IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 3, 2021, accepted February 13, 2021, date of publication February 16, 2021, date of current version February 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3059936

Design Optimization for Low-Complexity FPGA Implementation of Symbol-Level Multiuser Precoding

ALIREZA HAQIQATNEJAD, (Graduate Student Member, IEEE), JEVGENIJ KRIVOCHIZA, (Member, IEEE), JUAN CARLOS MERLANO DUNCAN, (Senior Member, IEEE), SYMEON CHATZINOTAS, (Senior Member, IEEE), AND BJÖRN OTTERSTEN, (Fellow, IEEE)
Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, L-1855 Luxembourg City, Luxembourg

Corresponding author: Alireza Haqiqatnejad (alireza.haqiqatnejad@uni.lu)

This work was supported in part by the Luxembourg National Research Fund (FNR), Enhanced Signal Space opTImization for satellite comMunication Systems (ESSTIMS), under CORE Junior Project C16/IS/11332341, in part by the AFR-PPP End-to-End Signal Processing Algorithms for Precoded Satellite Communications, under Grant FNR11481283, in part by the Exploiting Interference for Physical-Layer Security in 5G networks (CI-PHY), under Grant FNR11607830, and in part by the Cognitive Cohesive Networks of Distributed Units for Active and Passive Space Applications (COHESAT) under Grant CORE Junior FNR11689919.

ABSTRACT This paper proposes and validates a low-complexity FPGA design for symbol-level precoding (SLP) in multiuser multiple-input single-output (MISO) downlink communication systems. In the optimal case, the symbol-level precoded transmit signal is obtained as the solution to an optimization problem tailored for a given set of users' data symbols. This symbol-by-symbol design, however, imposes excessive computational complexity on the system. To alleviate this issue, we aim to reduce the per-symbol complexity of the SLP scheme by developing an approximate yet computationally-efficient closed-form solution. The proposed solution allows us to achieve a high symbol throughput in real-time implementations. To develop the FPGA design, we express the proposed solution in an algorithmic way and translate it to hardware description language (HDL). We then optimize the processing to accelerate the performance and generate the corresponding intellectual property (IP) core. We provide the synthesis report for the generated IP core, including performance and resource utilization estimates and interface descriptions. To validate our design, we simulate an uncoded transmission over a downlink multiuser channel using the LabVIEW software, where the SLP IP core is implemented as a clock-driven logic (CDL) unit. Our simulation results show that a throughput of 100 Mega symbols per second per user can be achieved via the proposed SLP design. We further use the MATLAB software to produce numerical results for the conventional zero-forcing (ZF) and the optimal SLP techniques as benchmarks for comparison. Thereby, it is shown that the proposed FPGA implementation of SLP offers an improvement of up to 50 percent in power efficiency compared to the ZF precoding. Remarkably, it enjoys the same per-symbol complexity order as that of the ZF technique. We also evaluate the loss of the real-time SLP design, introduced by the algebraic approximations and arithmetic inaccuracies, with respect to the optimal scheme.

INDEX TERMS Constructive interference, convex optimization, downlink multiuser multiple-input single-output (MISO) system, field-programmable gate array (FPGA), hardware description language (HDL), non-negative least squares (NNLS) problem, symbol-level precoding.

I. INTRODUCTION

Co-channel interference is one of the main limiting factors in wireless multiuser multi-input multi-output (MIMO) downlink channels. Multiuser precoding techniques address this issue by pre-processing and spatially multiplexing the users' intended data streams prior to transmission. In case perfect transmit-side channel state information (CSIT) is non-causally available, this processing is known to achieve the sum-rate capacity of the multiuser MIMO downlink

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

channel at an impractically high computational complexity [1]. In addition to simple linear precoding schemes such as maximum ratio transmission (MRT) [2], (regularized) zero-forcing (ZF) [3], [4], and minimum mean square error (MMSE) [5], extensive research focusing on practical yet efficient multiuser precoding has been reported in the literature; see, e.g., [6]–[11].

The existing multiuser precoding schemes can be broadly classified into two groups, namely, block-level and symbol-level techniques. Typically, a block-level precoding design exploits only the CSIT in order to suppress the co-channel interference, regardless of the users' data symbols. However, it has been shown that co-channel interference might not always be destructive. In fact, it is possible to exploit the constructive part of the interference, or even converting the interfering components into constructive interference (CI) by exploiting also the users' data symbols [12]. As a result, the conventional viewpoint on multiuser precoding can turn from block-level approaches towards a more sophisticated design that uses the data symbols in addition to the CSIT, which is known as symbol-level precoding (SLP) [13], [14].

The symbol-level design of a multiuser precoder can considerably improve the system's power efficiency. However, it comes with some practical challenges that need to be properly addressed. For example, some such challenges are a substantially increased computational burden at the transmitter, the need for setting the modulation scheme in advance, sensitivity of the design to CSIT errors, and sub-optimality of signal-to-interference-plus-noise ratio (SINR) pilots and log-likelihood ratio (LLR) calculation algorithms; see [15], [16]. Among the challenges mentioned above, we mainly focus on the high computation cost of SLP, which is primarily due to the fact that the design needs to be optimized specifically for every set of users' symbols. In high-throughput wireless communication systems, online computation of precoding may suffer from the high complexity of the symbol-level design. On the contrary, an offline (codebook design) computation may lead to an unfavorable computation cost for high-order modulation schemes, even with a moderate number of users [17], [18]. Therefore, the considerable performance gain offered by a symbol-level precoder has been motivating to find a more computationally efficient solution.

In this line of research, some effort has been made so far towards deriving low-complexity solutions to the SLP design problem, e.g., [19]–[23], and accordingly, some other studies have addressed efficient hardware demonstrations of these low-complexity SLP techniques, e.g., [24], [25]. In [20], the authors propose an iterative algorithm with a closed-form update equation for the SLP problem with a max-min fair design criterion, where the algorithm is shown to converge to the optimal solution in a few iterations. In another work [21], the power minimization SLP is addressed with strict phase constraints on the received signals. Then, a computationally-efficient approximate solution is suggested for this particular case for the phase-shift keying (PSK) modulation scheme. The authors demonstrated an FPGA-accelerated design of this computationally efficient solution in [26] and showed that it can provide a high symbol throughput in a real-time operation mode. In [22], another closed-form sub-optimal solution is obtained for the power minimization SLP using the Karush-Kuhn-Tucker (KKT) optimality conditions. The proposed solution is essentially based on distance-preserving CI regions [27], and applies to modulation schemes with any given constellation shape and order. This solution has been improved in [23] by considering an additional validation step in deriving the approximate solution at the cost of slightly increasing the computational complexity.

In this paper, we focus on the closed-form sub-optimal solution proposed in [22] obtained for the SNR-constrained power minimization SLP problem. Accordingly, the main contributions of this work are as follows:

- We further simplify this solution using some intermediate approximation steps and derive a new solution which has lower computational complexity. The approximations are mainly introduced to reduce the computation cost of the SLP design. This simplification further facilitates the design of a low-complexity algorithm operating in a real-time mode.
- We show through analytical evaluation of the computational complexity that the proposed approximate SLP solution has the same per-symbol complexity order as that of the conventional ZF precoding.
- To validate our design, we target FPGA implementation of the proposed SLP algorithm. First, we express the algorithm in C++ language and then convert it to hardware description language (HDL). The HDL implementation enables us to generate the intellectual property (IP) core targeted for a specific FPGA device. We analyze and compare two different cases: the original non-optimized HDL design and the case where the processing is optimized through function pipelining, loop unrolling and array partitioning. This indicates how optimizing the HDL design can accelerate the performance. We also provide the synthesis results for the generated IP core. In particular, the timing and latency estimates, the FPGA resource utilization ratios, and the register-transfer level (RTL) I/O ports specifications are reported.
- The synthesis and implementation results show that the proposed FPGA design is able to provide a high throughput of 100 Mega symbols per second per user for a 4×4 system with QPSK signaling. Furthermore, numerical results are obtained by simulating a multiuser downlink system in the LabVIEW and MATLAB environments and applying different precoding techniques. Our results show that the proposed low-complexity HDL implementation of the SLP algorithm substantially outperforms the ZF technique in terms of power efficiency.

Organization: The remainder of this paper is organized as follows. We describe the considered system and signal model in Section II. In Section III, we provide an overview on the power minimization SLP problem with distance-preserving

CI constraints. We then revisit the sub-optimal closed-form SLP solution, which is followed by proposing our simplified SLP design algorithm. In Section IV, we explain the HDL design and optimization steps for the proposed algorithm and report some performance estimates for the real-time FPGA implementation. In Section V, we evaluate our HDL design by presenting the results of simulation tests. Finally, we conclude the paper in Section VI.

Notation: We use uppercase and lowercase bold-faced letters to denote matrices and vectors, respectively. The sets of real and complex numbers are represented, respectively, by $\mathbb{R}$ and $\mathbb{C}$. For a complex input, $(\cdot)^*$ denotes the conjugate operator, and $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ respectively denote the real and the imaginary part. For matrices and vectors, $[\,\cdot\,]^{\mathrm{T}}$ denotes the transpose operation. For matrices, $[\,\cdot\,]^{\dagger}$ denotes the pseudo-inverse of the input matrix. For vectors, $\|\cdot\|$ stands for the Euclidean norm. Operator $\mathrm{diag}(\cdot)$ represents a diagonal matrix. We define $\mathrm{j} \triangleq \sqrt{-1}$ and use $\mathbf{I}$ and $\mathbf{0}$ to represent, respectively, the identity matrix and the zero matrix (or the zero vector, depending on the context) of appropriate dimension. The probability function and the statistical expectation are respectively denoted by $\mathrm{P}\{\cdot\}$ and $\mathrm{E}\{\cdot\}$. The operator $\otimes$ stands for the Kronecker product.

II. SYSTEM MODEL

We consider the downlink of a single-carrier multiuser multiple-input single-output (MISO) system where the base station (BS), which is equipped with an antenna array of N_{t} elements, simultaneously communicates to N_{u} single-antenna users through multiplexing independent data streams within the same time-frequency resource block. Let $s_i(t)$, $i = 1, 2, \ldots, N_{\mathrm{u}}$, denote the intended discrete-time complex modulated symbol for the ith user at symbol time $t = 0, 1, 2, \ldots$. We assume that each symbol $s_i(t)$ is taken from a finite equiprobable constellation set with unit average power, i.e., $\mathrm{E}\{s_i(t)s_i(t)^*\} = 1$, where the expectation is taken over t. For the brevity of notation, we focus on a specific symbol time and drop the time index t throughout the paper. Furthermore, for the sake of simplicity and without loss of generality, we assume that identical modulation schemes are used for all the users.

To manage the multiuser interference, the BS employs a non-linear symbol-level precoding scheme that calculates the transmit signal specifically for any set of input symbols. Let $\bar{\mathbf{u}} = [u_1, \ldots, u_{N_{\mathrm{t}}}]^{\mathrm{T}} \in \mathbb{C}^{N_{\mathrm{t}}\times 1}$ denote the precoded transmit signal to be propagated towards the users. Then, the baseband representation of the signal received by the user i is given as

$$r_i = \mathbf{h}_i^{\mathrm{T}}\mathbf{u} + z_i, \quad i = 1, 2, \ldots, N_{\mathrm{u}}, \tag{1}$$

where $\mathbf{h}_i \in \mathbb{C}^{N_{\mathrm{t}}\times 1}$ denotes the instantaneous coefficients of the frequency-flat Rayleigh block fading channel between the BS's antennas and the ith user which are distributed as $\mathbf{h}_i \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_{\mathrm{t}}})$, and z_k is the additive circularly symmetric complex Gaussian noise with distribution $z_i \sim \mathcal{CN}(0, \sigma_i^2)$. At the receiver side, the ith user may use the optimal single-user detector based on a maximum-likelihood (ML) decision rule to detect its intended symbol s_i, and therefore, the structure of the receiver is independent of the precoding design.

The symbol-level precoder calculates the transmit signal $\mathbf{u}$ through solving an objective-oriented optimization problem on a symbol-by-symbol basis. This nonlinear-precoded signal is directly designed without computing a precoding matrix, and therefore may not be uniquely decomposable as a linear combination of the users' precoding vectors.

To proceed, it is more convenient to define the following equivalent real-valued vectors:

$$\mathbf{u} = \begin{bmatrix}\mathrm{Re}(\mathbf{x})\\ \mathrm{Im}(\mathbf{x})\end{bmatrix} \in \mathbb{R}^{2N_{\mathrm{t}}\times 1},$$
$$\mathbf{s}_i = \begin{bmatrix}\mathrm{Re}(s_i)\\ \mathrm{Im}(s_i)\end{bmatrix} \in \mathbb{R}^2,\ i = 1, 2, \ldots, N_{\mathrm{u}},$$
$$\mathbf{s} = \left[\mathbf{s}_1^{\mathrm{T}}, \mathbf{s}_2^{\mathrm{T}}, \ldots, \mathbf{s}_{N_{\mathrm{u}}}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{2N_{\mathrm{u}}\times 1},$$
$$\mathbf{H}_i \triangleq \begin{bmatrix}\mathrm{Re}(\mathbf{h}_i^{\mathrm{T}}) & -\mathrm{Im}(\mathbf{h}_i^{\mathrm{T}})\\ \mathrm{Im}(\mathbf{h}_i^{\mathrm{T}}) & \mathrm{Re}(\mathbf{h}_i^{\mathrm{T}})\end{bmatrix} \in \mathbb{R}^{2\times 2N_{\mathrm{t}}},\quad i = 1, 2, \ldots, N_{\mathrm{u}}.$$

Using the above definitions, the real-valued noise-free signal received by the ith user can be represented as $\mathbf{H}_i\mathbf{u}$.

The convention in designing a symbol-level precoder is to find the transmit signal such that the noise-free received signal of each user locates in a specific region, called constructive interference (CI) region, that corresponds to the users' intended symbols. The CI regions are typically defined with the aim of improving the symbol detection accuracy at the receiver side, and hence, are specifically defined for the modulation scheme in use. Among a variety of definitions, e.g., in [13], [14], [27], we adopt a particular type of CI regions named distance-preserving CI regions [27], supporting generic constellation sets of any shape and size. By definition, any two points belonging to two distinct distance-preserving CI regions are distanced by at least the distance between the corresponding constellation points. Accordingly, the distance-preserving CI region associated with any constellation symbol is a subset of its ML decision region.

The above definition for the distance-preserving CI regions can be expressed in an explicit mathematical form as follows. For any $i \in \{1, 2, \ldots, N_{\mathrm{u}}\}$, the noise-free received signal $\mathbf{H}_i\mathbf{u}$ locates in the distance-preserving CI region of symbol $\mathbf{s}_i$ if the following equality condition is met:

$$\mathbf{A}_i\left(\mathbf{H}_i\mathbf{u} - \sigma_i\sqrt{\gamma_i}\,\mathbf{s}_i\right) = \mathbf{t}_i, \tag{2}$$

where γ_i denotes the target SNR for the ith user, $\mathbf{t}_i \in \mathbb{R}^2$ is a vector of non-negative design variables, and the 2×2 real-valued matrix $\mathbf{A}_i = [\mathbf{a}_{i,1}, \mathbf{a}_{i,2}]^{\mathrm{T}}$ contains the normal vectors of the associated ML decision boundaries for symbol $\mathbf{s}_i$. The two normal vectors $\mathbf{a}_{i,1}$ and $\mathbf{a}_{i,2}$ can simply be obtained using the following criteria:

- If $\mathbf{s}_i$ is an outer constellation point, we obtain $\mathbf{a}_{i,1}$ and $\mathbf{a}_{i,2}$ by subtracting symbol $\mathbf{s}_i$ from its two neighboring constellation points.
- If $\mathbf{s}_i$ is an inner constellation point, we set $\mathbf{a}_{i,1} = \mathbf{0}$ and $\mathbf{a}_{i,2} = \mathbf{0}$.

TABLE 1. Normal vectors corresponding to QPSK symbols.

$\mathbf{s}_i$	$\mathbf{a}_{i,1}^{\mathrm{T}}$	$\mathbf{a}_{i,2}^{\mathrm{T}}$
$0.7071+\mathrm{j}0.7071$	$[+1,0]$	$[0,+1]$
$-0.7071+\mathrm{j}0.7071$	$[-1,0]$	$[0,+1]$
$-0.7071-\mathrm{j}0.7071$	$[-1,0]$	$[0,-1]$
$0.7071-\mathrm{j}0.7071$	$[+1,0]$	$[0,-1]$

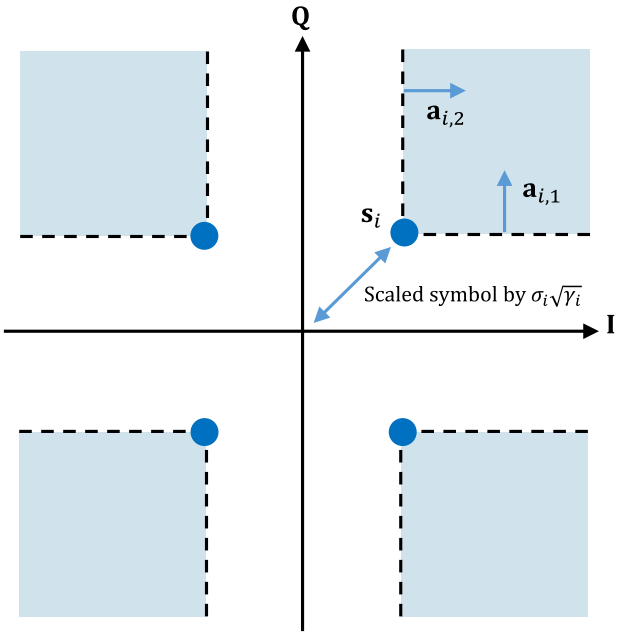

FIGURE 1. Distance-preserving CI regions, depicted as blue areas, for QPSK constellation.

As an example, in Table 1, we show the normal vectors that correspond to symbol $\mathbf{s}_i$ taken from a QPSK constellation. Note that the normal vectors given in Table 1 are normalized such that they have a unit Euclidean norm. It is further worth noting that there exist cases in which either $\mathbf{a}_{i,1}$ or $\mathbf{a}_{i,2}$ is set as $\mathbf{0}$ while the other is not, for instance, in case $\mathbf{s}_i$ is an outer but not corner point of a QAM constellation. An illustration of the distance-preserving CI regions for QPSK constellation is shown in Fig. 1.The interested readers are referred to [18] for a detailed discussion on the characteristics of distance-preserving CI regions.

We refer to the equality condition (2) as the CI constraint for the ith user. Collecting the CI constraints for all the users, we can write the stacked CI constraint in a compact form as

$$\mathbf{A}\left(\mathbf{H}\mathbf{u}-\Sigma\Gamma\mathbf{s}\right)=\mathbf{t}, \tag{3}$$

where we have used the following definitions:

$$\mathbf{A}\triangleq\begin{bmatrix}\mathbf{A}_1 & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_2 & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{N_{\mathrm{u}}}\end{bmatrix}\in\mathbb{R}^{2N_{\mathrm{u}}\times 2N_{\mathrm{u}}},$$

$$\mathbf{H}\triangleq\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\\ \vdots\\ \mathbf{H}_{N_{\mathrm{u}}}\end{bmatrix}\in\mathbb{R}^{2N_{\mathrm{u}}\times 2N_{\mathrm{t}}},$$

$$\mathbf{t}\triangleq\begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\\ \vdots\\ \mathbf{t}_{N_{\mathrm{u}}}\end{bmatrix}\in\mathbb{R}^{2N_{\mathrm{u}}\times 1},$$

$$\Sigma\triangleq\mathrm{diag}(\sigma_1,\sigma_2,\ldots,\sigma_{N_{\mathrm{u}}})\otimes\mathbf{I}_2\in\mathbb{R}^{2N_{\mathrm{u}}\times 2N_{\mathrm{u}}},$$

$$\Gamma\triangleq\mathrm{diag}(\sqrt{\gamma_1},\sqrt{\gamma_2},\ldots,\sqrt{\gamma_{N_{\mathrm{u}}}})\otimes\mathbf{I}_2\in\mathbb{R}^{2N_{\mathrm{u}}\times 2N_{\mathrm{u}}}.$$

It is shown in [18] that, for any given constellation set, the sub-matrices $\{\mathbf{A}_1,\ldots,\mathbf{A}_{N_{\mathrm{u}}}\}$ can be formed as non-singular matrices, and therefore, matrix $\mathbf{A}$ is always invertible. As a result, we can rewrite (3) as

$$\mathbf{H}\mathbf{u}=\Sigma\Gamma\mathbf{s}+\mathbf{A}^{-1}\mathbf{t}, \tag{4}$$

where, as a consequence of the block diagonal structure of matrix $\mathbf{A}$, we have

$$\mathbf{A}^{-1}\triangleq\begin{bmatrix}\mathbf{A}_1^{-1} & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_2^{-1} & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{N_{\mathrm{u}}}^{-1}\end{bmatrix}.$$

The compact CI constraint given in (4) will be used in the next section to cast the precoding optimization problem.

III. PRECODING DESIGN FORMULATION

We consider an SNR-constrained power minimization design criterion subject to CI constraints for all the users. Using the expression for CI constraints in (4), the corresponding design problem can be written as

$$\begin{aligned}&\min_{\mathbf{u},\mathbf{t}}\ \|\mathbf{u}\|^2\\ &\text{s.t. }\mathbf{H}\mathbf{u}=\Sigma\Gamma\mathbf{s}+\mathbf{A}^{-1}\mathbf{t},\quad \mathbf{t}\succeq\mathbf{0}.\end{aligned} \tag{5}$$

It is straightforward to show that the optimal vector $\mathbf{u}^*$ solving the optimization problem (5) is given by

$$\mathbf{u}^*=\mathbf{H}^{\dagger}\left(\Sigma\Gamma\mathbf{s}+\mathbf{A}^{-1}\mathbf{t}^*\right), \tag{6}$$

where $\mathbf{t}^*$ is the solution to the following non-negative least squares (NNLS) problem:

$$\min_{\mathbf{t}\succeq\mathbf{0}}\quad\|\mathbf{H}^{\dagger}\Sigma\Gamma\mathbf{s}+\mathbf{H}^{\dagger}\mathbf{A}^{-1}\mathbf{t}\|^2. \tag{7}$$

It follows that the precoding design problem of interest can be tackled through solving the NNLS optimization (7). For simplicity, we denote $\mathbf{B}\triangleq\mathbf{H}^{\dagger}\mathbf{A}^{-1}$ and $\mathbf{y}\triangleq-\mathbf{H}^{\dagger}\Sigma\Gamma\mathbf{s}$; thereby, the NNLS problem (7) can be rewritten in the standard form as

$$\min_{\mathbf{t}\succeq\mathbf{0}}\quad\|\mathbf{B}\mathbf{t}-\mathbf{y}\|^2. \tag{8}$$

The NNLS problem (8) is not amenable to a closed-form solution due to the non-negative constraints on $\mathbf{t}$. Various iterative algorithms exist to solve an NNLS problem, such as the well-known Lawson and Hanson method [28], the fast NNLS algorithm (FNNLS) [29], and those based on projected/proximal gradient method [30]–[32]. These NNLS algorithms, in the best-known case, require tens of iterations to converge. As an illustrative example, using the accelerated gradient method, which enjoys a superlinear convergence rate, it takes nearly 100 iterations to have a residual error of 10^{-3} with respect to the optimum. In a practical application of symbol-level precoding, this process has to be done either for every symbol period or every possible symbol set corresponding to N_{u} users. It is worth noting that there are plenty of efficient blocks capable of performing many iterations. For instance, if the processing is sequential, one can use pipelining, and if not, one can unroll the loops. However, each of these iterations, which may involve several matrix inversions and demanding computation steps, requires a total number of operations that occupy almost all the resources of modern FPGAs or CPUs/GPUs. As a consequence, no pipelining/unrolling can be implemented. Therefore, simplifying the design and optimizing the processing is essential to enable real-time operation of the precoder. In the first place, this motivates the need for a more computationally-efficient, though possibly approximate, solution for the precoding design problem.

A. APPROXIMATE SOLUTION

Let $\mathbf{t}^* = [t^*_{1,1}, t^*_{1,2}, \ldots, t^*_{N_{\mathrm{u}},1}, t^*_{N_{\mathrm{u}},2}]^{\mathrm{T}} \triangleq [t^*_1, t^*_2, \ldots, t^*_{2N_{\mathrm{u}}}]^{\mathrm{T}}$ denote the minimizer of the NNLS problem in (8). We refer to the set of indices n for which $t^*_n > 0$ as the support of $\mathbf{t}^*$, or the optimal support, denoted by

$$\Lambda^* = \{n : n = 1, 2, \ldots, 2N_{\mathrm{u}}, t^*_n > 0\}. \tag{9}$$

Given the optimal support Λ^*, the minimizer of (7) can be simply computed by $\mathbf{B}^{\dagger}_{\Lambda^*}\mathbf{y}$ with appropriate zero-padding, where $\mathbf{B}_{\Lambda^*}$ denotes the matrix composed of those columns of $\mathbf{B}$ that correspond to the indices in Λ^*. Hence, one may attempt to solve (7) equivalently by identifying Λ^*. However, finding Λ^* is as complex as solving (7) for the optimal solution. Alternatively, we can obtain an approximation of Λ^*, denoted by $\hat{\Lambda}$, in a non-iterative manner [22], [23]. This allows us to derive an approximate solution $\hat{\mathbf{t}}$ given in closed-form.

In this work, we mainly focus on the approximate closed-form solution proposed in [22] which is composed of two steps as follows:

i. Obtain an approximation of the support as

$$\hat{\Lambda} = \left\{n : n = 1, 2, \ldots, 2N_{\mathrm{u}},\ \mathbf{y}^{\mathrm{T}}\mathbf{b}_n \geq 0\right\}, \tag{10}$$

where $\mathbf{b}_n$ denotes the nth column of $\mathbf{B}$.

ii. Let $L \triangleq |\hat{\Lambda}|$ denote the length of the approximate support set. Build a $2N_{\mathrm{u}} \times L$ matrix $\mathbf{B}_{\hat{\Lambda}}$ consisting of those columns in $\mathbf{B}$ that are indexed in $\hat{\Lambda}$ and let the columns of $\mathbf{B}_{\hat{\Lambda}}$ be indexed as $\mathbf{b}_n$ where $n \in \hat{\Lambda}$. Then, calculate an approximate solution by solving a reduced system of linear equations as

$$\hat{t}_n = \left\{\left[\mathbf{B}^{\dagger}_{\hat{\Lambda}}\mathbf{y}\right]_n\right\}_+, \tag{11}$$

and $\hat{t}_n = 0$ otherwise, where $[\cdot]_n$ denotes the element that corresponds to the nth variable in $\mathbf{t}$, and operation $\{\cdot\}_+$ stands for $\max\{\cdot, 0\}$.

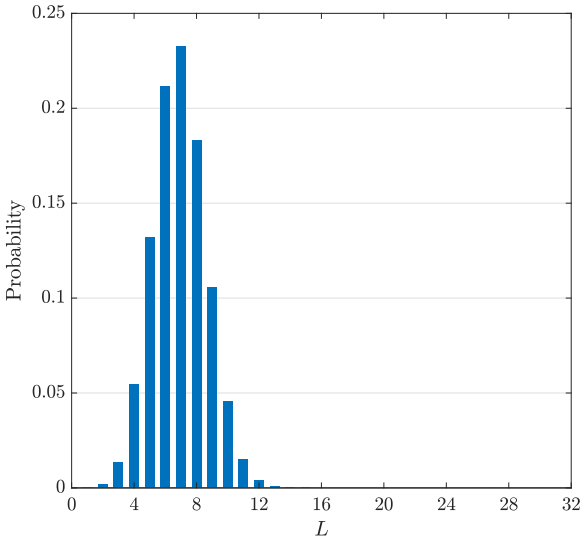

FIGURE 2. Experimental probability mass function of L.

This approximate closed-form solution involves a matrix pseudo-inverse operation as in (11), which is computationally costly in practice. In the sequel, we propose an approximate alternative operation to eliminate the need for computation of this matrix pseudo-inverse.

B. LOW-COMPLEXITY IMPLEMENTATION

Our experiments show that, on average, only a few number of inner products $\mathbf{y}^{\mathrm{T}}\mathbf{b}_n$ out of a total number of $2N_{\mathrm{u}}$ are non-negative, and hence, we usually have $L \ll 2N_{\mathrm{u}}$. As a consequence, the matrix $\mathbf{B}_{\hat{\Lambda}}$ has more rows than columns. In Fig. 2, we support this observation by plotting the empirical probability mass function of L which is obtained by averaging the realizations of L from 10^6 trials (10^3 symbol periods over 10^3 channel realizations) in a scenario with $N_{\mathrm{t}} = N_{\mathrm{u}} = 16$. It can be seen from Fig. 2 that $\mathrm{P}\{L \leq 3N_{\mathrm{u}}/4\} \approx 0.99$, i.e., the length of the approximated support is, with high probability, smaller than $3/4$ of the total number of elements. Based on this observation, we assume that the columns $\{\mathbf{b}_n : n \in \hat{\Lambda}\}$ are mutually orthogonal. Such an assumption leads us to the following approximation:

$$\left(\mathbf{B}_{\hat{\Lambda}}\mathbf{B}^{\mathrm{T}}_{\hat{\Lambda}}\right)^{-1} \approx \mathrm{diag}\left(\left\{\frac{1}{\|\mathbf{b}_n\|^2} : n \in \hat{\Lambda}\right\}\right). \tag{12}$$

As a result, the pseudo-inverse of matrix $\mathbf{B}_{\hat{\Lambda}}$ can be approximated as

$$\mathbf{B}^{\dagger}_{\hat{\Lambda}} = \mathbf{B}^{\mathrm{T}}_{\hat{\Lambda}}\left(\mathbf{B}_{\hat{\Lambda}}\mathbf{B}^{\mathrm{T}}_{\hat{\Lambda}}\right)^{-1} \approx \mathbf{B}^{\mathrm{T}}_{\hat{\Lambda}}\,\mathrm{diag}\left(\left\{\frac{1}{\|\mathbf{b}_n\|^2} : n \in \hat{\Lambda}\right\}\right). \tag{13}$$

Therefore, by plugging (13) into $\mathbf{B}_{\hat{\Lambda}}^{\dagger}\mathbf{y}$, we obtain

$$\hat{t}_n = \begin{cases} \mathbf{y}^{\mathrm{T}}\mathbf{b}_n/\|\mathbf{b}_n\|^2 & n \in \hat{\Lambda}, \\ 0 & n \notin \hat{\Lambda}. \end{cases} \tag{14}$$

Given the approximate solution $\hat{\mathbf{t}} = [\hat{t}_1, \hat{t}_1, \ldots, \hat{t}_{2N_{\mathrm{u}}}]^{\mathrm{T}}$, we use (6) to obtain the vector of precoded transmit signal. The pseudo-code of the proposed low-complexity approximate precoding solution is summarized in Algorithm 1. It is important to note that the non-negative constraints $\mathbf{t} \succeq \mathbf{0}$ are all satisfied by the SLP design in Algorithm 1. This implies that the approximation (13) does not lead to violation of the SNR constraints and the users' SNR requirements are guaranteed under the proposed approximate SLP solution.

Algorithm 1 Approximate Low-Complexity SLP Solution

Input : $\mathbf{H}^{\dagger}$, Σ, Γ, $\mathbf{s}$
Output : $\mathbf{u}$
$\mathbf{A}^{-1} \leftarrow$ lookup($\mathbf{s}$) ▷ Build matrix $\mathbf{A}^{-1}$
$\mathbf{B} \leftarrow \mathbf{H}^{\dagger}\mathbf{A}^{-1}$ ▷ Build matrix $\mathbf{B}$
$\mathbf{y} \leftarrow -\mathbf{H}^{\dagger}\Sigma\Gamma\mathbf{s}$ ▷ Build vector $\mathbf{y}$
for $n = 1$ **to** $2N_{\mathrm{u}}$ **do** ▷ Collect column-wise norms
    $c_n \leftarrow \mathbf{b}_n^{\mathrm{T}}\mathbf{b}_n$
end for
for $n = 1$ **to** $2N_{\mathrm{u}}$ **do**
    $d_n \leftarrow \mathbf{y}^{\mathrm{T}}\mathbf{b}_n$
    if $d_n \geq 0$ **then** ▷ Approximate the support
        $t_n \leftarrow \mathbf{y}^{\mathrm{T}}\mathbf{b}_n/c_n$ ▷ Compute vector $\mathbf{t}$
    else
        $t_n \leftarrow 0$
    end if
end for
$\mathbf{u}^{*} \leftarrow \mathbf{B}\mathbf{t} - \mathbf{y}$ ▷ Compute vector $\mathbf{u}$

We remark that in Algorithm 1, a simple lookup method is used to avoid calculation of matrix $\mathbf{A}^{-1}$ at each symbol period. More precisely, given the modulation scheme, one can extract all the possible realizations for a sub-matrix $\mathbf{A}_i$, which is equal to the modulation order, calculate its inverse and store them in a lookup table. At the time of implementation, each sub-matrix $\mathbf{A}_i^{-1}$ can be read from the lookup table with respect to the given symbol $\mathbf{s}_i$, and the entire matrix $\mathbf{A}$ can be constructed accordingly. Furthermore, for the ease of implementation, the matrices Σ and Γ can be incorporated into the symbol vector $\mathbf{s}$, as we will see in the FPGA design section.

The proposed solution in Algorithm 1 consists of a number of loops with known and constant number of iterations, each of which includes some basic arithmetic operations, e.g., addition and multiplication. We report in Table 2 the actual arithmetic complexity of Algorithm 1, including the separate complexity of each computation step as well as the overall complexity, in terms of the number of floating-point operations (FLOPs). It follows from Table 2 that Algorithm 1 has a dominating complexity order of $\mathcal{O}(N_{\mathrm{t}}N_{\mathrm{u}})$, in the limiting case where $N_{\mathrm{t}}, N_{\mathrm{u}} \to \infty$, and therefore, it enjoys the exact same per-symbol complexity order as that of the ZF precoding technique. Based on this comparison, we state that the proposed SLP solution has low computational complexity, and hence, is suitable for real-time implementation.

TABLE 2. Actual arithmetic complexity of Algorithm 1.

Computation	# Iterations	FLOPs
$\mathbf{H}^{\dagger}\mathbf{A}^{-1}$	1	$12N_{\mathrm{t}}N_{\mathrm{u}}$
$\mathbf{H}^{\dagger}\Sigma\Gamma\mathbf{s}$	1	$8N_{\mathrm{t}}N_{\mathrm{u}} + 4N_{\mathrm{u}} - 2N_{\mathrm{t}}$
$\mathbf{b}_n^{\mathrm{T}}\mathbf{b}_n$	$2N_{\mathrm{u}}$	$8N_{\mathrm{t}}N_{\mathrm{u}} - 2N_{\mathrm{u}}$
$\mathbf{y}^{\mathrm{T}}\mathbf{b}_n/c_n$	$2N_{\mathrm{u}}$	$8N_{\mathrm{t}}N_{\mathrm{u}}$
$\mathbf{B}\mathbf{t} - \mathbf{y}$	1	$8N_{\mathrm{t}}N_{\mathrm{u}}$
Overall		$44N_{\mathrm{t}}N_{\mathrm{u}} + 2N_{\mathrm{u}} - 2N_{\mathrm{t}}$

IV. FPGA DESIGN

To enable implementation of the proposed low-complexity SLP solution, we design the IP core using the Xilinx Vivado HLS tool. The Vivado HLS tool transforms a C specification, such as C, C++, or SystemC, into a register-transfer level (RTL) implementation that can be synthesized into Xilinx programmable devices. In this work, we have used version 2017.3 of the Xilinx Vivado HLS software and designed the IP core for Xilinx Kintex-7 xc7k410tffv900-2 FPGA part.

To generate the IP core, we have translated the algorithmic description of Algorithm 1 into C++ language. To achieve an accelerated performance and higher throughputs, we have optimized the code through many techniques, such as pipelining the functions, unrolling the loops, and partitioning the arrays. Pipelining and unrolling both improve the hardware function's performance by exploiting the parallelism between function and loop iterations. In particular, pipelining allows the operations in a function/loop to be implemented in a concurrent manner and unrolling creates multiple copies of the loop body and adjusts the loop iteration counter accordingly. These techniques have been applied to the design by adding the so-called "directives" into the C++ code. In the following, we refer to the design used in this work by applying the above techniques as the optimized HDL design. On the other hand, the original design without applying any of the above optimization techniques is referred to as the non-optimized design. Later in this section, we present the resource utilization and performance estimates for both non-optimized and optimized HDL implementations to emphasize how the design benefits from such code optimizations. We have further utilized the Vivado HLS matrix algebra library for efficient calculation of matrix multiplications. The C++ code has then been synthesized using the Vivado HLS tool, and the RTL implementation has been extracted as an intellectual property (IP) catalog.

A schematic block design of the IP core generated for a $(N_{\mathrm{t}}, N_{\mathrm{u}}) = (4, 4)$ system is depicted in Fig. 3. The design

TABLE 3. Interface specifications of the IP core.

RTL port	Direction	Bit width	Protocol	Description
ap_clk	Input	1	ap_ctrl_hs	Primary design clock
ap_rst_n	Input	1	ap_ctrl_hs	Interface reset (active-low)
ap_start	Input	1	ap_ctrl_hs	Block execution control (active-high)
ap_done	Output	1	ap_ctrl_hs	Complete-transaction indicator (active-high)
ap_idle	Output	1	ap_ctrl_hs	Operating/idle indicator (active-high)
ap_ready	Output	1	ap_ctrl_hs	Ready-for-new-inputs indicator (active-high)
pinvH_V	Input	$4BN_{\mathrm{t}}N_{\mathrm{u}}$	ap_none	Real-valued pseudo-inverse of the channel matrix
s_V_TDATA	Input	$2BN_{\mathrm{u}}$	axis	Real-valued vector of the users' symbols
s_V_TVALID	Input	1	axis	Data input valid
s_V_TREADY	Output	1	axis	Data input ready
u_V_TDATA	Output	$2BN_{\mathrm{t}}$	axis	Real-valued vector of precoded transmit signal
u_V_TVALID	Output	1	axis	Data output valid
u_V_TREADY	Input	1	axis	Data output ready

TABLE 4. Structure of the RTL data ports.

Data port	Format
s_V_TDATA	$\mathrm{Re}(s_1) \mid \mathrm{Im}(s_1) \mid \mathrm{Re}(s_2) \mid \mathrm{Im}(s_2) \mid \cdots \mid \mathrm{Re}(s_{N_{\mathrm{u}}}) \mid \mathrm{Im}(s_{N_{\mathrm{u}}})$
u_V_TDATA	$\mathrm{Re}(u_1) \mid \mathrm{Re}(u_2) \mid \cdots \mid \mathrm{Re}(u_{N_{\mathrm{t}}}) \mid \mathrm{Im}(u_1) \mid \mathrm{Im}(u_2) \mid \cdots \mid \mathrm{Im}(u_{N_{\mathrm{t}}})$

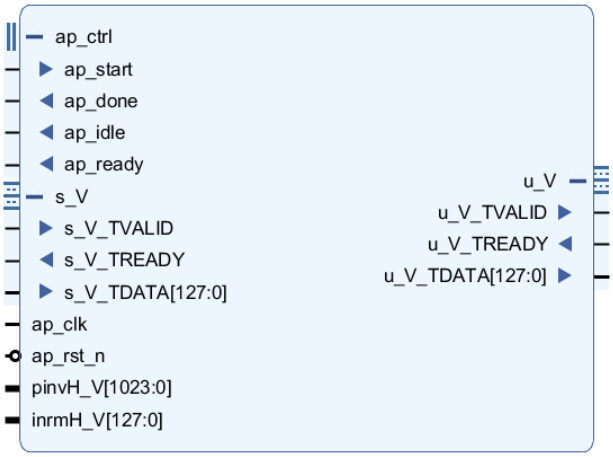

FIGURE 3. Block design of the proposed SLP IP core.

takes the matrix $\mathbf{H}^{\dagger}$ and the vector $\mathbf{s}$ as data inputs to execute Algorithm 1. These two inputs are shown as ports pinvH_V and s_V_TDATA in Fig. 3. Note that we do not consider dedicated inputs for the matrices Σ and Γ, but instead, we absorb the corresponding noise variances and target SNRs into the input vector $\mathbf{s}$. The only output of the HDL design is the precoded vector $\mathbf{u}$ which is placed on port u_V_TDATA of the IP core.

A. RTL I/O PORTS DESCRIPTION

The IP core associated with the HDL design has a number of data I/O ports. In addition, a block-level I/O control handshake protocol has been added to control the RTL design independently of the data ports. We summarize the specifications and behavior of all the HDL I/O ports in Table 3. Note that the bit width of a data port is determined by the bit width of the fixed-point format, which is denoted by B. In this work, we adopt a 6.10 signed fixed-point format for the RTL design such that it represents the integer and fraction parts, respectively, by 5 and 10 bits, and the sign is specified by one bit. Therefore, the real and imaginary parts are allocated 16 signed bits each, resulting in a total number of 32 bits for a single complex value.

To have an efficient data transfer towards and from the IP core, we adopt an AXIS handshake protocol for the I/O data ports. The precomputed pseudo-inverse of the real-valued channel matrix feeds the input data port pinvH_V, and therefore, this port does not need a handshake signaling. The data on this port must be ready before signaling to the port s_V_TREADY. The real and imaginary parts of each element of matrix $\mathbf{H}^{\dagger}$ are reshaped row by row into an array of length $4BN_{\mathrm{t}}N_{\mathrm{u}}$ bits. The first element of the first row starts at the the most significant bit, while the last element of the last row ends at bit 0. We further illustrate in Table 4 the formats of the data ports s_V_TDATA and u_V_TDATA. The s_V_TDATA port contains the elements of the symbol vector $\mathbf{s}$ in the order shown in Table 4, which are mapped to an array of length $2BN_{\mathrm{u}}$ bits. The real part of the first element starts at the most significant bit and the imaginary part of the last element ends at bit 0. The u_V_TDATA port, on the other hand, has

TABLE 5. Performance estimates of the non-optimized HDL design.

Timing/Clock period (ns)	
Target	10.00
C synthesis	8.63
Post-synthesis	5.18
Post-implementation	7.01
Latency (clock cycles)	
Latency	1493
Interval	1494

TABLE 6. Performance estimates of the optimized HDL design.

Timing/Clock period (ns)	
Target	10.00
C synthesis	8.72
Post-synthesis	5.52
Post-implementation	8.83
Latency (clock cycles)	
Latency	9
Interval	1

a different format from that of the s_V_TDATA port. The imaginary parts of all the elements of the precoded vector $\mathbf{u}$ are concatenated and appended to the real parts of all the elements. The first element's real part starts with the most significant bit and the the last element's imaginary part ends at bit 0.

B. RESOURCE UTILIZATION AND TIMING ESTIMATES

In designing the IP core for the Kintex-7 xc7k410tffv900-2 FPGA device, we have set a target clock period (CP) of 10 nanoseconds (ns), or equally, a 100 MHz clock rate. The estimate performance numbers, including timing and latency, produced by the C synthesis and implementation via the Vivado HLS tool are presented in Table 5 and Table 6 for both non-optimized and optimized designs, indicating that the required timing is perfectly met in both cases. In particular, the estimated timing performance after post-implementation of the optimized IP core is shown to be 8.83 ns, which is well smaller than the target CP of the HDL design.

We further report, in Table 5 and Table 6, the latency and the initiation interval (II) estimates for the non-optimized and optimized HDL functions, where latency refers to the number of clock cycles required for the design to complete the current transaction and compute all the output values (i.e., the number of clock cycles between the input and the corresponding output), and the II is the number of clock cycles before the design can accept new input data. Comparing these two tables, we see that the non-optimized HDL design has a latency of 1493 clock cycles, whereas the optimized design can achieve a far smaller latency of 9 cycles. This significant improvement in throughput is brought by optimizing the code through, e.g., exploiting the parallelism between function and loop iterations. More precisely, the IP core has been optimized to complete a transaction in 9 cycles, which means that, upon receiving data on the s_V_TDATA port, the precoded vector is valid on the u_V_TDATA output port after 9 clock cycles. In the meantime, the IP core can accept a new input data per cycle and performs the next transactions in parallel to compute the corresponding output values. Hence, the design can produce an output every clock cycle, allowing the IP core to operate at a rate of 100 Mega symbols per second per user, as we will see in Section V.

In Table 7 and Table 8, we present the estimated resource utilization on the Kintex-7 xc7k410tffv900-2 FPGA device, where the IP core is generated for two systems with $(N_t, N_u) = (2, 2)$ and $(N_t, N_u) = (4, 4)$. When comparing the non-optimized and optimized HDL designs, it can be seen that the latter design occupies more resources on the FPGA device. This originates from the well-known trade-off between area and performance in digital logic circuit design. More specifically, parallelization of functions and loops leads to higher throughputs, but it requires more resources to perform many concurrent operations. Nonetheless, the resource utilization estimates in Table 8 shows that the optimized design's total resource occupation is well below the available resource on this particular FPGA part for 2×2 and 4×4 systems. For larger system sizes, i.e., larger numbers of transmit antennas and users, one should either make a compromise between area and performance or use a more expensive FPGA with more available resources.

On the other hand, according to the utilization estimates in Table 8, for the 2×2 system, the design utilizes around 4% of the DSP blocks, 1% of the FFs, and 1% of the total LUTs that are available at this specific FPGA part, while for the 4×4 system, around 17% of the DSP blocks, 2% of the FFs, and 22% of the total LUTs are utilized by the design. This implies that, in general, the resource utilization ratios may not be linearly related to the system size. Roughly speaking, based on the estimates, it might be possible to support a larger system than the current design with this particular FPGA part or even use a cheaper FPGA with less available resources. For example, in the former case, the design might be able to treat several independent carriers or handle larger systems on the same FPGA. Note, further, that the design's resource utilization does not depend on the constellation size (i.e., the modulation order). More precisely, having a larger constellation does not affect the design complexity but increases the size of the lookup table to form the matrix $\mathbf{A}$, as described in Section III. Therefore, the same resource occupation estimates are valid also for larger signal constellations.

C. DESIGN VALIDATION

In this subsection, we assess the performance accuracy of the designed IP core. For this purpose, we validate our design

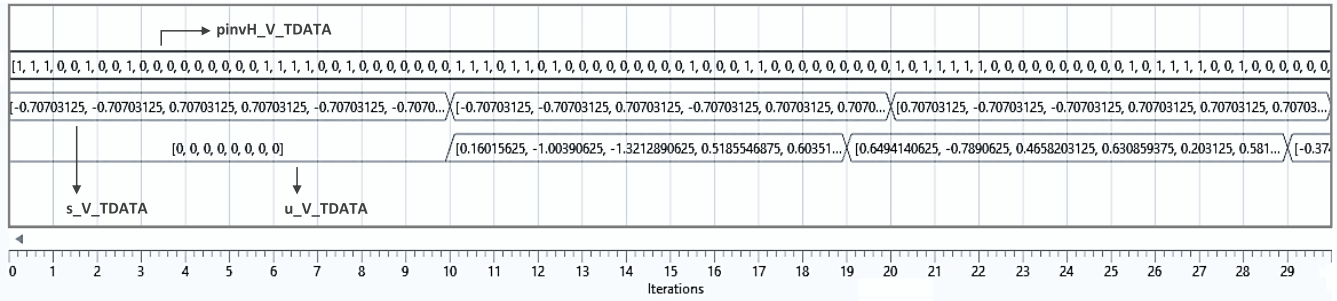

FIGURE 4. Data flow of the IP core sampled in the LabVIEW environment.

TABLE 7. Resource utilization of the non-optimized HDL design on the Xilinx Kintex-7 xc7k410tffv900-2 FPGA.

	2×2 system			4×4 system		
Resource	**DSP48E**	**FF**	**LUT**	**DSP48E**	**FF**	**LUT**
DSP	2	0	0	2	0	0
Expression	0	0	6038	0	0	11747
Instance	3	414	1522	3	6081	2248
Memory	0	26	4	0	410	37
Multiplexer	0	0	363	0	0	954
Register	0	1935	0	0	3670	0
Total	5	2375	7927	5	10161	14986
Available	1540	508400	254200	1540	508400	254200
Utilization (%)	0.3	0.5	3	0.3	1	5

TABLE 8. Resource utilization of the optimized HDL design on the Xilinx Kintex-7 xc7k410tffv900-2 FPGA.

	2×2 system			4×4 system		
Resource	**DSP48E**	**FF**	**LUT**	**DSP48E**	**FF**	**LUT**
DSP	68	0	0	72	0	0
Expression	0	0	1904	0	0	888
Instance	0	0	0	192	4224	54456
Memory	0	0	0	0	0	0
Multiplexer	0	0	62	0	0	62
Register	0	3590	768	0	8338	2560
Total	68	3590	2734	264	12562	57966
Available	1540	508400	254200	1540	508400	254200
Utilization (%)	4	0.7	1	17	2	22

using the LabVIEW software. The generated IP core is transformed to a design block and then imported to the LabVIEW environment. The validation steps, which are performed for a $(N_t, N_u) = (4, 4)$ system, are described in the following.

The input port pinvH_V is fed with the pseudo-inverse of the channel matrix given in (15), as shown at the bottom of the next page, and the symbol vector $\mathbf{s}$, taken from a normalized QPSK constellation set, is placed in order on the s_V_TDATA input port. We assume a unit noise variance and an equal target SNR of 0 dB for all the users. In the LabVIEW environment, we implement and run the imported IP core as a clock-driven logic (CDL) unit. The resulting flow of the data I/O ports is depicted in Fig. 4. According to the figure, it takes one iteration (clock cycle) for the IP core to read the data on input ports pinvH_V and s_V_TDATA. On the other hand, the IP core completes the current transaction after 9 cycles, and therefore, it generates the output data on the u_V_TDATA port after 10 cycles.

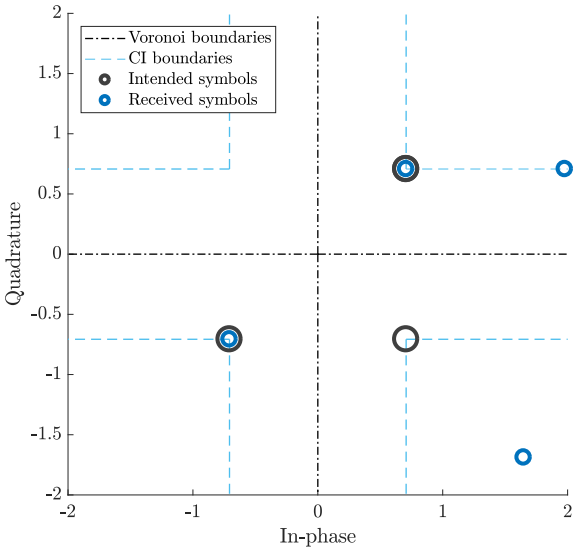

FIGURE 5. Intended symbols and noise-free received signals obtained by simulating the HDL design of Algorithm 1.

We particularly focus on the first transaction where the following symbol vector is placed on the data input port:

$$\text{s\_V\_TDATA} = \begin{bmatrix} -0.70703125 - \text{j}0.70703125 \\ 0.70703125 - \text{j}0.70703125 \\ 0.70703125 + \text{j}0.70703125 \\ 0.70703125 + \text{j}0.70703125 \end{bmatrix}. \tag{16}$$

The corresponding precoded vector generated on the output port u_V_TDATA of the IP core is

$$\text{u\_V\_TDATA} = \begin{bmatrix} 0.1601562500 + \text{j}0.6035156250 \\ -1.0039062500 - \text{j}1.6718750000 \\ -1.3212890625 + \text{j}3.1640625000 \\ 0.5185546875 + \text{j}1.0224609375 \end{bmatrix}. \tag{17}$$

This precoded vector is then passed through the multiuser channel $\mathbf{H}$, and eventually, the noise-free signals received by the users are plotted in Fig. 5. It can be seen that the received signal of each user is properly accommodated in the desired CI region. This verifies the accuracy of the designed IP core for implementation of the proposed low-complexity precoding solution.

V. NUMERICAL AND SIMULATION RESULTS

In this section, we provide some simulation results to assess the performance of the proposed low-complexity approximate SLP solution implemented as an IP core. We further compare the results with those obtained from the optimal SLP solution, the closed-form SLP solution in [22], and the ZF precoding technique. Note that the optimal SLP solution refers to the solution of problem (5). The precoding techniques of interest in this section are referred to as follows:

- ZF: zero-forcing precoding
- OPT-SLP: the optimal SLP solution of problem (5)
- CF-SLP: closed-form SLP solution in [22]
- HDL-CF-SLP: HDL implementation of Algorithm 1

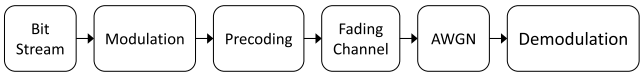

FIGURE 6. Block diagram of the simulated communication system.

The ZF, OPT-SLP and CF-SLP techniques are simulated using the MATLAB software, where a floating-point precision mode is considered by default. On the other hand, the HDL-CF-SLP technique is simulated in the LabVIEW environment, where the implementation uses fixed-point arithmetic as described in Subsection IV-A. As mentioned earlier in Section IV, to enable implementation of the HDL design in the LabVIEW environment, we have transformed the generated IP core into a design block using the Xilinx Vivado Design Suite tool, and then imported it as a CDL function into our LabVIEW simulation framework. The block diagram of the communication system, simulated in both MATLAB and LabVIEW environments, is shown in Fig. 6.

Our simulation setup is as follows. We consider a fully-loaded downlink multiuser MISO system with equal numbers of transmit antennas and users, i.e., $N_{\text{t}} = N_{\text{u}} = 4$. The BS uses QPSK signaling and an uncoded transmission scheme to communicate with the users. We assume a unit noise variance and equal target SNRs for all the users, i.e., $\sigma_i^2 = 1$ for all $i = 1, 2, \ldots, N_{\text{u}}$ and $\gamma_1 = \gamma_2 = \ldots = \gamma_{N_{\text{u}}}$. The presented plots in the following are obtained by averaging the results over 100 realizations of the Rayleigh block-fading channel matrix $\mathbf{H}$, where each realization consists of 100 symbols periods.

We show, in Fig. 7, the scatter plot of the users' noise-free and noisy received signals obtained from the HDL-CF-SLP technique for a target SNR of 0 dB. It can be seen that the users' noise-free received signals are properly located within the correct distance preserving CI region. As a result, the HDL implementation of our proposed approximate algorithm succeeded to satisfy the CI constraints of the SLP design problem. In the sequel, we evaluate the performance of our FPFA design in terms of average transmit power and symbol error rate.

In Fig. 8, we plot the average symbol error rate (SER) of each user versus the target SNR for different precoding

$$\mathbf{H}^{\dagger} = \begin{bmatrix} 0.2880 + \text{j}0.1221 & 0.1559 + \text{j}0.5371 & -0.8774 - \text{j}0.3437 & 0.1097 + \text{j}0.3331 \\ 0.3085 + \text{j}0.6187 & -0.7176 + \text{j}0.0683 & 0.8212 - \text{j}1.4356 & -0.6341 + \text{j}0.4036 \\ 0.1790 - \text{j}0.8406 & 0.6989 + \text{j}0.8182 & -2.2538 + \text{j}0.3444 & 0.4639 + \text{j}0.6017 \\ 0.0961 - \text{j}0.3560 & 0.3669 + \text{j}0.3207 & -1.0475 + \text{j}0.8989 & 0.6282 + \text{j}0.0833 \end{bmatrix}. \tag{15}$$

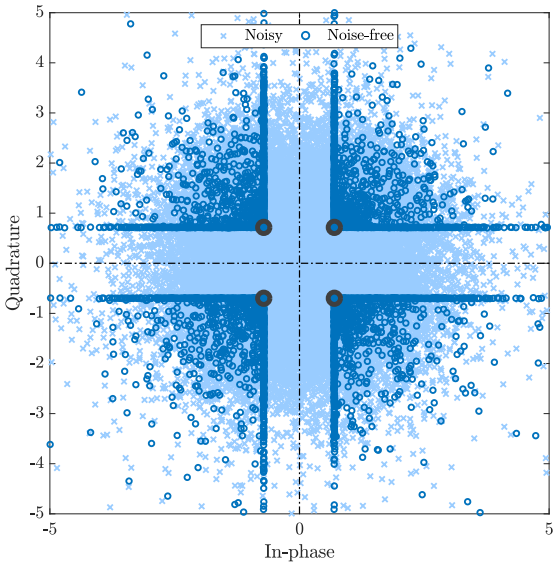

FIGURE 7. Scatter plot of the users' received signals at SNR = 0 dB.

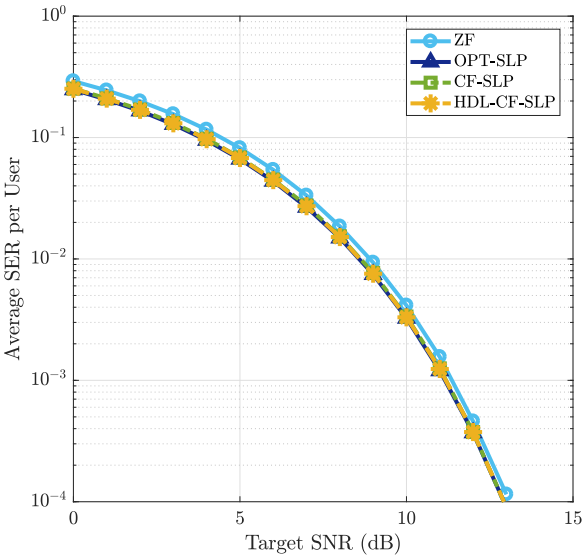

FIGURE 8. Average per-user symbol error rate as a function of target SNR with QPSK modulation and $N_t = N_u = 4$.

techniques of interest. It can be seen that all the techniques achieve almost the same SER performance, while the SLP techniques show slightly lower SER values compared to those of the ZF scheme. The reason for this lower SER is that the SLP techniques exploit the users' symbols to design the precoded vector such that it accommodates the noise-free received signal of each user in the distance-preserving CI region that corresponds to the user's intended symbol. Such a received signal has at least an equal or perhaps even an increased distance from the ML decision boundaries, which results in a higher accuracy for symbol detection at the user's receiver. It can further be seen from Fig. 8 that the FPGA simulation of the HDL-CF-SLP technique succeeds to achieve the same SER as that of the OPT-SLP. Therefore, the loss due to the approximate solution and the HDL implementation inaccuracies is not noticeable in terms of SER performance.

The average transmit power of each precoding technique corresponding to the SER performances in Fig. 8 is shown in Fig. 9 versus target SNR. All the SLP techniques, including the HDL-CF-SLP implementation, consume a lower power for precoded downlink transmission, compared to the ZF scheme. In particular, the HDL-CF-SLP implementation achieves 1.9 dBW gain in transmit power against the ZF technique. On the other hand, the FPGA simulation for the HDL-CF-SLP technique shows losses of 0.5 dBW and 0.85 dBW compared to the numerical results obtained for, respectively, the CF-SLP and the OPT-SLP techniques in the MATLAB environment. The loss compared to the CF-SLP technique originates from two facts. First, the HDL-CF-SLP implementation, which is based on Algorithm 1, uses the approximation (13) to avoid the pseudo-inverse calculation in the CF-SLP solution. Second, to design the HDL for Algorithm 1, we have used a fixed-point precision due to FPGA resource limitations which could be a source of inaccuracy in the values produced by the IP core, whereas simulating

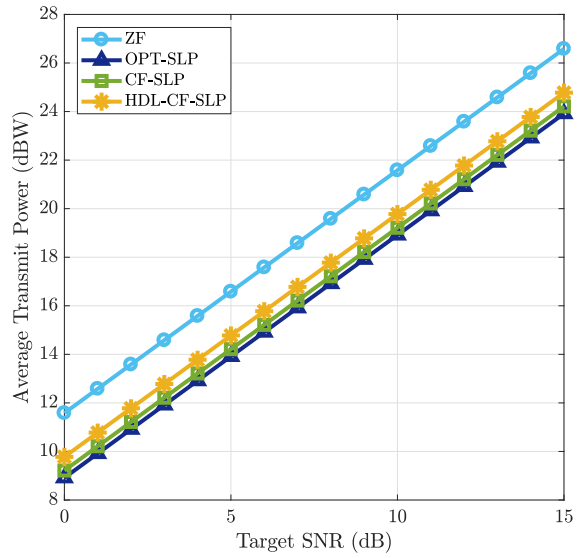

FIGURE 9. Average transmit power as a function of target SNR with QPSK modulation and $N_t = N_u = 4$.

the CF-SLP method via MATLAB uses floating-point arithmetic. However, one should notice that the HDL-CF-SLP implementation is designed for real-time applications on an FPGA and can provide a high throughput in practice, while the CF-SLP and the OPT-SLP techniques are not designed so. It should be further noted that the loss of the CF-SLP method compared to the OPT-SLP solution comes from the fact that the CF-SLP provides an approximate precoding solution in a two-step non-iterative way, while the OPT-SLP solution is obtained via an iterative optimization algorithm with a higher computational complexity.

Although all the precoding techniques of interest have shown comparable SER performances, they do not offer the same performance when it comes to the transmitted power.

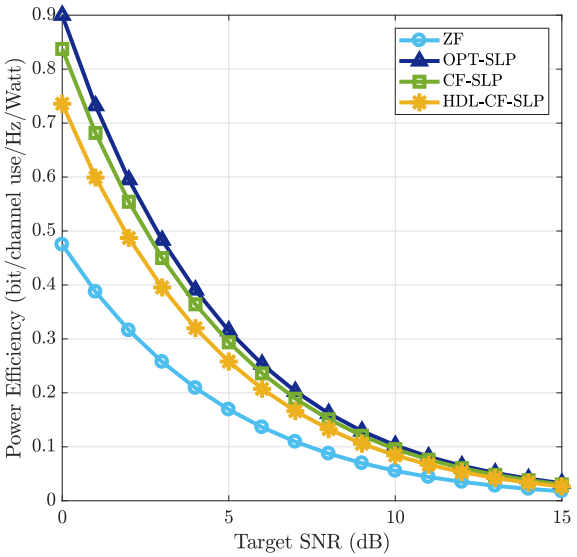

FIGURE 10. Power efficiency as a function of target SNR with QPSK modulation and $N_t = N_u = 4$.

In order to incorporate these two performance measures into a single figure of merit, we define power efficiency η as the ratio between the goodput and the transmit power, i.e.,

$$\eta \triangleq \frac{\log_2(M)(1 - \text{BER})}{\|\mathbf{u}\|^2}, \tag{18}$$

where M is the modulation order, $\|\mathbf{u}\|^2$ denotes the transmit power, and BER denotes the bit error rate which is simply obtained via dividing the SER by $\log_2(M)$.

We compare the power efficiencies of different precoding techniques in Fig. 10 as a function of target SNR. The HDL-CF-SLP implementation shows gains of up to 2 dB in power efficiency compared to the ZF scheme. When compared to the MATLAB implementation of SLP techniques, the OPT-SLP and the CF-SLP solutions outperform the HDL-CF-SLP implementation, but these techniques are not able to provide a high symbol throughput. In particular, the HDL implementation of Algorithm 1 shows at most 1 dB loss in the depicted range of target SNR, compared to the OPT-SLP technique. As mentioned earlier, this loss is due to the approximations used in deriving Algorithm 1 and also due to the adopted fixed-point precision. The latter drawback can be alleviated by increasing the bit width of the fixed-point format, but it comes with an excessive FPGA resource utilization. Furthermore, this performance loss is resulted in exchange for simplifying the design of the precoder. The simplified design enables implementation of the SLP algorithm on an actual FPGA. Our simulations in the LabVIEW environment indicate that the HDL design for Algorithm 1 allows data transmission with a high symbol throughput of 100 Mega symbols per second per user. In the considered system with $N_u = 4$ users and QPSK signaling, it translates to a sum-throughput of 800 Mbps which makes the proposed FPGA design suitable for realistic wireless communication applications.

VI. CONCLUSION AND FUTURE WORK

We developed an optimized FPGA design to enable low-complexity yet efficient implementation of SLP in a high-throughput downlink multiuser MISO system. The design is essentially based on [22] in which the authors proposed a sub-optimal closed-form solution to the power minimization SLP problem. In this work, we further simplified this solution by assuming mutually orthogonal channel vectors and proposed an approximate low-complexity design algorithm that can operate in a real-time mode. We analyzed the computational complexity of the proposed design and showed that it has the same per-symbol complexity order as that of the ZF precoding. We then used the Xilinx Vivado HLS tool to translate the design algorithm into an HDL code and also to optimize the design in order to achieve a low latency, and therefore, a higher throughput. The synthesis results, including performance, timing and resource utilization estimates verified the efficiency of our HDL design.

The generated IP core was evaluated in a simulation environment within the LabVIEW software. The simulations for a 4×4 system with QPSK signaling showed that the HDL design of our proposed algorithm is able to operate at a symbol rate of 100 Mega symbols per second per user when deployed on a specific Xilinx FPGA part, which makes it attractive for real-time implementations. Using the MATLAB software, we further evaluated the loss of our design algorithm with respect to the optimal SLP solution, where the loss is shown to be less than 1 dB according to our numerical results. This loss is mainly due to the approximation introduced when deriving the algorithm and also due to the adopted fixed-point arithmetic for the FPGA design. Furthermore, the simulation results indicated that the proposed HDL implementation of SLP outperforms the ZF scheme in terms of power efficiency, where an improvement of up to 50 percent can be achieved.

An interesting extension to this work could be to estimate the amount of power consumed by the FPGA, while running the IP block, and compare it with the saved power at the transmitter. Another future work is to further optimize the HDL code and seek possible improvements in the algorithm's accuracy. This will enable us to validate and implement the design for larger numbers of transmit antennas and users, for which the main restriction is the limited hardware resources. Another subsequent step is to conduct experimental validation of the proposed HDL design by deploying it on an actual FPGA.

ACKNOWLEDGMENT

The authors would also like to thank Dr. Farbod Kayhan and Dr. Jorge Querol Borras for facilitating this work through several discussions which helped us to improve this article.

REFERENCES

[1] M. Costa, "Writing on dirty paper (Corresp.)," *IEEE Trans. Inf. Theory*, vol. 29, no. 3, pp. 439–441, May 1983.
[2] T. K. Y. Lo, "Maximum ratio transmission," *IEEE Trans. Commun.*, vol. 47, no. 10, pp. 1458–1461, Oct. 1999.

[3] 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.
[4] A. Wiesel, Y. C. Eldar, and S. Shamai, "Zero-forcing precoding and generalized inverses," *IEEE Trans. Signal Process.*, vol. 56, no. 9, pp. 4409–4418, Sep. 2008.
[5] M. Joham, W. Utschick, and J. A. Nossek, "Linear transmit processing in MIMO communications systems," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2700–2712, Aug. 2005.
[6] M. Bengtsson and B. Ottersten, *Handbook of Antennas in Wireless Communications*. 2001.
[7] A. Gershman, N. Sidiropoulos, S. Shahbazpanahi, M. Bengtsson, and B. Ottersten, "Convex optimization-based beamforming," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 62–75, May 2010.
[8] 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.
[9] M. Schubert and H. Boche, "Solution of the multiuser downlink beamforming problem with individual SINR constraints," *IEEE Trans. Veh. Technol.*, vol. 53, no. 1, pp. 18–28, Jan. 2004.
[10] Y.-F. Liu, Y.-H. Dai, and Z.-Q. Luo, "Coordinated beamforming for MISO interference channel: Complexity analysis and efficient algorithms," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 1142–1157, Mar. 2011.
[11] A. Wiesel, Y. C. Eldar, and S. Shamai, "Linear precoding via conic optimization for fixed MIMO receivers," *IEEE Trans. Signal Process.*, vol. 54, no. 1, pp. 161–176, Jan. 2006.
[12] C. Masouros, "Correlation rotation linear precoding for MIMO broadcast communications," *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 252–262, Jan. 2011.
[13] C. Masouros and G. Zheng, "Exploiting known interference as green signal power for downlink beamforming optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3628–3640, Jul. 2015.
[14] 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.
[15] M. Alodeh, D. Spano, A. Kalantari, C. Tsinos, D. Christopoulos, S. Chatzinotas, and B. Ottersten, "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.
[16] A. Li, D. Spano, J. Krivochiza, S. Domouchtsidis, C. G. Tsinos, C. Masouros, S. Chatzinotas, Y. Li, B. Vucetic, and B. Ottersten, "A tutorial on interference exploitation via symbol-level precoding: Overview, State-of-the-Art and future directions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 796–839, 2nd Quart., 2020.
[17] M. Alodeh, S. Chatzinotas, and B. Ottersten, "Symbol-level multiuser MISO precoding for multi-level adaptive modulation," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5511–5524, Aug. 2017.
[18] A. Haqiqatnejad, F. Kayhan, and B. Ottersten, "Symbol-level precoding design based on distance preserving constructive interference regions," *IEEE Trans. Signal Process.*, vol. 66, no. 22, pp. 5817–5832, Nov. 2018.
[19] C. Masouros, M. Sellathurai, and T. Ratnarajah, "Vector perturbation based on symbol scaling for limited feedback MISO downlinks," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 562–571, Feb. 2014.
[20] 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, 2018.
[21] 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, pp. 1–6.
[22] 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.
[23] A. Haqiqatnejad, F. Kayhan, and B. Ottersten, "An approximate solution for symbol-level multiuser precoding using support recovery," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2019, pp. 1–5.
[24] 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.
[25] 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. Satell. Commun. Syst. Conf. (ICSSC)*, 2018, pp. 1–5.
[26] J. Krivochiza, J. Merlano Duncan, S. Andrenacci, S. Chatzinotas, and B. Ottersten, "FPGA acceleration for computationally efficient symbol-level precoding in multi-user multi-antenna communication systems," *IEEE Access*, vol. 7, pp. 15509–15520, 2019.
[27] A. Haqiqatnejad, F. Kayhan, and B. Ottersten, "Constructive interference for generic constellations," *IEEE Signal Process. Lett.*, vol. 25, no. 4, pp. 586–590, Apr. 2018.
[28] C. Lawson and R. Hanson, *Solving Least Squares Problems*. Philadelphia, PA, USA: SIAM, 1995.
[29] R. Bro and S. De Jong, "A fast non-negativity-constrained least squares algorithm," *J. Chemometrics*, vol. 11, no. 5, pp. 393–401, Sep. 1997.
[30] R. A. Polyak, "Projected gradient method for non-negative least square," *Contemp. Math.*, vol. 636, pp. 167–179, Mar. 2015.
[31] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate O $(1/k^2)$," *Dokl. Akad. Nauk SSSR*, vol. 269, pp. 543–547, Jan. 1983.
[32] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.

ALIREZA HAQIQATNEJAD (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from the Isfahan University of Technology, Iran, in 2012, and the M.Sc. degree in telecommunications engineering from the University of Isfahan, Iran, in 2015. He is currently pursuing the Ph.D. degree with the Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, Luxembourg, working on enhanced signal space design for multiuser MIMO interference channels with a focus on interference mitigation and multiuser precoding techniques. His research interests include signal processing and optimization for wireless communications and satellite communication systems.

JEVGENIJ KRIVOCHIZA (Member, IEEE) received the B.Sc. and M.Sc. degrees in electronic engineering in telecommunications physics and electronics from the Faculty of Physics, Vilnius University, in 2011 and 2013, respectively, and the Ph.D. degree in electrical engineering from the Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, in 2020. He is currently a Research Associate with the SNT, University of Luxembourg. His main research interests include coming from a development for FPGA silicon, software defined radios, digital signal processing, precoding, interference mitigation, DVB-S2X, DVB-S2, and LTE systems. He also works on DSP algorithms for SDR platforms for advanced precoding and beamforming techniques in the next-generation satellite communications.

JUAN CARLOS MERLANO DUNCAN (Senior Member, IEEE) received the Diploma degree in electrical engineering from the Universidad del Norte, Barranquilla, Colombia, in 2004, and the M.Sc. and Ph.D. Diploma *(cum laude)* 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 Institut National de la Recherche Scientifique, Montreal, QC, Canada, as a Research Assistant in the design and implementation of cognitive radio networks employing software development and FPGA programming. He joined the University of Luxembourg since 2016, where he currently works as a Research Scientist with the COMMLAB Laboratory working on SDR implementation of satellite and terrestrial communication systems. His research interests include wireless communications, remote sensing, distributed systems, frequency distribution and carrier synchronization systems, software-defined radios, and embedded systems.

SYMEON CHATZINOTAS (Senior Member, IEEE) 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 a Full Professor/Chief Scientist I and a Co-Head of the SIGCOM Research Group, SnT, University of Luxembourg. In the past, he has been a Visiting Professor with the University of Parma, Italy. He was involved in numerous Research and Development projects for the National Center for Scientific Research Demokritos, the Center of Research and Technology Hellas, and the Center of Communication Systems Research, University of Surrey. 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. He has (co-)authored more than 400 technical papers in refereed international journals, conferences and scientific books. He is currently in the Editorial Board of the IEEE Open Journal of Vehicular Technology and the *International Journal of Satellite Communications and Networking*.

BJÖRN OTTERSTEN (Fellow, IEEE) 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 1990. 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. He has been the Head of the Department for Signals, Sensors, and Systems, KTH, where he the Dean of the School of Electrical Engineering. He is currently the Director of the Interdisciplinary Centre for Security, Reliability, and Trust, University of Luxembourg. He is a Fellow of EURASIP. He was a recipient of the IEEE Signal Processing Society Technical Achievement Award, the EURASIP Group Technical Achievement Award, and the European Research Council advanced research grant twice. He has coauthored journal articles that received the IEEE Signal Processing Society Best Paper Award, in 1993, 2001, 2006, 2013, and 2019, and eight IEEE conference papers best paper awards. He has been a Board Member of IEEE Signal Processing Society, the Swedish Research Council, and also serves of the boards of EURASIP and the Swedish Foundation for Strategic Research. He has served as the Editor-in-Chief of *EURASIP Signal Processing*, and acted on the editorial boards of IEEE Transactions on Signal Processing, *IEEE Signal Processing Magazine*, IEEE Open Journal for Signal Processing, *EURASIP Journal on Advances in Signal Processing*, and *Foundations and Trends in Signal Processing*.