

Efficient, Nearly Optimum Addressing Schemes Based on Partitioning the Constellation into the Union of Blocks

A. K. Khandani¹ and P. Kabal^{1,2}

¹INRS-Telecommunications, 16 Place du Commerce, Verdun, Canada, H3E-1H6

²McGill University, 3480 University, Montreal, Canada, H3A-2A7

¹**Abstract:** In this paper, we introduce two efficient addressing schemes for the nearly optimum shaping of multi-dimensional signal spaces. Using K concentric circles, the 2-D (two-dimensional) subspaces are partitioned into K shells of equal volume. The 2-D shells are indexed in the radial direction from zero to $K-1$. The average energy of a 2-D shell is proportional to its index plus a fixed offset. In an $N=2n$ -D space, we obtain K^n shaping clusters of equal volume. Shaping is achieved by selecting $T \leq K^n$ of the N -D clusters with the least average energy (least sum of the 2-D indices). This results in a set of T integer n -tuples with components in the range $[0, K-1]$ and the sum of the components being at most a given number L_{\max} . The problem of addressing is to find a one-to-one mapping between the set of these n -tuples and the set of the integers $[0, T-1]$ such that the mapping and its inverse can be easily implemented. In the proposed schemes, the N -D clusters are grouped into blocks such that the addressing within the blocks, which is achieved using a common algorithm for all the blocks, has a low complexity. The addressing of the blocks is based on some recursive relationship which allows us to decompose the problem into smaller parts each of a low complexity. The overall scheme requires a moderate amount of memory and has a small computational complexity. The introduced methods are compared with the previously known schemes. The reduction in the complexity is substantial.

1 Introduction

In a data transmission system, the data is encoded such that in each signaling interval one of the M equiprobable symbols is produced. The overall transmission system can be modeled as a discrete-time system. In the discrete model, the channel provides us with a given number of dimensions, say N , per signaling interval. To achieve the transmission, we select M points over the channel space. Each of the source symbols is represented by one of these points. This is called a signal constellation.

In selecting the boundary of a constellation (shaping region), the objective is to minimize the average energy of the constellation for a given number of points from a given packing. In continuous approximation, the distribution of the constellation points is approximated by a continuous uniform density within the shaping region. The reduction in the average energy per two dimensions due to the use of the region \mathcal{C} as the boundary instead of using a hypercube is called the shaping gain of \mathcal{C} and is denoted as $\gamma_s(\mathcal{C})$.

The price to be paid for shaping ($\gamma_s > 1$) involves: (i) an increase in the factor CER_2 (Constellation-Expansion-Ratio) which is defined as the ratio of the number of points per two dimensions to

the minimum necessary number of points per two dimensions [1], (ii) an increase in the factor PAR_2 (Peak-to-Average-power-Ratio) which is defined as the ratio of the peak of energy per two dimensions to the average energy per two dimensions [1], and, (iii) an increase in the addressing complexity where addressing is the assignment of the source symbols to the constellation points².

In general, there exists a tradeoff between γ_s and CER_2 and also between γ_s and PAR_2 . An optimally shaped region is a region which optimizes both of these tradeoffs [2].

The major problem associated with shaping in a high dimensional space is the addressing complexity. For example, for 2-D subconstellations composed of 256 points in an $N=32$ -D space, a direct addressing scheme using a lookup table requires a block of memory with about 2^{128} memory locations where 128 arises from 8 bits/2-D times 16, 2-D subspaces per signaling interval.

Previous relevant works: Conway and Sloane in [3] introduced the idea of the Voronoi constellation based on using the Voronoi region of a lattice Λ , as the shaping region. The Voronoi constellations are further considered by Forney in [4]. In the work of Wei [5], shaping is a side effect of the method employed to transmit a nonintegral number of bits per two dimensions. Forney and Wei generalize this method in [1]. In [6], Calderbank and Ozarow introduced a shaping method which is directly achieved on the 2-D subspaces. The idea of the trellis shaping is introduced by Forney in [7]. Lang and Longstaff in [8] use an addressing scheme which is based on decomposing the space into lower-dimensional subspaces via generating function techniques. Prior to [8], a similar addressing scheme was used by Fischer in [9].

In [10], Kschischang and Pasupathy discuss a shaping method which is based on using the 2-D points with nonequal probability. In [11], Livingston discusses a shaping method in which the 2-D subspaces are partitioned into circular shells of increasing size. In this method, the 2-D shells are used with equal probability inducing a nonuniform distribution on the 2-D points. In a continuation to [6] and [11], Calderbank and Klimesh in [12] use a balanced binary code to select the sequence of the 2-D circular shells. In their scheme, as all the shaping code words have an equal number of zeros and ones independent of the size of the circular shells the data rate per signaling interval remains constant.

In our previous work [13], some practical addressing schemes to achieve (or approximate points) on the optimum tradeoff curves is given. The addressing scheme of Lang and Longstaff is further discussed (and applied to an optimally shaped constellation) by Kschischang and Pasupathy in [14] (also refer to [15]). Laroya, Farvardin and Tretter in [16] (also refer to [17]) apply ideas developed by the first two authors in the context of a type of structured

¹This work was supported by Natural Sciences and Engineering Research Council of Canada (NSERC).

²In a cubic constellation, addressing can be achieved independently along each dimension. This results in a low complexity. However, in a shaped constellation independent addressing is not applicable.

vector quantizer to constellation addressing. This results in an addressing decomposition scheme which has some similarities with the scheme of [8]. More recently, Kschischang in [18] discusses the structure of a prefix code which closely approximates the optimum nonuniform probabilities of the 2-D points.

In this paper, we introduce two efficient addressing schemes for the nearly optimum shaping of multi-dimensional signal spaces. The basic idea is to partition the signal space into the union of subsets (blocks) such that the addressing within the blocks, which is achieved using a common algorithm for all the blocks, has a low complexity. The addressing of the blocks is achieved using some recursive relationship which allows us to decompose the problem into some smaller parts each of a low complexity. The overall scheme requires a moderate amount of memory and has a small computational complexity.

First, we define two discrete sets (see also [13]):

$$F_n(K, L) \equiv \left\{ (J_0, \dots, J_{n-1}) + (0.5)^n : J_i \in [0, K-1], \& \sum_{i=0}^{n-1} J_i = L \right\}, \quad (1)$$

$$TC_n(K, L) \equiv \bigcup_{i=0}^L F_n(K, i). \quad (2)$$

We refer to (J_0, \dots, J_{n-1}) as the index vector of the points of F_n , TC_n sets.

2 Optimum shaping using a finite number of 2-D shells

In an optimally shaped region, the boundary of the 2-D subspaces is a circle, i.e., $S_2(R)$ [2]. Using K concentric circles, each $S_2(R)$ is partitioned into K shells of equal volume. The 2-D shells are indexed in the radial direction by $J \in [0, K-1]$. The average energy of the J 'th shell is $E(J) = (J + 0.5)(\Delta R)^2$ where $\Delta R = R/\sqrt{K}$, [13]. In an $N = 2n$ -D space, we obtain K^n clusters of equal volume. The average energy of the cluster indexed by (J_0, \dots, J_{n-1}) is equal to $E = (\sum_i J_i + 0.5n)(\Delta R)^2$. This means that the points of $F_n(K, L)$ represent the clusters with the equal energy. Using this fact, shaping is *optimally* achieved by selecting a subset of these points with the least $\sum_i J_i$. This results in the set $TC_n(K, L_{\max} - 1)$ and an arbitrary subset of $F_n(K, L_{\max})$ for some $L_{\max} \in [0, n(K-1)]$. This is denoted as the shaping set. If the cardinality of the shaping set is equal to T , we obtain,

$$CER_2 = \frac{K}{T^{1/n}}. \quad (3)$$

The overall shaping is optimum to the extent that the resolution of the partitioning of the 2-D subspaces allows. For small values of CER_2 (which are of practical interest), one can closely approximate the optimum tradeoff using a small K , [6] (asymptotic case), [13] (general case).

For the addressing, we need a one-to-one mapping between the set of the data values $0 \leq I \leq T-1$ and the elements of the shaping set such that the mapping and its inverse can be easily implemented. The basic idea behind the addressing methods introduced here is to partition the shaping set into the union of subsets (blocks), B_n 's, such that the addressing of the blocks (intra-block addressing) and also the addressing within the blocks (inter-block addressing) is an easy task.

The blocks are arranged in a preselected order and are indexed by $0 \leq b \leq B_{\max}$. For $b_2 > b_1$, the points of $B_n(b_2)$ correspond to larger data values comparing to the points of $B_n(b_1)$. The cardinality of the block b is denoted by $\Delta T_b = |B_n(b)|$. For each block, $0 \leq b \leq B_{\max}$, the total number of points in the preceding blocks is denoted by T_b , i.e., $T_b = \sum_{i=0}^{b-1} \Delta T_i$. For a given data value $0 \leq I \leq T-1$, we first find the index b such that $T_b \leq I < T_{b+1}$. Then the residue $R_{\alpha} = I - T_b$ is used to address a point within the block $B_n(b)$.

The shaping methods discussed here can be easily used in conjunction with the coding schemes of [19], [5]. One can also perform the shaping in a higher dimensional space and use its lower dimensional subspaces for the channel coding.

In explaining our schemes, for the sake of simplicity, we assume that the shaping set is equal to the set $TC_n(K, L_{\max})$. This can be easily generalized to the case that the $F_n(K, L_{\max})$ is partially included in the shaping set [20].

3 Blocks of identical first 2-D index

Define a block as the set of the index vectors with the same first (leftmost) component. This component serves as the block index. The key point to the addressing scheme to be introduced is that the vectors composed of the remaining $n-1$ components in block b spans the set $TC_{n-1}(K, L_{\max} - b)$. This results in $\Delta T_b = |TC_{n-1}(K, L_{\max} - b)|$. This fact is used in [13] to compute the shaping performance. Now, assume that the addressing of a data value has resulted in block index b . In this case, by replacing L_{\max} by $L_{\max} - b$, the addressing problem reduces to an equivalent form in $n-1$ dimensions. This procedure is repeated for $n-1$ steps until all the components of the index vector are computed (the n 'th component is equal to the $n-1$ 'th residue). This scheme results in a lexicographical ordering for the elements of the shaping set with respect to the components of the index vector.

In the following, we talk about the storage and the computational requirements of the method. In general, all over this paper, to facilitate the access to the lookup tables, we assume that: (i) each memory location is composed of an integral number of bytes (8 bits), and (ii) all the entries of a given table are of the same size. These restrictions increase the size of the memory.

3.1 Storage requirement

Assume that $s(i)$ is equal to the sum of the first (leftmost) i components of the index vector where $s(0) = 0$. To achieve the i 'th addressing step, $i = 1, \dots, n-1$, we need the values of $\sum_{j=0}^J |TC_{n-i}(K, L_{\max} - s(i-1) - j)|$, $J = 0, \dots, K-1$. We obtain an efficient method to store these values by considering that: (i) the preceding components affect the result only through their sum, namely $s(i-1)$, (ii) the $s(i-1)$ acts as an offset to the values and, consequently, its effect can be easily compensated by adding an appropriate offset to the corresponding data value. Using these facts, we employ a set of $n-1$ lookup tables where the i 'th table, $i = 1, \dots, n-1$, contains the values of

$$S_{s(i)} = \sum_{j=0}^{s(i)} |TC_{n-i}(K, L_{\max} - j)|, \quad 0 \leq s(i) \leq s_{\max}(i) - 1, \quad (4)$$

where, $s_{\max}(i) = \min[i(K-1), L_{\max}]$.

Considering that, $|TC_n(2^\alpha, \beta)| \leq 2^{n\alpha}$, $\forall \beta$, it is easy to show that the entries of the i 'th lookup table, $i = 1, \dots, n-1$, are at most,

$$m(i) = \min[\lceil \log_2 T \rceil, (n-i+1)\lceil \log_2 K \rceil], \quad (5)$$

bits long where T is the cardinality of the shaping set. The total memory size is equal to

$$M_t = \sum_{i=1}^{n-1} s_{\max}(i) \left\lceil \frac{m(i)}{8} \right\rceil, \quad (6)$$

bytes where $s_{\max}(i)$ and $m(i)$ are given in (4) and (5), respectively.

3.2 Computation requirement

To compensate the effect of the offset within the lookup tables, the data value involved in the i 'th addressing step, $i = 2, \dots, n-1$, is computed as,

$$I_i = R_{i-1} + \sum_{j=0}^{s(i-1)-1} |TC_{n-i}(K, L_{\max} - j)|, \quad (7)$$

where R_{i-1} is the residue of the previous step and $s(i-1)$ is the sum of all the previous components. The value of the summation in (7) is the $s(i-1)$ 'th element of the i 'th lookup table (element indexed by $s(i-1)-1$).

As the elements of the lookup tables are sorted, the search in the i 'th table requires at most $\lceil \log_2[1 + s_{\max}(i)] \rceil$ comparisons where $s_{\max}(i)$ is given in (4). We also need $n-1$ additions to determine the residues plus $n-2$ additions to determine the data values. The total number of the additions (including comparisons) per block is equal to

$$N_{add} = 2n - 3 + \sum_{i=1}^{n-1} \lceil \log_2[1 + s_{\max}(i)] \rceil. \quad (8)$$

Table (1) shows some examples of the storage and the computation requirements.

| N | L_{\max} | M_t | N_{add} | γ_s dB | PAR ₂ |
|-----|------------|-------|-----------|---------------|------------------|
| 16 | 17 | 0.2 k | 45 | 0.90 (0.92) | 3.52 |
| 32 | 29 | 1.3 k | 100 | 1.07 (1.09) | 3.66 |
| 64 | 54 | 8.9 k | 240 | 1.18 (1.21) | 3.76 |
| 128 | 104 | 65 k | 550 | 1.27 (1.30) | 3.84 |

Table 1: Parameters of the achieved point using $K=8$, $CER_2=1.5$. Column M_t denotes the memory size in bytes per N -D. Column N_{add} denotes the number of the additions per N -D (no multiplications). The values inside parenthesis are the optimum γ_s .

4 Blocks of identical binary coefficients

For a given $K = 2^k$, $N = 2n$ and integer $L \in [0, n(K-1)]$, consider k binary, n -D vectors \mathbf{g}_i , $i = 0, \dots, k-1$ where the weight of \mathbf{g}_i is equal to $w^{(i)}$ and,

$$L = \sum_{i=0}^{k-1} w^{(i)} 2^i, \quad (9)$$

where $0 \leq w^{(i)} \leq \min(n, L/2^i)$. The k -tuple $\mathbf{w} = (w^{(k-1)}, \dots, w^{(0)})$ is denoted as a binary coefficient vector of L . For $n=1$ this is the normal binary representation with k bits. For $n > 1$, the binary coefficient vector of L for $L \notin \{0, 1, n(K-1)-1, n(K-1)\}$ is not

unique. We use the set of the binary coefficient vectors of L to partition the $F_n(K, L)$ into blocks.

For a given k -tuple \mathbf{w} of the form mentioned, consider the set of the binary $k \times n$ matrices such that the weight of the i 'th row is equal to $w^{(i)}$. Assume that for a given matrix of this set, the (i, j) 'th element is the i 'th digit (coefficient of 2^j) in the binary representation of the shell index along the j 'th 2-D subspace. By permuting the elements of the rows, we obtain different matrices corresponding to a subset of points of $F_n(K, L)$. There are $\prod_{i=0}^{k-1} C_n^{w^{(i)}}$ such points, where $C_n^{w^{(i)}}$ is the combinatorial coefficient, and their union results in one block.

We assume that the blocks have a lexicographical ordering according to the elements of the binary coefficient vector. For a given data value I , the intra-block addressing is the determination of the label b (and \mathbf{w}_b) such that $T_b \leq I < T_{b+1}$. This is explained in the following:

4.1 Intra-block addressing

Assume that the shaping set is equal to $TC_n(2^k, L_{\max})$. If the most significant component of \mathbf{w} is known to be equal to $w^{(k-1)}$, the shaping set reduces to the union of $C_n^{w^{(k-1)}}$ of its subsets each of cardinality $|TC_n(2^{k-1}, L_{\max} - w^{(k-1)}2^{k-1})|$. Each of these subsets is the collection of the integer n -tuples with $n - w^{(k-1)}$ components having value in the range $[0, 2^{k-1} - 1]$ and $w^{(k-1)}$ components having value in the range $[2^{k-1}, 2^k - 1]$ and the sum of the components is less than or equal to L_{\max} .

In this case, considering the lexicographical ordering of the blocks, $w^{(k-1)}$ is selected as the largest integer satisfying,

$$R_{k-1} = I_{k-1} - \sum_{i=0}^{w^{(k-1)}-1} C_n^i |TC_n(2^{k-1}, L_{\max} - i2^{k-1})| \geq 0, \quad (10)$$

where $I_{k-1} = I$ is the data value. Then, the residue R_{k-1} is expanded as,

$$R_{k-1} = P_{k-1} |TC_n(2^{k-1}, L_{\max} - w^{(k-1)}2^{k-1})| + I_{k-2}, \quad (11)$$

where,

$$\begin{aligned} 0 \leq I_{k-2} &< |TC_n(2^{k-1}, L_{\max} - w^{(k-1)}2^{k-1})|, \\ 0 \leq P_{k-1} &< C_n^{w^{(k-1)}}. \end{aligned} \quad (12)$$

The value P_{k-1} is used in the inter-block addressing to permute the \mathbf{g}_{k-1} . After \mathbf{g}_{k-1} is known, the shaping set reduces to the set $TC_n(2^{k-1}, L_{\max} - w^{(k-1)}2^{k-1})$ shifted by an offset vector equal to $2^{k-1}\mathbf{g}_{k-1}$. The value I_{k-2} is used to address one point of this set. By replacing k by $k-1$ and L_{\max} by $L_{\max} - w^{(k-1)}2^{k-1}$ in the original problem, the same algorithm is used to find $w^{(k-2)}$. The procedure is repeated for k steps until all the elements of the coefficient vector are computed. The expansion of the residue in (11) needs one multiplication, one division and one addition per recursion step times $k-1$ steps (no expansion in the last step).

4.1.1 Tradeoff between the storage and the computational complexities

The major complexity of this method is that of calculating the summation in (10). The alternative is to use a lookup table to store the required results. We also need a block of memory to store the combinatorial coefficients. This block is used in the inter-block addressing and will be computed later. The total memory size is shown in column M_t of Tables (2) and (3). In this case, the

number of the multiplications, divisions per block is equal to $k - 1$. Columns N_{add} , N_{mul} in Tables (2) and (3) show the corresponding computational complexity.

As an intermediate solution, it is also possible to store a subset of the values of the summation and to compute the intermediate cases. This results in a tradeoff between the storage and the computational complexities.

| N | L_{max} | B_{max} | A_{max} | N_{add} | N_{mul} | M_t | γ_s dB | PAR ₂ |
|-----|-----------|-----------|-----------|-----------|-----------|--------|---------------|------------------|
| 16 | 17 | 139 | 12 | 50 | 4 | 0.11 k | 0.90(0.92) | 3.52 |
| 32 | 29 | 612 | 22 | 85 | 4 | 0.45 k | 1.07(1.09) | 3.66 |
| 64 | 54 | 3564 | 43 | 160 | 4 | 2.6 k | 1.18(1.21) | 3.76 |
| 128 | 104 | 23966 | 84 | 300 | 4 | 17 k | 1.27(1.30) | 3.84 |
| 256 | 202 | 170313 | 165 | 575 | 4 | 116 k | 1.32(1.35) | 3.88 |

Table 2: Parameters of the achieved point using $K=8$, $CER_2=1.5$. Column M_t denotes the total memory size in bytes per N -D. Column N_{add}/N_{mul} denotes the number of the additions/multiplications (including divisions) per N -D. The values inside parenthesis are the optimum γ_s .

| N | L_{max} | B_{max} | A_{max} | N_{add} | N_{mul} | M_t | γ_s dB | PAR ₂ |
|-----|-----------|-----------|-----------|-----------|-----------|--------|---------------|------------------|
| 16 | 9 | 25 | 8 | 35 | 2 | 0.04 k | 0.76(0.81) | 2.84 |
| 32 | 16 | 72 | 16 | 60 | 2 | 0.18 k | 0.90(0.96) | 2.94 |
| 64 | 28 | 210 | 28 | 100 | 2 | 1.1 k | 1.00(1.06) | 3.00 |
| 128 | 53 | 729 | 53 | 195 | 2 | 6.6 k | 1.07(1.15) | 3.05 |
| 256 | 103 | 2704 | 103 | 375 | 2 | 40 k | 1.12(1.20) | 3.09 |

Table 3: Parameters of the achieved point using $K=4$, $CER_2=1.25$. Column M_t denotes the total memory size in bytes per N -D. Column N_{add}/N_{mul} denotes the number of the additions/multiplications (including divisions) per N -D. The values inside parenthesis are the optimum γ_s .

4.2 Inter-block addressing

The next step is to provide a mapping between the integer numbers P_i , $0 \leq P_i < C_n^{w(i)}$, and the set of the binary n -tuples of weight $w^{(i)}$. Such a mapping is discussed in detail in [21]. The basic theorem is as follows [21]:

Theorem: Consider the set $G(n, w)$ of binary sequences $\mathbf{g} = (g_0, \dots, g_{n-1})$ of length n and weight w . Define the partial weight as, $w_k = \sum_{i=k}^{n-1} g_i$. The binary sequences $\mathbf{g} \in G(n, w)$ can be ordered according to

$$P(\mathbf{g}) = \sum_{k=0}^{n-1} g_k C_{n-k-1}^{w(k)}, \quad (13)$$

where $C_n^w = 0$ for $w > n$ and $0 \leq P(\mathbf{g}) < C_n^w$. A similarly motivated combinatorial coding scheme was employed in [22] in a speech coding context.

We assume that the C_{n-i}^w 's, $1 \leq i \leq n-1$, are precomputed and stored. In general, as $C_m^w = C_m^{m-w}$, just the values of C_m^w for $w \leq 1 + \lfloor m/2 \rfloor$ are stored. Using the identities $\sum_w C_m^{2w} = \sum_w C_m^{2w+1} = 2^{m-1}$, we obtain $C_m^w < 2^{m-1}$, $\forall w$. This means that the C_m^w 's, $\forall w$, can be represented with $m-1$ bits. This results in the memory size,

$$M_c = \sum_{i=1}^{n-1} \left(1 + \left\lfloor \frac{n-i}{2} \right\rfloor \right) \left\lceil \frac{\min[n-i-1, \lceil \log_2 T \rceil]}{8} \right\rceil, \quad (14)$$

bytes where T is the cardinality of the shaping set.

The whole mapping requires at most $k(n-1)$ comparisons and $A = \sum_i w_i$ additions. Column A_{max} in Tables (2), (3) shows the maximum value of $\sum_i w_i$ over $0 \leq L \leq L_{max}$.

Note: In [20], the addressing scheme using blocks of identical binary coefficients discussed here is generalized to the non-binary case.

5 Comparison with other techniques

In the following, we compare our schemes with that of [4], [6], [7], [13], [16]. As we are essentially able to closely approximate any point up to the knee of the optimum tradeoff curves, in Table (4), we have compared some of the values obtained in [4] and [6] with the optimum values calculated in [2]. A four state trellis diagram

| N | γ_s | VC, [4] | | C/O, [6] | | |
|-----|------------|------------|------------|------------|------------|------------|
| | | CER_s | PAR | γ_s | CER_s | PAR |
| 4 | 0.37 | 1.41(1.09) | 4.62(2.27) | — | — | — |
| 8 | 0.65 | 2.00(1.26) | 6.98(2.81) | — | — | — |
| 12 | 0.75 | 3.00(1.26) | 8.24(2.86) | 0.63 | 1.55(1.13) | 3.42(2.51) |
| 16 | 0.81 | 1.54(1.24) | 5.58(2.88) | 0.69 | 1.45(1.14) | 3.02(2.55) |
| 24 | 1.03 | 5.20(1.50) | 15.2(3.67) | 0.80 | 1.50(1.16) | 3.46(2.67) |
| 32 | 0.85 | 1.35(1.16) | 4.94(2.70) | 0.86 | 1.46(1.17) | 3.40(2.72) |

Table 4: Comparison between the Voronoi constellations (VC) and the Calderbank, Ozarow method (C/O) with the optimum constellations, the values in parenthesis are the optimum values of CER_s , PAR for the given γ_s .

of [7] (in conjunction with the peak constraint technique) achieves $\gamma_s = 0.97$ dB, $CER_s = 1.5$, $PAR_2 = 3.75$.

In [16], an example for $N = 64$ is given which needs 15 multiply-adds per 2-D, together with a memory of 1.5 kilo-bytes/ N -D, to achieve a tradeoff point with $CER_2 = 1.5$ near to the optimum curve (the optimum γ_s for $N = 64$, $CER_s = 1.5$ is equal to 1.2 dB).

Table (5) shows the performance and the complexity of our schemes for $N = 64$.

| Method | CER_s | N_{add} | N_{mul} | M_t | γ_s dB |
|---------------------|---------|-----------|-----------|-------|---------------|
| Blocks of identical | 1.25 | 210 | 0 | 3.0 k | 1.00(1.06) |
| first 2-D index | 1.5 | 240 | 0 | 8.9 k | 1.18(1.21) |
| Blocks of identical | 1.25 | 100 | 2 | 1.1 k | 1.00(1.06) |
| binary coefficients | 1.5 | 160 | 4 | 2.6 k | 1.18(1.21) |

Table 5: Performance and complexity of the two schemes for $N = 64$, $CER_s = 1.25/1.5$. Column M_t denotes the memory size in bytes per N -D. Column N_{add}/N_{mul} denotes the number of the additions/multiplications (including divisions) per N -D. The values inside parenthesis are the optimum γ_s .

The addressing decomposition method of [13] uses a pure lookup table (no associated computation) to achieve nearly optimum tradeoff points. Table (6) shows some examples of the performance of this scheme.

For $N \leq 32$, the selection between the methods introduced here and that of [13] is essentially a matter of tradeoff between the computational and the storage complexities. However, in higher dimensional spaces, the present methods are superior.

| N | CER _s | M_t | γ_s , dB | PAR |
|-----|------------------|-------|-----------------|------|
| 16 | 1.25 | 1.5 k | 0.79 (0.81) | 2.86 |
| 16 | 1.35 | 1.0 k | 0.84 (0.87) | 3.12 |
| 16 | 1.50 | 2.6 k | 0.90 (0.92) | 3.52 |
| 32 | 1.25 | 2.5 k | 0.92 (0.96) | 2.95 |
| 32 | 1.35 | 2.0 k | 0.95 (1.03) | 3.21 |
| 32 | 1.35 | 12 k | 1.00 (1.03) | 3.25 |

Table 6: Parameters of the point achieved using the addressing decomposition method of [13]. The values inside parenthesis are the optimum γ_s . Column M_t is memory size in kilo-bytes/ N -D (no computation).

6 Summary and conclusions

We have introduced two efficient addressing schemes for the nearly optimum shaping of multi-dimensional signal spaces. These schemes are based on the observation that the average energy of a 2-D shell can be replaced by its index. Using this fact, we have partitioned the signal space into blocks such that the inter-block addressing has a low complexity. The intra-block addressing is achieved using some recursive relationship which allows us to decompose the problem into smaller parts each of a low complexity.

As an example of performance, in an $N = 128$ dimensional space, by increasing the number of the 2-D points by 50% and using a relatively simple algorithm, we realize a shaping gain of 1.27 dB. It seems that after the initial coding gain associated the Ungerboeck's schemes [19] or with the Wei's schemes [5] this is the easiest way to obtain higher gains.

References

- [1] G. D. Forney, Jr. and L. F. Wei, "Multidimensional constellations—Part I: Introduction, figures of merit, and generalized cross constellations," *IEEE J. Select. Areas Commun.*, vol. SAC-7, pp. 877–892, August 1989.
- [2] A. K. Khandani and P. Kabal, "Shaping multi-dimensional signal spaces—Part I: optimum shaping, shell mapping," To appear in the *IEEE Trans. Inform. Theory*.
- [3] J. H. Conway and N. J. A. Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 106–109, January 1985.
- [4] G. D. Forney, Jr., "Multidimensional constellations—Part II: Voronoi constellations," *IEEE J. Select. Areas Commun.*, vol. SAC-7, pp. 941–958, August 1989.
- [5] L. F. Wei, "Trellis coded modulation with multidimensional constellations," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 483–501, July 1987.
- [6] A. R. Calderbank and L. H. Ozarow, "Nonequiprobable signaling on the Gaussian channel," *IEEE Trans. Inform. Theory*, vol. IT-36, pp. 726–740, July 1990.
- [7] G. D. Forney, "Trellis shaping," *IEEE Trans. Inform. Theory*, vol. IT-38, pp. 281–300, March 1992.
- [8] G. R. Lang and F. M. Longstaff, "A leech lattice modem," *IEEE J. Select. Areas Commun.*, vol. SAC-7, pp. 968–973, August 1989.
- [9] T. R. Fischer, "A pyramid vector quantizer," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 568–583, July 1986.
- [10] F. R. Kschischang and S. Pasupathy, "Optimal nonuniform signaling for Gaussian channels," Submitted to *IEEE Trans. Inform. Theory*, Nov. 1990.
- [11] J. R. Livingston, "Shaping using variable-size regions," *IEEE Trans. Inform. Theory*, vol. IT-38, pp. 1347–1353, July 1992.
- [12] A. R. Calderbank and M. Klimesh, "Balanced codes and nonequiprobable signaling," *IEEE Trans. Inform. Theory*, vol. IT-38, pp. 1119–1122, May 1992.
- [13] A. K. Khandani and P. Kabal, "Shaping multi-dimensional signal spaces—Part II: shell-addressed constellations," To appear in the *IEEE Trans. Inform. Theory*.
- [14] F. R. Kschischang and S. Pasupathy, "Optimal shaping properties of the truncated polydisc," Submitted to *IEEE Trans. Inform. Theory*, 1992.
- [15] F. R. Kschischang, *Shaping and coding gain criteria in signal constellation design*, Ph.D. Dissertation, Toronto Univ., June 1991.
- [16] R. Laroia, N. Farvardin and S. Tretter, "On SVQ shaping of multidimensional constellations—high-rate large-dimensional constellations," *Twenty-sixth annual conference on information sciences and systems*, Princeton, NJ, pp. 527–531, 19–20 March 1992.
- [17] R. Laroia, "On optimal shaping of multidimensional constellations—an alternative approach to lattice-bounded (voronoi) constellations," Submitted to *IEEE Trans. Inform. Theory*, November 1991.
- [18] F. R. Kschischang, "Huffman codes for shaping gain," *16th Biennial Symposium on Communications*, pp. 79–82, Kingston, Ontario, 27–29 May 1992.
- [19] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–67, 1982.
- [20] A. K. Khandani and P. Kabal, "Efficient Block-based Addressing Schemes for the Nearly Optimum Shaping of Multi-dimensional Signal Spaces, Combinatorial Addressing," Submitted to *IEEE Trans. Inform. Theory*.
- [21] J. P. M. Schalkwijk, "An algorithm for source coding," *IEEE Trans. Inform. Theory*, vol. IT-18, No. 3, pp. 395–399, May 1972.
- [22] M. Berouti, H. Garten, P. Kabal and P. Mermelstein, "Efficient computation and encoding of the multipulse excitation for LPC," *Proc. of IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (San Diego, CA), pp. 10.1.1–10.1.4, April 1984.