EFFICIENT SUBSPACE DETECTION FOR HIGH-ORDER MIMO SYSTEMS

Hadi Sarieddeen *Mohammad M. Mansour* *Ali Chehab*

Department of Electrical and Computer Engineering
American University of Beirut, Beirut 1107 2020, Lebanon
{has63, mmansour, chehab}@aub.edu.lb

ABSTRACT

In this paper, low-complexity multiple-input multiple-output (MIMO) subspace detection schemes are studied, which decompose a channel into multiple decoupled streams to be detected disjointly. Existing schemes require a number of matrix decomposition operations equal to the number of detected streams, which is computationally complex, especially in high-order MIMO systems. We propose two computationally efficient detection algorithms, based on a preprocessing stage that consists of special layer ordering, followed by permutation-robust QR decomposition (QRD) and elementary matrix operations. The algorithms are illustrated in the context of a 4-layer MIMO system, and their complexity is studied. Simulations demonstrate that using the proposed scheme, the QRD overhead is reduced by almost 50% for very high order MIMO, without incurring any performance degradation.

***Index Terms*—** MIMO systems, subspace detection, QR decomposition, column permutation

1. INTRODUCTION

Multiple-input multiple-output (MIMO) antenna systems are a key enabling technology in current and future wireless communication standards to increase spectral efficiency [1]. At the receiver side, different detection schemes provide different performance-complexity tradeoffs. At one end we have the linear detectors, such as zero forcing (ZF) and minimum mean square error (MMSE) [2], which are least-complex but least-efficient, and at the other end we have the most-complex optimal maximum likelihood (ML) detector. In between, many suboptimal detectors that offer complexity-performance tradeoffs exist, such as the sphere decoder (SD) and its variants [3–7].

An important MIMO detection class of algorithms that offers a good performance, and a complexity that only increases linearly with the number of transmit antennas, is subspace detection. Subspace detection consists of decomposing an effective channel matrix into lower order subchannels, to reduce the number of jointly detected streams. One way to decompose a channel matrix is by using geometric mean decomposition (GMD) [8–10], which can uniformly decomposes a channel into identical parallel subchannels. However, several subchannels are allowed to overlap by using QR decomposition (QRD) instead [11], which can result in additional diversity when less reliable data streams are put in multiple subchannel detection sets. The popular layered orthogonal lattice detector (LORD) [12] can be viewed as a special class of subspace detectors.

In [13,14], a comprehensive study on MIMO subspace detection was conducted, and a generic detector was proposed, that jointly detects subsets of decoupled streams, by transforming the channel matrix into a generalized elementary structure, through QL decomposition followed by elementary matrix operations. However, even when the subsets only include single streams, this approach requires a computationally intensive overhead of matrix decompositions in its preprocessing stage, which gets worse with higher order MIMO systems. In this paper, we build on the work in [13, 14], and propose two detection algorithms that can reduce this overhead.

Regarding notations, bold upper case, bold lower case, and lower case letters correspond to matrices, vectors, and scalars, respectively. Scalar and vector norms are represented by $|\cdot|$ and $\|\cdot\|$, respectively. $\mathsf{E}[\cdot]$, $(\cdot)^T$, $(\cdot)^*$, $(\cdot)^{-1}$, and $\mathbf{I}_M$ stand for the expected value, transpose, conjugate transpose, matrix inverse, and identity matrix of size M, respectively.

2. SYSTEM MODEL

We consider a MIMO system with N transmit antennas and $M \geq N$ receive antennas. The received complex symbol vector is represented as:

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

with $\mathbf{y} \in \mathcal{C}^{M\times 1}$ being the received complex vector, $\mathbf{H} \in \mathcal{C}^{M\times N}$ the spacially multiplexed complex channel matrix, $\mathbf{x} \in \mathcal{C}^{N\times 1}$ the transmitted symbol vector, and $\mathbf{n} \in \mathcal{C}^{M\times 1}$ the complex additive white Gaussian noise vector with zero mean and variance $\sigma_{\mathbf{n}}^2$ ($\mathsf{E}[\mathbf{n}\mathbf{n}^*] = \sigma_{\mathbf{n}}^2\mathbf{I}_M$). Each symbol x_n belongs to a normalized complex constellation $\mathcal{X}_n$ of size $Q_n = 2^{q_n}$, thus $\mathbf{x} = [x_1 x_2 \dots x_N]^T \in \mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_N$ and $\mathsf{E}[x_n^* \cdot x_n] = 1$. The noise variance is defined in terms of the signal to noise ration (SNR) as $\sigma_{\mathbf{n}}^2 = (N/\mathsf{SNR})$. The bit representation of a symbol is a coded bit-interleaved sequence $\mathbf{b}_n = (b_{n,1}, b_{n,2}, \dots, b_{n,q_n})$.

ML MIMO detection consists of exhaustively searching a lattice $\mathcal{X}$ to find a symbol vector that minimizes the Euclidean distance metric:

$$d^{\mathrm{ML}} = \min_{\mathbf{x}\in\mathcal{X}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \tag{2}$$

The unscaled log-likelihood ratio (LLR) of bit $b_{n,k}$, generated by the soft-output ML detector, is calculated as:

$$\lambda_{n,k}^{\mathrm{ML}} = \min_{\mathbf{x}\in\mathcal{X}_{n,k}^{(0)}} d(\mathbf{x}) - \min_{\mathbf{x}\in\mathcal{X}_{n,k}^{(1)}} d(\mathbf{x}) \tag{3}$$

where $n = 1, \dots, N$ and $k = 1, \dots, q_n$. The sets $\mathcal{X}_{n,k}^{(0)} = \mathbf{x} \in \mathcal{X} : b_{n,k} = 0$ and $\mathcal{X}_{n,k}^{(1)} = \mathbf{x} \in \mathcal{X} : b_{n,k} = 1$ correspond to subsets of symbol vectors in $\mathcal{X}$, having in the corresponding kth bit of the nth symbol a value of 0 and 1, respectively. ML detection requires computing $\prod_{n=1}^{N} Q_n$ distance metrics to achieve optimum performance.

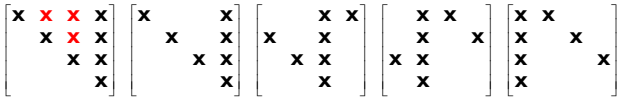

Fig. 1. QRD-based 4x4 channel matrix structures

3. SINGLE-STREAM SUBSPACE DETECTION

3.1. Generic WR Decomposition

The first step in subspace detection is channel matrix decomposition. Although the popular approach [12] is to undergo QRD, that results in an unpunctured upper triangular matrix as shown in Fig. 1(a), in what follows we consider a more powerful WR decomposition (WRD) scheme that punctures the red-marked entries above the diagonal. We consider $N=M$, and aim at transforming $\mathbf{H}$ into a punctured upper triangular matrix (UTM) $\mathbf{R}=[u_{ij}]\in\mathcal{C}^{N\times N}$ with $u_{ii}\in\mathcal{R}^{+}$, as shown in Fig. 1(b), through a matrix $\mathbf{W}=[\mathbf{w}_1\mathbf{w}_2\ldots\mathbf{w}_N]\in\mathcal{C}^{N\times N}$, such that $\mathbf{W}^{*}\mathbf{H}=\mathbf{R}$. We assume $\mathbf{H}=[\mathbf{h}_1\mathbf{h}_2\ldots\mathbf{h}_N]$ to have a full column rank. Setting $\mathbf{W}=(\mathbf{H}^{*}\mathbf{H})^{-1}\mathbf{H}^{*}$ to be the left Moore-Penrose pseudo-inverse of $\mathbf{H}$ results in $\mathbf{R}=\mathbf{I}_N$, and choosing $\mathbf{W}$ to be an orthonormal basis of the column space of $\mathbf{H}$, transforms it into an unpunctured UTM, with $\mathbf{W}$ being unitary (QRD). In general, if $\mathbf{R}$ is punctured, then $\mathbf{W}$ is non-unitary. However, if we impose the condition on the column vectors of $\mathbf{W}$ to have unit length, i.e., $\mathbf{w}_n^{*}\mathbf{w}_n=1$ for $n=1,\ldots,N$, the transformed noise vector is guaranteed to maintain an unaltered covariance matrix ($\mathsf{E}[\mathbf{W}^{*}\mathbf{n}\mathbf{n}^{*}\mathbf{W}]=\sigma_{\mathbf{n}}^2\mathbf{I}_N$).

Let $\mathbf{P}=\mathbf{H}(\mathbf{H}^{*}\mathbf{H})^{-1}\mathbf{H}^{*}$ be the orthogonal projection onto the column space of $\mathbf{H}$, and $\mathbf{P}^{\perp}=\mathbf{I}-\mathbf{H}(\mathbf{H}^{*}\mathbf{H})^{-1}\mathbf{H}^{*}$ be the orthogonal projection onto the left nullspace of $\mathbf{H}$. Let $\mathbf{H}_{\mathcal{I}}$ be the submatrix formed by the columns of $\mathbf{H}$ whose index $n\in\mathcal{I}$ (if $\mathcal{I}=1,3$, then $\mathbf{H}_{\mathcal{I}}=[\mathbf{h}_1\mathbf{h}_3]$). Denote by $\mathcal{I}_n$ the column index set of the entries in the nth row of $\mathbf{H}$ to be zeroed out, and define $\tilde{\mathbf{w}}_n=\mathbf{P}_{\mathcal{I}_n}^{\perp}\mathbf{h}_n$, where:

$$\mathbf{P}_{\mathcal{I}_n}^{\perp}=\mathbf{I}_N-\mathbf{H}_{\mathcal{I}_n}(\mathbf{H}_{\mathcal{I}_n}^{*}\mathbf{H}_{\mathcal{I}_n})^{-1}\mathbf{H}_{\mathcal{I}_n}^{*} \tag{4}$$

and $\mathbf{H}_{\mathcal{I}_n}=\{\mathbf{h}_m\,|\,m\in\mathcal{I}_n\}$. The normalized vector is derived as $\mathbf{w}=\tilde{\mathbf{w}}_n/\|\tilde{\mathbf{w}}_n\|$ with $\|\tilde{\mathbf{w}}_n\|=\sqrt{\mathbf{h}_n^{*}\mathbf{P}_{\mathcal{I}_n}^{\perp}\mathbf{h}_n}$. Let $\mathbf{D}=[d_n]\in\mathcal{R}^{+}$ be a diagonal matrix whose entries are given by $d_n=1/\sqrt{\mathbf{h}_n^{*}\mathbf{P}_{\mathcal{I}_n}^{\perp}\mathbf{h}_n}$, $n=1,\ldots,N$. The matrix that would zero out the entries in the rows of $\mathbf{H}$ at column positions given in $\mathcal{I}_n$ is:

$$\mathbf{W}^{*}=\mathbf{D}\begin{bmatrix}\mathbf{h}_1^{*}\mathbf{P}_{\mathcal{I}_1}^{\perp}\\ \mathbf{h}_2^{*}\mathbf{P}_{\mathcal{I}_2}^{\perp}\\ \vdots\\ \mathbf{h}_N^{*}\mathbf{P}_{\mathcal{I}_N}^{\perp}\end{bmatrix} \tag{5}$$

For example, in a 4×4 MIMO system, choosing the puncturing sets as $\mathcal{I}_1=2,3$, $\mathcal{I}_2=1,3$, $\mathcal{I}_3=1,2$, and $\mathcal{I}_4=1,2,3$, results in a punctured UTM $\mathbf{R}$ as follows:

$$\mathbf{W}^{*}\mathbf{H}=\mathbf{D}\begin{bmatrix}\mathbf{h}_1^{*}\mathbf{P}_{2,3}^{\perp}\\ \mathbf{h}_2^{*}\mathbf{P}_{1,3}^{\perp}\\ \mathbf{h}_3^{*}\mathbf{P}_{1,2}^{\perp}\\ \mathbf{h}_4^{*}\mathbf{P}_{1,2,3}^{\perp}\end{bmatrix}\mathbf{H}=\begin{bmatrix}\times & & & \times\\ & \times & & \times\\ & & \times & \times\\ & & & \times\end{bmatrix}=\mathbf{R} \tag{6}$$

3.2. Detection Algorithm

To generate soft-output LLRs for all layers, the N streams are decoupled, one at a time in N steps, by cyclically shifting the columns of $\mathbf{H}$ and generating the punctured UTMs, as shown in Fig. 1(b-e). We call this reference algorithm cyclical subspace detection (CYSD). Each permuted $\mathbf{H}$ at step t is WR-decomposed into $\mathbf{W}^{(t)}$ and $\mathbf{R}^{(t)}$. For simplicity, we assume $\mathcal{X}_n=\mathcal{M}$ for all n. We first partition $\tilde{\mathbf{y}}^{(t)}$, $\mathbf{R}^{(t)}$, and $\mathbf{x}$ as:

$$\tilde{\mathbf{y}}^{(t)}=\begin{bmatrix}\tilde{\mathbf{y}}_1^{(t)}\\ \tilde{y}_2^{(t)}\end{bmatrix},\ \mathbf{R}^{(t)}=\begin{bmatrix}\mathbf{A}^{(t)} & \mathbf{b}^{(t)}\\ 0 & c^{(t)}\end{bmatrix},\ \mathbf{x}=\begin{bmatrix}\mathbf{x}_1\\ x_2\end{bmatrix} \tag{7}$$

where $\tilde{\mathbf{y}}_1^{(t)}\in\mathcal{C}^{(N-1)\times1}$, $\tilde{y}_2^{(t)}\in\mathcal{C}^{1\times1}$, $\mathbf{A}^{(t)}\in\mathcal{R}^{(N-1)\times(N-1)}$, $\mathbf{b}^{(t)}\in\mathcal{C}^{(N-1)\times1}$, $c^{(t)}\in\mathcal{R}^{1\times1}$, $\mathbf{x}_1\in\mathcal{M}^{N-1}$, and $x_2\in\mathcal{M}$. Then the symbol vector with minimum distance for a structure t is:

$$\mathbf{x}_{(t)}^{\mathrm{WR}}=\arg\min_{\mathbf{x}\in\mathcal{X}}\left\|\tilde{\mathbf{y}}^{(t)}-\mathbf{R}^{(t)}\mathbf{x}\right\|^2 \tag{8}$$

$$=\arg\min_{x_2\in\mathcal{M}}\left(\left\|\tilde{y}_2^{(t)}-c^{(t)}x_2\right\|^2+\left\|\tilde{\mathbf{y}}_1^{(t)}-\mathbf{A}^{(t)}\hat{\mathbf{x}}_1-\mathbf{b}^{(t)}x_2\right\|^2\right) \tag{9}$$

where $\hat{\mathbf{x}}_1=\lfloor(\tilde{\mathbf{y}}_1^{(t)}-\mathbf{b}^{(t)}x_2)/\mathbf{A}^{(t)}\rceil_{\mathcal{M}^{N-1}}$ is the sliced output. Since $\mathbf{A}^{(t)}$ is a diagonal matrix, the slicing is applied to individual elements of the vector $\tilde{\mathbf{y}}_1^{(t)}$ over the constellation $\mathcal{M}$. In order to generate soft outputs, we compute two distance metrics defined as:

$$\mathbf{u}_{n,k,t}^{\mathrm{WR}}=\arg\min_{\mathbf{x}\in\mathcal{X}_{n,k}^{0}}\left\|\tilde{\mathbf{y}}^{(t)}-\mathbf{R}^{(t)}\mathbf{x}\right\|^2 \tag{10}$$

$$\mathbf{v}_{n,k,t}^{\mathrm{WR}}=\arg\min_{\mathbf{x}\in\mathcal{X}_{n,k}^{1}}\left\|\tilde{\mathbf{y}}^{(t)}-\mathbf{R}^{(t)}\mathbf{x}\right\|^2 \tag{11}$$

which can be expanded as in equation (8), and calculate the LLRs as:

$$\Lambda_{n,k,t}^{\mathrm{WR}}=\left\|\tilde{\mathbf{y}}^{(t)}-\mathbf{R}^{(t)}\mathbf{u}_{n,k,t}^{\mathrm{WR}}\right\|^2-\left\|\tilde{\mathbf{y}}^{(t)}-\mathbf{R}^{(t)}\mathbf{v}_{n,k,t}^{\mathrm{WR}}\right\|^2 \tag{12}$$

for $n=1,\ldots,N$, $k=1,\ldots,\log_2|\mathcal{M}|$, and $t=1,\ldots,N$.

Note that tighter LLRs can be computed by tracking global minimum distances rather than just minimizing over the per stream LLRs. Moreover, a multi-stream enumeration set can be exhaustively searched, where streams can be included in multiple detection sets to enhance performance.

4. PROPOSED DETECTION ALGORITHMS

4.1. Single-Permutation Subspace Detection

When cyclically shifting the columns of $\mathbf{H}$, the number of WRD operations required is equal to the number of layers to be detected, which is a significant computational burden that forms a bottleneck in high order MIMO. An alternative minimal swapping operation can reduce this computational overhead (Sec. 5). For example, in the case of 4×4 MIMO, if we want to compute the LLRs of the bits on layer 2, we can swap $\mathbf{h}_2$ with $\mathbf{h}_4$, and use the matrix decomposition of Fig. 1(b). We represent this swapping operation by a permutation:

$$\pi^{(t)}(i)=\begin{cases}N\ \text{if}\ i=t\\ t\ \text{if}\ i=N\\ i\ \text{otherwise}\end{cases} \tag{13}$$

for $t=1,\ldots,N$ and $i=1,\ldots,N$. The remainder of the derivation, equations (7) to (12), remains intact. We call this algorithm single-permutation subspace detection (SPSD).

4.2. Pairwise Subspace Detection

Another approach, which we will later argue to be of a practical interest, is what we call pairwise subspace detection (PWSD). This approach consists of lumping the channel columns in pairs (assuming N even), and handling each pair of layers at a time. First, the pair of interest is swapped with the rightmost two columns. Then, the columns of the pair get swapped so that each can be at position $i = N$. For example, in the case of 4×4 MIMO, the 4 permuted channels can be $\mathbf{H}_1 = [\mathbf{h}_3\mathbf{h}_4\mathbf{h}_1\mathbf{h}_2]$, $\mathbf{H}_2 = [\mathbf{h}_3\mathbf{h}_4\mathbf{h}_2\mathbf{h}_1]$, $\mathbf{H}_3 = [\mathbf{h}_1\mathbf{h}_2\mathbf{h}_3\mathbf{h}_4]$, and $\mathbf{H}_4 = [\mathbf{h}_1\mathbf{h}_2\mathbf{h}_4\mathbf{h}_3]$. After each of the N permutations, the permuted channel is decomposed, and the LLRs for the corresponding layer are computed (equations (7) to (12)).

5. EFFICIENT ARCHITECTURE

The brute force approach for computing $\mathbf{W}$ involves extensive matrix inversion, which is computationally expensive, and prone to numerical error when implemented with finite precision. However, there exist an alternative efficient scheme [14] to determine $\mathbf{W}$, which consists of QRD followed by elementary matrix operations.

5.1. Permutation-Robust Reduced-Complexity QRD

The QRD decomposes $\mathbf{H}$ into a unitary matrix $\mathbf{Q} = [\mathbf{q}_1\mathbf{q}_2 \dots \mathbf{q}_N]$ and an upper triangular matrix $\mathbf{R} = [r_{ij}]_{N\times N}$ with real and positive diagonal elements ($\mathbf{H} = \mathbf{QR}$). This can be computed using Givens rotation (GR), Gram-Schmidt (GS) orthogonalization, or Householder transformation (HT) [15]. While the hardware implementation of HT is very complex, GR reduces the hardware area, but at the expence of longer clock latency. The classical GS algorithm allows a memory efficient implementation due to its inherent parallelism, resulting in better regularity in data flow and a potential for better hardware-efficiency, however, due to fixed-precision computation and round off errors, it can not guarantee the orthogonality of $\mathbf{Q}$. This limitation was overcome by the numerically superior Modified Gram-Schmidt (MGS) algorithm.

The MGS-based QRD of $\mathbf{H}$ is illustrated in Fig. 2. The algorithm consists of two main parts. In the first part, the diagonal elements of $\mathbf{R}$ and the columns of $\mathbf{Q}$ are computed. In the second part, the non-diagonal elements of $\mathbf{R}$ are computed and the columns of $\mathbf{H}$ are updated. Considering a 4×4 complex matrix, in the first part of the first iteration, the norm of $\mathbf{h}_1$ is assigned to r_{11}, and $\mathbf{q}_1$ is calculated as $\mathbf{q}_1 = \mathbf{h}_1/r_{11}$. Then, in the second part r_{12} , r_{13} , and r_{14} are calculated using $\mathbf{q}_1$, $\mathbf{h}_2$, $\mathbf{h}_3$, and $\mathbf{h}_4$ follows:

$$r_{1j} = \mathbf{q}_1^T\mathbf{h}_j \quad 2 \le j \le 4 \tag{14}$$

and $\mathbf{H}$ is updated by setting its first column to zero and subtracting from the others the length of the projection of $\mathbf{q}_1$ on them, i.e.,

$$\mathbf{h}_j = \mathbf{h}_j - \mathbf{q}_1 r_{1j} \quad 2 \le j \le 4 \tag{15}$$

This procedure is repeated with one less column every new iteration.

Moreover, since in our proposed detection algorithms the $\mathbf{H}^{(t)}$ matrices are only one swap operation away, further simplifications can be introduced. In fact, when computing the QRD of a matrix, which is derived from another matrix, of known decomposition, by some column permutations, computational savings can be achieved. A part of the decomposition result remains unaltered under specific permutations. For example, assume as shown in Fig. 3, columns 3 and 4 in $\mathbf{H}$ (in blue) were permuted. The first two columns of $\mathbf{Q}$ and $\mathbf{R}$ (in red) depend only on the first two columns of $\mathbf{H}$, and hence there is no need to recompute them.

1: **procedure** MGS-QRD($\mathbf{H}$)
2: $k \leftarrow 1$;
3: **for** $k = 1 : N$ **do**
4: $r_{kk} \leftarrow \sqrt{\mathbf{h}_k^*\mathbf{h}_k}$
5: $\mathbf{q}_k \leftarrow \mathbf{h}_k/r_{kk}$
6: $j \leftarrow k + 1$
7: **for** $j = 1 : N$ **do**
8: $r_{kj} \leftarrow \mathbf{q}_k^T\mathbf{h}_j$
9: $\mathbf{h}_j \leftarrow \mathbf{h}_j - \mathbf{q}_k r_{kj}$
10: **end for**
11: **end for**
12: **end procedure**

Fig. 2. MGS QRD Algorithm

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ 0 & r_{22} & r_{23} & r_{24} \\ 0 & 0 & r_{33} & r_{34} \\ 0 & 0 & 0 & r_{44} \end{bmatrix}$$

Fig. 3. QRD Savings Under Column Permutations

5.2. Matrix Puncturing

Assume that $\mathbf{H}$ is QR-decomposed and we have $\mathbf{Q}_1^*\mathbf{H} = \mathbf{R}_1$. Obviously, $\mathbf{q}_N^*\mathbf{q}_N = 1$ and $\mathbf{q}_N^*\mathbf{h}_m = 0$ for all $m = 1, \dots, N-1$. Hence, $\mathbf{w}_N = \mathbf{q}_N$. Now consider row $1 < n \le N$ of $\mathbf{R_1}$, and assume the mth entry r_{nm}, $m > n$, is to be nulled. We have $\mathbf{q}_n^*\mathbf{h}_m = r_{nm} \in \mathcal{C}$ and $\mathbf{q}_m^*\mathbf{h}_m = r_{mm} \in \mathcal{R}^+$, from which it follows that $(\mathbf{q}_n^* - \mathbf{q}_m^*\frac{r_{nm}}{r_{mm}})\mathbf{h}_m = 0$. Therefore, the equations:

$$\mathbf{q}_n = \mathbf{q}_n - \mathbf{q}_m r_{nm}^*/r_{mm} \tag{16}$$

$$r_{nj} = r_{nj} - r_{mj}r_{nm}/r_{mm}, \quad \text{for } j = m, \dots, N \tag{17}$$

puncture the required entry and update $\mathbf{Q}_1$ accordingly. These operations are repeated for all other entries $m > n$ to be punctured in row n. Finally, $\mathbf{q}_n$ is normalized to have unit length, and the non-zero entries in row n of $\mathbf{R}_1$ are updated:

$$r_{nj} = r_{nj}/\left\|\mathbf{q}_n\right\|, \quad \text{for } j = n, \dots, N \tag{18}$$

$$\mathbf{q}_n = \mathbf{q}_n/\left\|\mathbf{q}_n\right\| \tag{19}$$

The operations in (16)-(17) followed by the normalization steps (18)-(19) are repeated for all rows n where puncturing is required. The resulting $\mathbf{Q}_1$ is $\mathbf{W}$, and $\mathbf{R}_1$ is the desired UTM $\mathbf{R}$.

In matrix form, we can write (16)-(17) using elementary matrices $\mathbf{E}_m = [e_{nj}], 1 \le m \le N$, which differ from $\mathbf{I}_N$ by a single elementary row operation, defined as follows:

$$e_{nj} = \begin{cases} 1 \text{ if } j = n \\ -r_{nm}/r_{mm} \text{ if } j = m,\ j \in \mathcal{I}_n \\ 0 \text{ otherwise} \end{cases} \tag{20}$$

The product of these elementary matrices forms the unscaled matrices $\mathbf{R}_2 = (\mathbf{E}_n \dots \mathbf{E}_1)\mathbf{R}_1$ and $\mathbf{Q}_2^* = (\mathbf{E}_n^* \dots \mathbf{E}_1^*)\mathbf{Q}_1^*$. The scaling operations (18)-(19) can be written using the diagonal matrix $\mathbf{D} = [d_n] \in \mathcal{R}^+$, where $d_n = 1/\sqrt{[\mathbf{Q}_1^*\mathbf{Q}_2]_{nn}}$ and $[\cdot]_{nn}$ denotes the nth diagonal element. The desired (scaled) matrices are given by $\mathbf{W}^* = \mathbf{D}\mathbf{Q}_2^*$ and $\mathbf{R} = \mathbf{D}\mathbf{R}_2$.

Unlike QRD, there is no permutation-robust implementation for puncturing. The punctured elements are in the upper rows, affecting the leftmost columns of $\mathbf{Q}$. We still call the overall decomposition a permutation-robust WRD (PR-WRD).

Table 1. Computational Savings in Proposed Schemes

	Permutations	Redundant	Saved
SPSD	1: $\mathbf{h}_1\mathbf{h}_2\mathbf{h}_3\mathbf{h}_4$	none	none
	2: $\mathbf{h}_1\mathbf{h}_2\mathbf{h}_4\mathbf{h}_3$	$\mathbf{q}_1, \mathbf{q}_2, r_{11}, r_{12}, r_{22}$	88 RML+40 RAD
	3: $\mathbf{h}_1\mathbf{h}_4\mathbf{h}_3\mathbf{h}_2$	$\mathbf{q}_1\ r_{11}$	28 RML+8 RAD
	4: $\mathbf{h}_4\mathbf{h}_2\mathbf{h}_3\mathbf{h}_1$	none	none
PWSD	1: $\mathbf{h}_3\mathbf{h}_4\mathbf{h}_1\mathbf{h}_2$	none	none
	2: $\mathbf{h}_3\mathbf{h}_4\mathbf{h}_2\mathbf{h}_1$	$\mathbf{q}_1, \mathbf{q}_2, r_{11}, r_{12}, r_{22}$	88 RML+40 RAD
	3: $\mathbf{h}_1\mathbf{h}_2\mathbf{h}_3\mathbf{h}_4$	none	none
	4: $\mathbf{h}_1\mathbf{h}_2\mathbf{h}_4\mathbf{h}_3$	$\mathbf{q}_1, \mathbf{q}_2, r_{11}, r_{12}, r_{22}$	88 RML+40 RAD

5.3. Channel Matrix Augmentation

For detection, the product $\mathbf{W}^*\mathbf{y}$ must also be formed. This can be efficiently computed by first right-augmenting $\mathbf{y}$ to $\mathbf{H}$, and then performing QRD on the augmented matrix to form $\tilde{\mathbf{Q}}\tilde{\mathbf{R}} = [\mathbf{H}|\mathbf{y}]$. When carrying out the orthogonalization procedure, the same operations applied to the columns of $\mathbf{H}$ are applied to the augmented column. This results in $\tilde{\mathbf{Q}} = [\mathbf{Q}|\mathbf{0}_{N\times 1}]$, with $\tilde{\mathbf{R}} = [\mathbf{R}|\tilde{\mathbf{y}}]$, where $\mathbf{H} = \mathbf{QR}$ and $\tilde{\mathbf{y}} = \mathbf{Q}^*\mathbf{y}$. Consequently, $\tilde{\mathbf{y}}$ is generated as a by-product. Then, carrying out the operations to puncture a given entry, these operations are also applied on the rightmost column of $\tilde{\mathbf{R}}$.

6. COMPLEXITY ANALYSIS

Table 1 summarizes the redundant QRD computations that can be saved in the efficient implementation, depending on the permutations and their order, for a 4×4 MIMO system. Note that this setup of permutations and their order is not unique, as other setups can result in the same savings. We analyze the complexity in terms of floating-point operations (flops) based on real multiplication (RML) and addition (RAD). Real division and square-root operations are assumed equivalent to a RML. Also, complex multiplication requires 4 RML and 2 RAD, while complex addition requires 2 RAD. The complete QRD requires a total of 304 RML and 176 RAD, and the savings in the permutation robust QRD are 88 RML and 40 RAD.

The reference CYSD with cyclic permutations does not allow further savings, since all column positions are altered from one permutation to another. However, parallelism is an inherent feature in it, where the process on each layer can run on a separate core. If we discard this parallelism, and use a pipelined architecture, the decomposition output from one layer can be fed to the subsequent layer, allowing computational savings. A two-stage architecture for PWSD is shown in Fig. 4. The first and third channel permutations can execute in parallel, using the efficient implementation of Sec. 5, but with no redundant computations to save. The LLRs of their corresponding layers are sent to a buffer, and the WRD output is passed to the next stage, to assist the WRD of the second and fourth permutations, respectively. A PR-WRD is thus applied in the second stage, making use of previous decompositions.

For simplicity, the proposed algorithms were illustrated for 4×4 MIMO systems. However, their impact is better achieved in higher order systems, 16×16 and 32×32 MIMO for example. With such systems, the total preprocessing QRD overhead can be reduced to almost 50%, since the non-redundant computations in the second stage will only correspond to a slice of 2 columns. Moreover, these savings are more profound with LORD detectors, where the matrix preprocessing consists of QRD only.

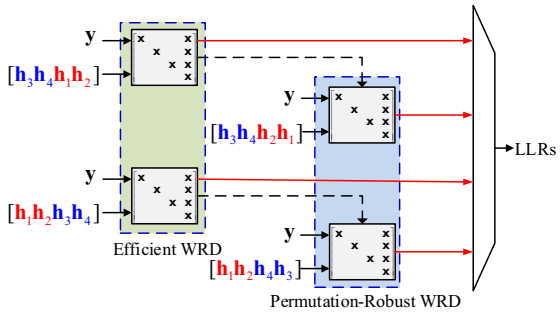

Fig. 4. Architecture for a 4×4 MIMO PWSD

7. SIMULATION RESULTS

The implementation followed the system model of Sec. 2. Turbo coding/decoding is used, with a code rate of $1/2$ and 8 decoding iterations. Moreover, we considered a zero-mean complex Gaussian circularly symmetric channel, that is independent and identically distributed from time to time, with unit variance, which corresponds to a case of rich scattering.

Figure 5 shows the bit error rate (BER) performance of the two proposed detection approaches, compared to that of CYSD, and the linear ZF detector. Note that the exhaustive ML detector is not shown in the results, since it has a prohibitive complexity. The PWSD and SPSD plots coincided with the reference subspace detector, which means that the savings came at no performance degradation cost. Moreover, the gap between the subspace detectors and ZF is around 3 dB, which is significant. Note that if the proposed algorithms are combined with layer ordering techniques, we could achieve both, complexity savings and performance enhancement.

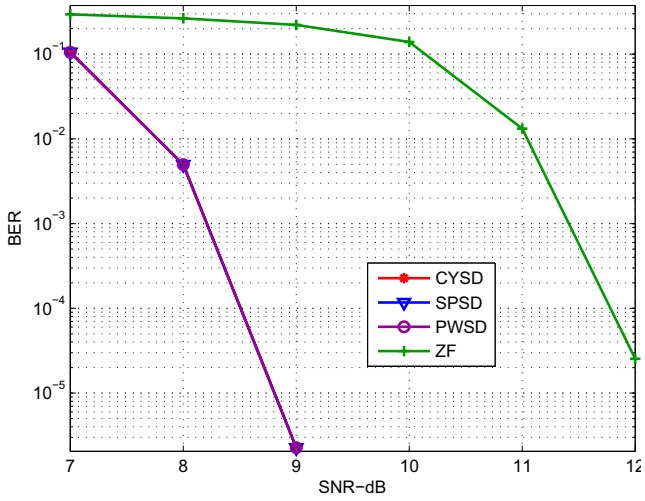

Fig. 5. BER performance of MIMO detectors with 16QAM

8. CONCLUSION

In this paper, two low-complexity MIMO subspace detection algorithms have been proposed and tested in a 4×4 MIMO scenario. It has been shown that with no performance degradation cost, subspace detection can be efficiently implemented. The proposed implementation avoids matrix inversion operations, and employs a number of permutation-robust matrix decomposition techniques. A two-stage architecture has been proposed, which allows the significant reduction in QRD overhead (almost 50% in very high order MIMO).

9. REFERENCES

[1] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[2] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, 2005.

[3] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1639–1242, July 1999.

[4] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005.

[5] H. Vikalo and B. Hassibi, "On the sphere-decoding algorithm II. Generalizations, second-order statistics, and applications to communications," *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 2819–2834, Aug. 2005.

[6] M. M. Mansour, S. Alex, and M. Jalloul, "Reduced complexity soft-output MIMO sphere detectors – Part I: Algorithmic optimizations," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5505–5520, Nov. 2014.

[7] M. M. Mansour, S. Alex, and M. Jalloul, "Reduced complexity soft-output MIMO sphere detectors – Part II: Architectural optimizations," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5521–5535, Nov. 2014.

[8] Y. Jiang, J. Li, and W. W. Hager, "Joint transceiver design for MIMO communications using geometric mean decomposition," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3791–3803, Oct. 2005.

[9] Y. Jiang, J. Li, and W. W. Hager, "Uniform channel decomposition for MIMO communications," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4283–4294, Nov. 2005.

[10] J. Zheng, E. Ojard, and J. Kim, "Subspace beamforming for near-capacity MIMO performance," *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5729–5733, Nov. 2008.

[11] Y. Chen and S. Brink, "Near-capacity MIMO subspace detection," *Proc. IEEE Int. Symp. Personal Indoor and Mobile Radio Commun. (PIMRC)*, Toronto, Canada, pp. 1733–1737, Sep. 2011.

[12] M. Siti and M. P. Fitz, "A novel soft-output layered orthogonal lattice detector for multiple antenna communications," *Proc. IEEE Int. Conf. Commun. (ICC)*, Istanbul, Turkey, vol. 4, pp. 1686–1691, Jun. 2006.

[13] M. M. Mansour, "A near-ML MIMO subspace detection algorithm," *IEEE Signal Process. Lett.*, vol. 22, no. 4, pp. 408–412, Apr. 2015.

[14] M. M. Mansour, "A Low-Complexity MIMO Subspace Detection Algorithm," *EURASIP Journal on Wireless Communications and Networking*, Mar. 2015.

[15] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996.