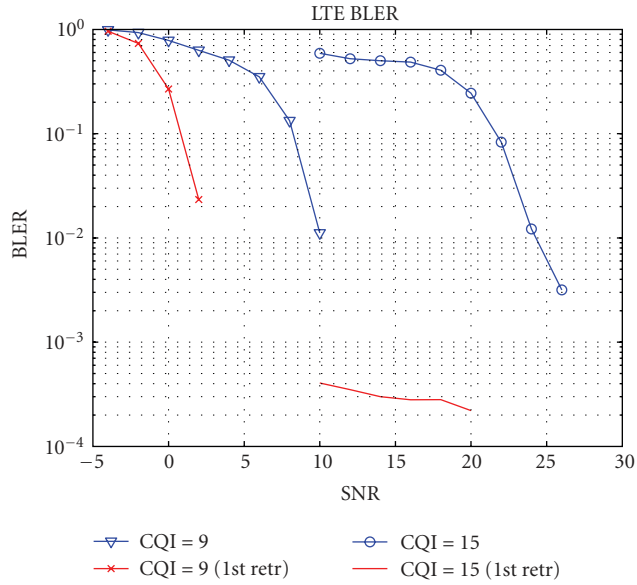


Figure 11: Block error ratio (2 × 2 SFBC, MMSE).

6. Performance Evaluation

In order to evaluate the performance of various MIMO detection algorithms, simulation is carried out using link-level 3GPP LTE and WiMAX simulators [17, 18]. The simulators are developed using MATLAB and C.

It includes the complete physical layer signal processing such as timing/frequency synchronization, channel estimation, subcarrier demapping, rate-matching, and turbo decoding. H-ARQ based on CRC of coded blocks is also enabled to support chase combine (CC) with up to three retransmissions. The bandwidth is set to be 5MHz in the simulation, the velocity of UE is 3 km/h and the scenario is urban micro [19]. Perfect synchronization and channel estimation are assumed to focus the simulation on detection performance. The Turbo decoder runs at most six iterations with early stopping. The WiMAX simulator [17] also works on 5MHz bandwidth. Two channel coding methods used in the simulation are Reed-Solomon with Convolutional (RS-Conv) and Low-Density Parity-Check (LDPC) coding. Two channel models namely the 3GPP SCME [19] and ITU Pedestrian B (PedB) [17] channel models are used in this paper. It is assumed the channel is quasistatic within one OFDM symbol duration. Note that the 1-TTI latency is introduced for uplink ACK/NACK in the simulation.

6.1. 3GPP LTE. Figure 7 shows the block error rate (BLER) of the LTE system with H-ARQ using different detection

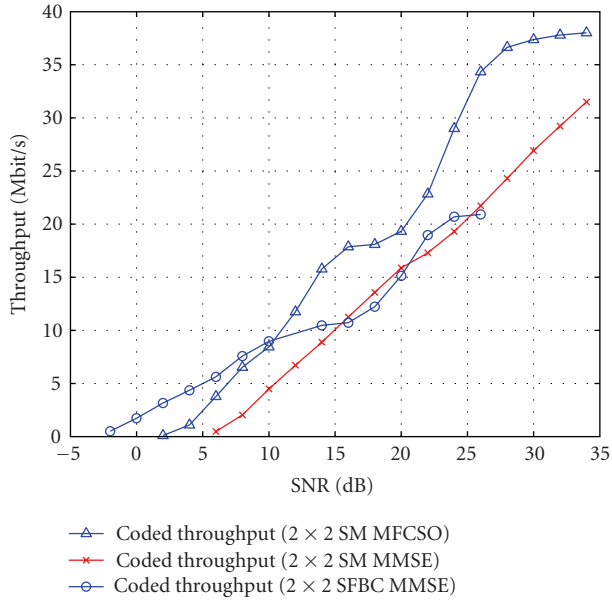


Figure 12: Coded throughput with 2-level AMC (CQI 15 and 9).

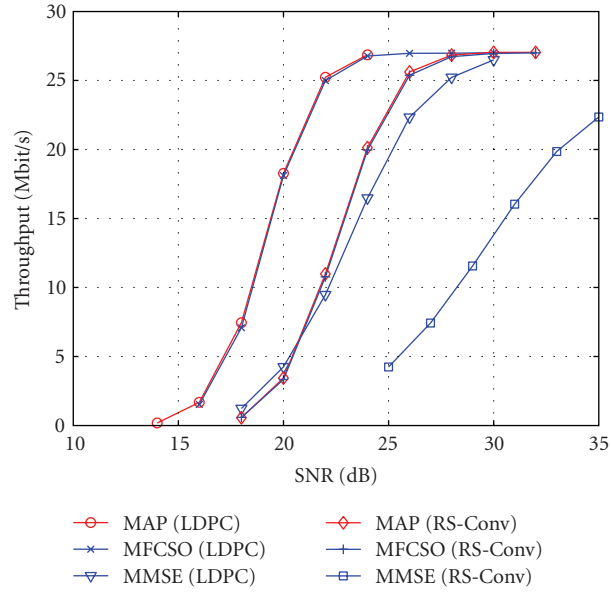


Figure 14: WiMAX coded throughput (rate 0.75, 64-QAM).

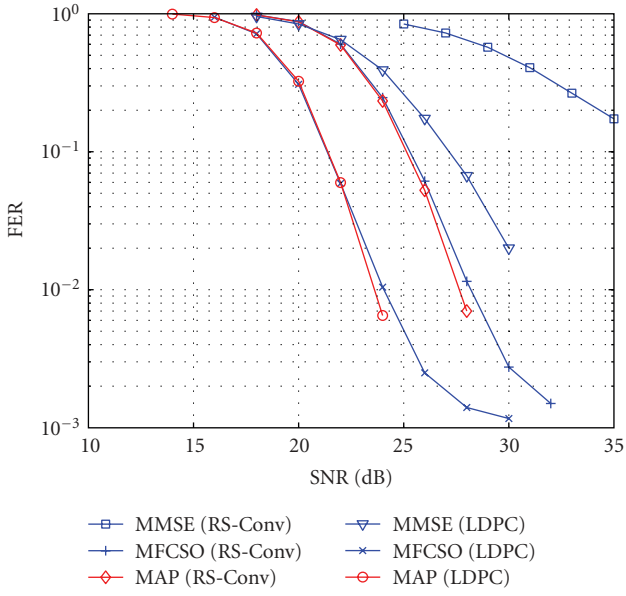


Figure 13: WiMAX coded frame error rate (rate 0.75, 64-QAM).

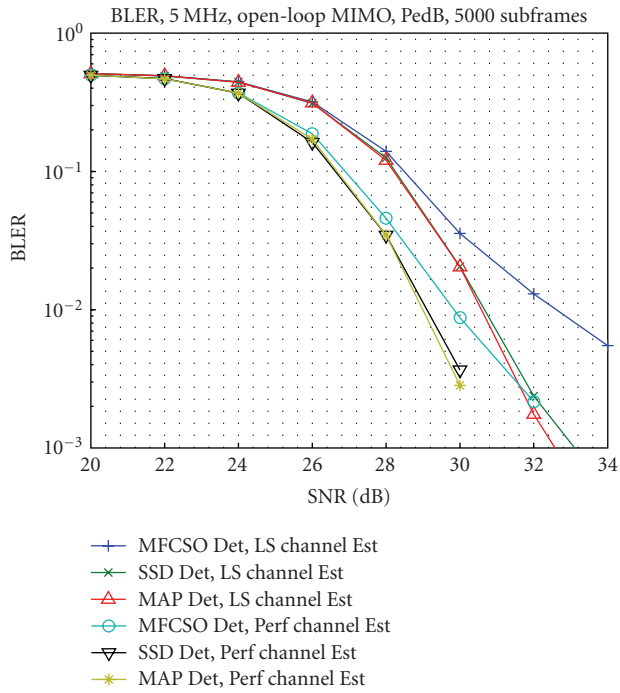


Figure 15: LTE bLock error rate with H-ARQ (CQI=14), PedB.

methods. The blue curves are the BLER of the first transmission while the red ones represent that of the first retransmission in H-ARQ. The figure shows that the BLER of the retransmission is drastically reduced compared to the first transmission which improves the throughput as shown later.

The result in Figures 8 and 9 shows that in case of 64-QAM and the weakest (rate 0.926) channel coding defined in LTE is used, for 2×2 SM, the FER performance of MAP is always better than that of MFCSO and K-best. MFCSO achieves lower FER than the K-best ($K = 16$) used in [10] until very high SNR. MMSE has the worst FER performance.

Note that in wireless systems, throughput is a more important performance factor than BER or FER because it has a direct effect on the user experience. Figure 9 shows that the gain in throughput brought by MFCSO against MMSE is significant (up to 12.6 Mbits/s, or 55% higher than the one achieved by MMSE). In comparison, the throughput performance degradation caused by the approximation in MFCSO is much smaller (up to 2.5 Mbits/s, or 7% lower than that achieved by MAP). The much smaller gap in

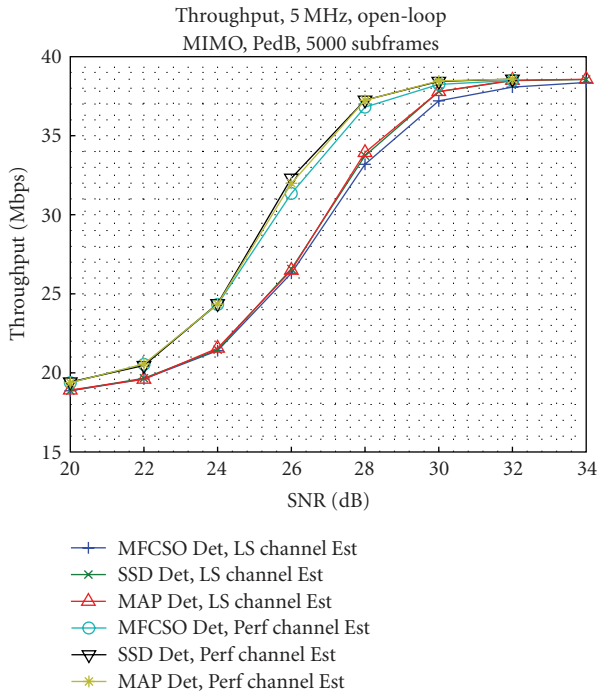


Figure 16: LTE throughput with H-ARQ (CQI=14), PedB.

Table 2: Minimum SNR to reach FER=0.01.

CQI	SFBC (MMSE)	SM (MFCSO)	SM (MMSE)
9	10 dB	17 dB	24 dB
15	24 dB	36 dB	N/A

throughput in comparison to that of FER mainly owes to the H-ARQ retransmission with chase combining. The result shows that even with a sub optimal detector (with much lower complexity than the optimal detector) and almost no channel coding, a throughput that is close to the one achievable by MAP detectors can still be reached when H-ARQ is used. The throughput gain of MFCSO over the K-best is as significant as 5 Mbits/s (14%), when SNR is 26 dB.

Figures 10 and 11 show the BLER and throughput of 2×2 SFBC with two different CQI values (9 and 15). The simulation shows that SFBC reaches FER = 0.01 at much lower SNR than SM as depicted in Table 2, though the throughput is half.

Figure 12 depicts the achievable throughput using two-level adaptive modulation and coding (AMC). The result shows that when SNR is worse than 10 dB, SFBC achieves both higher throughput and lower BLER than SM even if MAP detector is used.

6.2. WiMAX. The result in Figures 13 and 14 shows that when mild channel coding (e.g., RS-Conv 3/4) is used without H-ARQ in the WiMAX system, MFCSO still achieves near-MAP performance in FER and MAP performance in throughput. It has a gain of more than 9 dB compared to the MMSE detector. The use of stronger code (e.g. LDPC) will bring a gain of 4 dB in throughput compared to RS-Conv. This shows that MFCSO has a very promising performance/complexity trade-off taking the advance of channel coding into consideration. The result also shows that once FER reaches 0.01, any further improvement of FER gives only negligible increase in throughput.

6.3. Impact of Channel Estimation Error. In most of the literatures [1, 3, 5], perfect channel state information (CSI) is assumed which is never true in reality. In [4], channel estimation error is emulated with a randomly generated error constrained by the value of its average power, and the affected FER is plotted. However, how the channel estimation error affects the link-level performance of MIMO detection with the presence of H-ARQ has not been studied according to the best knowledge of the authors. In this paper, based on the least square (LS) channel estimation, the impact of channel estimation error on link-level performance is investigated, which provides a realistic measurement of the achievable performance of the MFCSO detector in a practical system. In this paper, an LTE system with CQI = 14 (coding rate 0.8547, 64-QAM) and open-loop 2×2 MIMO scheme is simulated using PedB channel. For comparison purposes, the MFCSO detector is benchmarked against the soft-output sphere decoding (SSD) in [1] and the MAP detector. However, note that no complexity reduction of SSD as used in [1] is applied in this paper, thus, the SSD performance reaches the upper bound. As depicted in Figure 15 and 16, regardless of the channel estimation error, SSD always achieves the same BLER and throughput performance as MAP detection. In Figure 15, the slope of the BLER curve of MFCSO will decrease when SNR reaches 28 dB. Considered from traditional point of view, the BLER performance of MFCSO is significantly worse than SSD and MAP (more than 2 dB). However, as shown in Figure 16, the throughput performance of MFCSO is only negligibly lower (0.3 dB) than that of SSD and MAP. This further proves that MFCSO has a better performance/complexity trade-off when taking system-level impact into consideration. Figure 16 also shows the throughput gap between the case assuming perfect CSI and the one with realistic LS estimated CSI is 1.5 dB in the active region for CQI = 14. In principle, channel estimation error will only cause the throughput curve to shift right by 1.5 dB.

7. Implementation Considerations

In LTE [11], taking a 5 MHz bandwidth LTE system as an example, up to 7 OFDM symbols need to be processed within one slot (0.5 ms) which contain 1900 data subcarriers. This means that there will be no more than 0.26 μs to finish the detection of each subcarrier on average. Therefore, proper detection methods have to be chosen in order to maximize the data rate at reasonable implementation cost.

As depicted in (7), for 2×2 SM, the MMSE detector needs to compute the inverse of a 2×2 matrix. It has been presented in [16] that the inversion of small matrices can be done using direct inversion which supplies sufficient precision for most of the channels. The FCSO and MFCSO detector involves the

Table 3: Complexity analysis for ASIC implementation (65 nm).

		MMSE	MFCSO	FCSO	MAP
Num nodes	16-QAM	1	18	32	256
	64-QAM	1	32	128	4096
Logic (mm^2)	64-QAM	0.08	0.2	0.6	20

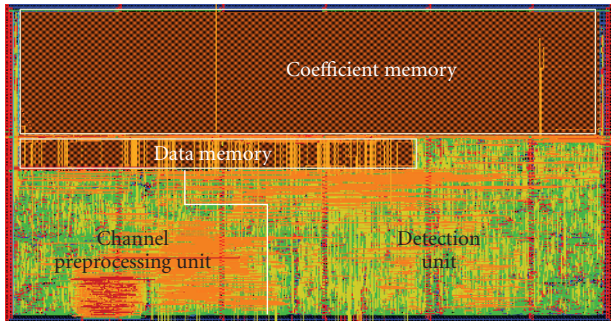


Figure 17: Layout of the MIMO Detector.

Table 4: Adaptive transmission and detection.

SNR range	SFBC	SM
High-end (MFCSO/MMSE)	−2 dB → 10 dB	≥10 dB
Low-end (MMSE only)	−2 dB → 26 dB	≥26 dB

Table 5: FPGA implementation result for real-time processing.

	This work	Ref [10]
Algorithm	MFCSO	K-best LSD
Modulation supported	up to 64-QAM	
FPGA type	Virtex2	
Datatype	fixed-point	
Wordlength (bits)	16	
Num of slices	4381	15662
Num of MULT18X18s	48	108
Block RAMs	3	61
Frequency (MHz)	85	70
Throughput for 64-QAM (Mbps)	67.5	6

search of a number of trellis nodes as depicted in Table 3. The FCSO detector always visits the complete constellation (e.g., 16 for 16-QAM and 64 for 64-QAM), while MFCSO only visits a subset of it (e.g., 9 for 16-QAM and 16 for 64-QAM). Note that MFCSO requires MMSE detection to compute the initial estimate (22) which is an extra cost compared to FCSO. To the knowledge of the author, SSD with complexity reduction [1] has a similar complexity compared to FCSO, which is not analyzed in this paper due to the limited space.

In practice, the hardware is usually implemented taking both the cost and performance issues into consideration. Based on the complexity analysis in Table 6 and the performance analysis in Section 6, MFCSO falls into the favor of the authors to be chosen as the target algorithm for ASIC implementation. Using ST 65 nm CMOS process, while meeting the 0.26 μs constraint, the implemented detector supporting both MMSE and MFCSO for 2×2 SM and up to 64-QAM modulation occupies less than 0.2 mm^2 as proven later.

8. Adaptive Transmission and Detection

As depicted in Table 3, a detector supporting dual-mode MFCSO/MMSE detection consumes 2.5 times the area of the one only supporting MMSE. Hence, the former one is assumed to target high-end users willing to pay more in area and power for performance (e.g., laptops). The MMSE single-mode detector is in favor of low-end users for connectivity with minimum cost (e.g., smartphones). Note that the user cares about latency as well as throughput, and latency is partly determined by the number of retransmissions. Hence, it is also important to keep the retransmissions to a minimum (which requires low FER). Figure 12 shows that with AMC, SM using MFCSO detector always brings higher throughput when SNR is greater than 10 dB. For both types of users, when SNR is worse than 10 dB (as in Figure 12), SFBC is preferred instead of SM. For low-end users, SM can be used when SNR ≥ 25 dB while SFBC is still preferable (due to the low FER thus fewer retransmissions resulting in low latency) to be used from 10 to 25 dB. For high-end users, SM is preferred when SNR is at least higher than 10 dB. On the other hand, the MMSE mode will consume substantiately lower power than the MFSCO mode, the high-end users might only want to switch to MFCSO-mode when there is enough battery power and high SNR (e.g., ≥25 dB). When SNR is very low, SFBC is also preferred due to its robustness (as depicted in Figure 12). The SNR ranges suggested for the mode switching of two types of detector hardware are shown in Table 4. The adaptive scheme brings power efficiency and can supply best-effort performance in an economic way.

9. Final VLSI Implementation

The implementation of our design is done in two steps. First, for fast prototype and to compare with the prior art in [10], the symbol detector is implemented using Xilinx FPGA. Second, ASIC flow including synthesis, floorplan, placement, and routing is carried out using ST 65 nm process libraries and Synopsys low-power design flow.

9.1. FPGA Prototype. Xilinx ISE and Core Generator were used to synthesize the design based on the Virtex2 xc2v6000 FPGA. The synthesis result is depicted in Table 5. The proposed implementation supports up to 64-QAM as described in Section 9.3. Table 5 shows that it consumes 72% fewer slices and 56% fewer embedded multipliers compared to the K-best detector presented in [10]. Note that the K-best FPGA implementation in [10] only supports the real-time detection of 2×2 QPSK spatial multiplexing in LTE. The FPGA-based detector presented in [8] covers a different antenna configuration, and most importantly the Virtex-5 FPGA used has a different architecture from the Virtex-2 FPGAs, which makes it difficult to make an area comparison.

Table 6: ASIC implementation result.

Area of channel preprocessing unit (kgate)	35
Area of detection unit (kgate)	55
Cycles for $\frac{1}{\sqrt{x}}$	3
Working frequency (MHz)	300
Throughput for 64-QAM (Mbps)	225

9.2. ASIC Implementation. Table 6 depicts the gate count, and working frequency of the ASIC implementation. In reality, the channel coefficients are updated less frequently than the received symbols, thus, they are saved in the coefficient memory which is not counted in [10]. In order to compare the area consumed by memory and the detector itself, a demo chip including a 172800 bit coefficient memory and a 19200 bit data memory for 5 MHz bandwidth is implemented using Cadence backend flow. As depicted in Figure 17, the total area of the detector is 0.37 mm^2 with half of it consumed by the actual logic ($\approx$0.2 mm^2) of the detector and the other half by the memory. Note that the microcode memory is implemented as a piece of logic in the chip. The size of the memories depends on the number of subcarriers (or bandwidth) to be supported. The K-best detector in [20] supports 4×4 MIMO and 100 Mbps data rate. As mentioned in Section 3.3, the complexity of MFCSO is proportional to n_{RX}^2. Hence, the area of the detection part for 4×4 will be four times of the presented 2×2 solution. Compared to the 31 mm^2 figure of a K-best detector for 4×4 MIMO in 0.13-μm running at 270 MHz (which is 7.5 mm^2 without memory in 65-nm according to CMOS scaling), the solution in this paper is 0.8 mm^2 without memory. Also note that [20] does not include the channel preprocessing part which is expected to give a major contribution in area (it already consumes half of the area of this solution).

9.3. Processing Throughput. Taking the assumption made in [10], for LTE system with 5 MHz bandwidth, there will be at most 300 data subcarriers to be processed within one OFDM symbol duration which is 83 μs. This requires the detection of each data subcarrier to be finished within 277 ns. For the FPGA implementation which has a clock frequency of 90 MHz, this amount of time is equal to around 25 clock cycles. Note that the detector can process two subcarriers in parallel which means each subcarrier can be finished within 16 cycles. For 2×2 spatial multiplexing and 64-QAM (12 bits per subcarrier), this corresponds to $(90/16) \times 12 = 67.5$ Mbps processing throughput.

The ASIC implementation can easily run at a clock frequency of 300 MHz which means 1570 data subcarriers can be computed within 83 μs. This corresponds to 225 Mbps processing throughput which allows real-time detection of 20 MHz bandwidth LTE downlink (containing up to 1200 data subcarriers) to be supported. Since the WiMAX 2004 [17] only uses 10 MHz bandwidth, it has a lower peak data rate than LTE, thus can be easily supported.

Note that the MFCSO detector can be switched to MMSE mode by poweringdown the major part of the DU. The detection in SFBC/STBC transmission schemes is in fact MMSE detection which can be handled by the MMSE mode. Since the MMSE mode will consume substantially less power than the MFCSO mode, the detector is switched to MMSE mode when the terminal enters power-saving mode.

10. Conclusion

In this paper, the VLSI implementation of a fixed complexity near-MAP MIMO detector ASIC is presented for multistandard wireless terminals. It achieves near-MAP throughput during LTE simulations, even with a relatively weak channel code and with high-order modulation (e.g., CQI = 15). Furthermore, based on the adaptive scheme proposed in Section 8, a good performance and power trade-off can be achieved. In comparison to prior art such as the K-best solution in [10], the detector presented achieves better performance and lower silicon cost. The impact of realistic channel estimation on detection performance is also presented.

Acknowledgments

The work of D. Wu, J. Eilert, R. Asghar, and D. Liu is supported by the Multibase Project from European Commission's 7th Framework in partner with Ericsson AB, Infineon AG, IMEC, Lund University, and KU-Leuven. The authors would like to thank ST Microelectronics for supplying 65nm process, ProfessorErik G. Larsson for discussion on MIMO detection, and Christian Mehlführer and the Christian Doppler Laboratory for Design Methodology of Signal Processing Algorithms at Vienna University of Technology, for contributions on the LTE simulation chain.

References

[1] C. Studer, M. Wenk, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: performance and implementation aspects," in *Proceedings of the 40th Asilomar Conference on Signals, Systems, and Computers (ACSSC '06)*, pp. 2071–2076, November 2006.

[2] M. Li, B. Bougard, W. Xu, D. Novo, L. Van Der Perre, and F. Catthoor, "Optimizing Near-ML MIMO detector for SDR baseband on parallel programmable architectures," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '08)*, pp. 444–449, March 2008.

[3] L. G. Barbero and J. S. Thompson, "Rapid prototyping of a fixed-throughput sphere decoder for MIMO systems," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, pp. 3082–3087, June 2006.

[4] E. G. Larsson and J. Jaldén, "Fixed-complexity soft MIMO detection via partial marginalization," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3397–3407, 2008.

[5] D. Wu, E. G. Larsson, and D. Liu, "Implementation aspects of fixed-complexity soft-output MIMO detection," in *Proceedings of the 69th IEEE Vehicular Technology Conference (VTC '09)*, April 2009.

[6] N. Moezzi-Madani, et al., "A low-area flexible MIMO detector for WiMAX/WiFi standards," in *Proceedings of the Conference*

on Design, Automation and Test in Europe (DATE '10), pp. 1637–1640, Dresden, Germany, March 2010.

[7] T. Cupaiuolo, et al., "Low-complexity high throughput VLSI architecture of soft-output ML MIMO detector," in *Proceedings of the IEEE Dessign, Test and Automation in Europe*, Dresden, Germany, March 2010.

[8] K. Amiri, J. R. Cavallaro, C. Dick, and R. M. Rao, "A high throughput configurable SDR detector for multi-user MIMO wireless systems," *Journal of Signal Processing Systems*. In press.

[9] D. Wu, J. Eilert, and D. Liu, "Evaluation of MIMO symbol detectors for 3GPP LTE terminals," in *Proceedings of the 17th European Signal Processing Conference (EUSIPCO '09)*, Glasgow, Scotland, 2009.

[10] J. Ketonen and M. Juntti, "SIC and K-best LSD receiver implementation for a MIMO-OFDM system," in *Proceedings of the 16th European Signal Processing Conference (EUSIPCO '08)*, August 2008.

[11] 3GPP, "Evolved Universal Terrestrial Radio Access (EUTRA): physical channels and modulation," Technical Specifications 36.211 V8.4.0, September 2008.

[12] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, 1998.

[13] M. Siti and M. P. Fitz, "A novel soft-output layered orthogonal lattice detector for multiple antenna communications," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, pp. 1686–1691, June 2006.

[14] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.

[15] D. Wu, J. Eilert, D. Liu, D. Wang, N. Al-Dhahir, and H. Minn, "Fast complex valued matrix inversion for multi-user STBC-MIMO decoding," in *Proceedings of the IEEE Computer Society Annual Symposium on VLSI: Emerging VLSI Technologies and Architectures (ISVLSI '07)*, pp. 325–330, March 2007.

[16] D. Wu, J. Eilert, and D. Liu, "Implementation of a high-speed MIMO soft-output symbol detector for software defined radio," *Journal of Signal Processing Systems*. In press.

[17] C. Mehlführer, S. Caban, and M. Rupp, "Experimental evaluation of adaptive modulation and coding in MIMO WiMAX with limited feedback," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, Article ID 837102, 2008.

[18] C. Mehlführer, M. Wrulich, J. C. Ikuno, D. Bosanska, and M. Rupp, "Simulating the long term evolution physical layer," in *Proceedings of the 17th European Signal Processing Conference (EUSIPCO '09)*, Glasgow, Scotland, 2009.

[19] D. S. Baum, J. Salo, M. Milojevic, P. Kyösti, and J. Hansen, "MATLAB implementation of the interim channel model forbeyond-3G systems (SCME)," May 2005.

[20] S. Chen, T. Zhang, and Y. Xin, "Relaxed K-best MIMO signal detector design and VLSI implementation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 3, pp. 328–337, 2007.

Preliminary call for papers

The 2011 European Signal Processing Conference (EUSIPCO-2011) is the nineteenth in a series of conferences promoted by the European Association for Signal Processing (EURASIP, www.eurasip.org). This year edition will take place in Barcelona, capital city of Catalonia (Spain), and will be jointly organized by the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) and the Universitat Politècnica de Catalunya (UPC).

EUSIPCO-2011 will focus on key aspects of signal processing theory and applications as listed below. Acceptance of submissions will be based on quality, relevance and originality. Accepted papers will be published in the EUSIPCO proceedings and presented during the conference. Paper submissions, proposals for tutorials and proposals for special sessions are invited in, but not limited to, the following areas of interest.

Areas of Interest

- Audio and electro-acoustics.
- Design, implementation, and applications of signal processing systems.
- Multimedia signal processing and coding.
- Image and multidimensional signal processing.
- Signal detection and estimation.
- Sensor array and multi-channel signal processing.
- Sensor fusion in networked systems.
- Signal processing for communications.
- Medical imaging and image analysis.
- Non-stationary, non-linear and non-Gaussian signal processing.

Submissions

Procedures to submit a paper and proposals for special sessions and tutorials will be detailed at www.eusipco2011.org. Submitted papers must be camera-ready, no more than 5 pages long, and conforming to the standard specified on the EUSIPCO 2011 web site. First authors who are registered students can participate in the best student paper competition.

Important Deadlines:

Proposals for special sessions	15 Dec 2010
Proposals for tutorials	18 Feb 2011
Electronic submission of full papers	**21 Feb 2011**
Notification of acceptance	23 May 2011
Submission of camera-ready papers	6 Jun 2011

Webpage: www.eusipco2011.org

Organizing Committee

Honorary Chair
Miguel A. Lagunas (CTTC)

General Chair
Ana I. Pérez-Neira (UPC)

General Vice-Chair
Carles Antón-Haro (CTTC)

Technical Program Chair
Xavier Mestre (CTTC)

Technical Program Co-Chairs
Javier Hernando (UPC)
Montserrat Pardàs (UPC)

Plenary Talks
Ferran Marqués (UPC)
Yonina Eldar (Technion)

Special Sessions
Ignacio Santamaría (Unversidad de Cantabria)
Mats Bengtsson (KTH)

Finances
Montserrat Nájar (UPC)

Tutorials
Daniel P. Palomar (Hong Kong UST)
Beatrice Pesquet-Popescu (ENST)

Publicity
Stephan Pfletschinger (CTTC)
Mònica Navarro (CTTC)

Publications
Antonio Pascual (UPC)
Carles Fernández (CTTC)

Industrial Liaison & Exhibits
Angeliki Alexiou (University of Piraeus)
Albert Sitjà (CTTC)

International Liaison
Ju Liu (Shandong University-China)
Jinhong Yuan (UNSW-Australia)
Tamas Sziranyi (SZTAKI -Hungary)
Rich Stern (CMU-USA)
Ricardo L. de Queiroz (UNB-Brazil)