

# FPGA Design of Low-Complexity Joint Channel Estimation and Data Detection for Large SIMO Wireless Systems

Oscar Castañeda<sup>1</sup>, Tom Goldstein<sup>2</sup>, and Christoph Studer<sup>1</sup>

<sup>1</sup>School of ECE, Cornell University, Ithaca, NY; e-mail: oc66@cornell.edu, studer@cornell.edu

<sup>2</sup>Department of CS, University of Maryland, College Park, MD; e-mail: tomg@cs.umd.edu

**Abstract**—Joint channel estimation and data detection (JED) enables near-optimal error-rate performance in realistic wireless communication systems that suffer from channel estimation errors. In this paper, we propose a new JED algorithm and a corresponding FPGA design for large single-input multiple-output (SIMO) wireless systems that use constant-modulus constellations. Our algorithm, referred to as PrOX (short for PRojection Onto conveX hull), relies on biconvex relaxation (BCR) in order to efficiently compute an approximate solution of the maximum-likelihood JED problem that exhibits prohibitive complexity. PrOX is a simple and hardware-friendly algorithm that achieves near-optimal error-rate performance for a wide-range of system configurations. To demonstrate the efficacy of PrOX, we develop a scalable VLSI architecture and present reference implementation results on a Xilinx Virtex-7 FPGA. Compared to a recently-reported reference JED design, PrOX achieves 3× higher throughput, 20× better hardware-efficiency (in terms of throughput per look-up tables), and 8× improved energy-efficiency.

## I. INTRODUCTION

Large wireless systems, in which the base-station (BS) is equipped with hundreds or thousands of antenna elements, have recently gained significant attention. The large number of BS antennas enables fine-grained beamforming and improved spectral-efficiency within each cell compared to traditional, small scale single-input multiple-output (SIMO) and multiple-input multiple-output (MIMO) wireless systems [1], [2]. *Pilot contamination* is known to be a limiting factor in such large wireless systems, i.e., channel training is contaminated by pilot or data transmission of users communicating in adjacent cells [3]. To mitigate the effects of pilot contamination, one can rely on joint channel estimation and data detection (JED), which is capable of approaching the performance of idealistic systems with perfect channel-state information [4], [5]. While a few JED algorithms have been proposed for traditional, small-scale SIMO and MIMO wireless systems [4]–[7], their complexity is typically very high and not much is known about their efficacy for systems with hundreds or thousands of antennas. Moreover, with the exception of the VLSI design in [8], no hardware designs for JED have been described in the literature.

### A. Contributions

We develop a novel JED algorithm and a VLSI design for large SIMO wireless systems that use constant-modulus constellations. Our algorithm, referred to as PrOX (short for PRojection Onto conveX hull), relies on the biconvex relaxation (BCR) framework [9], a general method for computing approximate solutions to semidefinite problems at low complexity. PrOX solves the biconvex problem that arises after relaxing and regularizing the optimal maximum-likelihood (ML) JED problem, enabling near ML-JED performance at low complexity. We develop a scalable VLSI architecture for PrOX that achieves high-throughput and low implementation complexity. We show reference

OC and CS were supported in part by Xilinx Inc. and by the US NSF under grants ECCS-1408006, CCF-1535897, and CAREER CCF-1652065. TG was supported by the US NSF under grant CCF-1535902 and by the US ONR under grant N00014-15-1-2676.

implementation results on a Xilinx Virtex-7 FPGA and compare our solution to the recently-reported JED design in [8].

### B. Notation

Lowercase and uppercase boldface letters stand for column vectors and matrices, respectively. For the matrix  $\mathbf{A}$ , the Hermitian is  $\mathbf{A}^H$  and the  $k$ th row and  $\ell$ th column entry is  $A_{k,\ell}$ . For the vector  $\mathbf{a}$ , the  $k$ th entry is  $a_k$ . The Euclidean norm of  $\mathbf{a}$  and the Frobenius norm of  $\mathbf{A}$  are  $\|\mathbf{a}\|_2$  and  $\|\mathbf{A}\|_F$ , respectively. The real and imaginary parts of the vector  $\mathbf{a}$  are  $\Re(\mathbf{a})$  and  $\Im(\mathbf{a})$ , respectively.

## II. SYSTEM MODEL AND ML-OPTIMAL JED

### A. Large SIMO System Model

We consider a large SIMO wireless uplink system in which a single-antenna user transmits data over  $K + 1$  time slots to  $B$  BS antennas. We model the input-output relation of a narrow-band, flat-fading wireless channel as follows [4]–[6]:  $\mathbf{Y} = \mathbf{h}\mathbf{s}^H + \mathbf{N}$ . Here, the matrix  $\mathbf{Y} \in \mathbb{C}^{B \times (K+1)}$  contains the  $B$ -dimensional receive vectors for the  $K + 1$  time slots,  $\mathbf{h} \in \mathbb{C}^B$  is the unknown SIMO channel vector (assumed block fading over  $K + 1$  time slots),  $\mathbf{s} \in \mathcal{O}^{K+1}$  contains the transmitted data symbols for all  $K + 1$  time slots, and  $\mathbf{N} \in \mathbb{C}^{B \times (K+1)}$  is i.i.d. complex circularly-symmetric Gaussian noise with variance  $N_0$  per entry. We assume constant-modulus constellations for the set  $\mathcal{O}$ .

### B. Joint Channel Estimation and Data Detection (JED)

Let  $\mathbf{h}$  be a deterministic, but unknown, channel vector with unknown statistics. Then, we have the following ML-JED problem [6]:

$$\{\hat{\mathbf{s}}^{\text{JED}}, \hat{\mathbf{h}}\} = \arg \min_{\mathbf{s} \in \mathcal{O}^{K+1}, \mathbf{h} \in \mathbb{C}^B} \|\mathbf{Y} - \mathbf{h}\mathbf{s}^H\|_F. \quad (1)$$

Note that, if  $\hat{\mathbf{s}}^{\text{JED}} e^{j\phi} \in \mathcal{O}^{K+1}$  for  $\phi \in [0, 2\pi)$ , then  $\hat{\mathbf{h}} e^{j\phi}$  is also a solution to (1). To avoid this ambiguity, we set the first entry of the transmit vector to  $\tilde{s} \in \mathcal{O}$  and exploit this knowledge at the receiver.<sup>1</sup>

Since we assumed the entries in  $\mathbf{s}$  to be constant-modulus, the ML-JED estimate for the transmit vector can be written as [6]:

$$\hat{\mathbf{s}}^{\text{JED}} = \arg \max_{\mathbf{s} \in \mathcal{O}^{K+1}} \|\mathbf{Y}\mathbf{s}\|_2. \quad (2)$$

This problem resembles the famous MaxCut problem that is known to be NP-hard [10]. For a small number of time slots  $K + 1$ , the problem in (2) can be solved exactly at low average complexity using sphere decoding (SD) methods [6]. For a large number of time slots, however, SD entails prohibitive complexity. Moreover, linear methods (which are widely used for coherent data detection in MIMO systems) are not useful for this case, since the entries of  $\mathbf{s}$  grow unbounded when relaxing the constraint  $\mathbf{s} \in \mathcal{O}^{K+1}$  to  $\mathbf{s} \in \mathbb{C}^{K+1}$ . To enable JED

<sup>1</sup>This approach differs from traditional, pilot-based schemes as all transmitted symbols are used to improve the channel estimate, which significantly improves the error-rate performance (see Section V for simulation results).

for SIMO systems with a large number of time slots, the design of novel, low-complexity algorithms is necessary. Recently, an algorithm and VLSI design that relies on semidefinite relaxation (SDR) was reported in [8]. We next propose a simpler and faster method to compute approximate solutions to the ML-JED problem in (2).

### III. PROX: PROJECTION ONTO CONVEX HULL

#### A. Biconvex Relaxation (BCR) of the ML-JED Problem

We start by reformulating the ML-JED problem in (2) using BCR [9]. We first include a regularization term that forces the transmit vector  $\mathbf{s} \in \mathcal{O}^{K+1}$  to be close to a copy  $\mathbf{q} \in \mathbb{C}^{K+1}$  as follows:

$$\arg \min_{\mathbf{s} \in \mathcal{O}^{K+1}, \mathbf{q} \in \mathbb{C}^{K+1}} -\|\mathbf{Y}\mathbf{q}\|_2^2 + \alpha\|\mathbf{q} - \mathbf{s}\|_2^2, \quad (3)$$

where  $\alpha$  is a suitably chosen regularization parameter. To ensure that the problem in (3) is convex in  $\mathbf{q}$ ,  $\alpha$  must be larger than the maximum singular value of the Gram matrix  $\mathbf{G} = \mathbf{Y}^H \mathbf{Y}$ . We next relax the finite-alphabet constraint to the convex hull  $\mathcal{C}_{\mathcal{O}}$  of the set  $\mathcal{O}$  [11]

$$\mathcal{C}_{\mathcal{O}} = \left\{ \sum_{i=1}^{|\mathcal{O}|} \alpha_i s_i \mid (\alpha_i \in \mathbb{R}^+, \forall i) \wedge \sum_{i=1}^{|\mathcal{O}|} \alpha_i = 1 \right\},$$

with  $s_i, i = 1, \dots, |\mathcal{O}|$ , being the constellation points in  $\mathcal{O}$ . The convex hull of the BPSK constellation set  $\mathcal{C}_{\text{BPSK}}$  is the line along the real axis in  $[-1, +1]$ ; the convex hull  $\mathcal{C}_{\text{QPSK}}$  for QPSK is the square with the four constellation points as corners. By relaxing  $\mathcal{O}$  to  $\mathcal{C}_{\mathcal{O}}$ , we arrive at the following relaxed ML-JED problem:

$$\arg \min_{\mathbf{s} \in \mathcal{C}_{\mathcal{O}}^{K+1}, \mathbf{q} \in \mathbb{C}^{K+1}} -\|\mathbf{Y}\mathbf{q}\|_2^2 + \alpha\|\mathbf{q} - \mathbf{s}\|_2^2.$$

Finally, we include a norm constraint that promotes large values in  $\mathbf{s}$ , with the goal of forcing the entries of  $\mathbf{s}$  to the boundary of the convex hull. This leads to the final BCR formulation:

$$\hat{\mathbf{s}}^{\text{BCR}} = \arg \min_{\mathbf{s} \in \mathcal{C}_{\mathcal{O}}^{K+1}, \mathbf{q} \in \mathbb{C}^{K+1}} -\|\mathbf{Y}\mathbf{q}\|_2^2 + \alpha\|\mathbf{q} - \mathbf{s}\|_2^2 - \beta\|\mathbf{s}\|_2^2, \quad (4)$$

where  $\beta < \alpha$  is a suitably chosen algorithm parameter.

#### B. Alternating Optimization

We solve the BCR problem in (4) using alternating minimization, in which we keep one of the variables fixed while solving for the other. The resulting iterative algorithm is given by

$$\begin{aligned} \mathbf{q}^{(t)} &= \arg \min_{\mathbf{q} \in \mathbb{C}^{K+1}} -\|\mathbf{Y}\mathbf{q}\|_2^2 + \alpha\|\mathbf{q} - \mathbf{s}^{(t-1)}\|_2^2 \\ \mathbf{s}^{(t)} &= \arg \min_{\mathbf{s} \in \mathcal{C}_{\mathcal{O}}^{K+1}} \alpha\|\mathbf{q}^{(t)} - \mathbf{s}\|_2^2 - \beta\|\mathbf{s}\|_2^2, \end{aligned}$$

where  $t = 1, 2, \dots, t_{\max}$  is the iteration counter. We initialize the above iterative procedure with  $\mathbf{s}^{(0)} = \check{s}(G_{1,1})^{-1}\mathbf{g}_1$ , where  $\mathbf{g}_1$  is the first column of  $\mathbf{G}$ , and  $\check{s} \in \mathcal{O}$  is the fixed symbol transmitted in the first time slot and known at the receiver.

Since the problem in (4) is biconvex in  $\mathbf{s}$  and  $\mathbf{q}$ , both of the above steps are convex and can be solved optimally and in closed form. Concretely, we get the following simple algorithm:

$$\begin{aligned} \mathbf{q}^{(t)} &= (\mathbf{I} - \alpha^{-1}\mathbf{G})^{-1}\mathbf{s}^{(t-1)} \\ \mathbf{s}^{(t)} &= \text{prox}_{\mathcal{C}_{\mathcal{O}}^{K+1}}(\theta\mathbf{q}^{(t)}), \end{aligned} \quad (5)$$

where  $\theta = \frac{\alpha}{\alpha - \beta} > 0$ ,  $\mathbf{I}$  is the identity, and the proximal operator [12]

$$\text{prox}_{\mathcal{C}_{\mathcal{O}}^{K+1}}(\mathbf{v}) = \arg \min_{\mathbf{s} \in \mathcal{C}_{\mathcal{O}}^{K+1}} \|\mathbf{v} - \mathbf{s}\|_2^2 \quad (6)$$

is the element-wise orthogonal projection onto the convex hull  $\mathcal{C}_{\mathcal{O}}$ . For BPSK, the proximal operator in (6) is given by  $\text{prox}_{\mathcal{C}_{\text{BPSK}}}(v_b) =$

#### Algorithm 1 PROjection Onto convex hull (PrOX)

- 1: **inputs:**  $\hat{\mathbf{G}}, \mathbf{s}^{(0)}$ , and  $\varrho$
- 2: **for**  $t = 1, \dots, t_{\max}$  **do**
- 3:  $\tilde{\mathbf{q}}^{(t)} = \hat{\mathbf{G}}\mathbf{s}^{(t-1)}$
- 4:  $\mathbf{s}^{(t)} = \text{prox}_{\mathcal{C}_{\mathcal{O}}^{K+1}}(\varrho\tilde{\mathbf{q}}^{(t)})$
- 5:  $s_1^{(t)} = \check{s}$
- 6: **end for**
- 7: **outputs:**  $\hat{\mathbf{s}}_k = \arg \min_{s \in \mathcal{O}} |s_k^{(t_{\max})} - s|, k = 2, \dots, K + 1$

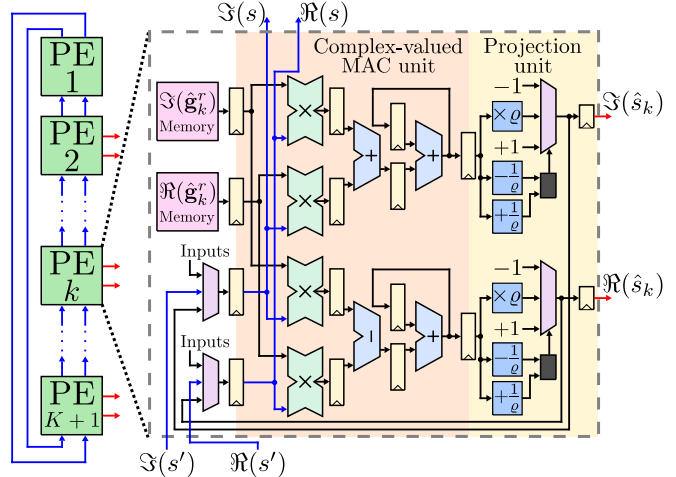


Fig. 1. Block diagram of PrOX. We use a linear array of processing elements (PEs) that achieves high throughput at low hardware complexity.

$\max\{\min\{\Re(v_b), +1\}, -1\}$ ; for QPSK, (6) corresponds to applying the same operation, independently, to  $\Re(v_b)$  and  $\Im(v_b)$ . In the case of (5),  $\mathbf{v} = \theta\mathbf{q}^{(t)}$ , so the proximal operator in (6) projects a scaled version of  $\mathbf{q}^{(t)}$  onto the convex hull of the constellation—this is why we call our algorithm “PROjection Onto convex hull” (PrOX).

#### C. Hardware Optimizations for PrOX

We now modify the algorithm in (5) to make it more hardware-friendly. Since the receiver knows the first entry of  $\mathbf{s}$ , there is no need to apply the algorithm on this particular entry—instead, we force  $s_1^{(t)}$  to be  $\check{s}$  at the end of each iteration. The matrix  $(\mathbf{I} - \alpha^{-1}\mathbf{G})^{-1}$  exhibits a large dynamic range for different system configurations and channel realizations. To facilitate fixed-point design, we divide all of its elements by a constant  $\gamma > 0$ , so that the entries of the resulting matrix are close to one in absolute value. This scaling procedure requires us to introduce a new vector  $\tilde{\mathbf{q}} = \gamma^{-1}\mathbf{q}$ , resulting in

$$\begin{aligned} \tilde{\mathbf{q}}^{(t)} &= \gamma^{-1}\mathbf{q}^{(t)} = \gamma^{-1}(\mathbf{I} - \alpha^{-1}\mathbf{G})^{-1}\mathbf{s}^{(t-1)} \\ \mathbf{s}^{(t)} &= \text{prox}_{\mathcal{C}_{\mathcal{O}}^{K+1}}(\theta\gamma\tilde{\mathbf{q}}^{(t)}) = \text{prox}_{\mathcal{C}_{\mathcal{O}}^{K+1}}(\varrho\tilde{\mathbf{q}}^{(t)}), \end{aligned} \quad (7)$$

where  $\varrho = \theta\gamma$ . By defining  $\hat{\mathbf{G}} = \gamma^{-1}(\mathbf{I} - \alpha^{-1}\mathbf{G})^{-1}$ , we arrive at the hardware-friendly version of PrOX summarized in Algorithm 1. We note that  $\alpha > 0$  and  $\varrho > 0$  are both algorithm parameters that can be tuned to empirically improve the performance of PrOX.

## IV. VLSI ARCHITECTURE

#### A. Architecture Overview

Figure 1 shows the proposed VLSI architecture for PrOX in a system with QPSK modulation. Our architecture is a linear array formed of  $N = K + 1$  processing elements (PEs), each associated with an entry of  $\mathbf{s}$  and  $\tilde{\mathbf{q}}$ . To execute Algorithm 1, each PE consists of three key components. The first component is the  $\hat{\mathbf{G}}$ -matrix memory: The  $k$ th PE contains memory for the real and imaginary parts of  $\hat{\mathbf{g}}_k^r$ ,

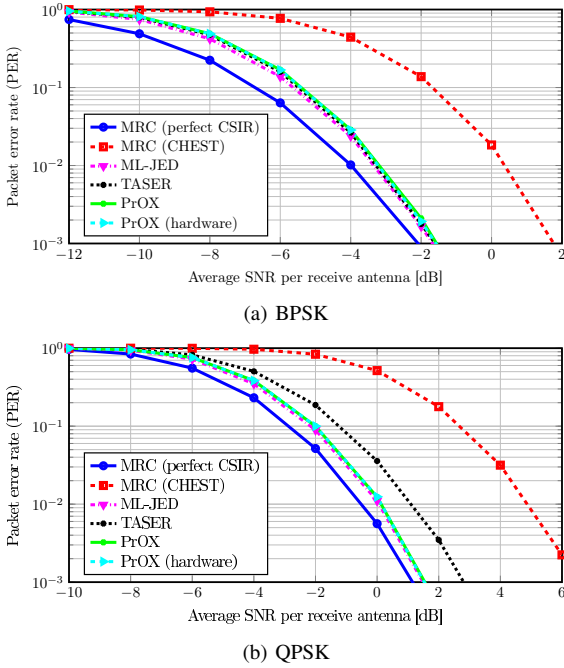


Fig. 2. Uncoded packet error rate (PER) for a SIMO system with  $B = 16$  BS antennas and transmission over  $K = 16$  time slots. PrOX achieves near-optimal PER performance (close-to perfect CSIR) and exhibits a performance similar to ML-JED. TASER [8] entails a 1 dB SNR loss for the QPSK case, while conventional MRC with CHEST suffers a 4 dB SNR loss.

the  $k$ th row of  $\hat{\mathbf{G}}$ . This memory is implemented using FPGA look-up tables (LUTs) as distributed RAM. We assume that the memory contents were computed during a preprocessing step. The second component is a complex-valued multiply-accumulate (MAC) unit, used to compute the matrix-vector product in line 3 of Algorithm 1. The third component is the projection unit, which executes line 4 of Algorithm 1. The hard-output estimates (see line 7 of Algorithm 1) are extracted from the sign bits of the outputs of the projection unit. As  $s_1^{(t)} = \tilde{s}$  is forced at the end of each iteration (line 5 of Algorithm 1), the first PE only contains two multiplexers and flip-flops that store  $\tilde{s}$ .

### B. Architecture Operation

We now detail the operating principle of the PrOX architecture shown in Figure 1. To prevent a high fan out, the entries of the  $\hat{\mathbf{G}}$  matrix are stored in a way that, for the  $k$ th PE, the first address of its memories contains  $\hat{G}_{k,k}$ ; the second address,  $\hat{G}_{k,k+1}$ , and so on. In the first clock cycle, the  $k$ th PE has access to  $s_k^{(t-1)}$ , so it can compute the product  $\hat{G}_{k,k} s_k^{(t-1)}$  and store the result in its accumulator. At the same time, this PE sends  $s_k^{(t-1)}$  to the  $(k-1)$ th PE. In the second clock cycle, the  $k$ th PE receives  $s_{k+1}^{(t-1)}$  from the  $(k+1)$ th PE, so it can compute  $\hat{G}_{k,k+1} s_{k+1}^{(t-1)}$  and add it to the previous result. This clock cycle is also used by this PE to transmit  $s_{k+1}^{(t-1)}$  to the  $(k-1)$ th PE. In the third clock cycle, the  $k$ th PE receives  $s_{k+2}^{(t-1)}$  from the  $(k+1)$ th PE (which received this value from the  $(k+2)$ th PE during the second cycle), so the  $k$ th PE can continue executing its complex-valued MAC operation. This process is replicated in all PEs, enabling each entry of  $\mathbf{s}^{(t-1)}$  to circulate through the linear array. After  $K+1$  cycles, all PEs have had access to all entries of  $\mathbf{s}^{(t-1)}$  and used them to compute their respective entry of  $\tilde{\mathbf{q}}^{(t)}$ . During this process, the first PE (which differs from the other PEs) executes the same procedure: it first sends  $\tilde{s}$  to the  $(K+1)$ th PE, and then forwards the  $\mathbf{s}^{(t-1)}$  entry received from the second PE to the  $(K+1)$ th PE.

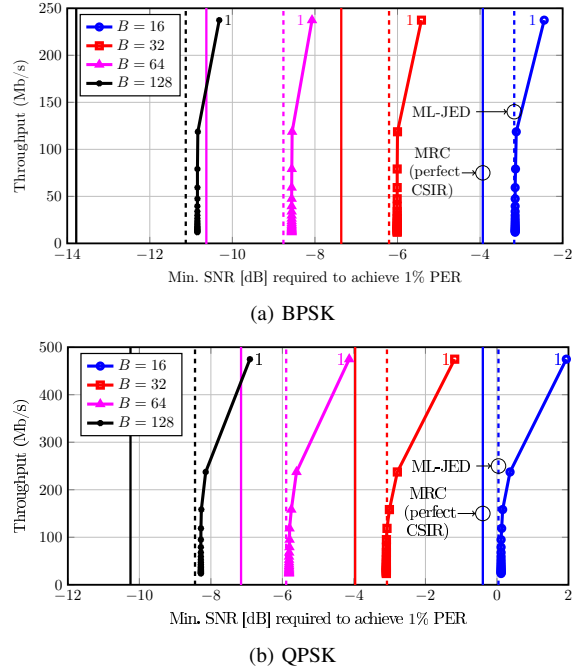


Fig. 3. Throughput vs. performance trade-off for  $K = 16$  time slots. Vertical solid lines represent MRC with CSIR; dashed lines represent ML-JED. The numbers next to the markers correspond to the number of PrOX iterations  $t_{\max}$ .

As the MAC units contain three pipeline stages, two clock cycles are required to flush the pipeline. Hence, the matrix-vector multiplication on line 3 of Algorithm 1 is computed in  $K+3$  clock cycles.

We now describe the operation of the projection unit. For QPSK constellations, the unit consists of two identical modules (cf. Figure 1). One of these modules takes  $\bar{q} = \Re(\tilde{q}^{(t)})$  as its input, while the other takes  $\bar{q} = \Im(\tilde{q}^{(t)})$ . Each module checks whether its input  $\bar{q}$ , after being multiplied with  $\varrho$ , is below  $-1$ , above  $+1$  or in-between these numbers, and outputs  $-1$ ,  $+1$ , or  $\varrho\bar{q}$ , respectively. To do so, each module takes its input and adds, separately,  $-1/\varrho$  and  $+1/\varrho$  to it. The two results are used to select the output value. The quantity  $\varrho\bar{q}$  is computed in parallel. By restricting  $\varrho$  to be a power of two, the multiplication can be carried out with arithmetic shifts. The projection unit uses one clock cycle to complete its operation, but it can start only after the  $K+3$  cycles used by the complex MAC unit. After the projection unit has finished, the new  $s_k^{(t)}$  will be available at the inputs of the complex-valued MAC unit, ready to start a new iteration. Consequently, each PrOX iteration requires  $K+4$  clock cycles.

### C. Fixed-Point Parameters

We use 6 bit fixed-point values for representing the  $\mathbf{s}^{(t)}$  entries, with 3 fraction bits. For the elements of  $\hat{\mathbf{G}}$ , 12 bit fixed-point values are used, with 11 fraction bits. The intermediate values and outputs of the complex-valued MAC unit use 15 bits with 11 fraction bits.

## V. IMPLEMENTATION RESULTS AND COMPARISON

### A. Error-Rate Performance

Figures 2(a) and 2(b) show packet error rate (PER) simulation results for PrOX as in (5) for  $t_{\max} = 5$  iterations. The results are obtained from 50,000 Monte-Carlo trials in a  $B = 16$  BS antenna SIMO system with  $K = 16$  time slots, using an i.i.d. flat Rayleigh block-fading channel model. We show the performance of maximum ratio combining (MRC) detection with both perfect receive-side channel state information (CSIR) and channel estimation (CHEST), ML-JED

TABLE I  
IMPLEMENTATION RESULTS ON A XILINX VIRTEX-7  
XC7VX690T FPGA FOR DIFFERENT PROX ARRAY SIZES

Array size ( $N = K + 1$ )	$N = 5$	$N = 9$	$N = 17$	$N = 33$
Time slots ( $K = N - 1$ )	$K = 4$	$K = 8$	$K = 16$	$K = 32$
Slices	327	658	1 491	3 018
LUTs	990	1 991	4 818	9 861
FFs	515	989	1 953	3 857
DSP48s	16	32	64	128
Max. clock freq. [MHz]	358	341	297	240
Min. latency [cycles]	8	12	20	36
Max. throughput <sup>a</sup> [Mb/s]	358	454	475	426
Power estimate <sup>b</sup> [W]	0.45	0.47	0.79	1.14

<sup>a</sup>Assuming QPSK modulation; for BPSK, the throughput is halved.

<sup>b</sup>Statistical power estimation at max. clock freq. and 1.0 V supply voltage.

detection using the method in [7], and the recent TASER algorithm [8] with 20 iterations. Note that MRC with perfect CSIR is optimal for this scenario. However, the assumption of perfect CSIR cannot be realized in practice and one must resort to CHEST, which entails a performance loss of more than 4 dB at 1% PER. ProX significantly outperforms MRC with CHEST and approaches near-ML-JED performance but, in contrast to ML-JED detection, at very low complexity. We also show the performance of the VLSI design of ProX as in Algorithm 1; our fixed-point design exhibits virtually no implementation loss.

Figures 3(a) and 3(b) show the trade-off between the throughput of ProX and the minimum SNR required to achieve 1% PER, for SIMO systems with  $K = 16$  time slots and different numbers of BS antennas. As a reference, we include the performance of MRC with CSIR and that of an optimal ML-JED detector. By increasing the number of ProX iterations  $t_{\max}$ , the desired PER is achieved at a lower SNR at the expense of lower throughput. Note that for BPSK modulation, 2 iterations are typically enough to reach near-ML-JED performance, which leads to a throughput of 118 Mb/s. For QPSK, 3 iterations are sufficient and achieve 158 Mb/s.

### B. Implementation Results

To demonstrate the efficacy of ProX, we developed FPGA designs for various array sizes  $N \in \{5, 9, 17, 33\}$  that support JED for  $K \in \{4, 8, 16, 32\}$  time slots, respectively. Our FPGA implementation results are shown in Table I. As expected, the resource utilization scales linearly with the array size  $N$ . For the  $N \in \{5, 9, 17\}$  arrays, the critical path is in the PE's projection unit; for the  $N = 33$  array, the critical path is in the distribution of control signals.

Table II compares ProX with TASER [8], which has been implemented on the same FPGA for a SIMO system with  $B = 128$  BS antennas and communication through  $K = 8$  time slots. ProX requires significantly fewer resources and a lower power than TASER, while achieving a higher throughput. This makes this design superior to TASER in terms of both hardware-efficiency (measured in throughput per FPGA LUTs) and energy per bit: ProX is  $20\times$  more hardware-efficient and  $8\times$  more energy-efficient than TASER for the considered scenario. At the same time, ProX exhibits a similar (for BPSK) or better (for QPSK) PER performance than TASER, while using fewer iterations, as shown in Figures 2(a) and 2(b). We note that while ProX is only suitable for JED in large SIMO, TASER can also perform near-ML data detection in coherent massive MIMO systems [8].

## VI. CONCLUSIONS

We have proposed ProX, a novel joint channel estimation and data detection (JED) algorithm for large SIMO systems with constant-

TABLE II  
COMPARISON OF JED DATA DETECTORS FOR A QPSK,  $B = 128$ ,  $K = 8$   
LARGE-SIMO SYSTEM ON A XILINX VIRTEX-7 XC7VX690T FPGA

Detection algorithm	PrOX	TASER [8]
Error-rate performance	Near ML-JED	Near ML-JED
Iterations $t_{\max}$	3	3
Slices	658 (0.61 %)	4 350 (4.02 %)
LUTs	1 991 (0.46 %)	13 779 (3.18 %)
FFs	989 (0.11 %)	6 857 (0.79 %)
DSP48s	32 (0.89 %)	168 (4.67 %)
Clock frequency [MHz]	341	225
Latency [clock cycles]	12	72
Throughput [Mb/s]	151	50
Power estimate <sup>a</sup> [W]	0.47	1.30
Throughput/LUTs	75 841	3 629
Energy/bit [nJ/b]	3.09	26.0

<sup>a</sup>Statistical power estimation at max. clock freq. and 1.0 V supply voltage.

modulus constellations, as well as a VLSI architecture. The low complexity of ProX translates into a low-area, high-throughput, and low-power FPGA design. Our FPGA implementation for a SIMO system with  $K = 8$  time slots achieves a maximum throughput of 454 Mb/s, while using only 2 k FPGA LUTs. In addition, ProX significantly outperforms TASER [8], the only other existing JED design, in all performance metrics.

We conclude by noting that ProX is a first step towards hardware accelerators that are able to find approximate solutions to the MaxCut problem at low complexity and in a hardware-efficient manner. The design of accelerators for MIMO JED is part of ongoing research.

## REFERENCES

- [1] 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.
- [2] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, Apr. 2013.
- [3] 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.
- [4] P. Stoica and G. Ganesan, "Space-time block codes: Trained, blind, and semi-blind detection," *Elsevier Dig. Signal Process.*, vol. 13, no. 1, pp. 93–105, Jan. 2003.
- [5] H. Vikalo, B. Hassibi, and P. Stoica, "Efficient joint maximum-likelihood channel estimation and signal detection," *IEEE Trans. Wireless Commun.*, vol. 5, no. 7, pp. 1838–1845, Jul. 2006.
- [6] 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*, Jul. 2015.
- [7] 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 Int. Symp. Inf. Theory (ISIT)*, Jul. 2008, pp. 2017–2021.
- [8] O. Castañeda, T. Goldstein, and C. Studer, "Data detection in large multi-antenna wireless systems via approximate semidefinite relaxation," *IEEE Trans. Circuits Syst. I*, no. 99, pp. 1–13, Nov. 2016.
- [9] S. Shah, A. Kumar, C. Castillo, D. Jacobs, C. Studer, and T. Goldstein, "Biconvex relaxation for semidefinite programming in computer vision," *European Conf. Comput. Vision (ECCV)*, pp. 717–735, Sep. 2016.
- [10] D. Steinberg, "Computation of matrix norms with applications to robust optimization," Master's thesis, Technion, Israel, 2005.
- [11] M. Wu, C. Dick, J. R. Cavallaro, and C. Studer, "High-throughput data detection for massive MU-MIMO-OFDM using coordinate descent," *IEEE Trans. Circuits Syst. I*, no. 99, Nov. 2016.
- [12] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends Optimization*, vol. 1, no. 3, pp. 123–231, Jan. 2013.