Approximate Matrix Inversion for High-Throughput Data Detection in the Large-Scale MIMO Uplink

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

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

***Abstract*—The high processing complexity of data detection in the large-scale multiple-input multiple-output (MIMO) uplink necessitates high-throughput VLSI implementations. In this paper, we propose—to the best of our knowledge—first matrix inversion implementation suitable for data detection in systems having hundreds of antennas at the base station (BS). The underlying idea is to carry out an approximate matrix inversion using a small number of Neumann-series terms, which allows one to achieve near-optimal performance at low complexity. We propose a novel VLSI architecture to efficiently compute the approximate inverse using a systolic array and show reference FPGA implementation results for various system configurations. For a system where 128 BS antennas receive data from 8 single-antenna users, a single instance of our design processes 1.9 M matrices/s on a Xilinx Virtex-7 FPGA, while using only 3.9% of the available slices and 3.6% of the available DSP48 units.**

I. Introduction

Multiple-input multiple-output (MIMO) combined with spatial multiplexing [1] is the key technology in most modern wireless communication standards, such as 3GPP LTE or IEEE 802.11n. MIMO technology offers improved link reliability and higher data rates compared to single-antenna systems by simultaneously transmitting multiple data streams in the same frequency band. However, conventional point-to-point and multi-user (MU) MIMO wireless systems already start to approach the theoretical throughput limits. Consequently, novel transmission technologies become necessary to meet the ever-growing demand for higher data rates without further increasing the communication bandwidth [2], [3].

Large-scale MIMO (or massive MIMO) is an emerging technology, which uses antenna arrays having orders of magnitude more elements at the base station (BS) compared to conventional (small-scale) MIMO systems, while simultaneously serving a small number of users in the same frequency band [2]. This technology promises further improvements in spectral efficiency and link reliability over conventional (small-scale) MIMO systems [3], [4]. In addition, large-scale MIMO has the potential to reduce the operational power consumption at the BS [2], [5].

Unfortunately, the benefits of large-scale MIMO come at the cost of significantly increased computational complexity in the BS compared to small-scale MIMO systems. Specifically, data detection in the large-scale MIMO uplink is among the most critical tasks, as the presence of hundreds of antennas at the BS requires novel detection algorithms that scale favorably to high-dimensional problems. Since optimal methods, such as maximum-likelihood (ML) detection or sphere decoding (SD) [6], would entail prohibitive complexity, one has to resort to low-complexity sub-optimal linear detection schemes [3] or stochastic techniques, such as Markov-chain Monte-Carlo (MCMC)-based detection methods [7].

Contributions: In this paper, we address the complexity issue associated with data detection in the large-scale MIMO uplink. We focus on linear soft-output data detection in combination with a novel approximate method for matrix inversion relying on Neumann series, which significantly reduces the computational complexity compared to exact inversion algorithms, while approaching the performance of SD methods. To demonstrate the efficacy of our inversion method, we present a novel systolic VLSI architecture for carrying out the inversion at high throughput for the high-dimensional problems arising in large-scale MIMO systems. Finally, we present reference implementation results on a Virtex-7 FPGA for various system configurations.

II. Large-Scale MIMO Uplink

We consider a large-scale multi-user (MU) MIMO system with N antennas at the BS communicating with $M < N$ single antenna users.[1] In what follows, we focus on the uplink, i.e., where M users are simultaneously transmitting to the BS.

A. Uplink System Model

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 $\mathbf{s} \in \mathcal{O}^M$ contains the transmit symbols for all M users. The vector $\mathbf{s}$ is then transmitted over the wireless channel modeled as $\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}$, 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; its entries are assumed to be i.i.d. zero-mean Gaussian with variance N_0. We furthermore assume that the transmit symbols satisfy $\mathbb{E}\{|s_i|^2\} = E_\mathrm{s}$, $\forall i$.

B. Soft-Output 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[2] $\mathbf{H}$ and the receive-vector $\mathbf{y}$, by means of a soft-output MIMO detection algorithm (see, e.g., [8]). Since

[1]Note that the model can easily be generalized to the case, where each user is equipped with multi-antenna terminals.

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

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, one must resort to low-complexity detection schemes because sophisticated detection methods, such as the k-Best algorithm or sphere decoding, would entail prohibitive complexity [3]. The most prominent low-complexity MIMO detection method is minimum mean-square error (MMSE) detection [6], which mitigates MU interference by inverting the effect of the channel matrix.

To arrive at low-complexity, we follow the linear detection approach of [9]. We start by computing the matched-filter output $\mathbf{y}^{\text{MF}} = \mathbf{H}^H\mathbf{y}$ and the $M \times M$ Gram matrix $\mathbf{G} = \mathbf{H}^H\mathbf{H}$. Then, we compute the following regularized matrix:

$$\mathbf{A} = \mathbf{G}E_{\text{s}} + N_0\mathbf{I}_M. \tag{1}$$

An estimate of the transmit vector can then be computed as

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

For soft-output detection, we further decompose the right-hand side of (2) for each user i as follows:

$$\hat{s}_i = \bar{\mathbf{a}}_i^H\mathbf{g}_i s_i + \textstyle\sum_{j\neq i}\bar{\mathbf{a}}_j^H\mathbf{g}_j s_j + \bar{\mathbf{a}}_i^H\mathbf{n} = \mu_i s_i + w_i, \tag{3}$$

where $\bar{\mathbf{a}}_i^H$ and $\mathbf{g}_i$ is the i^{th} row of $\mathbf{A}^{-1}$ and i^{th} column of $\mathbf{G}$, respectively, $\mu_i = \bar{\mathbf{a}}_i^H\mathbf{g}_i$ and w_i represents the noise-plus-interference (NPI) at user i. The decomposition (3) finally enables one to compute component-wise LLR-values by approximating the NPI as i.i.d. zero-mean Gaussian distributed with variance $\sigma_i^2 = \mathbb{E}\{|w_i|^2\}$ (see [9] for the details).

III. Low Complexity Matrix-Inversion for Large-Scale MIMO

The main computational complexity of the algorithm outlined above is caused by the inverse of $\mathbf{A}$. More specifically, computation of $\mathbf{A}^{-1}$ requires $O(M^3)$ number of operations, which quickly results in prohibitive complexity for the typical dimensions arising in large-scale MIMO. Hence, to arrive at cost-effective hardware implementations, an efficient inversion method is of paramount importance. We next present a novel method that approximately inverts the matrix $\mathbf{A}$ at low complexity, by exploiting the fact that in large-scale MIMO systems the channel matrices $\mathbf{H}$ are tall and skinny and therefore, well-conditioned, in general [2].

A. Neumann Series Approximation

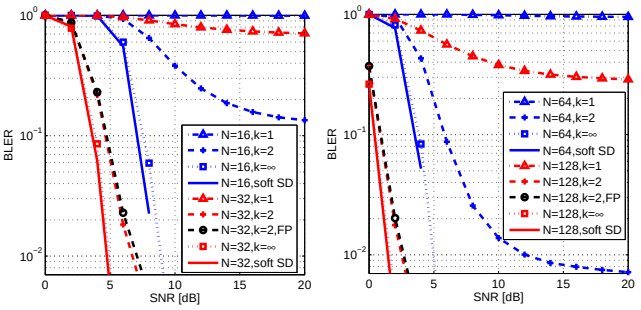
To reduce the complexity of computing $\mathbf{A}^{-1}$ compared to direct (and exact) matrix inversion methods, we propose to use the Neumann series approximation [10]. Concretely, if $\mathbf{A}$ is close to an invertible matrix $\mathbf{X}$ satisfying

$$\lim_{n\to\infty}(\mathbf{I} - \mathbf{X}^{-1}\mathbf{A})^n = 0 \;\text{ or }\; \lim_{n\to\infty}(\mathbf{I} - \mathbf{A}\mathbf{X}^{-1})^n = 0, \tag{4}$$

then the inverse of $\mathbf{A}$ can be rewritten using the following Neumann series [10]:

$$\mathbf{A}^{-1} = \textstyle\sum_{n=0}^{\infty}(\mathbf{X}^{-1}(\mathbf{X} - \mathbf{A}))^n\mathbf{X}^{-1}.$$

We can decompose $\mathbf{A}$ in (1) into its main diagonal $\mathbf{D}$ and off diagonal $\mathbf{E}$ such that $\mathbf{A} = \mathbf{D} + \mathbf{E}$. For $M \ll N$, the matrix $\mathbf{A}$ in (1) becomes a diagonally dominant matrix, i.e $\mathbf{A} \approx \mathbf{D}$. For i.i.d. Gaussian channel matrices $\mathbf{H}$ and in the large antenna limit, i.e., for $N \to \infty$ and a given M, it was shown in [2] that $\mathbf{A} \to \mathbf{I}_M$. This property of large-scale MIMO systems is the key to arrive at a low-complexity matrix-inversion method.



(a) $M = 4$ users. (b) $M = 8$ users.

Fig. 1. Uplink BLER performance in the large-scale MIMO uplink; the curves associated with $N = \infty$ correspond to exact matrix inversion and the curves associated with 'FP' correspond to fix-point implementations.

B. Low-Complexity Approximate Matrix Inversion

Since $\mathbf{A}$ is close to $\mathbf{D}$ for large-scale MIMO, we apply the Neumann series by letting $\mathbf{X} = \mathbf{D}$. By assuming that the matrix $\mathbf{D}$ is invertible and (4) holds, we can rewrite the inverse of $\mathbf{A} = \mathbf{D} + \mathbf{E}$ as

$$\mathbf{A}^{-1} = (\mathbf{D} + \mathbf{E})^{-1} = \textstyle\sum_{n=0}^{\infty}(-\mathbf{D}^{-1}\mathbf{E})^n\mathbf{D}^{-1}.$$

By keeping only the first k terms of the Neumann series, we arrive at the following k-term approximation of $\mathbf{A}^{-1}$:

$$\widetilde{\mathbf{A}}_k^{-1} = \textstyle\sum_{n=0}^{k-1}(-\mathbf{D}^{-1}\mathbf{E})^n\mathbf{D}^{-1}, \tag{5}$$

For $k = 1$, the inverse coincides to a MF detector, as (2) simply re-scales each entry of $\mathbf{y}^{\text{MF}}$. For $k = 2$, the inverse of $\mathbf{A}$ is approximated by $\widetilde{\mathbf{A}}_2^{-1} = \mathbf{D}^{-1} - \mathbf{D}^{-1}\mathbf{E}\mathbf{D}^{-1}$, which only requires $O(M^2)$ operations, in contrary to $O(M^3)$ operations required by an exact inversion algorithm. Subsequently, we use $\widetilde{\mathbf{A}}_2^{-1}$ in place of $\mathbf{A}^{-1}$, which significantly reduces the complexity of linear detection in large-scale MIMO systems.

C. Block Error-Rate (BLER) Performance in the Uplink

We now demonstrate the performance of the approximate inversion method for the large-scale MIMO uplink. 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 [11]. We use the WINNER-Phase-2 mode [12] to generate the channel matrices. At the BS, we use the soft-output MMSE detector described in Sec. II-B in combination with a rate-$5/6$ soft-input Viterbi decoder.

Figures 1(a) and 1(b) show the BLER performance of the proposed algorithm compared to an exact matrix inverse and soft-output single tree-search SD [13] for $M = 4$ and $M = 8$ users. Evidently, the proposed inversion method approaches the performance of an exact matrix inversion algorithm for a large number of BS antennas N. Moreover, the performance of linear detection approaches that of max-log optimal soft-output SD at significantly lower complexity. In addition,

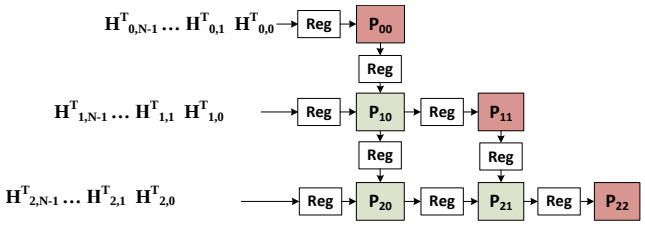


Fig. 2. High-level architecture of the systolic array for $M = 3$.

the proposed method significantly outperforms MF detection typically considered for low-complexity detection in large-scale MIMO [2]. For smaller systems, the approximate method incurs an error floor, which strongly depends on the number of BS antennas. We emphasize that practical communication standards commonly specify BLER $= 10^{-2}$, which is below the error floor in typical large-scale MIMO configurations.

IV. VLSI Architecture

We now present an architecture that is able to compute the 2-term approximation at very high throughput. In our architecture, we partitioned the inversion into two tasks, i.e., 1) computation of the Gram matrix $\mathbf{G} = \mathbf{H}^H\mathbf{H}$ and 2) computation of $\widetilde{\mathbf{A}}_2^{-1} = \mathbf{D}^{-1} - \mathbf{D}^{-1}\mathbf{E}\mathbf{D}^{-1}$, which are executed in pipelined fashion. Since both tasks produce symmetric matrices, we only process the lower-triangular part.

A. Gram-Matrix Computation Unit

The first unit computes the Gram matrix using an $M \times M$ lower-triangular systolic array. The architecture is shown in Fig. 2 for $M = 3$. Each processing element (PE, denoted by $\mathsf{P}_{k\ell}$ in Fig. 2) in the systolic array consists of a multiply-and-accumulate (MAC) unit. The transposed input matrix $\mathbf{H}^T$ is shifted one column at a time into the systolic array. Each processing element performs a MAC operation with both input operands. To ensure that each PE processes the correct set of operands, the values in i^{th} row of $\mathbf{H}^T$ are delayed by $i-1$ cycles. Once an input value reaches a diagonal PE, the value is conjugated and passed to the lower part of the systolic array.

There are two PE variants in the architecture. A PE on the main diagonal requires two multipliers to compute the squared absolute value of a complex number, $|a+bj|^2 = a^2 + b^2$. A PE on the off-diagonal computes the product of two complex numbers, i.e., $(a+bj)(c-bj)$. To reduce the number of multipliers, we deploy strength reduction, which requires three multipliers and five additions [14]. Consequently, the number of multipliers required in this unit is $(3M^2 + M)/2$. To minimize the critical path, each MAC unit is pipelined and has a throughput of one MAC operation per clock cycle.

B. Approximate Matrix Inversion Unit

To implement the approximate inverse, $\widetilde{\mathbf{A}}_2^{-1}$, we need to perform the following calculations. Let G_{ij} be the element in row i and column j of the Gram matrix $\mathbf{G}$. For the i^{th} diagonal entry of $\widetilde{\mathbf{A}}_2^{-1}$, we need to compute $D_{ii}^{-1} = (G_{ii} + N_0)^{-1}$. Since the diagonal elements of $\mathbf{G}$ are real positive numbers, D_{ii}^{-1} is computed with a reciprocal unit (see Sec. IV-C for the details). For the off-diagonal element in row i and column j of $\widetilde{\mathbf{A}}_2^{-1}$, we need to compute $D_{ii}^{-1}G_{ij}D_{jj}^{-1}$ which is one real multiplication followed by a real to complex multiplication. As a result, this module requires three multipliers in total.

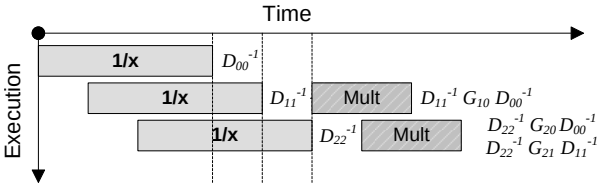


Fig. 3. Processing schedule for $\widetilde{\mathbf{A}}_2^{-1}$ computation ($M = 3$).

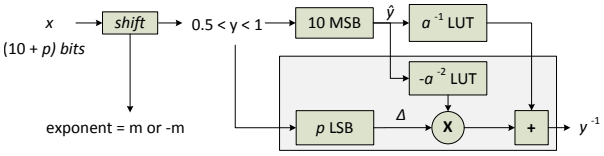


Fig. 4. Architecture of the FPGA-friendly reciprocal unit.

In the proposed architecture, we compute the entries of $\widetilde{\mathbf{A}}_2^{-1}$ row by row, from top to bottom, with the aid of a pipelined functional unit consisting of a single reciprocal unit and $M-1$ parallel multiplier units. Since the output of the Gram-matrix computational unit is lower-triangular, when D_{ii}^{-1} has been computed by the reciprocal unit, it is guaranteed that the D_{kk}^{-1} for $i < k < M$ are computed as well. As a result, the multipliers required to compute the off-diagonal elements in row i can proceed right after the computation of the diagonal element in the same row (D_{ii}^{-1}) is completed. This pipelined unit is able to compute one row of $\widetilde{\mathbf{A}}_2^{-1}$ per clock cycle. Figure 3 illustrates this processing schedule for $M = 3$.

C. FPGA-Friendly Reciprocal Unit

To compute D_{ii}^{-1} at high throughput, we designed a reciprocal unit that is particularly suitable for FPGA implementations. To improve numerical stability, we use a similar approach as described in [9] and shift the input value $D_{ii} = x$ such that it falls between 0.5 and 2, i.e., we get $x = 2^m y$, where $y \in [0.5, 1)$ and $m \in \mathbb{Z}$. We next rewrite $y = \hat{y} + \Delta$, where $\hat{y}$ corresponds to the 10 most significant bits of y and Δ the 10 remaining bits of y. We can now apply a first-order Taylor-series expansion to arrive at the approximation $a^{-1} \approx \hat{a}^{-1} - \Delta\hat{a}^{-2}$. Fig. 4 shows the block diagram of the reciprocal unit implementing this approximation. The terms $\hat{a}^{-1}$ and $\hat{a}^{-2}$ are obtained using two lookup tables (LUTs). The term $\hat{a}^{-2}$ is multiplied by Δ and added to $\hat{a}^{-1}$. Finally, the result is multiplied by 2^{-m} to compensate for the initial shift. Since fixed-point simulations suggest that only the first LUT is necessary, the shaded region in Fig. 4 was omitted in the final design. Note that the proposed reciprocal unit requires only one table look-up, which is in contrast to the architecture used in [9] that consists of two multipliers and one LUT.

V. FPGA Implementation Results

We implemented the approximate inversion architecture for large-scale MIMO systems on a Virtex-7 XC7VX1140T FPGA using Xilinx Vivado High-Level Synthesis 2012.

TABLE I
RESOURCE USAGE AND PERFORMANCE OF THE APPROXIMATE MATRIX-INVERSION UNIT ON A VIRTEX-7 XC7VX1140T FPGA

Unit	$N \times M$	Slices	LUTs	FFs	DSP48	BRAM	Max. freq.	Latency	Throughput
Gram matrix	32×4	824 (0.5%)	1784 (0.3%)	2106 (0.2%)	26 (0.8%)	0	302.29 MHz	46 cycles	6.57 M mat./s
Approx. inv.	4×4	1095 (0.6%)	2804 (0.4%)	2465 (0.2%)	9 (0.3%)	1	301.57 MHz	52 cycles	5.80 M mat./s
Combined	32×4	1919 (1.1%)	4588 (0.7%)	4571 (0.4%)	35 (1.1%)	1	301.57 MHz	52 cycles	5.80 M mat./s
Gram matrix	128×8	2946 (1.7%)	6873 (1.0%)	7786 (0.6%)	100 (3.0%)	0	299.76 MHz	150 cycles	2.00 M mat./s
Approx. inv	8×8	3985 (2.2%)	9476 (1.3%)	8094 (0.6%)	21 (0.6%)	1	285.46 MHz	55 cycles	5.18 M mat./s
Combined	128×8	6931 (3.9%)	16349 (2.3%)	15880 (1.2%)	121 (3.6%)	1	285.46 MHz	150 cycles	1.90 M mat./s
Gram matrix	128×16	11758 (6.6%)	26878 (3.8%)	30916 (2.2%)	392 (11.7%)	0	234.63 MHz	166 cycles	1.41 M mat./s
Approx. inv	16×16	5330 (3.0%)	11147 (1.6%)	15883 (1.1%)	45 (1.3%)	1	222.41 MHz	80 cycles	2.78 M mat./s
Combined	128×16	17088 (9.6%)	38025 (5.4%)	46796 (3.3%)	437 (13.0%)	1	222.41 MHz	166 cycles	1.34 M mat./s

A. Implementation Details and Fixed-Point Parameters

For the Gram matrix unit, the input values of the channel matrix $\mathbf{H}$ are 16 bit and we mapped all multiplications onto Xilinx DSP48E1 slices. As a result, the internal MAC registers are 48 bit. To reduce the output word-length, we truncate the final values of the Gram matrix to 20 bit for the next unit. For the inversion unit, we mapped all multiplications onto DSP48E1 slices and truncate the output to 16 bit. The LUT in the reciprocal unit has 1024 addresses and 18 bit outputs. Hence, we can implement it efficiently using one block-RAM (RAMB18E1) available on the FPGA. The resulting fixed-point performance is shown with dashed lines (and labeled by 'FP') in Fig. 1(a). The implementation loss for 4×32 is less than 0.2 dB SNR compared to a floating-point reference.

B. FPGA Implementation Results

We parameterized the architecture for N and M to explore the impact on the required FPGA resources and the maximum achievable throughput. The corresponding implementation results are detailed in Tbl. I. Note that—to the best of our knowledge—no implementation suitable for such high-dimensional problems has been described in the open literature and, hence, no comparison to existing designs can be given.

We can see that Gram-matrix unit requires more clock cycles than the approximate inversion unit as the dimensions of the MIMO system increases. The maximum clock frequency of the approximate inversion unit is slightly slower than that of the Gram-matrix unit. Assuming 64-QAM, the throughput of a single unit for a 32×4, 128×8, and 128×16 system is 139.2 Mb/s, 95.52 Mb/s, and 128.64 Mb/s, respectively. In the case of OFDM or SCFDMA, the throughput can be increased by simply instantiating more units that process multiple sub-carriers in parallel. As shown in Tbl. I, the inversion units are fairly small. Hence, the considered Virtex-7 FPGA has enough free resources to instantiate more than 10 units for a 32×4 and 128×8 system; the corresponding designs would easily reach throughputs exceeding 1 Gb/s. For 128×16 system, we could instantiate up to 7 units which corresponds to 0.9 Gb/s.

VI. CONCLUSIONS

We have developed—to the best of our knowledge—first implementation of a matrix inversion unit for linear data detection in the large-scale MIMO uplink. We have approximated the required matrix inversion using a small number of Neumann-series terms to reduces the complexity (compared to exact inversion) while only slightly degrading the error-rate performance in large-scale MIMO systems. To implement a fast approximate inversion unit, we have proposed a systolic VLSI architecture scaling favorably to large-dimensional problems. We have provided three reference designs, which enable data detection in large-scale MIMO systems at low complexity and high throughput with state-of-the-art FPGAs.

ACKNOWLEDGMENT

This work was supported in part by Renesas Mobile, Texas Instruments, Xilinx, Samsung, and by the US NSF under grants ECCS-1232274, EECS-0925942 and CNS-0923479.

REFERENCES

[1] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*. New York, USA: Cambridge University Press, 2008.
[2] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE TWC*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.
[3] 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.
[4] 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.
[5] 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.
[6] A. Burg, "VLSI circuits for MIMO communication systems," Ph.D. dissertation, ETH Zürich, Switzerland, 2006.
[7] T. Datta, N. Ashok Kumar, A. Chockalingam, and B. Sundar Rajan, "A novel MCMC algorithm for near-optimal detection in large-scale uplink mulituser MIMO systems," in *ITA 2012*, Feb. 2012, pp. 69 –77.
[8] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE TCOM*, vol. 51, no. 3, Mar. 2003.
[9] 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.
[10] G. Stewart, *Matrix Algorithms: Basic decompositions*, 1998.
[11] J. Hoydis, C. Hoek, T. Wild, and S. ten Brink, "Channel measurements for large antenna arrays," in *Proc. IEEE ISWCS*, Aug. 2012.
[12] L. Hentilä, P. Kyösti, M. Käske, M. Narandzic, and M. Alatossava. (2007, Dec.) Matlab implementation of the WINNER phase II channel model ver 1.1. [Online]. Available: https://www.ist-winner.org/phase_2_model.html
[13] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: Algorithms and VLSI implementation," *IEEE J. Sel. Areas in Comm.*, vol. 26, no. 2, pp. 290–300, Feb. 2008.
[14] V. Strassen, "Gaussian elimination is not optimal," *Numerische Mathematik*, vol. 13, pp. 354–356, 1969.