Finite-Alphabet MMSE Equalization for All-Digital Massive MU-MIMO mmWave Communication

Oscar Castañeda, *Graduate Student Member, IEEE*, Sven Jacobsson, *Graduate Student Member, IEEE*,
Giuseppe Durisi, *Senior Member, IEEE*, Tom Goldstein, *Member, IEEE*,
and Christoph Studer, *Senior Member, IEEE*

Abstract—We propose finite-alphabet equalization, a new paradigm that restricts the entries of the spatial equalization matrix to low-resolution numbers, enabling high-throughput, low-power, and low-cost hardware equalizers. To minimize the performance loss of this paradigm, we introduce FAME, short for finite-alphabet minimum mean-square error (MMSE) equalization, which is able to significantly outperform a naïve quantization of the linear MMSE matrix. We develop efficient algorithms to approximately solve the NP-hard FAME problem and showcase that near-optimal performance can be achieved with equalization coefficients quantized to only 1–3 bits for massive multi-user multiple-input multiple-output (MU-MIMO) millimeter-wave (mmWave) systems. We provide very-large scale integration (VLSI) results that demonstrate a reduction in equalization power and area by at least a factor of $3.9\times$ and $5.8\times$, respectively.

Index Terms—Millimeter wave (mmWave), massive multi-user MIMO, spatial equalization, minimum mean-square error (MMSE), quantization, hardware implementation.

I. INTRODUCTION

FUTURE wireless systems are expected to deliver even higher data-rates within the already crowded frequency spectrum. Emerging technologies, such as millimeter-wave (mmWave) communication [1], [2] and massive multi-user multiple-input multiple-output (MU-MIMO) [3], [4], have risen as promising candidates to provide such high data-rates. The abundance of available bandwidth at mmWave frequencies, combined with the fine-grained beamforming capabilities provided by massive MU-MIMO, enables high-throughput communication to multiple user equipments (UEs) in the same time-frequency resource. However, the presence of hundreds of antennas at the basestation (BS), each receiving a wideband signal, necessitates sophisticated radio frequency (RF) and digital baseband processing circuitry. As a result, circuit power consumption and system costs may increase significantly, which may hamper the success of this technology.

To reduce power consumption, the literature has largely focused on multi-antenna mmWave architectures that rely on hybrid analog-digital solutions [5]–[7]. Albeit energy efficient, such architectures have limited multiplexing capabilities as they are only capable of simultaneously combining signals coming from a restricted number of directions [7]–[10]; this key limitation may result in a reduced spectral efficiency. An emerging alternative is the use of all-digital BS architectures [11]–[13]. While it is commonly believed that all-digital BS designs would be energy inefficient, it has been shown recently that the power consumption of the RF and data-conversion elements in an all-digital BS is comparable to that of hybrid solutions, provided that the resolution of the data converters at the BS is suitably reduced [10], [12]. The power consumption and system costs of baseband processing for all-digital BS architectures is, however, largely unexplored.

A. The Case for Efficient Spatial Equalization

Spatial equalization in the uplink (UEs transmit to BS) is among the most power- and throughput-critical tasks in all-digital BS architectures. The purpose of spatial equalization is to collect the signals from all U UEs at the B BS antennas, while suppressing inter-UE interference. Mathematically, spatial equalization amounts to one or multiple $U \times B$ matrix-vector multiplications involving a $U \times B$ equalization matrix and the B-dimensional received vector. These multiplications need to be performed on a per-baseband-sample basis (at the sample rate of the analog-to-digital converters). Unfortunately, for a BS with $B = 256$ antennas serving $U = 16$ UEs, a conventional matrix-vector-product circuit

Oscar Castañeda and Christoph Studer are with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA, and also with Cornell Tech, New York, NY 10044 USA (e-mail: oc66@cornell.edu; studer@cornell.edu).

Sven Jacobsson is with Ericsson Research, 417 56 Gothenburg, Sweden, and also with the Department of Electrical Engineering, Chalmers University of Technology, 412 58 Gothenburg, Sweden (e-mail: sven.jacobsson@ericsson.com).

Giuseppe Durisi is with the Department of Electrical Engineering, Chalmers University of Technology, 412 58 Gothenburg, Sweden (e-mail: durisi@chalmers.se).

Tom Goldstein is with the Department of Computer Science, University of Maryland, College Park, MD 20742 USA (e-mail: tomg@cs.umd.edu).

A MATLAB simulator for the FAME approach proposed in this paper is available on GitHub: https://github.com/quantizedmassivemimo/fame

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2020.3000840

operating at $2\,\text{G}$ vectors/s consumes over $28\,\text{W}$ and occupies more than $128\,\text{mm}^2$ when implemented in $28\,\text{nm}$ CMOS (see Section V for the details). If more BS antennas and/or more UEs are considered, circuit power and area increase even further. Clearly, more efficient spatial equalization circuitry is necessary for all-digital BS architectures operating at mmWave frequencies in order to minimize power consumption and silicon area (which translate to system costs), while achieving high spectral efficiency.

The matrix-vector products required for spatial equalization involve multiplications and additions, where the hardware multipliers dominate power and area. The area and delay of a hardware multiplier scales with $O(mn)$ and $O(\log(\max\{m,n\}))$, respectively, where m and n are the number of bits of each operand [14]. As a consequence, circuit area, delay, and power consumption (which is roughly proportional to circuit area) of a matrix-vector-product engine can be minimized by using a low number of bits to represent both operands. The literature has extensively explored the efficacy of low-resolution data converters at the BS antennas of massive MU-MIMO systems [7], [10]–[12], [15]–[17]. Depending on the scenario, 3 to 8 bits were shown to achieve near-optimal spectral efficiency [10], [12], [15]–[17]. Such methods reduce the precision of one of the operands (i.e., that of the received vector) in a matrix-vector product. However, the coefficients of the equalization matrix (the other operand) are typically left at relatively high precision, e.g., 10 to 12 bits [18], [19].

B. Contributions

To reduce power consumption and implementation costs of spatial equalization, we propose to *coarsely* quantize the coefficients of spatial equalization matrices, a paradigm that we call *finite-alphabet equalization*. We emphasize that, in contrast to approaches that use low-resolution analog-to-digital converters (ADCs) to quantize the received vector to be equalized [7], [10]–[12], [15]–[17], finite-alphabet equalization intends to coarsely quantize the entries of the *spatial equalization matrix*. Although a straightforward concept, it turns out that obtaining low-resolution finite-alphabet equalization matrices that achieve high spectral efficiency is a hard problem. Figure 1 illustrates this claim for a case where the coefficients of a spatial equalization matrix are quantized using 1 bit per real and imaginary part. A naïve quantization of the linear minimum mean-square error (L-MMSE) equalization matrix to 1-bit leads to a finite-alphabet L-MMSE (FL-MMSE) equalizer, which, as we can see from Figure 1, suffers from a high error floor. To combat this problem, we propose *finite-alphabet minimum mean-square error equalization* (FAME), which leads to an NP-hard optimization problem that can be solved approximately (and efficiently) using forward-backward splitting (FBS). We refer to the resulting method as FAME-FBS. As shown in Figure 1, using FAME-FBS results in a substantially improved error rate compared to FL-MMSE equalization.

The main contributions of this paper are summarized as follows. We propose a specific finite-alphabet equalization-matrix structure that enables one to reduce the complexity of a
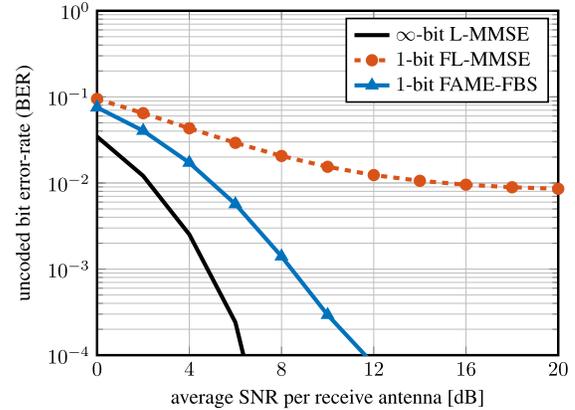

Fig. 1. Uncoded bit error-rate (BER) for a $B = 256$ BS antenna, $U = 16$ UE massive MU-MIMO system with 16-QAM in an i.i.d. Rayleigh-fading channel. The FAME-FBS algorithm proposed in this paper significantly outperforms the naïve FL-MMSE equalizer, which quantizes the real and imaginary parts of the conventional, high-resolution L-MMSE equalizer to 1 bit.

$U \times B$ matrix-vector product by using $U \times B$ low-resolution coefficients, while still being able to deliver a performance similar to conventional, high-resolution spatial equalization matrices. We derive the so-called FAME problem, whose solution leads to finite-alphabet equalization matrices that minimize the post-equalization mean-square error (MSE). We propose a range of algorithms that approximate the NP-hard FAME problem—some of these algorithms achieve excellent performance even for 1-bit resolution; some require very low complexity. We present simulation results for line-of-sight (LoS) and non-LoS mmWave channel models, which demonstrate the efficacy of FAME in terms of error-vector magnitude (EVM), beamforming capabilities, and uncoded bit error-rate (BER). Finally, we implement reference finite-alphabet equalization circuits for different numbers of bits in $28\,\text{nm}$ CMOS to demonstrate the effectiveness of FAME in practice.

C. Notation

Matrices and column vectors are represented with uppercase and lowercase boldface letters, respectively. The Hermitian transpose and the Frobenius norm of a matrix $\mathbf{A}$ are denoted by $\mathbf{A}^H$ and $\|\mathbf{A}\|_F$, respectively. The real part of a complex-valued matrix $\mathbf{A}$ is $\Re\{\mathbf{A}\}$ and the imaginary part is $\Im\{\mathbf{A}\}$. The $M \times M$ identity matrix is denoted by $\mathbf{I}_M$. The kth entry and the ℓ_2-norm of a vector $\mathbf{a}$ are a_k and $\|\mathbf{a}\|_2$, respectively; the entry-wise complex conjugate is denoted by $\mathbf{a}^*$. The kth standard basis vector is represented by $\mathbf{e}_k$. The signum function $\text{sgn}(\cdot)$ is defined as $\text{sgn}(a) = +1$ for $a \geq 0$ and $\text{sgn}(a) = -1$ for $a < 0$ and is applied entry-wise to vectors. We use $\mathbb{E}_\mathbf{x}[\cdot]$ to denote expectation with respect to the random vector $\mathbf{x}$. The set $\mathbb{S}_+$ contains all positive semidefinite matrices, and the set $\mathbb{R}_+$ contains all the non-negative real numbers.

D. Paper Outline

The rest of the paper is organized as follows. Section II introduces the system model and summarizes the basics of
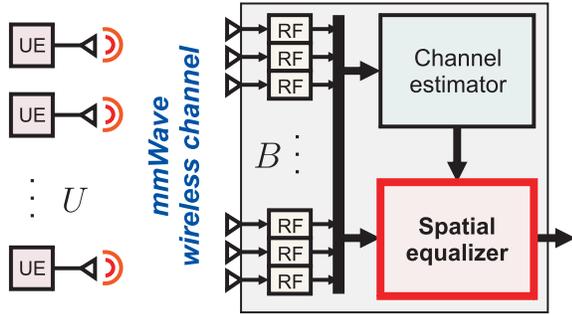
Fig. 2.   Uplink of a massive MU-MIMO mmWave wireless communication system. The U UEs transmit data in the same time-frequency resource to the B-antenna BS. After estimating the channel, the all-digital BS applies spatial equalization to collect the signals from the individual UEs and suppress inter-UE interference. In this work, we propose to use low-resolution spatial equalization matrices, an approach that we call finite-alphabet equalization.

L-MMSE equalization. Section III proposes the FAME problem and presents numerical experiments. Section IV develops low-complexity algorithms that approximate the NP-hard FAME problem. Section V shows hardware implementation results. We conclude in Section VI. Proofs and complexity counts are relegated to the appendices.

II. SYSTEM MODEL AND L-MMSE EQUALIZATION

We now introduce the system model considered in this work and briefly introduce the essentials of L-MMSE equalization.

A. System Model

We focus on the uplink of a massive MU-MIMO system with B BS antennas and $U \leq B$ single-antenna UEs, as illustrated in Figure 2. We consider the following narrowband input-output relation:

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}. \tag{1}$$

Here, $\mathbf{y} \in \mathbb{C}^B$ is the received signal vector at the BS, $\mathbf{H} \in \mathbb{C}^{B \times U}$ is the known uplink MIMO channel matrix, $\mathbf{s} \in \mathcal{S}^U$ is the transmit data vector, where $\mathcal{S}$ is the constellation set (e.g., 16-QAM), and $\mathbf{n} \in \mathbb{C}^B$ is i.i.d. circularly-symmetric complex Gaussian noise with covariance matrix $\mathbf{C_n} = \mathbb{E}_\mathbf{n}\big[\mathbf{n}\mathbf{n}^H\big] = N_0 \mathbf{I}_B$ per complex entry. In what follows, we assume that the transmit signals of the UEs, s_u, $u = 1, \ldots, U$, are i.i.d. zero mean with variance E_s so that $\mathbf{C_s} = \mathbb{E}_\mathbf{s}\big[\mathbf{s}\mathbf{s}^H\big] = E_s \mathbf{I}_U$.

Remark 1: The input-output relation (1) is not only valid to model narrowband transmission, but can also be used to model the subcarriers of a wideband massive MU-MIMO system that uses orthogonal frequency-division multiplexing (OFDM). The theory and algorithms developed in the remainder of the paper can be generalized for systems with inter-symbol interference—the details are left for future work.

Remark 2: We assume that the channel remains constant over multiple symbol transmissions and, hence, can be estimated. For our mathematical derivations, we assume (quantized) perfect channel state information at the BS. For systems in which the UEs use an antenna array to perform transmit

beamforming, the channel matrix $\mathbf{H}$ represents the joint effect of beamforming and the physical channel.

B. A Primer on L-MMSE Equalization

A key task of the BS is to form estimates $\hat{\mathbf{s}} \in \mathbb{C}^U$ of the transmitted data vector $\mathbf{s}$. To develop methods that are computationally efficient and hardware friendly, we focus on linear spatial estimators of the form $\hat{\mathbf{s}} = \mathbf{W}^H\mathbf{y}$, where $\mathbf{W}^H \in \mathbb{C}^{U \times B}$ is the L-MMSE equalization matrix that minimizes the post-equalization MSE defined as

$$MSE = \mathbb{E}_{\mathbf{s},\mathbf{n}}\big[\|\mathbf{W}^H\mathbf{y} - \mathbf{s}\|_2^2\big]. \tag{2}$$

Given the assumptions on the statistics of the transmit data and noise vectors, $\mathbf{s}$ and $\mathbf{n}$, we have that

$$MSE = E_s\|\mathbf{W}^H\mathbf{H} - \mathbf{I}_U\|_F^2 + N_0\|\mathbf{W}^H\|_F^2. \tag{3}$$

Hence, the L-MMSE equalization matrix can be obtained by solving the following matrix least-squares problem:

$$\mathbf{W}^H = \underset{\tilde{\mathbf{W}}^H \in \mathbb{C}^{U \times B}}{\arg\min} \|\mathbf{I}_U - \tilde{\mathbf{W}}^H\mathbf{H}\|_F^2 + \rho\|\tilde{\mathbf{W}}^H\|_F^2 \tag{4}$$

with regularization parameter $\rho = N_0/E_s$. This optimization problem has a well-known closed-form solution given by [20]

$$\mathbf{W}^H = (\rho\mathbf{I}_U + \mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H, \tag{5}$$

which can be computed efficiently in hardware [18].

We can alternatively compute the rows $\mathbf{w}_u^H$, $u = 1, \ldots, U$, of the L-MMSE equalization matrix $\mathbf{W}^H$ by decomposing (4) into U independent per-UE problems as follows:

$$\mathbf{w}_u = \underset{\tilde{\mathbf{w}} \in \mathbb{C}^B}{\arg\min} \|\mathbf{e}_u - \mathbf{H}^H\tilde{\mathbf{w}}\|_2^2 + \rho\|\tilde{\mathbf{w}}\|_2^2. \tag{6}$$

This alternative formulation of the L-MMSE equalizer will turn out useful in the next section.

III. FAME: FINITE-ALPHABET MMSE EQUALIZATION

We now propose the finite-alphabet equalization paradigm. We start by defining a finite-alphabet equalization matrix that enables efficient hardware for low-cardinality alphabets. We then formulate the FAME problem, which computes the finite-alphabet equalization matrix that minimizes the post-equalization MSE. Finally, we present a simple approach to compute finite-alphabet equalization matrices and compare its performance to the one of an equalizer that solves the FAME problem exactly.

A. Finite-Alphabet Equalization

Linear equalization in hardware requires the computation of an inner product $\hat{s}_u = \langle \mathbf{w}_u, \mathbf{y} \rangle = \mathbf{w}_u^H\mathbf{y}$ per UE for every received vector $\mathbf{y}$. As demonstrated in Section I-A, executing even such simple computations at the bandwidth offered by mmWave systems can result in excessively large area and high power consumption. To reduce both the area and power consumption, we propose to reduce the numerical precision of the equalization vectors $\mathbf{w}_u$, $u = 1, \ldots, U$. In the extreme case where the entries of $\mathbf{w}_u$ are quantized using 1-bit per real and imaginary component, an inner-product computation would

only require additions and subtractions; this is significantly less costly (in area and power) than using high-precision multipliers [14]. However, it is obvious that reducing the numerical precision of the equalization vectors $\mathbf{w}_u$ will affect the MSE and eventually the error-rate performance. Furthermore, quantization to, e.g., the finite alphabet $\mathcal{X} = \{+1 + j, +1 - j, -1 + j, -1 - j\}$, will result in numerical-range issues, meaning that such matrices will not be able to represent large or small entries. To mitigate both of these issues, we now develop a principled way to perform equalization with what we call *finite-alphabet matrices*. Concretely, we are interested in designing equalization matrices with the structure defined next.

Definition 1: We define a $U \times B$ finite-alphabet equalization matrix as follows:

$$\mathbf{V}^H = \mathrm{diag}(\boldsymbol{\beta}^*)\mathbf{X}^H. \tag{7}$$

Here, $\boldsymbol{\beta} \in \mathbb{C}^U$ is a vector that consists of post-equalization scaling factors and $\mathbf{X}^H \in \mathcal{X}^{U \times B}$ is an equalization matrix with entries chosen from the finite alphabet $\mathcal{X}$.

Remark 3: In this work, we are particularly interested in finite alphabets $\mathcal{X}$ of low cardinality and whose elements can be represented using a small number of bits. An example is the "1-bit" finite alphabet $\mathcal{X} = \{+1 + j, +1 - j, -1 + j, -1 - j\}$, which uses 1-bit per real and imaginary component.

With Definition 1, the equalized received symbol for the uth UE is given by

$$\hat{s}_u = \mathbf{v}_u^H \mathbf{y} = \beta_u^* \mathbf{x}_u^H \mathbf{y}, \tag{8}$$

where $\mathbf{v}_u^H \in \mathbb{C}^{1 \times B}$ and $\mathbf{x}_u^H \in \mathcal{X}^{1 \times B}$ are the uth rows of the matrices $\mathbf{V}^H$ and $\mathbf{X}^H$, respectively. It is now key to realize that spatial equalization as in (8) allows for efficient circuit implementations, especially for finite alphabets with low cardinality and regularly spaced elements. For such matrices, the inner product $\mathbf{x}_u^H \mathbf{y}$ can be implemented using low-resolution multipliers. As $\boldsymbol{\beta} \in \mathbb{C}^U$, the post-equalization scaling operation by the scalar factor β_u^* is performed using high-resolution multipliers. Nonetheless, this operation is executed only once per UE. In Section V, we show that equalizer implementations that leverage finite-alphabet equalization matrices enable significant area and power savings.

B. FAME: Finite-Alphabet MMSE Equalization

We now propose FAME, a principled method to compute MSE-optimal finite-alphabet equalization matrices. Analogous to the derivation of the L-MMSE equalizer in Section II-B, we are interested in minimizing the post-equalization MSE

$$FA\text{-}MSE = \mathbb{E}_{\mathbf{s},\mathbf{n}}\left[\|\mathbf{V}^H \mathbf{y} - \mathbf{s}\|_2^2\right], \tag{9}$$

which differs from the MSE in (2) as now $\mathbf{V}^H = \mathrm{diag}(\boldsymbol{\beta}^*)\mathbf{X}^H$ is a finite-alphabet equalization matrix as per Definition 1. The FAME matrix is the finite-alphabet equalization matrix that minimizes (9), i.e., it is the solution to the problem

$$\mathbf{V}^H = \underset{\tilde{\mathbf{X}} \in \mathcal{X}^{U \times B}, \tilde{\boldsymbol{\beta}} \in \mathbb{C}^U}{\arg\min} \|\mathbf{I}_U - \tilde{\mathbf{V}}^H \mathbf{H}\|_F^2 + \rho\|\tilde{\mathbf{V}}^H\|_F^2, \tag{10}$$

where $\tilde{\mathbf{V}}^H = \mathrm{diag}(\tilde{\boldsymbol{\beta}}^*)\tilde{\mathbf{X}}^H$. Clearly, the problem in (10) mirrors the one in (4). Hence, it follows from (6) that the rows

$\mathbf{v}_u^H = \beta_u^* \mathbf{x}_u^H$, $u = 1, \ldots, U$, of the desired FAME matrix can be computed by solving the following optimization problem:

$$\{\beta_u, \mathbf{x}_u\} = \underset{\tilde{\mathbf{x}} \in \mathcal{X}^B, \tilde{\beta} \in \mathbb{C}}{\arg\min} \|\mathbf{e}_u - \mathbf{H}^H \tilde{\beta}\tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\beta}\tilde{\mathbf{x}}\|_2^2. \tag{11}$$

Intuitively, we are interested in finding the finite-alphabet equalization vectors $\mathbf{v}_u^H = \beta_u^* \mathbf{x}_u^H$, $u = 1, \ldots, U$, that best mimic the infinite-precision L-MMSE equalizer.

Remark 4: For a fixed scaling factor β_u, the FAME problem in (11) is a closest vector problem, which is known to be NP-hard [21]–[23]. For example, for a system with 256 BS antennas using a 1-bit finite-alphabet equalization matrix, solving the FAME problem using an exhaustive search would require one to evaluate the objective function in (11) more than 10^{154} times for each UE. Clearly, without low-complexity algorithms, the FAME problem cannot be solved in practical massive MU-MIMO mmWave systems.

Since the FAME problem in (11) minimizes the cost function for two quantities at once, i.e., the scaling factor β_u and the low-resolution vector $\mathbf{x}_u$, it is not obvious how to solve it efficiently. To derive computationally efficient algorithms in Section IV, we will use the following equivalent form of the FAME problem in (11); the proof is given in Appendix A.

Lemma 1: The FAME problem in (11) is equivalent to solving the following optimization problem for each UE $u = 1, \ldots, U$:

$$\mathbf{x}_u = \underset{\tilde{\mathbf{x}} \in \mathcal{X}^B}{\arg\min} \frac{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2}{|\mathbf{h}_u^H \tilde{\mathbf{x}}|^2}, \tag{12}$$

where the associated optimal scaling factor is given by

$$\beta_u(\mathbf{x}_u) = \frac{\mathbf{x}_u^H \mathbf{h}_u}{\|\mathbf{H}^H \mathbf{x}_u\|_2^2 + \rho\|\mathbf{x}_u\|_2^2}. \tag{13}$$

This formulation of the FAME problem allows us to first find the optimal vector $\mathbf{x}_u$ using (12) and then compute the associated optimal scaling factor β_u using (13). Note that the equation in (13) represents the MSE-optimal scaling factor β_u for a given vector $\mathbf{x}_u$, regardless of how $\mathbf{x}_u$ was computed. Furthermore, the equivalent formulation in Lemma 1 is similar to a formulation proposed in [24] in the context of nonlinear quantized precoders.

C. FL-MMSE: A Baseline Finite-Alphabet Equalizer

Since the FAME problem is NP-hard, we now present a baseline method to compute finite-alphabet equalization matrices as per Definition 1 without having to solve the FAME problem in (12). We call our approach finite-alphabet L-MMSE (FL-MMSE), as it obtains the entries of the low-resolution matrix $\mathbf{X}^H$ by quantizing the L-MMSE equalizer in (5)—the corresponding scaling factors β_u, $u = 1, \ldots, U$, are then obtained via (13). In this work, we will use the FL-MMSE equalizer as a baseline to evaluate the performance of other, more sophisticated finite-alphabet equalizers that attempt to directly solve the FAME problem in (12).

For the 1-bit case, FL-MMSE applies the signum function $\mathrm{sgn}(\cdot)$ separately on the real and imaginary parts of the L-MMSE matrix $\mathbf{W}^H$ to obtain $\mathbf{X}^H$. Then, FL-MMSE uses the expression in (13) to compute the high-resolution scaling

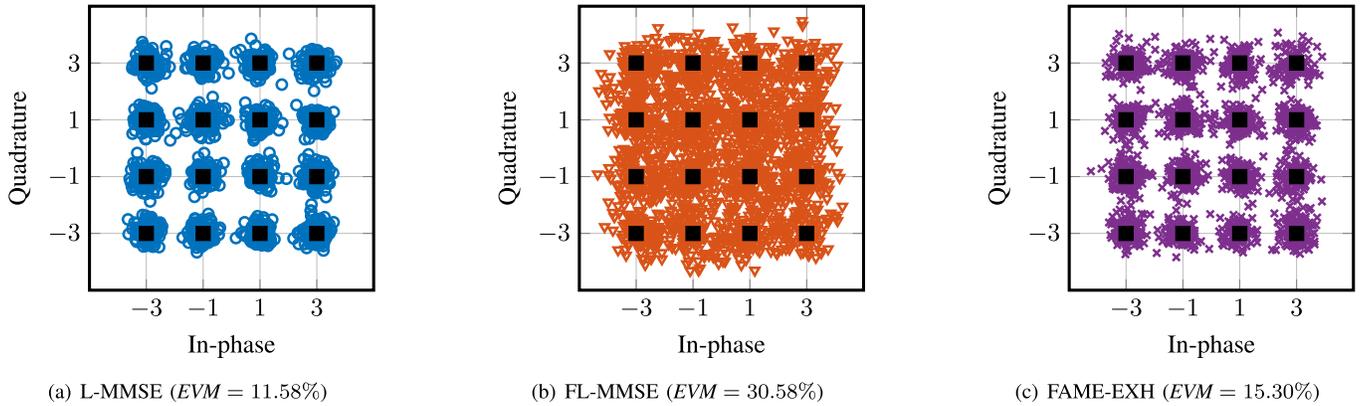(a) L-MMSE (*EVM* = 11.58%)   (b) FL-MMSE (*EVM* = 30.58%)   (c) FAME-EXH (*EVM* = 15.30%)

Fig. 3.   Error-vector magnitude (EVM) performance for different equalizers in a $B = 8$ BS antenna, $U = 2$ UE, 16-QAM system with i.i.d. Rayleigh-fading channels at 15 dB SNR. FL-MMSE corresponds to 1-bit quantization of the L-MMSE equalizer, which results in a significant EVM degradation. Solving the 1-bit FAME problem using an exhaustive search (FAME-EXH) yields an EVM comparable to that of infinite-precision L-MMSE equalization.

(a) L-MMSE (*SINR* = 21.02 dB)   (b) FL-MMSE (*SINR* = 6.72 dB)   (c) FAME-EXH (*SINR* = 21.02 dB)
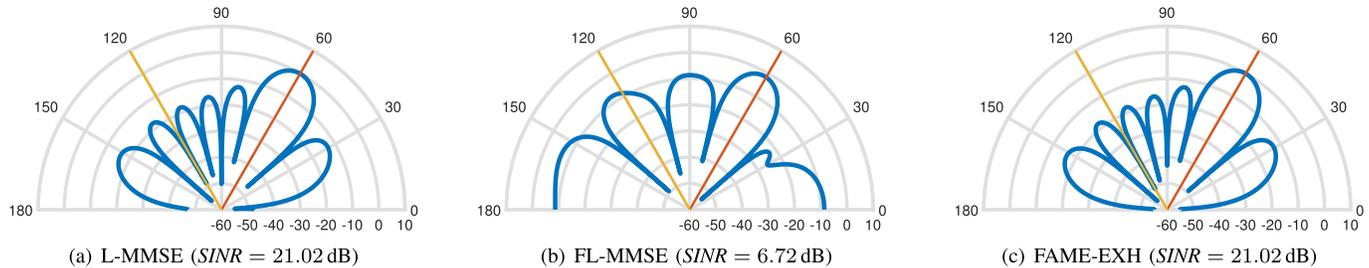
Fig. 4.   Beam- and null-forming capabilities for different equalizers in a $B = 8$ BS antenna, $U = 2$ UE system for textbook LoS channel at an SNR of 15 dB. The primary UE is located at $60°$ with respect to the ULA of antennas, whereas the secondary UE is at $120°$. We show the received signal power, in decibels, of the signals coming from different angles after being equalized for the primary UE at $60°$. A good equalizer gathers energy from this UE, while canceling interference from the UE at $120°$. FL-MMSE, which performs 1-bit quantization on the L-MMSE matrix, fails to reject the interference from the $120°$ UE. In contrast, 1-bit FAME-EXH achieves nearly identical beam- and null-forming performance to that of infinite-precision L-MMSE equalization.

factors in the vector $\boldsymbol{\beta}$. FL-MMSE can also be used with finite alphabets that have more than 1-bit per complex entry. In such cases, after computing the L-MMSE equalization matrix $\mathbf{W}^H$ in (5), we proceed to quantize its real and imaginary parts as follows. First, for each row $\mathbf{w}_u^H$ of $\mathbf{W}^H$, we identify the scalar $w_{\max}$ corresponding to the largest absolute value in $[\Re\{\mathbf{w}_u^H\}, \Im\{\mathbf{w}_u^H\}]$. Then, assuming that the targeted resolution is r bits, we divide the range $[-w_{\max}, +w_{\max}]$ into 2^r uniform-width bins and quantize the entries of $\Re\{\mathbf{w}_u^H\}$ and $\Im\{\mathbf{w}_u^H\}$ to the centroid values of these bins. For 2-bit resolution, for example, the centroid values of the bins are $\{-0.75, -0.25, +0.25, +0.75\}w_{\max}$. In hardware, one would scale these centroid values so that the minimum absolute value corresponds to 1. Following the previous example, one would use the values $\{-3, -1, +1, +3\}$ to represent the entries of $\Re\{\mathbf{x}_u^H\}$ and $\Im\{\mathbf{x}_u^H\}$. Note that this scaling does not affect the solution of the FAME problem in (12), as it is absorbed by the scaling factor β_u in (13). After obtaining the low-resolution vector $\mathbf{x}_u^H$, the corresponding scaling factor β_u is computed using the expression in (13).

D. EVM and Beamforming Performance of Exact FAME

We now assess the EVM and beamforming performance of the optimal FAME problem in (12) for a 1-bit finite alphabet. To solve the NP-hard problem, we resort to an

exhaustive search, which we call FAME-EXH. To keep the complexity within reasonable bounds, we simulate a small MU-MIMO system with $B = 8$ BS antennas. We also simulate the performance of conventional, infinite-precision L-MMSE equalization and 1-bit FL-MMSE equalization as proposed in Section III-C.

1) Error-Vector Magnitude: Figure 3 shows scatter plots of the equalization outputs $\hat{\mathbf{s}}$ of L-MMSE, FL-MMSE, and FAME-EXH equalization for 2 000 realizations in a $B = 8$ BS antenna, $U = 2$ UE system operating over an i.i.d. Rayleigh-fading channel at 15 dB SNR. While the infinite-precision L-MMSE equalizer achieves an EVM of 11.58%, quantizing its solution to 1-bit using FL-MMSE degrades the EVM to 30.58%, which blurs the decision regions of the considered 16-QAM constellation. In stark contrast, the 1-bit FAME-EXH equalizer achieves an EVM of only 15.30%, which is close to that of the infinite-precision L-MMSE equalizer; furthermore, the decision regions between constellation points are clearly visible. These results demonstrate the significant EVM advantage of solving the FAME problem over the simple FL-MMSE equalizer.

2) Beam- and Null-Forming Performance: Figure 4 illustrates the beam- and null-forming capabilities of FAME-EXH. For these plots, we consider a $B = 8$ BS antenna, $U = 2$ UE system operating at 15 dB SNR over a textbook LoS channel,

where the channel coefficient between the bth BS antenna and an UE located at an angle of ϕ is modeled as follows [25]:

$$h_b(\phi) = e^{-j\pi(b-1)\cos(\phi)}, \quad b = 1, \ldots, B. \tag{14}$$

Here, we assume a uniform linear array (ULA) of antennas with half-wavelength antenna spacing and constant path loss. We locate a primary UE at an angle of $\phi_1 = 60°$ and a secondary UE at $\phi_2 = 120°$. Next, we compute the corresponding equalization matrix using L-MMSE, FL-MMSE, and FAME-EXH equalization. We then evaluate how much the equalization vector $\mathbf{v}_1^H$ (which corresponds to the UE at $\phi_1 = 60°$) captures (or rejects) signals incoming from different incident angles by evaluating $|\mathbf{v}_1^H \mathbf{h}(\phi')|^2$ for $0 \leq \phi' \leq \pi$. The equalization vector $\mathbf{v}_1^H$ should amplify the signal from the primary UE at $\phi_1 = 60°$ but attenuate the signal from the secondary UE. The results shown in Figure 4 demonstrate that infinite-precision L-MMSE equalization is able to simultaneously beam-form towards the primary UE and null interference from the secondary UE. The 1-bit FL-MMSE equalizer is unable to reject interference from the secondary UE. In stark contrast, 1-bit FAME-EXH equalization is able to both beam-form towards the primary UE and null-form towards the secondary UE. We confirm these observations by computing the signal-to-interference-plus-noise ratio (SINR) for the primary UE.

Despite the significant performance advantages of 1-bit FAME-EXH over 1-bit FL-MMSE, solving 1-bit FAME-EXH for large-dimensional problems that arise in massive MU-MIMO mmWave systems is infeasible in practice. To this end, we next develop low-complexity FAME solvers that scale to large BS antenna arrays.

IV. FAST ALGORITHMS TO SOLVE FAME

We next present approximate algorithms to solve the FAME problem efficiently for systems with a large number of BS antennas. We start by proposing a semidefinite relaxation (SDR)-based method and then develop a much faster method that uses forward-backward splitting (FBS).

A. FAME With Semidefinite Relaxation (SDR)

We focus on using SDR [26] to solve the FAME problem in (12) for a 1-bit finite alphabet. To do so, we first re-express the FAME problem in the real domain using the quantities

$$\mathbf{x}_{\mathbb{R}} = \begin{bmatrix} \Re\{\mathbf{x}\} \\ \Im\{\mathbf{x}\} \end{bmatrix} \quad \text{and} \quad \mathbf{H}_{\mathbb{R}} = \begin{bmatrix} \Re\{\mathbf{H}\} & -\Im\{\mathbf{H}\} \\ \Im\{\mathbf{H}\} & \Re\{\mathbf{H}\} \end{bmatrix}. \tag{15}$$

Throughout, we will assume that $\Re\{\mathbf{x}\}$ and $\Im\{\mathbf{x}\}$ take values from the same alphabet $\mathcal{X}_{\mathbb{R}}$. For example, for 1-bit finite alphabets, we have $\mathcal{X}_{\mathbb{R}} = \{-1, +1\}$. We can now rewrite the FAME problem in (12) as

$$\mathbf{x}_{\mathbb{R},u} = \arg\min_{\tilde{\mathbf{x}}_{\mathbb{R}} \in \mathcal{X}_{\mathbb{R}}^{2B}} \frac{\|\mathbf{H}_{\mathbb{R}}^H \tilde{\mathbf{x}}_{\mathbb{R}}\|_2^2 + \rho\|\tilde{\mathbf{x}}_{\mathbb{R}}\|_2^2}{|\mathbf{h}_{\mathbb{R},u}^H \tilde{\mathbf{x}}_{\mathbb{R}}|^2}. \tag{16}$$

It is now key to realize that the vector $\tilde{\mathbf{x}}_{\mathbb{R}}$ can be scaled arbitrarily without changing the objective function of (16).

This observation enables us to state an equivalent optimization problem

$$\begin{cases} \underset{\bar{\mathbf{x}}_{\mathbb{R}} \in \mathcal{Z}_\alpha^{2B}, \alpha > 0}{\text{minimize}} & \|\mathbf{H}_{\mathbb{R}}^H \bar{\mathbf{x}}_{\mathbb{R}}\|_2^2 + \rho\|\bar{\mathbf{x}}_{\mathbb{R}}\|_2^2 \\ \text{subject to} & |\mathbf{h}_{\mathbb{R},u}^H \bar{\mathbf{x}}_{\mathbb{R}}|^2 = 1, \end{cases} \tag{17}$$

where the discrete set $\mathcal{Z}_\alpha$ is a scaled version of $\mathcal{X}_{\mathbb{R}}$; for 1-bit finite alphabets, we have $\mathcal{Z}_\alpha = \{-\alpha, +\alpha\}$ with $\alpha > 0$. This formulation enables us to formulate a semidefinite program to solve the FAME problem approximately.

By focusing on 1-bit finite alphabets, we can relax (17) by replacing the constraint $|\mathbf{h}_{\mathbb{R},u}^H \bar{\mathbf{x}}_{\mathbb{R}}|^2 = 1$ by $\mathbf{h}_{\mathbb{R},u}^H \bar{\mathbf{X}} \mathbf{h}_{\mathbb{R},u} = 1$, where the positive semidefinite matrix $\bar{\mathbf{X}} \in \mathbb{S}_+^{2B}$ should approximate $\bar{\mathbf{x}}_{\mathbb{R}} \bar{\mathbf{x}}_{\mathbb{R}}^H$. This SDR yields

$$\begin{cases} \underset{\bar{\mathbf{X}} \in \mathbb{S}_+^{2B}}{\text{minimize}} & \mathrm{tr}\left((\mathbf{H}_{\mathbb{R}} \mathbf{H}_{\mathbb{R}}^H + \rho\mathbf{I}_{2B})\bar{\mathbf{X}}\right) \\ \text{subject to} & \mathbf{h}_{\mathbb{R},u}^H \bar{\mathbf{X}} \mathbf{h}_{\mathbb{R},u} = 1 \\ & \bar{X}_{1,1} = \bar{X}_{b,b}, b = 2, \ldots, 2B, \end{cases} \tag{18}$$

where we ensure that the diagonal elements of $\bar{\mathbf{X}} \in \mathcal{S}_+^{2B}$ must be equal (but we do not specify their value). This constraint is a result of the fact that we are interested in a solution in the set $\mathcal{Z}_\alpha$, where the parameter α is not known. After solving the semidefinite program in (18), we compute the finite-alphabet vector by first extracting the leading eigenvector of the solution matrix $\bar{\mathbf{X}}$ followed by quantizing it to $\{-1, +1\}$ using the signum function $\mathrm{sgn}(\cdot)$. The equalization vector can then be scaled using the optimal FAME scaling parameter in (13). We refer to this procedure as FAME-SDR. A more detailed description of general SDR techniques can be found in [26].

Remark 5: While FAME-SDR can also be derived for multi-bit finite alphabets, e.g., using the techniques described in [27], we will not pursue this approach for the following reasons. As we will show in Section IV-C, the complexity of FAME-SDR does not scale well to a large number of BS antennas. Moreover, FAME-SDR cannot be applied to finite alphabets that are not separable into real and imaginary parts, such as a finite alphabet that contains the elements of an 8-PSK (phase shift keying) constellation. In addition, to the best of our knowledge, SDR can only handle finite alphabets with even cardinality that exclude a zero element. To avoid the drawbacks of SDR for FAME, we next present an alternative approach.

B. FAME With Forward-Backward Splitting (FBS)

Due to the high complexity of FAME-SDR and the fact that SDR solvers are notoriously difficult to implement in hardware [28], we next develop a low-complexity alternative for solving the FAME problem approximately. To do so, we start by assuming that, for each UE $u = 1, \ldots, U$, we know the optimal value of the objective in (12), which we denote with γ_u. Mathematically,

$$\gamma_u = \frac{\|\mathbf{H}^H \mathbf{x}_u\|_2^2 + \rho\|\mathbf{x}_u\|_2^2}{|\mathbf{h}_u^H \mathbf{x}_u|^2}, \tag{19}$$

where $\mathbf{x}_u$ is the solution to the problem in (12). Note that it follows from (19) that $\gamma_u > 1$. By rearranging (19), we obtain

$$0 = \|\mathbf{H}^H \mathbf{x}_u\|_2^2 + \rho\|\mathbf{x}_u\|_2^2 - \gamma_u|\mathbf{h}_u^H \mathbf{x}_u|^2. \tag{20}$$

Thus, if γ_u was known, solving the problem

$$\mathbf{x}_u = \arg\min_{\tilde{\mathbf{x}} \in \mathcal{X}^B} \frac{1}{2}\|\mathbf{H}^H\tilde{\mathbf{x}}\|_2^2 + \frac{\rho}{2}\|\tilde{\mathbf{x}}\|_2^2 - \frac{\gamma_u}{2}|\mathbf{h}_u^H\tilde{\mathbf{x}}|^2 \quad (21)$$

would yield the same solution as (12). As the optimal value of the objective in (12) is unknown in practice, we will use γ_u as an algorithm parameter that we tune to empirically improve the algorithm's performance.

Since the problem in (21) still contains a search over the finite alphabet $\mathcal{X}^B$, we relax the non-convex constraint $\tilde{\mathbf{x}} \in \mathcal{X}^B$ to $\tilde{\mathbf{x}} \in \mathcal{B}^B$. Here, $\mathcal{B}$ corresponds to the convex hull of the finite alphabet $\mathcal{X}$, which is defined as [29]

$$\mathcal{B} = \left\{ \sum_{i=1}^{|\mathcal{X}|} \alpha_i \bar{x}_i \mid (\alpha_i \in \mathbb{R}_+, \forall i) \wedge \sum_{i=1}^{|\mathcal{X}|} \alpha_i = 1 \right\}, \quad (22)$$

where $\bar{x}_i$ is the ith element of $\mathcal{X}$ and $i = 1, \ldots, |\mathcal{X}|$. After this relaxation step, the all-zeros vector $\mathbf{0}_{B\times 1}$ becomes a trivial solution. To prevent the algorithm from returning this trivial solution, we follow the approach put forward in [30] and include a term in (21) that encourages large entries in the vector $\tilde{\mathbf{x}}$. Specifically, we add $-\frac{\delta}{2}\|\tilde{\mathbf{x}}\|_2^2$ to the objective function, where $\delta > 0$ is a regularization parameter. The resulting optimization problem is given by

$$\mathbf{x}_u = \arg\min_{\tilde{\mathbf{x}} \in \mathcal{B}^B} \frac{1}{2}\|\mathbf{H}^H\tilde{\mathbf{x}}\|_2^2 - \frac{\gamma_u}{2}|\mathbf{h}_u^H\tilde{\mathbf{x}}|^2 + \frac{\rho-\delta}{2}\|\tilde{\mathbf{x}}\|_2^2. \quad (23)$$

To compute a solution to (23), we utilize FBS [31]–[33]. FBS is an efficient, iterative solver for convex optimization problems of the form

$$\hat{\mathbf{x}} = \arg\min_{\tilde{\mathbf{x}}} f(\tilde{\mathbf{x}}) + g(\tilde{\mathbf{x}}), \quad (24)$$

where both functions f and g are convex, but f is a smooth function and g is not necessarily smooth or bounded. FBS executes the following operations for $t = 1, 2, \ldots, t_{\max}$ iterations or until convergence [31], [33]:

$$\tilde{\mathbf{z}}^{(t+1)} = \tilde{\mathbf{x}}^{(t)} - \tau^{(t)}\nabla f(\tilde{\mathbf{x}}^{(t)}) \quad (25)$$

$$\tilde{\mathbf{x}}^{(t+1)} = \mathrm{prox}_g\left(\tilde{\mathbf{z}}^{(t+1)}; \tau^{(t)}\right). \quad (26)$$

Here, $\nabla f(\tilde{\mathbf{x}}^{(t)})$ is the gradient of the function f, $\{\tau^{(t)} > 0\}$ is a sequence of step sizes, and $\mathrm{prox}_g(\cdot)$ is the proximal operator of the function g, defined as [34]

$$\mathrm{prox}_g(\tilde{\mathbf{z}}; \tau) = \arg\min_{\tilde{\mathbf{x}}} \left\{ \tau g(\tilde{\mathbf{x}}) + \frac{1}{2}\|\tilde{\mathbf{x}} - \tilde{\mathbf{z}}\|^2 \right\}. \quad (27)$$

Our problem in (23) is not convex and hence, FBS is not guaranteed to converge to an optimal solution. Nevertheless, we can use FBS to find approximate solutions to (23) by setting

$$f(\tilde{\mathbf{x}}) = \frac{1}{2}\|\mathbf{H}^H\tilde{\mathbf{x}}\|_2^2 - \frac{\gamma_u}{2}|\mathbf{h}_u^H\tilde{\mathbf{x}}|^2 \quad (28)$$

$$g(\tilde{\mathbf{x}}) = \mathbf{1}_{\mathcal{B}^B}(\tilde{\mathbf{x}}) + \frac{\rho-\delta}{2}\|\tilde{\mathbf{x}}\|_2^2. \quad (29)$$

Here, the convex constraint $\tilde{\mathbf{x}} \in \mathcal{B}^B$ in (23) is incorporated into the function $g(\tilde{\mathbf{x}})$ via the indicator function $\mathbf{1}_{\mathcal{B}^B}(\tilde{\mathbf{x}})$, which is

zero if $\tilde{\mathbf{x}} \in \mathcal{B}^B$ and infinity otherwise. With these definitions, we arrive at

$$\nabla f(\tilde{\mathbf{x}}) = \mathbf{HH}^H\tilde{\mathbf{x}} - \gamma_u \mathbf{h}_u \mathbf{h}_u^H\tilde{\mathbf{x}} \quad (30)$$

$$\mathrm{prox}_g(\tilde{z}) = \mathrm{sgn}(\Re\{\tilde{z}\}) \min\left\{\nu^{(t)}|\Re\{\tilde{z}\}|, 1\right\}$$
$$+ j\,\mathrm{sgn}(\Im\{\tilde{z}\}) \min\left\{\nu^{(t)}|\Im\{\tilde{z}\}|, 1\right\}, \quad (31)$$

where $\nu^{(t)} = (1 + \tau^{(t)}(\rho - \delta))^{-1}$ and (31) is applied element-wise to the vector $\tilde{\mathbf{z}}$.

Note that we have introduced three sets of algorithm parameters: $\{\tau^{(t)}\}$, $\{\nu^{(t)}\}$, and $\{\gamma_u\}$, where $t = 1, \ldots, t_{\max}$ and $u = 1, \ldots, U$. While one could manually tune these parameters via numerical simulations, we automate the tuning process by using a neural-network-based approach as put forward in [35]. We note that such a neural-network-based approach is only used for determining the algorithm parameters, i.e., it is trained offline without affecting the runtime complexity of FAME-FBS. As the same algorithm parameters should work across several channel realizations, having a per-UE parameter such as $\{\gamma_u\}$ is meaningless. As a result, we set $\gamma = \gamma_u$ for $u = 1, \ldots, U$. Furthermore, to provide the neural network with greater flexibility during optimization, we allow γ to be different in each iteration; i.e., we introduce another set of per-iteration parameters $\{\gamma^{(t)}\}$, $t = 1, \ldots, t_{\max}$. We call the resulting algorithm FAME-FBS, which is summarized next.

Algorithm 1 (FAME-FBS): Initialize $\tilde{\mathbf{x}}^{(1)}$ with either the maximum-ratio combining (MRC) solution $\mathbf{h}_u$ or the low-resolution vector $\mathbf{x}_u$ computed by FL-MMSE, and fix the sets of parameters $\{\tau^{(t)}\}$, $\{\nu^{(t)}\}$, and $\{\gamma^{(t)}\}$. Then, for every iteration $t = 1, 2, \ldots, t_{max}$, compute

$$\tilde{\mathbf{z}}^{(t+1)} = \left(\mathbf{I}_B - \tau^{(t)}\mathbf{H}(\mathbf{I}_U - \gamma^{(t)}\mathbf{e}_u\mathbf{e}_u^H)\mathbf{H}^H\right)\tilde{\mathbf{x}}^{(t)} \quad (32)$$

$$\tilde{\mathbf{x}}^{(t+1)} = \mathrm{prox}_g(\tilde{\mathbf{z}}^{(t+1)}). \quad (33)$$

Here, the proximal operator $\mathrm{prox}_g(\cdot)$ is the element-wise function given by (31). The result $\tilde{\mathbf{x}}^{(t_{max}+1)}$ is quantized to the finite alphabet $\mathcal{X}$ to obtain $\mathbf{x}_u^H$. Then, the optimal FAME scaling parameter β_u is computed using (13).

Remark 6: FAME-FBS supports multi-bit finite-alphabet equalization matrices. This is achieved by uniformly quantizing, in the range $[-1, +1]$, the real and imaginary parts of the solution vector $\tilde{\mathbf{x}}^{(t_{max}+1)}$, similar to what is done by FL-MMSE equalization. As a consequence, unlike FAME-SDR, FAME-FBS can operate with finite alphabets that contain (i) an odd number of elements and (ii) a zero element. Furthermore, FAME-FBS (and FL-MMSE) can be applied to PSK-like finite alphabets following an approach related to the one used in [36].

Remark 7: Since FAME-FBS was obtained by relaxing the original optimization problem and by applying an iterative solver for convex optimization problems to the non-convex problem in (23), it is not guaranteed to converge to the optimal solution of the original problem in (12). Nonetheless, our simulation results in Section IV-D confirm that FAME-FBS achieves

TABLE I
COMPLEXITY FOR COMPUTING AN EQUALIZATION MATRIX

Algorithm	Computational complexity	Asymptotic scaling
L-MMSE	$2U^3 + 6BU^2 - 2BU - 2U + 1$	$O(BU^2)$
FL-MMSE	$10BU^2 + 2U^3 + 2U^2 + U + 1$	$O(BU^2)$
FAME-SDR	n.a.	$O(B^{4.5})$
FAME-FBS	$(8t_{max} + 4)BU^2 + 2U^2$ $+ (4t_{max} + 2)BU + (2t_{max} + 3)U$	$O(BU^2)$

TABLE II
COMPLEXITY OF FINITE-ALPHABET EQUALIZATION

Equalization	Real-valued multiplication count	
	High resolution	Low resolution
Traditional	$4BU$	0
Finite alphabet	$4U$	$4BU$

competitive performance for (i) different system configurations and (ii) realistic channel models.

C. Computational Complexity

We now assess the complexity of (i) computing the equalization matrix and (ii) performing equalization on a received vector $\mathbf{y}$, for high-resolution and finite-alphabet equalization approaches. We measure computational complexity as the number of real-valued multiplications performed by an algorithm.[1]

1) Computing the Equalization Matrix: Table I lists the computational complexity for computing a single equalization matrix using L-MMSE, FL-MMSE, FAME-SDR, and FAME-FBS. For the infinite-precision L-MMSE equalizer, the complexity corresponds to explicitly computing the equalization matrix $\mathbf{W}^H$. For the finite-alphabet equalizers (FL-MMSE and FAME-based algorithms), the complexity corresponds to the computation of the low-resolution matrix $\mathbf{X}^H$ and the scaling factors in the vector $\boldsymbol{\beta}$. Solving FAME-SDR results in the highest complexity, which asymptotically scales as $O(B^{4.5})$ unless specific problem structures can be exploited [26]. Since we do not have access to a particular SDR solver, we only provide this asymptotic scaling. Evidently, FAME-SDR does not scale well to systems with a large number of BS antennas. FAME-FBS has the same asymptotic scaling of $O(BU^2)$ as L-MMSE and FL-MMSE equalization, making it suitable for massive MU-MIMO mmWave systems. The exact complexity counts listed in Table I are derived in Appendix B.

Remark 8: While the constant associated with the term BU^2 is larger for FAME-FBS than for L-MMSE and FL-MMSE, the complexity of the latter algorithms appears to be higher in practice. Computing the L-MMSE (and the FL-MMSE) equalizer in hardware requires square roots and divisions, which result in high numerical precision requirements [18]. Furthermore, the Cholesky decomposition and forward- and back-substitution procedures required when computing the L-MMSE (and the FL-MMSE) equalization matrix result in stringent data dependencies that limit parallelism and, hence, reduce throughput. In contrast, FAME-FBS has a regular structure with few data dependencies and the matrix-vector multiplications can be parallelized easily. In addition, one can parallelize computation per UE as the FAME problem in (12) is independent for each $u = 1, \ldots, U$.

---

[1]For the remainder of the paper, we assume that complex-valued multiplications require four real-valued multiplications.

In fact, a simple hardware engine, similar to the one proposed in [37] for another FBS-based algorithm, could be used to efficiently execute FAME-FBS to determine the low-resolution equalization vectors $\mathbf{x}_u^H$.

2) Performing Equalization: After computing the equalization matrix, one must perform spatial equalization on the received signal vectors $\mathbf{y}$ at the rate of the ADCs. For the infinite-precision L-MMSE equalizer, this corresponds to computing one high-resolution matrix-vector product $\hat{\mathbf{s}} = \mathbf{W}^H \mathbf{y}$ per received vector. For finite-alphabet equalizers, this corresponds to a low-resolution matrix-vector product $\mathbf{z} = \mathbf{X}^H \mathbf{y}$, followed by U high-resolution products $\hat{s}_u = \beta_u^* z_u$, $u = 1, \ldots, U$. The complexity of equalization is summarized in Table II, where we distinguish between high resolution and low resolution multiplications. While finite-alphabet equalization performs more multiplications than a conventional equalizer, most of these multiplications are performed at low resolution. Thus, for sufficiently low resolution, finite-alphabet equalization effectively reduces the complexity of spatial equalization.

Remark 9: While spatial equalization must be carried out at symbol rate, the computation of the equalization matrix must only be carried out if the channel matrix changes. Given that we are considering operation at extremely high bandwidths, the complexity of performing equalization will dominate in most mmWave systems. For scenarios with short coherence times, methods that minimize the complexity of computing the equalization matrix are to be preferred.

D. Simulation Results

We now evaluate the uncoded BER performance of FAME-based algorithms, and compare it to infinite-precision L-MMSE equalization and FL-MMSE. The following simulation results are obtained by carrying out 10^4 Monte-Carlo trials. The per-iteration parameters $\{\gamma^{(t)}\}$, $\{\tau^{(t)}\}$, and $\{\nu^{(t)}\}$ of FAME-FBS are tuned using a neural network as in [35]; the neural network is trained using 10^4 channel realizations, which differ from the ones used to evaluate the BER. In practice, we have observed that $\gamma^{(t)} = 1.1$ and $\nu^{(t)} = 1.1$, $t = 1, \ldots, t_{max}$, constitute good values for initializing the neural network regardless of the system configuration; good initializers for $\tau^{(t)}$ vary from 2^{-9} to 2^{-4} for the systems considered in this work. For all equalizers, we quantize the entries of the channel matrices $\mathbf{H}$ to 8 bits per real and imaginary components. In addition, the received signal vectors are quantized to 7 bits per real and imaginary components, which is sufficient to achieve virtually the same performance as with double-precision representation.

(a) $B = 8$ BS antennas, $U = 2$ UEs, QPSK

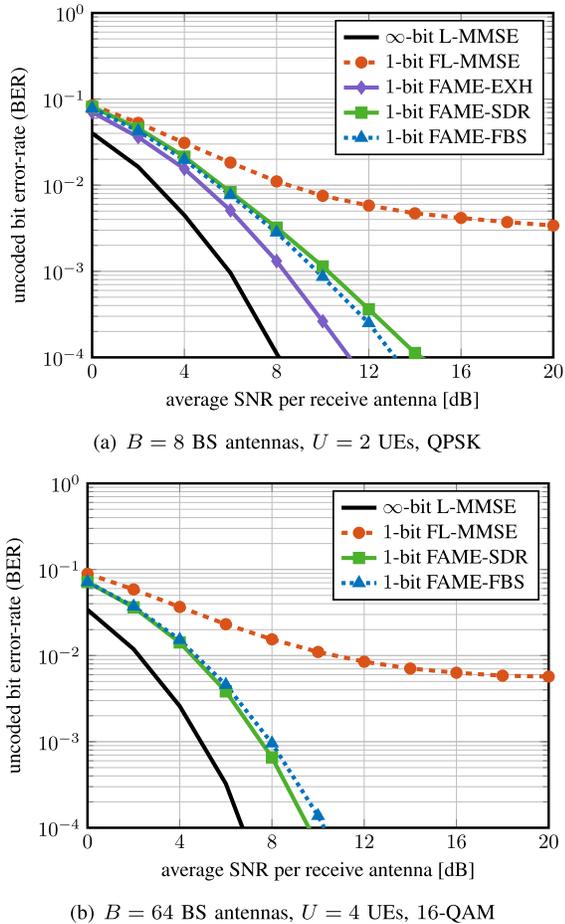(b) $B = 64$ BS antennas, $U = 4$ UEs, 16-QAM

Fig. 5. Uncoded bit error-rate (BER) for two MU-MIMO systems in an i.i.d. Rayleigh-fading scenario. All equalizers, except for L-MMSE, use 1-bit finite-alphabet equalization matrices. (a) FAME-SDR approaches the performance of an exhaustive search, while being scalable for systems with more BS antennas. (b) FAME-FBS with $t_{max} = 30$ iterations achieves similar performance as FAME-SDR but at a significantly lower complexity.

In Figure 5, we show uncoded BER for two different MU-MIMO systems in an i.i.d. Rayleigh-fading scenario. For the $B = 8$ BS antenna, $U = 2$ UE system shown in Figure 5(a), we see that 1-bit FAME-EXH significantly outperforms 1-bit FL-MMSE, which suffers from an error floor. Since the complexity of FAME-EXH scales exponentially in B, it cannot be used for significantly larger systems. Hence, we also show the performance of FAME-SDR, which has a 2 dB loss at a BER of 10^{-3} compared to FAME-EXH. Since FAME-SDR scales to systems with more BS antennas, we also show a $B = 64$ BS antenna, $U = 4$ UE system in Figure 5(b). In this scenario, FAME-SDR continues to substantially outperform 1-bit FL-MMSE. As discussed in Section IV-C, however, FAME-SDR does not scale to systems with more BS antennas, whereas FAME-FBS exhibits the same asymptotic complexity scaling as L-MMSE and FL-MMSE equalization. From Figure 5, we see that FAME-FBS performs on par with FAME-SDR but at much lower complexity.

In Figure 6(a), we show the BER performance of finite-alphabet equalization matrices in a $B = 256$ BS

antenna, $U = 16$ UE system using 16-QAM with i.i.d. Rayleigh-fading channels. The performance behavior of 1-bit FL-MMSE and 1-bit FAME-FBS is similar to what we have observed for smaller systems. The same figure also shows the performance of finite-alphabet matrices with resolutions larger than 1 bit. We see that the performance gap between FAME-FBS and FL-MMSE is more pronounced for 1-bit and 2-bit finite-alphabet equalization matrices than for 3-bit. We note that finite-alphabet equalizers achieve virtually the same performance as infinite-precision L-MMSE equalization when using 6 bits; nonetheless, 3 bits are sufficient to operate at SNRs lower than 4 dB.

Since i.i.d. Rayleigh-fading channels are a poor model for mmWave propagation conditions, we also show the performance of FAME-FBS in a $B = 256$ BS antenna, $U = 16$ UE system operating over more realistic mmWave channels generated using the QuaDRiGa model [39]. Concretely, in Figure 6(b) and Figure 6(c), we simulate mmWave systems with a carrier frequency of 60 GHz within the "mmMAGIC_UMi" scenario. We consider both non-LoS (shown in Figure 6(b)) and LoS (shown in Figure 6(c)) propagation conditions. We also model power control by scaling the QuaDRiGa-generated channels so that the received UE powers are in the range ±3 dB. Specifically, for each channel realization, the UE with highest power has 4× the power of the UE with the lowest power. Furthermore, the UEs are randomly placed in a sector of $120°$ in front of the BS antenna array with a distance ranging from 10 m to 110 m, and a minimum angular separation of $4°$. From Figure 6(b) and Figure 6(c), we observe that FAME-FBS outperforms FL-MMSE for both non-LoS and LoS channels—essentially the same trends as for Rayleigh-fading channels. These simulation results indicate that finite-alphabet equalization performs well under more realistic mmWave propagation conditions, while having the potential to significantly reduce power consumption and silicon area.

To further evaluate the performance of finite-alphabet equalizers, Figure 7 shows the EVM performance of FAME-FBS and FL-MMSE for the same system configuration and propagation conditions considered in Figure 6. The gray dashed lines indicate the EVM requirements for different modulation schemes as specified by the 3GPP 5G NR standard [38]. Figure 7 confirms the trends observed in the BER simulations. For example, while 1-bit FL-MMSE is not able to meet the EVM requirement for QPSK in Figure 7(a) and Figure 7(b), 1-bit FAME-FBS is almost able to reach the EVM requirement for 64-QAM. Moreover, while FAME-FBS significantly outperforms FL-MMSE when using 1 and 2 bits of resolution, their EVM performance is similar for 3 bits, in which case the performance of both finite-alphabet equalizers is close to that of infinite-precision L-MMSE.

Remark 10: FL-MMSE and FAME-based algorithms (FAME-EXH, FAME-SDR, and FAME-FBS) generate finite-alphabet equalization matrices as in Definition 1. This implies that, for a fixed equalizer resolution, all the algorithms proposed in this paper produce a low-resolution matrix $\mathbf{X}^H$ whose entries belong to the same finite alphabet $\mathcal{X}$, as well as a set of post-equalization scaling factors $\boldsymbol{\beta}$, which are
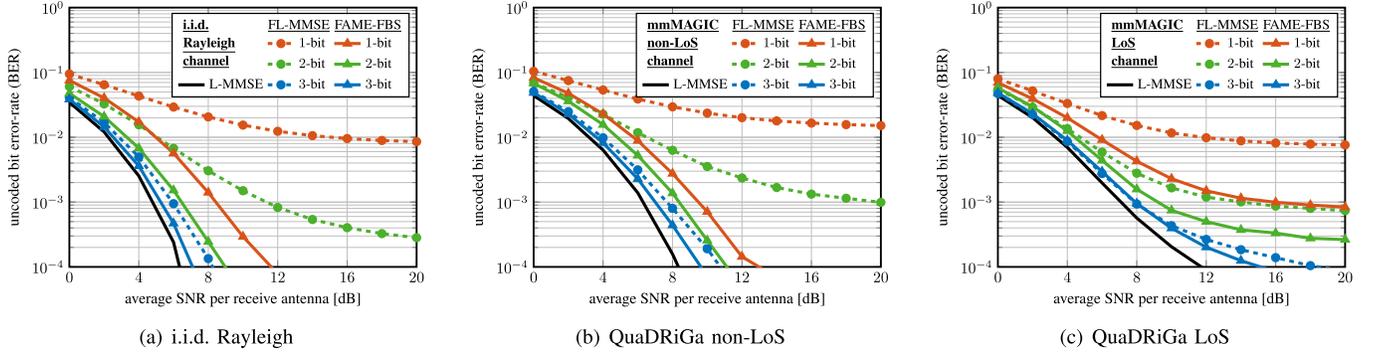
Fig. 6. Uncoded bit error-rate (BER) for a $B = 256$ BS antenna, $U = 16$ UE, 16-QAM system under different channel models. FAME-FBS runs $t_{max} = 5$ iterations for the i.i.d. Rayleigh channel and $t_{max} = 20$ iterations for both QuaDRiGa channels. FAME-FBS is initialized with the MRC equalizer $\mathbf{H}^H$ for all cases, except for the 3-bit QuaDRiGa LoS scenario, where FAME-FBS is initialized with the low-resolution matrix of FL-MMSE and runs for $t_{max} = 3$ iterations. FAME-FBS outperforms FL-MMSE in all considered scenarios. The performance of finite-alphabet equalizers meets that of L-MMSE with 6 bits.
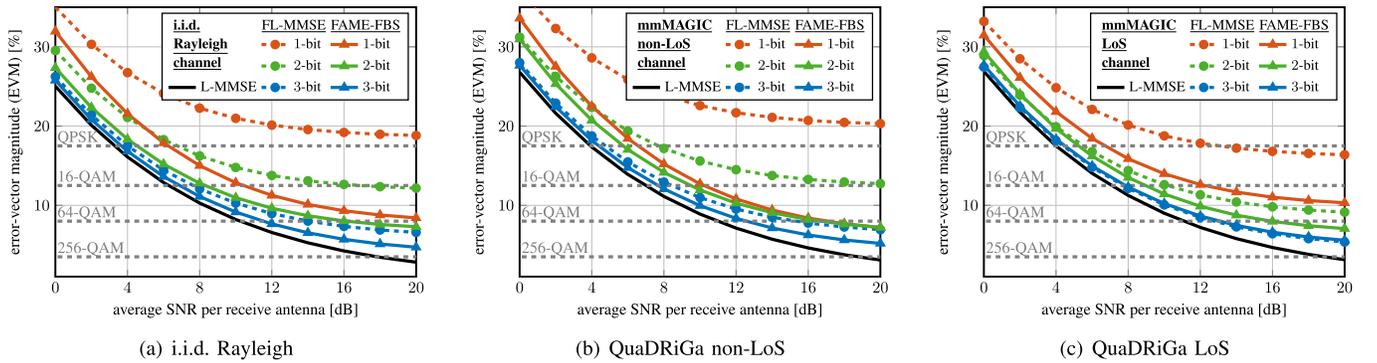

Fig. 7. Error-vector magnitude (EVM) for a $B = 256$ BS antenna, $U = 16$ UE system under different channel models. The gray dashed lines indicate the EVM requirements established by the 3GPP 5G NR technical specification [38]. The details for FAME-FBS are given in Figure 6. FAME-FBS significantly outperforms FL-MMSE in all considered scenarios when using 1- and 2-bit finite-alphabet equalization matrices.

computed via (13) once $\mathbf{X}^H$ has been determined. Even though all the algorithms use the same finite alphabet for the entries of $\mathbf{X}^H$, FAME-based algorithms are able to achieve a better performance as they are (approximately) solving the FAME problem in (12).

Remark 11: The improved performance of FAME-FBS over FL-MMSE comes at the cost of a higher complexity, as shown in Table I. Hence, there exists a performance-complexity trade-off between using FL-MMSE and FAME-FBS to generate finite-alphabet equalizers. In addition, the complexity and performance of FAME-FBS can be further tuned via the number of iterations t_{max}. Finally, the equalizer resolution offers another performance-complexity trade-off: The use of more bits for the finite-alphabet equalization matrix improves the performance, but also increases the circuit's power consumption and silicon area—a trade-off we will study next.

V. HARDWARE-LEVEL EVALUATION

To demonstrate the real-world benefits of finite-alphabet equalization, we now quantify the power and area savings that can be attained in comparison with conventional, high-resolution equalizers.

A. Equalizer Architectures

To arrive at a fair comparison between finite-alphabet equalization and conventional, high-resolution equalizers,

we implemented two equalization circuits: one for finite-alphabet equalization and one for high-resolution equalization.

The high-resolution equalizer computes a matrix-vector product between the $U \times B$ equalization matrix $\mathbf{W}^H$ and the received vector $\mathbf{y}$. The matrix-vector product is computed in a column-by-column fashion by using a linear array of U parallel multiply-accumulate (MAC) units over B clock cycles. The multipliers in the MAC units are high-resolution and take as input 10-bit numbers from the equalization matrix $\mathbf{W}^H$ and 7-bit numbers from the received vector $\mathbf{y}$. The accumulators in the MAC units use 18 bits. Finally, 9 bits are taken from both real and imaginary accumulators as the outputs of each MAC unit. These outputs correspond to the estimates $\hat{\mathbf{s}} = \mathbf{W}^H \mathbf{y}$.

The finite-alphabet equalizer computes a low-resolution matrix-vector product between the $U \times B$ finite-alphabet matrix $\mathbf{X}^H$ and the received vector $\mathbf{y}$. This matrix-vector product is implemented in the same way as in the traditional equalizer, with the difference that far fewer bits are used for the multipliers and accumulators. The multipliers take as input r-bit numbers from $\mathbf{X}^H$ and 7-bit numbers from $\mathbf{y}$, while the accumulators use $r + 13$ bits (except for the case where $r = 1$, where the accumulators use 13 bits). We take 9 bits from the accumulators in each MAC unit as the output of the low-resolution matrix-vector product $\mathbf{X}^H \mathbf{y}$. Unlike conventional equalization, the results of the U-dimensional vector $\mathbf{X}^H \mathbf{y}$ are scaled by the values in $\boldsymbol{\beta}^*$. This scaling

TABLE III

IMPLEMENTATION RESULTS IN 28 nm CMOS FOR ONE EQUALIZER
INSTANCE OPERATING IN A SYSTEM WITH $B = 256$ AND $U = 16$

Equalizer resolution r [bit]	1	2	3	4	5	10
Silicon area [mm^2]	0.06	0.08	0.10	0.14	0.16	0.26
Clock freq. [GHz]	1.33	1.25	1.25	1.16	1.16	1.05
Throughput [M vectors/s]	5.18	4.88	4.88	4.53	4.53	4.10
Power^a [mW]	18.5	29.2	38.8	42.6	51.3	57.1

^{a}Extracted from stimuli-based post-layout simulations in the typical-typical
process corner at 25°C with a nominal power supply of 0.9V.

operation is implemented with a high-resolution multiplier that
computes the product between the 9-bit $\mathbf{x}_u^H \mathbf{y}$ and the 10-bit
scaling factor β_u^*. The output of this multiplier is represented
using 9 bits per real and imaginary components and correspond
to the estimates $\hat{\mathbf{s}} = \mathbf{V}^H \mathbf{y}$.

B. Implementation Results

Table III lists post-layout implementation results for the
circuits discussed in Section V-A implemented for a $B = 256$
BS antenna, $U = 16$ UE system, using a 28 nm CMOS tech-
nology. The traditional, high-resolution equalizer corresponds
to the design with an equalization resolution r of 10 bits,
whereas the finite-alphabet equalizer was implemented for
$r = \{1, 2, \ldots, 5\}$ bits. To allow for a fair comparison between
the different equalization circuits, we consider a scenario in
which all of the designs support the same throughput. We
assume a throughput of 2 G (complex-valued) vectors/s, which
implies that the $2B$ ADCs at the BS run at 2 G samples/s. As
we can see from Table III, a single instance of our equalizer
designs reaches throughputs of the order of M vectors/s, which
is well below the target throughput of 2 G vectors/s. We can,
however, instantiate a time-multiplexed array of equalizers that
achieve the desired throughput (at the expense of increased
area). Assuming no overhead for this replication approach,
we can estimate the total silicon area and power consumption
required to perform equalization in a high-bandwidth mmWave
setting; Figure 8 shows the corresponding results.

Figure 8(a) shows the power consumption reduction
achieved by lowering the equalizer resolution; Figure 8(b)
shows the same effect but on silicon area. We see that halving
the number of bits used for the high-resolution equalizer
already introduces substantial gains of 19% and 44% lower
power and area, respectively. Further reducing the equalizer
resolution reduces the power and area by a factor of $3.9\times$
and $5.8\times$, respectively, when a 1-bit finite-alphabet equalizer
is used.

*Remark 12: We note that the power and area can poten-
tially be reduced much more. Once the number of bits in
the equalization matrix has been reduced to 5 bits or below,
emerging processing-in-memory architectures, such as the
one proposed in [40], potentially lower the area and power
(additionally to the savings above) by about $2\times$ to $4\times$. A
detailed analysis of such emerging multiplier-array architec-
tures in combination with finite-alphabet equalization is left
for future work.*
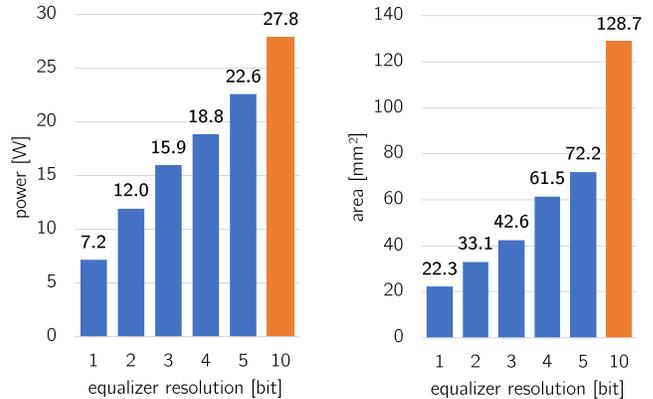
(a) Power vs. equalizer resolution   (b) Area vs. equalizer resolution

Fig. 8.   Power and area consumed by equalizer hardware designs in 28 nm
CMOS technology for a $B = 256$ BS antenna, $U = 16$ UE massive
MU-MIMO system with varying equalizer resolution. All equalizers operate
at a rate of 2 G vectors/s and 7 bits are used to represent the entries of the
received vector $\mathbf{y}$. For an equalizer resolution lower than 6 bits, we use a
finite-alphabet equalizer consisting of a low-resolution matrix-vector product,
followed by per-UE high-resolution scaling. The equalizer resolution of 10 bit
is executed with a high-resolution matrix-vector product only. Finite-alphabet
equalization (shown in blue) can reduce the power and area of conventional,
high-precision equalization (shown in orange) by a factor of $3.9\times$ and $5.8\times$,
respectively.

VI. CONCLUSION

We have proposed finite-alphabet equalization, a par-
adigm in which the spatial equalization matrix contains
low-resolution numbers in order to enable energy- and
area-efficient equalization hardware. To achieve an error-rate
performance similar to that of conventional, high-resolution
equalizers, such as the L-MMSE equalizer, we have formu-
lated the finite-alphabet MMSE equalization (FAME) prob-
lem, which minimizes the post-equalization MSE. We have
shown that solving the FAME problem yields significant
improvements over finite-alphabet matrices that are obtained
by naïvely quantizing the L-MMSE matrix in terms of EVM,
beamforming capabilities, and uncoded BER. Since the FAME
problem is NP-hard, we have proposed approximate algo-
rithms that trade-off performance with complexity. One of
the proposed algorithms, FAME-FBS, achieves a performance
that is on par with semidefinite relaxation while having the
same asymptotic complexity scaling as L-MMSE equalization.
We have shown that FAME-FBS significantly outperforms
a baseline finite-alphabet equalizer for LoS and non-LoS
massive MU-MIMO mmWave channel models in terms of
EVM and uncoded BER. In addition, our reference VLSI
implementation results in 28 nm CMOS have demonstrated
that the use of finite-alphabet equalization is able to reduce
the power and area of spatial equalization by at least a
factor of $3.9\times$ and $5.8\times$, respectively. Thus, finite-alphabet
equalization is a viable solution to combat the excessively high
power consumption and area of all-digital massive MU-MIMO
mmWave BS designs.

There are many avenues of future work. A theoreti-
cal convergence and performance analysis of FAME-FBS
is an interesting (but difficult) open problem. Moreover,

$$\left\| \mathbf{e}_u - \mathbf{H}^H \beta(\tilde{\mathbf{x}})\tilde{\mathbf{x}} \right\|_2^2 + \rho|\beta(\tilde{\mathbf{x}})|^2 \|\tilde{\mathbf{x}}\|_2^2 \tag{36}$$

$$= \left\| \mathbf{e}_u - \frac{\mathbf{H}^H \tilde{\mathbf{x}}\tilde{\mathbf{x}}^H \mathbf{H}}{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2}\mathbf{e}_u \right\|_2^2 + \rho\frac{|\tilde{\mathbf{x}}^H \mathbf{H}\mathbf{e}_u|^2}{(\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2)^2}\|\tilde{\mathbf{x}}\|_2^2 \tag{37}$$

$$= \left\| \left(\mathbf{I} - \frac{\mathbf{H}^H \tilde{\mathbf{x}}\tilde{\mathbf{x}}^H \mathbf{H}}{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2} \right)\mathbf{e}_u \right\|_2^2 + \rho\frac{|\tilde{\mathbf{x}}^H \mathbf{H}\mathbf{e}_u|^2}{(\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2)^2}\|\tilde{\mathbf{x}}\|_2^2 \tag{38}$$

$$= \|\mathbf{e}_u\|_2^2 - 2\frac{\mathbf{e}_u^H \mathbf{H}^H \tilde{\mathbf{x}}\tilde{\mathbf{x}}^H \mathbf{H}\mathbf{e}_u}{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2} + \frac{\mathbf{e}_u^H \mathbf{H}^H \tilde{\mathbf{x}}\tilde{\mathbf{x}}^H \mathbf{H}\mathbf{H}^H \tilde{\mathbf{x}}\tilde{\mathbf{x}}^H \mathbf{H}\mathbf{e}_u}{(\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2)^2} + \rho\frac{|\tilde{\mathbf{x}}^H \mathbf{H}\mathbf{e}_u|^2}{(\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2)^2}\|\tilde{\mathbf{x}}\|_2^2 \tag{39}$$

$$= 1 - 2\frac{|\tilde{\mathbf{x}}^H \mathbf{H}\mathbf{e}_u|^2}{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2} + \frac{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2|\tilde{\mathbf{x}}^H \mathbf{H}\mathbf{e}_u|^2}{(\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2)^2} + \rho\frac{|\tilde{\mathbf{x}}^H \mathbf{H}\mathbf{e}_u|^2}{(\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2)^2}\|\tilde{\mathbf{x}}\|_2^2 \tag{40}$$

$$= 1 + \frac{|\tilde{\mathbf{x}}^H \mathbf{H}\mathbf{e}_u|^2}{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2}\left(-2 + \frac{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2}{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2} + \frac{\rho\|\tilde{\mathbf{x}}\|_2^2}{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2} \right) \tag{41}$$

$$= 1 + \frac{|\tilde{\mathbf{x}}^H \mathbf{H}\mathbf{e}_u|^2}{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2}\left(-2\frac{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2}{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2} + \frac{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2}{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2} + \frac{\rho\|\tilde{\mathbf{x}}\|_2^2}{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2} \right) \tag{42}$$

$$= 1 - \frac{|\tilde{\mathbf{x}}^H \mathbf{H}\mathbf{e}_u|^2}{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2} = 1 - \frac{|\mathbf{h}_u^H \tilde{\mathbf{x}}|^2}{\|\mathbf{H}^H \tilde{\mathbf{x}}\|_2^2 + \rho\|\tilde{\mathbf{x}}\|_2^2}. \tag{43}$$

---

the development of algorithms that outperform FAME-FBS and further approach the performance of FAME-EXH at low complexity is a challenging open research direction. The finite-alphabet equalization paradigm is also applicable to downlink precoding in massive MU-MIMO mmWave systems [41] and could be used in other applications where matrix-vector products must be computed at high rates or with low power consumption—an investigation of other applications is part of ongoing research.

APPENDIX A
PROOF OF LEMMA 1

We start by deriving the expression for the optimal scaling factor β_u given an equalization vector $\mathbf{x}_u$ and then use the resulting quantity to simplify the optimization problem. We first take the Wirtinger derivative of the objective function in (11) in the complex-valued variable $\tilde{\beta}^*$ and set it to zero:

$$\frac{\partial}{\partial\tilde{\beta}^*}\left(\|\mathbf{e}_u - \mathbf{H}^H \tilde{\beta}\tilde{\mathbf{x}}\|^2 + \rho\|\tilde{\beta}\tilde{\mathbf{x}}\|^2 \right) = 0 \tag{34}$$

$$-\tilde{\mathbf{x}}^H \mathbf{H}\mathbf{e}_u + \tilde{\mathbf{x}}^H \mathbf{H}\mathbf{H}^H \tilde{\mathbf{x}}\tilde{\beta} + \rho\|\tilde{\mathbf{x}}\|^2\tilde{\beta} = 0. \tag{35}$$

Since $\mathbf{H}\mathbf{e}_u = \mathbf{h}_u$, we obtain (13) by solving for $\tilde{\beta}$. To obtain (12), we substitute (13) into (11) and simplify the resulting expression using algebraic manipulations. Concretely, we carry out the steps listed in (36)–(43), as shown at the top of this page.

APPENDIX B
COMPLEXITY COUNTS FOR COMPUTING EQUALIZATION MATRICES WITH DIFFERENT ALGORITHMS

In what follows, we ignore the complexity of reciprocals, square roots, and additions. The numbers in parentheses are real-valued multiplications, where we assume that a complex-valued multiplication requires four real-valued multiplications.

A. Complexity of Explicit L-MMSE

Explicit L-MMSE equalization corresponds to computing $\mathbf{W}^H = (\rho\mathbf{I}_U + \mathbf{H}^H \mathbf{H})^{-1}\mathbf{H}^H$, which can be achieved at low complexity using the approach detailed in [18]. First, we calculate $\mathbf{A} = \rho\mathbf{I}_U + \mathbf{H}^H \mathbf{H}$ ($2BU^2$). Then, we apply a Cholesky decomposition to $\mathbf{A}$ so that $\mathbf{A} = \mathbf{L}\mathbf{L}^H$ ($\frac{2}{3}U^3 - \frac{2}{3}U$). Next, we compute the inverse of $\mathbf{L}$ via back-substitution ($\frac{2}{3}U^3 - \frac{5}{3}U + 1$), to calculate $\mathbf{A}^{-1} = \mathbf{L}^{-H}\mathbf{L}^{-1}$ ($\frac{2}{3}U^3 + \frac{1}{3}U$). Finally, we obtain $\mathbf{W}^H = \mathbf{A}^{-1}\mathbf{H}^H$ ($4BU^2 - 2BU$). The total complexity of the explicit L-MMSE equalizer is $2U^3 + 6BU^2 - 2BU - 2U + 1$. Since in massive MU-MIMO systems we typically have $B \gg U$, the asymptotic complexity scales as $O(BU^2)$.

B. Complexity of FL-MMSE

We start by computing the explicit L-MMSE equalizer, which, as shown in Appendix B-A, entails a complexity of $2U^3 + 6BU^2 - 2BU - 2U + 1$. Then, we quantize the entries of the L-MMSE equalizer. We will not count the complexity of quantization as there are hardware-efficient ways to do so. Now that $\mathbf{X}^H$ has been determined, the next step is to compute the optimal scaling factor $\beta_u(\mathbf{x}_u)$ for each UE. We need to first calculate $\mathbf{H}^H \mathbf{x}_u$ ($4BU$) from which we also extract $\mathbf{x}_u^H \mathbf{h}_u$. Then, we compute the ℓ_2-norm of $\mathbf{H}^H \mathbf{x}_u$ and $\mathbf{x}_u$ ($2U$ and $2B$, respectively). The next steps are to scale $\|\mathbf{x}_u\|_2^2$ by ρ (1 multiplication), and obtain $\beta_u(\mathbf{x}_u)$ by multiplying $\mathbf{x}_u^H \mathbf{h}_u$ and the multiplicative inverse of $\|\mathbf{H}^H \mathbf{x}_u\|_2^2 + \rho\|\mathbf{x}_u\|_2^2$ (2 multiplications). Then, computing $\beta_u(\mathbf{x}_u)$ for all UEs $u = 1, \ldots, U$ has a complexity of $4BU^2 + 2BU + 2U^2 + 3U$. Thus, computing $\mathbf{X}^H$ and $\boldsymbol{\beta}$ for FL-MMSE equalization has a total complexity of $10BU^2 + 2U^3 + 2U^2 + U + 1$. As expected, FL-MMSE has the same asymptotic complexity scaling $O(BU^2)$ as the L-MMSE equalizer.

C. Complexity of FAME-FBS

For each UE $u = 1, \ldots, U$, one instance of FAME-FBS is executed. Each FAME-FBS instance requires an iterative procedure with t_{max} iterations followed by the computation of $\beta_u(\mathbf{x}_u)$. In the iterative procedure, we compute $\mathbf{H}^H \tilde{\mathbf{x}}$ ($4BU$) and scale one of its entries with $\gamma^{(t)}$ (2 multiplications). With this, we have computed the vector $(\mathbf{H}^H - \gamma^{(t)} \mathbf{e}_u \mathbf{e}_u^H \mathbf{H}^H) \tilde{\mathbf{x}}$, which we now multiply with $\mathbf{H}$ ($4BU$) to obtain the gradient $\nabla f(\tilde{\mathbf{x}})$ in (30). The next step in FAME-FBS is to scale $\nabla f(\tilde{\mathbf{x}})$ with $\tau^{(t)}$ ($2B$) to compute $\tilde{\mathbf{z}}^{(t+1)}$ in (25). Then, the entries of $\tilde{\mathbf{z}}^{(t+1)}$ are scaled by $\nu^{(t)}$ ($2B$), completing one FAME-FBS iteration. Hence, to compute $\mathbf{x}_u$, FAME-FBS requires $(8BU + 4B + 2)t_{max}$ real-valued products. As shown in Appendix B-B, computing the optimal scaling factor $\beta_u(\mathbf{x}_u)$ requires $4BU + 2B + 2U + 3$ real-valued multiplications per UE. Thus, the total computational complexity of FAME-FBS is $(8t_{max} + 4)BU^2 + 2U^2 + 2(2t_{max} + 1)BU + (2t_{max} + 3)U$. Hence, FAME-FBS has an asymptotic complexity of $O(BU^2)$.

REFERENCES

[1] T. S. Rappaport *et al.*, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.

[2] A. L. Swindlehurst, E. Ayanoglu, P. Heydari, and F. Capolino, "Millimeter-wave massive MIMO: The next wireless revolution?" *IEEE Commun. Mag.*, vol. 52, no. 9, pp. 56–62, Sep. 2014.

[3] F. Rusek, D. Persson, B. Kiong Lau, E. G. Larsson, T. L. Marzetta, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

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

[5] W. Roh *et al.*, "Millimeter-wave beamforming as an enabling technology for 5G cellular communications: Theoretical feasibility and prototype results," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 106–113, Feb. 2014.

[6] B. Sadhu *et al.*, "7.2 a 28 GHz 32-element phased-array transceiver IC with concurrent dual polarized beams and 1.4 degree beam-steering resolution for 5G communication," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2017, pp. 128–129.

[7] A. Alkhateeb, J. Mo, N. González-Prelcic, and R. W. Heath, Jr., "MIMO precoding and combining solutions for millimeter-wave systems," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 122–131, Dec. 2014.

[8] Z. Pi and F. Khan, "An introduction to millimeter-wave mobile broadband systems," *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 101–107, Jun. 2011.

[9] E. Björnson, L. Van der Perre, S. Buzzi, and E. G. Larsson, "Massive MIMO in sub-6 GHz and mmWave: Physical, practical, and use-case differences," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 100–108, Apr. 2019.

[10] S. Dutta, C. N. Barati, D. Ramirez, A. Dhananjay, J. F. Buckwalter, and S. Rangan, "A case for digital beamforming at mmWave," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 756–770, Feb. 2020.

[11] J. Mo and R. W. Heath, Jr., "Capacity analysis of one-bit quantized MIMO systems with transmitter channel state information," *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5498–5512, Oct. 2015.

[12] K. Roth and J. A. Nossek, "Achievable rate and energy efficiency of hybrid and digital beamforming receivers with low resolution ADC," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2056–2068, Sep. 2017.

[13] S. Jacobsson, G. Durisi, M. Coldrey, U. Gustavsson, and C. Studer, "Throughput analysis of massive MIMO uplink with low-resolution ADCs," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 4038–4051, Jun. 2017.

[14] R. Zimmerman, "Computer arithmetic: Principles, architectures, and VLSI design," Integr. Syst. Lab., ETH Zürich, Tech. Rep., 1999.

[15] C. Studer and G. Durisi, "Quantized massive MU-MIMO-OFDM uplink," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2387–2399, Jun. 2016.

[16] J. Mo, A. Alkhateeb, S. Abu-Surra, and R. W. Heath, Jr., "Hybrid architectures with few-bit ADC receivers: Achievable rates and energy-rate tradeoffs," *IEEE Trans. Wireless Commun.*, vol. 16, no. 4, pp. 2274–2287, Apr. 2017.

[17] H. Yan, S. Ramesh, T. Gallagher, C. Ling, and D. Cabric, "Performance, power, and area design trade-offs in millimeter-wave transmitter beamforming architectures," *IEEE Circuits Syst. Mag.*, vol. 19, no. 2, pp. 33–58, May 2019.

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

[19] M. Wu, B. Yin, G. Wang, C. Dick, J. Cavallaro, and C. Studer, "Large-scale MIMO detection for 3GPP LTE: Algorithm and FPGA implementation," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 916–929, Oct. 2014.

[20] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

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

[22] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comput.*, vol. 44, no. 170, pp. 463–471, Apr. 1985.

[23] S. Verdú, "Computational complexity of optimum multiuser detection," *Algorithmica*, vol. 4, nos. 1–4, pp. 303–312, Jun. 1989.

[24] S. Jacobsson, G. Durisi, M. Coldrey, T. Goldstein, and C. Studer, "Quantized precoding for massive MU-MIMO," *IEEE Trans. Commun.*, vol. 65, no. 11, pp. 4670–4684, Nov. 2017.

[25] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[26] Z.-Q. Luo, W.-K. Ma, A. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.

[27] T.-H. Chang, C.-W. Hsin, W.-K. Ma, and C.-Y. Chi, "A linear fractional semidefinite relaxation approach to maximum-likelihood detection of higher-order QAM OSTBC in unknown channels," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2315–2326, Apr. 2010.

[28] O. Castañeda, T. Goldstein, and C. Studer, "Data detection in large multi-antenna wireless systems via approximate semidefinite relaxation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 12, pp. 2334–2346, Dec. 2016.

[29] M. Wu, C. Dick, J. R. Cavallaro, and C. Studer, "High-throughput data detection for massive MU-MIMO-OFDM using coordinate descent," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 12, pp. 2357–2367, Dec. 2016.

[30] S. Shah, A. K. Yadav, C. D. Castillo, D. W. Jacobs, C. Studer, and T. Goldstein, "Biconvex relaxation for semidefinite programming in computer vision," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2016, pp. 717–735.

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

[32] T. Goldstein and S. Setzer, "High-order methods for basis pursuit," UCLA, Los Angeles, CA, USA, CAM Tech. Rep., 2010.

[33] T. Goldstein, C. Studer, and R. Baraniuk, "A field guide to forward-backward splitting with a FASTA implementation," Nov. 2014, *arXiv:1411.3406*. [Online]. Available: http://arxiv.org/abs/1411.3406

[34] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, Jan. 2014.

[35] A. Balatsoukas-Stimming, O. Castañeda, S. Jacobsson, G. Durisi, and C. Studer, "Neural-network optimized 1-bit precoding for massive MU-MIMO," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2019, pp. 1–5.

[36] O. Castañeda, S. Jacobsson, G. Durisi, T. Goldstein, and C. Studer, "VLSI design of a 3-bit constant-modulus precoder for massive MU-MIMO," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5.

[37] O. Castañeda, S. Jacobsson, G. Durisi, M. Coldrey, T. Goldstein, and C. Studer, "1-bit massive MU-MIMO precoding in VLSI," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 7, no. 4, pp. 508–522, Dec. 2017.

[38] *5G; NR; Base Station (BS) Radio Transmission and Reception*, document TS 38.104 version 15.5.0 Rel. 15, 3GPP, May 2019.

[39] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, Jun. 2014.

[40] O. Castañeda, M. Bobbett, A. Gallyas-Sanhueza, and C. Studer, "PPAC: A versatile in-memory accelerator for matrix-vector-product-like operations," in *Proc. IEEE 30th Int. Conf. Appl.-Specific Syst., Archit. Processors (ASAP)*, Jul. 2019, pp. 149–156.

[41] O. Castañeda, S. Jacobsson, G. Durisi, T. Goldstein, and C. Studer, "Finite-alphabet Wiener filter precoding for mmWave massive MU-MIMO systems," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019, pp. 178–183.

Oscar Castañeda (Graduate Student Member, IEEE) received the B.S. degree in electronics engineering from Del Valle de Guatemala University, Guatemala City, Guatemala in 2016. He is currently pursuing the Ph.D. degree with the VLSI Information Processing (VIP) Group, School of Electrical and Computer Engineering, Cornell University. In 2017, he was a Visiting Researcher with the Integrated Systems Laboratory, ETH Zurich. His research interests include digital signal processing, emerging computer architectures, and digital VLSI circuit and system design. In 2019, he received a Qualcomm Innovation Fellowship, as well as a Top 10 Best Student Presentation Award at SRC TECHCON.

Sven Jacobsson (Graduate Student Member, IEEE) received the Ph.D. degree in electrical engineering from the Chalmers University of Technology, Gothenburg, Sweden, in 2019. In 2015, he joined Ericsson Research, where he is currently an Experienced Researcher. His research interests include advanced antenna systems and hardware-constrained communications.

Giuseppe Durisi (Senior Member, IEEE) received the Laurea *(summa cum laude)* and Ph.D. degrees from Politecnico di Torino, Italy, in 2001 and 2006, respectively. From 2006 to 2010, he was a Post-Doctoral Researcher with ETH Zurich, Zurich, Switzerland. Since 2010, he has been with the Chalmers University of Technology, Gothenburg, Sweden, where he is currently a Professor with the Communication Systems Group. He is also a Guest Researcher with Ericsson, Sweden, and a Co-Director of the Chalmers Information and Communication Technology Area of Advance. His research interests are in the areas of communication and information theory. He was a recipient of the 2013 IEEE ComSoc Best Young Researcher Award for the Europe, Middle East, and Africa Region, and is co-author of an article that won a student paper award at the 2012 International Symposium on Information Theory, and of a paper that won the 2013 IEEE Sweden VT-COM-IT joint chapter best student conference paper award. In 2015, he joined the editorial board of the IEEE Transactions on Communications as an Associate Editor. From 2011 to 2014, he served as a Publications Editor of the IEEE Transactions on Information Theory.

Tom Goldstein (Member, IEEE) received the Ph.D. degree from UCLA. He is currently an Associate Professor with the Department of Computer Science, University of Maryland. He held research positions at Stanford University and Rice University. His research focuses on optimization for machine learning and signal processing. His work ranges from large-scale computing on distributed architectures to inexpensive power-aware algorithms for small-scale embedded systems. Applications of his work include scalable machine learning, computer vision, and methods for wireless communications. He has been a recipient of several awards, including SIAM's DiPrima Prize, a DARPA Young Faculty Award, and a Sloan Fellowship.

Christoph Studer (Senior Member, IEEE) received the Ph.D. degree in information technology and electrical engineering from ETH Zurich in 2009. In 2005, he was a Visiting Researcher with the Smart Antennas Research Group, Stanford University. From 2006 to 2009, he was a Research Assistant in both the Integrated Systems Laboratory and the Communication Technology Laboratory (CTL) at ETH Zurich. From 2009 to 2012, he was a Post-Doctoral Researcher with CTL, ETH Zurich, and the Digital Signal Processing Group, Rice University. In 2013, he was a Research Scientist with Rice University. From 2014 to 2019, he was an Assistant Professor with Cornell University. Since 2019, he has been an Associate Professor with Cornell University and Cornell Tech. Since 2014, he has also been an Adjunct Professor with Rice University. His research interests include the design of very large-scale integration (VLSI) circuits, as well as wireless communications, optimization, and machine learning. He received ETH Medals for his M.S. and Ph.D. theses in 2006 and 2009, respectively. He received a Swiss National Science Foundation fellowship for Advanced Researchers in 2011 and a U.S. National Science Foundation CAREER Award in 2017. He won a Michael Tien '72 Excellence in Teaching Award from the College of Engineering, Cornell University, in 2016. He shared the Swisscom/ICTnet Innovations Award in 2010 and 2013. He was the winner of the Student Paper Contest of the 2007 Asilomar Conference on Signals, Systems, and Computers, received a Best Student Paper Award of the 2008 IEEE International Symposium on Circuits and Systems (ISCAS), and shared the best Live Demonstration Award at the IEEE ISCAS in 2013. In 2019, he was the Technical Program Chair of the Asilomar Conference on Signals, Systems, and Computers, and a Technical Program Co-Chair of the IEEE International Workshop on Signal Processing Systems (SiPS). He is currently an Associate Editor of the IEEE Open Journal of Circuits and Systems.