Signal Detection for 3GPP LTE Downlink: Algorithm and Implementation

Huan Xuan Nguyen
School of Engineering and Information Sciences
Middlesex University
The Burroughs, London, NW4 4BT, United Kingdom
Email: H.Nguyen@mdx.ac.uk

***Abstract*—In this paper[1], we investigate an efficient signal detection algorithm, which combines lattice reduction (LR) and list decoding (LD) techniques for the 3rd generation long term evolution (LTE) downlink systems. The resulting detector, called LRLD based detector, is carried out within the framework of successive interference cancellation (SIC), which takes full advantages of the reliable LR detection. We then extend our studies to the implementation possibility of the LRLD based detector and provide reference for the possible real silicon implementation. Simulation results show that the proposed detector provides a near maximum likelihood (ML) performance with a significantly reduced complexity.**

***Index Terms*—3GPP LTE downlink, signal detection, lattice reduction, successive interference cancellation, implementation study.**

I. Introduction

The 3rd generation partnership project (3GPP) [2] is in the process of defining the long-term evolution (LTE) and Advanced-LTE for 3G radio access, in order to maintain the future competitiveness of 3G technology. The main targets for this evolution concern increased data rates, improved spectrum efficiency, improved coverage, and reduced latency. The LTE downlink is based on orthogonal frequency division multiple access (OFDMA) that allows multiple access on the same channel [3]. This allows simple receivers in case of large bandwidth, frequency selective scheduling and adaptive modulation and coding. The LTE uplink is based on single carrier frequency division multiple access (SC-FDMA) technique [4].

In order to fulfill the requirements on coverage, capacity, and high data rates, novel multiple input multiple output (MIMO) schemes need to be supported as part of the long-term 3G evolution. Signal detection in MIMO systems have recently drawn significant attention. If the maximum likelihood (ML) detection is used, the complexity grows exponentially with the number of transmit antennas. Thus, various approaches are devised to reduce the complexity. The successive interference cancellation (SIC) approach is employed in [5]. The relation between the SIC based MIMO detection and the decision feedback equalizer (DFE) is exploited in [6]. A probabilistic data association (PDA) algorithm, which was devised for the multiuser detection in [7], is applied to the MIMO detection in [8]. In [9], the partial maximum a posteriori probability (MAP) principle is derived to discuss the optimality of the SIC based detection. List decoding (LD) based detectors are also considered for the MIMO detection to obtain soft-decision in [10] and [11]. In [12], a lattice reduction (LR) based MIMO detector used as a low complexity MIMO detector is first discussed. In [13], more LR based MIMO detectors are proposed. Following this trend, this paper considers the signal detection in the LTE downlink, where an efficient signal detection algorithm based on the LR and LD techniques is investigated. The resulting detector (called LRLD detector) produces a list in the LR domain, which results in a much more reliable list and thus is efficient in mitigating error propagation when the SIC based detection is employed. Simulation results show that the LRLD detector provides a near ML performance with a significantly reduced complexity.

However, the potential capacity of the MIMO channel can only be exploited if implementable hardware architecture is available. The main issue in implementing the MIMO detector is the latency incurred by preprocessing the channel matrices

[1]This work was partly presented at the 2010 International Conference on Digital Communications (see reference [1].)

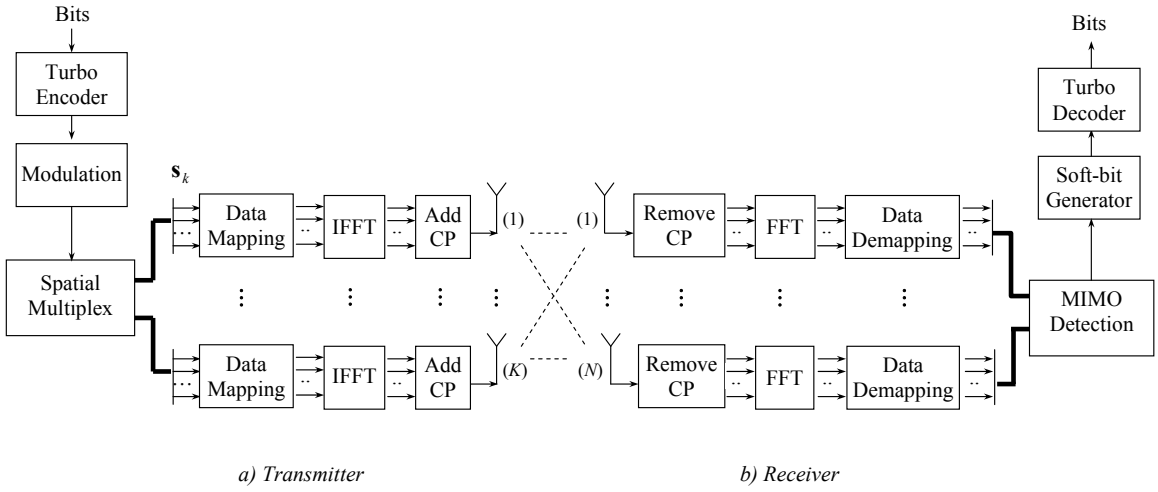

Fig. 1. Block diagram of a MIMO-OFDMA LTE downlink.

[14]. There have been extensive work on the implementation of the MIMO detection either with minimum mean square error-successive interference cancellation (MMSE-SIC) [15], vertical-Bell Laboratories layered space-time (V-BLAST) [16] or Maximum Likelihood (ML) receivers [17]-[22]. However, while the formers usually provide an inferior performance, the latter demandingly requires a large silicon complexity. Thus, finding a reasonable trade-off between an implementable architecture of the MIMO detector and a near ML performance is always a motivation. We therefore extend our studies to the implementation possibility of the proposed detector and then provide references for the possible real silicon implementation.

The rest of the paper is structured as follows. Section II describes the system and channel models. The signal detection algorithm is designed and discussed in Section III. Section IV studies the implementation possibility of the proposed detector. Section V provides simulation results and some concluding remarks are provided in Section VI.

Notation: Bold-face upper (lower) letters denote matrices (column vectors); $(\cdot)^*$, $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{\mathrm{H}}$ denote complex conjugation, transpose and Hermitian transpose, respectively; $\mathbf{I}$ is the identity matrix; $\mathrm{E}[\cdot]$ denotes statistical expectation; $\mathrm{Diag}(\mathbf{x})$ denotes a matrix with vector $\mathbf{x}$ being its diagonal; $\mathcal{N}(\mu, \sigma^2)$ denotes Gaussian distribution with mean μ and variance σ^2; $\delta_{n,n'}$ denotes Kronecker delta; $J_0(\cdot)$ denotes zero-order Bessel function of the first kind; $|\cdot|$ denotes absolute value; and $\|\cdot\|$ denotes Frobenius norm.

II. System and Channel Models

The MIMO-OFDMA LTE system is a parallel of single-input single-output OFDMA (SISO-OFDMA) where blocks of K data symbols are mapped onto the spatial multiplexing (SM) module followed by the data mapping and inverse fast Fourier transform (IFFT) operations, as shown in Figure 1. Note that we do not consider MIMO encoding (e.g., space-time coding) in this work. The data mapping operation is used for subcarrier mapping (e.g., distributed or localized mapping in multiple access [4]). Reversed operations are carried out at the receiver, which are then followed by the signal detection and MIMO processing. Assume that there are K transmit antennas and N receive antennas. Let P and Q denote the number of subcarriers used in one orthogonal frequency division multiplexing (OFDM) symbol for the user of interest and the size of the IFFT, respectively. We denote

$$\mathbf{s}_{P,k} = [s_{1,k}, s_{2,k}, \cdots, s_{P,k}]^{\mathrm{T}} \tag{1}$$

as the transmitted signal vector from the kth transmit antenna. For convenience, it is assumed that $E[s_{p,k}s_{p,k}^*] = 1$ for $1 \leq p \leq P, 1 \leq k \leq K$.

Assuming that the guard interval (i.e., cyclic prefix (CP)) is longer than the maximum channel span, the received signal vector after removing CP and taking fast Fourier transform (FFT) at the nth receive antenna can be written as

$$\mathbf{r}_{P,n} \triangleq [r_{1,n}, r_{2,n}, \cdots, r_{P,n}]^{\mathrm{T}} \tag{2}$$

$$= \sum_{k=1}^{K} \mathrm{Diag}(\mathbf{h}_{n,k})\mathbf{s}_{P,k} + \mathbf{w}_n \tag{3}$$

where $\mathbf{h}_{n,k} = [h_{n,k}(i_1), h_{n,k}(i_2), \cdots, h_{n,k}(i_P)]^T$ is the frequency-domain channel vector from the kth transmit antenna to the nth receive antenna and $\mathbf{w}_n$ is a zero-mean complex Gaussian vector with variance σ_w^2. Here, $i_p = \mathcal{P}(p)$ where $\mathcal{P}(\cdot)$ is the subcarrier mapping function that maps a data symbol onto one of the Q subcarriers. Obviously, i_p is obtained depending on the subcarrier mapping pattern and $i_p \in \{1, 2, \cdots, Q\}$. Note that

$$h_{n,k}(i_p) = \sum_{l=1}^{L} g_{n,k}(l) e^{-\frac{2\pi j}{Q}(l-1)(i_p-1)}$$

where $g_{n,k}(l)$ is the lth tap of the fading channel from kth transmit antenna to the nth receive antenna and L is the number of paths. We can rewrite the received signal for each subcarrier as follow

$$\mathbf{r}_{p,N} = \mathbf{H}(i_p)\mathbf{s}_{p,K} + \mathbf{w}_p \tag{4}$$

where $\mathbf{r}_{p,N} = [r_{p,1}, r_{p,2}, ..., r_{p,N}]^T, p = 0, 1, ..., P-1$, is the signal vector at the i_pth subcarrier received through the N receive antennas. $\mathbf{s}_{p,K}$= $[s_{p,1}, s_{p,2}, ..., s_{p,K}]^T$ is the data symbol vector at the i_pth subcarrier transmitted through K transmit antennas. $\mathbf{w}_p$ is also the complex Gaussian noise vector. $\mathbf{H}(i_p)$ is the frequency-domain channel matrix at the i_pth subcarrier given as

$$\mathbf{H}(i_p) = \begin{pmatrix} h_{1,1}(i_p) & h_{1,2}(i_p) & \cdots & h_{1,K}(i_p) \\ h_{2,1}(i_p) & h_{2,2}(i_p) & \cdots & h_{2,K}(i_p) \\ \vdots & \vdots & \ddots & \vdots \\ h_{N,1}(i_p) & h_{N,2}(i_p) & \cdots & h_{N,K}(i_p) \end{pmatrix}. \tag{5}$$

We assume that the channel is unchanged during one OFDM symbol interval and $g_{n,k}(l)$ is independent and has identical Gaussian distribution $g_{n,k}(l) \sim \mathcal{N}(0, \sigma_l^2)$. Here, σ_l^2 is the normalized average power of each propagation path with

$$\sum_{l=0}^{L-1} \sigma_l^2 = 1. \tag{6}$$

Typical urban (TU) [23] and spatial channel model (SCM) [24] power delay profiles are used in this paper.

1) Typical Urban: We consider the time varying channel whose channel impulse response (CIR) is modeled by L propagation paths,

$$g(\tau, t) = \sum_{l=0}^{L-1} \gamma_l(t)\delta(\tau - \tau_l). \tag{7}$$

Assume that the channel is a wide-sense stationary uncorrelated scattering (WSSUS) Rayleigh fading and unchanged during one OFDM symbol interval. The maximum channel impulse span is also assumed to be within the guard interval. For convenience, let $\tau_l = lT_s$, $T_b = T + T_g$ where $T_s = T/Q$. Here, T, T_b and T_g denote the useful OFDM symbol interval, the whole OFDM symbol interval and the guard interval, respectively. Then, the channel impulse vector at each (OFDM symbol) time index n, denoted by $\mathbf{g}(t) = [g_0(t), g_1(t), ..., g_{L-1}(t)]^T$, can represent the discrete CIR. The autocorrelation function of $g_l(t) = g(lT_s, tT_b)$ is expressed as

$$E\{g_l(t)g_{l'}^*(t')\} = \sigma_l^2 J_0(2\pi f_D(t-t')T_b)\delta_{l,l'}, \tag{8}$$

where f_D is the maximum Doppler frequency and σ_l^2 is the normalized average power of each propagation tap with

$$\sum_{l=0}^{L-1} \sigma_l^2 = 1. \tag{9}$$

An typical urban (TU) power delay profile [23] is used to model $\{\sigma_l^2\}$.

2) Spatial Channel Model: SCM was proposed by the 3GPP for both link- and system-level simulations. The 3GPP SCM emulates the double-directional and clustering effects of small scale fading mechanisms in a variety of environments, such as suburban macrocell, urban macrocell, and urban microcell. It considers N clusters of scatterers. A cluster can be considered as a resolvable path. Within a resolvable path (cluster), there are M subpaths which are regarded as the unresolvable rays. A simplified plot of the SCM is given in Figure 2, where only one cluster of scatterers is shown as an example. Here, θ_v is the angle of the mobile station (MS) velocity vector with respect to the MS broadside, $\theta_{n,m,AoD}$ is the absolute angle of departure (AoD) for the mth ($m = 1, ..., M$) subpath of the nth ($n = 1, ..., N$) path at the base station (BS) with respect to the BS broadside, and $\theta_{n,m,AoA}$ is the absolute angle of arrival (AoA) for the mth subpath of the nth path at the MS with respect to the MS broadside. Details of the generation of SCM simulation parameters can be found in [24].

III. Signal Detection

For convenience, the indices in (4) are omitted. The $N \times 1$ received signal vector $\mathbf{r}_{p,N}$, now denoted by $\mathbf{r}$, is given by

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \mathbf{w}, \tag{10}$$

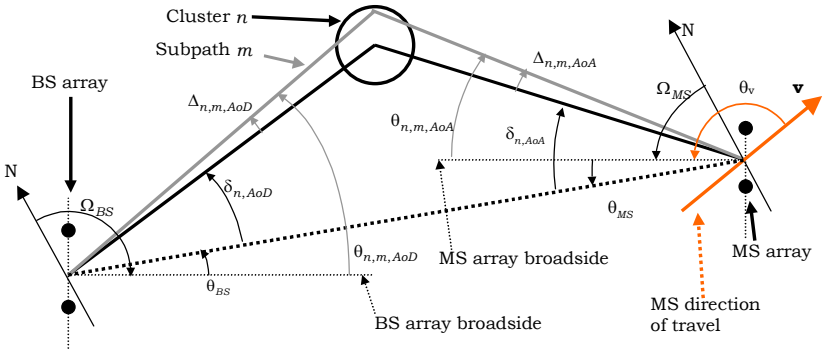

Fig. 2. BS and MS angle parameters in the 3GPP SCM with one cluster of scatterers [24].

where $\mathbf{H}$, $\mathbf{s}$, and $\mathbf{w}$ are the $N \times K$ channel matrix, the $K \times 1$ transmitted signal vector, and the $N \times 1$ noise vector, respectively. Let $\mathcal{S}$ denote the signal alphabet for symbols, i.e., $s_k \in \mathcal{S}$, where s_k denotes the kth element of $\mathbf{s}$, and its size is denoted by M, i.e., $M = |\mathcal{S}|$.

A. Conventional Detectors

We consider two conventional detection approaches: ML and MMSE.

1) ML Detection: The ML detection finds the data symbol vector that maximizes the likelihood function as follows:

$$\begin{aligned} \mathbf{s}_{\rm ml} &= \arg\max_{\mathbf{s}\in\mathcal{S}^K} f(\mathbf{r}|\mathbf{s}) \\ &= \arg\min_{\mathbf{s}\in\mathcal{S}^K} ||\mathbf{r}-\mathbf{H}\mathbf{s}||^2. \end{aligned} \quad (11)$$

To identify the ML vector, an exhaustive search is required. Because the number of candidate vectors for $\mathbf{s}$ is M^K, the complexity grows exponentially with K.

If the a priori probability of $\mathbf{s}$ is available, the maximum a posteriori (MAP) sequence detection can be formulated. Suppose that $\mathbf{b}$ is a bit-level symbol vector representation of $\mathbf{s}$. The elements of $\mathbf{b}$ are binary and the size of $\mathbf{b}$ is $(K \log_2 M) \times 1$. With the a priori probability of $\mathbf{b}$, the MAP vector (at the bit-level) becomes

$$\begin{aligned} \mathbf{b}_{\rm map} &= \arg\max_{\mathbf{b}} \Pr(\mathbf{b}|\mathbf{r}) \\ &= \arg\max_{\mathbf{b}} f(\mathbf{r}|\mathbf{b})\Pr(\mathbf{b}), \end{aligned} \quad (12)$$

where $\Pr(\mathbf{b})$ denotes the a priori probability of $\mathbf{b}$. In addition, the a posteriori probability of each bit can be found by marginalization as

$$\Pr(b_i = +1|\mathbf{r}) = \sum_{\mathbf{b}\in\mathcal{B}_i^+} \Pr(\mathbf{b}|\mathbf{r})$$

$$\Pr(b_i = -1|\mathbf{r}) = \sum_{\mathbf{b}\in\mathcal{B}_i^-} \Pr(\mathbf{b}|\mathbf{r}), \quad (13)$$

where $\mathcal{B}_i^{\pm} = \{[b_1 \ b_2 \ \ldots \ b_{\bar{K}}]^{\rm T} \mid b_i = \pm 1, b_m \in \{+1,-1\}, \ \forall m \neq i\}$ and $\bar{K} = K \log_2 M$.

2) MMSE Detection: It is easy to perform the (linear) MMSE detection if the constraint on the symbol vector, $s_k \in \mathcal{S}$, $\forall k$, is not imposed. Using the orthogonality principle, the MMSE estimator for $\mathbf{s}$ can be found as

$$\begin{aligned} \mathbf{W}_{\rm mmse} &= \arg\min_{\mathbf{W}} E[||\mathbf{s}-\mathbf{W}^{\rm H}\mathbf{r}||^2] \\ &= \left(E[\mathbf{r}\mathbf{r}^{\rm H}]\right)^{-1} E[\mathbf{r}\mathbf{s}^{\rm H}]. \end{aligned} \quad (14)$$

We can show that

$$\begin{aligned} E[\mathbf{r}\mathbf{r}^{\rm H}] &= \mathbf{H}\mathbf{H}^{\rm H} + \sigma_w^2\mathbf{I} \\ E[\mathbf{r}\mathbf{s}^{\rm H}] &= \mathbf{H}. \end{aligned}$$

It follows that

$$\mathbf{W}_{\rm mmse} = (\mathbf{H}\mathbf{H}^{\rm H} + \sigma_w^2\mathbf{I})^{-1}\mathbf{H}$$

and

$$\begin{aligned} \hat{\mathbf{s}}_{\rm mmse} &= \mathbf{W}_{\rm mmse}^{\rm H}\mathbf{r} \\ &= \mathbf{H}^{\rm H}(\mathbf{H}\mathbf{H}^{\rm H} + \sigma_w^2\mathbf{I})^{-1}\mathbf{r}. \end{aligned} \quad (15)$$

B. Proposed Detector

We assume that $N \geq K$ and consider the QR factorization of the channel matrix as $\mathbf{H} = \mathbf{Q}\mathbf{R}$, where $\mathbf{Q}$ is unitary and $\mathbf{R}$ is upper triangle. We have

$$\mathbf{x} = \mathbf{Q}^{\rm H}\mathbf{r} = \mathbf{R}\mathbf{s} + \mathbf{Q}^{\rm H}\mathbf{w}. \quad (16)$$

Since the statistical properties of $\mathbf{Q}^{\rm H}\mathbf{w}$ are identical to that of $\mathbf{w}$, $\mathbf{Q}^{\rm H}\mathbf{w}$ will be denoted by $\mathbf{w}$. If $N = K$, there is no zero rows in $\mathbf{R}$, otherwise the last $N - K$ rows would be zero. Thus, the last $N - K$ elements of $\mathbf{x}$ would be ignored

for the detection if $N > K$. Accordingly, the first K rows of $\mathbf{R}$ would be considered. If there is no risk of confusion, hereafter, we assume that the sizes of $\mathbf{x}$, $\mathbf{R}$, and $\mathbf{w}$ are $K \times 1$, $K \times K$, and $K \times 1$, respectively.

The complexity of the conventional LR based detector can grows significantly with the number of basis vectors. To avoid this problem, we propose an LRLD based detection algorithm, which breaks a high dimensional MIMO detection problem into multiple lower dimensional MIMO sub-detection problems.

To perform the proposed LRLD based detection, we consider the partition of $\mathbf{x}$ as follows:

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_3 \\ \mathbf{0} & \mathbf{R}_2 \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}, \qquad (17)$$

where $\mathbf{x}_i$, $\mathbf{s}_i$, and $\mathbf{w}_i$ denote the $K_i \times 1$ ith subvectors of $\mathbf{x}$, $\mathbf{s}$, and $\mathbf{w}$, $i = 1, 2$, respectively. Note that $K_1 + K_2 = K$. From (17), we can have two lower dimensional MIMO sub-detection problems to detect $\mathbf{s}_1$ and $\mathbf{s}_2$. It is straightforward to extend the partition into more than two groups. However, for the sake of simplicity, we only consider the partition into two groups as in (17).

In the proposed LRLD based detection, the sub-detection of $\mathbf{s}_2$ is carried out first using the LR based detector. Then, a list of candidate vectors of $\mathbf{s}_2$ is generated. With the list of $\mathbf{s}_2$, the sub-detection of $\mathbf{s}_1$ is performed with the LR based detector. The candidate vector in the list is used for the SIC to mitigate the interference from $\mathbf{s}_2$. The algorithm steps (AS) of the proposed LRLD based detector is summarized as follows.

AS1) The LR based detection of $\mathbf{s}_2$ is performed with the received signal $\mathbf{x}_2$, i.e.,

$$\tilde{\mathbf{c}}_2 = \text{LRDet}(\mathbf{x}_2),$$

where LRDet$(\cdot)$ is the function of the LR detection operation (see Appendix A for details of the LR detection), and $\tilde{\mathbf{c}}_2$ is the estimated vector of $\mathbf{s}_2$ in the corresponding LR domain. Note that there is no interference from $\mathbf{s}_1$ in detecting $\mathbf{s}_2$.

AS2) A list of candidate vectors in the lattice-reduced domain is generated by

$$\mathcal{C}_2 = \text{List}(\tilde{\mathbf{c}}_2),$$

where List is a function that chooses the Q closest vectors to $\tilde{\mathbf{c}}_2 (1 \le Q \le M^{K_2})$ in the LR domain. The details of the list generation is discussed in Appendix B.

AS3) The list of candidates of $\mathbf{s}_2$, denoted by $\mathcal{S}_2$, can be converted from $\mathcal{C}_2$. For convenience, denote $\mathcal{S}_2 = \{\tilde{\mathbf{s}}_2^{(1)}, \tilde{\mathbf{s}}_2^{(2)}, \cdots, \tilde{\mathbf{s}}_2^{(Q)}\}$.

AS4) Once $\mathcal{S}_2$ is available, the LR-based detection of $\mathbf{s}_1$ can be carried out with SIC, i.e.,

$$\tilde{\mathbf{c}}_1^{(q)} = \text{LRDet}(\mathbf{x}_1 - \mathbf{R}_3 \tilde{\mathbf{s}}_2^{(q)}),$$

where $\tilde{\mathbf{s}}_2^{(q)}$ is the qth decision vector of $\mathbf{s}_2$ from list $\mathcal{S}_2$.

AS5) Let $\tilde{\mathbf{s}}_1^{(q)}$ denote the signal vector corresponding to $\tilde{\mathbf{c}}_1^{(q)}$ in the LR domain and $\tilde{\mathbf{s}}^{(q)} = [(\tilde{\mathbf{s}}_1^{(q)})^{\mathrm{T}} \ (\tilde{\mathbf{s}}_2^{(q)})^{\mathrm{T}}]^{\mathrm{T}}$, the final decision of $\mathbf{s}$ is found as

$$\tilde{\mathbf{s}} = \arg \min_{q=1,2,\cdots,Q} \left\| \mathbf{x} - \mathbf{R}\tilde{\mathbf{s}}^{(q)} \right\|^2.$$

Softbit Generation: As we are using turbo code for channel coding, its inputs should be soft bits. The probability of the qth candidate $\hat{\mathbf{s}}^{(q)}$ in the list can be found as

$$P(\hat{\mathbf{s}}^{(q)}) = C_Q \exp\left(-\frac{1}{\sigma_w^2} ||\mathbf{x} - \mathbf{R}\hat{\mathbf{s}}^{(q)}||^2 \right), \qquad (18)$$

where C_Q is the normalization constant, which is given by

$$C_Q = \frac{1}{\sum_{q=1,\cdots,Q} \exp\left(-\frac{1}{\sigma_w^2} ||\mathbf{x} - \mathbf{R}\hat{\mathbf{s}}^{(q)}||^2 \right)}.$$

Note that

$$\sum_{q=1,\cdots,Q} P(\hat{\mathbf{s}}^{(q)}) = 1. \qquad (19)$$

Suppose that $\hat{\mathbf{b}}^{(q)}$ is a bit-level symbol vector representation of $\hat{\mathbf{s}}^{(q)}$, i.e., $\hat{\mathbf{s}}^{(q)} = \mathcal{M}(\hat{\mathbf{b}}^{(q)})$ where $\mathcal{M}(\cdot)$ denotes the mapping rule. The elements of $\hat{\mathbf{b}}^{(q)}$ are binary and the size of $\hat{\mathbf{b}}^{(q)}$ is $\bar{K} \times 1$ where $\bar{K} = K \log_2 M$. Correspondingly, the probability of $\hat{\mathbf{b}}^{(q)}$ can be written as

$$P(\hat{\mathbf{b}}^{(q)}) = C_Q \exp\left(-\frac{1}{\sigma_w^2} ||\mathbf{x} - \mathbf{R}\mathcal{M}(\hat{\mathbf{b}}^{(q)})||^2 \right), \qquad (20)$$

The soft log-likelihood ratio (LLR) value of the ith bit b_i $(i = 1, 2, \cdots, \bar{K})$ can then be obtained as

$$\Lambda(b_i) = \log \frac{\sum_{\hat{\mathbf{b}}^{(q)} \in \mathcal{B}_i^+} P(\hat{\mathbf{b}}^{(q)})}{\sum_{\hat{\mathbf{b}}^{(q)} \in \mathcal{B}_i^-} P(\hat{\mathbf{b}}^{(q)})}, \qquad (21)$$

where $\mathcal{B}_i^{\pm} = \{[b_1 \ b_2 \ \ldots \ b_{\bar{K}}]^{\mathrm{T}} \mid b_i = \pm 1, b_m \in \{+1, -1\}, \ \forall m \neq i\}$.

IV. Implementation Study of the Proposed Detector

In this section, we study the implementation possibility of the proposed LRLD detector. Note that some details of the proposed detector and definition of certain parameters, e.g., α, β, are presented in Appendix A and B.

A. Detector Structure

For convenience, we outline the implementation steps (IS) required for the proposed detector as follows.

IS1) *QR decomposition*:

$$\mathbf{H} = \mathbf{QR},$$

where

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_3 \\ \mathbf{0} & \mathbf{R}_2 \end{bmatrix}.$$

IS2) *Gaussian lattice reduction*:

$$\bar{\mathbf{R}}_1 = \mathbf{R}_1\mathbf{U}_1,$$

$$\bar{\mathbf{R}}_2 = \mathbf{R}_2\mathbf{U}_2.$$

IS3) *MMSE filtering weight matrices*:

$$\mathbf{W}_1 = (\mathbf{R}_1\mathbf{R}_1^{\mathrm{H}}\alpha^2 E_s + |\alpha|^2\sigma_w^2\mathbf{I})^{-1}\mathbf{R}_1\mathbf{U}_1^{-\mathrm{H}}\alpha^2 E_s,$$
$$\mathbf{W}_2 = (\mathbf{R}_2\mathbf{R}_2^{\mathrm{H}}\alpha^2 E_s + |\alpha|^2\sigma_w^2\mathbf{I})^{-1}\mathbf{R}_2\mathbf{U}_2^{-\mathrm{H}}\alpha^2 E_s.$$

IS4) *Unitary transformation*:

$$\begin{aligned}\mathbf{x} &= \mathbf{Q}^{\mathrm{H}}\mathbf{r} \\ &= \mathbf{Rs} + \mathbf{w},\end{aligned}$$

or

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_3 \\ \mathbf{0} & \mathbf{R}_2 \end{bmatrix}\begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}.$$

IS5) *Scaling/shifting*:

$$\begin{aligned}\mathbf{d}_2 &= \alpha\mathbf{x}_2 + \beta\mathbf{R}_2\mathbf{1}, \\ \mathbf{b}_2 &= \alpha\mathbf{s}_2 + \beta\mathbf{1}, \\ \mathbf{d}_1^{(q)} &= \alpha(\mathbf{x}_1 - \mathbf{R}_3\hat{\mathbf{s}}_2^{(q)}) + \beta\mathbf{R}_1\mathbf{1}, \\ \mathbf{b}_1 &= \alpha\mathbf{s}_1 + \beta\mathbf{1}.\end{aligned}$$

IS6) *LR based list detection*: This step includes three stages:

- one MMSE filtering operation to estimate $\mathbf{c}_2$ (i.e., signal vector $\mathbf{s}_2$ in the LR domain):

$$\begin{aligned}\tilde{\mathbf{c}}_2 &= \mathbf{W}_2^H(\mathbf{d}_2 - \beta\mathbf{R}_2\mathbf{1}) + \mathbf{U}_2^{-1}\beta\mathbf{1} \\ &= \alpha\mathbf{W}_2^H\mathbf{x}_2 + \mathbf{U}_2^{-1}\beta\mathbf{1}.\end{aligned}$$

- sorting and storing the list of $\mathbf{c}_2$ (of length Q):

$$\mathcal{C}_2 = \{\mathbf{c}_2 \mid ||\tilde{\mathbf{c}}_2 - \mathbf{c}_2|| < r(Q)\}.$$

- Q parallel MMSE filtering operations to estimate $\mathbf{c}_1$ with respect to each candidate of the list of $\mathbf{c}_2$:

$$\tilde{\mathbf{c}}_1^{(q)} = \alpha\mathbf{W}_1^H(\mathbf{x}_1 - \mathbf{R}_3\hat{\mathbf{s}}_2^{(q)}) + \mathbf{U}_1^{-1}\beta\mathbf{1},$$

where $\mathbf{s}_2^{(q)} = (\mathbf{U}_2\mathbf{c}_2^{(q)} - \beta\mathbf{1})/\alpha$ and $\mathbf{c}_2^{(q)} \in \mathcal{C}_2$.

The implementation operations can be classified into two types: Pre-processing and detection processing.

Pre-processing: This is often referred to as channel-rate processing, in which all operations need to be carried out only when there is a new channel update. All steps from IS1) to IS3) belong to this type.

Detection Processing: This can be referred to as symbol-rate processing. This type of processing includes all operations that are carried out after each received signal vector arrives. In our proposed detector, the received data will be processed in a first in first out (FIFO) manner. The FIFO buffer is used to bridge the latency incurred among the received signals. All steps from IS4) to IS6) belong to this type.

Figure 3 shows a high-level structure of the proposed detector with respect to hardware implementation. We will describe each major operation next. Some operations such as unitary transformation, shifting/scaling and final decision are straightforward and thus ignored. Since memory is nowadays not a big issue in the hardware implementation, we assume that a certain amount of memory is available wherever needed.

B. Pre-Processing

In our proposed detector, there are three dominant components in the pre-processing stage – QR decomposition, Gaussian lattice reduction and matrix inversion operations. It is always desirable to have a low latency in preprocessing the channel matrices. Thus, selection of algorithm to be implemented for each of the three above operations may well decide the real silicon complexity. We will consider each operation in details next.

1) QR Decomposition: As shown in [25], QR decomposition is preferred to Cholesky decomposition due to the numerical stability. In our detection algorithm, although the QR operation is required only once for each channel update, it still provides a significant load of computations as the operation is carried out to the channel matrix of full size. We therefore study different algorithms in the literature for the QR decomposition.

Gram-Schmidt:

The Gram-Schmidt (GS) procedure finds the QR decomposition of a matrix $\mathbf{H}$ such that $\mathbf{H} = \mathbf{QR}$, where $\mathbf{Q}$ is unitary and $\mathbf{R}$ is upper triangular. An obvious drawback of

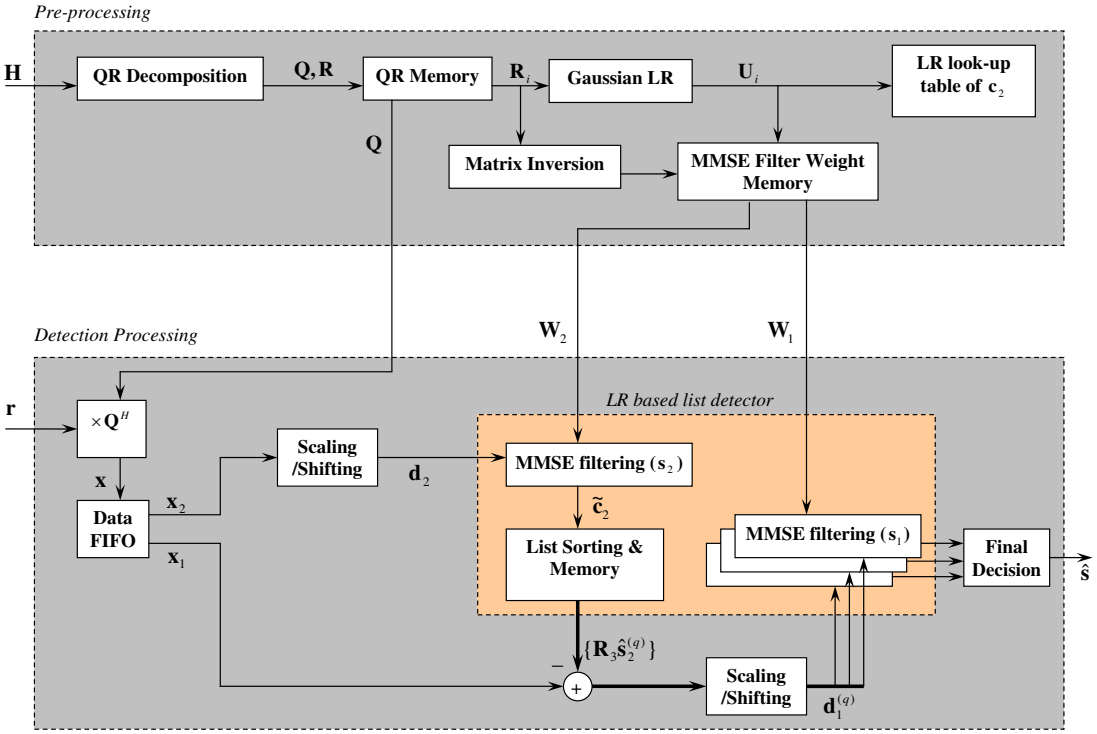

Fig. 3. High-level structure diagram of the implementation of the proposed LR based list detector.

the GS algorithm is the fact that it requires costly square-root and division operations and that the overall computational complexity is high. Thus, a modified version of the GS is presented (see [26]). The details of the modified GS are discussed in [27], [28]. The corresponding algorithm proceeds as follows.

Gram-Schmidt algorithm:

1) `initialize:` $\mathbf{Q} = \mathbf{H}, \mathbf{R} = \mathbf{0}$
2) `for` $k = 1$ `to` K
3) $\quad [\mathbf{R}]_{k,k} = \sqrt{\mathbf{q}_k^{\mathrm{H}} \mathbf{q}_k}$
4) $\quad \mathbf{q}_k = \mathbf{q}_k / [\mathbf{R}]_{k,k}$
5) $\quad$ `for` $i = k+1$ `to` K
6) $\quad\quad [\mathbf{R}]_{k,i} = \mathbf{q}_k^{\mathrm{H}} \mathbf{q}_i$
7) $\quad\quad \mathbf{q}_i = \mathbf{q}_i - [\mathbf{R}]_{k,i} \mathbf{q}_k$
8) $\quad$ `end for`
9) `end for`

Generally, the GS is accurate to the floating-point precision. For fixed-point arithmetic, the problem of quantization and round-off errors is not ignorable and therefore there is loss in accuracy (e.g., loss in the orthogonality of $\mathbf{Q}$) [27]. It was shown in [29] that the orthogonalization error (ϵ_o) in fixed-point version of the GS algorithm is bounded by the product of condition number $\kappa(\mathbf{H})$ of matrix $\mathbf{H}$ and machine precision ε, as follows

$$\begin{aligned} \epsilon_o &= \| \mathbf{I} - \mathbf{Q}^{\mathrm{H}} \mathbf{Q} \| \\ &\leq \zeta(K) \times \varepsilon \times \kappa(\mathbf{H}), \end{aligned}$$

where $\zeta(K)$ is a low degree polynomial in K depending only on details of computer arithmetic. This implies that for a well-conditioned matrix, fixed-point architecture for the GS is still accurate to the integer multiples of the machine precision ε. However, for ill-conditioned matrices, the computed $\mathbf{Q}$ can be very far from orthogonal. Thus, we can consider the numerically more favorable scheme, Householder Transformation, which is based on unitary transformation.

Householder Transformation:

The use of unitary transformations instead of the conventional methods is to alleviate the numerical problem such as requirement of high number precision, i.e., large silicon area in fixed-point very-large-scale integration (VLSI) implementation is required. The reason for this more favorable behavior is that unitary transformations do not alter the length of a vector

and thus cannot lead to an excessive increase in dynamic range or to an enhancement of quantization noise. Two typical algorithms using unitary transformations are Householder Transformation and Givens Rotation. For illustrative purpose, we overview the Householder Reflection algorithm only.

The Householder Transformation algorithm recursively applies a sequence of unitary transformations $\mathbf{Q}_i^{\mathrm{H}}$ to matrix $\mathbf{H}$ as follows:

$$\mathbf{R}^{(k+1)} = \mathbf{Q}_k^{\mathrm{H}}\mathbf{R}^{(k)},$$

where $\mathbf{R}^{(1)} = \mathbf{H}$. Each transformation will eliminate more subdiagonal entries until finally $\mathbf{R} = \mathbf{R}^{(K-1)} = \mathbf{Q}_{K-1}^{\mathrm{H}} \cdots \mathbf{Q}_1^{\mathrm{H}}\mathbf{H}$. The unitary matrix $\mathbf{Q}^{\mathrm{H}}$ is readily obtained from

$$\mathbf{Q}^{\mathrm{H}} = \mathbf{Q}_{K-1}^{\mathrm{H}} \cdots \mathbf{Q}_1^{\mathrm{H}}.$$

The algorithm can be described in details as follows.

Householder Transformation algorithm:

1) `initialize:` $\mathbf{Q}^{(0)} = \mathbf{I}, \mathbf{R}^{(1)} = \mathbf{H}$
2) `for` $k = 1$ `to` $K-1$
3) $\bar{\mathbf{q}}_k = \mathbf{r}_k + \| \mathbf{r}_k \| \mathbf{1}$
4) $\bar{\mathbf{Q}}_k = \mathbf{I} - 2\frac{\bar{\mathbf{q}}_k \bar{\mathbf{q}}_k^{\mathrm{H}}}{\|\bar{\mathbf{q}}_k\|^2}$
5) $\mathbf{P}_k = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{Q}}_k \end{bmatrix}$
6) $[\mathbf{R}]_{k+1}^{H} = \mathbf{P}_k\mathbf{R}^{(k)}$
7) $\mathbf{Q}^{(k)} = \mathbf{P}_k\mathbf{Q}^{(k-1)}$
8) `end for`
9) $\mathbf{Q}^{\mathrm{H}} = \mathbf{Q}^{(K-1)}$

We compare the complexity of the two methods in Table I. The Householder Reflection algorithm provides a slightly lower number of complex multiplications (CMs), divisions and square root operations compared to the Gram-Schmidt algorithm. In addition, for fixed-point implementation, the Householder Reflection algorithm is supposed to be more stable.

Note that $(K^2 + K(K+1)/2)$ words of memory[2] are required to store matrices $\mathbf{Q}$ and $\mathbf{R}$ at the output of the QR decomposition operation.

2) Lattice Reduction Using Gaussian Method: In the proposed LR based list detector, the LR is applied to the sub-channel matrix $\mathbf{R}_1$ and $\mathbf{R}_2$. For convenience, we consider these matrices of size 2×2 only. Thus, this basis-2 LR can be carried out using the simple Gaussian method. We can limit the maximum number of iterations in this Gaussian lattice reduction algorithm to a small number (e.g., 2 iterations is reasonable) while keeping the overall performance almost the same. For the implementation purpose, we can fix the maximum number of iterations to T, and the Gaussian LR algorithm is summarized as follows.

1) `Input` $(\mathbf{b}_1, \mathbf{b}_2, T)$
2) `Set` $\mathbf{J} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ `and` $\mathbf{U} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
3) i = 0
4) `do`
5) `if` $||\mathbf{b}_1|| > ||\mathbf{b}_2||$
6) `swap` $\mathbf{b}_1$ `and` $\mathbf{b}_2$, `and` $\mathbf{U} = \mathbf{U}\mathbf{J}$
7) `end if`
8) `if` $|< \mathbf{b}_2, \mathbf{b}_1 >| > 1/2$
9) $\hat{t} = \lfloor \frac{< \mathbf{b}_2, \mathbf{b}_1 >}{||\mathbf{b}_1||^2} \rceil$
10) $\mathbf{b}_2 = \mathbf{b}_2 - \hat{t}\mathbf{b}_1$ and $\mathbf{U} = \mathbf{U}\begin{bmatrix} 1 & -t \\ 0 & 1 \end{bmatrix}$
11) `end if`
12) $i = i + 1$
13) `while` $(||\mathbf{b}_1|| < ||\mathbf{b}_2||)\&\&(i \leq T)$
14) `return` $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{U})$

In the worst case where the Gaussian LR algorithm runs until the maximum iteration $i = T$, the number of CMs required for the Gaussian LR is $4T$. Six words of memory are required to store data of the unimodular matrix at the output.

3) Matrix Inversion: In our proposed detector, the dominant complexity component in obtaining the MMSE filtering weight matrices is the matrix inversion operations, $(\mathbf{R}_1\mathbf{R}_1^{\mathrm{H}}\alpha^2 E_s + |\alpha|^2\sigma_w^2\mathbf{I})^{-1}$ and $(\mathbf{R}_2\mathbf{R}_2^{\mathrm{H}}\alpha^2 E_s + |\alpha|^2\sigma_w^2\mathbf{I})^{-1}$. Fortunately, the fact that the size of these submatrices to be inverted is reasonably small leads to a reasonably low load of computations. For example, a 2×2 matrix $\mathbf{R} = \begin{bmatrix} r_{1,1} & r_{1,2} \\ r_{2,1} & r_{2,2} \end{bmatrix}$ can be simply inverted using adjoint method

$$\mathbf{R}^{-1} = \frac{1}{r_{1,2}r_{2,1} - r_{1,1}r_{1,1}} \begin{bmatrix} r_{2,2} & -r_{2,1} \\ -r_{1,2} & r_{1,1} \end{bmatrix},$$

which requires 1 division and 6 CMs.

In a general case of matrix $\mathbf{H}$ of size $K \times K$, the complexity of inversion operation may vary depending on implementation method. We overview some typical methods:

[2]The term 'word of memory' is referred to the amount of memory required to store one complex number. The number of bits in one word may vary depending on the dynamic range of the observing data. Thus, throughout the section, we use 'word' as a unit of memory.

TABLE I
COMPLEXITY COMPARISON OF THE TWO METHODS: GRAM-SCHMIDT (GS) AND HOUSEHOLDER REFLECTION (HR)

Algorithm	**Division**	**Square root**	**Complex multiplications (CMs)**	**CMs with $K = 4$**
GS	K	K	$2K^2 + 2\sum_{k=1}^{K} K(K-k)$	80
HR	$K-1$	$K-1$	$2\sum_{k=1}^{K-1}(K-k+1)^2$	78

a) Adjoint Method:

$$\mathbf{H}^{-1} = \frac{\text{adj}(\mathbf{H})}{\det(\mathbf{H})}.$$

Unfortunately, for the matrix inversion using adjoint method, there is no generic expression for the number of CMs as it depends heavily on the dimension K. However, the approximated number of CMs can be of up to scale in 2^K as [30]

$$\mathcal{C}_m \approx a2^K + K^2 + K.$$

b) LR Decomposition: Matrix $\mathbf{H}$ is decomposed into a lower-triangular matrix $\mathbf{L}$ and a upper-triangular matrix $\mathbf{R}$, i.e., $\mathbf{H}^{-1} = \mathbf{R}^{-1}\mathbf{L}^{-1}$. The algorithm is as follows

1) Initiate $\mathbf{L} = \mathbf{H}, \mathbf{R} = \mathbf{I}$
2) For $i = 1$ to K
3) For $j = 1$ to K
4) $[\mathbf{R}]_{j,i} = [\mathbf{L}]_{j,i} - \sum_{k=1}^{j-1}[\mathbf{L}]_{j,k}[\mathbf{R}]_{k,j}$
5) $[\mathbf{L}]_{j,i} = \frac{[\mathbf{R}]_{j,i}}{[\mathbf{R}]_{i,i}}$
6) end for
7) end for

The number of CMs for matrix inversion using LR decomposition is $4(K^3 - K)/3$.

c) QR Decomposition: Matrix $\mathbf{H}$ can be inverted using QR decomposition as $\mathbf{H}^{-1} = \mathbf{R}^{-1}\mathbf{Q}^H$. If Gram-Schmidt algorithm is used for QR decomposition, the total number of CMs required for matrix inversion is $(9K^3 + 10K^2 - K)/6$.

In general, a major concern with matrix inversion algorithms is the need for a high number precision which gives rise to a large silicon area in fixed-point VLSI implementations. The two main reasons for these numerical requirements are: i) the use of costly operation such as square root and divisions, which leads to a significant increase of the dynamic range for some intermediate variables; and ii) the desire to replace repeated divisions by multiplications with the corresponding inverse in order to reduce the number of costly operations. Unfortunately, multiplications often results in an enhancement of the quantization noise and thus requires a high fixed point precision.

A VLSI architecture has therefore been proposed in [28] to deal with numerical problems for fixed-point implementation. It was based on the QR decomposition with modified Gram-Schmidt algorithm. The results showed that for typical 4×4 MIMO channel matrices, the architecture was able to achieve a clock rate of 277 MHz with a latency of 18 time units and area of 72K gates using $0.18 - \mu m$ CMOS technology, which is impressive compared to previously known architectures. In other direction, the architecture can be designed focusing on reducing number of matrix inversions, which is well-suited to the systems with multiple channels to be processed such as MIMO-OFDM systems [31], [30].

C. Detection Processing

This is where all operations are carried out when a new set of received signal symbols arrives. The resources required for the detection processing is in fact much less compared to the preprocessing stage. In addition, the hardware for preprocessing can be conveniently reused for the detection processing. As a result, the latency in the detection processing is reasonably low. Two operations will be discussed in this section: List sorting in the lattice domain and MMSE filtering to find the estimates of $\mathbf{s}_1$ and $\mathbf{s}_2$.

1) List Sorting in LR Domain: The list of candidate vectors in the LR domain is formed by

$$\mathcal{C}_2 = \{\mathbf{c}_2 \mid ||\tilde{\mathbf{c}}_2 - \mathbf{c}_2|| < r(Q)\}.$$

The problem is that the alphabet of signal in LR domain ($\mathbf{c}_2$) varies depending on channel. For example, while the alphabet of $\mathbf{s}_2$ is known, that of $\mathbf{c}_2 = \mathbf{U}_2^{-1}(\alpha\mathbf{s}_2 + \beta\mathbf{1})$ depends on $\mathbf{U}_2$. However, with Gaussian reduction method, $\mathbf{U}_2$ has always a form of

$$\mathbf{U}_2 = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix},$$

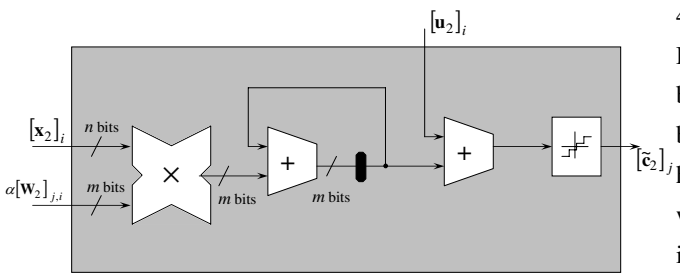

Fig. 4. Block diagram of the linear filtering operation: Inputs are $\mathbf{x}_2$, α, $\mathbf{W}_2$ and $\mathbf{u}_2$ while output is $\tilde{\mathbf{c}}_2$.

where t is an integer. As the maximum number of iterations in the Gaussian LR algorithm is limited to $T = 2$ or 3 only, we can easily obtain a known set of t (and accordingly $\mathbf{U}_2$). Thus, a look-up table can be formed for the alphabet of $\mathbf{c}_2$. This look-up table is formed in the pre-processing stage after the Gaussian LR algorithm is carried out to subchannel matrix $\mathbf{R}_2$. Memory is required to store this pre-calculated data. For example, it requires TM words of memory to store the alphabet of $\mathbf{c}_2$, where M is the size of alphabet of $\mathbf{s}_2$. In addition, $2Q$ words are required for storing $\mathcal{C}_2$.

2) MMSE filtering: This is a matrix-multiplication based operation. One MMSE filtering operation to estimate $\mathbf{c}_2$ is applied to received signal vector $\mathbf{x}_2$:

$$\tilde{\mathbf{c}}_2 = \alpha \mathbf{W}_2^H \mathbf{x}_2 + \mathbf{u}_2,$$

where $\mathbf{u}_2 = \mathbf{U}_2^{-1}\beta\mathbf{1}$. Q times of same operation are applied to received signal vector $\mathbf{x}_1$:

$$\tilde{\mathbf{c}}_1^{(q)} = \alpha \mathbf{W}_1^H \bar{\mathbf{x}}_1^{(q)} + \mathbf{u}_1, \tag{22}$$

where $\mathbf{u}_1 = \mathbf{U}_1^{-1}\beta\mathbf{1}$ and $\bar{\mathbf{x}}_1^{(q)} = \mathbf{x}_1 - \mathbf{R}_3\hat{\mathbf{s}}_2^{(q)}$. Note that Q operations in (22) can be carried out in parallel (see Figure 3). The parallel structure often allows low latency and high throughput. The most complex steps can then be processed in a single cycle, however, at the expense of large silicon area. In addition, with parallel structure, memories need to be implemented based on register files for sufficient access bandwidth. Thus, trade-off between latency/throughput and silicon area needs to be considered.

The weight matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ are pre-calculated and stored in the pre-processing stage. Note that only 8 words of memory are needed for this storage requirement. A simple VLSI architecture for MMSE filtering of $\mathbf{x}_2$ is shown in Figure 4. Filtering operation for $\mathbf{x}_1^{(q)}$ can be carried out similarly. Due to different dynamic ranges, variables can be represented by different numbers of bits (e.g., n bits for $\mathbf{x}_2$ whereas m bits for $\mathbf{W}_2$). It is expected that $m > n$ as entries of $\mathbf{W}_2$ has a larger dynamic range, thus they should be presented with considerable number of bits for the accurate fixed-point implementation.

Memory-wise, there are $2Q$ words required to store the outputs $\{\tilde{\mathbf{c}}_1, \tilde{\mathbf{c}}_2, \cdots, \tilde{\mathbf{c}}_Q\}$.

D. Fixed-Point Considerations

A critical issue in fixed-point arithmetic is the difference in dynamic ranges of variables. Number of integer and fractional bits for each variable should be carefully determined to avoid overflows and, at the same time, not to waste hardware resources.

For example, entries of channel matrix $\mathbf{H}$ is usually assumed to be Gaussian distributed, thus has a infinite dynamic range. To deal with this problem, two common approaches can be employed:

- A sufficiently large number of integer bits is used for representing $\mathbf{H}$ to ensure that overflows occur only rarely. At the same time, the round-off error (i.e., accumulation of rounding errors during fixed point arithmetic operations) should be purely due to loss in fractional precision. In this case, it is shown in [27] that the error variance varies only with the number of fractional bits, η, in the form:
$$\sigma_e^2 = 2^{-2\eta}/3.$$
- Automatic gain control adjusts the data of $\mathbf{H}$ to the available number of integer bits with an appropriate scaling factor γ in which the new channel matrix become $\bar{\mathbf{H}} = \gamma\mathbf{H}$. γ can be chosen as
$$\gamma = \frac{1}{\max|[\mathbf{H}]_{i,j}|}.$$

Depending on hardware resources, each approach can be applied. However, practical systems tend to compromise between the two approaches.

V. Simulation Results

We run simulations for MIMO-OFDMA LTE downlink system with parameters being given in Table II.

Figures 5 and 6 show bit error rate (BER) performance of different detectors for TU and SCM channels. 4-QAM is

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Center Frequency	3.5GHz
Bandwidth	10MHz
Subcarrier Spacing	15kHz
FFT size	1024
Number of usable subcarriers	601
Cyclic Prefix (CP)	FFT size / 8
Channel Model & Velocity	TU-30km/h and SCM-3km/h
Modulation	16-QAM, Gray Mapping
Channel Coding	Turbo Coding, Code Rate 1/2
Channel Estimation	Ideal
Data Mapping	Localized Subcarrier Pattern

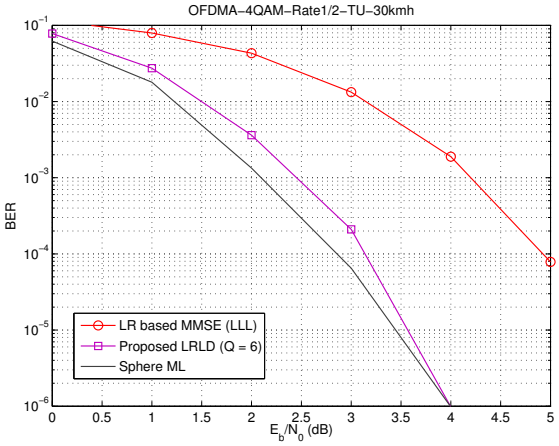

Fig. 5. BER performance comparison of different detectors with 4QAM modulation and TU channel (receiver velocity of 30kmh.)

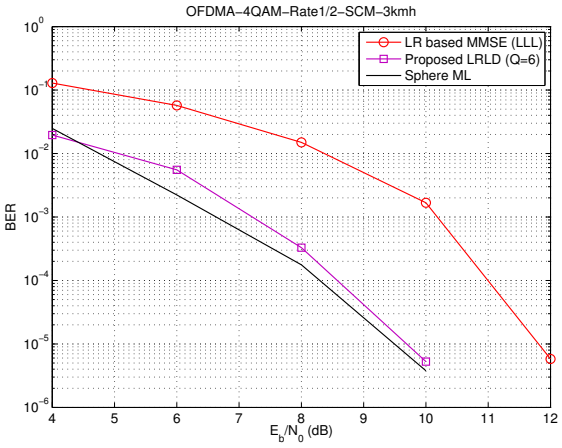

Fig. 6. BER performance comparison of different detectors 4QAM modulation and SCM channel (receiver velocity of 3kmh.)

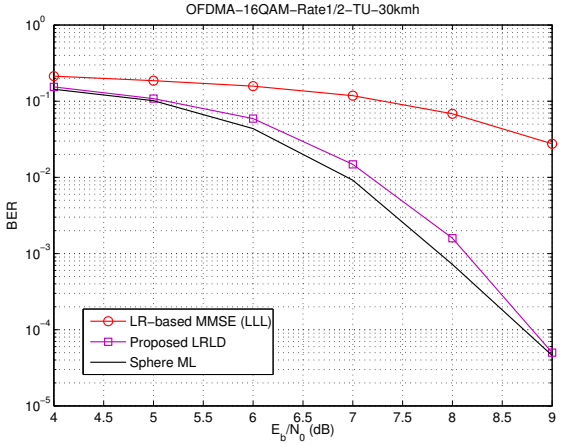

Fig. 7. BER performance comparison of different detectors with 16QAM modulation and TU channel (receiver velocity of 30kmh.)

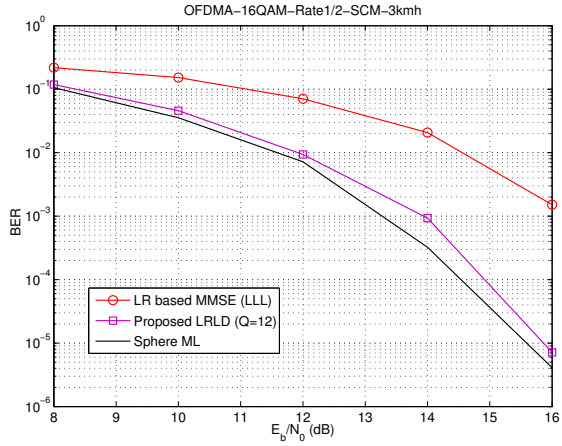

Fig. 8. BER performance comparison of different detectors with 16QAM modulation and SCM channel (receiver velocity of 3kmh.)

used for modulation. We compare the proposed LRLD based detector with the conventional LR based Minimum Mean Square Error (MMSE) detector that uses LenstraLenstraLovsz (LLL) algorithm [32] and the optimal sphere ML detector. It can be seen that the proposed detector provides a near ML performance and outperform the conventional LR based MMSE detector. The same behaviour is observed with 16-QAM modulation in Figures 7 and 8.

Complexity comparison: To fully examine the complexity of different detection methods, simulation is considered and results are shown in Figure 9 where the estimated flops using MATLAB execution time were obtained over all operations for each detector under the same environment. The execution time

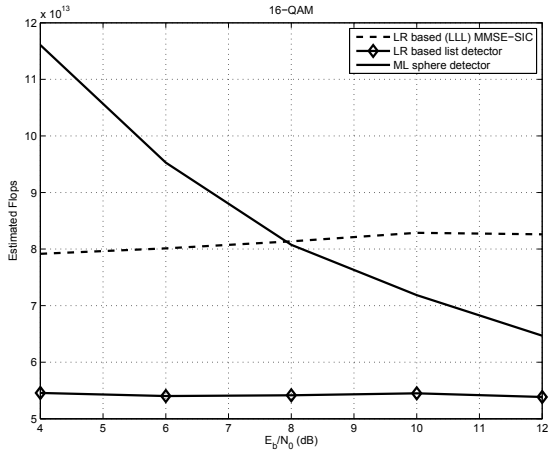

Fig. 9. Complexity comparison.

is averaged over hundreds of thousands of channel realizations. Note that Schnorr-Euchner algorithm [33] is used for sphere ML detector. The LLL-reduced algorithm with reduction factor $\delta = 3/4$ [32] is chosen for the LR based MMSE-SIC detector. No limitation on the number of iterations is imposed for any LR algorithm. The proposed LRLD based detector clearly requires the lowest number of flops. We can also see that the number of flops of the proposed detector is slightly higher than half of that of the LR based MMSE-SIC detector where the LLL-reduced algorithm is used.

VI. Conclusion

An efficient signal detector based on two techniques, namely LR and LD, has been investigated in this paper for the MIMO-OFDMA LTE downlink systems. By generating the list in LR domain, a more reliable list detection is obtained to facilitate SIC detection. As a result, the proposed detector outperforms conventional LR based detectors and provides a near ML performance with significantly reduced complexity. The implementation possibility was then studied to provide references for the real silicon implementation.

Appendix A
LR based Signal Detection

We describe the LR based detection that is used in Steps AS1 and AS4. Let $\mathbb{C}$ denote the set of complex integers or Gaussian integers, $\mathbb{C} = \mathbb{Z} + j\mathbb{Z}$, where $\mathbb{Z}$ denotes the set of integers and $j = \sqrt{-1}$. We assume that $\{\alpha s + \beta | s \in \mathcal{S}\} \subseteq \mathbb{C}$, where α and β are the scaling and shifting coefficients, respectively. For example, for M-QAM, if $M = 2^{2m}$, we have

$$\mathcal{S} = \{s = a + jb | a, b \in \{\pm A, \pm 3A, \ldots, \pm(2m-1)A\}\},$$

where $A = \sqrt{(3E_s/2(M-1))}$ and $E_s = E[|s|^2]$ denotes the symbol energy. Thus, $\alpha = 1/(2A)$ and $\beta = ((2m-1)/2)(1+j)$. Note that the pair of α and β is not uniquely decided.

Consider the MIMO detection with the following signal:

$$\mathbf{y} = \mathbf{A}\mathbf{z} + \mathbf{v}, \tag{23}$$

where $\mathbf{A}$ is a MIMO channel matrix, $\mathbf{z} \in \mathcal{S}^{K_i}$ is the signal vector, and $\mathbf{v}$ is a zero-mean Gaussian noise with $E[\mathbf{v}\mathbf{v}^{\mathrm{H}}] = \sigma_w^2\mathbf{I}$. We scale and shift $\mathbf{y}$ as

$$\mathbf{d} = \alpha\mathbf{y} + \beta\mathbf{A}\mathbf{1} = \mathbf{A}(\alpha\mathbf{z} + \beta\mathbf{1}) + \alpha\mathbf{v} = \mathbf{A}\mathbf{b} + \alpha\mathbf{v}, \tag{24}$$

where $\mathbf{1} = [1\ 1\ \ldots\ 1]^{\mathrm{T}}$, and $\mathbf{b} = \alpha\mathbf{z} + \beta\mathbf{1} \in \mathbb{C}^{K_i}$. Let $\bar{\mathbf{A}} = \mathbf{A}\mathbf{U}$ where $\mathbf{U}$ is a unimodular matrix. Using any LR algorithm including LLL algorithm [32], we can find $\mathbf{U}$ that makes the column vectors of $\bar{\mathbf{A}}$ shorter. It follows that

$$\mathbf{d} = \mathbf{A}\mathbf{U}\mathbf{U}^{-1}\mathbf{b} + \alpha\mathbf{v} = \bar{\mathbf{A}}\mathbf{c} + \alpha\mathbf{v}, \tag{25}$$

where $\mathbf{c} = \mathbf{U}^{-1}\mathbf{b}$. The MMSE filter to estimate $\mathbf{c}$ is given by

$$\begin{aligned}\mathbf{W}_{\mathrm{mmse}} &= \min_{\mathbf{W}} E[||\mathbf{W}^{\mathrm{H}}(\mathbf{d} - \bar{\mathbf{d}}) - (\mathbf{c} - \bar{\mathbf{c}})||^2] \\ &= (\mathbf{A}\mathbf{A}^{\mathrm{H}}\alpha^2 E_s + |\alpha|^2\sigma_w^2\mathbf{I})^{-1}\mathbf{A}\mathbf{U}^{-\mathrm{H}}\alpha^2 E_s, \end{aligned} \tag{26}$$

where $\bar{\mathbf{d}} = E[\mathbf{d}] = \beta\mathbf{A}\mathbf{1}$, $\bar{\mathbf{c}} = E[\mathbf{c}] = \mathbf{U}^{-1}\beta\mathbf{1}$, and $\mathrm{Cov}(\mathbf{c}) = |\alpha|^2\mathbf{U}^{-1}\mathbf{U}^{-\mathrm{H}}E_s$. The estimate of $\mathbf{c}$ is given by:

$$\tilde{\mathbf{c}} = \bar{\mathbf{c}} + \mathbf{W}_{\mathrm{mmse}}^{\mathrm{H}}(\mathbf{d} - \bar{\mathbf{d}}).$$

In Table III, the signals and parameters for the LR based MMSE detection for each step are shown.

TABLE III
Signals and parameters for the LR-based detection

Steps	$\mathbf{y}$	$\mathbf{A}$	$\mathbf{z}$	$\tilde{\mathbf{c}}$	K_i
AS1)	$\mathbf{x}_2$	$\mathbf{R}_2$	$\mathbf{s}_2$	$\tilde{\mathbf{c}}_2$	K_2
AS4)	$\mathbf{x}_1 - \mathbf{R}_2\hat{\mathbf{s}}_2^{(q)}$	$\mathbf{R}_1$	$\mathbf{s}_1$	$\tilde{\mathbf{c}}_1^{(q)}$	K_1

Appendix B
List Generation in the LR Domain

To avoid or mitigate the error propagation, the use of a list of candidate vectors of $\mathbf{s}_2$ in detecting $\mathbf{s}_1$ is crucial. Using the

ML metric, we can find the candidate vectors in the list, $\mathcal{S}_2$. Let

$$||\mathbf{r} - \mathbf{R}_2\hat{\mathbf{s}}_2^{(1)}||^2 \leq ||\mathbf{r} - \mathbf{R}_2\hat{\mathbf{s}}_2^{(2)}||^2 \leq \ldots \leq ||\mathbf{r} - \mathbf{R}_2\hat{\mathbf{s}}_2^{(M^{K_2})}||^2,$$

where $\hat{\mathbf{s}}_2^{(q)}$ denotes the symbol vector that corresponds to the qth largest likelihood. Therefore, an ideal list would be

$$\mathcal{S}_2 = \{\hat{\mathbf{s}}_2^{(1)}, \hat{\mathbf{s}}_2^{(2)}, \ldots, \hat{\mathbf{s}}_2^{(Q)}\}. \quad (27)$$

However, this requires an exhaustive search, which results in a high computational complexity due to computing of $\mathbf{R}_2\mathbf{s}_2$ for all $\mathbf{s}_2 \in \mathcal{S}^{K_2}$.

To avoid a high computational complexity, we can find a suboptimal list in the LR domain with low complexity. Consider (24). According to Table III, let $\mathbf{A} = \mathbf{R}_2$, $\mathbf{d} = \alpha\mathbf{x}_2 + \beta\mathbf{A}\mathbf{1}$, and $\mathbf{b} = \alpha\mathbf{s}_2 + \beta\mathbf{1}$. Then, since $\bar{\mathbf{A}} = \mathbf{A}\mathbf{U}$, we can see that the ML metric to construct the list is given by

$$||\mathbf{d} - \mathbf{A}\mathbf{b}|| = ||\mathbf{d} - \bar{\mathbf{A}}\mathbf{c}||. \quad (28)$$

It is noteworthy that the metric on the right hand side in (28) is defined in the LR domain. Let $\tilde{\mathbf{s}}_2$ be the signal vector in $\mathcal{S}^{K_2}$ corresponding to $\tilde{\mathbf{c}}_2$ and assume that $\tilde{\mathbf{s}}_2$ is sufficiently close to $\hat{\mathbf{s}}_2^{(1)}$. Then, we can have $\mathbf{d} \simeq \bar{\mathbf{A}}\tilde{\mathbf{c}}_2$. From this, the ML metric (ignoring a scaling factor) for constructing the list in the LR domain becomes

$$||\mathbf{d} - \bar{\mathbf{A}}\mathbf{c}|| = ||\bar{\mathbf{A}}\tilde{\mathbf{c}}_2 - \bar{\mathbf{A}}\mathbf{c}|| = ||\tilde{\mathbf{c}}_2 - \mathbf{c}||_{\bar{\mathbf{A}}^{\mathrm{H}}\bar{\mathbf{A}}}, \quad (29)$$

where $||\mathbf{x}||_{\mathbf{A}} = \sqrt{\mathbf{x}^{\mathrm{H}}\mathbf{A}\mathbf{x}}$ is a weighted norm. The list in the LR domain becomes

$$\mathcal{C}_2 = \{\mathbf{c} \mid ||\tilde{\mathbf{c}}_2 - \mathbf{c}||_{\bar{\mathbf{A}}^{\mathrm{H}}\bar{\mathbf{A}}} < r_{\bar{\mathbf{A}}}(Q)\}, \quad (30)$$

where $r_{\bar{\mathbf{A}}}(Q) > 0$ is the radius of an ellipsoid centered at $\tilde{\mathbf{c}}_2$, which contains Q elements in the LR domain. If the column vectors of $\bar{\mathbf{A}}$ or the basis vectors in the LR domain are orthogonal, $\bar{\mathbf{A}}^{\mathrm{H}}\bar{\mathbf{A}}$ becomes diagonal. Furthermore, if they have the same norm, $\bar{\mathbf{A}}^{\mathrm{H}}\bar{\mathbf{A}} \propto \mathbf{I}$. Thus, for nearly orthogonal basis vectors of almost equal norm, the list of $\mathbf{c}_2$ can be approximated as

$$\mathcal{C}_2 \simeq \{\mathbf{c} \mid ||\tilde{\mathbf{c}}_2 - \mathbf{c}|| < r(Q)\}, \quad (31)$$

where $r(Q) > 0$ is the radius of a sphere centered at $\tilde{\mathbf{c}}_2$, which contains Q elements. Since the LR provides a set of nearly orthogonal basis vectors for the LR based detection, we can see that the column vectors in $\bar{\mathbf{A}}$ are nearly orthogonal with a a two-basis system. Let $\mathcal{S}_2$ denotes the list in the original domain converted from $\mathcal{C}_2$ as in step AS3. Since no matrix-vector multiplications are required to generate $\mathcal{C}_2$ or $\mathcal{S}_2$, we can use $\mathcal{S}_2$ as the list in the proposed detector to reduce computational complexity. Note that the list generated in the LR domain is much more reliable than the list generated in the original domain (this list is different from $\mathcal{S}_2$).

REFERENCES

[1] H. X. Nguyen, "An efficient signal detection algorithm for 3GPP LTE downlink," in *Proc. IEEE International Conf. on Digital Telecommunications (ICDT 2010)*, Athens, Greece, Jun. 2010, pp. 77-81.

[2] 3rd Generation Partnership Project (3GPP) TR 25.814, "Technical specification group radio access network: Physical layer aspects for Evolved UTRA," *http://www.3gpp.org/ftp/Specs/html-info/25814.htm*.

[3] H. Ekstrom, A. Furuskar, J. Karlsson, M. Meyer, S. Parkvall, J. Torsner, and M. Wahlqvist, "Technical solutions for the 3G Long-Term Evolution," *IEEE Commun. Mag.*, vol. 44, pp. 38-45, Mar. 2006.

[4] H. G. Myung, J. Lim, and D. J. Goodman, "Single carrier FDMA for uplink wireless transmission," *IEEE Veh. Technol. Mag.*, vol. 1, pp. 30-38, Sep. 2006.

[5] G. J. Foschini, G. Golden, R. Valenzuela, and P. Wolniansky, "Simplified processing for wireless communication at high spectral efficiency," *IEEE J. Select. Areas Commun.*, no. 11, pp. 1841-1852, 1999.

[6] W. J. Choi, R. Negi, and J. Cioffi, "Combined ML and DFE decoding fo the V-BLAST system," in *Proc. IEEE International Conf. Communications*, New Orleans, LA, 2000, pp. 1243-1248.

[7] J. Luo, K. Pattipati, P. Willett, and F. Hasegawa, "Near optimal multiuser detection in synchronous CDMA using probabilistic data association," *IEEE Commun. Lett.*, vol. 5, pp. 361-363, Sep. 2001.

[8] D. Pham, K. R. Pattipati, P. K. Willett, and J. Luo "A generalized probabilistic data association detector for multiple antenna systems," *IEEE Commun. Lett.*, vol. 8, no. 4, April 2004.

[9] J. Choi, "On the partial MAP detection with applications to MIMO channels," *IEEE Trans. Signal Proc.*, vol.53, pp.158-167, Jan. 2005.

[10] D. J. Love, S. Hosur, A. Batra, and R. W. Heath, "Chase decoding for space-time codes," in *Proc. IEEE Vehicular Technology Conf.*, vol. 3, Nov. 2004, pp. 1663-1667.

[11] D. W. Waters and J. R. Barry, "The Chase family of detection algorithms for multiple-input multiple-output channels," *IEEE Trans. Signal Proc.*, vol. 56, No. 2, pp. 739-747, February 2008.

[12] H. Yao and G. W. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *Proc. IEEE Global Telecommunications Conf.*, Taiwan, Nov. 2002, pp. 424-428.

[13] D. Wubben, R. Bohnke, V. Kuhn and K. -D. Kammeyer, "Near-maximum-likelihood detection of MIMO systems using MMSE-based lattice reduction" in *Proc. IEEE International Conf. Communications*, vol. 2, Paris, Jun. 2004. pp. 798-802.

[14] H. Bolcskei, "MIMO-OFDM wireless systems: Basics, perspectives, and challenges," *IEEE Wireless Commmun.*, vol. 13, pp. 31-37, Aug. 2006.

[15] D. Perels, S. Haene, P. Luethi, A. Burg, N. Felber, W. Fichtner, and H. Bolcskei, "ASIC Implementation of a MIMO-OFDM Transceiver for 192 Mbps WLANs," *Proc. ESSCIRC*, Grenoble, France, 2005, pp. 215-218.

[16] Z. Guo and P. Nilsson, "A VLSI implementation of MIMO detection for future wireless communications," in *Proc. 14th IEEE 2003 Int. Symp. Personal, Indoor and Mobile Radio Communication*, 2003, pp. 2852 - 2856.

[17] G. Knagge, L. Davis, G. Woodwar, S. R. Weller, "VLSI preprocessing techniques for MUD and MIMO sphere detection," in *Proc. 6th Australian Communications Theory Workshop*, Feb. 2005, pp. 221 - 228.

[18] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, pp. 1566 - 1577, Jul. 2005.

[19] A. Burg, M. Borgmann, M. Wenk, C. Studer, and H. Bolcskei, "Advanced receiver algorithms for MIMO wireless communications," in *Proc. Design, Automation and Test in Europe (DATE '06)*, vol. 1, Mar. 2006.

[20] C. Studer, A. Burg, and H. Bolcskei, "Soft-output sphere decoding: Algorithms and VLSI implementation," submitted to *IEEE J. Select. Areas Commun.*, Apr. 2007.

[21] D. Garrett, L. Davis, S. ten Brink, B. Hochwald, and G. Knagge, "Silicon complexity for maximum likelihood MIMO detection using spherical decoding," *IEEE J. Solid-State Circuits*, vol. 39, pp. 1544 - 1552, Sep. 2004.

[22] S. Chen, T. Zhang, and Y. Xin, "Relaxed K-Best MIMO signal detector design and VLSI implementation," *IEEE Trans. VLSI Syst.*, vol. 15, pp. 328 - 337, Mar. 2007.

[23] R. Steele, *Mobile Radio Communications*, New York: IEEE Press, 1992.

[24] 3GPP, TR 25.996, "Spatial channel model for multiple input multiple output (MIMO) simulations (Rel. 6)," 2003.

[25] L. M. Davis, "Scaled and decoupled cholesky and QR decompositions with application to spherical MIMO detection," in *Proc. IEEE Wireless Communications and Networking Conf.*, vol. 1, Mar. 2003, pp. 326-331.

[26] G. H. Golub and C. F. V. Loan, *Matrix computations*, 3rd ed. Baltimore, MD: John Hopkins University Press, 1996.

[27] C. K. Singh, S. H. Prasad, and P. T. Balsara, "A fixed-point implementation for QR Decomposition," in *Proc. 2006 IEEE Dallas/CAS Workshop Design, Applications, Integration and Software*, Oct. 2006, pp. 75-78.

[28] C. K. Singh, S. H. Prasad, and P. T. Balsara, "VLSI architecture for matrix inversion using modified Gram-Schmidt based QR decomposition," in *Proc. 20th IEEE Int. Conf. VLSI Design*, Jan. 2007, pp. 836-841.

[29] A. Bjorck, and C. Paige, "Loss and recapture of orthogonality in the modified gram-schmidt algorithm," *SIAM J. Matrix Anal. Appl.*, vol. 13(1), pp. 176-190, 1992.

[30] M. Borgmann and H. Bolcskei, "Interpolation-based efficient matrix inversion for MIMO-OFDM receivers," in *Proc. 38th Asilomar Conf. Signals, Systems, Computers*, vol. 2, Pacific Grove, CA, Nov. 2004, pp. 1941-1947.

[31] D. Cescato, M. Borgmann, H. Bolcskei, J. Hansen, and A. Burg, "Interpolation-based QR decomposition in MIMO-OFDM systems," in *Proc. 6th IEEE Workshop Signal Processing Advances in Wireless Communications (SPAWC)*, New York, NY, Jun. 2005, pp. 945-949.

[32] A. K. Lenstra, J. H. W. Lenstra, and L. Lovasz, "Factorizing polynomials with rational coefficients," *Math. Ann.*, vol. 261, pp. 515-534, 1982.

[33] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math.Programming*, vol. 66, pp. 181-191, 1994.