VLSI Design of Large-Scale Soft-Output MIMO Detection Using Conjugate Gradients

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

[1]Department of ECE, Rice University, Houston, TX; e-mail: {by2, mbw2, cavallar}@rice.edu
[2]School of ECE, Cornell University, Ithaca, NY; e-mail: studer@cornell.edu

***Abstract*—We propose an FPGA design for soft-output data detection in orthogonal frequency-division multiplexing (OFDM)-based large-scale (multi-user) MIMO systems. To reduce the high computational complexity of data detection, our design uses a modified version of the conjugate gradient least square (CGLS) algorithm. In contrast to existing linear detection algorithms for massive MIMO systems, our method avoids two of the most complex tasks, namely Gram-matrix computation and matrix inversion, while still being able to compute soft-outputs. Our architecture uses an array of reconfigurable processing elements to compute the CGLS algorithm in a hardware-efficient manner. Implementation results on Xilinx Virtex-7 FPGA for a 128 antenna, 8 user large-scale MIMO system show that our design only uses 70% of the area-delay product of the competitive method, while exhibiting superior error-rate performance.**

I. Introduction

Large-scale (or massive) MIMO is an emerging technology for next generation wireless communication systems [1]. In such systems, the base station (BS) is equipped with hundreds of antennas and serves a small number of users in the same frequency band [2]–[4]. Compared to conventional small-scale MIMO systems, massive MIMO promises improved link reliability, higher spectral efficiency, and superior energy efficiency. Large-scale MIMO also enables low-complexity detection algorithms (such as the matched filter) to achieve close-to-optimum error-rate performance—this, however, only holds for systems with an excessive number of BS antennas.

Data detection in realistic systems: For realistic antenna configurations (e.g., a hundred BS antennas with ten users), computationally expensive methods such as linear data detection, become necessary to achieve near-optimal performance [5]. To reduce the computational complexity of data detection, several approximate methods have been proposed recently [5]–[9]. All these methods either use the Cholesky decomposition or a Neumann series expansion to compute or approximate the matrix inverse that is required for linear data detection. As shown in [5], in systems with a large BS-to-user antenna ratio, the approximate Neumann-series method is able to achieve near-optimal performance. In order to improve the error-rate performance for systems with medium BS-to-user ratios, a conjugate-gradient (CG)-based soft-output data detection algorithm was proposed in [10]. More recently, another CG-based detector was proposed in [11]; this detection algorithm, however, is unable to compute soft-outputs.

Contributions: We develop a VLSI architecture for the CG-based soft-output data-detection algorithm proposed in [10]. In particular, we develop an architecture consisting of a reconfigurable array of processing elements (PEs) to compute the CGLS algorithm, as well as the necessary post-equalization signal-to-interference-and-noise-ratio (SINR) information that is crucial for soft-output detection. Our implementation results for a Xilinx Virtex-7 FPGA demonstrate that our detector achieves superior error-rate performance compared to existing Neumann-series-based detectors [5], while only using 70% of the hardware complexity (in terms of the area–delay product).

II. System Model

We consider a massive MIMO orthogonal frequency-division multiplexing (OFDM) system with B antennas at the BS and U single-antenna user-terminals. At the transmit-side, each user encodes its own information bit stream using a channel encoder. The encoded bits are then mapped onto constellation points in the set $\mathcal{S}$. The resulting data is then transformed from the frequency to the time domain and transmitted over the wireless channel. At the BS, the received signals are transformed back into the frequency domain. For each subcarrier (or tone), we model the input-output relation of the wireless link as

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

where, for the sake of simplicity, we omit the subcarrier index. In (1), $\mathbf{y} \in \mathbb{C}^B$ corresponds to the received vector at the BS, $\mathbf{H} \in \mathbb{C}^{B\times U}$ is the MIMO channel matrix, $\mathbf{x} \in \mathcal{S}^U$ is the transmit vector containing signals from all users, and $\mathbf{n} \in \mathbb{C}^B$ models additive noise. In what follows, we assume that the channel matrices $\mathbf{H}$ are generated from the WINNER-Phase-2 channel model [12] and the channel matrices are perfectly known at the BS. The entries of the noise vector $\mathbf{n}$ are assumed to be i.i.d. circularly-symmetric complex Gaussian with variance N_0. The transmit power of each user and the average SNR per receive antenna is defined as $\mathbb{E}\{|x_i|^2\} = E_\text{s}$ and $\mathit{SNR} = UE_s/N_0$, respectively.

Soft-output MMSE detection: As demonstrated in [5], linear soft-output minimum mean square error (MMSE) data detection is able to achieve near-optimal error-rate performance in

This work was supported in part by Xilinx Inc., and by the US National Science Foundation under grants ECCS-1408370, CNS-1265332, ECCS-1232274, and ECCS-1408006.

massive MIMO systems. One of the simplest soft-output MMSE detection methods was put forward in [13] and requires computation of the MMSE equalization matrix

$$\mathbf{W} = \mathbf{A}^{-1}\mathbf{H}^H, \tag{2}$$

where $\mathbf{A} = \mathbf{G} + N_0/E_\mathrm{s}\mathbf{I}_U$ and $\mathbf{G} = \mathbf{H}^H\mathbf{H}$ is the Gram matrix. One can then equalize the receive-vector $\mathbf{y}$ as $\hat{\mathbf{x}} = \mathbf{W}\mathbf{y}$. By modeling the equalized symbol of the ith user as $\hat{x}_i = \mu_i x_i + z_i$, where the so-called equalized interference and noise is modeled as $z_i = \sum_{j,j\neq i} \mathbf{w}_i^H\mathbf{h}_j x_j + \mathbf{w}_i^H\mathbf{n}$, we can compute soft-output information in the form of log-likelihood ratio (LLR) values. In particular, we use the max-log approximation for each bit b and user i and compute the LLR values as follows [13]:

$$L_{i,b} = \rho_i \left(\min_{a\in\mathcal{S}_b^0} \left| \frac{\hat{x}_i}{\mu_i} - a \right|^2 - \min_{a'\in\mathcal{S}_b^1} \left| \frac{\hat{x}_i}{\mu_i} - a' \right|^2 \right). \tag{3}$$

Here, $\rho_i = \mu_i^2/\nu_i^2$ represents post-equalization signal-to-interference-and-noise-ratio (SINR). The sets $\mathcal{S}_b^0$ and $\mathcal{S}_b^1$ represent subsets of the constellation set $\mathcal{S}$ for which the b-th bit is 0 and 1, respectively.

In massive MIMO systems, soft-output MMSE detection as outlined above is computational expensive. In particular, computation of the Gram matrix $\mathbf{G}$ and the matrix inversion $\mathbf{A}^{-1}$ in (2) result in excessive computational complexity, especially in systems with a large number of antennas at both ends of the wireless link [5]. More specifically, computing the matrices $\mathbf{G}$ and $\mathbf{A}^{-1}$ requires $\mathcal{O}(BU^2)$ and $\mathcal{O}(U^3)$ operations, respectively. To significantly reduce the complexity of soft-output MMSE detection in large-scale MIMO systems, we therefore use our soft-output CGLS detection method in [10], which avoids an explicit computation of $\mathbf{G}$ and $\mathbf{A}^{-1}$ altogether.

III. CGLS-Based Soft-Output Data Detection

CGLS-based data detection: CGLS is an efficient iterative method to solve large systems of linear equations [14]. For our application, we use CGLS to solve the following regularized least-squares optimization problem:

$$\hat{\mathbf{x}} = \operatorname*{arg\,min}_{\tilde{\mathbf{x}}\in\mathbb{C}^U} \|\mathbf{y} - \mathbf{H}\tilde{\mathbf{x}}\|_2^2 + \tfrac{N_0}{E_\mathrm{s}}\|\tilde{\mathbf{x}}\|_2^2, \tag{4}$$

where the result $\hat{\mathbf{x}}$ is equivalent to that of the linear MMSE estimate $\hat{\mathbf{x}} = \mathbf{W}\mathbf{y}$ in (2). By directly solving (4) in an iterative manner, CGLS is able to compute $\hat{\mathbf{x}}$ at low complexity, without ever forming the Gram matrix $\mathbf{G}$ and the inverse $\mathbf{A}^{-1}$. Furthermore, each iteration of CGLS requires rather inexpensive and regular matrix-vector multiplications, which involve only $\mathcal{O}(BU)$ operations. In addition, for a $B\times U$-dimensional system, CGLS converges to the exact solution (4) in U-iterations [14]. In practice, however, far fewer iterations are necessary to obtain accurate approximations to (4).

Our detection method is summarized in Algorithm 1 and is based on the following (equivalent) formulation of (4):

$$\hat{\mathbf{x}} = \operatorname*{arg\,min}_{\tilde{\mathbf{x}}\in\mathbb{C}^U} \|\bar{\mathbf{y}} - \overline{\mathbf{H}}\tilde{\mathbf{x}}\|_2, \tag{5}$$

where $\bar{\mathbf{y}} = \left[\mathbf{y}^T, \mathbf{0}_{1\times U}\right]^T$ is the so-called augmented received vector, and $\overline{\mathbf{H}} = \left[\mathbf{H}^T, \sqrt{N_0/E_s}\mathbf{I}_U\right]^T$ is a suitably augmented channel matrix $\mathbf{H}$.

Algorithm 1 CGLS-based soft-output MMSE detection

```
1: inputs: $\mathbf{H}$ and $\mathbf{y}$
2: initialization:
3:     $\mathbf{b} = \mathbf{H}^H\mathbf{y}$ and $\overline{\mathbf{H}} = \left[\mathbf{H}^T, \sqrt{N_0/E_s}\mathbf{I}_U\right]^T$
4:     $\mathbf{v}_0 = \mathbf{0}$, $\mathbf{r}_0 = \mathbf{b}$, $\mathbf{p}_0 = \mathbf{r}_0$, and $\mathbf{t}_0 = \overline{\mathbf{H}}\mathbf{p}_0$
5: for $k = 1, \ldots, K$ do
6:     $\alpha_k = \|\mathbf{r}_{k-1}\|_2^2/\|\mathbf{t}_{k-1}\|_2^2$
7:     $\mathbf{v}_k = \mathbf{v}_{k-1} + \alpha_k\mathbf{p}_{k-1}$
8:     $\mathbf{e}_k = \overline{\mathbf{H}}^H\mathbf{t}_{k-1}$
9:     $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k\mathbf{e}_k$
10:    $\beta_k = \|\mathbf{r}_k\|_2^2/\|\mathbf{r}_{k-1}\|_2^2$
11:    $\mathbf{p}_k = \mathbf{r}_k + \beta_k\mathbf{p}_{k-1}$
12:    $\mathbf{t}_k = \overline{\mathbf{H}}\mathbf{p}_k$
13:    compute $\mu_{i|k}$, $\forall i$, as in Sec. III
14:    compute $\rho_{i|k}$, $\forall i$, as in Sec. III
15: end for
16: outputs: $\hat{\mathbf{x}}_K = \mathbf{v}_K$, $\mu_{i|K}$, $\forall i$, and $\rho_{i|K}$, $\forall i$
```

SINR Computation: The key to attain low complexity soft-output MMSE detection (and to avoid an explicit computation of $\mathbf{A}^{-1}$) consists of the steps carried out on lines 13 and 14 of Algorithm 1, where the quantities $\mu_{i|k}$ as well as $\rho_{i|k}$ are computed. As detailed in [10], one can iteratively compute $\mu_{i|k}$ and $\rho_{i|k}$ directly within CGLS, which significantly reduces the computational complexity compared to methods that require the computation of $\mathbf{A}^{-1}$ (see, e.g., [13]). In essence, in every CGLS iteration k, we compute

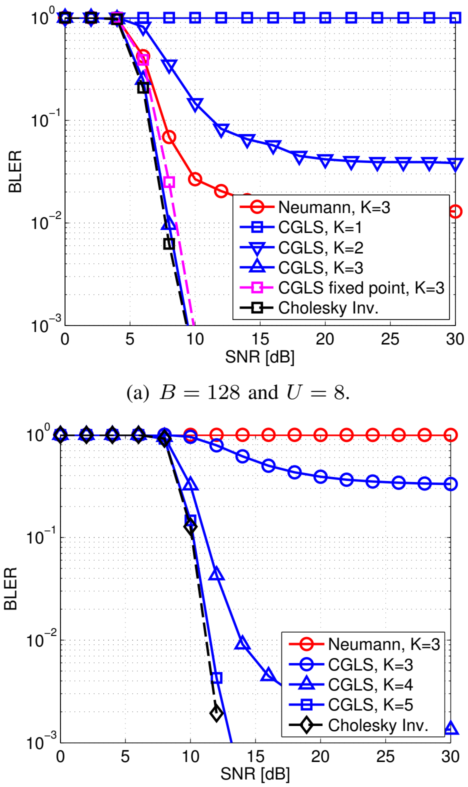
$$\widetilde{L}_{i|k} = \widetilde{L}_{i|k-1} + \left(\frac{\alpha_k(1+\beta_{k-1})}{\alpha_{k-1}} - \alpha_k D_{i,i}\right)(\widetilde{L}_{i|k-1} - \widetilde{L}_{i|k-2}) - \frac{\alpha_k\beta_{k-2}}{\alpha_{k-2}}(\widetilde{L}_{i|k-2} - \widetilde{L}_{i|k-3}), \tag{6}$$

with $\widetilde{L}_{i|1} = \alpha_1$, and $\widetilde{L}_{i|k} = 0$ for $k < 1$, $\forall i$. The quantity $D_{i,i}$ corresponds to the ith diagonal element of the regularized Gram matrix $\mathbf{A}$. Using $\widetilde{L}_{i|k}$, we can then approximate $\mu_{i|k}$ with $\mu_{i|k} \approx \widetilde{L}_{i|k}G_{i,i}$, where $G_{i,i}$ is the ith diagonal entry of the Gram matrix $\mathbf{G}$. The post-equalization SINR required in (3) can then be approximated as

$$\rho_{i|k} = \mu_{i|k}^2/\nu_{i|k}^2 \approx G_{i,i}/N_0, \tag{7}$$

which does not depend on the iteration index k. Using the approximation $\rho_{i|k}$ in (7) results in low computational complexity, i.e., only requires $\mathcal{O}(U)$ operations. Fortunately, the loss in terms of error-rate performance due to the approximation is negligible in large-scale MIMO systems (see [10] for the details). Note that in contrast to the method detailed in [10], Algorithm 1 never computes the matrices $\mathbf{A}$ or $\mathbf{G}$. In particular, the proposed approximation (7) only relies on the diagonal elements of $\mathbf{A}$ and $\mathbf{G}$, which requires only $\mathcal{O}(BU)$ operations.

Error-rate performance: To assess the performance of our CGLS-based soft-output detection algorithm, we simulate the block-error rate (BLER) performance and compare it to that



(a) $B = 128$ and $U = 8$.

(b) $B = 256$ and $U = 32$.

Fig. 1. Block error rate (BLER) performance comparison.

of existing methods, namely an exact soft-output MMSE detector [13] and the Neumann-series approximation proposed in [5]–[8]. All simulations are carried out in a large-scale MIMO-OFDM system with 128 subcarriers and for 64-QAM. We furthermore use a rate-5/6 convolutional code and a soft-input log-MAP Viterbi decoder.

As shown in Fig. 1, the proposed CGLS-based soft-output detector achieves a BLER performance that is close to that of the exact MMSE detector (which uses a Cholesky-based matrix inversion) with only $K = 3$ iterations in a 128 BS antenna, 8 user large-scale MIMO system, and with $K = 5$ iterations for a system consisting of 256 BS antennas and 32 users. In contrast, the Neumann-series-based approach exhibits an error floor for the considered antenna configurations.

IV. Reconfigurable VLSI Architecture

We propose a low complexity VLSI architecture that utilizes the regularity of the proposed CGLS detection algorithm summarized in Algorithm 1. As shown in Fig. 2, our design consists of a reconfigurable processing element (PE) array. The array can be reconfigured on the fly to perform matrix-vector multiplications (line 3, line 4, line 8, and line 12), vector inner products (line 6, line 10, line 13, line 14), scaled vector additions/subtractions (line 7, line 9, and line 11), and scalar divisions (line 6 and line 10). A global finite state machine (FSM) controls the sequence of operations to perform the proposed soft-output detection algorithm.

Reconfigurable processing elements: To support the above operations using a PE array, each PE supports the following elementary operations: vector inner product, scaled vector addition/subtraction, and scalar division. As shown in Fig. 2, each PE consists of five components: A multiplier array, an add/subtract array, an accumulator array, an adder tree, and a scalar division module. These five components work together to perform the three elementary operations:

Vector inner product: Given two vectors, $\mathbf{a} = [a_1, \ldots, a_n]$ and $\mathbf{b} = [b_1, \ldots, b_n]$, the PE first performs element-wise complex-valued multiplication, $[a_1 b_1, \ldots, a_n b_n]$ using the multiplier array and add/subtract array. The PE then accumulates the element-wise products using the accumulator array and an adder tree. The vector inner product $c = \sum_{i=1}^{n} a_i b_i$ then corresponds to the output of the adder tree.

Scaled vector addition/subtraction: Given a real-valued scalar α and two complex vectors $\mathbf{a}$ and $\mathbf{b}$, the PE computes $\alpha\mathbf{a}$ using the multiplier array. The PE then computes $\mathbf{c} = \alpha\mathbf{a} + \mathbf{b}$ using the add/subtract array. The vector $\mathbf{c}$ is the output from the add/subtract array.

Scalar Division: The scalar division module consists of a lookup table and a real-valued multiplier. Given two real-valued scalars, a and b, the PE first computes $1/a$ using a lookup table (see, e.g., [5]). The PE then multiplies $1/a$ by b with a the real-valued multiplier to obtain b/a.

Array of reconfigurable PEs: In our design, we use one PE to perform the vector dot products, scaled vector additions/subtractions, and scalar divisions required by Algorithm 1. While a single PE can perform the required matrix-vector multiplication (line 3, line 4, line 8, and line 12) as a series of vector dot products, we can speed up matrix-vector multiplications by instantiating an array of PEs. For example, to compute $\mathbf{t}_k = \overline{\mathbf{H}}\mathbf{p}_k$ (a $U \times 1$ vector multiplied by a $B \times U$ matrix; line 12) with N PEs in a parallel manner, we assign the j-th PE to compute the dot-product between the j-th row of $\overline{\mathbf{H}}$ and $\mathbf{p}_k$ to generate the j-th element of $\mathbf{t}_k$.

Architecture Optimizations: A straightforward implementation of Algorithm 1 would result in an inefficient VLSI design. We therefore deploy several optimizations that reduce the silicon area and the processing latency.

We first reduce the memory requirements. An naïve implementation of Algorithm 1 requires storage of a B-dimensional vector $\mathbf{t}_k$ (line 9). To reduce the memory consumption, we interleave the computation on line 9 of Algorithm 1 with the computation of lines 6 and 8 of the next iteration. Each time we compute U elements of $\mathbf{t}_k$ (line 9), we use them immediately to compute $\|\mathbf{t}_{k-1}\|^2$ (line 6) and $\mathbf{e}_k$ (line 8). We then compute the next U elements of $\mathbf{t}_k$. In this way, we only need to store U elements of $\mathbf{t}_k$ instead of B elements of $\mathbf{t}_k$.

We also reduce the circuit complexity by reducing the dynamic range of the stored data. As shown in [5], the Gram matrix $\mathbf{H}^H\mathbf{H}$ is a diagonally dominant matrix with diagonal entries close to B. In CGLS, this property leads to a large dynamic range. Some vectors, such as $\mathbf{e}_k$, consist of relatively large values. Some other vectors, such as $\mathbf{v}_k$, consist of relatively small values. Similar to [5], in order to process

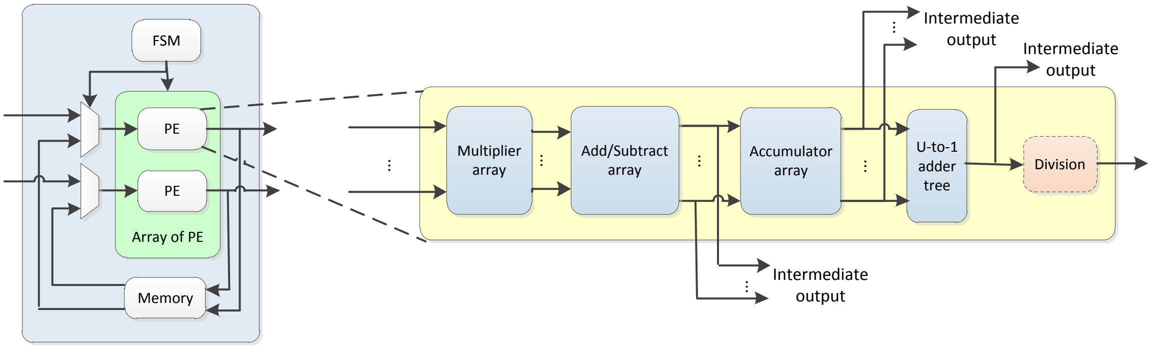


Fig. 2. Reconfigurable array of processing elements (PEs) for CGLS-based soft-output data detection.

TABLE I
Implementation results on a Xilinx Virtex-7 XC7VX690T FPGA

Antenna configuration	128 BS antennas, 8 users	
Detection algorithm	CGLS ($K = 3$)	Neumann ($K = 3$)
Slices	1 094 (1 %)	48 244 (44.6 %)
LUTs	3 324 (0.76 %)	148 797 (34.3 %)
FFs	3 878 (0.44 %)	161 934 (18.7 %)
DSP48s	33 (0.9 %)	1 016 (28.3 %)
Block RAMs	1 (0.03 %)	16 (1.1 %)
Latency (clock cycles)	951	196
Maximum clock frequency	412 MHz	317 MHz
Throughput	20 Mb/s	621 Mb/s
Area-delay product	**7 672 631**	**11 500 083**

both kinds of vectors with low complexity, a scaling factor κ is applied to $\mathbf{b}_k$ and $\mathbf{e}_k$ to reduce the dynamic range. This scaling cancels out at the end of every CGLS iteration. The scaling factor is obtained by selecting p such that $1/2^p$ is close to $1/B$, which implies that all required scaling operations are carried out by simple arithmetic shifts.

V. Implementation Results and Conclusion

To enable a fair comparison with the Neumann-series-based detector in [5], we implemented our CGLS large-scale MIMO detector for a 128×8 system on a Xilinx Virtex-7 XC7VX690T FPGA. The input values and the output values contain 15 bits per complex dimension. The output of multipliers in the multiplier array is truncated to 15 bits. The accumulators in the accumulator array use 20 bits. The LUT within the scalar division module is 12 bits wide and 1024 entries deep. The resulting fixed-point performance is shown in Fig. 1; we see that the implementation loss is negligible.

Table I compares the resource usage of our CGLS-based detector with the resource usage of the Neumann-series-based detector [5]. While our implementation achieves a lower throughput than that of [5], the hardware efficiency (area-delay product) is significantly better. Note that multiple instances of the CGLS-based detector can be used to match the throughput of the design in [5] at significantly smaller area. We use area-delay product to arrive at a fair comparison. The area corresponds to the number of FPGA LUTs, whereas the delay simply corresponds to the processing latency of both engines. This comparison shows that our CGLS-based detection requires 70% of the area-delay product of the design in [5]. From Fig. 1, we furthermore see that the CGLS-based method outperforms the Neumann-series-based approach in term of error-rate performance. Hence, the proposed detector design outperforms existing solutions for large-scale MIMO systems in terms of implementation complexity *and* error-rate performance.

References

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

[2] 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.

[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," *IEEE Trans. Wireless Commun.*, vol. 11, no. 9, pp. 3266–3239, Sept. 2012.

[4] 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," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[5] M. Wu, B. Yin, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, "Large-scale MIMO detection for 3GPP LTE: algorithms and FPGA implementations," *IEEE J. Sel. Topics in Sig. Proc.*, vol. 8, no. 5, pp. 916–929, Oct. 2014.

[6] B. Yin, M. Wu, C. Studer, J. R. Cavallaro, and C. Dick, "Implementation trade-offs for linear detection in large-scale MIMO systems," in *Proc. IEEE ICASSP*, Vancouver, Canada, May 2013, pp. 2679–2683.

[7] 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, pp. 2155–2158.

[8] B. Yin, M. Wu, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, "A 3.8 Gb/s large-scale MIMO detector for 3GPP LTE-Advanced," in *Proc. IEEE ICASSP*, May 2014, pp. 3907–3911.

[9] H. Prabhu, J. Rodrigues, O. Edfors, and F. Rusek, "Approximative matrix inverse computations for very-large MIMO and applications to linear pre-coding systems," in *Proc. IEEE WCNC*, 2013, pp. 2710–2715.

[10] 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*, Dec 2014, pp. 4287–4292.

[11] Y. Hu, Z. Wang, X. Gaol, and J. Ning, "Low-complexity signal detection using CG method for uplink large-scale MIMO systems," in *Proc. IEEE International Conference on Communication Systems (ICCS)*, Nov 2014, pp. 477–481.

[12] L. Hentilä, P. Kyösti, M. Käske, M. Narandzic, and M. Alatossava, "Matlab implementation of the WINNER phase II channel model ver 1.1," Dec. 2007.

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

[14] C. Paige and M. A. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Trans. Math. Softw.*, vol. 8, pp. 43–71, 1982.