IMPLEMENTATION TRADE-OFFS FOR LINEAR DETECTION
IN LARGE-SCALE MIMO SYSTEMS

Bei Yin[1], Michael Wu[1], Christoph Studer[1], Joseph R. Cavallaro[1], and Chris Dick[2]

[1]Rice University, Houston, TX, USA; e-mail: {by2, mbw2, studer, cavallar}@rice.edu
[2]Xilinx, San Jose, CA, USA; e-mail: chrisd@xilinx.com

ABSTRACT

In this paper, we analyze the VLSI implementation trade-offs for linear data detection in the uplink of large-scale multiple-input multiple-output (MIMO) wireless systems. Specifically, we analyze the error incurred by using the sub-optimal, low-complexity matrix inverse proposed in Wu *et al.*, *2013, ISCAS*, and compare its performance and complexity to an exact matrix inversion algorithm. We propose a Cholesky-based reference architecture for exact matrix inversion and show corresponding implementation results on an Virtex-7 FPGA. Using this reference design, we perform a performance/complexity trade-off comparison with an FPGA implementation for the proposed approximate matrix inversion, which reveals that the inversion circuit of choice is determined by the antenna configuration (base-station antennas vs. number of users) of large-scale MIMO systems.

Index Terms— Large-scale MIMO, linear detection, approximate matrix inversion, FPGA implementation.

1. INTRODUCTION

Large-scale multiple-input multiple-output (MIMO) is an emerging wireless data transmission technique, which uses a large number of the antennas at the base station (BS) to serve a small number of users simultaneously and in the same frequency band [1, 2]. Besides improving the spectral efficiency and link reliability compared to conventional (small-scale) MIMO systems [2, 3], large-scale MIMO enables the deployment of low-complexity precoding methods in the downlink and inexpensive radio-frequency circuitry that have the potential to reduce the operational power consumption and hardware costs in the base station (BS) [1, 4].

1.1. Low-complexity data detection

For the large-scale MIMO uplink, where the users transmit data to the BS, low-complexity and sub-optimal data detection methods are of paramount importance, as optimal or even sub-optimal detection methods such as maximum-likelihood (ML) or linear minimum mean-squared error (MMSE) detection result in excessive computational complexity [2]. In order to overcome the complexity bottleneck of linear data detection methods in large-scale MIMO systems, we recently proposed a low-complexity, approximate inversion method in [5]. However, the impact of the antenna configuration on the performance and hardware complexity of this approximate inversion method has not been analyzed systematically.

1.2. Contributions

In this paper, we systematically analyze the implementation trade-offs associated with linear data detection in the large-scale MIMO uplink. We show analytically that the approximation error caused by the approximate inversion method of [5] is proportional to the number of users squared and inversely proportional to the number of BS antennas. We then compare the approximate inversion method to a Cholesky-based exact inverse and investigated the associated computational complexity. We furthermore present FPGA implementation results for both inversion methods for various antenna configurations, which we use to study the associated hardware complexity/SNR performance trade-offs. We finally demonstrate that the approximate inverse is preferred for systems having large BS-antenna-to-user ratios, whereas using an exact matrix inversion circuit is beneficial in systems having small BS-antenna-to-user ratios.

2. LARGE-SCALE MIMO UPLINK

We next introduce the system model and outline the linear detection algorithm investigated in the remainder of the paper.

2.1. System model

We consider the large-scale multi-user MIMO uplink with N antennas at the BS communicating with $M < N$ single antenna users. The transmitted bit stream for each user is first encoded using a channel encoder and then mapped to constellation points in the set $\mathcal{O}$. The transmit vector $\mathbf{s} = [s_1, \ldots, s_M]^T$ with $\mathbf{s} \in \mathcal{O}^M$ containing the transmit symbols for all M users, is then transmitted over the wireless

channel modeled as $\mathbf{y} = \mathbf{Hs} + \mathbf{n}$, where $\mathbf{y} = [y_1, \ldots, y_N]^T$ corresponds to the vector received at the BS, $\mathbf{H} \in \mathbb{C}^{N \times M}$ is the (tall and skinny) uplink channel matrix, and $\mathbf{n} \in \mathbb{C}^N$ models additive noise at the BS; the entries of $\mathbf{H}$ and of $\mathbf{n}$ are assumed to be i.i.d. zero-mean Gaussian with unit variance and N_0, respectively. We furthermore set $\mathbb{E}\{|s_i|^2\} = E_s, \forall i$.

2.2. Linear detection for large-scale MIMO

The task of the BS is to compute soft-estimates in the form of log-likelihood ratios (LLRs) for the coded bits given the channel matrix[1] $\mathbf{H}$ and the receive-vector $\mathbf{y}$ [6]. Since the number of BS antennas N and the number of users M is expected to be much larger than that of conventional (small-scale) MIMO systems, low-complexity detection algorithms are mandatory for practical realizations of large-scale MIMO [2].

To this end, we deploy the low-complexity, linear detection method proposed in [7]. The algorithm starts by computing the matched-filter (MF) output $\mathbf{y}^{\mathrm{MF}} = \mathbf{H}^H \mathbf{y}$ and the $M \times M$ Gram matrix $\mathbf{G} = \mathbf{H}^H \mathbf{H}$, followed by computing

$$\mathbf{A} = \mathbf{G} E_s + N_0 \mathbf{I}_M. \tag{1}$$

This regularized matrix is then used to generate an estimate of the transmit vector $\mathbf{s}$ as

$$\hat{\mathbf{s}} = \mathbf{A}^{-1} \mathbf{y}^{\mathrm{MF}} = \mathbf{A}^{-1} \mathbf{Gs} + \mathbf{A}^{-1} \mathbf{n}. \tag{2}$$

From this estimate, entry wise LLR-values can be computed by approximating the residual noise plus interference as i.i.d. zero-mean Gaussian distributed (see [7] for the details).

3. ERROR ANALYSIS OF APPROXIMATE MATRIX INVERSION

Computation of the inverse $\mathbf{A}^{-1}$ of (1) causes the main complexity of the detection algorithm outlined above. Inverting $\mathbf{A}$ can result in very high complexity for large-scale MIMO systems as it requires $O(M^3)$ operations. Hence, an efficient matrix inversion approximation method was proposed in [5] to arrive at cost-effective hardware implementations. We next summarize the idea behind this approximation method and then provide a corresponding error analysis.

3.1. k-term Neumann series approximation

To reduce the complexity of computing $\mathbf{A}^{-1}$ compared to an exact inversion, we start by the following Neumann series proposed in [5]:

$$\mathbf{A}^{-1} = \sum_{n=0}^{\infty} (\mathbf{X}^{-1}(\mathbf{X} - \mathbf{A}))^n \mathbf{X}^{-1}, \tag{3}$$

which holds if $\mathbf{A}$ satisfies $\lim_{n \to \infty} (\mathbf{I} - \mathbf{X}^{-1}\mathbf{A})^n = 0$. By decomposing $\mathbf{A}$ in (1) into $\mathbf{A} = \mathbf{D} + \mathbf{E}$ with $\mathbf{D}$ being the main

---

[1] In practice, channel-state information is acquired through training pilots.

and $\mathbf{E}$ the off diagonal of $\mathbf{A}$, and by keeping the first k terms of (3) only, we obtain the following k-term approximation:

$$\widetilde{\mathbf{A}}_k^{-1} = \sum_{n=0}^{k-1} (-\mathbf{D}^{-1}\mathbf{E})^n \mathbf{D}^{-1}. \tag{4}$$

For a $k = 2$ term approximation, for example, we get

$$\widetilde{\mathbf{A}}_2^{-1} = (\mathbf{I}_M - \mathbf{D}^{-1}\mathbf{E})\mathbf{D}^{-1} = \mathbf{D}^{-1} - \mathbf{D}^{-1}\mathbf{E}\mathbf{D}^{-1}, \tag{5}$$

which only requires $O(M^2)$ operations compared to the $O(M^3)$ complexity required by an exact inversion.

3.2. Error analysis

We next analyze the impact of the approximate inverse in (4) when used as a substitute for $\mathbf{A}^{-1}$. To this end, let

$$\mathbf{\Theta}_k = \sum_{n=k}^{\infty} (-\mathbf{D}^{-1}\mathbf{E})^n \mathbf{D}^{-1} = (-\mathbf{D}^{-1}\mathbf{E})^k \mathbf{A}^{-1}$$

be the residual error of $\widetilde{\mathbf{A}}_k^{-1}$. Using $\widetilde{\mathbf{A}}_k^{-1}$ rather than $\mathbf{A}^{-1}$ in the detection process (2) leads to

$$\widetilde{\mathbf{s}} = \widetilde{\mathbf{A}}_k^{-1} \mathbf{y}^{\mathrm{MF}} = (\mathbf{A}^{-1} - \mathbf{\Theta}_k) \mathbf{y}^{\mathrm{MF}} = (\mathbf{I} - \mathbf{\Theta}_k \mathbf{A}) \mathbf{A}^{-1} \mathbf{y}^{\mathrm{MF}},$$

which reveals that the approximation error depends on $\mathbf{\Theta}_k \mathbf{A}$. The following theorem characterizes the probability that this approximation error exceeds a given threshold $\alpha \geq 0$.

Theorem 3.1. *Let $N > 4$ and $\mathbf{H} \in \mathbb{C}^{N \times M}$ be i.i.d. complex Normal distributed. Then, for $k > 0$ and $N_0 \geq 0$, we have*

$$\Pr[\|\mathbf{\Theta}_k \mathbf{A}\|_F^2 > \alpha] \leq \frac{\sqrt{2} M^2 (N+1)}{(N-4)^2 \alpha^{\frac{1}{k}}}. \tag{6}$$

Proof. The proof follows from standard norm inequalities and Markov's inequality applied to the left-hand side of (6), and by bounding $\mathbb{E}[\|\mathbf{\Theta}_1 \mathbf{A}\|_F^2]$ using properties of Chi-squared random variables for the entries in $\mathbf{D}$ and $\mathbf{E}$ [8]. The details of the proof are omitted due to space constraints. $\square$

Theorem 3.1 reveals that for a given threshold α, the approximation error can be arbitrarily small by increasing the number of BS antennas N, while keeping the number of users M fixed. Furthermore, the approximation error decreases with M^2/N, which implies that a small number of Neumann series terms (e.g., $k = 2$) results in a good approximation to the exact inverse $\mathbf{A}^{-1}$ in large-scale MIMO systems having significantly more BS antennas than users.

4. REFERENCE IMPLEMENTATION AND COMPLEXITY COMPARISON

To compare the above approximate inversion with exact matrix inversion on a hardware-implementation level, we next present a Cholesky decomposition-based exact matrix inversion architecture. We then compare its implementation with the approximate inversion architecture proposed in [5].

Table 1. Complexity comparison of the k-term Neumann series approximation and the Cholesky decomposition.

	Multiplications	Additions
$k = 1$ approx.	0	0
$k = 2$ approx.	$4M(M-1)$	$2(M^2 - M)$
Cholesky dec.	$2M^3 + M^2 + M/3$	$2M^3/3 + 4M/3$

Table 2. Implementation results of approximate inversion and Cholesky decomposition on a Virtex-7 XC7VX1140T FPGA.

Unit	$N \times M$	Slices / DSP48	Freq. / Latency [MHz] / [cycles]
Approx. [5]	32×4	1095 (0.6%) / 9 (0.3%)	301 / 52
Cholesky	32×4	555 (0.3%) / 36 (1.1%)	281 / 94
Approx. [5]	128×8	3985 (2.2%) / 21 (0.6%)	285 / 55
Cholesky	128×8	5083 (2.9%) / 280 (8.3%)	219 / 202

4.1. Exact matrix inversion via Cholesky decomposition

We chose the Cholesky decomposition as a baseline for exact matrix inversion, since it exhibits a smaller number of operations compared with other inversion algorithms such as direct matrix inversion, QR decomposition, or LU factorization [9]. In the uplink, the detector requires the explicit inverse (mainly to compute the noise-plus-interference variance), which requires N instances of forward/backward substitution. In what follows, we exclusively focus on the Cholesky decomposition, as the complexity required by forward and backward substitution is negligible in large-scale MIMO systems, i.e., of order $O(N^2)$, compared to the $O(N^3)$ complexity required by the Cholesky decomposition itself.

In Table 1, we compare complexity of the Cholesky decomposition with the approximate inversion for the cases $k = 1, 2$. Since channel matrices are complex-valued, we convert the complex-valued operations to their real-valued equivalents, i.e., we assume 4 real-valued multiplications and 2 additions for 1 complex multiplication, and 2 real-valued additions for 1 complex addition. We see that the approximate matrix inversion requires less operations than the Cholesky decomposition for $M > 0$; in addition, for $k = 2$ the approximate inverse scales with $O(M^2)$, whereas the Cholesky decomposition scales with $O(M^3)$. We note that if considering the complexity required by forward/backward substitution, the complexity difference is even more pronounced.

4.2. Cholesky decomposition architecture

To compare the performance and complexity of both inversion circuits, we next detail a VLSI architecture of the Cholesky decomposition suitable for large-scale MIMO systems. The proposed Cholesky decomposition unit factorizes $\mathbf{A}$ into $\mathbf{LL}^H$ using a systolic array, where $\mathbf{L}$ is a lower-triangular matrix. The jth diagonal entry of $\mathbf{L}$ is computed as $L_{jj} = \sqrt{(A_{jj} - \sum_{k=1}^{j-1} L_{jk}L_{jk}^H)}$; the off-diagonal element in row i and column j below the diagonal $i > j$, is computed as $L_{ij} = (A_{ij} - \sum_{k=1}^{j-1} L_{ik}L_{jk}^H)/L_{jj}$, which requires one real-valued reciprocal. This algorithm is sequential in nature, as the division operation is executed after the square root operation. To arrive at high-throughput architecture, we parallelize these two operations. Specifically, we first compute $S_j = A_{jj} - \sum_{k=1}^{j-1} L_{jk}L_{jk}^H$ and $M_{ij} = A_{ij} - \sum_{k=1}^{j-1} L_{ik}L_{jk}^H$ in parallel. Then, $L_{jj} = \sqrt{S_j}$ and $R_j = 1/\sqrt{S_j}$ are calculated in parallel. Finally, $L_{ij} = M_{ij}R_j$ is computed.

4.3. Square root and inverse square root units

To further enhance the throughput of the systolic array, we designed dedicated square root and inverse square root units for $\sqrt{S_j}$ and $1/\sqrt{S_j}$, respectively. We follow the approach put forward in [7] to improve numerical stability. Concretely, the value $S_j = 2^m y$ is shifted such that $y \in [1, 4)$, with m being an even number. The results of $\sqrt{y}$ and $1/\sqrt{y}$ are obtained using separate lookup tables (LUTs). Finally, the result of $\sqrt{y}$ is multiplied by $2^{-m/2}$ to compensate for the initial shift; similarly, the result of $1/\sqrt{y}$ is multiplied by $2^{m/2}$.

4.4. Approximate matrix inversion architecture

As detailed in [5], the approximate matrix inversion implementation consists of a Gram-matrix computation unit and a unit computing the $k = 2$ approximate inverse in (5). The Gram matrix unit corresponds to an $M \times M$ lower-triangular systolic array, where each processing element (PE) consists of a multiply-and-accumulate (MAC) unit. This architecture requires two different PEs, for the diagonal and off-diagonal elements in $\mathbf{G}$, requiring a total of $(3M^2 + M)/2$ multipliers.

In the approximate inversion unit, the ith diagonal entry of $\widetilde{\mathbf{A}}_2^{-1}$, i.e., $D_{ii}^{-1} = (G_{ii} + N_0)^{-1}$ is computed with a dedicated reciprocal unit. For the off-diagonal element in row i and column j of $\widetilde{\mathbf{A}}_2^{-1}$, we compute $D_{ii}^{-1}G_{ij}D_{jj}^{-1}$, which requires one real-valued multiplication and a real-by-complex multiplication. In total, this module needs three multipliers.

4.5. Fixed-point design and FPGA implementation

We implemented both the Cholesky decomposition unit and the approximate inverse each with a Gram computation unit on a Virtex-7 FPGA using Xilinx Vivado High-Level Synthesis 2012. The input and output word length is 18 bit. All multiplications have been mapped to DSP48E1 slices. Each LUT is realized using a block-RAM (RAMB18E1) having 1024 entries and a word length of 18 bit.

We parameterized the approximate inverse and Cholesky decomposition unit for different antenna configurations to explore the resulting FPGA resources and decomposition throughput. The results for 32×4 and 128×8 are summarized in Table 2, and include the Gram computation unit detailed in [5]. The Cholesky decomposition unit requires considerably more clock cycles than the approximate inversion unit; in addition, the maximum clock frequency of
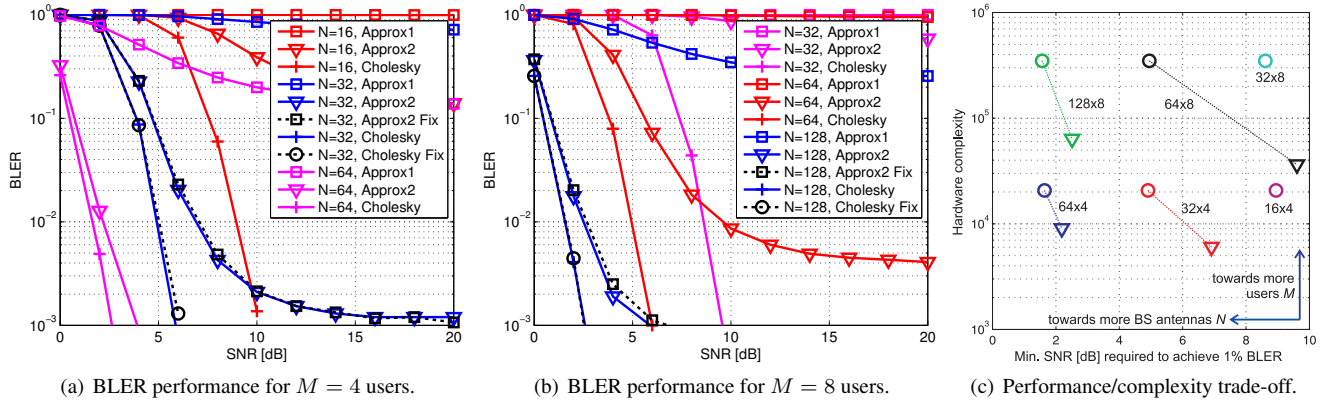
Fig. 1. (a), (b) Large-scale MIMO uplink block error-rate (BLER) performance; "Fix" indicates fixed-point performance. (c) Performance/complexity trade-off between the approximate matrix inversion ($k = 2$) and Cholesky decomposition units; circles indicate the performance and complexity of exact inversion using the Cholesky decomposition and triangles represent the approximate inversion unit.

Cholesky decomposition is lower. Moreover, the area utilization of the approximate inversion is much lower than that of the Cholesky decomposition in the used Virtex-7 FPGA.

5. PERFORMANCE/COMPLEXITY TRADE-OFFS

We first analyze the resulting block error rate (BLER) performance using numerical simulations for both algorithms. Then, we characterize the performance/complexity trade-offs using simulation and FPGA implementation results.

5.1. Uplink block error-rate performance

We simulate a coded MIMO-OFDM system with 128 subcarriers, 16-QAM, and assume a 10 m linear antenna array, where the antennas are equally spaced similarly to [10]. We use the WINNER-Phase-2 model [11] to generate the channel matrices. At the BS, we use the soft-output MMSE detector outlined in Sec. 2.2 (see [7] for the details) in combination with a rate-$5/6$ soft-input Viterbi decoder.

The resulting block error-rates (BLERs) are shown in Figs 1(a) and 1(b) for $M = 4$ and $M = 8$ users. The approximate inversion method with $k = 2$ ("Approx2" in Fig. 1) is able to approach the performance of an exact matrix inversion ("Cholesky" in Fig. 1) for systems having a large number of BS antennas N. The approximate method incurs an error floor or small-scale systems. In addition, for a given N, the error floor increases as the number of users M increases. These trends are consistent with Theorem 3.1, where the error bound scales with M^2/N. Finally, we see that the approximate inverse significantly outperforms the MF detector ("Approx1" in Fig. 1), which is typically considered for low-complexity detection in large-scale MIMO systems [1].

5.2. Performance/complexity trade-off

Based on the simulated BLER and our FPGA implementation results, we now analyze the performance/complexity trade-offs associated with the approximate matrix inversion and the Cholesky decomposition. To this end, we plot the associated hardware complexity against the minimum SNR required to achieve 1% BLER[2] in Fig. 1(c). Since both designs are dominated by multipliers, we define hardware complexity as the product of the number of multipliers and the time required to compute the Gram matrix plus the (approximate) inverse.

From Fig. 1(c), we see that the hardware complexity of the Cholesky-based inverse is very large and does not depend on the number of BS antennas. The hardware cost of the exact inverse is dominated by Cholesky decomposition, which consumes a significant portion of the available multipliers. In contrast, the hardware complexity for the approximate inverse is roughly one order of magnitude smaller.

In addition, for large ratios between the number of BS antennas to the number of users N/M, we see that the SNR performance of the approximate inverse and the exact inverse are very similar. For small ratios N/M, the performance difference between the approximate inverse and the exact inverse is significantly larger, caused by the error induced by the 2-term approximation (5). In fact, for 16×4 and 32×8, the approximate inverse is unable to achieve 1% BLER (cf. Figs. 1(a) and 1(a)). In these cases, an exact inverse (e.g., using our Cholesky decomposition circuits) is necessary.

5.3. Discussion

We conclude that for large-scale MIMO systems where the number of BS antennas is much larger than the number of users, the approximate matrix inversion proposed in [5] outperforms an exact Cholesky-based method in terms of the performance/complexity trade-off, i.e., both methods achieve similar BLER performance, while the approximate inverse requires one order of magnitude less hardware complexity. For MIMO systems where the number of BS antennas is not much larger than the number of users—resembling to conventional (small-scale) MIMO systems—an exact inverse is necessary to avoid poor error-rate performance.

---

[2]A minimum BLER of 1% is specified in IEEE 802.11n, for example.

6. REFERENCES

[1] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE TWC*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[2] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large arrays," *arXiv preprint: 1201.3210v1*, Jan. 2012.

[3] H. Huh, G. Caire, H. C. Papadopoulos, and S. A. Ramprashad, "Achieving "massive MIMO" spectral efficiency with a not-so-large number of antennas," *arXiv preprint: 1107.3862v2*, Sept. 2011.

[4] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *arXiv preprint: 1112.3810v2*, May 2012.

[5] M. Wu, B. Yin, A. Vosoughi, C. Studer, J. R. Cavallaro, and C. Dick, "Approximate matrix inversion for high-throughput data detection in the large-scale mimo uplink," in *Proc. IEEE ISCAS*, Beijing, China, May 2013.

[6] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Comm.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.

[7] C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation," *IEEE JSSC*, vol. 46, no. 7, pp. 1754–1765, July 2011.

[8] A. Papoulis and S. Pillai, *Probability, random variables, and stochastic processes*. McGraw-Hill, 2002.

[9] G. Stewart, *Matrix Algorithms: Basic decompositions*, 1998.

[10] J. Hoydis, C. Hoek, T. Wild, and S. ten Brink, "Channel measurements for large antenna arrays," in *Proc. IEEE ISWCS*, Aug. 2012.

[11] L. Hentilä, P. Kyösti, M. Käske, M. Narandzic, and M. Alatossava. (2007, December) Matlab implementation of the WINNER phase II channel model ver 1.1. [Online]. Available: https://www.ist-winner.org/phase_2_model.html