A Near-Optimal Detection Scheme Based on Joint Steepest Descent and Jacobi Method for Uplink Massive MIMO Systems

Xiaobo Qin, Zhiting Yan, and Guanghui He, *Member, IEEE*

***Abstract*—A new approach based on joint steepest descent algorithm and Jacobi iteration is proposed to iteratively realize linear detections for uplink massive multiple-input multiple-output (MIMO) systems. Steepest descent algorithm is employed to obtain an efficient searching direction for the following Jacobi iteration to speed up convergence. Moreover, promising initial estimation and hybrid iteration are utilized to further accelerate the convergence rate and reduce the complexity. Simulation results show that the proposed method outperforms Neumann Series, Richardson method, and conjugate gradient based methods, while achieving the near-optimal performance of linear detectors with a small number of iterations. Meanwhile, the FPGA implementation results demonstrate that our proposed method can achieve high throughput owing to its high parallelism.**

***Index Terms*—Massive MIMO, steepest descent, Jacobi iteration, matrix inversion.**

I. Introduction

MASSIVE multiple-input multiple-output (MIMO) is an emerging technology for the future wireless standards (e.g. the fifth generation cellular systems), due to its high spectral efficiency [1]. The basic idea behind the massive MIMO systems is that numerous antennas at the base station (BS) serve for multi-users. Because of the redundancy of the BS antennas, simple linear detection algorithms, for example, zero-forcing (ZF) and minimum mean square error (MMSE), can achieve near-optimal performance [2]. Unfortunately, these linear detection methods are involved with complex matrix inversion, especially when the dimension of massive MIMO systems is going large.

Recently, several approximated MMSE detection approaches are proposed to avoid exact massive MIMO matrix inversion [3]–[6]. However, truncated Neumann series (NS) approximation suffers from significant performance loss when massive MIMO scales up and marginal complexity is reduced [3]. Richardson method is proposed to achieve a near-optimal performance [4], but it requires large numbers of iterations. In addition, Conjugate gradient (CG) based algorithms still require a large number of iterations and involve many divisions [5]. To further reduce the iteration number, the approach based on Gauss-Seidel (GS) [6] is proposed and only a small number of iterations is needed to achieve an even better performance. Unfortunately, the internal iterations within each GS iteration make it hard for parallel implementation.

Manuscript received July 25, 2015; accepted November 17, 2015. Date of publication December 1, 2015; date of current version February 12, 2016. This work was supported by the National Natural Science Foundation of China under Grant 61306026. The associate editor coordinating the review of this paper and approving it for publication was C. Masouros.

The authors are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: guanghui.he@sjtu.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2015.2504506

Contributions: In this letter, a low-complexity near-optimal approach is proposed based on joint steepest descent algorithm and Jacobi iteration. In the proposed scheme, steepest descent provides an efficient searching direction for Jacobi iteration, resulting an even better performance than both of them. Moreover, two optimization strategies including proper initial estimation and hybrid iteration are proposed to further improve the performance. The numerical results verify that the proposed algorithm outperforms NS, RI and CG in terms of bit error rate (BER), achieving near-optimal performance. The FPGA implementation results show that the proposed method can achieve high throughput due to its high parallelism.

The remainder of this letter is organized as follows. Section-II reviews the massive MIMO model and the details of the proposed scheme are depicted in Section-III. Then, Section-VI shows the numerical results of the proposed algorithm. Finally, the conclusions are drawn in Section-V.

Notation: $(\cdot)^T$, $(\cdot)^H$, $(\cdot)^{-1}$ and $|\cdot|$ denote the transpose, conjugate transpose, matrix inversion and absolute value operator, respectively. $\mathbf{I}_N$ denotes the N dimensional unit diagonal matrix. x_i and $\mathbf{x}^{(i)}$ are the i-th entry of $\mathbf{x}$ and the result of i-th iteration, respectively. A_{ij} is the i-th row, j-th column element of matrix $\mathbf{A}$.

II. System Model

Consider the uplink of a massive MIMO system equipped with B antennas at BS and U single-antenna users ($U \leq B$, e.g. $U = 16$, $B = 128$). The U users encode their own information bits and map to the constellation points in the finite alphabet $\mathcal{O}$ with cardinality $|\mathcal{O}|$ and average transmit power E_s per symbol. The modulated symbols are transmit over the massive MIMO channel, received by the BS. The received symbols at the BS can be modeled as

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

where $\mathbf{x} \in \mathbf{C}^U$ and $\mathbf{y} \in \mathbf{C}^B$ denote the transmitted and received signal vector, respectively. The entries of the flat Rayleigh fading channel matrix $\mathbf{H} \in \mathbf{C}^{B\times U}$ are independent and identically distributed (i.i.d.) with zero mean and unit variance [6]. In what follows, $\mathbf{H}$ is assumed perfectly estimated at BS. $\mathbf{n}$ denotes the additive white Gaussian noise (AWGN) with N_0 variance for each element. The signal-to-noise ratio (SNR) is defined as UE_s/N_0 [5].

A. MMSE Detection

The typical MMSE detection is considered for (1), which can be denoted by

$$\hat{\mathbf{x}} = \left(\mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I}_U\right)^{-1}\mathbf{H}^H\mathbf{y} = \mathbf{A}^{-1}\mathbf{b} \tag{2}$$

where $\hat{\mathbf{x}}$ is the estimated vector and $\mathbf{b} = \mathbf{H}^H\mathbf{y}$ is the matched-filter output of $\mathbf{y}$, and $\mathbf{A} = \mathbf{G} + \sigma^2\mathbf{I}_U$ is the MMSE filtering matrix which is hermitian positive definite (HPD) [4] and diagonally dominant [6] where $\mathbf{G} = \mathbf{H}^H\mathbf{H}$. The estimated variance σ^2 is assumed as N_0/E_s. To avoid the exact matrix inversion, the MMSE algorithm is converted into solving the linear equation (3) [4].

$$\mathbf{Ax} = \mathbf{b} \tag{3}$$

It is proven that MMSE algorithm (2) can achieve near-optimal detection performance for massive MIMO systems [2]. However, inverting matrix $\mathbf{A}$ is complex since the dimension is very large. For example, the typical complexity to invert a matrix is $O(U^3)$ [3]. Fortunately, the equation (3) can be solved in a iterative way [4]–[6].

B. Log-Likelihood Ratio (LLR) Generation

After the MMSE detection, LLRs can be extracted for the channel decoder. Let $\mathbf{W} = \mathbf{A}^{-1}\mathbf{G}$, and rewrite (2) as

$$\hat{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{G}\mathbf{x} + \mathbf{A}^{-1}\mathbf{H}^H\mathbf{n} = \mathbf{W}\mathbf{x} + \mathbf{A}^{-1}\mathbf{H}^H\mathbf{n} \tag{4}$$

Denote $\hat{x}_i = \mu_i x_i + e_i$, where $\mu_i = W_{ii}$ and e_i is the noise term of $\hat{x}_i$ with variance of $\nu_i^2 = \sum_{j\neq i}^{U} \left|W_{ji}\right|^2 + E_{ii}\sigma^2$ and $\mathbf{E} = \mathbf{A}^{-1}\mathbf{H}^H(\mathbf{A}^{-1}\mathbf{H}^H)^H = \mathbf{A}^{-1}\mathbf{G}\mathbf{A}^{-1}$. The LLR $L_{i,b}$ of b-th bit for the i-th user is generated as (5) based on a Gaussian approximation [3].

$$L_{i,b} = \gamma_i \left(\min_{a\in\mathcal{O}_b^0} \left| \frac{\hat{x}_i}{\mu_i} - a \right|^2 - \min_{a'\in\mathcal{O}_b^1} \left| \frac{\hat{x}_i}{\mu_i} - a' \right|^2 \right) \tag{5}$$

where and $\gamma_i = \mu_i^2/\nu_i^2$ is the signal-to-interference-plus-noise ratio (SINR) for i-th user. $\mathcal{O}_b^0$ and $\mathcal{O}_b^1$ are the symbol subsets of $\mathcal{O}$ where the b-th bit of the symbol is 0 and 1, respectively.

III. Proposed Scheme

In this section, we introduce the proposed joint algorithm, two optimization strategies and approximated LLR computation. The convergence analysis is given in the following.

A. The Proposed Joint Algorithm

It is proven that steepest descent algorithm goes along a rapidly convergent direction at the beginning of the iteration when the matrix is HPD [7]. Meanwhile, Jacobi iteration performs excellently when the matrix is diagonally dominant [8]. Therefore, to exploit the HPD and diagonally dominant property of matrix $\mathbf{A}$ simultaneously, we propose a joint algorithm adopting one steepest descent iteration to obtain an efficient searching direction for the following Jacobi iterations to achieve a fast convergence rate and near-optimal performance. Basically, the joint algorithm consists of the following three steps.

Step-1: Initialization. Set up the model of equation (3).

Step-2: Performing steepest descent algorithm once. In practice, a smaller iteration number is more favorable to reducing complexity when the performance is acceptable.

Step-3: Employing (K-1)-time Jacobi iterations. Jacobi iteration has low complexity while it is able to converges fast to exact solution because $\mathbf{A}$ is diagonally dominant and the searching direction provided by steepest descent is efficient.

B. Initial Estimation

To reduce the iteration number, instead of a zero vector assumption, a promising initial estimation of $\mathbf{x}$ in (3) is employed on the proposed algorithm in *Step-1*. Due to the diagonally dominant property of $\mathbf{A}$, the diagonal-approximating [6] can obtain considerable performance improvement while causing little additional complexity, which is denoted as (6).

$$\mathbf{x}^{(0)} = \mathbf{D}^{-1}\mathbf{b} = \mathbf{D}_{inv}\mathbf{b} \tag{6}$$

where matrix $\mathbf{D}$ is the diagonal component of $\mathbf{A}$ and $\mathbf{D}_{inv}$ is the inverse matrix of diagonal matrix $\mathbf{D}$. Obviously, the complexity to invert $\mathbf{D}$ is cheap.

C. Hybrid Iteration

The hybrid iteration is proposed to replace *Step-2* in Section III-A. Rewrite the first Jacobi iteration [8] as

$$\begin{aligned}\mathbf{x}^{(2)} &= \mathbf{D}_{inv}\left[(\mathbf{D}-\mathbf{A})\,\mathbf{x}^{(1)} + \mathbf{b}\right] \\ &= \mathbf{x}^{(1)} + \mathbf{D}_{inv}\left(\mathbf{b} - \mathbf{A}\mathbf{x}^{(1)}\right) = \mathbf{x}^{(1)} + \mathbf{D}_{inv}\mathbf{r}^{(1)}\end{aligned} \tag{7}$$

where $\mathbf{r}^{(1)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(1)}$. Suppose that $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + u\mathbf{r}^{(0)}$ is the result of the steepest descent algorithm [7], and we have

$$\begin{aligned}\mathbf{r}^{(1)} &= \mathbf{b} - \mathbf{A}\left(\mathbf{x}^{(0)} + u\mathbf{r}^{(0)}\right) \\ &= \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)} - u\mathbf{A}\mathbf{r}^{(0)} = \mathbf{r}^{(0)} - u\mathbf{p}^{(0)}\end{aligned} \tag{8}$$

where $u = \frac{\mathbf{r}^{(0)H}\mathbf{r}^{(0)}}{(\mathbf{A}\mathbf{r}^{(0)})^H\mathbf{r}^{(0)}}$. Substituting (8) and $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + u\mathbf{r}^{(0)}$ into (7), we can merge the steepest descent algorithm and first jacobi iteration into a hybrid iteration (9).

$$\mathbf{x}^{(2)} = \mathbf{x}^{(0)} + u\mathbf{r}^{(0)} + \mathbf{D}_{inv}\left(\mathbf{r}^{(0)} - u\mathbf{p}^{(0)}\right) \tag{9}$$

Therefore, within K iterations, the improved joint algorithm realizes 1-time steepest descent algorithm and K-time Jacobi iterations.

D. LLR Computation

In Section II-B, $\mathbf{W}$ and $\mathbf{E}$ must be obtained first to compute the LLRs, which again involves with the inversion of $\mathbf{A}^{-1}$. Thus, we employ an approximated method to generate LLRs without matrix inversion [6]. Concretely, we derive

$$\mathbf{W} = \mathbf{A}^{-1}\mathbf{G} \approx \mathbf{D}^{-1}\mathbf{G} \tag{10}$$

$$\mathbf{E} = \mathbf{A}^{-1}\mathbf{G}\mathbf{A}^{-1} \approx \mathbf{D}^{-1}\mathbf{G}\mathbf{D}^{-1} = \mathbf{W}\mathbf{D}^{-1} \tag{11}$$

Since $\mathbf{A}$ is diagonally dominant, $\mathbf{D}$ is utilized to approximate $\mathbf{A}$ with small error. Therefore, the matrix inversion and multiplication in (10) and (11) introduce low complexity.

The complete proposed detection scheme is depicted as the following **Algorithm 1**.

Algorithm 1. Proposed Optimized Joint Algorithm

Input: $\mathbf{H}$, $\mathbf{y}$, σ^2 and iteration number K.
Output: $L_{i,b}$,$(i = 1, 2 \cdots U, b = 1, 2 \cdots log_2^{|\mathcal{O}|})$
Initialization
1: $\mathbf{G} = \mathbf{H}^H\mathbf{H}$, $\mathbf{A} = \mathbf{G} + \sigma^2\mathbf{I}_U$, $\mathbf{b} = \mathbf{H}^H\mathbf{y}$, $\mathbf{D}_{inv} = \mathbf{D}^{-1}$;
2: $\mathbf{x}^{(0)} = \mathbf{D}_{inv}\mathbf{b}$; {Initial Estimation in Section II-B}
3: $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$;
First iteration: hybrid iteration in Section II-C
4: $\mathbf{p}^{(0)} = \mathbf{A}\mathbf{r}^{(0)}$;
5: $u = \frac{\mathbf{r}^{(0)H}\mathbf{r}^{(0)}}{\mathbf{p}^{(0)H}\mathbf{r}^{(0)}}$;
6: $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + u\mathbf{r}^{(0)} + \mathbf{D}_{inv}\left(\mathbf{r}^{(0)} - u\mathbf{p}^{(0)}\right)$;
Other iterations: Jacobi
for $(k = 2; k <= K; k++)$ **do**
7: $\mathbf{x}^{(k)} = \mathbf{D}_{inv}\left[(\mathbf{D} - \mathbf{A})\,\mathbf{x}^{(k-1)} + \mathbf{b}\right]$;
endfor
LLR computation
8: $\mathbf{W} = \mathbf{D}^{-1}\mathbf{G}$, $\mathbf{E} = \mathbf{W}\mathbf{D}^{-1}$, γ_i, μ_i, $\mathcal{O}_b^0$, $\mathcal{O}_b^1$;
9: $L_{i,b} = \gamma_i\left(\min_{a\in\mathcal{O}_b^0}\left|\frac{x_i^{(k)}}{\mu_i} - a\right|^2 - \min_{a'\in\mathcal{O}_b^1}\left|\frac{x_i^{(k)}}{\mu_i} - a'\right|^2\right)$;
Return $L_{i,b}$

E. Convergence Analysis

Suppose the exact solution of the linear equation (3) is $\mathbf{x}^{(*)} = \mathbf{A}^{-1}\mathbf{b}$, and the error after kth iteration is denoted as

$$\begin{aligned}\mathbf{x}^{(k)} - \mathbf{x}^{(*)} &= \mathbf{x}^{(k-1)} + \mathbf{D}_{inv}\left(\mathbf{b} - \mathbf{A}\mathbf{x}^{(k-1)}\right) - \mathbf{x}^{(*)} \\ &= \left(\mathbf{x}^{(k-1)} - \mathbf{x}^{(*)}\right) + \mathbf{D}_{inv}\mathbf{A}\left(\mathbf{A}^{-1}b - \mathbf{x}^{(k-1)}\right) \\ &= (\mathbf{I} - \mathbf{D}_{inv}\mathbf{A})\left(\mathbf{x}^{(k-1)} - \mathbf{x}^{(*)}\right) \\ &= (\mathbf{I} - \mathbf{D}_{inv}\mathbf{A})^{k-1}\left(\mathbf{x}^{(1)} - \mathbf{x}^{(*)}\right) \end{aligned} \tag{12}$$

The error for the hybrid iteration is similarly computed as

$$\mathbf{x}^{(1)} - \mathbf{x}^{(*)} = (\mathbf{I} - \mathbf{D}_{inv}\mathbf{A})(\mathbf{I} - u\mathbf{A})\left(\mathbf{x}^{(0)} - \mathbf{x}^{(*)}\right) \tag{13}$$

Denote $\mathbf{M} = \mathbf{I} - \mathbf{D}_{inv}\mathbf{A}$, $\mathbf{N} = \mathbf{I} - u\mathbf{A}$, and we have

$$\mathbf{x}^{(k)} - \mathbf{x}^{(*)} = \mathbf{M}^k\mathbf{N}\left(\mathbf{x}^{(0)} - \mathbf{x}^{(*)}\right) \tag{14}$$

Note that $\mathbf{A}$ is diagonally dominant HPD for massive MIMO systems and $\mathbf{D}$ is the diagonal component of $\mathbf{A}$. Thus, $\mathbf{M}$ has a matrix norm less than one [8] (as well as $\mathbf{N}$ [7]) and

$$\lim_{k\to\infty}\mathbf{M}^k\mathbf{N} = \mathbf{0} \tag{15}$$

As a consequence, the error $\mathbf{x}^{(k)} - \mathbf{x}^{(*)}$ converges to zero with increasing k and promising initial estimation. In other words, the proposed algorithm converges to the exact solution $\mathbf{x}^{(*)}$. Moreover, the hybrid iteration accelerates the convergence rate of the proposed algorithm.

IV. Numerical Results

A. BER Performance

The simulation results of the BER performance against the SNR are given to demonstrate the validity of the proposed algorithm and compare to other recently reported ones. In all simulations, we consider the 64-QAM modulation scheme, rate-1/2 convolutional code with $[133_o\ 171_o]$ polynomial and BCJR decoder with soft-bits extracted from the estimated symbols. The performances of exact matrix inversion is given as the benchmark [6]. Note that Proposed-1 corresponds to the basic joint algorithm in Section III-A, while Proposed-2 refers to **Algorithm 1** adopting initial estimation in Section III-B and hybrid iteration in Section III-C.

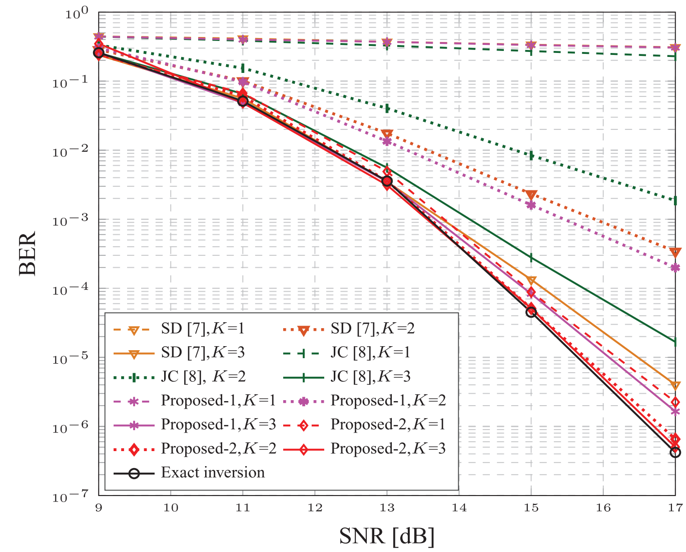

Fig. 1. The BER performance of the proposed algorithms for $B \times U = 128 \times 16$ massive MIMO systems.

1) Comparison With Steepest Descent and Jacobi Iteration: Fig. 1 shows the comparison between the proposed algorithms and the steepest descent (SD) algorithm and Jacobi iteration (JC). Given an identical K, Proposed-1 outperforms both steepest descent algorithm and Jacobi iteration, which reveals that steepest descent and Jacobi iteration benefit each other, making the joint algorithm achieve an even better performance than both of them. Meanwhile, the performance of Proposed-2 with $K = 2$ is better than that of Proposed-1 with $K = 3$, which shows a significant improvement of the two proposed optimization strategies.

2) Comparison With Other Reported Schemes: As shown in Fig. 2, the proposed algorithm (referring to Proposed-2 in what follows) converges fast to exact solution when K is large enough (e.g. $K \geq 2$). For example, when K equals to 2, the performance loss of the proposed algorithm compared with exact inversion method is within 0.1dB at BER=10^{-6}. Meanwhile, our proposed algorithm outperforms other algorithms such as NS, RI and CG given an identical iteration number.

B. Computational Complexity

The computational complexity is analyzed in terms of the number of real-valued multiplications [5]. The total computational complexity is split into two part. For the first part, the initialization step computes multiplication of $\mathbf{H}^H\mathbf{H}$, $\mathbf{H}^H\mathbf{y}$, $\mathbf{D}_{inv}\mathbf{b}$, etc. The second part comes from the K iterations. Table I shows the comparison between our proposed algorithm and other ones in the literature for $B \times U$ MIMO systems.

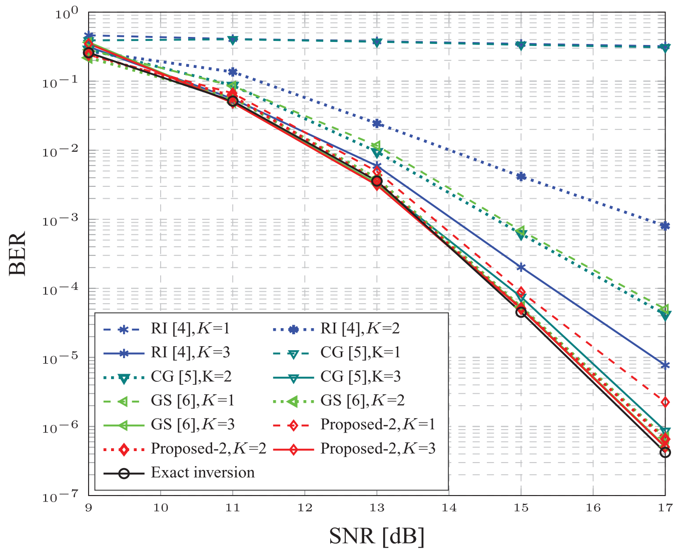

Fig. 2. The BER performance comparison for 128 ×16 MIMO systems.

TABLE I

NUMBER OF REAL-VALUED MULTIPLICATIONS

	initialization	K-time iteration
NS [3]	$2BU^2+4BU$	$(8U^3-8U^2+2U)(K-2)$ $+(4U^2-4U)(K>1)$
RI [4]	$2BU^2+4BU$	$(4U^2+2U)K$
CG [5]	$2BU^2+4BU$	$(4U^2+10U)K$
GS [6]	$2BU^2+4BU+2U$	$4U^2K$
SD [7]	$2BU^2+4BU+4U^2$	$(4U^2+8U)K$
JC [8]	$2BU^2+4BU$	$(4U^2-2U)K$
Proposed-1	$2BU^2+4BU+4U^2$	$(4U^2-2U)K+6U$
Proposed-2	$2BU^2+4BU+4U^2+2U$	$(4U^2-2U)K+10U$

Given an identical K, the proposed algorithm has approximately the same low computational complexity as RI, CG and GS. Meanwhile, compared to Proposed-1, Proposed-2 introduces marginal additional complexity since $\mathbf{D}_{inv}$ is diagonal and u is a scalar in (9).

C. FPGA Implementation

It is shown in section (IV-A) that the GS method has close BER performance to the proposed approach. However, it works in an iterative fashion within one external iteration to avoid computing the lower matrix inversion. For our proposed method, no inner loop is employed. Especially, each Jacobi iteration can enable extensive parallel computation (16) which is quite friendly to hardware implementation [8].

$$x_i^{(k+1)} = \frac{1}{A_{ii}}\left(\sum_{j=1,j\neq i}^{U} A_{ij}x_i^{(k)} + b_i\right), \mathrm{i}=1,2\ldots U \quad (16)$$

In order to verify the friendliness to hardware implementation, we implement the mentioned methods on a Virtex-7 XC7VX1140T FPGA using Xilinx Vivado High-Level Synthesis 2015. The associated fixed-point parameters are set as follows. The channel matrix, received vector and noise variance are all quantized to 16bit, while the estimated vector and intermediate values are quantized to 20bit. The basic implementation optimization approaches, including unrolling and array partitioning, are similarly employed in the implementation of different detection methods. No pipeline technique is considered here. Note that only the iteration part of the detection methods is implemented since they have approximately the same initialization step.

TABLE II

COMPARISON OF RESOURCE USAGE AND PERFORMANCE OF THE IMPLEMENTATION RESULTS

	LUTs	FFs	DSP48	Latency	Throughput[1]
Cholesky[2]	176304	274472	5456	3591	0.06M/s
NS [3]	53802	186563	6112	1356	0.15M/s
RI [4]	8586	68049	3200	112	1.78M/s
CG [5]	15706	57408	2720	375	0.53M/s
GS [6]	8602	44550	2112	1409	0.14M/s
Proposed-2	6330	28010	1312	121	1.65M/s

[1]The clock frequency is set as 200MHz for all implementations.
[2]Cholesky decomposition based exact matrix inversion.
[3]The iteration number is $K=3$ for NS, RI and CG, while $K=2$ for GS and the proposed approach according to the BER performance.

Table II shows the implementation results for 128×16 massive MIMO systems. It can be concluded that our proposed method outperforms other ones in terms of hardware consumption while obtaining high throughput. The method based on RI exhibits slightly higher throughput than the proposed algorithm, whereas it suffers from performance loss as shown in Fig. 2. The GS method obtains much lower throughput due to its iterative computation.

V. CONCLUSION

By exploiting the property of the MMSE filtering matrix, we propose a novel low-complexity near-optimal algorithm for massive MIMO detection without complicated matrix inversion. The convergence analysis for the proposed algorithm is also provided. The numerical results reveal that our detection algorithm converges to exact matrix inversion detection with a small number of iterations, while keeping low complexity. In addition, the FPGA implementation results verify that the proposed approach is hardware friendly because of its high parallelism.

REFERENCES

[1] J. Thompson et al., “5G wireless communication systems: Prospects and challenges,” *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 62–64, Feb. 2014.
[2] F. Rusek *et al.*, “Scaling up MIMO: Opportunities and challenges with very large arrays,” *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.
[3] B. Yin, M. Wu, C. Studer, J. R. Cavallaro, and C. Dick, “Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA implementations,” *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 916–929, Mar. 2014.
[4] X. Gao, L. Dai, Y. Ma, and Z. Wang, “Low-complexity near-optimal signal detection for uplink large-scale MIMO systems,” *Electron. Lett.*, vol. 50, no. 18, pp. 1326–1328, Aug. 2014.
[5] B. Yin, M. Wu, J. R. Cavallaro, and C. Studer, “Conjugate gradient-based soft-output detection and precoding in massive MIMO systems,” in *Proc. IEEE GLOBECOM’14*, Austin, TX, USA, Dec. 2014, pp. 3696–3701.
[6] L. Dai, X. Gao, X. Su, S. Han, C. I, and Z. Wang, “Low-complexity soft-output signal detection based on gauss-seidel method for uplink multiuser large-scale MIMO systems,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4839–4845, Oct. 2015.
[7] D. M. Young, *Iterative Solution of Large Linear Systems*. New York, NY, USA: Academic Press, 1971.
[8] M. Burger, B. Kaltenbacher, and A. Neubauer, *Iterative Solution Methods*. New York, NY, USA: Springer, 2011.