Data Detection in Large Multi-Antenna Wireless Systems via Approximate Semidefinite Relaxation

Oscar Castañeda, Tom Goldstein, and Christoph Studer

***Abstract*—Practical data detectors for future wireless systems with hundreds of antennas at the base station must achieve high throughput and low error rate at low complexity. Since the complexity of maximum-likelihood (ML) data detection is prohibitive for such large wireless systems, approximate methods are necessary. In this paper, we propose a novel data detection algorithm referred to as Triangular Approximate SEmidefinite Relaxation (TASER), which is suitable for two application scenarios: (i) coherent data detection in large multi-user multiple-input multiple-output (MU-MIMO) wireless systems and (ii) joint channel estimation and data detection in large single-input multiple-output (SIMO) wireless systems. For both scenarios, we show that TASER achieves near-ML error-rate performance at low complexity by relaxing the associated ML-detection problem into a semidefinite program, which we solve approximately using a preconditioned forward-backward splitting procedure. Since the resulting problem is non-convex, we provide convergence guarantees for our algorithm. To demonstrate the efficacy of TASER in practice, we design a systolic architecture that enables our algorithm to achieve high throughput at low hardware complexity, and we develop reference field-programmable gate array (FPGA) and application-specific integrated circuit (ASIC) designs for various antenna configurations.**

***Index Terms*—FPGA and ASIC design, data detection, joint channel estimation and data detection, large single-input and multiple-input multiple-output (SIMO and MIMO) wireless systems, semidefinite relaxation.**

I. Introduction

LARGE multiple-input multiple-output (MIMO) and single-input multiple-output (SIMO) wireless technology, where the base station (BS) is equipped with hundreds or thousands of antennas, are widely believed to play a major role in fifth-generation (5G) cellular communication systems [2]–[7]. Such large wireless systems promise improved spectral efficiency, coverage, and range compared to traditional small-scale systems. However, the extremely large number of BS antennas requires the design of high-performance data-detection algorithms that can be implemented efficiently in very-large scale integration (VLSI) circuits [8]. In fact, data detection is among the most critical baseband-processing tasks in terms of implementation complexity, power consumption, throughput, and error-rate performance for such systems [9], [10].

To enable high-throughput uplink communication for massive multi-user (MU) MIMO wireless systems (where tens of user terminals transmit data to a BS with hundreds of antennas), a variety of low-complexity data-detection algorithms [11]–[19], as well as a few field-programmable gate array (FPGA) implementations [8], [20]–[22] and application-specific integrated circuit (ASIC) designs [23] have been proposed recently. To date, all data detectors that have been implemented in VLSI for such high-dimensional problems rely on (approximate) linear data detection [8], [20]–[23]. Such linear methods are known to suffer from a significant error-rate performance loss for more realistic systems with a not-so-large number of antennas at the BS or where the number of user terminals is comparable to that of the number of BS antennas [8]. Furthermore, the literature on large MU-MIMO data detection almost exclusively relies on the assumption of perfect channel state information (CSI) at the BS—an assumption that cannot be satisfied in practice.

A. Contributions

In this paper, we propose a novel data detection algorithm and corresponding VLSI designs for large wireless systems. Our algorithm, referred to as Triangular Approximate SEmidefinite Relaxation (TASER), can be deployed in two different application scenarios: (i) coherent data detection in massive MU-MIMO wireless systems and (ii) joint channel estimation and data detection (JED) in large SIMO wireless systems. Our detector builds upon semidefinite relaxation [24], which enables near maximum-likelihood (ML) data detection performance at polynomial (in the number of transmit antennas) complexity for systems that communicate with low-rate, constant-modulus modulation schemes [25]. TASER approximates the semidefinite relaxation (SDR) formulation of both the coherent ML and the JED ML problems using a Cholesky factorization, and solves the resulting non-convex problem using a preconditioned forward-backward splitting (FBS) procedure [26], [27]. We provide theoretical convergence guarantees for our algorithm, and we develop a corresponding systolic array that enables high-throughput data detection at low silicon area in an energy-efficient manner. We provide reference VLSI implementation results for a Xilinx Virtex-7 FPGA and for a 40 nm CMOS technology, and we perform an extensive comparison in terms of performance and complexity with recently-proposed data detector implementations for large MU-MIMO wireless systems [8], [20]–[23].

O. Castañeda and C. Studer are with the School of ECE, Cornell University, Ithaca, NY; e-mail: oc66@cornell.edu, studer@cornell.edu

T. Goldstein is with the Department of CS, University of Maryland, College Park, MD; e-mail: tomg@cs.umd.edu

A short version of this paper summarizing the TASER FPGA design for large MU-MIMO data detection has been presented at the IEEE International Symposium on Circuits and Systems (ISCAS) 2016 [1].

The system simulator for TASER used in this paper will be made available on GitHub after (possible) acceptance of the paper.

B. Relevant Prior Art

1) Data detection in large MU-MIMO: The literature on data detection in large (or massive) MU-MIMO wireless systems describes only a few algorithms that are able to achieve near-optimal error-rate performance [11], [13], [18]. For these algorithms, however, no hardware designs have been described in the open literature. So far, only sub-optimal, linear data detection algorithms have been integrated successfully in FPGAs [8], [20]–[22] or ASICs [23]. Unfortunately, such linear data detection algorithms suffer from a significant error-rate performance loss in "square" systems, where the number of users is comparable to the number of BS antennas [8]. In contrast, the proposed TASER algorithm achieves near-optimal error-rate performance, even in symmetric large MU-MIMO systems where the BS-to-user-antenna ratio is one.

2) SDR-based data detection: SDR is a well-known technique for achieving near-ML performance in multi-user code division multiple access (MU-CDMA) [28], [29] and traditional, small-scale MIMO [24], [30]–[36] wireless systems. Most results on SDR-based data detection rely on computationally inefficient, general-purpose convex solvers that require either the solution to a linear system or an eigenvalue decomposition per iteration—both of these operations entail prohibitive complexity when implemented in hardware. As an exception, the algorithm in [36] relies on block-coordinate descent, which avoids the solution to a full linear system per iteration. While computationally efficient, this method exhibits stringent data dependencies, requires a high number of multiplications per iteration, and consumes a large amount of memory, which renders corresponding VLSI designs inefficient. TASER, in contrast, is highly parallelizable and hardware friendly, and is—to the best of our knowledge—the first SDR-based data detector that has been successfully implemented in VLSI.

3) Joint channel estimation and data detection: JED is known to significantly outperform traditional data detection schemes that separate channel estimation from data detection. We believe that JED is a promising solution for large cellular systems, where pilot-contamination (i.e., pilot-based training for users in adjacent cells interferes with the training pilots in the current cell) poses a fundamental performance bottleneck [2]. The computational complexity of exact JED via an exhaustive search grows exponentially in the number of transmission time slots [37]. Hence, sphere-decoding (SD)-based methods have been proposed for JED in the SIMO [37]–[40] and MIMO [41] literature to reduce the computational complexity. Nevertheless, the design of hardware implementations of high-throughput sphere-decoders is challenging, and most existing designs only achieve a few hundred Mb/s for small MIMO systems (see [10], [42] for more details on SD-based data detectors). In addition—to the best of our knowledge—no hardware design for JED has been proposed in the open literature. In this paper, we show that (i) JED can be performed using SDR and (ii) TASER enables near-optimal, high-throughput JED for realistic large SIMO wireless systems.

C. Notation

Lowercase boldface letters stand for column vectors; uppercase boldface letters denote matrices. For a matrix $\mathbf{A}$, we denote its transpose, adjoint, and trace by $\mathbf{A}^T$, $\mathbf{A}^H$, and $\mathrm{Tr}(\mathbf{A})$, respectively. We use $A_{k,\ell}$ for the entry in the kth row and ℓth column of the matrix $\mathbf{A}$; the kth entry of a vector $\mathbf{a}$ is denoted by $a_k = [\mathbf{a}]_k$. The Frobenius norm of the matrix $\mathbf{A}$ is $\|\mathbf{A}\|_F = \sqrt{\sum_{k,\ell} |A_{k,\ell}|^2}$ and the ℓ_2-norm of the vector $\mathbf{a}$ is $\|\mathbf{a}\|_2 = \sqrt{\sum_k |a_k|^2}$. The identity matrix and all-ones vector are denoted by $\mathbf{I}$ and $\mathbf{1}$, respectively. The real and imaginary part of a complex-valued matrix $\mathbf{A}$ are denoted by $\Re(\mathbf{A})$ and $\Im(\mathbf{A})$, respectively.

D. Paper Outline

The rest of the paper is organized as follows: Section II introduces the large MU-MIMO and SIMO system models and discusses coherent ML data detection as well as JED. Section III introduces the TASER algorithm and provides a theoretical convergence analysis. Section IV details our systolic architecture. Section V shows reference implementation results and provides a comparison with existing data detectors for large MU-MIMO. Concluding remarks are presented in Section VI.

II. Data Detection in Large Multi-Antenna Wireless Systems

The algorithm and VLSI designs proposed in this paper are suitable for two application scenarios: (i) coherent data detection in large MU-MIMO systems and (ii) JED in large SIMO systems. We next describe the corresponding system models and show how both problems can be relaxed to a semidefinite program (SDP) of the same form.

A. Coherent Data Detection for Large MU-MIMO Systems

The first application scenario is data detection in the large (or massive) MU-MIMO wireless uplink with B BS antennas and U user antennas. We consider the standard input-output relation to model a flat-fading[1] MIMO wireless channel [43]: $\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}$. Here, $\mathbf{y} \in \mathbb{C}^B$ is the BS receive-vector, $\mathbf{H} \in \mathbb{C}^{B\times U}$ is the MIMO channel matrix, $\mathbf{s} \in \mathcal{O}^U$ is the transmit vector containing the data symbols from all users ($\mathcal{O}$ refers to the constellation set), and $\mathbf{n} \in \mathbb{C}^B$ is i.i.d. circularly-symmetric Gaussian with variance N_0 per entry. Assuming that an estimate of the channel matrix $\mathbf{H}$ was acquired during a dedicated training phase, ML data detection corresponds to the following problem [44]:

$$\hat{\mathbf{s}}^{\mathrm{ML}} = \arg\min_{\mathbf{s}\in\mathcal{O}^U} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|_2. \tag{1}$$

A number of computationally efficient sphere-decoding algorithms have been proposed to solve the combinatorial problem in (1) for conventional, small-scale MIMO systems [10], [45]–[47]. Unfortunately, the worst-case and average computational complexity of these exact methods still scales exponentially

[1]Our algorithm and circuit design are also suitable for frequency-selective channels in combination with orthogonal frequency-division multiplexing (OFDM), where we consider the same input-output relation per subcarrier.

with the number of users U [48], [49]. For large MU-MIMO systems, where the BS-to-user-antenna ratio exceeds a factor of two, recently-developed linear algorithms have been shown to achieve near-ML performance [4], [5], [8]. For systems with a large number of users where the BS-to-user-antenna ratio is close to one, however, linear methods are known to deliver poor error-rate performance [8].

To enable near-optimal error-rate performance at low complexity for such scenarios, we can relax the ML problem in (1) into an SDP [24]. This relaxation step requires us to reformulate the ML detection problem as follows. By assuming constant-modulus QAM constellations, such as BPSK and QPSK, we first perform the real-valued decomposition of the system model $\overline{\mathbf{y}} = \overline{\mathbf{H}}\overline{\mathbf{s}} + \overline{\mathbf{n}}$ using the following definitions:

$$\overline{\mathbf{y}} = \begin{bmatrix} \Re(\mathbf{y}) \\ \Im(\mathbf{y}) \end{bmatrix}, \qquad \overline{\mathbf{H}} = \begin{bmatrix} \Re(\mathbf{H}) & -\Im(\mathbf{H}) \\ \Im(\mathbf{H}) & \Re(\mathbf{H}) \end{bmatrix},$$
$$\overline{\mathbf{s}} = \begin{bmatrix} \Re(\mathbf{s}) \\ \Im(\mathbf{s}) \end{bmatrix}, \qquad \overline{\mathbf{n}} = \begin{bmatrix} \Re(\mathbf{n}) \\ \Im(\mathbf{n}) \end{bmatrix}.$$

This decomposition enables us to reformulate the ML problem in (1) into the following equivalent form:

$$\bar{\mathbf{s}}^{\text{ML}} = \underset{\tilde{\mathbf{s}} \in \mathcal{X}^N}{\arg\min}\ \text{Tr}(\tilde{\mathbf{s}}^T \mathbf{T} \tilde{\mathbf{s}}). \tag{2}$$

For QPSK, the matrix $\mathbf{T} = [\overline{\mathbf{H}}^T\overline{\mathbf{H}}, -\overline{\mathbf{H}}^T\overline{\mathbf{y}}; -\overline{\mathbf{y}}^T\overline{\mathbf{H}}, \overline{\mathbf{y}}^T\overline{\mathbf{y}}]$ is of dimension $N \times N$ with $N = 2U + 1$ and $\mathcal{X} \in \{-1, +1\}$ with $\tilde{\mathbf{s}} = [\Re(\mathbf{s}); \Im(\mathbf{s}); 1]$. The solution $\bar{\mathbf{s}}^{\text{ML}}$ can then be converted back into the complex-valued ML solution as $[\hat{\mathbf{s}}^{\text{ML}}]_i = [\bar{\mathbf{s}}^{\text{ML}}]_i + j[\bar{\mathbf{s}}^{\text{ML}}]_{i+U}$ for $i = 1, \ldots, U$. For BPSK, the matrix $\mathbf{T} = [\underline{\mathbf{H}}^T\underline{\mathbf{H}}, -\underline{\mathbf{H}}^T\overline{\mathbf{y}}; -\overline{\mathbf{y}}^T\underline{\mathbf{H}}, \overline{\mathbf{y}}^T\overline{\mathbf{y}}]$ is of dimension $N \times N$ with $N = U + 1$ and $\tilde{\mathbf{s}} = [\Re(\mathbf{s}); 1]$. Here, we define the $2B \times U$ matrix $\underline{\mathbf{H}} = [\Re(\mathbf{H}); \Im(\mathbf{H})]$. Since $\Im(\mathbf{s}) = \mathbf{0}$ in this case, $[\hat{\mathbf{s}}^{\text{ML}}]_i = [\bar{\mathbf{s}}^{\text{ML}}]_i$ for $i = 1, \ldots, U$. In Section II-C, we detail how the problem in (2) can be relaxed into an SDP.

B. Joint Channel Estimation and Data Detection

The second application scenario is JED in large SIMO wireless uplink systems where one single-antenna user communicates over $K + 1$ time slots with B BS antennas. We use the following input-output relation to model the (flat-fading) SIMO wireless channel [37]–[40]: $\mathbf{Y} = \mathbf{h}\mathbf{s}^H + \mathbf{N}$. Here, $\mathbf{Y} \in \mathbb{C}^{B \times (K+1)}$ contains the received vectors acquired over all $K + 1$ time slots, $\mathbf{h} \in \mathbb{C}^B$ is the unknown SIMO channel vector that is assumed to be block fading, i.e., constant over $K + 1$ time slots, $\mathbf{s}^H \in \mathcal{O}^{1 \times (K+1)}$ is the transmit vector containing the data symbols from all $K + 1$ time slots, and $\mathbf{N} \in \mathbb{C}^{B \times (K+1)}$ is i.i.d. circularly-symmetric Gaussian with variance N_0 per entry. By assuming that $\mathbf{h}$ is a deterministic but unknown channel vector with unknown prior statistics, we can formulate the following ML JED problem [40]:

$$\{\hat{\mathbf{s}}^{\text{JED}}, \hat{\mathbf{h}}\} = \underset{\mathbf{s} \in \mathcal{O}^{K+1},\, \mathbf{h} \in \mathbb{C}^B}{\arg\min} \|\mathbf{Y} - \mathbf{h}\mathbf{s}^H\|_F. \tag{3}$$

It is important to note that there exists a phase ambiguity between both outputs of JED because $\hat{\mathbf{h}}e^{j\phi}$ is also a solution whenever $\hat{\mathbf{s}}^{\text{JED}}e^{j\phi} \in \mathcal{O}^{K+1}$ for some phase ϕ. As a consequence, one may convey information either as phase changes in the vector $\mathbf{s}^H$ over time slots (known as differential encoding) or "pin down" the phase of one entry of the transmit vector; in what follows, we assume that the first transmitted entry is known to the receiver.[2]

By assuming that the entries in $\mathbf{s}$ are constant modulus (e.g., BPSK or QPSK), the ML JED estimate of the transmit vector reduces to [40]:

$$\hat{\mathbf{s}}^{\text{JED}} = \underset{\mathbf{s} \in \mathcal{O}^{K+1}}{\arg\max}\ \|\mathbf{Y}\mathbf{s}\|_2, \tag{4}$$

and $\hat{\mathbf{h}} = \mathbf{Y}\hat{\mathbf{s}}^{\text{JED}}$ is the estimate of the channel vector. For a small number of time slots $K + 1$, the problem in (4) can be solved exactly at low average complexity using SD methods [40]. For systems with many time slots, however, the computational complexity of such algorithms becomes prohibitive. In contrast to the coherent ML detection problem described in (2), linear methods that approximate (4) are unavailable as relaxing the constraint $\mathbf{s} \in \mathcal{O}^{K+1}$ to $\mathbf{s} \in \mathbb{C}^{K+1}$ causes the entries of $\mathbf{s}$ to grow without bound.

We now show how the ML JED problem in (4) can be transformed into the same structure of the coherent ML problem in (2), which enables SDR. Since the receiver is assumed to know the first transmitted symbol s_0, we rewrite the objective in (4) as $\|\mathbf{Y}\mathbf{s}\|_2 = \|\mathbf{y}_0 s_0 + \mathbf{Y}_r\mathbf{s}_r\|_2$, where $\mathbf{Y}_r = [\mathbf{y}_1, \ldots, \mathbf{y}_K]$ and $\mathbf{s}_r = [s_1, \ldots, s_K]^T$. Similarly to the coherent ML problem, we perform the real-valued decomposition by defining:

$$\overline{\mathbf{y}} = \begin{bmatrix} \Re(\mathbf{y}_0 s_0) \\ \Im(\mathbf{y}_0 s_0) \end{bmatrix}, \ \overline{\mathbf{H}} = \begin{bmatrix} \Re(\mathbf{Y}_r) & -\Im(\mathbf{Y}_r) \\ \Im(\mathbf{Y}_r) & \Re(\mathbf{Y}_r) \end{bmatrix}, \ \overline{\mathbf{s}} = \begin{bmatrix} \Re(\mathbf{s}_r) \\ \Im(\mathbf{s}_r) \end{bmatrix},$$

which allows us to rewrite $\|\mathbf{y}_0 s_0 + \mathbf{Y}_r\mathbf{s}_r\|_2 = \|\overline{\mathbf{y}} + \overline{\mathbf{H}}\overline{\mathbf{s}}\|_2$. We can now reformulate (4) in a form that is equivalent to (2) as

$$\bar{\mathbf{s}}^{\text{JED}} = \underset{\tilde{\mathbf{s}} \in \mathcal{X}^N}{\arg\min}\ \text{Tr}(\tilde{\mathbf{s}}^T \mathbf{T} \tilde{\mathbf{s}}). \tag{5}$$

For QPSK, the matrix $\mathbf{T} = -[\overline{\mathbf{H}}^T\overline{\mathbf{H}}, \overline{\mathbf{H}}^T\overline{\mathbf{y}}; \overline{\mathbf{y}}^T\overline{\mathbf{H}}, \overline{\mathbf{y}}^T\overline{\mathbf{y}}]$ is of dimension $N \times N$ with $N = 2K + 1$ and $\mathcal{X} \in \{-1, +1\}$ with $\tilde{\mathbf{s}} = [\Re(\mathbf{s}_r); \Im(\mathbf{s}_r); 1]$; for BPSK, the matrix $\mathbf{T} = -[\underline{\mathbf{H}}^T\underline{\mathbf{H}}, \underline{\mathbf{H}}^T\overline{\mathbf{y}}; \overline{\mathbf{y}}^T\underline{\mathbf{H}}, \overline{\mathbf{y}}^T\overline{\mathbf{y}}]$ is of dimension $N \times N$ with $N = K + 1$ and $\tilde{\mathbf{s}} = [\Re(\mathbf{s}_r); 1]$. Here, we define the $2B \times K$ matrix as $\underline{\mathbf{H}} = [\Re(\mathbf{Y}_r); \Im(\mathbf{Y}_r)]$. Analogously to the coherent ML case, the solution $\bar{\mathbf{s}}^{\text{JED}}$ can then be used to construct the complex-valued ML JED solution of (4).

Evidently, the problems described in (2) and (5) exhibit the same structure—we next show how both of these problems can be solved approximately using the same SDR-based method.

C. Semidefinite Relaxation of the Problems in (2) and (5)

SDR is a well-known approximation to the coherent ML problem [24], [28]–[30] and enables significantly lower (i.e., polynomial) computational complexity for systems employing BPSK and QPSK constellations.[3] SDR not only provides near-ML performance, but also achieves the same diversity order as the ML detector [25]. In contrast, the use of SDR for solving

[2]For SIMO systems, this approach resembles that of pilot-based transmission—the difference to JED is, however, that we also use all transmitted information symbols to improve the channel estimate and hence, to improve the error-rate performance.

[3]SDR methods for higher-order constellations (such as 16-QAM) exist; see, e.g., [32], [33] for more details.

the ML JED problem as proposed in Section II-B appears to be novel.

SDR-based data detection starts by reformulating the problems in (2) and (5) in the following equivalent form [24]:

$$\widehat{\mathbf{S}} = \underset{\mathbf{S}\in\mathbb{R}^{N\times N}}{\arg\min}\ \mathrm{Tr}(\mathbf{TS}) \ \text{ subject to } \mathrm{diag}(\mathbf{S}) = \mathbf{1},\ \mathrm{rank}(\mathbf{S}) = 1. \tag{6}$$

Here, we used $\mathrm{Tr}(\mathbf{s}^T\mathbf{T}\mathbf{s}) = \mathrm{Tr}(\mathbf{T}\mathbf{s}\mathbf{s}^T) = \mathrm{Tr}(\mathbf{TS})$, where $\mathbf{S} = \mathbf{s}\mathbf{s}^T$ is a rank-1 matrix and $\mathbf{s} \in \mathcal{X}^N$ is of appropriate dimension N. Unfortunately, the rank-one constraint in (6) makes this problem at least as hard as the original two problems in (2) and (5). The key idea of SDR is to relax this rank constraint, which results in an SDP that can be solved in polynomial time. Specifically, SDR applied to (6) results in the following well-known optimization problem [24]:

$$\widehat{\mathbf{S}} = \underset{\mathbf{S}\in\mathbb{R}^{N\times N}}{\arg\min}\ \mathrm{Tr}(\mathbf{TS}) \ \text{ subject to } \mathrm{diag}(\mathbf{S}) = \mathbf{1},\ \mathbf{S} \succeq 0, \tag{7}$$

where the constraint $\mathbf{S} \succeq 0$ ensures that the matrix $\mathbf{S}$ is positive semidefinite (PSD). If the result of the problem in (7) is rank one, then $\widehat{\mathbf{S}} = \hat{\mathbf{s}}\hat{\mathbf{s}}^H$ where $\hat{\mathbf{s}}$ contains the exact estimate to (2) and (5), i.e., SDR solves the original problem optimally. If the resulting matrix $\widehat{\mathbf{S}}$ has a higher rank, then an estimate of the ML solution can be obtained by taking the signs of the leading eigenvector of $\widehat{\mathbf{S}}$ or by using randomization schemes [24].

While (7) can be solved exactly using interior-point methods [24], such algorithms typically require (i) a large number of iterations, where each iteration requires either the solution to a linear system or an eigenvalue decomposition, and (ii) high numerical precision, which renders fixed-point hardware challenging. We believe that these are the main reasons why—until now—*no* VLSI design of an SDR-based data detector has been proposed in the open literature.

III. TASER: Triangular Approximate Semidefinite Relaxation

We now detail TASER, a novel algorithm for approximately solving the SDP presented in (7) using hardware accelerators.

A. Triangular SDP Formulation

The key idea of TASER builds on the fact that real-valued PSD matrices $\mathbf{S} \succeq 0$ can be factorized using the Cholesky decomposition $\mathbf{S} = \mathbf{L}^T\mathbf{L}$, where $\mathbf{L}$ is an $N\times N$ lower-triangular matrix with non-negative entries on the main diagonal. With this result, we can reformulate the SDP shown in (7) using the following equivalent form:

$$\widehat{\mathbf{L}} = \underset{\mathbf{L}}{\arg\min}\ \mathrm{Tr}(\mathbf{LTL}^T) \text{ subject to } \|\boldsymbol{\ell}_k\|_2 = 1, \forall k. \tag{8}$$

Here, we replaced the constraint $\mathrm{diag}(\mathbf{L}^T\mathbf{L}) = \mathbf{1}$ of (7) by the equivalent ℓ_2-norm constraint on the kth column $\boldsymbol{\ell}_k = [\mathbf{L}]_k$. To obtain (approximate) solutions to the ML or JED ML problems in either (2) or (5), respectively, we can take the signs of the last row of the solution matrix $\widehat{\mathbf{L}}$ from (8). In fact, if the solution matrix $\widehat{\mathbf{S}} = \widehat{\mathbf{L}}^T\widehat{\mathbf{L}}$ has rank one (this implies that TASER identified the ML solution), then the last row of $\widehat{\mathbf{L}}$ must contain the associated eigenvector as this is the only vector of dimension N. If, however, the solution matrix $\widehat{\mathbf{S}} = \widehat{\mathbf{L}}^T\widehat{\mathbf{L}}$ has a higher rank, an approximate ML solution must be extracted somehow. As suggested in [50], [51], taking the last row of the Cholesky decomposition results in accurate rank-one approximations of PSD matrices. Our own simulations in Section V-A confirm that this approximation yields excellent error-rate performance, i.e., close to that of the exact SDR detector followed by an eigenvalue decomposition. We emphasize that this approach avoids costly eigenvalue decompositions and randomization strategies that are required by conventional solvers that compute $\widehat{\mathbf{S}}$ exactly using SDR.

B. Forward-Backward Splitting

We now develop a computationally efficient algorithm that directly solves the triangular SDP formulation in (8). Unfortunately, the problem described in (8) is non-convex in the matrix $\mathbf{L}$ and hence, computing an optimal solution is difficult. For TASER, we apply FBS [27], a computationally efficient method to solve convex optimization problems, to the non-convex problem in (8). While this approach is not guaranteed to converge to the optimal solution of the non-convex problem posed by (8), we show in Section III-E that TASER converges to a critical point of (8). Furthermore, our simulation results in Section V demonstrate near-ML error-rate performance.

FBS is an efficient, iterative method to solve convex optimization problems of the form $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x})$, where the function f is smooth and convex, and g is convex but not necessarily smooth or bounded. FBS performs the following steps for $t = 1, 2, \ldots$ [26], [27]:

$$\mathbf{x}^{(t)} = \mathrm{prox}_g(\mathbf{x}^{(t-1)} - \tau^{(t)}\nabla f(\mathbf{x}^{(t-1)}); \tau^{(t-1)})$$

until convergence or a maximum number of iterations $t_{\max}$ is reached. Here, $\{\tau^{(t)} > 0\}$ is a suitably-chosen sequence of step size parameters, $\nabla f(\mathbf{x})$ is the gradient of the function f, and the so-called proximal operator for the function g is defined as [26], [27]:

$$\mathrm{prox}_g(\mathbf{z}; \tau) = \underset{\mathbf{x}}{\arg\min}\left\{\tau g(\mathbf{x}) + \tfrac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2\right\}. \tag{9}$$

In order to approximately solve (8) using FBS, we define $f(\mathbf{L}) = \mathrm{Tr}(\mathbf{LTL}^T)$ and incorporate the constraint using $g(\mathbf{L}) = \chi(\|\boldsymbol{\ell}_k\|_2 = 1, \forall k)$, where χ is the characteristic function (which is zero if the constraint is met and infinity otherwise). The gradient is given by $\nabla f(\mathbf{L}) = \mathrm{tril}(2\mathbf{LT})$, where $\mathrm{tril}(\cdot)$ extracts the lower-triangular part of the argument. Even though the function g is non-convex, the proximal operator defined in (9) has a closed form and is given by $\mathrm{prox}_g(\boldsymbol{\ell}_k; \tau) = \boldsymbol{\ell}_k/\|\boldsymbol{\ell}_k\|_2$, $\forall k$; in words, the proximal operator simply rescales the columns of $\mathbf{L}$ to have unit ℓ_2-norm.

In order to arrive at a hardware-friendly algorithm, we avoid sophisticated step size rules such as the ones proposed in [27]. We use a fixed step size proportional to the reciprocal of the Lipschitz constant of the gradient $\nabla f(\mathbf{L})$ as proposed in [26]. Our step size corresponds to $\tau = \alpha/\|\mathbf{T}\|_2$, where $\|\mathbf{T}\|_2$ is the spectral norm of the matrix $\mathbf{T}$ and $0 < \alpha < 1$ is a system-dependent tuning parameter that we use to improve the empirical convergence rate when running TASER for a small number of iterations (see Section III-E for a discussion).

Algorithm 1 TASER

inputs: $\widetilde{\mathbf{T}}$, $\mathbf{D}$, and $\tau = \alpha/\|\widetilde{\mathbf{T}}\|_2$
initialization: $\widetilde{\mathbf{L}}^{(0)} = \mathbf{D}$
for $t = 1, \ldots, t_{\max}$ **do**
  $\mathbf{V}^{(t)} = \widetilde{\mathbf{L}}^{(t-1)} - \operatorname{tril}(2\tau\widetilde{\mathbf{L}}^{(t-1)}\widetilde{\mathbf{T}})$
  $\widetilde{\mathbf{L}}^{(t)} = \operatorname{prox}_{\tilde{g}}(\mathbf{V}^{(t)})$
end for
outputs: $\overline{s}_k = \operatorname{sign}(\widetilde{L}_{N,k}^{(t_{\max})}), k = 1, \ldots, N-1$

C. Jacobi Preconditioning

To improve the convergence rate of FBS, we precondition the problem presented in (8). To this end, we compute a diagonal scaling matrix $\mathbf{D} = \operatorname{diag}(\sqrt{T_{1,1}}, \ldots, \sqrt{T_{M,M}})$, which we use to scale the matrix $\mathbf{T}$ as $\widetilde{\mathbf{T}} = \mathbf{D}^{-1}\mathbf{T}\mathbf{D}^{-1}$ so that $\widetilde{\mathbf{T}}$ has an all-ones main diagonal. The purpose of this so-called Jacobi preconditioner is to improve the condition number of the original PSD matrix $\mathbf{T}$ [52]. We then run FBS to recover a normalized version[4] of the lower-triangular matrix $\widetilde{\mathbf{L}} = \mathbf{L}\mathbf{D}$. We emphasize that preconditioning also requires us to modify the proximal operator, which turns out to be $\operatorname{prox}_{\tilde{g}}(\tilde{\boldsymbol{\ell}}_k) = D_{k,k}\tilde{\boldsymbol{\ell}}_k/\|\tilde{\boldsymbol{\ell}}_k\|_2$, where $\tilde{\boldsymbol{\ell}}_k$ is the kth column of $\widetilde{\mathbf{L}}$. Since we only rely on the signs of the last row of $\widehat{\mathbf{L}}$ to obtain an estimate of the ML problems, we can simply take the signs of the normalized triangular matrix $\widetilde{\mathbf{L}}$.

D. The TASER Algorithm

We now have all the necessary ingredients for TASER, which is summarized in Algorithm 1. The inputs of the algorithm are the preconditioned matrix $\widetilde{\mathbf{T}}$, the scaling matrix $\mathbf{D}$, and the step size τ. We initialize the FBS procedure by $\widetilde{\mathbf{L}}^{(0)} = \mathbf{D}$, which resulted in excellent performance for all considered scenarios. The main loop of TASER then performs the gradient and proximal steps as discussed in Sections III-B and III-C until a maximum number of iterations $t_{\max}$ is reached. For most situations, only a few iterations are sufficient to achieve near-ML error rate performance (see Section V for numerical results). The TASER algorithm computes an estimate for the coherent ML and ML JED problems in (2) and (5), respectively.

E. Convergence Theory

The TASER algorithm tries to solve a non-convex problem using FBS. Hence, our approach raises two questions, namely (i) whether we should expect the minimization algorithm to converge, and (ii) whether the local minima of the non-convex problem correspond to minimizers of the convex SDP. We now investigate both of these questions.

While the application of FBS for minimizing the proposed semidefinite program is new, the convergence of FBS for non-convex problems is well-studied. Reference [53] presents conditions for which FBS converges with non-convex constraints. In particular, the problem must be semi-algebraic, meaning both the constraints and the epigraph of the objective can be written as the set of solutions to a system of polynomial equations.[5] Fortunately, such results apply to the formulation (8). The following result makes this statement rigorous.

Proposition 1. *Suppose we apply FBS (Algorithm 1) to solve the problem stated in (8). If we use the step size $\tau = \alpha/\|\mathbf{T}\|_2$ with $0 < \alpha < 1$, then the sequence of iterates $\{\mathbf{L}^{(t)}\}$ converges to a critical point of the problem in (8).*

PROOF. The function $\|\boldsymbol{\ell}_k\|_2^2$ is a polynomial in the entries of $\mathbf{L}$. The constraint set in (8) is the solution to the polynomial system $\|\boldsymbol{\ell}_k\|_2^2 = 1, \forall k$ and is thus semi-algebraic. The objective function, being a quadratic form, is also trivially semi-algebraic. By Theorem 5.3 of [53], we know that the sequence of iterates $\{\mathbf{L}^{(t)}\}$ converges, provided the step size is bounded from above by the inverse of the Lipschitz constant of the gradient of the objective. For our quadratic objective, the Lipschitz constant is merely the spectral radius (ℓ_2-norm) of $\mathbf{T}$. □

Note that the Jacobi preconditioner in Section III-C results in a problem of the same form as (8), but with constraints of the form $\|\tilde{\boldsymbol{\ell}}_k\|_2^2 = D_{k,k}^2$ and the step size $\tau = \alpha/\|\widetilde{\mathbf{T}}\|_2$. Consequently, Proposition 1 still applies. Note that this result has the caveat that we are not guaranteed to find a (global) minimizer, but rather stationary points, although we generally observe minimizers in practice. Nonetheless, this convergence guarantee is considerably stronger than what is known for other low-complexity SDP methods, such as those inspired by Burer and Montiero [54], which rely on non-convex augmented Lagrangian schemes for which no guarantees currently exist.

The second question to ask is whether the local minima of our non-convex formulation in (8) correspond to minimizers of the convex SDP shown in (7). Interestingly, when the factors $\mathbf{L}$ and $\mathbf{L}^T$ are not constrained to be triangular, local minimizers of (8) are known to yield optimal minimizers for the SDP (7) (see [55], Corollary 3.6). Nevertheless, we have found that it is better to enforce the triangular constraint in practice as it substantially simplifies the architecture detailed next.

IV. VLSI Architecture

We now propose a systolic VLSI architecture that implements TASER and enables high-throughput data detection at low hardware complexity.

A. Architecture Overview

Figure 1 shows the proposed triangular systolic array consisting of $N(N+1)/2$ processing elements (PEs), which mainly perform multiply-accumulate (MAC) operations. Each PE is associated with an entry $\widetilde{L}_{i,j}^{(t-1)}$ of the lower-triangular matrix $\widetilde{\mathbf{L}}^{(t-1)}$ and stores $\widetilde{L}_{i,j}^{(t-1)}$ as well as the value $V_{i,j}^{(t)}$ of the $\mathbf{V}^{(t)}$ matrix (cf. Algorithm 1). All PEs that are part of the same column receive data from a column-broadcast unit (CBU); all PEs that are part of the same row receive data from a row-broadcast unit (RBU).

In the kth cycle during the tth TASER iteration, the ith RBU sends the value $\widetilde{L}_{i,k}^{(t-1)}$ to all PEs on row i, while the jth CBU

[4] In the conference paper [1], we mistakenly stated $\widetilde{\mathbf{L}} = \mathbf{D}\mathbf{L}$.

[5] The authors of [53] actually prove results for the broader class of Kurdyka-Łojasiewicz functions, of which semi-algebraic functions are a special case.

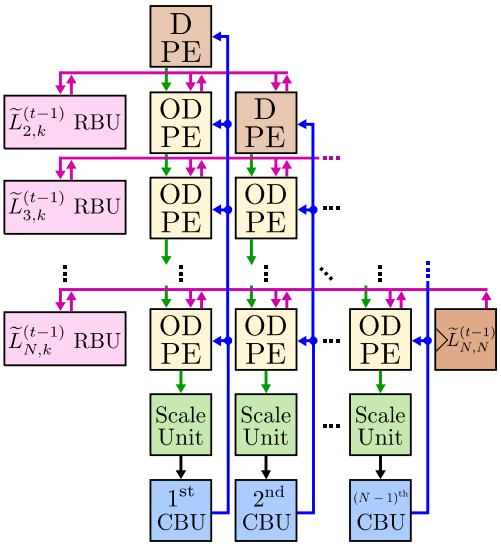


Fig. 1. High-level block diagram of TASER. We use a systolic array of processing elements (PEs) for the diagonal (D) and off-diagonal (OD) elements, which enables high throughput at low hardware complexity.

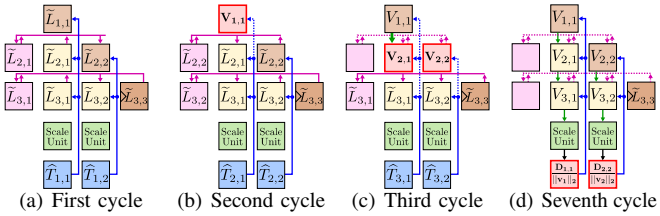


Fig. 2. Different cycles for the tth iteration on an $N = 3$ TASER array. The symbols inside the PEs correspond to the quantity of interest, for each cycle, stored in the PE, while the symbols inside the RBUs and CBUs correspond to the quantity being transmitted by these units. All the $\widetilde{L}$ values correspond to the $(t-1)$th iteration, while the V values are from the tth iteration.

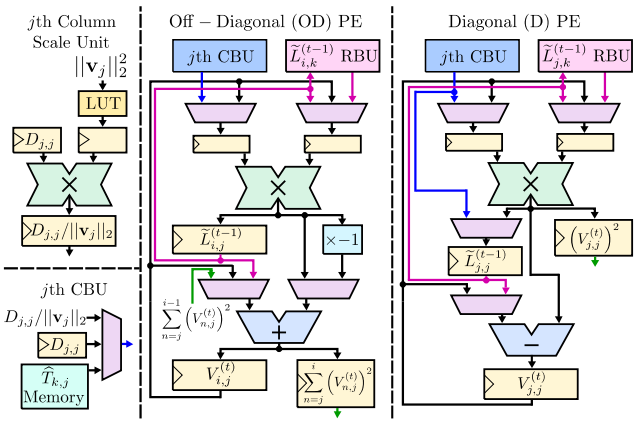


Fig. 3. Architecture details of the column-scale unit (CSU), the column-broadcast unit (CBU), and the off-diagonal (OD) and diagonal (D) processing elements (PEs).

sends $\widehat{T}_{k,j}$ to all PEs on column j. We assume that the (scaled) matrix $\widetilde{\widehat{\mathbf{T}}} = 2\tau\widetilde{\mathbf{T}}$ has been computed in a pre-processing step and is stored in a memory (see Section IV-C for more details on the memory implementation). The $\widetilde{L}_{i,k}^{(t-1)}$ value coming from each RBU is taken from the (i,k) PE and sent to all other PEs in the same row.

With the data received from the CBU and the RBU, each PE performs MAC operations until the result $\widetilde{\mathbf{L}}^{(t-1)}\widehat{\mathbf{T}}$ on line 4 of Algorithm 1 is computed. To include the subtraction on line 4, the operation $\widetilde{L}_{i,j}^{(t-1)} - \widetilde{L}_{i,1}^{(t-1)}\widehat{T}_{1,j}$ is performed in the first cycle of each TASER iteration and stored in the accumulator. During subsequent cycles, the products $\widetilde{L}_{i,k}^{(t-1)}\widehat{T}_{k,j}$, with $2 \le k \le N$, are sequentially subtracted from the accumulator. Since the matrix $\widetilde{\mathbf{L}}$ is lower-triangular, we have $\widetilde{L}_{i,k'} = 0$ if $i < k'$. Hence, we avoid the subtraction of $\widetilde{L}_{i,k'}^{(t-1)}\widehat{T}_{k',j}$ as they are zero. This implies that the $V_{i,j}^{(t)}$ values of the PEs in the ith row of the systolic array are computed after only i clock cycles, so the matrix $\mathbf{V}^{(t)}$ on line 4 is completed after N cycles.

An example for an $N = 3$ array is shown in Figure 2(a). In the first cycle of the tth iteration, the $(1,1)$ PE has access to the values $\widetilde{L}_{1,1}^{(t-1)}$ and $\widehat{T}_{1,1}$, so it can compute $V_{1,1}^{(t)} = \widetilde{L}_{1,1}^{(t-1)} - \widetilde{L}_{1,1}^{(t-1)}\widehat{T}_{1,1}$. In the same cycle, the PEs on the second row perform their first MAC operation, which leaves $\widetilde{L}_{2,j}^{(t-1)} - \widetilde{L}_{2,1}^{(t-1)}\widehat{T}_{1,j}$ in their accumulators.

In the second cycle, the PEs on the second row receive $\widetilde{L}_{2,2}^{(t-1)}$ via the RBU and $\widehat{T}_{2,j}$ via the CBUs (see Figure 2(b)), so they can finish computing $V_{2,j}^{(t)} = \widetilde{L}_{2,j}^{(t-1)} - \widetilde{L}_{2,1}^{(t-1)}\widehat{T}_{1,j} - \widetilde{L}_{2,2}^{(t-1)}\widehat{T}_{2,j}$. In addition, in this same cycle, the (1,1) PE can use its MAC unit to square its $V_{1,1}^{(t)}$ value. This result will be available and sent to the next PE in the same column on the following cycle, which is represented with the green arrow in Figure 2(c).

In the third cycle, the $(2,1)$ PE has access to ${V_{1,1}^{(t)}}^2$ (from the $(1,1)$ PE) and $V_{2,1}^{(t)}$ (stored internally), so it can use its MAC unit to square $V_{2,1}^{(t)}$ and add it to ${V_{1,1}^{(t)}}^2$ (see Figure 2(c)). The result will be the sum of the squares of the first two elements of the first column of $\mathbf{V}^{(t)}$ and, in the next cycle, the result will be available and sent to the next PE in the same column (for this example, the $(3,1)$ PE), so it can repeat the same procedure. This process is replicated in all the columns and repeated until all the PEs of the array have executed it. By doing so, the squared ℓ_2-norm of each column of $\mathbf{V}^{(t)}$ is computed after $N+1$ clock cycles, just one cycle after $\mathbf{V}^{(t)}$ is completed. In the $(N+2)$th cycle, the squared ℓ_2-norm for the jth column is passed to a scale unit, which computes its inverse square root and multiplies the result with $D_{j,j}$. This operation takes two clock cycles to complete, so its result is ready in the $(N+4)$th cycle.

In the $(N+4)$th cycle, the scaling factor $D_{j,j}/\|\mathbf{v}_j\|_2$ (where $\mathbf{v}_j$ is the jth column of $\mathbf{V}^{(t)}$) is sent to all the PEs in the same column via the CBU, as shown in Figure 2(d). Then, in the $(N+5)$th and final cycle of the iteration, all PEs multiply the received scaling factor to their associated $V_{i,j}^{(t)}$ value to obtain the next iterate $\widetilde{L}_{i,j}^{(t)}$, thus completing the proximal step on line 5 of Algorithm 1.

Prior to decoding the next symbol, line 2 of Algorithm 1 must be executed; this is accomplished using the CBUs, which send the $D_{j,j}$ values to the diagonal PEs, while the off-diagonal PEs clear their $\widetilde{L}_{i,j}^{(t-1)}$ registers.

B. Processing Element

We use two slightly distinct types of PEs in our systolic array: (i) off-diagonal (OD) PEs and (ii) diagonal (D) PEs (see Figure 3). Both PE types support the following four operation modes:

1) Initialization of $\widetilde{\mathbf{L}}$*:* This mode is used for line 2 of Algorithm 1. All off-diagonal PEs initialize $\widetilde{L}_{i,j}^{(t-1)} = 0$; the diagonal PEs initialize their states with $D_{j,j}$ received from the CBU.

2) Matrix multiplication: This mode is used to compute line 4 of Algorithm 1. The multiplier uses the inputs from both broadcast signals. In the first cycle of the matrix-matrix multiplication procedure, the multiplier's output is subtracted from $\widetilde{L}_{i,j}^{(t-1)}$; in all other cycles, it is subtracted from the accumulator. Since each PE stores its own $\widetilde{L}_{i,j}^{(t-1)}$ value, in the kth cycle, all the PEs in the kth column use their internal $\widetilde{L}_{i,k}^{(t-1)}$ to feed the multiplier, instead of the signals coming from the RBU.

3) Squared ℓ_2-norm calculation: This mode is used for line 5 of Algorithm 1. Both of the multiplier's inputs are $V_{i,j}^{(t)}$. For the D-PEs, the result is passed to the next PE in the same column. For the OD-PEs, the output of the multiplier is added to the $\sum_{n=j}^{i-1} \left(V_{n,j}^{(t)}\right)^2$ value from the preceding PE in the same column; the result $\sum_{n=j}^{i} \left(V_{n,j}^{(t)}\right)^2$ is sent to the next PE or to the scale unit, if the PE is in the last row.

4) Scaling: This mode completes line 5 of Algorithm 1. One of the multiplier's inputs is $V_{i,j}^{(t)}$ and the other is the value $D_{j,j}/\|\mathbf{v}_j\|_2$, which was computed previously by the scale unit and received through the CBU. The result is $\widetilde{L}_{i,j}^{(t)}$ and is stored in every PE as the $\widetilde{L}_{i,j}^{(t-1)}$ of the next iteration.

C. Implementation Details

To demonstrate the efficacy of TASER and the proposed triangular systolic array, we implemented FPGA and ASIC reference designs for various array sizes N. All designs were designed and optimized in Verilog on register-transfer level (RTL). The implementation details are as follows:

1) Fixed-point design parameters: To minimize the hardware complexity while maintaining near-optimal error-rate performance, all our designs use 14 bit fixed-point numbers. All PEs, except for the ones in the bottom row of the triangular array, use 8 fraction bits to represent $\widetilde{L}_{i,j}^{(t-1)}$ and $V_{i,j}^{(t)}$; the PEs in the bottom row use 7 fraction bits. For the element $\widetilde{L}_{N,N}$, we do not use a PE and store the value (which remains constant) in a register with 5 fraction bits.

2) Inverse square-root computation: The inverse square-root operation in the scale unit is implemented using a look-up table (LUT), which we synthesized using random logic. Each LUT consists of 2^{11} entries with 14 bit per word, of which 13 are fraction bits.

3) $\widehat{\mathbf{T}}$*-matrix memories:* For the FPGA designs, the $\widehat{T}_{k,j}$ memories are implemented on LUTs used as distributed RAM (i.e., no block RAMs were used); for the ASIC designs, we use latch arrays built from standard cells [56] in order to minimize the circuit area.

4) RBU and CBU design: The RBUs are implemented differently for the FPGA and ASIC designs. For the FPGA designs, the RBU of the ith row is an i-input multiplexer that receives data from all the PEs on its row, and also sends the appropriate $\widetilde{L}_{i,k}^{(t-1)}$ to these PEs. For the ASIC designs, the RBU consists of a bidirectional bus, where each PE on its row uses a tri-state buffer to send data through it one at a time, while all the PEs on the same row acquire data from it. A similar approach is used for the CBUs: We use multiplexers for the FPGA designs and busses for the ASIC designs. For both target architectures, the output of the ith RBU connects to i PEs. This path suffers from large fan-out for large values of i, eventually becoming the critical path for large systolic arrays. The same behavior applies to the CBUs. In order to shorten these critical paths in our architecture, we place pipeline registers at the inputs and outputs of the respective broadcast units. While this approach entails two penalty cycles per TASER iteration, the overall detection throughput is increased as we achieve a substantially higher clock frequency.

V. Implementation Results and Comparison

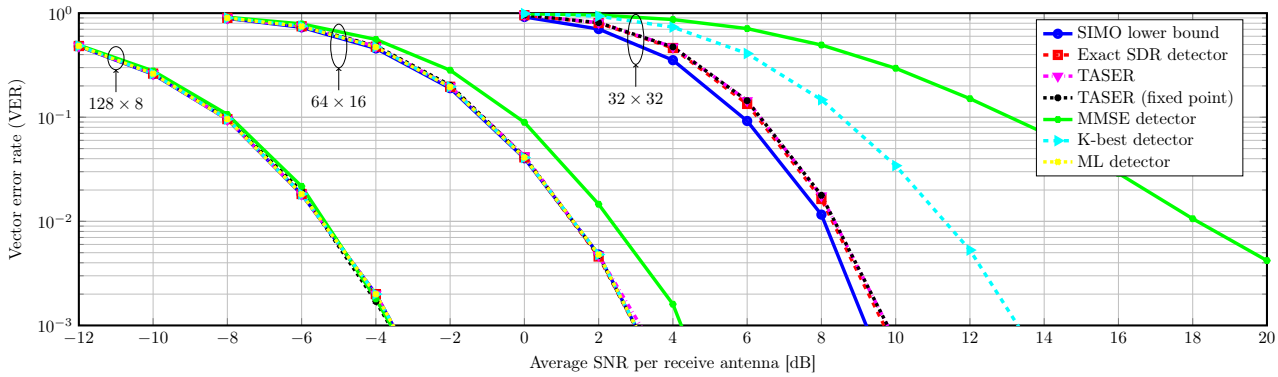
We now provide error-rate performance results for coherent data detection in massive MU-MIMO systems and for JED in massive SIMO systems. We then show reference FPGA and ASIC implementation results which we compare to existing designs for massive MU-MIMO systems.

A. Error-Rate Performance

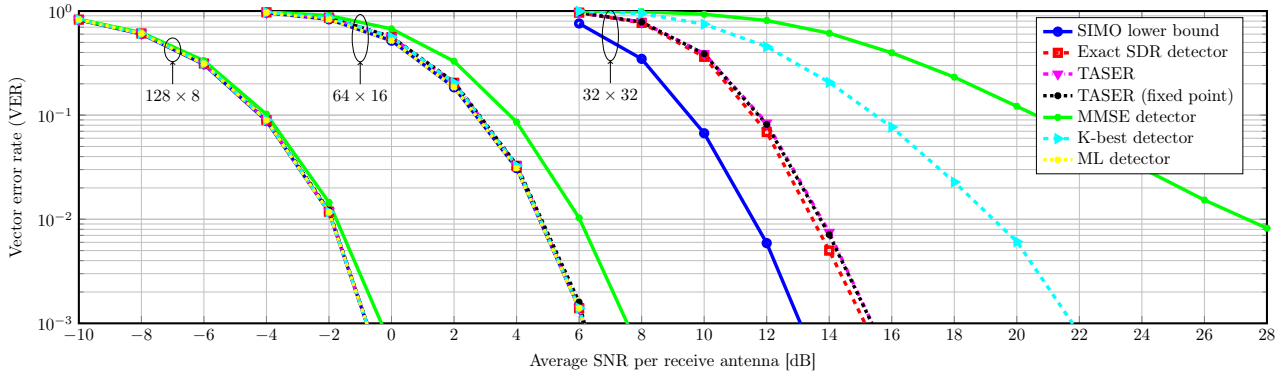
1) Coherent massive MU-MIMO data detection: Figures 4(a) and 4(b) show vector error rate (VER) simulation results for TASER with BPSK and QPSK modulation, respectively.[6] We show simulation results for coherent data detection with i.i.d. flat Rayleigh fading in tall 128×8 and 64×16 systems, as well as a square 32×32 large MU-MIMO system (we use the notation $B \times U$). We show the performance of ML detection (only for the $U = 8$ and $U = 16$ systems; computed using the sphere-decoding algorithm in [10]), exact SDR detection from (6), linear MMSE detection, and the real-valued K-best algorithm as detailed in [57] with $K = 5$. As a baseline, we also include the performance of the SIMO lower bound.

For the 128×8 massive MIMO system, we see that all detectors approach optimal performance (even the SIMO lower bound); this is a well-known result from the large MIMO literature [2]–[5]. For the 64×16 massive MIMO system, only the linear MMSE detector suffers from a (rather small) performance loss; all the other detectors perform equally well. For the more challenging square 32×32 massive MIMO system, we see that TASER achieves near-ML performance and significantly outperforms linear MMSE detection and the K-best algorithm (note that, even with the sphere decoder, ML detection exhibits prohibitive complexity). We also show the fixed-point performance of our TASER hardware design, denoted by "fp" in Figures 4(a) and 4(b), which demonstrates a small implementation loss (less than 0.2 dB SNR at 1% VER).

[6]The vector error rate (VER) corresponds to $\mathrm{P}[\mathbf{s} \neq \hat{\mathbf{s}}]$, which is the probability of detecting a different vector $\hat{\mathbf{s}}$ than the transmitted one $\mathbf{s}$.



(a) BPSK



(b) QPSK

Fig. 4. Vector error rate (VER) for three different $B \times U$ large-MIMO system configurations. Even for square massive MU-MIMO systems (see the 32×32 system), TASER achieves near-optimal VER performance (close-to-ML and the SIMO lower bound) and approaches the performance of the exact SDR detector. For systems with more BS antennas than users (see the 128×8 system), all of the considered data detectors approach optimal performance.

Figures 5(a) and 5(b) show the trade-off between the throughput of TASER for the FPGA design (see Section V-C for the details) and the minimum SNR required to achieve 1% VER for the coherent data detection in large-MIMO systems. We also include the SIMO lower bound and the performance of linear MMSE detection as a reference. The MMSE detector serves as a fundamental performance limit of the conjugate gradient least-squares (CGLS) detector [20], the Neumann-series detector [8], the optimized coordinate-descent (OCD) detector [21], and the Gauss-Seidel (GS) detector [22]. The maximum number of TASER iterations t_{max} enables us to tune the performance/complexity trade-off; only a few iterations are sufficient to outperform linear detection. We also see that TASER delivers near-ML performance and achieves throughputs from 10 Mb/s to 80 Mb/s for the FPGA design.

2) JED in massive SIMO systems: Figures 6(a) and 6(b) show VER simulation results for TASER with BPSK and QPSK modulation, respectively. The simulations are for a $B = 16$ BS antenna and $K = 16$ time slot SIMO system; we perform $t_{\text{max}} = 5$ iterations and use an i.i.d. flat Rayleigh block-fading channel model. We include the performance of the SIMO detection with both perfect receiver channel state information (CSIR) and channel estimation (CHEST), exact SDR detection from (6), and ML JED detection (which is computed using the algorithm proposed in [41]). We see that TASER achieves near-optimal performance, as it is close to a system with perfect CSIR, and outperforms detection via SIMO CHEST, while offering performance similar to the ML JED and exact SDR detection at a manageable complexity. We note that the trade-offs between throughput and SNR performance behave analogously to the massive MU-MIMO case.

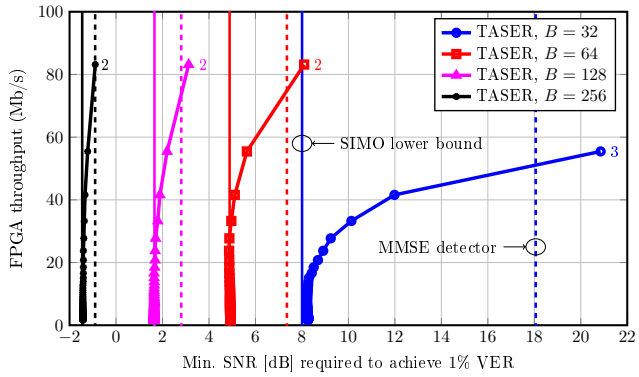
TABLE I
COMPUTATIONAL COMPLEXITY OF DIFFERENT DATA DETECTION ALGORITHMS FOR MASSIVE MIMO SYSTEMS

Algorithm	Computational complexity[a]
BPSK TASER	$t_{\text{max}}(\frac{1}{3}U^3 + \frac{5}{2}U^2 + \frac{37}{6}U + 4)$
QPSK TASER	$t_{\text{max}}(\frac{8}{3}U^3 + 10U^2 + \frac{37}{3}U + 4)$
CGLS [20]	$(t_{\text{max}} + 1)(4U^2 + 20U)$
Neumann [8]	$(t_{\text{max}} - 1)2U^3 + 2U^2 - 2U$
OCD [21]	$t_{\text{max}}(8BU + 4U)$
GS [22]	$t_{\text{max}}6U^2$

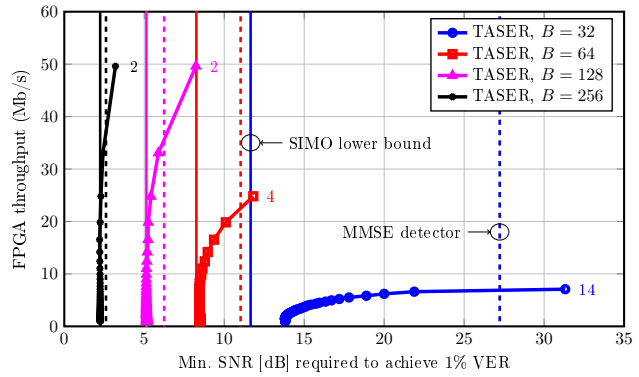
[a]The complexity is measured by the number of real-valued multiplications for t_{max} iterations. Complex-valued multiplications are assumed to require four real-valued multiplications. All results ignore the preprocessing complexity.

B. Computational Complexity

We now compare the computational complexity of TASER with other large-scale MIMO data-detection algorithms proposed in the literature, namely the CGLS detector [20], the Neumann-series detector [8], the OCD detector [21], and the

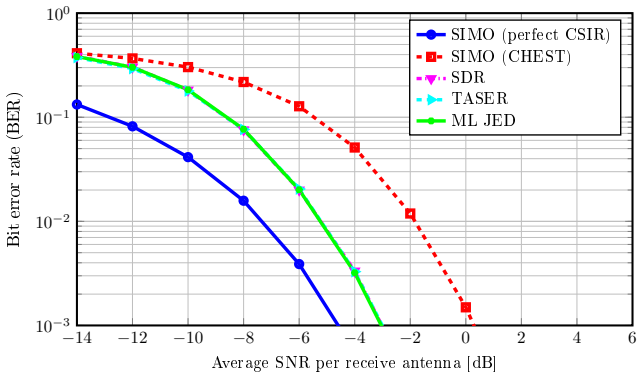


(a) BPSK

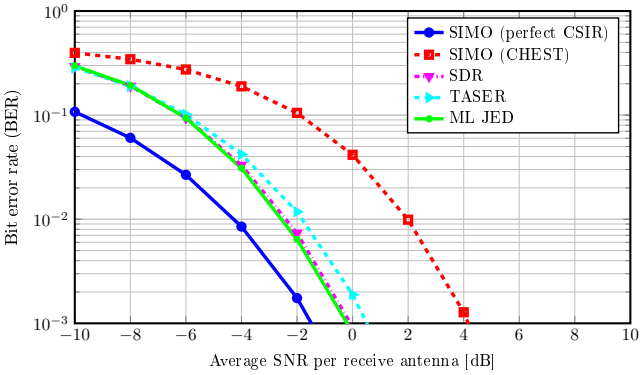


(b) QPSK

Fig. 5. Throughput for the FPGA design vs. performance trade-off for a 32-user system. Vertical solid lines represent the SIMO lower bound; dashed lines represent linear MMSE performance. TASER outperforms linear detectors in almost all operation regimes. The number next to the markers corresponds to the number of TASER iterations $t_{\max}$.



(a) BPSK



(b) QPSK

Fig. 6. Vector error rate (VER) for a SIMO system with 16 BS antennas with transmission over $K = 16$ time slots. TASER-based JED achieves near-optimal VER performance (close-to perfect CSIR) and achieves performance similar to the exact SDR detector and ML JED; channel estimation (CHEST) followed by SIMO detection entails a 3 dB SNR loss.

GS detector [22]. Table I shows the number of real-valued multiplications for $t_{\max}$ iterations. We see that the complexity of TASER (for BPSK and QPSK) and the Neumann-series detector scales with $t_{\max}U^3$, whereas TASER is slightly more complex; CGLS and GS both scale with $t_{\max}U^2$, whereas GS is slightly more complex; OCD scales with $t_{\max}BU$. Evidently, the near-ML performance of TASER comes at the cost of high computational complexity. In contrast, CGLS, OCD, and GS are rather inexpensive, but also perform poorly in square systems (see the 32×32 results in Figure 4). We finally note that TASER can be used for JED—the other (approximate) linear detectors cannot be used for this application.

C. FPGA Implementation Results

To demonstrate the effectiveness of TASER, we developed several FPGA designs for systolic array sizes of $N = 9$, $N = 17$, $N = 33$, and $N = 65$. The FPGA designs were implemented using Xilinx Vivado Design Suite and optimized for a Xilinx Virtex-7 XC7VX690T FPGA. The associated implementation results are shown in Table II. As expected, the resource utilization increases quadratically with the array size N. For the $N = 9$ and $N = 17$ arrays, the critical path is in the PEs' MAC unit; for the $N = 33$ and $N = 65$ arrays, the critical path is in the row broadcast multiplexers, which limits the throughput of the $N = 65$ array.

TABLE II
IMPLEMENTATION RESULTS ON A XILINX VIRTEX-7 XC7VX690T FPGA FOR DIFFERENT TASER ARRAY SIZES

Array size	$N = 9$	$N = 17$	$N = 33$	$N = 65$
BPSK users / time slots	8	16	32	64
QPSK users / time slots	4	8	16	32
Slices	1 467	4 350	13 787	60 737
LUTs	4 790	13 779	43 331	149 942
FFs	2 108	6 857	24 429	91 829
DSP48s	52	168	592	2 208
Max. clock freq. [MHz]	232	225	208	111
Min. latency [clock cycles]	16	24	40	72
Max. throughput [Mb/s]	116	150	166	98
Power estimate[a] [W]	0.6	1.3	3.6	7.3

[a]Statistical power estimation at max. clock freq. and 1.0 V supply voltage.

In Table III, we compare TASER to the few existing large MIMO data detector designs, namely CGLS detector [20], the Neumann-series detector [8], the OCD detector [21], and the GS detector [22]. All of these detectors have been implemented on the same FPGA and for a 128×8 large-MIMO system. TASER achieves comparable throughput to the CGLS and GS designs and significantly lower latency than the Neumann-series and CD detectors. In terms of the hardware efficiency (measured in terms of throughput per FPGA LUTs), our design performs similarly to CGLS, Neumann and GS, and inferior to the CD

TABLE III
COMPARISON OF 128 × 8 LARGE-MIMO DETECTORS ON A XILINX VIRTEX-7 XC7VX690T FPGA

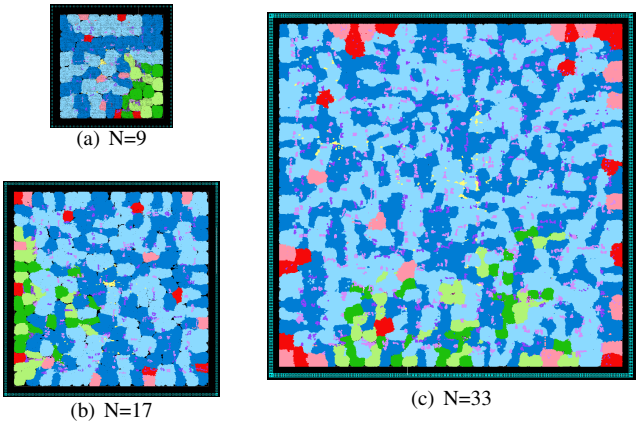
Detection algorithm	TASER	TASER	CGLS [20]	Neumann [8]	OCD [21]	GS [22]
Error-rate performance	Near-ML	Near-ML	Near-MMSE	Near-MMSE	Near-MMSE	Near-MMSE
Modulation scheme	BPSK	QPSK	64-QAM	64-QAM	64-QAM	64-QAM
Preprocessing	Not included	Not included	Included	Included	Included	Included
Iterations $t_{\max}$	3	3	3	3	3	1
Slices	1 467 (1.35 %)	4 350 (4.02 %)	1 094 (1 %)	48 244 (44.6 %)	13 447 (12.4 %)	n.a.
LUTs	4 790 (1.11 %)	13 779 (3.18 %)	3 324 (0.76 %)	148 797 (34.3 %)	23 955 (5.53 %)	18 976 (4.3 %)
FFs	2 108 (0.24 %)	6 857 (0.79 %)	3 878 (0.44 %)	161 934 (18.7 %)	61 335 (7.08 %)	15 864 (1.8 %)
DSP48s	52 (1.44 %)	168 (4.67 %)	33 (0.9 %)	1 016 (28.3 %)	771 (21.5 %)	232 (6.3 %)
BRAM18s	0 (0 %)	0 (0 %)	1 (0.03 %)	32[a] (1.08 %)	1 (0.03 %)	12[a] (0.41 %)
Clock frequency [MHz]	232	225	412	317	263	309
Latency [clock cycles]	48	72	951	196	795	n.a.
Throughput [Mb/s]	38	50	20	621	379	48
Throughput/LUTs	7 933	3 629	6 017	4 173	15 821	2 530

[a]These designs use BRAM36s, which are equal to two BRAM18s.

TABLE IV
ASIC IMPLEMENTATION RESULTS FOR DIFFERENT TASER ARRAY SIZES

Array size	$N = 9$	$N = 17$	$N = 33$
BPSK users / time slots	8	16	32
QPSK users / time slots	4	8	16
Core area [μm^2]	149 738	482 677	1 382 318
Core density [%]	69.86	68.89	72.89
Cell area [GE[a]]	148 264	471 238	1 427 962
Max. clock freq. [MHz]	598	560	454
Min. latency [clock cycles]	16	24	40
Max. throughput [Mb/s]	298	374	363
Power estimate[b] [mW]	41	87	216

[a]One gate equivalent (GE) refers to the area of a unit-sized NAND2 gate.
[b]Post-place-and-route power estimation at max. clock freq. and 1.1 V.



(a) N=9 (b) N=17 (c) N=33

Fig. 7. Layout of the TASER ASIC designs for $N = 9$, $N = 17$ and $N = 33$ array sizes. The different modules of the design were colored in the following way: PEs are colored in blue, memories in red, the scale units in green, the busses of the RBUs and CBUs in purple, and the control unit in yellow. Light and dark versions of the same color are alternated according to the order in which the modules appear in the hardware description code.

design. For the 128×8 massive MIMO system, all detectors achieve near-ML performance. However, when considering the 32×32 large MIMO system (see Figures 4(a) and 4(b)), TASER significantly outperforms the error-rate performance of all these reference designs. We conclude by noting that the CGLS, Neumann, OCD, and GS detectors are able to support 64-QAM, whereas TASER is limited to either BPSK or QPSK. This limitation negatively affects the throughput and hardware-efficiency of TASER, as the throughput of the other (approximate) methods scales linearly in the number of bits per symbol—the provided throughput and hardware-efficiency results favor the CGLS, Neumann, OCD, and GS detectors.

D. ASIC Implementation Results

We also developed several reference ASIC designs for systolic array sizes of $N = 9$, $N = 17$ and $N = 33$. The ASIC designs were implemented using Synopsys DC and IC Compiler and optimized for a TSMC 40nm CMOS process. The associated implementation results are shown in Table IV. As for our FPGA designs, the silicon area increases quadratically with the array size N. This can be verified both visually in Figure 7 as well as numerically in Table V, where we see that the unit areas of each PE and scale unit remain nearly constant, while the total area of these units increases with N^2. As expected, the total area of the $\widehat{T}_{k,j}$ memories increases with N, as these memories contain an $N \times N$ matrix. The critical path for the $N = 9$, $N = 17$, and $N = 33$ array is within the PE's MAC unit, the inverse square root unit, and the row broadcasting bus, respectively.

In Table VI, we compare our TASER ASIC implementation to the Neumann-series detector in [23], which is—to the best of our knowledge—the only ASIC design for massive MU-MIMO systems. While the latter offers a significantly higher throughput than our design, TASER's reduced area and power consumption result in superior hardware-efficiency (measured in throughput per cell area) and power-efficiency (measured in energy per bit). Furthermore, TASER enables near-ML performance for massive MU-MIMO systems where the number of users is in the same range as the number of BS antennas (see Figures 4(a) and 4(b)). We note that the comparison presented on Table VI is not entirely fair. TASER does not include preprocessing circuitry, whereas the Neumann-series detector [23] includes preprocessing circuitry

TABLE V
AREA BREAKDOWN FOR DIFFERENT TASER ASIC ARRAY SIZES IN GATE EQUIVALENTS (GES)

Array size	$N = 9$		$N = 17$		$N = 33$	
Element	Unit area	Total area	Unit area	Total area	Unit area	Total area
PEs	2 391 (1.6 %)	105 198 (70.9 %)	2 404 (0.5 %)	365 352 (77.5 %)	2 086 (0.1 %)	1 168 254 (81.8 %)
Scale units	6 485 (4.4 %)	25 941 (17.5 %)	6 315 (1.3 %)	50 521 (10.7 %)	5 945 (0.4 %)	95 125 (6.6 %)
$\widehat{T}_{k,j}$ memories	734 (0.5 %)	5 873 (4.0 %)	1 451 (0.3 %)	23 220 (4.9 %)	2 888 (0.2 %)	92 426 (6.5 %)
Control unit	459 (0.3 %)	459 (0.3 %)	728 (0.2 %)	728 (0.2 %)	1 259 (0.1 %)	1 259 (0.1 %)
Miscellaneous	–	10 793 (7.3 %)	–	31 417 (6.7 %)	–	70 898 (5.0 %)

TABLE VI
COMPARISON OF DATA DETECTION ASICS FOR 128 BS ANTENNA, 8 USER LARGE-MIMO SYSTEMS

Detection algorithm	TASER	TASER	Neumann [23]
Error-rate performance	Near-ML	Near-ML	Near-MMSE
Modulation scheme	BPSK	QPSK	64-QAM
Preprocessing	Not included	Not included	Included
Iterations	3	3	3
CMOS technology [nm]	40	40	45
Supply voltage [V]	1.1	1.1	0.81
Clock freq. [MHz]	598	560	1 000 (1 125[a])
Throughput [Mb/s]	99	125	1 800 (2 025[a])
Core area [mm^2]	0.150	0.483	11.1 (8.77[a])
Core density [%]	69.86	68.89	73.00
Cell area[b] [kGE]	142.4	448.0	12 600
Power[c] [mW]	41.25	87.10	8 000 (13 114[a])
Throughput/cell area[a] [b/(s×GE)]	695	279	161
Energy/bit[a] [pJ/b]	417	697	6 476

[a]Technology scaling to 40 nm and 1.1 V assuming: $A \sim 1/\ell^2$, $t_{pd} \sim 1/\ell$, and $P_{dyn} \sim 1/(V_\ell^2 \ell)$ [58].
[b]Excluding the gate count of memories.
[c]At maximum clock frequency and given supply voltage.

and was optimized for wideband systems that use single-carrier frequency-division multiple access.

We finally note that there exists a plethora of data-detector ASICs for traditional, small-scale MIMO systems (see [9], [10], [42], [47], [59]–[63] and the references therein). While most of these designs achieve near-ML performance and/or throughputs in the Gb/s regime in small-scale MIMO systems, their efficacy for large MIMO is unexplored—a corresponding algorithm and hardware-level comparison is left for future work.

VI. CONCLUSIONS

We have proposed—to the best of our knowledge—the first data-detector implementation that uses semidefinite relaxation. Our novel algorithm, referred to as Triangular Approximate SEmidefinite Relaxation (TASER), is suitable for coherent data detection in massive MU-MIMO systems, as well as joint channel estimation and data detection (JED) in large SIMO systems. We have developed a corresponding systolic VLSI architecture and implemented FPGA and ASIC reference designs. Our results have shown that TASER achieves comparable hardware-efficiency as existing massive MU-MIMO data detectors, while providing near-ML performance, even for systems where the number of users is comparable to the number of BS antennas. Hence, for systems supporting a large number of low-rate users (e.g., 16 users or more) where BPSK and QPSK transmission is sufficient, TASER provides a viable alternative to sub-optimal, linear data-detection methods, or optimal but computationally expensive non-linear methods. We also note that TASER can be used in so-called overloaded systems, i.e., systems with more users than BS antennas—such a scenario may be of interest in large sensor networks or for the internet of things (IoT).

There are many avenues for future work. Traditional SDR-based data detection only supports BPSK and QPSK transmission and hard-output data detection. Extending TASER to support higher-order modulation schemes using the methods in [32], [33] is the subject of ongoing research. Furthermore, developing efficient ways to compute soft-output values (in the form of log-likelihood ratio values) within TASER is left for future work. Finally, SDR-based data detection for JED in MU-MIMO systems is an interesting open research topic.

REFERENCES

[1] O. Castañeda, T. Goldstein, and C. Studer, "FPGA design of approximate semidefinite relaxation for data detection in large MIMO wireless systems," in *Proc. IEEE Intl. Conf. on Circuits and Systems (ISCAS)*, May 2016.

[2] 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.

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

[4] J. Hoydis, S. Ten Brink, and M. Debbah, "Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 2, pp. 160–171, Feb. 2013.

[5] E. Larsson, O. Edfors, F. Tufvesson, and T. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[6] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5G be?," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[7] L. Lu, G. Li, A. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 742–758, Oct. 2014.

[8] 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.

[9] K. Wong, C. Tsui, R. Cheng, and W. Mow, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," in *Proc. IEEE International Conference on Circuits and Systems (ISCAS)*, May 2002, vol. 3, pp. 273–276.

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

[11] K. Vardhan, S. Mohammed, A. Chockalingam, and B. Rajan, "A low-complexity detector for large MIMO systems and multicarrier CDMA systems," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 3, pp. 473–485, Apr. 2008.

[12] 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.

[13] S. Wu, L. Kuang, Z. Ni, J. Lu, D. Huang, and Q. Guo, "Low-complexity iterative detection for large-scale multiuser MIMO-OFDM systems using approximate message passing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 902–915, Oct. 2014.

[14] P. Svac, F. Meyer, E. Riegler, and F. Hlawatsch, "Soft-heuristic detectors for large MIMO systems," *IEEE Transactions on Signal Processing*, vol. 61, no. 18, pp. 4573–4586, Sep. 2013.

[15] 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 ICCS*, Nov 2014, pp. 477–481.

[16] 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.

[17] L. Liu, C. Yuen, Y. L. Guan, Y. Li, and Y. Su, "A low-complexity Gaussian message passing iterative detector for massive MU-MIMO systems," in *Proc. IEEE ICICS*, Dec 2015.

[18] C. Jeon, R. Ghods, A. Maleki, and C. Studer, "Optimality of large MIMO detection via approximate message passing," in *IEEE International Symposium on Information Theory (ISIT)*, June 2015, pp. 1227–1231.

[19] K. Li, B. Yin, M. Wu, J. R. Cavallaro, and C. Studer, "Accelerating massive MIMO uplink detection on GPU for SDR systems," in *IEEE Dallas Circuits and Systems Conference (DCAS)*, Oct. 2015.

[20] B. Yin, M. Wu, J. Cavallaro, and C. Studer, "VLSI Design of Large-Scale Soft-Output MIMO Detection Using Conjugate Gradients," in *Proc. IEEE ISCAS*, May 2015, pp. 1498–1501.

[21] M. Wu, C. Dick, J. Cavallaro, and C. Studer, "FPGA design of a coordinate-descent detector for large-MIMO," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016.

[22] Z. Wu, C. Zhang, Y. Xue, S. Xu, and Z. You, "Efficient architecture for soft-output massive MIMO detection with Gauss-Seidel method," in *Proc. IEEE Intl. Conf. on Circuits and Systems (ISCAS)*, May 2016.

[23] 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.

[24] Z.-Q. Luo, W.-k. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Sig. Proc. Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.

[25] J. Jaldén and B. Ottersten, "The diversity order of the semidefinite relaxation detector," *IEEE Transactions on Information Theory*, vol. 54, no. 4, pp. 1406–1422, Apr. 2008.

[26] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, Jan. 2009.

[27] T. Goldstein, C. Studer, and R. G. Baraniuk, "A field guide to forward-backward splitting with a FASTA implementation," *arXiv preprint: 1411.3406*, Nov. 2014.

[28] P. H. Tan and L. K. Rasmussen, "The application of semidefinite programming for detection in CDMA," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 8, pp. 1442–1449, Aug. 2001.

[29] W.-K. Ma, T. N. Davidson, K. M. Wong, Z.-Q. Luo, and P.-C. Ching, "Quasi-maximum-likelihood multiuser detection using semi-definite relaxation with application to synchronous CDMA," *IEEE Transactions on Signal Processing*, vol. 50, no. 4, pp. 912–922, Apr. 2002.

[30] B. Steingrimsson, Z.-Q. Luo, and K. M. Wong, "Soft quasi-maximum-likelihood detection for multiple-antenna wireless channels," *IEEE Trans. Sig. Proc.*, vol. 51, no. 11, pp. 2710–2719, Nov. 2003.

[31] J. Jaldén, C. Martin, and B. Ottersten, "Semidefinite programming for detection in linear systems-optimality conditions and space-time decoding," in *Proc. IEEE Inernational Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003, vol. 4.

[32] A. Wiesel, Y. C. Eldar, and S. Shamai, "Semidefinite relaxation for detection of 16-QAM signaling in MIMO channels," *IEEE Sig. Proc. Letters*, vol. 12, no. 9, pp. 653–656, Sep. 2005.

[33] N. Sidiropoulos and Z.-Q. Luo, "A semidefinite relaxation approach to MIMO detection for high-order QAM constellations," *IEEE Sig. Proc. Letters*, vol. 13, no. 9, pp. 525–528, Sep. 2006.

[34] Y. Yang, C. Zhao, P. Zhou, and W. Xu, "MIMO detection of 16-QAM signaling based on semidefinite relaxation," *IEEE Signal Processing Letters*, vol. 11, no. 14, pp. 797–800, 2007.

[35] W.-K. Ma, C.-C. Su, J. Jaldén, T.-H. Chang, and C.-Y. Chi, "The equivalence of semidefinite relaxation MIMO detectors for higher-order qam," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 6, pp. 1038–1052, 2009.

[36] H.-T. Wai, W.-K. Ma, and A. M.-C. So, "Cheap semidefinite relaxation MIMO detection using row-by-row block coordinate descent," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 3256–3259.

[37] H. Vikalo, B. Hassibi, and P. Stoica, "Efficient joint maximum-likelihood channel estimation and signal detection," *IEEE Transactions on Wireless Communications*, vol. 5, no. 7, pp. 1838–1845, 2006.

[38] P. Stoica and G. Ganesan, "Space–time block codes: Trained, blind, and semi-blind detection," *Elsevier Digital Signal Processing*, vol. 13, no. 1, pp. 93–105, Jan. 2003.

[39] P. Stoica, H. Vikalo, and B. Hassibi, "Joint maximum-likelihood channel estimation and signal detection for SIMO channels," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003, vol. 4, pp. IV–13.

[40] H. A. J. Alshamary, M. F. Anjum, T. Al-Naffouri, A. Zaib, and W. Xu, "Optimal non-coherent data detection for massive SIMO wireless systems with general constellations: A polynomial complexity solution," *arXiv preprint:1507.02319*, 2015.

[41] W. Xu, M. Stojnic, and B. Hassibi, "On exact maximum-likelihood detection for non-coherent MIMO wireless systems: a branch-estimate-bound optimization framework," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2008, pp. 2017–2021.

[42] C. Studer, M. Wenk, and A. Burg, "VLSI implementation of hard-and soft-output sphere decoding for wide-band MIMO systems," in *VLSI-SoC: Forward-Looking Trends in IC and Systems Design*, pp. 128–154. Springer, 2010.

[43] D. Gesbert, M. Shafi, D.-s. Shiu, P. J. Smith, and A. Naguib, "From theory to practice: an overview of mimo space-time coded wireless systems," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 3, pp. 281–302, 2003.

[44] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*, Cambridge University Press, New York, USA, 2008.

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

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

[47] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: Algorithms and VLSI implementation," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 2, pp. 290–300, Feb. 2008.

[48] J. Jaldèn and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.

[49] D. Seethaler, J. Jaldén, C. Studer, and H. Bölcskei, "On the complexity distribution of sphere decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 5754–5768, Sept. 2011.

[50] F. R. Bach and M. I. Jordan, "Predictive low-rank decomposition for kernel methods," in *Proc. 22nd International Conference on Machine Learning (ICML)*, Aug. 2005, pp. 33–40.

[51] H. Harbrecht, M. Peters, and R. Schneider, "On the low-rank approximation by the pivoted Cholesky decomposition," *Elsevier Applied Numerical Mathematics*, vol. 62, no. 4, pp. 428–440, Apr. 2012.

[52] M. Benzi, "Preconditioning techniques for large linear systems: a survey," *Elsevier Journal of Computational Physics*, vol. 182, no. 2, pp. 418–477, Nov. 2002.

[53] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods," *Mathematical Programming*, vol. 137, no. 1-2, pp. 91–129, Feb. 2013.

[54] S. Burer and R. D. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical Programming*, vol. 95, no. 2, pp. 329–357, 2003.

[55] N. Boumal, "A Riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints," *arXiv preprint: 1506.00575*, June 2015.

[56] P. Meinerzhagen, C. Roth, and A. Burg, "Towards generic low-power area-efficient standard cell based memory architectures," in *53rd IEEE Intern. Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug. 2010, pp. 129–132.

[57] M. Wenk, M. Zellweger, A. Burg, N. Felber, and W. Fichtner, "K-best MIMO detection VLSI architectures achieving up to 424 Mbps," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2006.

[58] B. Razavi, *Design of Analog CMOS Integrated Circuits*, New York: McGraw-Hill, 2002.

[59] C. H. Liao, T. P. Wang, and T. D. Chiueh, "A 74.8 mW soft-output detector IC for 8×8 spatial-multiplexing MIMO communications," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 2, pp. 411–421, Feb. 2010.

[60] C. H. Yang, T. H. Yu, and D. Marković, "A 5.8 mW 3GPP-LTE compliant 8×8 MIMO sphere decoder chip with soft-outputs," in *2010 Symposium on VLSI Circuits*, June 2010, pp. 209–210.

[61] E. M. Witte, F. Borlenghi, G. Ascheid, R. Leupers, and H. Meyr, "A scalable VLSI architecture for soft-input soft-output single tree-search sphere decoding," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 9, pp. 706–710, Sep. 2010.

[62] C.-F. Liao, J.-Y. Wang, and Y.-H. Huang, "A 3.1 Gb/s 8×8 sorting reduced K-Best detector with lattice reduction and QR decomposition," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 12, pp. 2675–2688, Feb. 2014.

[63] C. Senning, L. Bruderer, J. Hunziker, and A. Burg, "A lattice reduction-aided MIMO channel equalizer in 90 nm CMOS achieving 720 Mb/s," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 6, pp. 1860–1871, June 2014.