TECHNICAL RESEARCH REPORT

SYSTEMS RESEARCH CENTER

Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

On SVQ Shaping of Multidimensional Constellations – High-Rate Large-Dimensional Constellations

by R. Laroia, N. Farvardin and S. Tretter

TR 92-5 r2

On SVQ Shaping of Multidimensional Constellations — High-Rate Large-Dimensional Constellations†

by

Rajiv Laroia, Nariman Farvardin and Steven Tretter
Electrical Engineering Department
University of Maryland
College Park, Maryland 20742

Abstract

An optimal shaping scheme for multidimensional constellations, motivated by some ideas from a fixed-rate structured vector quantizer (SVQ), was recently proposed by Laroia. It was shown that optimal shaping could be performed subject to a constraint on the CER_2 or PAR_2 by expressing the (optimally shaped) constellation as the codebook of an SVQ and using the SVQ encoding/decoding algorithms to index the constellation points. Further, compatibility with trellis coded modulation was demonstrated. The complexity of the proposed scheme was reasonable but dependent on the data transmission rate. In this paper, we use recent results due to Calderbank and Ozarow to show that complexity of this scheme can be reduced and made independent of the data rate with essentially no effect on the shaping gain. Also, we modify the SVQ encoding/decoding algorithms to reduce the implementation complexity even further. It is shown that SVQ shaping can achieve a shaping gain of about 1.2 dB with a PAR_2 of 3.75 at a very reasonable complexity (about 15 multiply-adds/baud and a memory requirement of 1.5 kbytes). Further, a shaping gain of 1 dB results in a PAR_2 of less than 3. This is considerably less than a PAR_2 of 3.75 for Forney's trellis shaping scheme that gives about 1 dB shaping gain.

Index Terms: Multidimensional constellations; SVQ shaping; Optimal shaping; Shaping gain.

† The first two authors are affiliated with the Systems Research Center at the University of Maryland; their work was supported in part by National Science Foundation grants NSFD MIP-9109109, MIP-86-57311 and CDR-85-00108.

I. Introduction

It was shown in [1] that an N-sphere shaped N-dimensional cubic lattice ($\mathbf{Z}^N$) based constellation can be described as the codebook of a structured vector quantizer (SVQ) [2]. The codevector encoding/decoding algorithms of the SVQ (given in [1]) can be used to index the points of such a constellation. Since an N-sphere has the smallest average energy for a given volume, N-sphere SVQ shaping leads to optimal shaping gains in N-dimensions. Such shaping is however not very useful because it results in a high constellation expansion ratio (CER_2) and a high peak-to-average power ratio (PAR_2) of the constituent 2D constellation. For a given constraint on the CER_2 or PAR_2, the optimal shaping region is the intersection of the interiors of an N-sphere and $\mathcal{C}^{N/2}$ (even N) [1], where $\mathcal{C}$ is a circle bounding the constituent 2D constellation and $\mathcal{C}^{N/2}$ is the $N/2$-fold cartesian product of $\mathcal{C}$ with itself. A $\mathbf{Z}^N$ based constellation bounded by this region can be represented as the codebook of an SVQ making it possible to achieve the optimal shaping gain for a given CER_2 or PAR_2. This is very useful because as shown in [3], a significant reduction in both CER_2 and PAR_2 is possible with only a small loss in shaping gain. A cubic lattice by itself offers no coding gain but a trellis code realizing a significant coding gain can be constructed from a redundant cubic lattice [4]. Constellations based on trellis codes therefore perform significantly better than those based on $\mathbf{Z}^N$. Compatibility of SVQ shaping with trellis coding was also demonstrated in [1]. The SVQ-shaped trellis-coded constellations thus offer optimal shaping gains and significant coding gains.

The memory requirement of the SVQ shaping scheme presented in [1] is cubic in the constellation dimension N and exponential in the rate r (in bits/2D). The computational requirement per 2D is linear in N and exponential in r. Although this shaping scheme gives higher shaping gains than trellis shaping [5], its complexity for a shaping gain of about 1 dB is considerably higher than that of the 4-state trellis shaping scheme.

In this paper, we deal with ways to reduce the implementation complexity of SVQ shaping without significantly affecting the shaping gain. In the next section we present SVQ encoding and decoding algorithms that are more efficient than those given in [1]. These algorithms reduce the storage complexity of SVQ shaping from cubic to quadratic in the constellation dimension N making it more practical to implement large-dimensional constellations and achieve high shaping gains. The complexity however is still exponential in the rate r (in bits/2D) of the constellation. This problem is solved in Section III by using the results of Calderbank and Ozarow [6] (also see [7]) to show that for high rates ($r > 6$ bits/2D) the SVQ shaping complexity can be made independent of the data rate with a negligible loss in shaping gain. Finally, in Section IV an example of a rate $r = 7$ bits/2D 64-dimensional SVQ-shaped trellis-coded constellation with a 384 point circular constituent 2D constellation is given. This corresponds to a shaping $\text{CER}_2 = 1.5$ and a coding $\text{CER}_2 = 2$ and achieves a shaping gain of 1.20 dB with a PAR_2 of about 3.75. The computational and storage complexities associated with the shaping of this constellation are also reported.

II. SVQ and SVQ Shaping

We start with brief descriptions of the SVQ and SVQ shaping, following which we present the modified SVQ encoding and decoding algorithms.

II.1 The Structured Vector Quantizer

The SVQ is a special kind of vector quantizer (VQ) in which the codebook structure is derived from a variable-length quantizer $\mathcal{S}$. Let $\mathcal{Q} \equiv \{q_1, q_2, ..., q_n\}$ be the set of n elements in the alphabet of the quantizer $\mathcal{S}$ ($\mathcal{Q}$ is also referred to as the SVQ alphabet) and $\mathcal{L} \equiv \{\ell_1, \ell_2, ..., \ell_n\}$ be the corresponding set of positive integer lengths, where $\ell_i,\ i \in J_n \equiv \{1, 2, ..., n\}$ is the length associated with the element q_i. The codebook $\mathcal{Z}$ of an m-dimensional SVQ $\mathcal{V}$ derived from $\mathcal{S} \equiv (\mathcal{Q}, \mathcal{L})$ is a subset of $\mathcal{Q}^m$ consisting of only those points (m-tuples) that have a total length no greater than an integer threshold L. The total length is defined as the sum of the lengths of the individual components and the threshold L is chosen such that the codebook $\mathcal{Z}$ contains (at most) 2^{mr} of the n^m total points in $\mathcal{Q}^m$, where r is the desired rate of the SVQ $\mathcal{V}$ in bits/sample (r here should not be confused with its earlier use as the rate of a constellation). This is formally described by the following definition.

Definition: An m-dimensional SVQ $\mathcal{V}$ derived from a variable-length quantizer $\mathcal{S} \equiv (\mathcal{Q}, \mathcal{L})$ is a VQ with a codebook $\mathcal{Z}$ given as,

$$\mathcal{Z} = \{\mathbf{z} \equiv (z_1, z_2, \ldots, z_m) \in \mathcal{Q}^m \ :\ \ell_{f(z_1)} + \ell_{f(z_2)} + \cdots + \ell_{f(z_m)} \leq L\}, \tag{1}$$

where the index function $f : \mathcal{Q} \rightarrow J_n$ is defined as,

$$f(q_i) = i,\ i \in J_n\ . \tag{2}$$

For a rate r bits/sample SVQ, the threshold L is selected as the largest integer such that the cardinality of $\mathcal{Z}$ is no greater than 2^{mr}.

With this structure of the SVQ codebook, there exist fast codevector encoding and decoding algorithms that can index (label) the codevectors of the SVQ. The codevector encoding and decoding algorithms presented in [2] were modified in [1] to make them suitable for data transmission purposes. Later in this section we present yet another version of these algorithms that has a smaller storage complexity.

Since in the present work we use the SVQ in the data transmission context, we define the threshold L as the smallest integer such that the cardinality of $\mathcal{Z}$ is no less than 2^{mr}. This ensures that the SVQ-shaped constellations described next have the required number of points to transmit at the desired rate.

II.2 Constellation Shaping Using the SVQ

As shown in [1], the optimally shaped constellations can be represented as the codebook of an SVQ and the SVQ encoding/decoding algorithms can be used to index the constellation points. In this paper we confine our attention to quadrature amplitude modulation (QAM) based systems. An N-dimensional constellation can hence be expressed as the codebook of an $(N/2)$-dimensional SVQ ($m = N/2$) with an alphabet consisting of points in the constituent 2D constellation.

Under the shaping $\text{CER}_2 \leq \beta$ constraint, the optimally shaped N-dimensional constellation has a constituent 2D constellation $\mathbf{A}_0$ ($\subset \mathbf{Z}^2$) contained inside a circle $\mathcal{C}$ and consisting of (at most) $\delta = \beta 2^r$ points [1],[3], where r is the constellation rate in bits/2D. The optimal shaping region in this case is the intersection of the interior of an N-sphere with the interior of $\mathcal{C}^{N/2}$ [1]. The optimally shaped constellation can be expressed as the codebook of an $(N/2)$-dimensional SVQ with: (i) an alphabet consisting of the δ points $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_\delta$ of $\mathbf{A}_0$, i.e., $\mathcal{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_\delta\}$; (ii) a set of lengths $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_\delta\}$, with the length ℓ_i, $i \in J_\delta$, of the point $\mathbf{q}_i$ given as, $\ell_i = \|\mathbf{q}_i\|^2$, where $\|\mathbf{q}_i\|^2$ is the squared Euclidean distance of $\mathbf{q}_i$ from the origin; and (iii) a threshold L which is determined such that the constellation has at least 2^{mr} points.

Compatibility of SVQ shaping with trellis coding was also demonstrated in [1]. The SVQ-shaped trellis-coded constellations can realize optimal shaping and significant coding gains.

II.3 Efficient SVQ Encoding and Decoding Algorithms

The encoding and decoding algorithms described here are for the SVQ defined in II.1 and are generalizations of the corresponding algorithms in [1] (for 2-dimensions). These algorithms assume that the codebook dimension m is a power of 2, i.e., $m = 2^K$. The understanding of these algorithms will probably be aided by the knowledge of the corresponding algorithms in [1].

For $m_i = m/2^{(i-1)}$, $i \in J_{K+1} \equiv \{1, 2, \ldots, K+1\}$, let $M^j_{m_i}$ represent the number of m_i-vectors $\mathbf{v} \equiv (v_1, v_2, \ldots, v_{m_i}) \in \mathcal{Q}^{m_i}$ that have a total length $\ell_{f(v_1)} + \ell_{f(v_2)} + \cdots + \ell_{f(v_{m_i})} = j$. Clearly, $M^j_{m_{(K+1)}} = M^j_1$ is the number of elements in the SVQ alphabet $\mathcal{Q}$ that have a length equal to j. The $M^j_{m_i}$ can be determined by the following recursive equation:

$$M^j_{m_i} = \sum_{k=1}^{j-1} M^{j-k}_{m_{(i+1)}} M^k_{m_{(i+1)}}, \; \forall i \in J_K, \tag{3}$$

where $m_i = 2m_{(i+1)}$ and each step doubles the dimension. The sequence $M^j_{m_i}$, $j = 1, 2, 3, \ldots$, hence results from the convolution of the sequence $M^j_{m_{(i+1)}}$, $j = 1, 2, 3, \ldots$, with itself. If $C^j_{m_i}$ is the number of m_i-vectors $\mathbf{v} \in \mathcal{Q}^{m_i}$ with a total length no greater than j, then

$$C^j_{m_i} = \sum_{k=1}^{j} M^k_{m_i}. \tag{4}$$

The SVQ threshold L is the minimum value of j for which $C^j_{m_1} = C^j_m \geq 2^{mr}$, where r is the desired rate of the SVQ in bits per SVQ dimension.

The encoding and decoding algorithms described below assume that the $M^j_{m_i}$ $\forall i \in \{2, 3, \ldots, K+1\}$, $\forall j \in J_L$ and C^j_m, $\forall j \in J_L$, are computed once and stored in the memory. This takes up considerably less storage (especially for a large m) than storing the M^j_k for all $k \in J_{m-1}$, as in [1]. It is further assumed that the SVQ alphabet $q_1, q_2, \ldots, q_n$, is indexed such that the corresponding lengths $\ell_1, \ell_2, \ldots, \ell_n$, form a non-decreasing sequence, i.e., the smaller lengths are assigned a smaller index.

<u>The SVQ Encoding Algorithm</u> *(constellation point to binary index)*

The encoding function assigns a unique mr-bit binary number to every codevector of the SVQ. There are several mappings that can accomplish this, two such mapping are described in [1] and [2], respectively. The algorithm given here encodes an $m =$

2^K-dimensional codevector by recursively splitting it into two equal parts. Let ${}^i\mathbf{v} \equiv (v_1, v_2, \ldots, v_{m_i})$ denote an $m_i = m/2^{(i-1)}$-dimensional vector in $\mathcal{Q}^{m_i}$. Also, let ${}^i\mathbf{v}_1 \equiv (v_1, v_2, \ldots, v_{m_{(i+1)}})$ and ${}^i\mathbf{v}_2 \equiv (v_{(m_{(i+1)}+1)}, v_{(m_{(i+1)}+2)}, \ldots, v_{m_i})$ be the first and the second halves of ${}^i\mathbf{v}$, respectively. If $\mathbf{z} \equiv (z_1, z_2, \ldots, z_m) \in \mathcal{Q}^m$ is the codevector to be encoded, then let ${}^1\mathbf{v} = \mathbf{z}$.

Represent by $E^i : \mathcal{Q}^{m_i} \rightarrow \{0, 1, 2, \ldots\}$, the encoding function that encodes a vector ${}^i\mathbf{v} \in \mathcal{Q}^{m_i}$ into a non-negative integer. Define $E^{K+1} : \mathcal{Q} \rightarrow \{0, 1, 2, \ldots\}$, as $E^{K+1}(v) = f(v) - 1,\ \forall v \in \mathcal{Q}$, where $f(.)$ is the index function in the SVQ definition.

Assume that all ${}^i\mathbf{v}_1$ and ${}^i\mathbf{v}_2$ in $\mathcal{Q}^{m_{(i+1)}}$ have been indexed using E^{i+1}, i.e., $E^{i+1}({}^i\mathbf{v}_1)$ and $E^{i+1}({}^i\mathbf{v}_2)$ are the indexes of ${}^i\mathbf{v}_1$ and ${}^i\mathbf{v}_2$, respectively. Order all vectors in $\mathcal{Q}^{m_i}$ according to the following three rules: (i) a vector ${}^i\mathbf{u} \in \mathcal{Q}^{m_i}$ is "smaller than" ${}^i\mathbf{w} \in \mathcal{Q}^{m_i}$ (i.e., ${}^i\mathbf{u} < {}^i\mathbf{w}$) if $T^i({}^i\mathbf{u}) < T^i({}^i\mathbf{w})$, where the function $T^i(.)$ gives the total length of an m_i-vector $\in \mathcal{Q}^{m_i}$; (ii) if $T^i({}^i\mathbf{u}) = T^i({}^i\mathbf{w})$, then ${}^i\mathbf{u} < {}^i\mathbf{w}$ if $E^{i+1}({}^i\mathbf{u}_1) < E^{i+1}({}^i\mathbf{w}_1)$; and (iii) if $T^i({}^i\mathbf{u}) = T^i({}^i\mathbf{w})$ and $E^{i+1}({}^i\mathbf{u}_1) = E^{i+1}({}^i\mathbf{w}_1)$, then ${}^i\mathbf{u} < {}^i\mathbf{w}$ if $E^{i+1}({}^i\mathbf{u}_2) < E^{i+1}({}^i\mathbf{w}_2)$.

The encoding function $E^i({}^i\mathbf{v})$ is now given as the number of vectors in $\mathcal{Q}^{m_i}$ that are smaller than ${}^i\mathbf{v}$. This can be expressed as,

$$E^i({}^i\mathbf{v}) = \mathcal{E}^i({}^i\mathbf{v}) + C_{m_i}^{T^i({}^i\mathbf{v})-1}, \tag{5}$$

where $\mathcal{E}^i({}^i\mathbf{v})$ is the number of length $T^i({}^i\mathbf{v})$ vectors in $\mathcal{Q}^{m_i}$ that are smaller than ${}^i\mathbf{v}$ and is given as,

$$\mathcal{E}^i({}^i\mathbf{v}) = \sum_{k=1}^{T^{i+1}({}^i\mathbf{v}_1)-1} M_{m_{(i+1)}}^k M_{m_{(i+1)}}^{T^i({}^i\mathbf{v})-k} + \mathcal{E}^{i+1}({}^i\mathbf{v}_1) M_{m_{(i+1)}}^{T^i({}^i\mathbf{v})-T^{(i+1)}({}^i\mathbf{v}_1)} + \mathcal{E}^{i+1}({}^i\mathbf{v}_2). \tag{6}$$

The encoding operation is hence performed by partitioning the input m-tuples into $m/2$ pairs and encoding the pairs using $E^K(.)$. The pairs are then grouped into 4-tuples and encoded using $E^{K-1}(.)$ and so on. The dependence of the storage complexity of this algorithm on the dimension is $m^2 \log m$. Its computational complexity is slightly higher than the corresponding algorithm in [1], but as shown in Section IV, this can be substantially reduced by repeated use of the ideas of [6] presented in the next section.

Note that to reduce the storage requirement, the $C_{m_i}^j,\ \forall j \in J_L$ are stored in the memory only for $i = 1$. For $i \in \{2, 3, \ldots, K+1\}$, the $C_{m_i}^j$ can be sparsely stored — as an example, for values of j that are multiples of 4. These can then be used together with the stored $M^j_{m_i}$ to obtain the $C^j_{m_i}$ for other values of j.

The SVQ Decoding Algorithm (binary index to constellation point)

The decoding function takes mr-bit binary numbers and converts them into SVQ codevectors in a one-to-one manner and is implemented as the inverse of the encoding function described above. The aim is to determine $\mathbf{z} = {}^1\mathbf{v} \in \mathcal{Q}^m$ given an mr-bit binary number $E^1({}^1\mathbf{v})$. This is accomplished by determining $E^2({}^1\mathbf{v}_1)$ and $E^2({}^1\mathbf{v}_2)$ from $E^1({}^1\mathbf{v})$. The problem now reduces to an equivalent $(m/2)$-dimensional problem which can be similarly handled.

To determine $E^2({}^1\mathbf{v}_1)$ and $E^2({}^1\mathbf{v}_2)$ from $E^1({}^1\mathbf{v})$, first determine $\mathcal{E}^1({}^1\mathbf{v})$ and the length $T^1({}^1\mathbf{v})$ of ${}^1\mathbf{v}$ using the stored values of C^j_m, $j \in J_L$. The values $T^2({}^1\mathbf{v}_1)$, $\mathcal{E}^2({}^1\mathbf{v}_1)$, $T^2({}^1\mathbf{v}_2)$ and $\mathcal{E}^2({}^1\mathbf{v}_2)$ are next determined form $\mathcal{E}^1({}^1\mathbf{v})$ by repeated subtraction and a division (see Equation (6)).

The complexity of this decoding algorithm is approximately the same as that of the encoding algorithm.

III. SVQ Shaping at High Rates

Calderbank and Ozarow in [6] have shown that it is in principle possible to achieve most of the maximum shaping gain of 1.53 dB by partitioning a circular 2D constellation (with a large number of points) into a small number, t, of equal area regions (circular shells) and using all the constellation points in a region with the same probability. Virtually all of the shaping gain can be achieved with just $t = 16$ regions and a CER_2 less than about 2. For $t = 8$ a shaping gain of over 1.4 dB can be realized. The shaping gain vs. PAR_2 plot for $t = 32$ given in Fig. 4(e) in [6] (the notation in [6] is different from that used here) is claimed to be indistinguishable from the optimal shaping gain vs. PAR_2 plot in [3].

These results suggest that nearly optimal SVQ shaping can be performed by partitioning a circular constituent 2D constellation into a maximum of $t = 16$ regions and requiring that all the points in a region have the same length. This ensures that all points in the same region of the constituent 2D constellation occur with the same frequency in the multidimensional constellation. Since SVQ shaping can minimize the average constellation energy subject to the above constraint, for a given t, it can asymptotically (in dimension) achieve the performance described in [6]. The shaping gain for a given t is independent of the number of points in the constituent 2D constellation (assuming a large number of points). Therefore, the complexity of this SVQ shaping scheme for nearly optimal shaping does not continue to increase with the constellation rate r. Numerical results show that for r greater than about 6 bits/2D, the complexity is independent of the rate.

The above reduction in complexity when combined with the efficient SVQ encoding/decoding algorithms described in Section II makes it possible to achieve a higher shaping gain for a given complexity (or a given CER_2, PAR_2 or shaping delay) than any other shaping scheme proposed so far. In the next section we consider an example that demonstrates this.

IV. Example of an SVQ-Shaped Constellation

Assume that it is desired to transmit binary data using a 64-dimensional SVQ-shaped uncoded (trellis-coded constellations are considered later) $\mathbf{Z}^{64}$ based constellation at the rate of 8 bits/2D. The constituent 2D constellation in this case must consist of at least 256 points. A circular 256-point 2D constellation however results in only 0.2 dB shaping gain (that of a circle over a square). To achieve higher shaping gains, the constituent 2D constellation must be expanded to have more than 256 points. In this example, we assume that a shaping CER_2 of 1.5 (corresponding to a 384-point 2D constellation) is acceptable. The 2D constellation $\mathbf{A}_0$ hence consists of 384 points on the translated lattice $\mathbf{Z}^2 + (1/2, 1/2)$ that are enclosed inside a circle of appropriate radius. The circular constellation $\mathbf{A}_0$ is partitioned into $t = 12$ subsets (regions) $R_1, R_2, \ldots, R_{12}$, each containing 32 points. The subset R_1 consists of the 32 lowest energy (smallest squared-distance from the origin) points in $\mathbf{A}_0$, R_2 consists of the 32 next higher energy points in $\mathbf{A}_0$ and so on. There are many different ways to pick the subsets $R_1, R_2, \ldots, R_{12}$, and any of these that preserves the $\pi/2$ rotational symmetry of $\mathbf{A}_0$ can be chosen.

As suggested by the results of [6], close to optimal shaping gain can be achieved by using all 32 points in any given subset R_i, $i \in J_{12}$ with the same probability. In the context of SVQ shaping, this can be accomplished by taking the SVQ alphabet as the 384 points in $\mathbf{A}_0$ and assigning the same length to all points in the same subset. The threshold L can be determined such that the 32-dimensional SVQ codebook (corresponding to a 64-dimensional constellation) has $2^{32\times 8} = 2^{256}$ codevectors (constellation points). Alternatively, in a more efficient formulation, the SVQ alphabet is taken as the 12 ($n = t = 12$) subsets, i.e., $\mathcal{Q} \equiv \{q_1, q_2, \ldots, q_{12}\} = \{R_1, R_2, \ldots, R_{12}\}$ and to every subset R_i, $i \in J_{12}$, is assigned a length $\ell_i = i$. The threshold L in this case is determined such that the codebook $\mathcal{Z}$ of the 32-dimensional SVQ ($m = 32$) consists of $2^{32\times 3} = 2^{96}$ codevectors in $\mathcal{Q}^{32}$. Each codevector here represents $2^{32\times 5} = 2^{160}$ constellation points.

The transmitter in a QAM based system accepts binary data in blocks of $32 \times 8 = 256$ bits and transmits each block using 32 2D-points (equivalent to one point on the 64-dimensional constellation). Out of the 256 bits in each block, the SVQ decoder decodes 96 bits into a codevector in $\mathcal{Q}^{32}$. The additional 5 bits/2D (a total of 160 bits) are used to determine which point ($\in \mathbf{A}_0$) of each 32-point subset is transmitted. In the receiver, 5 bits/2D are recovered by determining which subset point was received and 96 bits are recovered by using the SVQ encoder to encode the received codevector.

The 64-dimensional constellation in this example is a cubic lattice based constellation and realizes no coding gain. As shown in [1], SVQ shaping is compatible with trellis coding and it is possible to construct a 64-dimensional SVQ-shaped trellis-coded constellation from the 384-point circular 2D constellation $\mathbf{A}_0$ considered above. Assuming that the trellis code used has a redundancy of 1 bit/2D, a rate of 7 bits/2D coded constellation then has the same shaping gain as the 8 bits/2D uncoded constellation.

Numerical evaluation shows that the above SVQ-shaped constellations achieve a shaping gain of 1.20 dB and have a $\text{PAR}_2 = 3.76$. The shaping operation requires about 60 multiply-adds/2D and 3 kbytes of memory. For $t = 12$, a shaping gain of up to 1.26 dB can be realized with a PAR_2 of 5.25 by a 64-dimensional constellation (optimal 64-dimensional shaping results in up to 1.31 dB shaping gain). This is about 0.25 dB more than the 1 dB gain of Forney's 4-state trellis shaping scheme ($\text{PAR}_2 \approx 3.75$) [5]. Using 64-dimensional SVQ shaping, a gain of 1 dB results in a PAR_2 of only 2.9. This is even smaller than the PAR_2 of the baseline constellation that gives no shaping gain. Such small values of the PAR_2 can be useful for transmission over channels that introduce harmonic distortion at high signal levels.

The complexity of the shaping scheme in the example above can be further reduced with little or no effect on the shaping gain. This is done by repeatedly applying the ideas of Section III to constituent constellations in 4, 8, 16 and 32 dimensions. For instance, the number M_8^j is the total number of length $T^3({}^3\mathbf{v}) = j$ points ($K = 5$ and ${}^3\mathbf{v}$ represents an 8-tuple in $\mathcal{Q}^8$) in the constituent 16-dimensional constellation. The constituent 16-dimensional constellation can be divided into subsets (regions) of say, 1024 points each.

The 1024 points with the smallest lengths belong to the first subset, the 1024 points with the next higher lengths belong to the next subset and so on. All points in the same subset are assigned the same length (this may be different from the original length of the points). This corresponds to modifying the numbers M_8^j, $j = 1, 2, 3, \ldots$, to a new set $\mathcal{M}_8^j$, where $\mathcal{M}_8^j$ is the new number of points in the constituent 16-dimensional constellation with a modified length $\mathcal{T}^3({}^3\mathbf{v}) = j$. Also, let $\mathcal{C}_8^j$ be the (new) number of points in the 16-dimensional constellation with (modified) length $\mathcal{T}^3({}^3\mathbf{v}) \leq j$.

The codeword $E^3({}^3\mathbf{v})$ of ${}^3\mathbf{v}$ can now be used to determine $\mathcal{T}^3({}^3\mathbf{v})$ and $e^3({}^3\mathbf{v})$ which is the number of modified length $\mathcal{T}^3({}^3\mathbf{v})$ points in the constituent 16-dimensional constellation that are smaller than ${}^3\mathbf{v}$.

For this approach, the encoding function of Section II is modified as follows:

$$E^i({}^i\mathbf{v}) = \mathcal{E}^i({}^i\mathbf{v}) + C_{m_i}^{T^i({}^i\mathbf{v})-1}, \tag{7}$$

where,

$$\mathcal{E}^i({}^i\mathbf{v}) = \sum_{k=1}^{\mathcal{T}^i({}^i\mathbf{v}_1)-1} \mathcal{M}_{m_{(i+1)}}^k \mathcal{M}_{m_{(i+1)}}^{T^i({}^i\mathbf{v})-k} + e^{i+1}({}^i\mathbf{v}_1)\mathcal{M}_{m_{(i+1)}}^{T^i({}^i\mathbf{v})-\mathcal{T}^{(i+1)}({}^i\mathbf{v}_1)} + e^{i+1}({}^i\mathbf{v}_2), \tag{8}$$

with

$$M_{m_i}^j = \sum_{k=1}^{j-1} \mathcal{M}_{m_{(i+1)}}^{j-k} \mathcal{M}_{m_{(i+1)}}^k, \tag{9}$$

$$C_{m_i}^j = \sum_{k=1}^{j} M_{m_i}^k, \tag{10}$$

and

$$T^i({}^i\mathbf{v}) = \mathcal{T}^{i+1}({}^i\mathbf{v}_1) + \mathcal{T}^{i+1}({}^i\mathbf{v}_2). \tag{11}$$

With the above modification, a variety of trade-offs between shaping gain, computational complexity and storage requirement are possible. Numerical evaluation shows that with 64-dimensional SVQ shaping, a shaping gain of about 1.2 dB can be realized at a PAR_2 of 3.75 with a worst case computational complexity of about 15 multiply-adds/2D and a storage requirement of around 1.5 kbytes.

V. Conclusions

In this paper we have shown that a considerable reduction in complexity of the SVQ shaping scheme proposed in [1] is possible with very little effect on the shaping gain. To reduce the storage complexity, modified SVQ encoding and decoding algorithms were presented. Next, the results of Calderbank and Ozarow on nonequiprobable signaling over Gaussian channels [6] were used to show that the complexity of SVQ shaping can be made independent of the rate of the constellation.

It was shown that shaping gains of up to 1.25 dB can be realized at a very reasonable complexity. This is considerably higher than any other similar complexity shaping scheme proposed so far. Further, for a given shaping gain SVQ-shaped constellations result in the smallest PAR_2, CER_2 and shaping delay of any shaping scheme. For example, a gain of 1 dB with 64-dimensional SVQ shaping results in a PAR_2 of only 2.9. This is significantly less than the PAR_2 of about 3.75 for a 4-state trellis shaping scheme [5] with a similar shaping gain and complexity. The small PAR_2 offers a considerable advantage for transmission over channels that introduce harmonic distortion at high signal levels.

For transmission over ISI channels, SVQ shaping can be used together with the precoding scheme presented in [8] to realize shaping gains up to 1.15 dB.

References

1. R. Laroia, "On Optimal Shaping of Multidimensional Constellations — An Alternative Approach to Lattice-Bounded (Voronoi) Constellations," submitted to *IEEE Trans. Inform. Theory*, Nov. 1991.
2. R. Laroia and N. Farvardin, "A Structured Fixed-Rate Vector Quantizer Derived from a Variable-Length Scalar Quantizer," submitted to *IEEE Trans. Inform. Theory*, Aug. 1991.
3. G. D. Forney, Jr. and L. F. Wei, "Multidimensional Constellations – Part I: Introduction, Figures of Merit, and Generalized Cross Constellations," *IEEE J. Select. Area Commun.*, Vol. 7, No. 6, pp. 877-892, Aug. 1989.
4. G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," *IEEE Trans. Inform. Theory*, Vol. IT-28, pp. 55-67, Jan. 1982.
5. G. D. Forney, Jr., "Trellis Shaping," *IEEE Trans. Inform. Theory*, to appear.
6. A. R. Calderbank and L. H. Ozarow, "Nonequiprobable Signaling on the Gaussian Channel," *IEEE Trans. Inform. Theory*, Vol. 36, pp. 726-740, July 1990.

7. A. R. Calderbank, "Binary Covering Codes and High Speed Data Transmission," *International Symposium on Coding Theory and Applications*, Italy, Nov. 1990.

8. R. Laroia, S. Tretter and N. Farvardin, "A Simple and Effective Precoding Scheme for Noise Whitening on Intersymbol Interference Channels," submitted to *IEEE Trans. Commun.*, Jan. 1992.