Designing Voronoi Constellations to Minimize Bit Error Rate

Shen Li*, Ali Mirani†, Magnus Karlsson‡, and Erik Agrell*

*Dept. of Electrical Engineering, Chalmers University of Technology, 412 96 Gothenburg, Sweden
†Dept. of Microtechnology and Nanoscience, Chalmers University of Technology, 412 96 Gothenburg, Sweden
Email: {shenl,mirani,magnus.karlsson,agrell}@chalmers.se

Abstract—In a classical 1983 paper, Conway and Sloane presented fast encoding and decoding algorithms for a special case of Voronoi constellations (VCs), for which the shaping lattice is a scaled copy of the coding lattice. Feng generalized their encoding and decoding methods to arbitrary VCs. Kurkoski proposed encoding and decoding algorithms called “rectangular encoding” in [18] for VCs whose shaping lattice and coding lattice both have triangular generator matrices. These algorithms are equally simple but less general than Feng’s algorithms, and are applicable to a variety of coding lattices and shaping lattices.

Conway and Sloane defined VCs in 1983 by selecting a finite set of points of a lattice (possibly translated) that belong to a scaled-up version of its Voronoi region [12]. They presented simple and elegant algorithms for encoding and decoding VCs, i.e., mapping integers to constellation points and vice versa. Mirani et al. designed multidimensional scaled VCs with up to $10^{28}$ constellation points utilizing these algorithms for the AWGN channel and nonlinear fiber channel, and showed significant bit error rate (BER) gains over QAM in uncoded systems [13].

Forney generalized the concept in 1989 by considering two possibly different lattices, a coding lattice and a shaping lattice. The VCs are constructed by selecting the points of the (translated) coding lattice that belong to a the Voronoi region of the shaping lattice [14]. The only constraint is that the shaping lattice is a sublattice of the coding lattice. A method to enumerate the constellation points in an arbitrary VC was presented by Feng et al. [15]. This enumeration admits very fast encoding and decoding algorithms, which are reviewed by Zamir in [16, Ch. 9], [17].

Kurkoski proposed encoding and decoding algorithms called “rectangular encoding” in [18] for VCs whose shaping lattice and coding lattice both have triangular generator matrices. These algorithms are equally simple but less general than Feng’s algorithms, and are applicable to a variety of coding lattices and shaping lattices.

Ferdinand et al. proposed a two-step “systematic Voronoi shaping” method in [19], based on the concept of “systematic shaping” proposed by Sommer et al. in [20]. A coding lattice defined by a lower-triangular parity-check matrix and a shaping lattice satisfying certain constraints related to the coding lattice were combined to achieve high coding and shaping gains, and the symbol error rate (SER) performance was evaluated. For the shaping step, the algorithms to map the integers to points in VCs with a cubic coding lattice and vice versa were explicitly described in [21].

As far as we know, no study apart from the one by Mirani et al. [13] has been reported about designing VCs with good BER performance and low complexity. Hence, in this paper, we focus on the VCs with a cubic coding lattice and try to minimize the BER both in uncoded and coded systems. Although such VCs have no coding gain, the high shaping gain is maintained, and their decoding is much simpler than for VCs with rescaled coding and shaping lattices [12], [13]. They also allow more flexibility in spectral efficiencies for mapping
Integers to bits. The considered VCs in this paper are very large, with up to \(9 \times 10^{46}\) points, but they are nevertheless useful in communications with moderate complexity, since none of the involved algorithms need to store or search all constellation points. We apply pseudo-Gray labeling on top of Kurkoski’s encoding and decoding. Our designed VCs show better BER performance than Feng’s and Ferdinand’s algorithms for the same VCs in uncoded systems. Also, in combination with a low-density parity-check (LDPC) code, the designed VCs can have better BER performance than the scaled VCs, due to the efficient pseudo-Gray labeling for the cubic coding lattice.

**Notation:** Bold lowercase symbols denote row vectors and bold uppercase symbols denote matrices. The elements of a vector \(u\) are denoted by \(u_i\), the rows of a matrix \(P\) are denoted by \(p_i\), and the elements of a matrix \(P\) are denoted by \(P_{ij}\). Uppercase Greek or calligraphic letters denote sets.

## II. Preliminaries

Given a set of \(n\) linearly independent basis vectors, a lattice \(\Lambda\) is the set of all linear combinations of these vectors with integer coefficients. If the basis vectors are arranged row-wise into a matrix \(G\), then the lattice is

\[
\Lambda \triangleq \{ uG : u \in \mathbb{Z}^n \}. \tag{1}
\]

Without loss of generality, we assume that the generator matrix has dimension \(n \times n\). From the definition, any lattice includes the all-zero vector \(0\).

The generator matrix of a given lattice \(\Lambda\) is not unique. Two generator matrices \(G\) and \(G'\) generate the same lattice if and only if \(G' = UG\), where \(U\) is an integer matrix with determinant \(±1\) [22, p. 10].

The **fundamental Voronoi region** of a lattice \(\Lambda\) is the set of vectors in Euclidean space having the all-zero vector as its closest lattice point, i.e.,

\[
\Omega(\Lambda) \triangleq \{ x \in \mathbb{R}^n : \| x \| \leq \| x - \lambda \|, \forall \lambda \in \Lambda \}. \tag{2}
\]

Given an \(n\)-dimensional coding lattice \(\Lambda\), an \(n\)-dimensional shaping lattice \(\Lambda_s\) which is a sublattice of \(\Lambda\), and an offset vector \(a \in \mathbb{R}^n\), a **Voronoi constellation** (VC) in its general form defined by Forney [14] is

\[
\Omega(\Lambda) \triangleq \{ x : x \in \mathbb{Z}^n \}\cap \Omega(\Lambda_s). \tag{3}
\]

We assume that no points in \(\Lambda - a\) fall on the boundary of \(\Omega(\Lambda_s)\). The number of points in the VC is

\[
M \triangleq |\Gamma| = \frac{|\det G_s|}{|\det G|}. \tag{4}
\]

where \(G_s\) is a generator matrix of \(\Lambda_s\). This relation can be verified by recognizing \(|\det G|\) and \(|\det G_s|\) as the volumes of \(\Omega(\Lambda)\) and \(\Omega(\Lambda_s)\), respectively [22, p. 4].

Fig. 1 illustrates a two-dimensional VC of lattice partition \(\mathbb{Z}^2/2D_2\), where \(D_2\) is the two-dimensional checkerboard lattice. In this simple example, we can choose the generator matrices of \(\Lambda\) and \(\Lambda_s\) to be

\[
G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad G_s = \begin{bmatrix} 4 & 0 \\ 2 & 2 \end{bmatrix}, \tag{5}
\]

and there are \(M = 8\) constellation points, with \(a = (-\frac{1}{2}, 0)\). The offset vector \(a\) is optimized to minimize the average symbol energy of the constellation, and can be obtained using an iterative algorithm given by [12]. The algorithm may converge to a suboptimal vector for large-size constellations, but as the constellation size increases, the performance difference between VC generated using optimal \(a\) and a random \(a \in \Omega(\Lambda)\) decreases, and can be neglected for large VCs [13, Fig. 3]. In this paper, for a small or moderate-size VC (\(M \leq 2^{17} \approx 1.3 \times 10^5\)), \(a\) was optimized using the method in [12]; and for a very large VC where we can only approximate the average symbol energy by Monte Carlo simulations, we selected a random \(a\) uniformly in \(\Omega(\Lambda)\).

## III. Encoding and Decoding

We adopt Kurkoski’s algorithms for encoding and decoding, and also review Feng’s algorithms in this section.

**Feng’s algorithms:** Given \(\Lambda\) and its sublattice \(\Lambda_s\) with their generator matrices \(G\) and \(G_s\), respectively, since all basis vectors of \(\Lambda_s\) belong to \(\Lambda\), there exists by definition (1) an integer matrix \(U\) such that \(G_s = UG_s\) for any choice of \(\Lambda\) and \(\Lambda_s\). Then \(U\) has a Smith normal form [23, Ch. 15]

\[
J \triangleq \det(J_{11}, \ldots, J_{nn}) = STU, \tag{6}
\]

where \(S\) and \(T\) are integer matrices with determinant \(±1\) and \(J_{ii} \in \mathbb{Z}^+\) for all \(i = 1, \ldots, n\). Then alternative generator matrices of \(\Lambda\) and \(\Lambda_s\) can be constructed as \(G' = T^{-1}G\) and \(G'_s = SG_s\), with the relation [15, Th. 6]

\[
G'_s = JG'. \tag{7}
\]
Algorithm 1 Feng’s encoding [15]
Preprocessing: Given generator matrices $G$ and $G_s$, find two integer matrices $S$ and $T$ with determinant $±1$ such that $J = SG_sG_s^{-1}T$ is the Smith normal form of $G_sG_s^{-1}$. Then let $G' = T^{-1}G$.
1: Let $x \leftarrow uG' - a$
2: Let $z \leftarrow \arg \min_{\lambda \in \Lambda_i} \|x - \lambda\|^2$
3: Let $c \leftarrow x - z$

Algorithm 2 Feng’s decoding [15]
Input: $y \in \mathbb{R}^n$, which is a noisy version of $c$. Output: $u$.
Preprocessing: as in Algorithm 1.
1: Let $x \leftarrow \arg \min_{\lambda \in \Lambda} \|y + a - \lambda\|^2$
2: Let $u \leftarrow xG' - 1$
3: Let $u_i \leftarrow u_i \mod J_i, \forall i = 1, \ldots, n$

Every point $c \in \Gamma$ can be uniquely enumerated by vectors $u$ and $v$ such that
$$c = uG' + vG_s' - a,$$
where $u_i \in \{0, \ldots, J_i - 1\}$ for $i = 1, \ldots, n$ and $v \in \mathbb{Z}^n$. There are $M = \det J = \prod_i J_i$, possible values of $u$, and each of them occurs exactly once among all points $c \in \Gamma$. Hence, $u$ is used to label $c$, regardless of $v$.

In encoding, $u$ and $a$ in (8) are known, and the unique value of $z = -vG_s'$ that fulfills $c \in \Gamma$ is found. In decoding, given $c$ and $a$,
$$u = (c + a)G' - 1 - vJ,$$
which can be solved by setting $u_i$ to the $i$th element of $(c + a)G' - 1$ modulo $J_i$. If the decoder input vector is noisy, then it is first rounded to the nearest point in the translated lattice $\Lambda - a$. The encoding and decoding are summarized in Algorithm 1 and 2.

The arg min operations are carried out by an algorithm to find the closest point in a given lattice, which is a well-studied problem. Specific algorithms are available for many common lattices [22, Ch. 20], [24], [25], and other lattices can be handled by general algorithms [26].

Kurkoski’s algorithms: Given $\Lambda$ and its sublattice $\Lambda_s$, Kurkoski’s algorithms are applicable only when $\Lambda$ and $\Lambda_s$ both have triangular generator matrices $G$ and $G_s$, respectively. Conventionally, lower-triangular $G$ and $G_s$ are used. Then the diagonal elements of the lower-triangular integer matrix $L = G_sG_s^{-1}$ can be used to enumerate the $M = \prod_i L_{ii}$ constellation points as
$$c = uG + vG_s - a,$$
where $u_i \in \{0, \ldots, L_{ii} - 1\}$ for $i = 1, \ldots, n$ and $v \in \mathbb{Z}^n$.

The encoding is similar to Algorithm 1, but there is no need to calculate $G'$. In decoding, given $c$ and $a$, (10) is solved for $u$. This can be done sequentially, beginning from $u_n$, thanks to the triangular structure of $L$. Specifically, the algorithms operate as in Algorithm 3 and 4. If $L = rI$ for an identity matrix $I$ and a positive integer $r$, then the algorithms specialize into the classical encoding and decoding algorithms [12].

Specifically for the application of Kurkoski’s algorithms to VCs with $\Lambda = \mathbb{Z}^n$, different triangular generator matrices $G$ result in different encoding. It is important that $G$ is set to $I_n$ for pseudo-Gray labeling, where $I_n$ is an $n$-dimensional identity matrix, which is discussed in section V.

Ferdinand et al. proposed equally simple but less general encoding and decoding algorithms than Feng’s and Kurkoski’s specifically for VCs with a cubic coding lattice, which is applicable for most commonly used shaping lattices, e.g., the 4-dimensional checkerboard lattice $D_4$, 8-dimensional lattice $E_8$, 16-dimensional Barnes–Wall lattice $\Lambda_{16}$, and the 24-dimensional Leech lattice $\Lambda_{24}$ [22, Ch. 4]. Ferdinand’s algorithms can have the same mapping rule as Feng’s, when the generator matrices of shaping lattices are written in nice lower-triangular matrices as in [22, Ch. 4]. For the explicit algorithms, see [21].

IV. VORONOI CONSTELLATIONS WITH CUBIC CODING LATTICE

In this section, we study a specific kind of VC, whose coding lattice $\Lambda = \mathbb{Z}^n$. One reason why this kind of VC is interesting is that the search for the closest lattice point in decoding (the arg min operation in Algorithm 2 and 4), which usually dominates the complexity for the encoding and decoding, especially for high-dimensional lattices, is just to round the received noisy vector to an integer vector. With this trivial search algorithm, the decoding is a set of very low-complexity linear operations.

Apart from its simplicity in decoding, this VC maintains the asymptotic shaping gain. To see this, two figures of merit that are useful for comparing different modulation formats are considered as in [8], [27]: the 

spectral efficiency $SE = 2 \log_2(M)/n$ bits/symbol/dimension-pair and the power efficiency $PE = d_{\min}^2 \log_2(M)/(4E_s)$, where $d_{\min}$ is the minimum Euclidean distance of the constellation, and
where \( E_s = (1/M) \sum_{c \in \Gamma} ||c||^2 \) is the average symbol energy. Then the power efficiency of the one-dimensional pulse amplitude modulation (PAM) is \( PEP_{\text{PAM}} = 3SE/(2(2^{2SE} - 1)) \). The ratio \( PE/PEP_{\text{PAM}} \) in dB is the gain that a constellation can obtain over cubic constellations. At high spectral efficiencies, the \( PE/PEP_{\text{PAM}} \) ratio of VCs of the lattice partition \( \mathbb{Z}^n/\Lambda_p \) should converge to the asymptotic shaping gain \( \gamma_p(\Lambda_p) = 1/(12G(\Omega(\Lambda_p))) \) defined in [8], where \( G(\Omega(\Lambda_p)) \) is the normalized second moment of the Voronoi region \( \Omega(\Lambda_p) \) [28, Eq. (9)]. Forney listed \( G(\Omega(\Lambda_p)) \) and \( \gamma_p(\Lambda_p) \) of Voronoi regions of some classical multidimensional shaping lattices in [14, Table I]. In Fig. 2, we present \( PE/PEP_{\text{PAM}} \) as a function of \( SE \) for VCs with a cubic coding lattice and \( D_4 \), \( E_8 \), \( A_{16} \), and \( A_{24} \) as shaping lattices.

In addition, the separate shaping lattice and coding lattice allow for improved granularity in spectral efficiencies than the scaled VC, which gives us more flexibility in choosing different data rates as needed.

V. LABELING OF CONSTELLATION POINTS

The Gray penalty \( G_p \) of a constellation was defined as the average number of different bits per pair of adjacent symbols [30], [31]. It has been extensively used in the literature [32]–[34], since it predicts the asymptotic ratio of the BER \( P_b \) and SER \( P_s \) for the AWGN channel:

\[
P_b = \frac{G_p}{\log_2(M)} P_s. \tag{11}
\]

Clearly \( G_p \geq 1 \) and we want \( G_p \) to be close to 1. Gray penalty is a good measure for selecting labeling schemes for VCs. However, its calculation requires enumeration of all constellation points, which is infeasible for very large VCs.

We propose a new method to accurately estimate \( G_p \) for constellations too large to enumerate. With the cubic coding lattice, each symbol \( c \in \Gamma \) (except those on the boundary of the constellation) has \( 2n \) neighbors, forming a set \( \text{nbr}(c) \triangleq \{ h : ||h - c||^2 = 1, h \in \mathbb{Z}^n \} \), which can be easily enumerated by adding all permutations and sign changes of the \( n \)-tuple \((\pm 1, 0, \ldots, 0)\) to \( c \). Thus \( G_p \) can be estimated by Monte Carlo simulations for very large \( M \) as follows. We define \( \text{enc}(\cdot) \) as the encoding function performing Algorithm 1 or 3, \( \text{dec}(\cdot) \) as the decoding function performing Algorithm 2 or 4, and \( \text{map}(\cdot) \) as the mapping function converting each element \( u_i \) of an integer input vector \( u = (u_1, u_2, \ldots, u_n) \) to a binary vector of length \( \log_2(J_{ii}) \) or \( (\log_2(L_{ii})) \) for all \( i = 1, 2, \ldots, n \) using a one-dimensional natural binary code (NBC) or binary reflected Gray code (BRGC), and concatenating them together to a binary vector of length \( \log_2(M) \) in total. The explicit estimating process of \( G_p \) is described in Algorithm 5, where \( d_{\text{Ham}} \) denotes the Hamming distance between two binary vectors.

In Table I, we list the estimated \( G_p \) values for very large multidimensional VCs, when Feng’s and Kurkoski’s encoding and decoding algorithms and both NBC and BRGC are used. For the constellations with \( M \leq 2^{17} \), we validated that using \( N_s = 10^4 \) symbols, Algorithm 5 estimates the Gray penalties accurately with an error less than 0.5% compared with the exact \( G_p \) values calculated by enumerating all constellation points. The reason why Kurkoski’s algorithms always yield a smaller \( G_p \) is that the generator matrix \( G \) of the coding lattice \( \mathbb{Z}^n \) is \( I_n \), whereas when Feng’s algorithms are used, in order to fulfill (7) when \( D_4, E_8, A_{16}, \) and \( A_{24} \) are used as shaping lattices, \( G' \) has to be a non-identity matrix, which will change the natural or Gray ordering of \( u \) after multiplication. Specifically in this paper, we write \( G_s = G_s' \) as lower-triangular matrices [22, Ch. 4], then \( G' \) must also be lower-triangular. For the example VC in Fig. 1, in Feng’s encoding,

\[
G' = \begin{bmatrix}
1 & 0 \\
0 & 1 \\
1 & 1 \\
\end{bmatrix}.
\tag{12}
\]

**Algorithm 5 Gray penalty estimation**

**Input:** Generate \( N_s \) random integer vectors \( u_r \), \( r = 1, 2, \ldots, N_s \), of which where each element \( u_{r,i} \) is uniformly distributed in \( \{0,1,\ldots,J_{ii}-1\} \) for all \( i = 1, 2, \ldots, n \) (replace \( J_{ii} \) with \( L_{ii} \) if Algorithm 4 is used). **Output:** Estimated Gray penalty \( G_p \).

1. \( \text{counter}_1 = 0 \)
2. \( \text{counter}_2 = 0 \)
3. for \( r = 1, 2, \ldots, N_s \) do
4. \( c \leftarrow \text{enc}(u_r) \)
5. \( H \leftarrow \text{nbr}(c) \)
6. for \( h \in H \) do
7. \( x \leftarrow \arg\min_{\lambda \in \Lambda_s} ||h - \lambda||^2 \)
8. if \( x \neq 0 \) then remove \( h \) from \( H \)
9. end for
10. \( \text{counter}_1 \leftarrow \text{counter}_1 + |H| \)
11. \( m \leftarrow \text{map}(u_r) \)
12. \( K \leftarrow \{ \text{dec}(c) : c \in H \} \)
13. \( B \leftarrow \{ \text{map}(u) : u \in K \} \)
14. \( \text{counter}_2 \leftarrow \text{counter}_2 + \sum_{b \in B} d_{\text{Ham}}(m, b) \)
15. end for
16. \( G_p = \frac{\text{counter}_1}{\text{counter}_2} \)
TABLE I: Estimated Gray penalties of VCs of lattice partitions $Z^n/\Lambda$, at high spectral efficiencies.

<table>
<thead>
<tr>
<th>$Z^n/\Lambda_n$</th>
<th>$Z^3/D_4$</th>
<th>$Z^8/E_8$</th>
<th>$Z^{16}/\Lambda_{16}$</th>
<th>$Z^{24}/\Lambda_{24}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE [bit/symbol/dimension-pair]</td>
<td>12.5</td>
<td>12</td>
<td>11.5</td>
<td>13</td>
</tr>
</tbody>
</table>

Feng’s NBC          3.46   5.36    8.88    13.13
Kurkoski’s NBC      1.98   2.01    2.04    2.07
Feng’s BRGC        1.75   2.81    4.86    7.40
Kurkoski’s BRGC    1.02   1.08    1.16    1.17

Similarly, in Ferdinand’s encoding, which is equivalent to Feng’s encoding for the considered VCs in Table I, u also needs to multiply with a non-identity matrix, see the example in [21, Eq. (14)]. Ferdinand’s decoding is equivalent to Kurkoski’s decoding, and yields the same $G_p$ as Feng’s algorithms. As the constellation dimension increases, their $G_p$ grows rapidly with the number of dimensions. However, with Kurkoski’s algorithms, $G_p$ stays steady at different dimensions both for NBC and BRGC.

According to Table I, Kurkoski’s algorithms are expected to reduce the BER by a factor up to 6.3 compared with Feng’s and Ferdinand’s algorithms. Again, we implemented Ferdinand’s algorithms, and find the same BER performance as Feng’s. In Fig. 3, we illustrate the required $E_b/N_0$ gain (in dB) of Kurkoski’s algorithms over Feng’s as a function of $SE$, at a BER of $10^{-4}$, where $E_b = E_s/\log(2^M)$ is the energy per bit and $N_0$ is the noise variance of the Gaussian noise per two dimensions. As $n$ and $SE$ increase, higher gains are obtained both using NBC and BRGC. The curves with BRGC have a larger slope than NBC, and reach up to 1.1 dB for $Z^{24}/\Lambda_{24}$ at 13 bits per symbols per dimension-pair. Higher gains are expected if we further increase $n$ and $SE$.

We also investigate the BER performance of the designed VCs based on Kurkoski’s algorithms, with BRGC applied to integer vectors in coded systems for 4-dimensional case. LDPC codes from the digital video broadcasting (DVB-S2) standard [35] with a codeword length of 64800, 50 decoding iterations, and the max-log approximation of the log-likelihood ratio [36, Eq. (6)] is applied to 1) the designed VCs, (2) the same VCs based on Feng’s encoding and decoding, and (3) the scaled VCs. Different code rates $R_c$ are used to compare these VCs under almost the same information rate, defined as $R = SE \cdot R_c$ bits/symbol/dimension-pair. The labeling of the scaled VCs follows the “quasi-Gray labeling” in [13].

Fig. 4 shows 0.8–1 dB $E_b/N_0$ gains of Kurkoski’s algorithms over Feng’s at a BER of $10^{-4}$ after LDPC decoding. The designed VCs also outperform the scaled VCs, which means that the loss of coding gain due to cubic coding lattice is more than compensated by the LDPC code, implying no need for including more complex coding lattices. The better BER performance might come from the pseudo-Gray labeling for the cubic coding lattice, as our VCs have lower Gray penalties than the scaled VCs. Also, the higher spectral efficiencies allow us to use a higher overhead error-correction code.

VI. CONCLUSION

We designed low-complexity VCs with a cubic coding lattice based on Kurkoski’s encoding and decoding algorithms. With the pseudo-Gray labeling, the designed VCs can reduce the BER up to 6.3 times at the same SNR or reduce the required SNR by up to 1.1 dB at the same BER, compared with the two benchmark algorithms in the literature in uncoded systems. The Gray penalty estimation algorithm for large VCs with cubic coding lattice can be used to choose good labeling schemes. In combination with a LDPC code, the designed VCs can have better BER performance with lower decoding complexity compared with the scaled VCs, due to the efficient pseudo-Gray labeling for the cubic coding lattice. In the future, the mutual information of such VCs is a worthy topic to study.

ACKNOWLEDGMENT

This work was supported by the Swedish Research Council (VR) under grant no. 2017–03702.