Lattice Quantization of Phases for Equal Gain Transmission

Shang-Ho Tsai, *Member, IEEE*, and Yuan-Pei Lin, *Senior Member, IEEE*

***Abstract*—Equal gain transmission (EGT) requires only the feedback of phase information and is known to enjoy several implementational advantages compared to maximum ratio transmission (MRT). This paper investigates the use of lattice quantization for quantizing phases in EGT systems. In practice, the feedback rate is finite and the lattices need to be truncated to have finite code rate. The truncation of lattices results in boundary effect; hence, earlier results developed for untruncated lattices, e.g., fast algorithm and mean-square error (MSE) analysis, cannot be applied directly. Exploiting the fact that the quantized signals are phases, we show how to quantize using truncated lattices so that there is no boundary effect. Moreover, we propose efficient methods to convert between lattice codewords and binary digits so that the lattice quantization can be used in practical systems. Furthermore, the mean-square quantization error is analyzed for several useful truncated lattices. We also show how to analyze the signal-to-noise ratio (SNR) loss when the proposed quantization is applied to an EGT system.**

***Index Terms*—Beamforming, EGT, equal gain transmission, Grassmannian, lattice, Lloyd, phase quantization, precoding.**

I. Introduction

MULTIPLE-INPUT multiple-output (MIMO) beamforming (precoding) techniques are widely used in current wireless communication standards, e.g., IEEE 802.11n, IEEE 802.16 family and Long Term Evolution (LTE). The diversity gain provided by beamforming techniques enables the systems to achieve reliable link quality for long transmission distance without increasing transmit power and bandwidth.

When the transmitter has complete channel information, the optimal beamforming scheme, i.e., maximum ratio transmission (MRT), can be used. MRT maximizes the received signal-to-noise ratio (SNR), by using the right singular vector corresponding to the largest singular value of channel matrix. In a wireless system where complete channel information is not available to the transmitter, the beamforming vectors need to be quantized before they are sent back to the transmitter. Since both magnitude and phase information need to be quantized, codebook-based vector quantization (VQ) is usually used for quantizing MRT. In [1], the authors proposed beamforming schemes with finite-rate feedback to achieve the minimum outage probability. In [2], a codebook-based VQ scheme called Grassmannian beamforming was proposed to minimize the chordal distance between the true and quantized beamforming vectors for a given feedback rate. When the number of transmit antennas N_t is large, a codebook with a large size is needed. However, constructing large Grassmannian codebooks may not be an easy task [2]. Also, large amount of memory is required at the transmitter and receiver for codebook storage. In addition, exhaustive search is usually needed to determine the closest codeword in the codebook; the computational complexity increases with the code size and so dose the feedback latency.

When the elements of the beamforming vectors have the same magnitude, the beamforming scheme is called equal gain transmission (EGT) [3], [4]. Although EGT is not the optimal beamforming scheme, it only leads to a 1.05-dB SNR loss compared to the optimal MRT [5]. Moreover, the design of power amplifier is easier for EGT than that for MRT, because the transmit power of EGT is a constant [6]. It was shown in [3] that full diversity order can be achieved by quantized EGT. Scalar quantization (SQ) has been suggested for quantizing the beamforming vectors of EGT; e.g., see [4]–[6]. The capacity loss of quantized EGT with SQ was analyzed in [4]. The SNR loss of the quantized EGT with SQ was analyzed in [5]. Since there is no closed-form solution for the best MIMO EGT, an iterative MIMO EGT design and the corresponding bit allocations with SQ were proposed in [6]. When the code size is small, it was shown in [7] that quantized EGT beamforming can achieve the same optimal performance as quantized MRT by carefully choosing the columns from a discrete Fourier transform (DFT) matrix; an extended discussion on this topic was given in [8].

One advantage of SQ is that it does not require pre-computation of codebooks and the result can be used for arbitrary number of transmit antennas. However, when SQ is used in EGT, the code rate B is restricted to be an integer multiple of $(N_t - 1)$, where N_t is the number of transmit antennas. This limits the choice of quantization rates, especially for large N_t. More flexible code rate and better performance can be achieved using VQ [3], [4], [9]. Although VQ may be designed for an arbitrary code rate and has better performance than SQ, in general it is a codebook-based quantization and hence requires codebook storage, more computations for codeword selection. It is also more difficult to design for large N_t.

Lattice quantizer is a special class of vector quantization, which has a highly regular structure. Due to the regular structure, there is no need to store the codebook. Moreover, fast

Manuscript received August 10, 2010; revised March 22, 2011 and August 30, 2011; accepted October 13, 2011. Date of publication October 24, 2011; date of current version January 13, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Gerald Matz. This work is supported by the National Science Council (NSC), Taiwan, Cooperative Agreement No. 99-2221-E-009-100-MY3.

The authors are with the Department of Electrical Engineering, National Chiao Tung University, Hsinchu 30050, Taiwan (e-mail: shanghot@mail.nctu.edu.tw; ypl@mail.nctu.edu.tw).

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

Digital Object Identifier 10.1109/TSP.2011.2173333

quantization algorithms are available for several kinds of lattices [10]. The mean-square error (MSE) for lattice quantization has been analyzed in [11]–[13]. However, in the development of fast algorithms and MSE analysis [10], [11], [14], [15], it is generally assumed that the lattice is not truncated and the codebook size is infinite. In practice, truncation of lattices is needed so that the codebook size M is finite. A constant 2-norm criterion was proposed in [11] for lattice truncation. The corresponding code rate $B = \log_2 M$ is in general not an integer, which is more difficult to implement in practical applications. Truncation of lattices leads to boundary regions, for which the regularity of lattice is no longer followed. In these boundary regions, fast algorithms developed for untruncated lattices cannot be applied and extra computations are needed [15]. As the Voronoi regions are of irregular shapes and of different areas, the MSE analysis for untruncated lattices does not carried over.

In this paper, we propose a lattice truncation scheme for quantizing phase vectors by exploiting the wrapping-around property of phases, i.e., π and $-\pi$ yielding the same phase. With the proposed lattice truncation, we can avoid boundary effect all together. All Voronoi regions are of the same shapes and areas; fast algorithms designed for untruncated lattices [10] can be applied in a straightforward manner. Moreover, integer code rate B is achievable using the proposed lattice truncation. MSE can be given as a closed form due to the regular structure of lattices. The MSE result provides a useful design reference. Analytical results show that the MSE of the proposed lattice quantization is 0.4–0.65 dB smaller than that of a scalar quantizer. Furthermore, we will analyze the SNR loss when the proposed method is used to quantize phase vectors in EGT. Simulation results show that the performance of the proposed lattice quantizer is comparable to those of the Lloyd-codebook [4] EGT beamforming and the Grassmannian beamforming [2].

The rest of this paper is organized as follows. The system model is presented in Section II. Lattice quantization is reviewed in Section III. In Section IV, we introduce the quantization of phase vectors using the lattice D_n. These quantization using truncated lattices D_n^* and E_8 are introduced in Section V. The mean-square quantization error of phase quantizers using truncated lattices is analyzed in Section VI and the SNR loss of EGT with lattice quantization is analyzed in VII. Simulation results are provided in Section VII. Finally, concluding remarks are given in Section IX.

Notations: Boldfaced lowercases and boldfaced uppercases denote vectors and matrices, respectively. $\mathbb{E}\{x\}$ is the expectation of random variable x. $\mathbf{A}^*$ and $\mathbf{A}^T$ denote the conjugate and transpose of $\mathbf{A}$, respectively. $\mathbf{A}^\dagger$ is the conjugate transposition of $\mathbf{A}$. σ_x^2 is the variance of x. $\lfloor x \rfloor$ is the floor function of x.

II. System Model of EGT System

The block diagram of an EGT beamforming system with N_t transmit antennas and one receive antenna is shown in Fig. 1. At the first stage, one transmit symbol x is sent to N_t branches for beamforming. The symbol is multiplied by $e^{j\theta_i}$ in the ith branch. The transmitted vector $\mathbf{s} = (s_1\, s_2 \cdots s_{N_t})^t$ is given by $\mathbf{s} = \frac{1}{\sqrt{N_t}}\mathbf{p}x$, where $\mathbf{p} = \left(e^{j\theta_1}\, e^{j\theta_2} \cdots e^{j\theta_{N_t}}\right)^T$. At the receive side, the received scalar r is

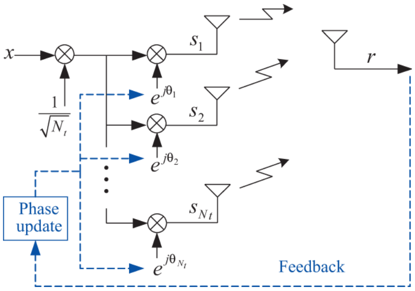

Fig. 1. A block diagram of the EGT beamforming system.

$$r = \mathbf{h}^T\mathbf{s} + e \tag{1}$$

where $\mathbf{h} = (h_1\, h_2 \cdots h_{N_t})$ is an $N_t \times 1$ channel vector and e is the additive channel noise with a complex Gaussian distribution. Detection is applied on $z = (\mathbf{h}^T\mathbf{p})^* r$, where $*$ denotes conjugation. We can express z as the sum of a signal term z_x and a noise term z_e, i.e.,

$$z = (\mathbf{h}^T\mathbf{p})^* r = \underbrace{\frac{1}{\sqrt{N_t}}|\mathbf{h}^T\mathbf{p}|^2 x}_{z_x} + \underbrace{(\mathbf{h}^T\mathbf{p})^* e}_{z_e}. \tag{2}$$

The instantaneous SNR of z, defined as $\frac{\sigma_{z_x}^2}{\sigma_{z_e}^2}$, is given by [5]

$$\rho = \frac{1}{N_t}\frac{\sigma_x^2}{\sigma_e^2}\frac{(|\mathbf{h}^T\mathbf{p}|^2)^2}{(\mathbf{h}^T\mathbf{p})^*\mathbf{h}^T\mathbf{p}} = \frac{1}{N_t}\frac{\sigma_x^2}{\sigma_e^2}|\mathbf{h}^T\mathbf{p}|^2. \tag{3}$$

The instantaneous SNR can be maximized by choosing (see [3], [5], and [6]))

$$\mathbf{p} = (1\, e^{-j(\angle h_2 - \angle h_1)} \cdots e^{-j(\angle h_{N_t} - \angle h_1)})^T. \tag{4}$$

For notation convenience, let $n = N_t - 1$ and define the phase vector

$$\begin{aligned}\boldsymbol{\theta} &= (\theta_1\, \theta_2 \cdots \theta_n) \\ &= ((\angle h_2 - \angle h_1)\, (\angle h_3 - \angle h_1) \cdots (\angle h_{N_t} - \angle h_1)).\end{aligned} \tag{5}$$

The transmitter needs to know the phase vector $\boldsymbol{\theta}$ to achieve the best EGT performance. It is generally reasonable to assume that the channel information is available to the receiver, and the transmitter relies on the feedback from the receiver. In practice, however, the number of feedback bits is limited; hence quantization of $\boldsymbol{\theta}$ is needed before feedback. When vector quantization is used, the receiver looks into the codebook and finds the codeword that best represents $\boldsymbol{\theta}$. Then, the index of the codeword is sent back to the transmitter.

III. Review of Lattice Quantization

The review material in this section can be found in [10], [11], and [13]. A lattice is a set of vectors that can be generated using a generation matrix. Let $\mathbf{T}$ be an $m \times n$ generation matrix of a

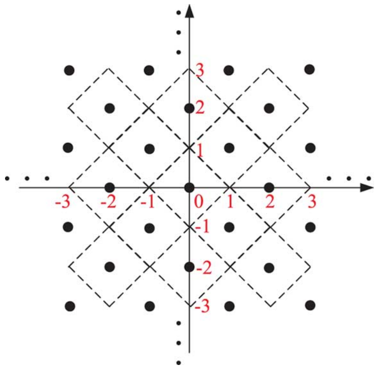

Fig. 2. An untruncated D_2 lattice.

lattice with $m \geq n$. The corresponding lattice $\mathcal{L}$ is the collection of vectors of the form $\mathbf{Tu}$, where the elements of $\mathbf{u}$ are integers, i.e., $\mathcal{L} = \{\mathbf{Tu} : u_i \in Z\}$. An example of lattice generated by $\mathbf{T} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ is given in Fig. 2, which is a special use of the D_n lattice defined by [11]

$$D_n = \{(w_1\, w_2 \cdots w_n) : w_1 + \cdots + w_n \text{ is even}, w_i \in Z\}. \quad (6)$$

An n-dimensional generation matrix of D_n is given by

$$\mathbf{T} = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ -1 & -1 & 1 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & -1 \end{pmatrix}_{n\times n}. \quad (7)$$

Lattice quantization of a vector $\mathbf{x}$ finds the lattice point that is closest to $\mathbf{x}$, i.e.,

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathcal{L}} \|\mathbf{w} - \mathbf{x}\|^2 \quad (8)$$

where $\hat{\mathbf{w}}$ is called the closest lattice point given the vector $\mathbf{x}$. Note that the design criterion in (8) is MSE instead of minimizing the maximum chordal distance in [2] and [7]. Hence, the EGT using lattice quantization in this paper is suboptimal. However, the beamforming vectors obtained by minimum MSE criterion can serve as a good candidate of optimal solution (see [7, p. 1904, column 1]). Simulation in Section VIII will show that with the proposed lattice quantization, the EGT can achieve performance comparable to the benchmark Lloyd EGT proposed in [4].

Exploiting the regular structure of lattices, Conway and Sloane have developed fast algorithm to find the closest lattice point (encoding) [10]. Let us use D_n as an example to explain this. To quantize a vector $\mathbf{x}$ to the closest vector in D_n, we first compute $f(\mathbf{x})$, where $f(\mathbf{x})$ quantizes $\mathbf{x}$ to the closest integer vector; then we find the element that has the largest absolute quantization error, requantize this element the wrong way and call the resulting vector $g(\mathbf{x})$. Thus, $f(\mathbf{x})$ and $g(\mathbf{x})$ are the same except that the worst element of $\mathbf{x}$ is rounded the wrong way. Next, we compute the summations of the elements for $f(\mathbf{x})$ and $g(\mathbf{x})$, respectively. One of the two sums will be even and the other will be odd. The one with an even sum is the lattice point in D_n. For instance, let $\mathbf{x} = (0.6 - 1.1\ 1.7\ 0.1)$. Since the first element has the largest absolute error 0.4, we have $f(\mathbf{x}) = (1 - 1\ 2\ 0)$ and $g(\mathbf{x}) = (0 - 1\ 2\ 0)$. Since the sum of $f(\mathbf{x})$ is 2 and that of $g(\mathbf{x})$ is 1, $f(\mathbf{x})$ is the lattice point closest to $\mathbf{x}$. Thanks to the fast algorithms, there is no need of codebook storage. However, when lattice VQ is applied to practical systems, we need to truncate the lattice to have a finite code size. Lattice truncation using the 2-norm criterion, i.e., $\mathcal{C} = \{\mathbf{w},\ \|\mathbf{w}\|^2 \leq c\}$ was considered in [11]; the code size M can be given in terms of c. Truncation of lattices usually leads to boundary effects [9] since the Voronoi regions of the lattice points on the boundary are not of the same shape as others, i.e., irregular. Also, the code rate $B = \log_2 M$ is generally not an integer using the 2-norm criterion. To have an integer code rate B, some codewords need to be removed or added. Also the constant 2-norm criterion may not be suitable for quantizing phase vectors. For instance, if the dimension is $n = 2$, the region to be quantized for the phase vector $(x_1,\ x_2)^T$ is a square, i.e., $-\pi \leq x_i \leq \pi$, instead of a circle. In the following sections, we propose a novel lattice truncation that is more suitable for quantizing phases and integer code rate can be attained without boundary effect.

IV. Phase Quantization Using Lattice D_n

In this section, we consider the use of lattice D_n for quantizing phase vectors. The n-dimensional lattice D_n is defined by (6) and a generation matrix is given in (7). The lattice D_4 achieves the smallest mean-square quantization error among all lattices known for dimension $n = 4$ [11] if the code size is infinite, i.e., the overload MSE of the boundary lattice points is ignored (see [11, p. 61]).

A. *Proposed Truncation of D_n*

Now, we would like to truncate D_n for the quantization of the phase vector $\boldsymbol{\theta}$ in (5). The elements in $\boldsymbol{\theta}$ are bounded by $-\pi \leq \theta_i \leq \pi$, where $1 \leq i \leq n$. As each θ_i has the same lower and upper bounds, we propose to truncate the lattice by keeping lattice points in an n-dimensional cube. Such a codebook is given by

$$\mathcal{C}_{D_n} = \{\mathbf{w} : \mathbf{w} \in D_n, -c \leq w_i \leq c\} \quad (9)$$

where c is a positive value related to code size, to be chosen later. We scale θ_i by

$$x_i = \frac{\theta_i}{\alpha}, \quad \text{where } \alpha = \frac{\pi}{c}. \quad (10)$$

Then, x_i is bounded between c and $-c$. The vector $\mathbf{x}$ is quantized to the closest codeword in $\mathcal{C}_{D_n}$, i.e.,

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathcal{C}_{D_n}} \|\mathbf{w} - \mathbf{x}\|^2. \quad (11)$$

The reconstructed phase vector is then given by

$$\hat{\boldsymbol{\theta}} = \alpha\hat{\mathbf{w}}. \quad (12)$$

Given the lattice D_n and scalar c, we can determine the number of vectors in $\mathcal{C}_{D_n}$ in a straightforward manner. However, the application here is the quantization of phases. Because π and $-\pi$ give the same phase, some vectors may lead to identical phase

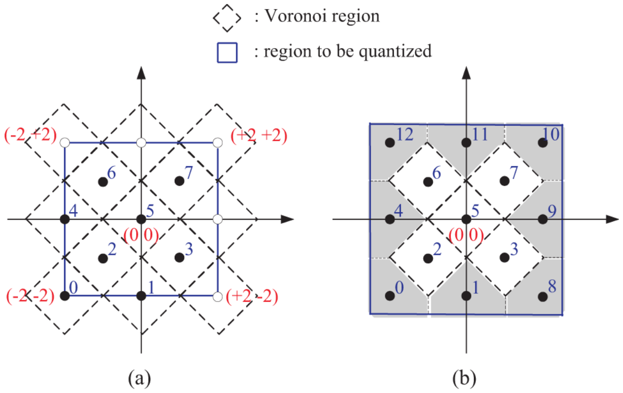

Fig. 3. Truncation for lattice D_2: (a) $c = 2$ and (b) $c = \frac{5}{2}$.

vectors. For instance, let $n = 2$ and $c = 2$, then $\alpha = \frac{\pi}{2}$. The lattice points in $\mathcal{C}_{D_2}$ are shown in Fig. 3(a). Consider the following four vectors in $\mathcal{C}_{D_2}$: $(2\ 2), (-2\ 2), (-2\ -2)$, and $(2\ -2)$. The four reconstructed phase vectors are $(\pi\ \pi)$, $(-\pi\ \pi)$, $(-\pi\ -\pi)$ and $(\pi\ -\pi)$, respectively. They are identical phase vectors and only one codeword is needed. There are 13 lattice points in $\mathcal{C}_{D_2}$, but these 13 lattice points yield only 8 distinct phase vectors (filled circles) and we can use code size $M = 8$ in this case. Therefore, for the quantization of phases, the code size M is the number of distinct phase vectors represented by the lattice points in $\mathcal{C}_{D_n}$, rather than the number of vectors in $\mathcal{C}_{D_n}$. It turns out that for lattice $\mathcal{C}_{D_n}$, M is related to c in a very simple manner as shown in the following proposition.

Proposition 1: When c is a positive integer, the number of distinct phase vectors represented by the lattice points in $\mathcal{C}_{D_n}$ in (9), i.e., the code size M, is given by

$$M = 2^{n-1}c^n. \tag{13}$$

Proof: Let us first show that if $\mathbf{w}$ is a lattice point of D_n with the ith element $w_i = c$, then, the vector $\mathbf{w}'$, with the ith element equals to $-c$ and all the other elements the same as $\mathbf{w}$, is also a lattice point of D_n. As $\mathbf{w}$ is a lattice point of D_n, the sum of the elements is even, i.e.,

$$\left(c + \sum_{j=1,j\neq i}^{n} w_j\right) \text{ is even.} \tag{14}$$

Summing up the elements of $\mathbf{w}'$, we have

$$-c + \sum_{j=1,j\neq i}^{n} w_j = \left(c + \sum_{j=1,j\neq i}^{n} w_j\right) - 2c. \tag{15}$$

Since $2c$ is even, the value in (15) is also even. Then $\mathbf{w}'$ is also a lattice of D_n. Consider the reconstructed phase vector $\boldsymbol{\theta} = \alpha\mathbf{w}$ and $\boldsymbol{\theta}' = \alpha\mathbf{w}'$. Observe that $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are identical phase vector due to the wrapping-around property. Hence, the codebook can be written as

$$\mathcal{C}_{D_n} = \{\mathbf{w} : \mathbf{w} \in D_n, -c \le w_i \le c-1\}. \tag{16}$$

In this case, all the lattice points in the set lead to distinct codewords and the codebook size is equal to the number of vectors in $\mathcal{C}_{D_n}$. Hence, all the elements of the lattice points in $\mathcal{C}_{D_n}$ only have allowable values $-c, \ldots 0, \ldots, c-1$, i.e., total $2c$ possible values. For an n-dimensional lattice point, there are $(2c)^n$ possible combinations. By [11], D_n can be obtained by picking up one for every two from $(2c)^n$ points. Therefore, the code size of D_n is $M = \frac{(2c)^n}{2}$. ■

Choice of c. Using (13), we can choose c so that M is a power of 2; the resulting B is then an integer that can be implemented straightforwardly in practical systems. For instance, suppose $n = 3$, $c = 2$, then the code size $M = 2^5$ and code rate $B = 5$, an integer. Note that the choice of c affects not only the code size but also the computational complexity. When c is a positive integer, the Voronoi regions of distinct codewords are of identical shapes due to the wrapping-around property. Take D_2 as an example, the use of $c = 2$ results in 8 distinct codewords as shown in Fig. 3(a). The Voronoi regions of the 8 distinct codewords labelled with numbers from 0 to 7 are of identical shapes, similar to the case of untruncated lattice. For instance, codeword 0 covers the quantization regions originally covered by lattice points $(-2\ 2)$, $(2\ 2)$, and $(2\ -2)$ and hence the Voronoi region of codeword 0 are identical to other codewords. Since all codewords are of identical shapes, the fast algorithm developed for the untruncated D_n in [10] can be used for lattice quantization. Now consider an example where c is not an integer, e.g., $c = \frac{5}{2}$ in Fig. 3(b). There are 13 distinct codewords in this case and 8 of them are in the boundary regions. The Voronoi regions of the boundary lattice points are of different shapes; the fast algorithm in [10] cannot be directly applied and extra computation is needed to deal with those lattice points.

B. Codewords Ordering and Generation

In EGT application, the phase vector is quantized to the closest lattice point (codeword) and the index of the codeword needs to be fed back to the transmitter. For a truncated lattice, the codebook depends on the way truncation is applied. Using the definition of D_n in (6), we can obtain the codewords by generating all the n tuples $(w_1\ w_2 \cdots w_n)$ with $-c \le w_i \le c-1$ and keeping only the vectors such that $w_1 + w_2 \cdots + w_n$ is even. It turns out that the proposed truncation of D_n allows a more efficient codebook generation. To see this we will introduce a systematic ordering of the codewords. Each codeword in the codebook is associated with an index between 0 and $M-1$. Given a codeword the index can be found efficiently; and given an index the codeword can be generated in a real-time manner. Therefore, there is no need of codebook storage at the transmitter and receiver.

The indexing of codewords is given in Algorithm 1. Given a codeword $\mathbf{w}$ from the codebook in (16), we first add c to the elements of $\mathbf{w}$; then $v_i = w_i + c$ satisfies $0 \le v_i \le 2c-1$. Next we multiply each v_i by a weight a_i and then sum them up to form the codeword index s, i.e., $s = \sum_{i=1}^{n} a_i v_i$. The value of a_i can be determined as follows: Beginning with a_2, set all v_i to zero except v_1, i.e., $v_n = v_{n-1} = \cdots v_2 = 0$. In this case, there are $\frac{2c}{2}$ possible lattice points since v_1 has $2c$ possible values, i.e., $0 \le v_i \le 2c-1$, and v_1 needs to be even to satisfy the even-sum condition in (6). Hence, we choose a_2

to be $\frac{2c}{2}$. Now consider a_3. Setting all v_i to zero except v_1 and v_2, i.e., $v_n = v_{n-1} = \cdots v_3 = 0$, there are $\frac{(2c)^2}{2}$ possible lattice points. Hence, the value of a_3 can be $\frac{(2c)^2}{2}$. Using a similar argument, we can obtain the value of a_n to be $\frac{(2c)^{n-1}}{2}$. Therefore, we have a general form for weight $a_i = \frac{(2c)^{i-1}}{2}$. By this general form $a_1 = \frac{1}{2}$. However, the index should be an integer and so is $a_1 v_1$. Hence, we add floor function for $a_1 v_1$. There are $\frac{(2c)^n}{2}$ codewords in the codebook using the proposed codeword ordering, which matches the codebook size in (13).

Algorithm 1: Codewords indexing for $\mathbf{C}_{D_n}$.

```
Suppose the codeword is $\hat{\mathbf{w}} = (\hat{w}_1\ \hat{w}_2 \cdots \hat{w}_n)$.
    1: $v_i = \hat{w}_i + c, i = 1, 2, \ldots n$.
    2: The index $s = \lfloor \frac{v_1}{2} \rfloor + \sum_{i=2}^{n} a_i v_i$, where $a_i = \frac{1}{2}(2c)^{i-1}$.
```

By reversing Algorithm 1, we can generate the codeword given the index, in Algorithm 2. The index s is first divided by a_n to obtain the quotient v_n and residue r_{n-1}. Then, r_{n-1} is divided by a_{n-1} to obtain the quotient v_{n-1} and residue r_{n-2}. Using a similar procedure, we can obtain v_i and r_{i-1} until $i = 2$. Note that when c is a power of 2, we can use a shifter instead of a divider to obtain the quotients and residues. To determine v_1, we notice that $\sum_{i=1}^{n} v_i$ is even if nc is even, and it is odd if nc is odd. This is due to the following two reasons: First, $\sum_{i=1}^{n} v_i = nc + \sum_{i=1}^{n} \hat{w}_i$, where $\sum_{i=1}^{n} \hat{w}_i$ is even since it is a lattice point of D_n. Second, an even value plus an even value results in an even value, and an even value plus an odd value results in an odd value. Therefore, for an even nc, we first check whether $\sum_{i=2}^{n} v_i$ is even; if it is, $v_1 = 2r_1$; otherwise $v_1 = 2r_1 + 1$. On the other hand, for an odd nc, $v_1 = 2r_1 + 1$ if $\sum_{i=2}^{n} v_i$ is even; otherwise $v_1 = 2r_1$. The reconstructed codeword is obtained by $\hat{w}_i = v_i - c$.

Algorithm 2: Generation of codewords from indexes for $\mathbf{C}_{D_n}$

```
Suppose $s$ is the codeword index.
  1: Let $a_i = \frac{1}{2}(2c)^{i-1}$. If $nc$ is even, $u_1 = 0$ and $u_2 = 1$;
     otherwise $u_2 = 0$ and $u_1 = 1$. Compute $v_i$(with the order
     from $i = n$ to $i = 1$) using
       $v_n = \left\lfloor \frac{s}{a_n} \right\rfloor$ and $r_{n-1} = s - a_n v_n$,
       $v_i = \left\lfloor \frac{r_i}{a_i} \right\rfloor$ and $r_{i-1} = r_i - a_i v_i$, for $2 \le i \le n-1$;
       $v_1 = \begin{cases} 2r_1 + u_1, & \text{if } \sum_{i=2}^{n} v_i \text{ is even;} \\ 2r_1 + u_2, & \text{otherwise.} \end{cases}$
  2: The codeword is $\hat{\mathbf{w}} = (\hat{w}_1\ \hat{w}_2 \cdots \hat{w}_n)$, where $\hat{w}_i = v_i - c$.
```

Example 1: **Design example of EGT using D_n lattice**. Let $n = 3$ and $c = 2$ (thus $\alpha = \frac{\pi}{2}$). Using Proposition 1 the corresponding code size is $M = 32$. Let the channel $\mathbf{h}$ be

$$\mathbf{h} = \begin{pmatrix} 1.6895 + 1.4217i \\ 0.2088 - 0.8009i \\ -0.6855 + 2.0595i \\ -0.5601 - 0.3890i \end{pmatrix}.$$

Then, the corresponding phase vector $\boldsymbol{\theta}$ to be quantized is given by

$$\boldsymbol{\theta} = \pi \begin{pmatrix} -0.6415 \\ 0.3796 \\ 0.9706 \end{pmatrix}.$$

The scaled phase vector is

$$\mathbf{x} = \frac{\boldsymbol{\theta}}{\alpha} = (-1.2830\ 0.7592\ 1.9411)^T.$$

Using the fast algorithm in [10] and let $-c \le w_i \le c - 1$, $\mathbf{x}$ is quantized as the lattice point $\hat{\mathbf{w}} = (-1\ 1 - 2)^T$. Correspondingly, $\hat{\boldsymbol{\theta}} = \alpha\hat{\mathbf{w}} = \pi(-\frac{1}{2}\ \frac{1}{2} - 1)^T$. From (4), the optimal beamforming vector is thus given by $\mathbf{p} = (1\ e^{j\frac{\pi}{2}}\ e^{-j\frac{\pi}{2}}\ e^{j\pi})^T = (1\ j - j - 1)^T$.

Now consider the feedback of the quantized phase $\hat{\mathbf{w}}$ to the transmitter. From Algorithm 1 , we can obtain the following parameters:

$$(a_2\ a_3) = (2\ 8),\ \mathbf{v} = (1\ 3\ 0) \text{ and } s = 6$$

where s is the codeword index sent to the transmitter. At the transmitter side, the received index s is decoded using Algorithm 2, resulting in the following parameters:

$$(u_1\ u_2) = (0\ ,1),\ (a_2\ a_3) = (2\ 8), \\ \mathbf{v} = (1\ 3\ 0) \text{ and } \hat{\mathbf{w}} = (-1\ 1 - 2)^T.$$

Using the proposed conversions in Algorithms 1 and 2 and the fast algorithm in [10], there is no need for codebook storage and exhaustive codeword search. ■

V. Phase Quantization Using Lattices D_n^* and E_8

Among the lattices that have existing fast algorithms, we found lattices D_n^* and E_8 have nice results like lattice D_n when the lattices are truncated to have a finite codebook. That is, there is no boundary effect due to lattice truncation, fast algorithms developed for untruncated lattice can be applied directly, and it is very easy to obtain a codebook with integer code rate B.

A. Phase Quantization Using Lattice D_n^*

The lattice D_n^* is called the dual lattice of D_n. It is the union of the n-dimensional integer lattice Z_n and the translate of Z_n by the vector $\mathbf{\frac{1}{2}} = (\frac{1}{2}\ \frac{1}{2} \cdots \frac{1}{2})^T$(see [10] and [11]), i.e.,

$$D_n^* = Z_n \bigcup \left(\mathbf{\frac{1}{2}} + Z_n\right). \tag{17}$$

The lattices D_3^*, D_4^*, and D_5^* are the best lattice quantizers known for dimensions 3, 4, and 5, respectively, in terms of mean-square quantization error if infinite code size is used (see [11, p. 61]). A generation matrix of D_n^* is given by

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & \cdots & 0 & \frac{1}{2} \\ 0 & 1 & \cdots & 0 & \frac{1}{2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & \frac{1}{2} \\ 0 & 0 & \cdots & 0 & \frac{1}{2} \end{pmatrix}_{n \times n}. \tag{18}$$

Similar to the case of D_n in (9), we propose to truncate D_n^* using an $n-$ dimensional cube, i.e.,

$$\mathcal{C}_{D_n^*} = \{\mathbf{w} : \mathbf{w} \in D_n^*, -c \le w_i \le c\} \tag{19}$$

where c is either a positive integer or a positive half integer. We scale θ_i by the formula in (10), and derive the number of distinct codewords in $\mathcal{C}_{D_n^*}$ in the following proposition.

Proposition 2: When c is either a positive integer or a positive half integer, the number of distinct phase vectors represented by the codewords of $\mathcal{C}_{D_n^*}$ in (19), i.e., the code size M, is given by

$$M = c^n 2^{n+1}. \tag{20}$$

Proof: Again, let us first show that if $\mathbf{w}$ is a lattice point of D_n^* with the ith element $w_i = c$, the vector $\mathbf{w}'$, with the ith element $w_i' = -c$ and all the other elements the same as $\mathbf{w}$, is also a lattice point of D_n^*. From (17), all the elements of a lattice point in D_n^* should be either integers or positive half integers. If c is an integer, $-c$ is also an integer; $\mathbf{w}'$ is, hence, still a lattice point in D_n^*. Similar argument can be applied to the case when c is a positive half integer.

Suppose the phase vectors corresponding to $\mathbf{w}$ and $\mathbf{w}'$ are $\boldsymbol{\theta} = \frac{\mathbf{w}}{\alpha}$ and $\boldsymbol{\theta}' = \frac{\mathbf{w}'}{\alpha}$, respectively. Then, $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ represent the same phase. Hence, for either $c \in Z$ or $c \in Z + \frac{1}{2}$, possible values of elements for $\mathbf{w} \in Z_n$ can be $-c, -c+1, \ldots, c-2, c-1$, i.e., a total of $2c$ values; possible values of elements for $\mathbf{w} \in \frac{\mathbf{1}}{\mathbf{2}} + Z_n$ can be $-c, -c+1, \ldots, c-2, c-1$; there are total $2c$ values as well. Thus, there are $(2c)^n$ possible combinations for either $\mathbf{w} \in Z_n$ or $\mathbf{w} \in \frac{\mathbf{1}}{\mathbf{2}} + Z_n$. Therefore, there are a total of $2(2c)^n$ lattice points in the codebook. ■

From (20), if c is chosen as a power of 2, the code rate $B = \log_2 M$ is an integer. Similar to the case of D_n, there is no need to store the codebook. The codewords can be indexed efficiently (Algorithm 3), and conversely given an index the codeword can be generated easily (Algorithm 4).

Algorithm 3: Codewords indexing for $\boldsymbol{C}_{D_n^*}$.

```
Suppose the codeword is $\hat{\mathbf{w}} = (\hat{w}_1\, \hat{w}_2 \cdots \hat{w}_n)$.
1: Compute the two scalars $e$ and $\delta$ by
     $\begin{cases} e = \lfloor c \rfloor \text{ and } \delta = 1, & \text{for } \hat{\mathbf{w}} \in Z_n \\ e = \lceil c \rceil - \frac{1}{2} \text{and } \delta = 0, & \text{for } \hat{\mathbf{w}} \in Z_n + \frac{\mathbf{1}}{\mathbf{2}}. \end{cases}$
2: $v_i = \hat{w}_i + e,\ i = 1, 2, \ldots n.$
3: Compute $s = \delta + 2\sum_{i=1}^{n} a_i v_i$, where $a_i = (2c)^{i-1}$.
```

Algorithm 4: Generation of codewords from indexes for $mbiC_{D_n^*}$.

```
Suppose $s$ is the codeword index.
1: Let $a_i = (2c)^{i-1}$, and let $\delta = 0$ for even $s$ and $\delta = 1$ for
   odd $s$. Compute $v_i$(with order from $i = n$ to $i = 1$) using
     $v_n = \left\lfloor \frac{s-\delta}{2a_n} \right\rfloor$ and $r_{n-1} = \frac{s-\delta}{2} - a_n v_n,$
     $v_i = \left\lfloor \frac{r_i}{a_i} \right\rfloor$ and $r_{i-1} = r_i - a_i v_i,$
       for $1 \le i \le n-1$.
2: The codeword is $\hat{\mathbf{w}} = (\hat{w}_1\, \hat{w}_2 \cdots \hat{w}_n)$, where $\hat{w}_i = v_i - c$.
```

B. Phase Quantization Using Lattice E_8

The lattice E_8 in the even coordinate system is defined as

$$E_8 = \left\{ (w_1\, w_2 \cdots\, w_8) \in Z_8 \text{ or } Z_8 + \frac{1}{2} : \sum_{i=1}^{8} w_i \text{ is even} \right\}. \tag{21}$$

The lattice E_8 in the odd coordinate system can be defined similarly by constraining the sum to be odd instead of even. Note that E_8 is the best lattice quantizer known for dimension $n = 8$ in terms of minimum mean-square quantization error if the code size is infinite (see [11, p. 61]). A generation matrix of E_8 is given by

$$\mathbf{T} = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}. \tag{22}$$

Similar to the case of D_n in (9), we propose to truncate E_8 using an $8-$ dimensional cube, i.e.,

$$\mathcal{C}_{E_8} = \{\mathbf{w} : \mathbf{w} \in E_8, -c \le w_i \le c\} \tag{23}$$

where c is a positive integer. We scale θ_i by the formula in (10) and derive the number of distinct codewords in $\mathcal{C}_{E_8}$ in the following proposition.

Proposition 3: When c is a positive integer, the number of distinct phase vectors represented by the codewords of $\mathcal{C}_{E_8}$ in (23), i.e., the code size M, is given by

$$M = 256c^8. \tag{24}$$

Proof: From the definition in (21), the lattice points in E_8 can be divided into two groups. The first group is the collection of lattice points satisfying

$$(w_1\, w_2 \cdots\, w_8) \in Z_8 : w_1 + w_2 + \cdots + w_8 \text{ is even}, \tag{25}$$

and the second group is the set of lattice points satisfying

$$(w_1\, w_2 \cdots\, w_8) \in Z_8 + \frac{\mathbf{1}}{\mathbf{2}} : w_1 + w_2 + \cdots + w_8 \text{ is even}. \tag{26}$$

The first group is actually equivalent to D_8. From Proposition 1, the number of lattice points in the first group is $\frac{(2c)^8}{2}$. Now, let us examine the second group. All the elements w_i of codewords in the second group are half integer. Possible values of w_i are $-c + \frac{1}{2}, \ldots, -\frac{1}{2}, +\frac{1}{2}, \ldots, c - \frac{1}{2}$, i.e., a total of $2c$ values. To satisfy the condition in (26), we should keep one out of every two lattice points of $Z_8 + \frac{\mathbf{1}}{\mathbf{2}}$. Similarly, the number of lattice points in the second group is $\frac{(2c)^8}{2}$. The total number of lattice points in the first and the second groups is thus $M = (2c)^8$. ■

If c is chosen as a power of 2, the code rate B is an integer. Similar to the case of D_n, the codewords of E_8 can also be indexed efficiently, i.e., see Algorithm 5. Given an index, the corresponding codeword can also be generated using Algorithm 6.

Algorithm 5: Codewords indexing for C_{E_8}.

```
Suppose the codeword is $\hat{\mathbf{w}} = (\hat{w}_1\ \hat{w}_2 \cdots \hat{w}_8)$.
1: Compute the two scalars $e$ and $\delta$ by
     $\begin{cases} e = \lfloor c \rfloor \text{ and } \delta = 1, & \text{for } \hat{\mathbf{w}} \in Z_n; \\ e = \lceil c \rceil - \frac{1}{2} \text{ and } \delta = 0, & \text{for } \hat{\mathbf{w}} \in Z_n + \mathbf{\frac{1}{2}}. \end{cases}$
2: $v_i = \hat{w}_i + e, i = 1, 2, \ldots 8$.
3: The index $s = \delta + 2(\lfloor \frac{v_1}{2} \rfloor + \sum_{i=2}^{8} a_i v_i)$, where
   $a_i = (2c)^{i-1}$.
```

Algorithm 6: Generation of codewords from indexes for $\mathbf{C}_{E_8}$.

```
Suppose $s$ is the codeword index.
1: Let $a_i = (2c)^{i-1}$, and let $\delta = 0$ for even $s$ and $\delta = 1$ for
   odd $s$. Compute $v_i$(starting from $i = 8$ to $i = 1$) using
     $v_8 = \left\lfloor \frac{s-\delta}{2a_8} \right\rfloor \text{ and } r_7 = \frac{s-\delta}{2} - a_8 v_8,$
     $v_i = \left\lfloor \frac{r_i}{a_i} \right\rfloor \text{ and } r_{i-1} = r_i - a_i v_i,\ 2 \le i \le 7;$
     $v_1 = \begin{cases} 2r_1, \text{ if } \sum_{i=2}^{8} v_i & \text{is even;} \\ 2r_1 + 1, & \text{otherwise.} \end{cases}$
2: The codeword is $\hat{\mathbf{w}} = (\hat{w}_1\ \hat{w}_2 \cdots \hat{w}_8)$, where $\hat{w}_i = v_i - c$.
```

VI. MSE Analysis

The MSE of untruncated lattice quantization was analyzed in [11], i.e., infinite code size. However, the MSE in [11] was given as a dimensionless quantity and hence it is independent of the choice of the scaling factor α (see p. 59 in [11]). In this section, we would like to include the scaling factor into the MSE analysis, so that we can see how MSE is related to the code rate B. The average mean-square error per dimension is given by [12]

$$\sigma_\epsilon^2 = \frac{1}{n} \sum_{i=1}^{M} \int_{S_i} \frac{1}{\mathbb{V}(S_i)} ||\boldsymbol{\theta} - \boldsymbol{\theta}_i||^2 \mathbb{P}(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{27}$$

where S_i is the Voronoi region for codeword $\boldsymbol{\theta}_i$, $\mathbb{V}(S_i)$ is the volume of the Voronoi region S_i, and $\mathbb{P}(\boldsymbol{\theta})$ is the probability density function of $\boldsymbol{\theta}$. Assuming that $\mathbb{P}(\boldsymbol{\theta})$ is a constant $\frac{1}{M}$, (27) can be rewritten as

$$\sigma_\epsilon^2 = \frac{1}{nM} \sum_{i=1}^{M} \int_{S_i} \frac{1}{\mathbb{V}(S_i)} ||\boldsymbol{\theta} - \boldsymbol{\theta}_i||^2 d\boldsymbol{\theta}. \tag{28}$$

From the discussion in Sections IV and V, we know the Voronoi regions of all codewords are of identical shape and area when c is chosen properly. Hence, $\mathbb{V}(S_i) = \mathbb{V}_0$, where $\mathbb{V}_0$ is the volume of the Voronoi region of the all-zero codeword. In addition, using the equality [14]

$$\frac{1}{\mathbb{V}_0} \int_{\mathbf{S}_i} ||\boldsymbol{\theta} - \boldsymbol{\theta}_i||^2 d\boldsymbol{\theta} = n \mathbb{V}_0^{\frac{2}{n}} \mathbb{G}(\mathcal{L}) \tag{29}$$

where $\mathbb{G}(\mathcal{L})$ is the second moment of the Voronoi region of lattice $\mathcal{L}$, which could be Z_n (scalar quantization), D_n, D_n^* or E_8. Values of $\mathbb{G}(\mathcal{L})$ for various lattices can be found in [11, p. 61]. The MSE in (28) can be expressed as

$$\sigma_\epsilon^2 = \mathbb{G}(\mathcal{L}) \mathbb{V}_0^{\frac{2}{n}}. \tag{30}$$

By [14], $\mathbb{V}_0$ is equal to the determinant of the generation matrix. As discussed in Section IV-A, a scaled version of the phase vector $\frac{\boldsymbol{\theta}}{\alpha}$ is quantized. This is equivalent to quantizing the phase vector $\boldsymbol{\theta}$ using scaled lattice codewords. That is, the effective generation matrix is $\alpha \mathbf{T}$. Hence, $\mathbb{V}_0$ is given by

$$\mathbb{V}_0 = \alpha^n \det \mathbf{T}. \tag{31}$$

The following proposition summarizes the above discussion.

Proposition 4: The mean-square error of phase quantization using the proposed truncated lattices for D_n, D_n^*, and E_8 can be expressed as

$$\sigma_\epsilon^2 = \alpha^2 \mathbb{G}(\mathcal{L}) (\det \mathbf{T})^{\frac{2}{n}}. \tag{32}$$

■

Let $b = \frac{B}{n}$ be the average number of bits per dimension for quantizing the phase vector. Given σ_ϵ^2, the following corollary computes the required b.

Corollary 1: The required number of bits per dimension b as a function of the mean-square error σ_ϵ^2 for D_n, D_n^* and E_8 can be expressed by

$$b = 1 + \frac{1}{2} \log_2 \mathbb{G}(\mathcal{L}) + \log_2 \pi - \frac{1}{2} \log_2 \sigma_\epsilon^2. \tag{33}$$

Proof: We prove the case for D_n. The proof for other types of lattices can be done using a similar procedure. Combining (10) and (13), we have

$$\alpha = \pi 2^{1 - \frac{1}{n} - b}. \tag{34}$$

For D_n, we know $\det \mathbf{T} = 2$ (see [11, pp. 5 and 117]). From (7), (32), and (34), we have

$$\sigma_\epsilon^2 = \mathbb{G}(D_n) 2^{\frac{2}{n}} (\pi 2^{1 - \frac{1}{n} - b})^2 = \mathbb{G}(D_n) (\pi 2^{1-b})^2. \tag{35}$$

Taking $\log_2$ of both sides of (35), we can arrive at (33). ■

We see that increasing b by one will decrease the mean-square error by 6 dB, which is consistent with the result for scalar quantization in [16]. To compare the mean-square quantization error of lattice quantizers and scalar quantizers, we compute the average mean-square error per dimension for scalar quantizers, which can be shown to be

$$\sigma_\epsilon^2 = \mathbb{G}(Z_n) (\pi 2^{1-b})^2 = \frac{1}{12} (\pi 2^{1-b})^2. \tag{36}$$

The mean-square error for different quantizers depend on $\mathbb{G}(\mathcal{L})$. For the introduced lattice quantizers in this paper, the best known lattice quantizers for $n = 3$, 4, 5, and 8 are D_3^*, $D_4^* \cong D_4$, D_5^* and E_8; the corresponding $\mathbb{G}(\mathcal{L})$ are $\mathbb{G}(D_3^*) = 0.0785$, $\mathbb{G}(D_4^*) = \mathbb{G}(D_4) = 0.0766$, $\mathbb{G}(D_5^*) = 0.0758$ and $\mathbb{G}(E_8) = 0.0717$[11]. When $n = 5$, using D_5^* leads to a 0.4098 dB smaller mean-square error per dimension than scalar quantization; when $n = 8$, using E_8 leads to a 0.6513-dB reduction in mean-square error per

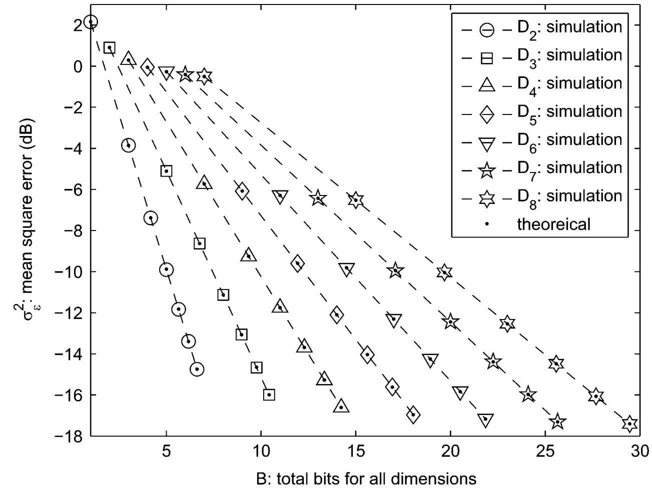

Fig. 4. Mean square error for lattice D_n as a function of B for various dimension n: Theoretical versus simulation results.

dimension. On the other hand, the required number of bits for a given mean-square error can also be used for comparison; that is, from (33), for the same variance the extra number of bits per dimension required for scalar quantizer is given by

$$\frac{1}{2}\log_2\frac{\mathbb{G}(Z_n)}{\mathbb{G}(\mathcal{L})}. \tag{37}$$

For example, scalar quantization requires extra $8\cdot\frac{1}{2}\log_2\frac{\frac{1}{12}}{0.0717}=0.87$ bit to achieve the same MSE as E_8. These results show the advantages of using lattice quantizers in higher dimension quantization.

Example 2: **MSE: Theoretical versus simulation results**. In this example, the theoretical result of σ_ϵ^2 derived in (35) is demonstrated to be the same as the simulation result for lattice D_n. In the simulations, the channel is assumed to have complex Gaussian distribution and $\boldsymbol{\theta}$ is defined as in (5). The fast algorithm in [10] is used to determine the closest codewords. The simulated value of σ_ϵ^2 is obtained by averaging $\|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\|^2$ for 60 000 channel realizations. For the theoretical result, (35) is used and $B=nb=(N_t-1)b$ is the code rate. The theoretical and simulated results are shown in Fig. 4. The simulated result corroborates the theoretical result.

VII. SNR Loss of EGT Using Lattice Quantizers

Let us analyze the SNR loss between the EGT without quantization and the EGT using lattice quantization. Let the number of transmit antennas be N_t. From (5), the number of elements in the phase vector to be quantized is $n=N_t-1$. Define the ith quantization error of phase vector as $\epsilon_i=\hat{\theta}_i-\theta_i$. Assume each element of the channel vector $\mathbf{h}=(h_1\,h_2\cdots h_{N_t})$ is complex Gaussian with zero mean and variance σ_h^2. The instantaneous SNR of the unquantized EGT is as given in (3). According to the encoding procedure for lattice D_n in Section IV-A, we may argue that for each beamforming vector only one element of ϵ_i depends on $\mathbf{x}$ and the other elements should be uncorrelated and uniformly distributed. When n grows larger, the effect of correlation becomes less pronounced and the distribution of ϵ_i tends to be uniform. Thus, we may assume that ϵ_i is uniformly distributed and is uncorrelated for different i. Similar arguments can be applied to lattices D_n^* and E_8. Let $\widetilde{\rho}$ be the instantaneous SNR of EGT with quantized phase vector; it is shown in the Appendix A that the expectation of $\widetilde{\rho}$ may be approximated by

$$\mathbb{E}\{\tilde{\rho}\}\approx\frac{\sigma_x^2\sigma_h^2}{\sigma_\epsilon^2}\Big(1+c_1\frac{\pi}{4}\mathbb{E}\{\cos(\epsilon_j)\}+c_2\frac{\pi}{4}\mathbb{E}\{\cos(\epsilon_j-\epsilon_i)\}\Big),\ i\neq j \tag{38}$$

where $c_1=\frac{2(N_t-1)}{N_t}$, $c_2=\frac{(N_t-1)(N_t-2)}{N_t}$. When $\epsilon_i=0$, the above equation reduces to the case of no quantization, i.e., $\mathbb{E}\{\tilde{\rho}\}=\mathbb{E}\{\rho\}=\frac{\sigma_x^2\sigma_h^2}{\sigma_\epsilon^2}(1+\frac{\pi}{4}(N_t-1))$ [5]. The quantities $\mathbb{E}\{\cos(\epsilon_j-\epsilon_i)\}$ and $\mathbb{E}\{\cos(\epsilon_i)\}$ in (38) can be approximated as follows. Using the Taylor series expansion, we can approximate $\mathbb{E}\{\cos(\epsilon_j-\epsilon_i)\}$ as

$$\mathbb{E}\{\cos(\epsilon_j-\epsilon_i)\}\approx\sum_{k=0}^{K}(-1)^k\frac{\mathbb{E}\{(\epsilon_j-\epsilon_i)^{2k}\}}{(2k)}. \tag{39}$$

It is shown in Appendix B that

$$\mathbb{E}\{(\epsilon_j-\epsilon_i)^{2k}\}=\sigma_\epsilon^{2k}\sum_{m=0}^{k}\binom{2k}{2m}\frac{3^k}{(2m+1)(2(k-m)+1)}. \tag{40}$$

Using (39) and (40), $\mathbb{E}\{\cos(\epsilon_j-\epsilon_i)\}$ can be approximated as a function of σ_ϵ^{2k} given by

$$\sum_{k=0}^{K}\frac{(-1)^k\sigma_\epsilon^{2k}}{(2k)}\sum_{m=0}^{k}\binom{2k}{2m}\frac{3^k}{(2m+1)(2(k-m)+1)}. \tag{41}$$

We have verified by the Monte Carlo simulation that using $K=3$ in (41), we can achieve a satisfactory approximation. Similarly using the Taylor series expansion and (53) in Appendix B, $\mathbb{E}\{\cos(\epsilon_i)\}$ can be approximated as

$$\sum_{k=0}^{K}(-1)^k\frac{\mathbb{E}\{\epsilon_i^{2k}\}}{(2k)}=\sum_{k=0}^{K}\frac{3^k}{2k+1}\frac{(-1)^k\sigma_\epsilon^{2k}}{(2k)}. \tag{42}$$

Proposition 5: The SNR loss of the EGT due to lattice quantization can be approximated by

$$\frac{\mathbb{E}\{\rho\}}{\mathbb{E}\{\tilde{\rho}\}}\approx\frac{1+\frac{\pi}{4}(N_t-1)}{1+\frac{\pi}{4}c_3} \tag{43}$$

where

$$c_3=(c_1+c_2)-(0.5c_1+c_2)\sigma_\epsilon^2+(0.075c_1+0.4c_2)(\sigma_\epsilon^2)^2-(0.0054c_1+0.0857c_2)(\sigma_\epsilon^2)^3,$$

and σ_ϵ^2 depends on the type of lattice, given by $\sigma_\epsilon^2=\mathbb{G}(\mathcal{L})(\pi 2^{1-b})^2$(see (35)).

Proof: Letting $K=3$ in (41), we have

$$\mathbb{E}\{\cos(\epsilon_j-\epsilon_i)\}\approx 1-\sigma_\epsilon^2+0.4(\sigma_\epsilon^2)^2-0.0875(\sigma_\epsilon^2)^3. \tag{44}$$

From (42) and letting $K=3$, we have

$$\mathbb{E}\{\cos(\epsilon_i)\}\approx 1-0.5\sigma_\epsilon^2+0.075(\sigma_\epsilon^2)^2-0.0054(\sigma_\epsilon^2)^3. \tag{45}$$

From (38), (44), and (45), and the derivation of the SNR for EGT without quantization in [5], the SNR loss due to the lattice quantization can be calculated as in (43).

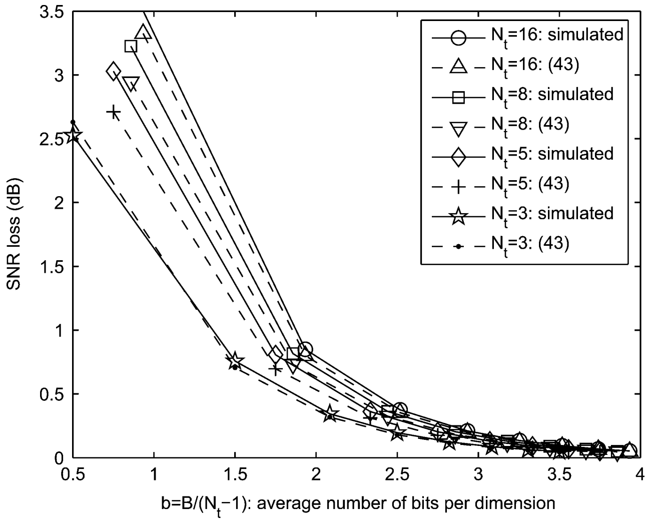

Fig. 5. SNR loss as a function of b for different N_t: theoretical result versus simulated result.

Note that when b tends to ∞, $\sigma_\epsilon^2 = 0$ and thus $c_3 = c_1 + c_2 = N_t - 1$. In this case, the SNR loss is 0 dB. Also, for $b \geq 3$, we may approximate c_3 to the first order of σ_ϵ^2 and the corresponding SNR loss can be approximated by $10\log_{10}\frac{1+\frac{\pi}{4}(N_t-1)}{1+\frac{\pi}{4}(N_t-1)\left(1-\frac{N_t-1}{N_t}\sigma_\epsilon^2\right)}$.

VIII. Simulation Result

In the simulations, the channel coefficients are i.i.d. complex Gaussian random variables with zero mean and unit variance. One receive antenna is used. The symbol transmitted is rectangular 16-QAM with equally probable constellation points (see [17, p. 175]). There are 60 ,000 different channel realizations used, and 6000 different 16-QAM symbols are used in each channel. The SNR is defined as $\frac{\sigma_x^2}{\sigma_\epsilon^2}$. The number of total feedback bits is B and the dimension of phase vectors is $n = N_t - 1$. The average number of bits per dimension is $b = \frac{B}{n}$.

Example 3: **SNR loss of EGT due to quantization effect**. The SNR loss of the EGT using lattice D_n, as a function of b for different N_t is shown in Fig. 5. The theoretical result is obtained by (43). The theoretical result is close to the simulation result; hence (43) can be used as a useful reference in practical design. Moreover, the SNR loss is within 1 dB for $b \geq 2$. As b increases, the performance gap becomes smaller.

The bit error rate (BER) performance of D_n with different B for $N_t = 3$, 5, 8, and 16 is shown in Fig. 6. The code rate B is chosen according to (13) for $c = 2$ and $c = 4$, which corresponds to $B = 2(N_t - 1) - 1$ and $B = 3(N_t - 1) - 1$, respectively. The SNR loss in Fig. 6 generally matches the result in Fig. 5. For instance consider $N_t = 5$. The code rates are $B = 7$ for $c = 2$ and $B = 11$ for $c = 4$. The corresponding SNR losses are around 0.8 and 0.2 dB, respectively, observed from Fig. 6 at BER $= 10^{-5}$. In Fig. 5, the SNR losses are around 0.8 dB for $N_t = 5$ and $B = 7$, and 0.2 dB for $N_t = 5$ and $B = 11$ (see the diamond-curve for $\frac{B}{(N_t-1)} = \frac{7}{4} = 1.75$ and $\frac{B}{(N_t-1)} = \frac{11}{4} = 2.75$, respectively). This result shows that the analysis of SNR loss can be a satisfactory estimation for the BER performance (see also [2] and [18]). Note that in the BER simulation, Gray mapping is used and thus if symbol error rate performance P_s is needed, it can be approximated by bit error rate P_b using the relationship $P_s \approx 4P_b$.

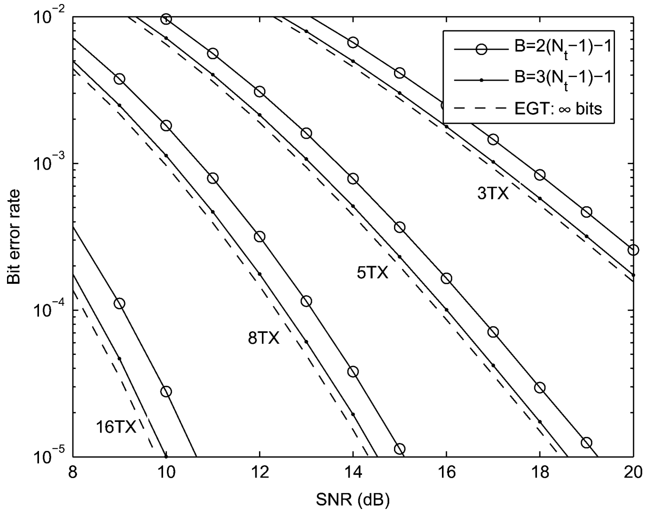

Fig. 6. Quantization effect using lattice D_n.

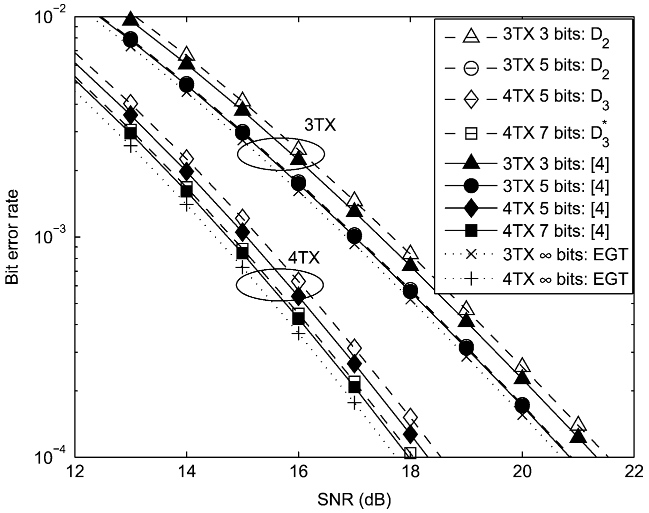

Fig. 7. BER comparison between lattices and Lloyd vector quantizations [4] in 3T1R and 4T1R EGT beamforming.

Example 4: **BER performance**. In this example, we compare the proposed lattice quantization with codebook design using Lloyd algorithm in [4]. The codebooks generated by Lloyd algorithm can in general achieve nearly minimum mean-square quantization error in vector quantization [4], [9]. The BER comparison is shown in Fig. 7, for $N_t = 3$ and $N_t = 4$. The figure shows that the performance gap between these two quantization schemes is small. With minor performance loss, the proposed lattice quantizations enjoy great implementational advantages. That is, codebook storage in both transmitter and receiver is not required. Also, there is no need to search exhaustively to find the closest codeword.

We also compare the proposed lattice codebook with other vector quantization schemes including the Grassmannian codebook in [2] and [21], and the DFT-based EGT codebook in [3]. The simulation result is shown in Fig. 8. We observe that for

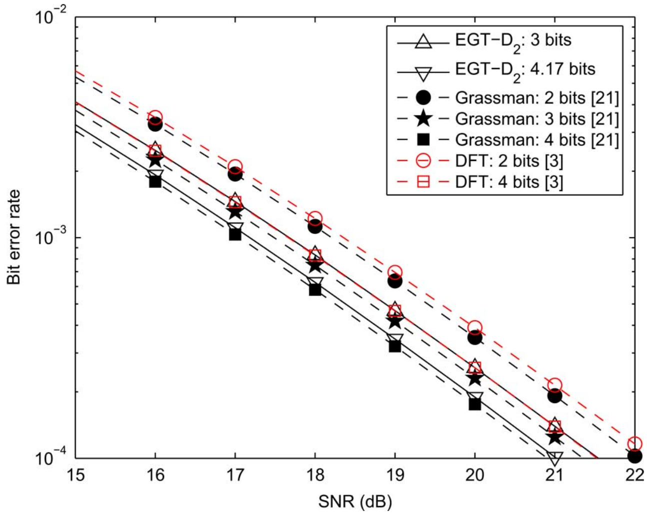

Fig. 8. Comparison of various beamforming schemes in a 3T1R channel environment.

$B = 3$, the proposed D_2 EGT codebooks performs only about 0.2 dB worse than the Grassmannian codebook, which quantizes both magnitude and phase information; also D_2 lattice with $B = 4.17$ performs about 0.1 dB worse than the Grassmannian codebook with $B = 4$. Moreover, the lattice D_2 with $B = 3$ can achieve nearly the same performance as DFT-based EGT codebook with $B = 4$.

Example 5: **EGT with large numbers of transmit antennas**. In some modern MIMO systems, a large number of transmit antennas is used to increase the diversity or multiplexing gain [19]. To achieve full diversity order for EGT, it is suggested in [3] that around at least one bit per transmit antenna is needed. Hence, the number B of quantization bits needs to be increased as the number of transmit antennas increases. For a large B, the design of codebook-based EGT is more difficult and even impractical since it needs large memory size to store codebooks and exhaustive-search computations. In this case, non-codebook-based quantizations such as scalar quantization and the proposed lattice quantizations are more suitable. This example evaluates non-codebook-based quantizations for EGT with large numbers of transmit antennas.

The BER performance of EGT with scalar and the proposed lattice quantizations is shown in Fig. 9, for $N_t = 9$ and $N_t = 16$. When $N_t = 9$ and $B = 8$, the proposed lattice E_8 outperforms scalar quantization by around 0.75 dB. With one-bit increase in B, i.e., $\frac{1}{9}$-bit increase per transmit antenna, D_8^* with $B = 9$ has a gain of around 2 dB, compared to scalar quantization with $B = 8$ at BER $= 10^{-5}$. Note that we could not use $B = 9$ for scalar quantization when $N_t = 9$. This is because for scalar quantization, each antenna uses the same number of bits for quantization; hence B in scalar quantization is a multiple of $(N_t - 1)$, i.e., 8, 16, 24, $\cdots$ for $N_t = 9$. By using a combination of different lattices, we can have a more flexible code rate. Let $N_t = 16$. We also show the BER of D_n^* for $B = 16$, and scalar quantization for $B = 15$. In the same figure, one-bit increase in B, i.e., $\frac{1}{16}$-bit increase per transmit antenna, D_{15}^* with $B = 16$ can result in a 1.5-dB performance gain, compared to scalar quantization with $B = 15$ at BER $= 10^{-5}$.

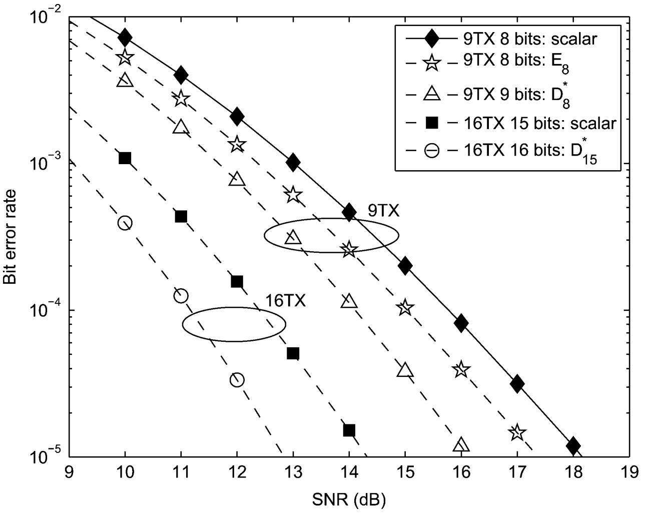

Fig. 9. BER comparison between lattices and scalar quantizations in 9T1R and 16T1R EGT beamforming.

IX. Conclusion

We proposed truncations of various lattices for quantizing phase vectors in EGT systems. There is no boundary effect using the proposed truncations and integer code rate can be easily obtained. In addition, fast algorithms can be used to quantize phase vectors; hence there is no need to search exhaustively and the feedback latency is small. In addition, we found the codewords can be ordered efficiently so that each codeword is associated with a specified codeword index. That is, given a codeword, the index can be obtained with few computations. Conversely given an index, the codeword can be generated easily. As a result, codebook storage is not needed at the transmitter and receiver. Moreover, the mean-square quantization error was analyzed for several truncated lattices. We also showed how to analyze the SNR loss when the proposed quantization is applied to EGT system. Simulation results showed that EGT with lattice quantization can achieve satisfactory performance comparable to Lloyd EGT beamforming; also, the use of lattices not only outperforms scalar quantization but also provides more flexible number of code sizes. Furthermore, the application of the proposed lattice quantizer for phase vectors is not limited to EGT only; it can be used in any system that requires quantization of phase vectors. For instance, the compressed beamforming in [20] decomposes the singular vectors using Givens rotation. The signals to be quantized are also phase vectors. Finally, it is interesting to observe that equal-gain Grassmannian codebooks have regular structure. For instance, the codebook with $N_t = 4$ and $M = 64$ in [21] is an equal gain codebook. This codebook has regular structure and the 4×1 codeword can be generated by $\frac{1}{2}\left(e^{j\frac{2\pi}{64}28k}\ e^{j\frac{2\pi}{64}16k}\ e^{j\frac{2\pi}{64}32k}\ e^{j\frac{2\pi}{64}k}\right)^T$, $0 \leq k \leq 63$. Although such codebooks cannot be generated by truncating the three types of lattices discussed in this paper, it is still interesting to explore whether such codebooks have lattice-like structure, and this problem is still open.

$$\begin{aligned}\mathbb{E}\left\{\tilde{\rho}\right\} &= \frac{1}{N_t}\frac{\sigma_x^2}{\sigma_e^2}\mathbb{E}\left\{\sum_{i=1}^{N_t}|h_i|^2 + \sum_{i=1}^{N_t}\sum_{j\neq i}^{N_t} h_i^* h_j e^{j(\hat{\theta}_j-\hat{\theta}_i)}\right\}\\ &= \frac{1}{N_t}\frac{\sigma_x^2}{\sigma_e^2}\mathbb{E}\left\{\sum_{i=1}^{N_t}|h_i|^2 + 2\Re\left\{\sum_{i=1}^{N_t}\sum_{j>i}^{N_t} h_i^* h_j e^{j(\hat{\theta}_j-\hat{\theta}_i)}\right\}\right\}\\ &= \frac{1}{N_t}\frac{\sigma_x^2}{\sigma_e^2}\left(\sum_{i=1}^{N_t}\mathbb{E}\left\{|h_i|^2\right\} + 2\sum_{i=1}^{N_t}\sum_{j>i}^{N_t}\mathbb{E}\left\{\Re\left\{h_i^* h_j e^{j(\hat{\theta}_j-\hat{\theta}_i)}\right\}\right\}\right). \qquad (46)\end{aligned}$$

Appendix

A. Proof of (38)

See (46) at the top of the page. Let $h_i = |h_i|e^{j\angle h_i}$, and define the quantization error of phase as $\epsilon_i = \hat{\theta}_i - \theta_i$, $1 \le i \le N_t$, where θ_i is the optimal phase. In MISO channels, $\theta_i = -(\angle h_i - \angle h_1)$, and thus $\hat{\theta}_i = -(\angle h_i - \angle h_1) + \epsilon_i$. Since $\hat{\theta}_1 = 0$ and $\epsilon_1 = 0$, we separate the terms for $i = 1$ and $i \neq 1$ as follows:

$$\begin{aligned}\mathbb{E}\left\{\tilde{\rho}\right\} = \frac{1}{N_t}\frac{\sigma_x^2}{\sigma_e^2}\Bigg(&\sum_{i=1}^{N_t}\mathbb{E}\left\{|h_i|^2\right\}\\ &+2\sum_{j>1}^{N_t}\mathbb{E}\left\{\Re\left\{h_1^* h_j e^{-j(\angle h_j-\angle h_1)}e^{j\epsilon_j}\right\}\right\}\\ &+2\sum_{i=2}^{N_t}\sum_{j>i}^{N_t}\mathbb{E}\left\{\Re\left\{h_i^* h_j e^{-j(\angle h_j-\angle h_i)}e^{j(\epsilon_j-\epsilon_i)}\right\}\right\}\Bigg). \qquad (47)\end{aligned}$$

Assume the quantization error is independent of channel realization and using the fact that $h_i^* h_j e^{-j(\angle h_j-\angle h_i)}$ is real, we have

$$\begin{aligned}\mathbb{E}\left\{\tilde{\rho}\right\} = \frac{1}{N_t}\frac{\sigma_x^2}{\sigma_e^2}\Bigg(&\sum_{i=1}^{N_t}\mathbb{E}\{|h_i|^2\} + 2\sum_{j>1}^{N_t}\mathbb{E}\{|h_1^* h_j|\}\,\mathbb{E}\{\Re\left\{e^{j\epsilon_j}\right\}\}\\ &+2\sum_{i=2}^{N_t}\sum_{j>i}^{N_t}\mathbb{E}\left\{|h_i^* h_j|\right\}\mathbb{E}\left\{\Re\left\{e^{j(\epsilon_j-\epsilon_i)}\right\}\right\}\Bigg). \qquad (48)\end{aligned}$$

Using the assumption that ϵ_i tends to be uniformly distributed and uncorrelated for different i as N_t grows, and using the property that $\frac{\mathbb{E}\{|h_i h_j|\}}{\mathbb{E}\{|h_i|^2\}} = \frac{\pi}{4}$ [5], we can obtain the approximation in (38). ■

B. Proof of (40)

By using the binomial theorem, we have

$$\mathbb{E}\left\{(\epsilon_j - \epsilon_i)^{2k}\right\} = \sum_{m=0}^{2k}\binom{2k}{m}\mathbb{E}\left\{\epsilon_i^m \epsilon_j^{2k-m}\right\}. \qquad (49)$$

Assume ϵ_i is uncorrelated for $i \neq j$, $\mathbb{E}\left\{\epsilon_i^m \epsilon_j^{2k-m}\right\} = \mathbb{E}\left\{\epsilon_i^m\right\}\mathbb{E}\left\{\epsilon_j^{2k-m}\right\}$. Assume ϵ_i is uniformly distributed in $[-\frac{\Delta}{2}\,\frac{\Delta}{2}]$. The probability density function of ϵ_i^m is an even function; hence $\mathbb{E}\left\{\epsilon_i^m\right\} = 0$, for m is odd; and $\mathbb{E}\left\{\epsilon_i^m \epsilon_j^{2k-m}\right\} = 0$, for m is odd. We can rewrite (49) as

$$\begin{aligned}\mathbb{E}\left\{(\epsilon_j - \epsilon_i)^{2k}\right\} &= \sum_{m=0,\ m\ even}^{2k}\binom{2k}{m}\mathbb{E}\left\{\epsilon_i^m\right\}\mathbb{E}\left\{\epsilon_j^{2k-m}\right\}\\ &= \sum_{m=0}^{k}\binom{2k}{2m}\mathbb{E}\left\{\epsilon_i^{2m}\right\}\mathbb{E}\left\{\epsilon_j^{2k-2m}\right\}. \qquad (50)\end{aligned}$$

Using the property that the kth moment (k is even) of a random variable x uniformly distributed in $[a\ b]$ is

$$\mathbb{E}\left\{x^k\right\} = \frac{1}{k+1}\sum_{i=0}^{k} a^i b^{k-i}. \qquad (51)$$

The $2m$th moment of ϵ_i, i.e., $\mathbb{E}\left\{\epsilon_i^{2m}\right\}$, can be written as

$$\frac{1}{2m+1}\left(\frac{\Delta}{2}\right)^{2m}\sum_{i=0}^{2m}(-1)^{2m-i} = \frac{1}{2m+1}\left(\frac{\Delta}{2}\right)^{2m}. \qquad (52)$$

Since ϵ_i is uniformly distributed in $[-\frac{\Delta}{2}\,\frac{\Delta}{2}]$, the 2nd moment of ϵ_i is $\sigma_\epsilon^2 = \mathbb{E}\left\{\epsilon_i^2\right\} = \frac{1}{12}\Delta^2$. We can rewrite (52) as

$$\mathbb{E}\left\{\epsilon_i^{2m}\right\} = \frac{3^m}{2m+1}\sigma_\epsilon^{2m}. \qquad (53)$$

Substituting (53) into (50), we have the equality in (40). ■

References

[1] K. K. Mukkavilli, A. Sabharwal, E. Erkip, and B. Aazhang, "On beamforming with finite rate feedback in multiple antenna systems," *IEEE Trans. Inf. Theory*, vol. 49, pp. 2562–2579, Oct. 2003.

[2] D. J. Love, R. W. Heath, and T. Strohmer, "Grassmannian beamforming for multiple-input multiple-output wireless systems," *IEEE Trans. Inf. Theory*, vol. 49, pp. 2735–2747, Oct. 2003.

[3] D. J. Love and R. W. Heath, "Equal gain transmission in multiple-input multiple-out wireless systems," *IEEE Trans. Commun.*, vol. 51, pp. 1102–1110, Jul. 2003.

[4] C. R. Murthy and B. D. Rao, "Quantization methods for equal gain transmission with finite rate feedback," *IEEE Trans. Signal Process.*, vol. 55, no. 1, pp. 233–245, Jan. 2007.

[5] S.-H. Tsai, "Transmit equal gain precoder in Rayleigh fading channels," *IEEE Trans. Signal Process.*, vol. 57, pp. 3717–3721, Sep. 2009.

[6] X. Zheng, Y. Xie, J. Li, and P. Stoica, "MIMO transmit beamforming under uniform elemental power constraint," *IEEE Trans. Signal Process.*, vol. 55, pp. 5395–5406, Nov. 2007.

[7] P. Xia, S. Zhou, and G. B. Giannakis, "Achieving the welch bound with difference sets," *IEEE Trans. Inf. Theory*, vol. 51, pp. 1900–1907, May 2005.

[8] C. Ding, "Complex codebooks from combinatorial designs," *IEEE Trans. Inf. Theory*, vol. 52, pp. 4229–4235, Sep. 2006.

[9] K. Sayood, *Introduction to Data Compression*. San Mateo, CA: Morgan Kaufmann, 2000.

[10] J. H. Conway and N. J. A. Sloane, "Fast quantization and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inf. Theory*, vol. 28, pp. 227–232, Mar. 1982.

[11] J. H. Conway and N. J. A. Sloane, *Shpere Packings, Lattices and Groups*. New York: Springer, 1998.

[12] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inf. Theory*, vol. 25, pp. 373–380, Jul. 1979.

[13] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.

[14] J. H. Conway and N. J. A. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE Trans. Inf. Theory*, vol. 28, pp. 211–226, Mar. 1982.

[15] J. H. Conway and N. J. A. Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inf. Theory*, vol. 29, pp. 820–824, Nov. 1983.

[16] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[17] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2001.

[18] P. A. Dighe, R. K. Mallink, and S. S. Jamuar, "Analysis of transmit-receiver diversity in rayleigh fading," *IEEE Trans. Commun.*, vol. 51, pp. 694–703, Apr. 2003.

[19] A. Pidwerbetsky, "Application of BLAST to tactical communications," *Proc. IEEE MILCOM 2004*, pp. 510–516, Oct. 2004.

[20] *Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput*, IEEE Standard 802.11n, 2009.

[21] Grassmannian codebook [Online]. Available: http://cobweb.ecn.purdue.edu/~djlove/grass

Shang-Ho (Lawrence) Tsai (S'04–M'06) was born in Kaohsiung, Taiwan, R.O.C., in 1973. He received the Ph.D. degree in electrical engineering from the University of Southern California (USC), Los Angeles, in 2005.

From June 1999 to July 2002, he was with the Silicon Integrated Systems Corporation (SiS), where he participated in the VLSI design for DMT-ADSL systems. From September 2005 to January 2007, he was with the MediaTek Incorporated (MTK) and participated in the VLSI design for MIMO-OFDM systems. In February 2007, he joined the Department of Electrical and Control Engineering (now the Department of Electrical Engineering) at the National Chiao Tung University, Hsinchu, Taiwan, R.O.C., where he is currently an Associate Professor. His research interests include signal processing for communications, statistical signal processing, and signal processing for VLSI designs.

Dr. Tsai was awarded a government scholarship for overseas study from the Ministry of Education, Taiwan, in 2002–2005.

Yuan-Pei Lin (S'93–M'97–SM'03) was born in Taipei, Taiwan, R.O.C., in 1970. She received the B.S. degree in control engineering from the National Chiao-Tung University, Taiwan, R.O.C., in 1992 and the M.S. and Ph.D. degrees, both in electrical engineering, from the California Institute of Technology, Pasadena, in 1993 and 1997, respectively.

She joined the Department of Electrical and Control Engineering of National Chiao-Tung University in 1997. Her research interests include digital signal processing, multirate filter banks, and signal processing for digital communications. She has coauthored two books: *Signal Processing and Optimization for Transceiver Systems* (Cambridge Univ. Press, 2010) and *Filter Bank Transceivers for OFDM and DMT Systems* (Cambridge Univ. Press, 2010).

Dr. Lin is a recipient of the 2004 Ta-You Wu Memorial Award. She served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, the IEEE SIGNAL PROCESSING LETTERS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I, the *EURASIP Journal on Applied Signal Processing*, and *Multidimensional Systems and Signal Processing*. She was a Distinguished Lecturer of the IEEE Circuits and Systems Society for 2006–2007.