A Grouped-Iterative Framework for MIMO Detection

Di-You Wu and Lan-Da Van
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
E-mail: dywu@viplab.cs.nctu.edu.tw, ldvan@cs.nctu.edu.tw

Abstract—**In this paper, we propose one grouped-iterative framework to generate a family of the MIMO detection algorithms. The presented framework not only includes the conventional iterative, grouped, and Chase detection algorithms, but also derives a new low-complexity MIMO detection algorithm. The proposed detection can adjust some parameters to achieve a range of trade-offs between performance and complexity. In (4,4) system with uncoded 16-QAM inputs, one instance of the proposed detection algorithm not only substantially reduces the multiplication complexity by 26.3% but also outperforms the BODF algorithm about 5dB. Another instance of the proposed algorithm can save multiplication complexity by 34% at the penalty of 1 dB loss compared with the B-Chase detector.**

I. Introduction

Multiple-Input-Multiple-Output (MIMO) technology can significantly improve data transmission rate in the limited bandwidth wireless communications without increasing transmitted power. Previous researchers have shown that the channel capacity increases with the number of antennas [1], [2]. Because of the above benefit, the MIMO technique has been considered in modern high-speed wireless communication standard including wireless LAN and mobile wireless MAN. For MIMO communication systems, the detection scheme is more complex than that in Single-Input-Single-Output (SISO) communication systems. Since the MIMO communications transmit information at very high data rate, reasonable computational complexity of the detection algorithm using in the receiver is essentially considered. In terms of bit-error rate (BER) performance, the maximum likelihood (ML) detection scheme is an optimum solution for the MIMO detection. However, it is manifest that the detection complexity would be raised rapidly as the number of antennas and the constellation size increases. Therefore, the computational complexity of the ML scheme is too high to be applied to real-time communications. Many researchers currently concentrate on developing low-complexity detection algorithms [3], [4] to reduce the computational complexity and save power for larger number of antenna systems. The Bell Laboratories layered space-time (BLAST) wireless communication system [1] uses multi-element antenna arrays at both the transmitter and receiver to achieve high spectral efficiency. This technology is referred to as the diagonal BLAST (DBLAST). The DBLAST theoretically approaches the Shannon capacity for multiple transmitters and receivers, but the DBLAST still possesses high computational complexity. A simplified architecture known as vertical BLAST (VBLAST) and a detection algorithm called BLAST-ordered decision feedback (BODF) has been proposed in [3]; however, the BER performance will be largely degraded. Until now, some research uses grouped interference suppression (GIS) [5] technique to divide system into two lower dimensional sub-systems. The most important sub-system applies the ML detection and another employs a suboptimal algorithm with lower complexity. Although the previously published method using the ML and BODF detection schemes can improve performance improvement, the high complexity is still remained. Thus, we are motivated to devise a MIMO detection algorithm that possesses the features of the low complexity and satisfactory performance.

This work was supported in part by the MOEA96-EC-17-A-01-S1-048, National Science Council (NSC) Grant NSC-96-2221-E-009-220, and NSC-96-2220-E-009-038. The authors are with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

This paper is organized as follows: Brief review of MIMO detection algorithms is described in Section II. In Section III, one generalized grouped-iterative framework is presented, and shows how to generate existing algorithms through this framework. In Section IV, we propose a new detection algorithm that belongs to this framework. The complexity analysis and simulation results are presented in Section IV. Last, the conclusion is given.

II. Brief Review of MIMO Detection Algorithms

A MIMO system with N transmit antennas and M receive antennas is considered in this paper. The discrete-time received signal $\mathbf{r}$ can be written as

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

where $\mathbf{s}$ is the $N\times 1$ vector of the simultaneous transmitted symbols that selecting from constellation C, and we denotes $|C|$ is the constellation size. $\mathbf{H}$ is the $M\times N$ equivalent channel transfer matrix, $\mathbf{n}$ is the $M\times 1$ complex white Gaussian noise vector with zero mean and variance of σ_n^2. In this paper, the elements in $\mathbf{H}$ are assumed to be independent identically distributed (IID) complex Gaussian random variable with zero mean. It is assumed that the receiver knows channel matrix $\mathbf{H}$ perfectly. This is known that the ML detector is an optimum solution for the receiver in which the scheme detects all sub-stream symbols jointly by choosing the symbol vector, which maximizes likelihood function. However, the higher computational complexity of the ML scheme blocks the VLSI implementation. Several low-complexity detection algorithms [3-10] have been widely studied. Herein, we briefly review the complexity-oriented algorithms as follows.

978-1-4244-1722-3/08/$25.00 ©2008 IEEE.

A. Grouped Detection

The group detection algorithm [5] consists of the ordering, GIS, ML algorithm for the first group symbols, interference canceling (IC), and BODF algorithm for the second group symbols. The GIS plays the role of dividing symbols into two groups and suppressing the performance influence with the low SNR signals. After ordering symbols, the ML detection algorithm is employed at the first group to detect higher SNR signals. Because of the property of the ML algorithm, we can detect symbols at early stage and guarantee the performance without error propagation. Although the remaining symbols detected at the second group disturbed by high noise power can be performed in a suboptimal way like linear filtering, the difference to the ML approach is very small in this case. Thus, the suboptimal algorithms can be resulted such as the BODF [3], [4] detection algorithms.

B. Iterative Detection

The iterative detection algorithm proposed in [6], [7] iterates the BODF algorithm recursively. At the second round of the BODF algorithm, it reverses the detection order of the BODF algorithm at first around to retrieve the high diversity gain. Enhancing diversity for all symbols can decrease error propagation.

C. Chase Detection

The Chase detection [9] based on the chase algorithm determines which symbol detected first, list length, and sub-detector algorithm for MIMO detection application. Many detection algorithms including the ML, BODF, and B-chase can be derived from the chase algorithm by adjusting the above three parameters. The B-Chase detector provides a tradeoff between the complexity and performance by choosing the list length. When the list length equals the constellation size, the performance of the B-Chase detection is close to that of the ML detection. Although the sphere decoder (SD) [10] has better performance than that of the above chase detectors, the SD needs larger and uncertain computational complexity.

III. A Generalized Grouped-Iterative MIMO Detection Framework

In this section, we propose a grouped-iterative framework as shown in Fig. 1 such that several previously reported detection algorithms can be fit into. As we know that the grouped detection outperforms iterative detection in high SNR environment; on the other hand, the grouped detection has weaker performance than that of the iterative detection in low SNR environment. We are motivated to combine both algorithms to attain better performance. Herein, the proposed framework not only includes the existing conventional grouped and iterative detection algorithms, but also derives a new scheme. In order to improve performance, we list more candidates to look for more possible solutions. The proposed framework is introduced as follows.

Step1: Order the sequence of detecting symbols, and divide all symbols into two groups. Group I has *K* symbols $\{s_{n_1}, s_{n_2}, \cdots, s_{n_k}\}$ of the highest order, and the residual (*N*-*K*) symbols $\{s_{n_{k+1}}, s_{n_{k+2}}, \cdots, s_{n_N}\}$ are allocated into group II, where $\{n_1, n_2, \ldots, n_N\}$ denotes the detection order index.

Step2: Make a list of partial candidates $\{s'_{I_1}, s'_{I_2}, \cdots, s'_{I_\ell}\}$ by ordering in terms of minimum mean squared error, where s'_{I_i} consists of *K* symbols $\{s'_{i,n_1}, s'_{i,n_2}, \cdots, s'_{i,n_k}\}$.

Step3: Cancel the interference of $\mathbf{r}$ from the *K* symbols s'_{I_i} to derive $\mathbf{r}'_i$, and determine the remaining (*N*-*K*) symbols $s_{II_i} = [s_{i,n_{k+1}}, s_{i,n_{k+2}}, \cdots, s_{i,n_N}]^T$, where x^T denotes the transpose of x.

Step4: Cancel the interference of $\mathbf{r}$ from the (*N*-*K*) symbols s_{II_i} to derive $\mathbf{r}''_i$, and determine the *K* symbols $s_{I_i} = [s_{i,n_k}, s_{i,n_{k-1}}, \cdots, s_{i,n_1}]^T$.

Step5: Determine whether the iterative operation is needed by detection algorithm. If iterating is selected, continue to update values of parameters. The equation of decision is presented in Table I if the detection needed. The iterative decision feedback (IDF) consists of steps 3~5 as shown in Fig. 1.

Step6: Choose the best hard decision $\tilde{s}$ among the candidates $\{\tilde{s}_{n_1}, \tilde{s}_{n_2}, \cdots, \tilde{s}_{n_\ell}\}$, and reorder $\tilde{s}$ into $\mathbf{s}$.

To configure different grouped-iterative detectors, several parameters will be used and defined in the following.

- *K*: The number of symbols in group I. ($1 \leqq K < N$).
- Sub-algorithm1, 2, and 3: The detection algorithm used in step 2, 3, and 4, respectively.
- ℓ: The list length. ($1 \leqq \ell \leqq |C|^K$)
- *MI*: The maximum number of iteration. ($0 \leqq MI$)

Table I summarizes how the BODF, grouped, iterative, and B-Chase detector are specified in the group-iterative detection framework using these parameters. Identity means that the symbols are bypassed at this stage and fed to the next step. The new proposed detector will be illustrated in the next section.

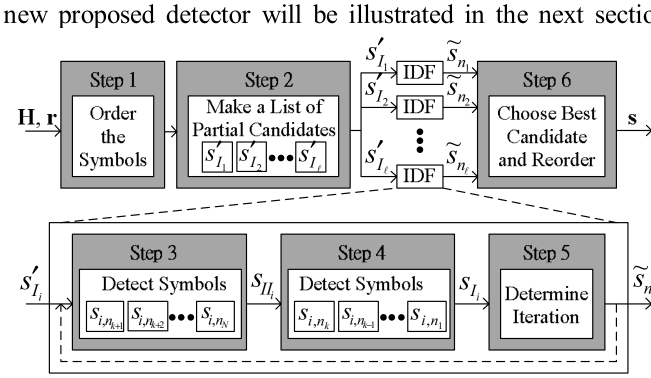

Fig. 1. Flowchart of grouped-iterative framework.

IV. A New Grouped-Iterative Detector

In this section, we explore the above framework by

TABLE I

CASES OF GROUPED-ITERATIVE FRAMEWORK FOR MIMO DETECTION

Detector	The number of symbols in group I : K	Algorithm used in Step 2 (Sub-Algo.1)	List length ℓ	Algorithm used in (Step 3, Step 4) (Sub-Algo.2, Sub-Algo.3)	Iteration Determination in Step 5 (MaxIteNum=MI)
BODF [3]	$1 \leqq K < N$	BODF	1	(BODF, Identity)	No
Grouped [5]	$1 < K < N$	ML (ZF-GIS)	1	(BODF, Identity)	No
Iterative [7]	$K=(N-1)$	BODF	1	(BODF, BODF)	Yes($1 \leqq MI$) $*_1$
B-Chase [9]	$K=1$	BODF	$1 \leqq \ell \leqq \|C\|$	(BODF, Identity)	No
Proposed Work	$1 < K < N$	B-Chase (ZF-GIS)	$1 \leqq \ell \leqq \|C\|$	(SQRDF, SQRDF)	Yes($1 \leqq MI$) $*_2$ or No

$*_1$: If ($s'_{i,n_1} == s_{i,n_1}$ or IteNum(IN) $==MI$), end; else set $s'_{i,n_1} = s_{i,n_1}$ and iterate.

$*_2$: If ($\{s'_{i,n_1}, s'_{i,n_2}, \cdots, s'_{i,n_k}\} == \{s_{i,n_1}, s_{i,n_2}, \cdots, s_{i,n_k}\}$ or $IN == MI$), end; else set $\{s'_{i,n_1}, s'_{i,n_2}, \cdots, s'_{i,n_k}\} = \{s_{i,n_1}, s_{i,n_2}, \cdots, s_{i,n_k}\}$ and iterate.

configuring the parameters and then propose a new detection algorithm. After investigating the grouped-iterative framework, the configuration parameters including K, sub-algorithm1, -algorithm2, -algorithm3, ℓ, MI can be further changed to optimize the complexity and performance. In the proposed work, the ML algorithm used in step 2 at grouped detection is replaced by the B-Chase detection, where the performance of the B-Chase detection is close to that of the ML algorithm with the low computational complexity. For regularity and low computational complexity, we use the sorted QR decision feedback algorithm (SQRDF) [8] in step 3, and 4 instead of the zero-forcing BODF algorithm used in grouped detection. Finally, we give a wide range of parameters K, ℓ, and MI to be changed for complexity and performance trade-off. Hence, we can largely reduce complexity through the presented configurations as shown in Table 1. The detailed operations at each design step of the proposed MIMO detection algorithm are described as follows.

Step 1: In this step, we sort the detection order firstly by values of signal to noise ratio (SNR) in order to early detect higher SNR signals.

$$p_i = \| \mathbf{h}_i \|^2 \quad \text{for } i = 1, 2, \ldots, N, \tag{2}$$

where $\mathbf{h}_i$ is the i-th column of $\mathbf{H}$. According to the value of each p_i, we can sort the values and obtain (3)

$$p_{n_1} \geq p_{n_2} \geq \ldots \geq p_{n_N} \ , \tag{3}$$

where $\{n_1, n_2, \ldots, n_N\}$ denotes the detection order index.
We reorder all symbols $\mathbf{s}$ and the channel matrix $\mathbf{H}$ into $\tilde{s} = \{s_{n_1}, s_{n_2}, \cdots, s_{n_N}\}$ and $\tilde{\mathbf{H}} = [\mathbf{h}_{n_1}\ \mathbf{h}_{n_2} \cdots \mathbf{h}_{n_N}]$. The system will be changed to

$$\tilde{\mathbf{r}} = \tilde{\mathbf{H}}\tilde{s} + \tilde{\mathbf{n}}\,. \tag{4}$$

According to the number of K, $\tilde{s}$ can be divided into two groups $s_I = [s_{n_1}, s_{n_2}, \cdots, s_{n_k}]^T$ and $s_{II} = [s_{n_{k+1}}, s_{n_{k+2}}, \cdots, s_{n_N}]^T$.

Step 2: In order to reduce complexity, we divide original system into two lower dimensional sub-systems through applying the GIS technique. And we modify the computation of the ZF-GIS [5] to possess lower complexity. Without loss of the generality, we illustrate the computation in (4,4) MIMO system and set K=N/2 at this step. In this case, the ordered channel matrix $\tilde{\mathbf{H}}$ can be written as

$$\tilde{\mathbf{H}} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} = [\mathbf{h}_{n_1}\ \mathbf{h}_{n_2}\ \mathbf{h}_{n_3}\ \mathbf{h}_{n_4}] = [\mathbf{H}'\ \mathbf{H}''], \tag{5}$$

where $\mathbf{H}' = [\mathbf{h}_{n_1}\ \mathbf{h}_{n_2} \cdots \mathbf{h}_{n_k}]$ and $\mathbf{H}'' = [\mathbf{h}_{n_{k+1}}\ \mathbf{h}_{n_{k+2}} \cdots \mathbf{h}_{n_N}]$. In the proposed detection algorithm, we employ the matrix $\mathbf{H}_{\text{br}}$ to obtain a left null matrix $\mathbf{L}$ of $\mathbf{H}''$. The left null matrix $\mathbf{L}$ and the matrix $\mathbf{H}_{\text{br}}$ on the bottom right corner of $\tilde{\mathbf{H}}$ can be respectively defined in (6) and (7).

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & A & B \\ 0 & 1 & C & D \end{bmatrix}, \tag{6}$$

and

$$\mathbf{H}_{br} = \begin{bmatrix} h_{33} & h_{34} \\ h_{43} & h_{44} \end{bmatrix}, \tag{7}$$

where A, B, C, D can be calculated via the following matrix computation.

$$\begin{bmatrix} A \\ B \end{bmatrix} = (\mathbf{H}_{br}^T)^{-1} \begin{bmatrix} -h_{13} \\ -h_{14} \end{bmatrix}, \tag{8a}$$

$$\begin{bmatrix} C \\ D \end{bmatrix} = (\mathbf{H}_{br}^T)^{-1} \begin{bmatrix} -h_{23} \\ -h_{24} \end{bmatrix}. \tag{8b}$$

By this computation, we can quickly retrieve the left null matrix $\mathbf{L}$ and then apply Gram-Schmidt orthogonalization to $\mathbf{L}$. Then $\mathbf{L}$ is multiplied on both sides of (4). We can derive the following equation as

$$\hat{\mathbf{r}} = \hat{\mathbf{H}}s'_I + \hat{\mathbf{n}}\,, \tag{9}$$

where $\hat{\mathbf{n}} = \mathbf{L}\tilde{\mathbf{n}}$ and $\hat{\mathbf{H}} = \mathbf{L}\mathbf{H}'$ with dimension of $K \times K$. After the operation of the ZF-GIS, we use the B-Chase detector to detect the sub-system in (9) with choosing best ℓ candidates which have minimum mean squared error. Then we can derive an ordered list of partial candidates $\{ s'_{I_1}, s'_{I_2}, \cdots, s'_{I_\ell} \}$.

Steps 3, 4, and 5: It is convenient for introduce steps 3, 4 and 5 together, and just describe the operation in *i*-th iterative decision feedback (IDF). At step 3 of the proposed work, we apply the SQRDF algorithm to detect sub-system in (10).

$$\mathbf{r}'_i = \mathbf{r} - \mathbf{H}' s'_{I_i} = \mathbf{H}'' s_{II_i} + \mathbf{n}', \quad (10)$$

We divide the algorithm into two parts: sorted QR decomposition (SQRD) and decision feedback (DF). After the SQRD of $\mathbf{H}''$, we can derive $\mathbf{H}'' = \mathbf{Q}''\mathbf{R}''$, where $\mathbf{Q}''$ denotes unitary matrix and $\mathbf{R}''$ represents upper triangular matrix with positive and real diagonal elements. The partial operation of the SQRD can be obtained from (2) such that we can save above computation at this stage. The systems can be changed to (11).

$$\mathbf{y}''_i = \mathbf{Q}''^{*}\mathbf{r}'_i = \mathbf{Q}''^{*}\mathbf{r} - \mathbf{Q}''^{*}\mathbf{H}' s'_{I_i} = \mathbf{R}'' \bar{s}_{II_i} + v'', \quad (11)$$

where $\bar{s}_{II_i} = [\bar{s}_{i,n_{k+1}}, \bar{s}_{i,n_{k+2}}, \cdots, \bar{s}_{i,n_N}]^T$ is ordered from s_{II_i}, where x^* denotes the conjugate transpose of x. We can obtain $\bar{s}_{II_i}$ from DF operation as follows:

$$\bar{s}_{i,n_b} = \mathbf{dec}\left(\frac{\mathbf{y}''_{i,b-k} - \sum_{j=b-k+1}^{N-k} \mathbf{R}''_{b-k,j} \bar{s}_{i,n_{j+k}}}{\mathbf{R}''_{b-k,b-k}} \right),$$
$$\text{for } b = N, N-1, \cdots, k+1, \quad (12)$$

where $\mathbf{dec}(x)$ denotes the quantization function which quantizes the value x to the nearest constellation point. The symbols s_{II_i} can be obtained by reordering $\bar{s}_{II_i}$. In similar manner, at step 4, we can obtain following equations from (13) to (15).

$$\mathbf{r}''_i = \mathbf{r} - \mathbf{H}'' s_{II_i} = \mathbf{H}' s_{I_i} + \mathbf{n}''. \quad (13)$$

$$\mathbf{y}'_i = \mathbf{Q}'^{*}\mathbf{r}''_i = \mathbf{Q}'^{*}\mathbf{r} - \mathbf{Q}'^{*}\mathbf{H}'' s_{II_i} = \mathbf{R}' \bar{s}_{I_i} + v'. \quad (14)$$

$$\bar{s}_{i,n_c} = \mathbf{dec}\left(\frac{\mathbf{y}'_{i,k-c+1} - \sum_{j=k-c+2}^{k} \mathbf{R}'_{k-c+1,j} \bar{s}_{i,n_{k-j+1}}}{\mathbf{R}'_{k-c+1,k-c+1}} \right),$$
$$\text{for } c = 1, 2, \cdots, k, \quad (15)$$

where $\mathbf{H}' = \mathbf{Q}'\mathbf{R}'$, and $\bar{s}_{I_i} = [\bar{s}_{i,n_k}, \bar{s}_{i,n_{k-1}}, \cdots, \bar{s}_{i,n_1}]^T$ is ordered from s_{I_i}. Observing (11) and (14), we can reuse tentative calculations for parallel and iterative computing such that we just compute the SQRD of $\mathbf{H}'$ and $\mathbf{H}''$, $\mathbf{Q}''^{*}\mathbf{r}$, $\mathbf{Q}''^{*}\mathbf{H}'$, $\mathbf{Q}'^{*}\mathbf{r}$, and $\mathbf{Q}'^{*}\mathbf{H}''$ once. The maximum iterative (*MI*) number will affect the computational complexity. The initial iterative number (*IN*) is set to zero. When executing step 4 once, *IN* is increased by one. If s'_{I_i} equals s_{I_i} or *IN* equals *MI*, we obtain the candidate $\tilde{s}_{n_i} = [s_{I_i}, s_{II_i}]^T$ or $\tilde{s}_{n_i} = [s'_{I_i}, s_{II_i}]^T$ when *MI*=0. Else, let $s'_{I_i} = s_{I_i}$ and repeat steps 3, and 4. Note that if *MI*=0, (13)~(15) can be skipped.

Step 6: In the last step, we choose the final hard decision $\tilde{s}$ with the minimum mean squared error among the candidates $\{ \tilde{s}_{n_1}, \tilde{s}_{n_2}, \cdots, \tilde{s}_{n_\ell} \}$. The minimum mean squared error of each candidate can be attained via pruning and threshold-tightening strategy and temporary variance in the DF operation given in [9]. According to the detection order sequence at step 1, we rank the detected symbols $\tilde{s}$ to obtain the final symbols $\mathbf{s}$.

V. Complexity Analysis and Simulation Results

This section explores the complexity and performance of the proposed detector and compares with the existing detection scheme. We define the complexity measurement in terms of number of complex multiplications required in the worst case. We ignore the number of divisions and square roots because it is small compared with the number of multiplies. The multiplication of a number and a constellation point can be implemented by scaled integers [11] such that we can reduce the multiplications. For simplicity, we set *K*=*N*/2 and *M*=*N*. The comparisons of the worst computational complexity of the proposed detection, and the B-Chase detection schemes are tabulated in Table 2. On the other hand, we show simulation results to sustain the performance of the proposed detection algorithm, and the simulation environment is assumed Rayleigh flat-fading channel and no correlation between sub-channels. Figs. 2 and 3 show the BER performance of the proposed detection and the existing detections. The performance measurement targets at the SNR that reaches BER=10^{-3}.

From the comparison results, we can find out the performance of the proposed work can be improved as ℓ or *MI* increases. Even slightly increasing the list length ℓ, BER performance can be significantly enhanced. For example, the proposed work (*K*=2, ℓ=2, *MI*=0) outperforms the proposed work (*K*=2, ℓ=1, *MI*=0) by about 4dB with respect to QPSK and 16-QAM inputs in (4,4) system, and just increases complexity 2.2% and 1.7%. The better performance gains in longer list length; the proposed work (*K*=2, ℓ=16, *MI*=0) outperforms the proposed work (*K*=2, ℓ=1, *MI*=0) by about 7dB with 16-QAM inputs. On the other hand, better BER performance can be obtained by increasing *MI* under the same list length. For instance, the proposed work (*K*=2, ℓ=1, *MI*=1) outperforms the proposed work (*K*=2, ℓ=1, *MI*=0) by 1~2dB with QPSK inputs.

Next, we begin to show how the proposed detection attains better complexity-performance trade-off at the slight penalty of BER performance degradation compared with the B-Chase detection. Note that we use the B-Chase detection as a reference instead of the ML or SD detections due to higher computational complexity. Our proposed work (*K*=2, ℓ=1, *MI*=0) not only reduces the complexity of 41.5% and 26.3% but also outperforms about 5dB compared with the BODF (B-Chase (ℓ=1)) with respect to QPSK and 16-QAM inputs, respectively, in (4,4) system. The other proposed work (*K*=2, ℓ=16, *MI*=0) and proposed work (*K*=2, ℓ=4, *MI*=0) reduce

TABLE II
THE COMPARISON OF COMPUTATIONAL COMPLEXITY

Detector	Number of Complex Multiplications
B-Chase	$\left[\frac{11}{3}N^3+5N^2+\frac{1}{3}N+2\ell N\right]*_1$
Proposed Work	$\left[\frac{17}{8}N^3+\frac{9}{2}N^2+\left(\lvert C\rvert-\frac{1}{2}\right)N+\left(MI+\frac{1}{2}\right)\ell N\right]$ or $\left[\frac{13}{8}N^3+\frac{31}{8}N^2+\left(\lvert C\rvert-\frac{1}{4}\right)N+\ell N\right]*_2$

$*_1$The computational complexity when ℓ equals to $\lvert C\rvert$.

$*_2$The computational complexity when MI equals to zero.

34% and 44.8% complexity while falling 1dB and 2dB short of the B-Chase (ℓ =16) detector with 16-QAM inputs respectively. We do not present the comparison with the grouped and iterative detection here since both algorithms require more computational complexity compared with the BODF algorithm and possess worse BER performance than the corresponding cases of the proposed work.

VI. CONCLUSION

In this work, a low-complexity framework of MIMO detection algorithm using grouped and iterative approach for MIMO communications has been presented. Based on the grouped-iterative framework, we propose new low-complexity detection that trade-offs the complexity and performance by modifying the list length and the number of maximum iteration. The proposed detection significantly reduces the multiplication complexity and has comparable BER performance compared with the existing detections. For example, in (4,4) system with 16-QAM inputs, the proposed work (K=2, ℓ =1, MI=0) can reduce the multiplication complexity by 26.3% and outperform about 5dB compared with the BODF detection algorithm at low complexity end. At high performance end, the proposed work (K=2, ℓ =16, MI=0) can reduce the multiplication complexity by 34% at the penalty of 1dB loss compared with the B-Chase (ℓ =16) detection.

REFERENCES

[1] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multiple antennas," *Bell Labs Tech. J.*, vol. 1, no. 2, pp. 41-59, Autumn 1996.

[2] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *Eur. Trans. Telecommum.*, vol.10, no. 6, pp. 585-595, Nov. 1999.

[3] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel," in *Proc. ISSSE*, 1998, pp. 295–300.

[4] B. Hassibi, "An efficient square-root algorithm for BLAST," in *Proc. IEEE Int.Conf. Acoustics,Speech, Signal Processing*, vol. 2, Jun. 2000, pp. 737-740.

[5] L. Yang, M. Chen, S. Cheng, and H. Wang, "Combined maximum likelihood and ordered successive interference cancellation grouped detection algorithm for multistream MIMO," in *Proc.IEEE Eighth Int. Sym. on Spread Spectrum Tech. and Appl.*, 30 Aug.-2 Sept. 2004, pp. 250-254.

[6] Cong Shen, Hairuo Zhang, Lin Dai, and Shidong Zhou, "Detection algorithm improving V-BLAST performace over error propagation" *Electronic Letters*, vol. 39, no. 13, pp. 1107-1108, June 2003.

[7] D. Li, L. Cai and H. Yang, "New iterative detection algorithm for V-BLAST," *IEEE Veh. Technol. Conf.*, vol. 4, Sept. 2004, pp.2444-2448.

[8] D. Wübben, R. Böhnke, J. Rinas, V. Kühn and K. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *Electronic Letters*, vol.37, no. 22, pp.1348-1350, Oct 25, 2001.

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

[10] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[11] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, ''VLSI implementation of MIMO detection using the sphere decoding algorithm,'' *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp.1566–1577, Jul. 2005.

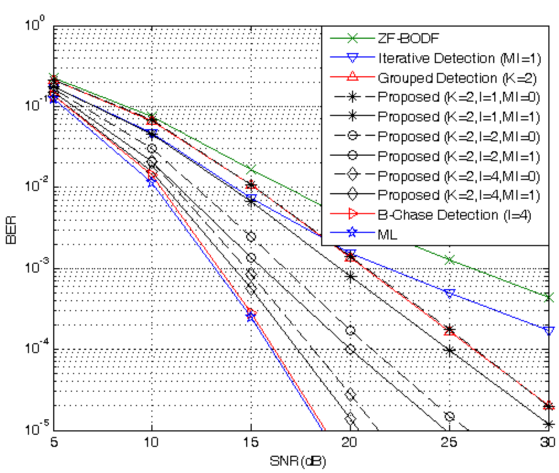

Fig. 2. BER comparison for (4,4) MIMO system with QPSK inputs.

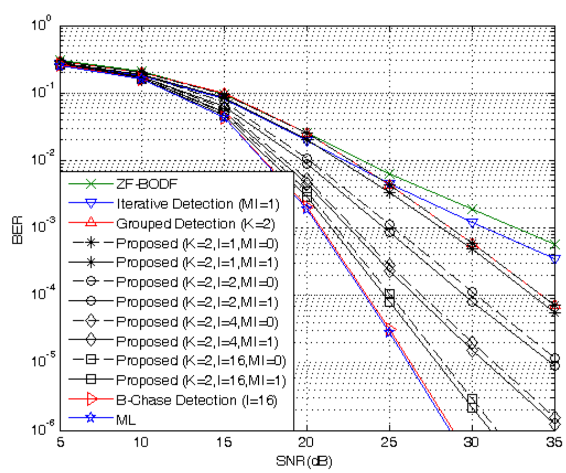

Fig. 3. BER comparison for (4,4) MIMO system with 16-QAM inputs.