IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 21, 2018, accepted January 9, 2019, date of publication January 23, 2019, date of current version February 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2894181

FPGA Acceleration for Computationally Efficient Symbol-Level Precoding in Multi-User Multi-Antenna Communication Systems

JEVGENIJ KRIVOCHIZA, (Graduate Student Member, IEEE), JUAN MERLANO DUNCAN, (Member, IEEE), STEFANO ANDRENACCI, (Member, IEEE), SYMEON CHATZINOTAS, (Senior Member, IEEE), AND BJÖRN OTTERSTEN, (Fellow, IEEE)
SnT Department, University of Luxembourg, L-1855 Luxembourg City, Luxembourg

Corresponding author: Jevgenij Krivochiza (jevgenij.krivochiza@uni.lu)

This work was supported in part by the Luxembourg National Research Fund under Grant FNR11481283 "End-to-End Signal Processing Algorithms for Precoded Satellite Communications," Grant FNR11306457 "Energy And Complexity Efficient Millimiter-Wave Large-Array Communications," Grant FNR11504382 "On-board processing techniques for high throughput satellites," and in part by the European Research Council Advanced Grant 742648 "Actively Enhanced Cognition based Framework for Design of Complex Systems."

ABSTRACT In this paper, we demonstrate an FPGA-accelerated design of the computationally efficient symbol-level precoding (SLP) for high-throughput communication systems. The SLP technique recalculates the optimal beam-forming vectors by solving a non-negative least squares problem per every set of transmitted symbols. It exploits the advantages of constructive inter-user interference to minimize the total transmitted power and increase service availability. The benefits of using SLP come with a substantially increased computational load at the gateway. The FPGA design enables the SLP technique to perform in real-time operation mode and provide a high symbol throughput for the multiple receive terminals. We define the SLP technique in a closed-form algorithmic expression and translate it to hardware description language (HDL) and build an optimized HDL core for an FPGA. We evaluate the FPGA resource occupation, which is required for the high throughput multiple-input-multiple-output (MIMO) systems with sizeable dimensions. We describe the algorithmic code, the I/O ports mapping, and the functional behavior of the HDL core. We deploy the IP core to an actual FPGA unit and benchmark the energy efficiency performance of the SLP. The synthetic tests demonstrate a fair energy efficiency improvement of the proposed closed-form algorithm compared to the best results obtained through the MATLAB numerical simulations.

INDEX TERMS Convex programming, field programmable gate arrays, hardware resources, multicast communication, MIMO, optimization, precoding, power minimization, interference, wireless channels.

I. INTRODUCTION

Precoding is an important technological enabler to fully exploit the full frequency reuse in the next iterations of the modern wireless terrestrial [1] and multi-user multi-beam satellite communications [2]. MIMO precoded communications are also promissory to be applied in other multi-channel interference scenarios such as in the case of Very-high-bit-rate digital subscriber line (VDSL) [3] and Powerline communications [4].

The conventional channel-based precoding techniques use the knowledge of the Channel State Information (CSI) in order to generate the transmitted precoded signals. The most common channel-based strategies are Zero Forcing (ZF) and the Minimal Means Square Error (MMSE) precoding methods [5]. These methods were further studied and extended in the recent works [6], [7]. Advanced approaches, such as SLP, use in addition to the CSI the knowledge of the transmitted data symbols to each user to achieve more power efficient signaling and service availability.

Precoding techniques are deployed at the gateway side and introduce additional computational complexity on top of the existing signal preprocessing algorithms. The increased computational complexity involves defining and solving complex optimization problems at the system's symbol rate.

Solving optimization algorithms [8], [9] for large-scale problems is not a trivial task in real-time operations and is a barrier to the implementation of SLP techniques. Numerous studies have been conducted to address implementation and demonstration of computational-complexity aware precoding techniques. In [10] the closed-loop ZF precoding communications are demonstrated over-the-air satellite link showing the practical application. In [11] it is demonstrated that SLP design can be approached as ZF precoding with transmitted symbols perturbations. Li and Masouros [12] devised a novel closed-form solution to exploit constructive interference in precoding by using a similar approach. Haqiqatnejad *et al.* [13] proposed another closed-form sub-optimal solution for power minimizing SLP. At the same time, the SLP technique for large-scale antenna arrays is shown in [14].

In [15] we described the computationally efficient SLP technique, which optimizes the sum power of the precoded signal per each set of the transmitted symbols. The theoretical and experimental validations conducted in [16] showed that the SLP technique improves the received signal-to-noise ratio (SNR), service availability and energy efficiency of the transmitter. In [17] we demonstrated a 2×2 MIMO precoded real-time transmission system by making use of lookup tables (LUTs) for storing SLP optimized symbol mapping. While the use of the LUTs is an efficient solution for small systems with few transmitters, the large sized LUTs are needed for the large numbers of transmitters and receivers in the system. The size of the LUTs increases as a function of M^N for M-th modulation order and N number of receiver terminals. In [18] a real-time satellite precoded transmission hardware demonstrator is presented, where a gateway has 6 transmitting antennas and simultaneously serves 6 receiver terminals with up to 32-APSK modulated signals. In this case, the required size of LUTs would be more than $6^{32} \approx 7.95 * 10^{24}$ elements. It is inefficient to implement and handle LUTs at such a scale at a gateway.

In [19] we developed a novel closed-form solution of a NNLS problem for the computationally efficient SLP. The closed-form sub-optimal solution showed a very promising trade-off of the SLP technique performance and processing time when benchmarked against the conventional convex optimization Fast NNLS algorithm [20]. Its computational complexity is in the same order as one of ZF and does not require additional linear algebra operations. Numerical tests revealed a comparable processing time per set of symbols in both ZF and the SLP techniques.

In this work we expand the computationally efficient SLP design to operate in the real-time regime. We develop a complete FPGA-accelerated closed-form algorithm of the SLP technique and optimize it for an actual model of an FPGA silicon chip. For this, we use Vivado High-Level Synthesis (HLS) to translate the algorithm into HDL core and integrate the design into an FPGA. We estimate the resource utilization and cycle period. We deploy the HDL core on an actual FPGA board and benchmark its performance in terms of energy efficiency and compare the results with numerical estimations. We draw our conclusion based on the benchmark results and show that the closed-form solution fairly improves energy efficiency of precoded communications and utilizes a reasonable amount of FPGA resources.

Notation: Upper-case and lower-case bold-faced letters are used to denote matrices and column vectors. The superscripts $(\cdot)^H$, $(\cdot)^\dagger$ and $(\cdot)^{-1}$ represents Hermitian matrix, matrix transpose and inverse operations. $\|\cdot\|_2$ is the Euclidean norm, $|\cdot|$ is an absolute value of a complex value. The real and imaginary parts of a complex value are defined as $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$. The imaginary unit is denoted as $\iota^2 = -1$. The operator $(\bullet)$ denotes element-wise vector multiplication.

II. COMPUTATIONALLY EFFICIENT SYMBOL-LEVEL PRECODING

In [19] we benchmarked the symbols throughput of the proposed SLP algorithm in MATLAB environment and achieved over 200 kSymbols per second by running the closed-form algorithm on a standard Intel Central processing unit (CPU). It was shown, that the throughput is only 2 times slower than the performance of the conventional ZF algorithm [5] running on the same CPU. For comparison, by solving the same optimization problem with Fast NNLS algorithm [21] we could reach only around 5 kSymbols per second. For multi-level constellations, a power minimization problem cannot be solved using the Fast NNLS and more complicated algorithms are required. In [15] we benchmarked the symbol throughput of around 10 Symbols per second in case of 16-APSK modulation while running the optimization code in the similar environment. On the other hand, the closed-form SLP technique can be universally applied for single-level and multi-level modulations. It is a good candidate for a realistic real-time hardware implementation in a condition of limited FPGA resources as the same code can be used for multiple modulation types. The same algorithm is optimized for single- and multi-level modulations and thus no additional algorithm must be developed. We devise an FPGA-accelerated design of the precoding technique. Towards a better comprehension of the FPGA code design, in this section, we cover the main implementation aspects of SLP and the approximate closed-form solution.

A. SYSTEM MODEL

We consider a system model, which focuses on the forward link of a multi-user multi-antenna wireless communication system. We assume the full frequency reuse scenario, in which all the antennas transmit in the same frequency and time. The multi-user interference is mitigated using precoding. We define the number of transmitting antenna as N_t and the total number of receiver terminals as N_u in the coverage area. In the specified MIMO channel model, the received signal at the i-th terminal is given by $y_i = \mathbf{h}_i^\dagger \mathbf{x} + n_i$, where $\mathbf{h}_i^\dagger$ is a $1 \times N_t$ vector representing the complex channel coefficients between the i-th terminal and the N_t antennas of the transmitter, $\mathbf{x}$ is defined as the $N_t \times 1$ vector of the

transmitted symbols at a certain symbol period and n_i is the independent complex circular symmetric (c.c.s.) independent identically distributed (i.i.d) zero mean Additive White Gaussian Noise (AWGN) measured at the i-th terminal's receive antenna.

Looking at the concatenated formulation of the received signal, which includes the whole set of receiver terminals, the linear signal model is

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

where $\mathbf{y} = [y_1, y_2, \ldots, y_i] \in \mathbb{C}^{N_u \times 1}$, $\mathbf{n} = [n_1, n_2, \ldots, n_i] \in \mathbb{C}^{N_u \times 1}$, $\mathbf{x} \in \mathbb{C}^{N_t \times 1}$, and $\mathbf{s} \in \mathbb{C}^{N_u \times 1}$ and $\mathbf{H} = [\mathbf{h}_i^\dagger, \mathbf{h}_i^\dagger, \ldots, \mathbf{h}_i^\dagger] \in \mathbb{C}^{N_u \times N_t}$. In this scenario, we define a precoding matrix $\mathbf{W} \in \mathbb{C}^{N_t \times N_u}$ which maps the information symbols $\mathbf{s}$ into precoded symbols $\mathbf{x}$. We consider the data symbols $\mathbf{s}$ to be unit variance complex vectors $|s_i| = 1$ for every $i = 1, 2, \ldots, N_u$.

B. OPTIMIZATION PROBLEM DEFINITION

In this section, we define an optimization problem of the computationally efficient SLP technique, which aims to minimize the sum power of the precoded symbols at the gateway side. The technique is applicable on the M-th order phase-shift keying (M-PSK) modulations. It reduces the sum power of the precoded symbols by optimally increasing the amplitudes of the initial data symbols to exploit the constructive interference between at the receiver side. The method optimally preserves constructive interference components to decrease the total transmitted power at the transmitter side. The essential difference of the SLP technique from a linear precoding method is the optimization vector $\mathbf{u} = [u_1, u_2, \ldots, u_i] \in \mathbb{C}^{N_u \times 1}$, which is recalculated for every set of symbols $\mathbf{s}$ to construct the optimized precoded signal given by

$$\mathbf{x} = \mathbf{W}(\Gamma \bullet \mathbf{s} + \mathbf{u}), \tag{2}$$

where $\Gamma = [\Gamma_1, \Gamma_2, \ldots, \Gamma_i] \in \mathbb{R}^{N_u \times 1}$ is per terminal SNR requirements. The following formulation allows to split the problem of constructing an optimal beamforming into two independent tasks: channel orthogonalization and optimal symbol mapping for energy efficiency. In this scenario, we define the precoding matrix ($\mathbf{W}$) as the Zero-Forcing linear precoder:

$$\mathbf{W}_{\mathrm{ZF}} = \hat{\mathbf{H}}^\dagger(\hat{\mathbf{H}}\hat{\mathbf{H}}^\dagger)^{-1}, \tag{3}$$

where $\hat{\mathbf{H}}$ is the channel matrix estimated from the channel state information (CSI). We choose ZF for its properties to orthogonalize the channel so that in the case of $\hat{\mathbf{H}} = \mathbf{H}$ the received symbols are a summation of the transmitted symbols, the optimization vector and Gaussian noise:

$$\mathbf{y} = \mathbf{H}\mathbf{W}_{\mathrm{ZF}}(\Gamma \bullet \mathbf{s} + \mathbf{u}) + \mathbf{n} = \Gamma \bullet \mathbf{s} + \mathbf{u} + \mathbf{n}. \tag{4}$$

ZF is not an optimal precoder in a sense of energy efficiency and there are more efficient techniques in the literature [22], which provide better power and BER performance on a frame basis. By using the ZF precoding matrix we guarantee to meet the SNR constraints in the design of SLP and simplify the precoder. The optimal symbol mapping we derive in the following paragraphs.

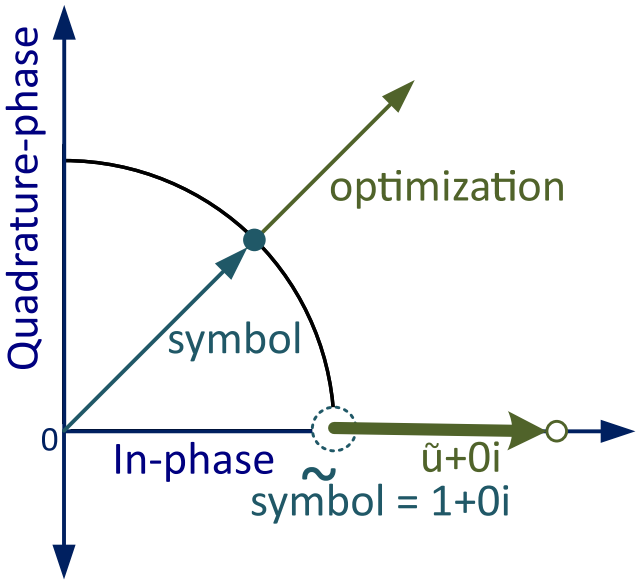

FIGURE 1. Symbol optimization of the proposed Symbol-Level Precoding.

In Fig. 1 we demonstrate an impact of the optimization u_k on a single complex data symbol s_i with unit power. The optimization vector increases the absolute magnitude of the symbol and keeps its phase in the fixed direction. In a case of multi-level constellation we consider, that power of the symbols, which are mapped to the external level, is $|s_i| \geq 1$. All the symbols on internal level with power $|s_i| < 1$ retain their original position.

To avoid operations with complex numbers in the optimization problem we reformulate the input data to real-defined values and keep the rest of the optimization problem relevant. We replace the complex data symbols with equivalent symbols $\tilde{\mathbf{s}} \in \mathbb{R}^{N_u}$, where $\tilde{s}_i = 1 + \imath 0$ for every $i = 1, 2, \ldots, N_u$, by introducing the following transformation

$$\Gamma \bullet \mathbf{s} = \mathbf{B}\tilde{\mathbf{s}}, \tag{5}$$

where $\mathbf{B}$ is a diagonal matrix, where elements of the vector $\Gamma \bullet \mathbf{s}$ are its diagonal elements such as:

$$\mathbf{B} = \begin{bmatrix} \Gamma_1 s_1 & 0 & 0 & \ldots & 0 \\ 0 & \Gamma_2 s_2 & 0 & \ldots & 0 \\ 0 & 0 & \Gamma_3 s_3 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & 0 & \Gamma_i s_i. \end{bmatrix} \tag{6}$$

We also replace the optimization vector $\mathbf{u}$ with a new vector $\tilde{\mathbf{u}} = [\tilde{u_1}, \tilde{u_2}, \ldots, \tilde{u_i}] \in \mathbb{R}_{\geq 0}^{N_u \times 1}$ and rewrite the equation (2) as

$$\mathbf{x} = \mathbf{W}_{\mathrm{ZF}}\mathbf{B}(\tilde{\mathbf{s}} + \tilde{\mathbf{u}}). \tag{7}$$

The new vector optimization $\tilde{\mathbf{u}}$ can only acquire zero or positive real values, which accommodates the objective to increase the absolute magnitude and keep the phase

fixed of the data symbols received by the terminals while pushing the sum power of the transmitted precoded symbols to its minimum.

In a case of a single level M-PSK modulation, we define the optimization problem to minimize the sum power of the precoded symbols vector $\mathbf{x}$ as

$$\begin{aligned} \min_{\tilde{\mathbf{u}}} \quad & \|\mathbf{x}\|_2 \\ \text{s. t.} \quad & \tilde{u}_i \geq 0, \end{aligned} \tag{8}$$

for all $i = 1, 2, \ldots, N_u$.

By substituting (7) in (8) we get

$$\begin{aligned} \min_{\tilde{\mathbf{u}}} \quad & \|\mathbf{A}\tilde{\mathbf{u}} - \mathbf{d}\|_2 \\ \text{s. t.} \quad & \tilde{u}_i \geq 0, \end{aligned} \tag{9}$$

where $\mathbf{A} = \mathbf{W}_{\text{ZF}}\mathbf{B}$ and $\mathbf{d} = -\mathbf{W}_{\text{ZF}}\mathbf{B}\tilde{\mathbf{s}}$. Finally, we transform the objective function in (9) from the complex domain to the real domain. In this case, we apply an equality between the Euclidean norm of a complex vector $\tilde{z} = [\tilde{z_1}, \tilde{z_2}, \ldots, \tilde{z_i}]$ and a real vector $z = [z_1, z_2, \ldots, z_i]$, where $\tilde{z_i} = a_i + \imath b_i$ and $z_i = [a_i, b_i]$, to rewrite (9) as

$$\begin{aligned} \min_{\tilde{\mathbf{u}}} \quad & \|\widetilde{\mathbf{A}}\tilde{\mathbf{u}} - \tilde{\mathbf{d}}\|_2 \\ \text{s. t.} \quad & \tilde{u}_i \geq 0, \end{aligned} \tag{10}$$

where $\widetilde{\mathbf{A}} = [\text{Re}(\mathbf{A}); \text{Im}(\mathbf{A})] \in \mathbb{R}^{2N_t \times N_u}$ and $\tilde{\mathbf{d}} = [\text{Re}(\mathbf{d}^\dagger), \text{Im}(\mathbf{d}^\dagger)]^\dagger$.

In a case where the symbols generated from the multi-level amplitude and phase-shift keying (M-APSK) constellation, we need to fix the symbols on the internal levels ($\tilde{u}_i = 0$) and optimize the symbols only on the external level by increasing their absolute amplitude ($\tilde{u}_i \geq 0$). For this, we need to define the optimization problem by constraining the external and internal symbols separately as:

$$\begin{aligned} \min_{\tilde{\mathbf{u}}} \quad & \|\mathbf{x}\|_2 \\ \text{s. t.} \quad & \tilde{u}_i \geq 0, |s_i| \geq 1, \\ & \tilde{u}_i = 0, |s_i| < 1. \end{aligned} \tag{11}$$

We follow the same derivation steps as in the case of M-PSK modulation and get the following optimization expression:

$$\begin{aligned} \min_{\tilde{\mathbf{u}}} \quad & \|\widetilde{\mathbf{A}}\tilde{\mathbf{u}} - \tilde{\mathbf{d}}\|_2 \\ \text{s. t.} \quad & \tilde{u}_i \geq 0, |s_i| \geq 1, \\ & \tilde{u}_i = 0, |s_i| < 1. \end{aligned} \tag{12}$$

We can see that the problem (10) is a subset of the more general problem formulation (12).

The problem (10) is NNLS optimization problem. It can be solved in different ways found in the literature [21], [23]. The problem (12) can be solved by using CVX [24], [25]. After the optimization vector $\tilde{\mathbf{u}}$ is found, the gateway constructs the precoded signal using the equation (7). If the optimal solution is not found, then all the elements of $\tilde{\mathbf{u}}$ are equal to zero. In this case, the SLP technique is equivalent to the conventional ZF precoding technique

$$\mathbf{x} = \mathbf{W}_{\text{ZF}}\mathbf{B}(\tilde{\mathbf{s}} + \tilde{\mathbf{u}}^0) = \mathbf{W}_{\text{ZF}}(\Gamma \bullet \mathbf{s}). \tag{13}$$

Thus, in the worst case scenario the proposed SLP technique performs the same as the ZF precoding in terms of energy efficiency and sum power rate.

C. APPROXIMATE CLOSED-FORM SOLUTION

In this section, we propose an approximate closed-form algorithm to efficiently solve the optimization problem (10). The throughput performance of the Fast NNLS algorithm is not sufficient to operate in a real-time regime as we showed in [19]. The more complex convex optimization solver demonstrates even lower throughput [15], [26]. Instead, we devise a Fast NNLS based closed-form optimization algorithm, which gives a trade-off of lower power minimization and much faster processing time.

The conventional Fast NNLS algorithm finds the optimal regression coefficients through a number of iteration. In every iteration it dynamically chooses and solves a subset of quadratic equations from a complete set defined as

$$\tilde{\mathbf{u}} = (\widetilde{\mathbf{A}}^\dagger\widetilde{\mathbf{A}})^{-1}\widetilde{\mathbf{A}}^\dagger\tilde{\mathbf{d}}. \tag{14}$$

For Fast NNLS algorithm to converge the number of iteration is not fixed and can reach up to N_u. In every iteration the equation (14) is partially solved thought QR decomposition, which asymptotic complexity alone is of $O(N_t \times N_u^2)$.

We propose to substantially relax the complexity of the optimization problem by the assumption that the regression coefficients are mutually uncorrelated. In this case, the off-diagonal elements of the matrix product $(\widetilde{\mathbf{A}}^\dagger\widetilde{\mathbf{A}})$ are equal to zero as

$$\widetilde{\mathbf{A}}^\dagger\widetilde{\mathbf{A}} \approx \begin{bmatrix} \sum_{j=1}^{2N_t} \widetilde{A}_{j,1}^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sum_{j=1}^{2N_t} \widetilde{A}_{j,N_u}^2. \end{bmatrix} \tag{15}$$

By inserting (15) into (14) we derive an approximate closed-form solution for the optimization problem (10) as

$$\tilde{u}_i = \frac{1}{\sum_{j=1}^{2N_t} \widetilde{A}_{j,i}^2} \sum_{j=1}^{2N_t} \widetilde{A}_{j,i}\tilde{d}_j \geq 0, \tag{16}$$

for each element i of the vector $\tilde{\mathbf{u}}$. The solution of the equation (16) must be equal or greater than zero and cannot take negative values.

In order to solve (12) we extend (16) to differentiate symbols from external and internal constellation layers as

$$\tilde{u}_i = \begin{cases} \dfrac{1}{\sum_{j=1}^{2N_t} \widetilde{A}_{j,i}^2} \sum_{j=1}^{2N_t} \widetilde{A}_{j,i}\tilde{d}_j \geq 0, & |s_i| \geq 1 \\ 0, & |s_i| < 1. \end{cases} \tag{17}$$

In the extended expression the solution must be equal to zero for every symbol in internal layer $|s_i| < 1$. In this case the

Algorithm 1 Approximate Closed-Form Solution Algorithm

1: Input: $(\widetilde{\mathbf{A}} \in \mathbb{R}^{2N_t \times N_u}, \tilde{\mathbf{d}} \in \mathbb{R}^{N_t \times 1})$
2: Output: $\tilde{\mathbf{u}} \in \mathbb{R}^{N_u \times 1}$
3: **for** $i = 1, 2, \ldots, N_u$ **do**
4: $\quad a \leftarrow 0$
5: $\quad b \leftarrow 0$
6: $\quad$ **for** $j = 1, 2, \ldots, N_t$ **do**
7: $\quad\quad a \leftarrow a + \tilde{d}_j \widetilde{A}_{j,i}$
8: $\quad\quad b \leftarrow b + \widetilde{A}^2_{j,i}$
9: $\quad$ **end for**
10: $\quad \tilde{u}_i \leftarrow a/b$
11: $\quad$ **if** $\tilde{u}_i < 0$ **then**
12: $\quad\quad \tilde{u}_i \leftarrow 0$
13: $\quad$ **end if**
14: **end for**

internal constellation symbols are fixed to their original position. The (17) can be also applied to solve the problem (10), thus it is a complete solution for any type of PSK and APSK modulations.

The approximate solution (17) is solved and considered as converged in a single iteration contrary to the Fast NNLS. The asymptotic complexity of the complete approximate closed-form solution is of $O(N_t \times N_u)$, which is considerably less complex than Fast NNLS.

III. FPGA IMPLEMENTATION DESIGN

We use Vivado HLS to design the HDL core. Vivado HLS accelerates IP creation by enabling C, C++ and System C specifications to be directly targeted into Xilinx programmable devices without the need to manually create RTL. Thus, in this section, we translate the computationally efficient SLP technique to a pseudo-code and analyze its computational complexity. We optimize the core for Xilinx Kintex-7 xc7k410TFFG-2 FPGA model. This particular model is installed in a wide set of commercially available software defined radios (SDR) by National Instruments, like NI USRP (Universal Software Radio Peripheral) 2954R and FlexRIO (Reconfigurable IO) 7976R.

A. ALGORITHM DESCRIPTION

For the convenience of the implementation analysis, we rewrite the equation (16) as a pseudo-code algorithm (1). The algorithm consists of only two **for** loops, where which of them has a constant number of iterations, which allows to design the FPGA core at the target symbol throughput. The input arguments of the algorithm is the matrix $\widetilde{\mathbf{A}}$ and the vector $\tilde{\mathbf{d}}$. The output is a vector of the regression coefficients $\tilde{\mathbf{u}}$.

Finally, we derive the complete pseudo-code of the computationally efficient SLP technique in Algorithm 2. The input arguments of the algorithm is a Zero-Forcing precoding matrix $\mathbf{W_{ZF}}$, a vector of data symbols $\mathbf{s}$. There is no a dedicated input for the vector of SNR requirements Γ as it can be directly incorporated into the matrix $\mathbf{W_{ZF}}$. The output is

Algorithm 2 Computationally Efficient SLP Algorithm

1: Input: $(\mathbf{W_{ZF}} \in \mathbb{R}^{N_t \times N_u}, \mathbf{s} \in \mathbb{R}^{N_u \times 1})$
2: Output: $\mathbf{x} \in \mathbb{R}^{N_t \times 1}$
3: **for** $j = 1, 2, \ldots, N_t$ **do** ▷ Compute matrix $\mathbf{A}$
4: $\quad$ **for** $i = 1, 2, \ldots, N_u$ **do**
5: $\quad\quad A_{j,i} \leftarrow W_{\mathrm{ZF}j,i} s_i$
6: $\quad$ **end for**
7: **end for**
8: **for** $j = 1, 2, \ldots, N_t$ **do** ▷ Build matrix $\widetilde{\mathbf{A}}$
9: $\quad$ **for** $i = 1, 2, \ldots, N_u$ **do**
10: $\quad\quad \widetilde{A}_{j,i} \leftarrow \mathrm{Re}(A_{j,i})$
11: $\quad\quad \widetilde{A}_{j+N_t,i} \leftarrow \mathrm{Im}(A_{j,i})$
12: $\quad$ **end for**
13: **end for**
14: **for** $j = 1, 2, \ldots, 2\,N_t$ **do** ▷ Compute vector $\tilde{\mathbf{d}}$
15: $\quad \tilde{d}_i \leftarrow 0$
16: $\quad$ **for** $i = 1, 2, \ldots, N_u$ **do**
17: $\quad\quad \tilde{d}_j \leftarrow \tilde{d}_j + \widetilde{A}_{j,i}$
18: $\quad$ **end for**
19: **end for**
20: $\tilde{\mathbf{u}} \leftarrow$ **Algorithm** $1(\widetilde{\mathbf{A}}, (-\tilde{\mathbf{d}}))$ ▷ Compute vector $\tilde{\mathbf{u}}$
21: **for** $i = 1, 2, \ldots, N_u$ **do**
22: $\quad$ **if** $|s_i| < 1$ **then** ▷ Eq. (17) condition
23: $\quad\quad \tilde{u}_i \leftarrow 0$
24: $\quad$ **end if**
25: **end for**
26: **for** $j = 1, 2, \ldots, N_u$ **do** ▷ Compute vector $\mathbf{x}$
27: $\quad x_j \leftarrow 0$
28: $\quad$ **for** $i = 1, 2, \ldots, N_t$ **do**
29: $\quad\quad x_j \leftarrow x_j + W_{\mathrm{ZF}j,i} s_i + W_{\mathrm{ZF}j,i} \tilde{u}_i$
30: $\quad$ **end for**
31: **end for**

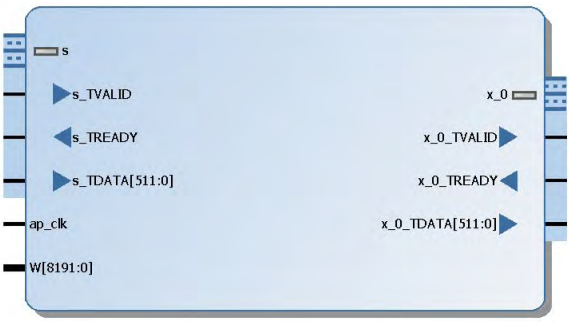

FIGURE 2. Core schematic symbol.

a vector of precoded symbols $\mathbf{x}$. We implement the condition check for multi-level modulation at the line 22 to fully implement the approximate closed-form solution (17). Therefore, the described algorithm does not need a configuration parameter to indicate the type of a symbol modulation at the input.

B. HDL CORE I/O PORTS DESCRIPTION

The input and output (I/O) ports of the HDL core is presented in Table 1. We designed the core using AXIS handshake for

TABLE 1. HDL Core I/O Ports.

Symbol	I/O	Width	Name	Description
ap_clk	I	1	Primary clock	Primary system clock. The system is synchronous and operates at the rising edge of this clk signal.
W	I	c_f_p*N_t*N_u	Precoding matrix	Complex values of the Zero-Forcing precoding matrix reshaped to a vector.
s_TDATA	I	c_f_p*N_u	Vector of input symbols	Complex values of the input symbols.
s_TVALID	I	1	AXIS input valid	Data input valid. When asserted the data are valid for input on the port s_TDATA.
s_TREADY	O	1	AXIS input ready	Data input ready. When high the core signals that the port s_TDATA is ready for input.
x0_TDATA	O	c_f_p*N_u	Vector of output symbols	Complex values of the output symbols.
x0_TVALID	O	1	AXIS output valid	Data output valid. When high the core signals that the data are valid for output on port x0_TDATA.
x0_TREADY	I	1	AXIS output ready	Data output ready. When asserted the data are ready to output on the port s_TDATA.

the optimal data transfer towards and from the core. The input port W receives a precomputed precoding matrix and has no handshake signaling. The data on this port should be ready before signaling to the port s_TREADY. The bit width of the data ports depends on the bit width of the complex fixed-point format (c_f_p).

TABLE 2. Data Port W Format.

W[MSB downto 0]				
$\mathrm{Im}(W_{\mathrm{ZF}N_t,N_u})$	$\mathrm{Re}(W_{\mathrm{ZF}N_t,N_u})$	...	$\mathrm{Im}(W_{\mathrm{ZF}1,1})$	$\mathrm{Re}(W_{\mathrm{ZF}1,1})$

The detailed format of the port W is described in Table 2. The real and imaginary parts of each entry of the matrix $\mathbf{W}_{\mathrm{ZF}}$ are concatenated and are mapped to a vector in the order as shown in the table. The entries of the matrix are concatenated row by row so that the first row should start at the bit 0, following by the second row and the last row should end at the most significant bit (MSB).

The format of the port s_TDATA is described in Table 3. The real and imaginary parts of each entry of the vector $\mathbf{s}$ are concatenated and are mapped to a vector in the order as shown in the table. The first entry starts at the bit 0, following by the second entry and the last entry ends at the most significant bit.

TABLE 3. Data Port s_TDATA Format.

s_TDATA[MSB downto 0]				
$\mathrm{Im}(s_{N_u})$	$\mathrm{Re}(s_{N_u})$	...	$\mathrm{Im}(s_1)$	$\mathrm{Re}(s_1)$

The format of the port x0_TDATA is described in Table 4. The real part of the first symbol is placed at the bit 0. Its imaginary part is appended after the real part. The rest of the symbols are concatenated in the same order until the MSB is the imaginary part of the last symbol.

TABLE 4. Data Port x0_TDATA Format.

x0_TDATA[MSB downto 0]				
$\mathrm{Im}(x_{N_u})$	$\mathrm{Re}(x_{N_u})$	...	$\mathrm{Im}(x_1)$	$\mathrm{Re}(x_1)$

C. FUNCTIONAL BEHAVIOR DESCRIPTION

In Fig. 3 we see the complete flow of the core functional behavior for multiple sets of symbols. In this demonstration, we feed the port W of the core with an identity matrix $\mathcal{I}_N \in \mathbb{R}^{N\times N}$ during all the cycles defined as

$$\mathcal{I}_N = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 1. \end{bmatrix} \tag{18}$$

The core reads $N_t \times N_u$ elements of the matrix and a set of N_u symbols in parallel in a single cycle. During the iteration cycle 0 we feed the s_TDATA port with a vector of symbols $\mathbf{s}_1 = [0.7071 + 0.7071i, 0.7071 + 0.7071i, \dots, 0.7071 + 0.7071] \in \mathbb{C}^N$ and switch s_TVALID from low to high. At the interval cycle 1 we switch s_TVALID back to low. We feed the port s_TDATA with a new set of symbols $\mathbf{s}_2 = [-0.7071 - 0.7071i, -0.7071 - 0.7071i, \dots, -0.7071 - 0.7071] \in \mathbb{C}^N$ and switch s_TVALID again from low to high for the period of interval cycle 2. We can see, that the port x0_TVALID switches from low to high during the same iteration cycle. We can read the data on the port x0_TDATA. At the interval cycle 4 we feed the port s_TDATA again with the set of symbols $\mathbf{s}_1$. We can see, that the output data corresponds to the input data delayed by 2 cycles as $\mathbf{x} = \mathcal{I}_N\mathbf{s}_1$ at the cycle 2 and $\mathbf{x} = \mathcal{I}_N\mathbf{s}_2$ at the cycle 4. The core produces

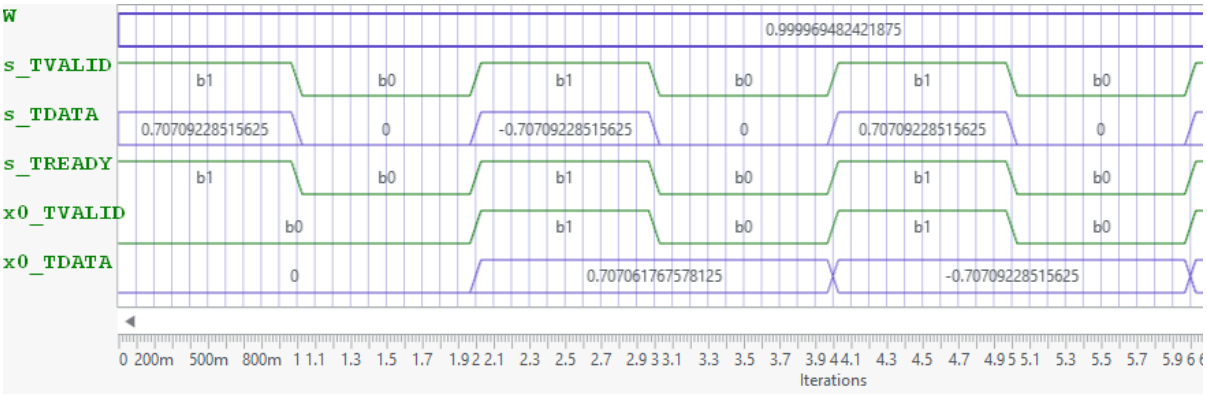

FIGURE 3. Functional behavior of the control ports for the first 7 iteration cycles.

TABLE 5. HDL Core Resource Occupation on Kintex-7 (xc7k410TFFG-2).

N^o of Beams	DSP48E	Slices	LUT	Cycles@Clock
2	16	479	216	2@166 MHz
6	72	2019	2488	2@166 MHz
12	288	9891	9938	2@166 MHz
16	512	11683	19010	2@166 MHz
20	800	21187	27602	2@166 MHz
Available	1540	508400	254200	

output data every 2 cycles, which are delayed by 2 cycles with respect to the corresponding input data.

D. FPGA RESOURCE AND TIMING PERFORMANCE

We define the complex fixed-point format as 16 signed bits (C1.15) for the design of the HDL core. The format allocates 16 signed bits to real values and 16 bits to complex values, which results in a total of 32 bits for a single complex value. We target the HDL core to operate at the symbol rate of 83 MSymbols per second. The motivation behind the target is the new symbols rates, which are considered in the DVB-S2X standard [27]. We estimate the resource consumption by the core design for a number of transmitting antennas and receiver terminals $N = N_t = N_u = 2$, 6, 12, 16 and 20. In Table 5 we see the numerical estimation of the FPGA resource utilization for a different number of beams. For all the scenarios the core is optimized to operate at a 166 MHz clock (≈6 ns per cycle) with a cycle interval 2. The clock allows to operate at the $166\,\text{MHz}/2 = 83$ MSymbols per second symbol rate per each receiver terminal. For the 20 transmitting antennas and 20 receiver terminals case, the design utilizes around 50 percent of the DSP blocks available at the given FPGA model (xc7k410TFFG-2).

IV. NUMERICAL VALIDATION

In the numerical validation, we consider the MIMO system, which has an equal number of the transmit and receive antennas $N_t = N_u = N$. We accordingly generate a full rank $N \times N$ MIMO channel matrix with a 2-norm matrix condition number defined as

$$\kappa_2(\mathbf{H}) = ||\mathbf{H}||_2 \cdot ||\mathbf{H}^{-1}||_2. \tag{19}$$

The matrix condition number corresponds to the ratio of the largest singular value of that matrix to smallest singular value. In the case of the MIMO system, the matrix condition number describes the power imbalance in the channel [16]. We average the results over 50 channel matrices with defined condition number. We benchmark the proposed approximate closed-form, Fast NNLS, and CVX optimization algorithms and measure the total average power of the precoded symbols generated by the techniques in selected channel scenarios. We set the SNR requirement to $\Gamma_i = 1$ for every $i = 1, 2, \ldots, N_u$.

In Fig. 4 we can see the normalized total average power of the approximate closed-form and Fast NNLS optimization algorithms. The power of all the techniques is normalized point by point in reference to the power of the precoded symbols generated by ZF precoder. This way we can directly compare the increase in the performance of the techniques in the same conditions. The condition number of the channel matrices used in the benchmarks is set as a function of $0.5\,N$, $1\,N$ and $3\,N$. For example, for 20 antennas at the transmitter, the condition number of all the 20×20 channel matrices is 10, 20 and 60 accordingly. It is evident that the sum power minimization results are better for channel matrices with larger condition numbers. The approximate closed-form algorithm performs very closely to the full solution of Fast NNLS up to certain dimensions of the channel matrices. We can see that the point where the Fast NNLS substantially outperforms the proposed closed-form algorithm in each case depends on the condition number of the channel matrices. The lower the condition number the larger channel matrix dimensions can be successfully handled by the closed-form algorithm. It is also evident, that with a greater condition number of the channel matrices we achieve larger power reduction for both algorithms.

The approximate closed-form algorithm demonstrates a fair performance than benchmarked against the full Fast NNLS solution. It is evident that the condition number of

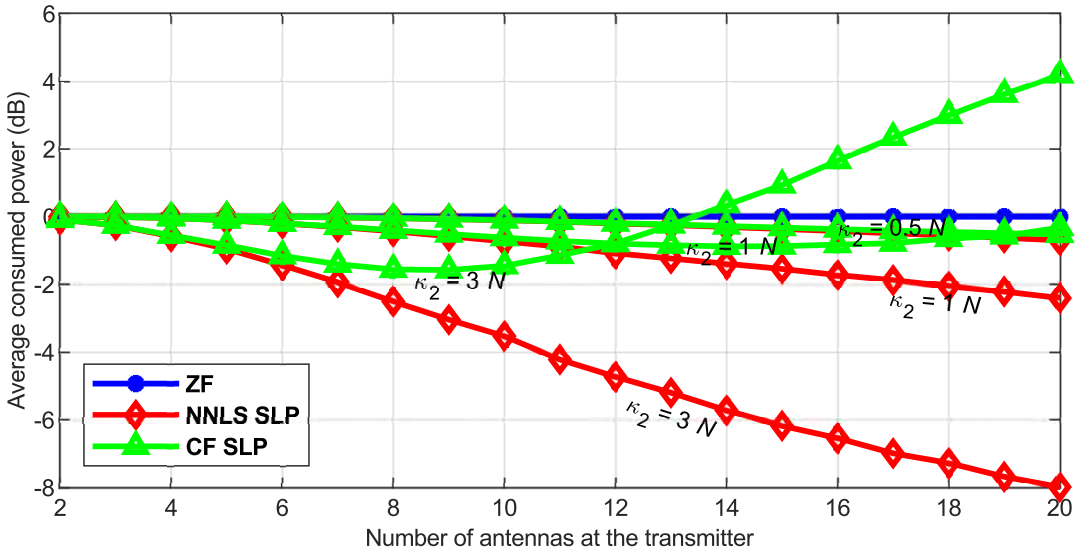

FIGURE 4. Average total power of the precoded symbols calculated by ZF, the approximate close-form and Fast NNLS algorithms.

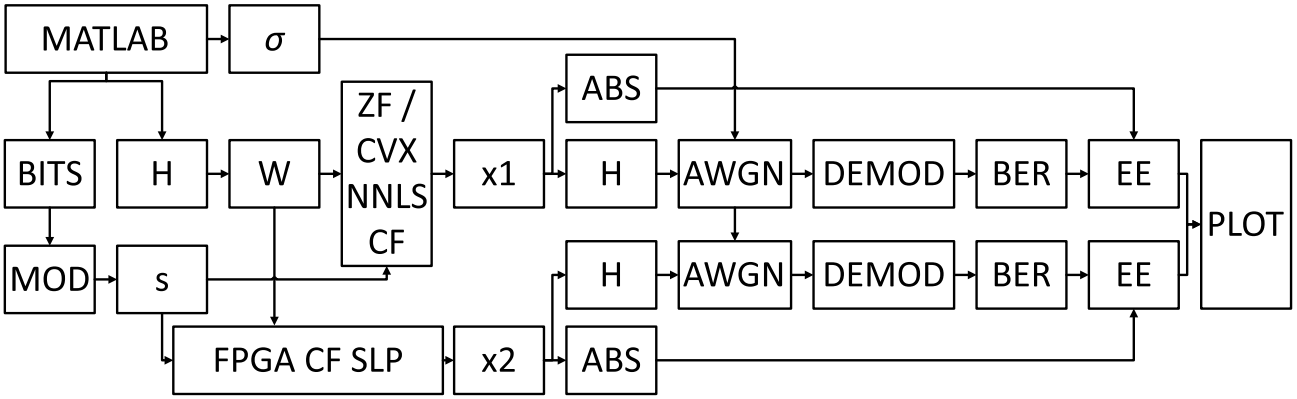

FIGURE 5. Schematic block diagram of the benchmark of the SLP implementation.

the channel matrix used in the optimization has an important influence on the demonstrated results of both benchmarked algorithms. The greater condition number gives better power minimization results in both techniques, but at the same time, the approximate closed-form solution can not efficiently handle the large dimensions of the channel matrices with large condition numbers. The issue of a high condition number can be addressed through channel aware user scheduling as in [28]. In this way, we can always select channels, which have plausible condition numbers for the approximate closed-form algorithm to operate with its best performance.

V. EXPERIMENTAL VALIDATION

In this section, we benchmark the performance of Algorithm 2 implemented on the HDL core and deployed on an actual FPGA against the same algorithm running in a MATLAB environment. The HDL core is implemented using fixed-point arithmetic, while MATLAB is running in a float-point precision mode. We estimate the difference of the arithmetic precision implementations.

In Fig. 5 we show the block diagram of the conducted benchmark. We benchmark energy efficiency (EE) of the presented SLP technique implemented on FPGA and in MATLAB. In MATLAB we generate data bits, a channel matrix $\mathbf{H}$, a precoding matrix $\mathbf{W}$ and modulated data symbols $\mathbf{s}$. We generate a $N_t \times N_u$ channel matrix with a specific 2-norm matrix condition number.

The MATLAB calculates precoded symbols **x1** using the ZF or the computationally efficient SLP implemented with MATLAB CVX (CVX SLP), Fast NNLS (NNLS SLP) and the approximate closed-form (CF SLP) optimization algorithms. At the same time, MATLAB transfers the same set of the generated symbols with the precoding matrix to the FPGA node, which runs the HDL core to calculate precoded symbols **x2** using the approximate closed-form algorithm (FPGA SLP). The two versions **x1** and **x2** of the precoded symbols are multiplied by the channel matrix and mixed

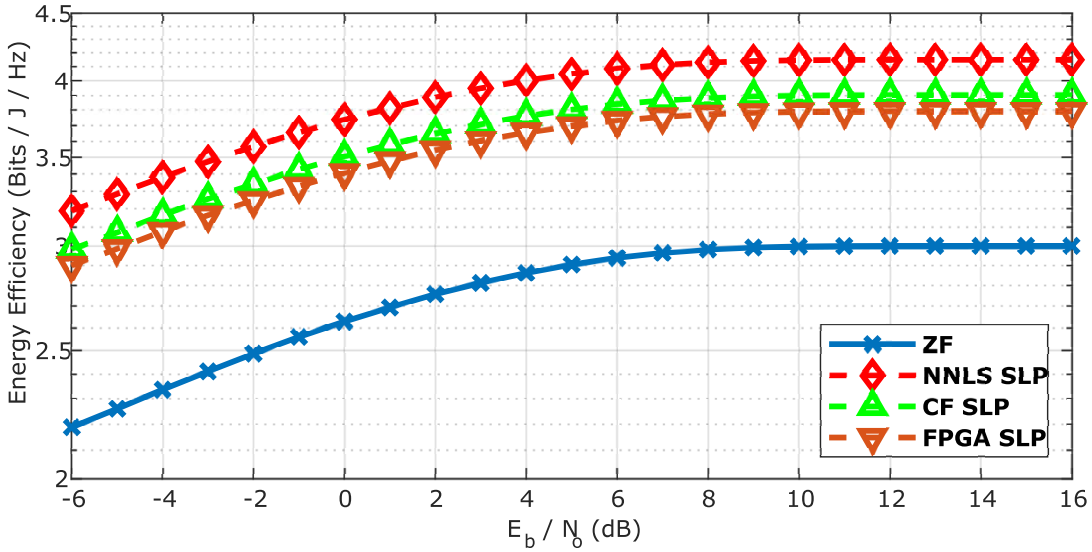

FIGURE 6. Energy efficiency curves of 8-PSK for FPGA implementation of FPGA SLP compared to ZF, NNLS SLP and CF SLP on MATLAB.

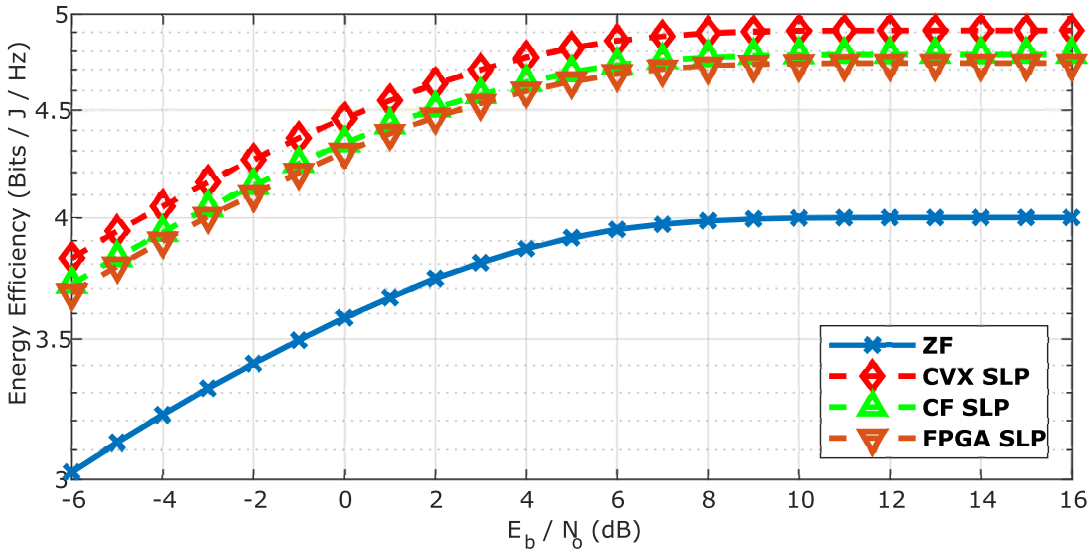

FIGURE 7. Energy efficiency curves of 16-APSK ($\gamma = 3.15$) for FPGA implementation of FPGA SLP compared to ZF, CVX SLP and CF SLP on MATLAB.

with the AWGN noise. MATLAB demodulates the resulting signals and calculates BER scores. Finally, we calculate EE as

$$\mathrm{EE}(E_b/N_0) = \frac{\log_2(M)(1-\mathrm{BER}(E_b/N_0))}{\|\mathbf{x}_{\mathrm{norm}}\|_2}, \qquad (20)$$

where $\|\mathbf{x}_{\mathrm{norm}}\|_2$ is the normalized average sum power of the precoded symbols and $E_b/N_0 = 10\log_{10}(\frac{1}{3\sigma^2})$ is the energy per bit to noise power spectral density ratio.

In Fig. 6 we can see the energy efficiency curves as a function of E_b/N_0 of the ZF, NNLS SLP and CF SLP algorithms running in MATLAB and on FPGA. We generate Gray mapped M-PSK modulation symbols and average the benchmarks over 50 iterations of $N_t = N_u = 6$ channel matrix with a condition number ($\kappa_2(\mathbf{H})$) fixed to 18. The difference between the performance of the Fast NNLS and the CF algorithms running on MATLAB is around 2.5 dB due to the approximation method used in the closed-form solution. We also observe an additional 1 dB difference between the MATLAB and FPGA implementations of the CF algorithm due to losses in fixed-point arithmetic.

In Fig. 7 we demonstrate energy efficiency benchmarks as a function of E_b/N_0 of the ZF, CVX SLP and CF SLP algorithms running in MATLAB and on FPGA. We generate symbols with 16-APSK constellation with constellation radius ratio $\gamma = 3.15$ and average the benchmarks over 10 iterations of $N_t = N_u = 6$ channel matrix with a condition number $\kappa_2(\mathbf{H}) = 18$. In this case, we observe that the CVX optimization is 1 dB more efficient than the CF algorithm in

the MATLAB environment. But as we previously discussed and demonstrated in [15] CVX has much lower symbol throughput than the CF algorithm and ZF. The FPGA and MATLAB implementations of the CF algorithm demonstrate similar results of an additional 1 dB performance difference due to losses in fixed-point arithmetic.

The CF algorithm designed for FPGAs delivers considerably improved energy efficiency when compared to ZF in all the benchmarks. The Fast NNLS and CVX solutions are shown to outperform the CF algorithm, but they are not designed to run in real-time on an FPGA. The FPGA implementation operates at a high symbol throughput but has an additional energy efficiency loss of 1 dB due to losses in fixed-point arithmetic. This drawback can be addressed if we choose to increase the number of bits in the fixed-point arithmetic at the cost of extra FPGA resources.

VI. CONCLUSION

In this paper, we devised an FPGA-accelerated design of the energy and computationally efficient symbol-level precoding operating on a real-time operation mode, the first such case to our knowledge. We successfully deployed and validated the design on an actual FPGA platform.

We developed an approximate closed-form solution and showed that it can fairly improve energy efficiency in comparison to the conventional optimization algorithms. The performance of the proposed algorithm was shown to be sensitive towards a condition number of the channel matrix. Therefore, a channel aware user scheduling must be applied together with the proposed technique.

We describe the algorithmic code, the I/O ports mapping and the functional behavior of the FPGA design. We optimized the design of the HDL core to operate at up to 83 MSymbols per second throughput per each receiver terminal with up to 20 simultaneously operating terminal units while utilizing a reasonable amount of the FPGA resources. The achieved symbol throughput is considered for the DVB-S2X standard [27] communications. The designed HDL core universally supports single- and multi-level symbol modulations with fixed-phase optimization. It can directly operate with any M-PSK and M-APSK constellation and does not need to reconfigure. The approximate closed-form algorithm, which we developed for the FPGA design, demonstrated a 2 dB loss of energy efficiency during the conducted benchmarks against conventional Fast NNLS and CVX optimization algorithms. We also measured an additional 1 dB loss of energy efficiency of the approximate closed-form algorithm when deployed on an actual FPGA platform. This can be addressed with an increased precision of the fixed-point implementation at the expense of FPGA resources.

REFERENCES

[1] *Mobile and Wireless Communications Enablers for Twenty-Twenty (2020) Information Society*, Eurescom, Berlin, Germany 2014.
[2] S. Chatzinotas, B. Ottersten, and R. De Gaudenzi, *Cooperative and Cognitive Satellite Systems*, 1st ed. Orlando, FL, USA: Academic, 2015.
[3] I. Binyamini and I. Bergel, "Adaptive precoder using sign error feedback for FEXT cancellation in multichannel downstream VDSL," *IEEE Trans. Signal Process.*, vol. 61, no. 9, pp. 2383–2393, May 2013.
[4] Y. Qian, X. Zhou, J. Li, F. Shu, and D. N. K. Jayakody, "A novel precoding and impulsive noise mitigation scheme for MIMO power line communication systems," *IEEE Syst. J.*, pp. 1–12, 2018.
[5] C. B. Peel, B. M. Hochwald, and A. L. Swindlehurst, "A vector-perturbation technique for near-capacity multiantenna multiuser communication—Part I: Channel inversion and regularization," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 195–202, Jan. 2005.
[6] E. Björnson, M. Bengtsson, and B. Ottersten, "Optimal multiuser transmit beamforming: A difficult problem with a simple solution structure [lecture notes]," *IEEE Signal Process. Mag.*, vol. 31, no. 4, pp. 142–148, Jul. 2014.
[7] M. Alodeh *et al.*, "Symbol-level and multicast precoding for multiuser multiantenna downlink: A state-of-the-art, classification, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1733–1757, 3rd Quart., 2018.
[8] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[9] K. Lange and H. Zhou, "MM algorithms for geometric and signomial programming," *Math. Program.*, vol. 143, nos. 1–2, pp. 339–356, Feb. 2014. [Online]. Available: https://doi.org/10.1007/s10107-012-0612-1
[10] B. Hamet, T. Kolb, C. Rohde, F. Leschka, M. U. P. Bhave, and A. Lidde, "Over-the-air operation of mobile and multicast linear precoding for a multi-spot-beam mobile satellite service," in *Proc. 24th Ka Broadband Commun. Conf.*, Oct. 2018. [Online]. Available: http://proceedings.kaconf.org/lista.php?trova=S
[11] Y. Liu and W.-K. Ma, "Symbol-level precoding is symbol-perturbed ZF when energy efficiency is sought," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 3869–3873.
[12] A. Li and C. Masouros, "Interference exploitation precoding made practical: Optimal closed-form solutions for PSK modulations," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7661–7676, Sep. 2018.
[13] A. Haqiqatnejad, F. Kayhan, and B. Ottersten, "Power minimizer symbol-level precoding: A closed-form suboptimal solution," *IEEE Signal Process. Lett.*, vol. 25, no. 11, pp. 1730–1734, Nov. 2018.
[14] C. G. Tsinos, A. Kalantari, S. Chatzinotas, and B. Ottersten, "Symbol-level precoding with low resolution DACs for large-scale array MU-MIMO systems," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2018, pp. 1–5.
[15] J. Krivochiza, A. Kalantari, S. Chatzinotas, and B. Ottersten, "Low complexity symbol-level design for linear precoding systems," in *Proc. Symp. Inf. Theory Signal Process. Benelux*, 2017, p. 117.
[16] J. C. Merlano-Duncan, J. Krivochiza, S. Andrenacci, S. Chatzinotas, and B. Ottersten, "Computationally efficient symbol-level precoding communications demonstrator," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–5.
[17] J. Krivochiza, J. C. Merlano-Duncan, S. Andrenacci, S. Chatzinotas, and B. Ottersten, "Computationally and energy efficient symbol-level precoding communications demonstrator," *Phys. Commun.*, vol. 28, pp. 108–115, Jun. 2018.
[18] J. Duncan, J. Krivochiza, S. Andrenacci, S. Chatzinotas, and B. Ottersten, "Hardware demonstration of precoded communications in multi-beam UHTS systems," in *Proc. 36th Int. Commun. Satell. Syst. Conf. (ICSSC)*, Oct. 2018, pp. 1–5.
[19] J. Krivochiza, J. C. Merlano-Duncan, S. Andrenacci, S. Chatzinotas, and B. Ottersten, "Closed-form solution for computationally efficient symbol-level precoding," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018.
[20] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Philadelphia, PA, USA: SIAM, 1995. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9781611971217
[21] R. Bro and S. D. Jong, "A fast non-negativity-constrained least squares algorithm," *J. Chemometrics*, vol. 11, no. 5, pp. 393–401, 1997.
[22] M. Alodeh, S. Chatzinotas, and B. Ottersten, "Constructive multiuser interference in symbol level precoding for the MISO downlink channel," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2239–2252, May 2015.
[23] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Philadelphia, PA, USA: SIAM, 1995.
[24] M. C. Grant and S. P. Boyd. (Mar. 2014). *CVX: MATLAB Software for Disciplined Convex Programming, Version 2.1*. [Online]. Available: http://cvxr.com/cvx

[25] M. C. Grant and S. P. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control* (Lecture Notes in Control and Information Sciences), V. Blondel, S. Boyd, and H. Kimura, Eds. Berlin, Germany: Springer-Verlag, 2008, pp. 95–110. [Online]. Available: http://stanford.edu/~boyd/graph_dcp.html

[26] M. Alodeh, S. Chatzinotas, and B. Ottersten, "Constructive interference through symbol level precoding for multi-level modulation," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[27] *Digital Video Broadcasting (DVB); Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications; Part 2: DVB-S2 Extensions (DVB-S2X)*, ETSI, Sophia Antipolis, France, 2015.

[28] A. Bandi, B. S. M. R, S. Maleki, S. Chatzinotas, and B. Ottersten, "A novel approach to joint user selection and precoding for multiuser MISO downlink channels," in *Proc. 6th IEEE Global Conf. Signal Inf. Process.*, Nov. 2018, pp. 1–5.

JEVGENIJ KRIVOCHIZA received the B.Sc. and M.Sc. degrees in electronic engineering with the focus on telecommunications physics and electronics from the Faculty of Physics, Vilnius University, in 2011 and 2013, respectively. He is currently pursuing the Ph.D. degree with the Signal Processing and Communications Group-SIGCOM, Interdisciplinary Centre for Security, Reliability, and Trust, University of Luxembourg. He has experience in research on, FPGA development, software-defined radios (SDR), digital signal processing (DSP), interference mitigation, DVB-S2X, DVB-S2, and LTE systems. He is currently involved in the research field of DSP on an SDR platform, and develops advanced precoding and beamforming techniques for the next generation satellite communications.

JUAN MERLANO DUNCAN received the Diploma degree in electrical engineering from the Universidad del Norte, Barranquilla, Colombia, in 2004, the M.Sc. and Ph.D. Diploma degrees from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2009 and 2012, respectively. At UPC, he was responsible for the design and implementation of a radar system known as SABRINA, which was the first ground-based bistatic radar receiver using spaceborne platforms, such as ERS-2, ENVISAT, and TerraSAR-X as opportunity transmitters (C and X bands). He was also in charge of the implementation of a ground-based array of transmitters which was able to monitor land subsidence with sub-wavelength precision. These two implementations involved FPGA design, embedded programming, and analog RF/microwave design. In 2013, he joined the Institute National de la Recherche Scientifique, Montreal, Canada, as a Research Assistant in the design and implementation of the cognitive radio networks by means of software development and FPGA programming. His research interests are wireless communications, remote sensing, distributed systems, frequency distribution and carrier synchronization systems, software-defined radios, and embedded systems.

STEFANO ANDRENACCI received the M.S. degree in telecommunication engineering *(cum laude)* from the Polytechnic University of Marche, Ancona, Italy, in 2008, and the Ph.D. degree in telecommunication engineering from the Department of Biomedical Engineering, Electronics and Telecommunications from the Polytechnic University of Marche, in 2011. From 2011 to 2014, he was a Post-Doctoral Researcher with the Interdepartmental Centre for Industrial Research on Information and Communication Technologies, University of Bologna, where he worked on interference management techniques under some ESA research projects and on hardware implementation of satellite terminals under some ESA technological development projects. From 2014 to 2015, he was a Post-Doctoral Researcher with the Department of Electrical and Information Engineering "Guglielmo Marconi", University of Bologna. Since 2015, he has been a Research Associate with the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg. His research activities are mainly focused on interference management techniques, synchronization chain design and synchronization techniques for digital receivers, DVB-S2/S2x systems, DVB-RCS2 systems, software-defined radios (SDR), and spread spectrum systems. During his Ph.D., he collaborated with Aethra spa (Ancona) on immersive telepresence systems and with Mavigex srl (Bologna), on the implementation of digital satellites receivers.

SYMEON CHATZINOTAS (S'06–M'09–SM'13) received the M.Eng. degree in telecommunications from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2003, and the M.Sc. and Ph.D. degrees in electronic engineering from the University of Surrey, Surrey, U.K., in 2006 and 2009, respectively. He is currently the Deputy Head of the SIGCOM Research Group, Interdisciplinary Centre for Security, Reliability, and Trust, University of Luxembourg, Luxembourg, and also a Visiting Professor with the University of Parma, Italy. He was involved in numerous research and development projects for the Institute of Informatics Telecommunications, National Center for Scientific Research Demokritos, the Institute of Telematics and Informatics, Center of Research and Technology Hellas, and the Mobile Communications Research Group, Center of Communication Systems Research, University of Surrey. He has over 300 publications, 3000 citations, and an H-Index of 30 according to Google Scholar. He was a co-recipient of the 2014 IEEE Distinguished Contributions to Satellite Communications Award, the CROWNCOM 2015 Best Paper Award, and the 2018 EURASIC JWCN Best Paper Award.

BJÖRN OTTERSTEN (S'87–M'89–SM'99–F'04) was born in Stockholm, Sweden, in 1961. He received the M.S. degree in electrical engineering and applied physics from Linköping University, Linköping, Sweden, in 1986, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1989. He has held research positions at the Department of Electrical Engineering, Linköping University, the Information Systems Laboratory, Stanford University, the Katholieke Universiteit Leuven, Leuven, Belgium, and the University of Luxembourg, Luxembourg. From 1996 to 1997, he was the Director of research with ArrayComm Inc., a start-up in San Jose, CA, USA, based on his patented technology. In 1991, he was appointed as a Professor of signal processing with the Royal Institute of Technology (KTH), Stockholm, Sweden. From 1992 to 2004, he was the Head of the Department for Signals, Sensors, and Systems, KTH, and from 2004 to 2008, he was the Dean of the School of Electrical Engineering, KTH. He is currently the Director of the Interdisciplinary Centre for Security, Reliability, and Trust, University of Luxembourg. As a Digital Champion of Luxembourg, he acts as an Adviser to the European Commission. He is a Fellow of the EURASIP. He has co-authored journal papers that received the IEEE Signal Processing Society Best Paper Award in 1993, 2001, 2006, and 2013, respectively, and three IEEE conference papers receiving Best Paper Awards. In 2011, he received the IEEE Signal Processing Society Technical Achievement Award. He has received the European Research Council Advanced Research Grant twice, from 2009 to 2013 and from 2017 to 2021. He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and on the Editorial Board of the *IEEE Signal Processing Magazine*. He is currently an Editor-in-Chief of the *EURASIP Signal Processing Journal*, and a member of the editorial boards of the *EURASIP Journal of Advances Signal Processing* and *Foundations and Trends of Signal Processing*.