

Application of Shaping Technique to Multi-level Turbo-coded Modulation

Amir K. Khandani and W. Tong

Department of Electrical and Computer Engineering

University of Waterloo

Waterloo, Ontario, Canada, N2L 3G1

Technical Report UW-E&CE# 02-07

May 30, 2002

Abstract

This article presents a method for the combined Turbo-code and QAM modulation using coset coding. A symbol-based Turbo-code acting on symbols of size 4 is used to select a coset along each one-D sub-constellation (using the partition $Z/4Z$). The proposed method satisfies the conditions for the separability of shaping and coding, and consequently, can be easily integrated with a shaping algorithm. Numerical results are presented which show that the proposed configuration outperforms the best known techniques in terms of the complexity and BER performance. The proposed method also demonstrates an excellent performance near the error floor region.

Keywords: Combined Coding and Modulation, Shaping, Turbo-codes, Symbol-based Interleaving

I. INTRODUCTION

A. Combined Coding and Modulation

Coded modulation goes back to 1974, when Massey suggested the notion of improving system performance by looking at coding and modulation as a combined entity [1]. The most successful step towards implementing Massey's thoughts was the invention of *Trellis-Coded Modulation* (TCM) by Ungerboeck [2]. A distinct feature of TCM is the application of labeling by set partitioning where the constellation is successively binary partitioned into congruent subsets defining a mapping of binary addresses to the signal points. Almost at the same time and independent of [2], Imai and Hirakawa [3] proposed *Block-Coded Modulation* (BCM) using multi-level binary coding. This is based on using a set of parallel binary block codes to specify different bit values in a given binary labeling of the constellation points. The basic structure proposed in [3] has been further studied and generalized in a number of subsequent research works. Reference [4] which is the state of the art article in this category includes a detailed review of the relevant literature. It is also a well established fact that such a multi-level coding scheme can approach the maximum theoretically achievable rate associated with a set of constellation points¹ if: (i) the rate of the underlying binary codes are adjusted properly, and (ii) these codes are decoded sequentially where each decoder takes advantage of the decoded bit value produced by its earlier stages [4].

It should be also mentioned that some of the generalizations reported in the literature (e.g., [5][6][7]) almost remove the thin line of distinction between the above two families of schemes (namely those emerging from [2] and [3], respectively) and explain all known methods of combined coding and modulation under the same umbrella.

¹ This is defined as the mutual information across the channel assuming that the constellation points are used with equal probability.

B. Shaping

Shaping concerns the selection of the boundary of a multi-dimensional signal constellation to reduce its average energy. It is well known that by using shaping techniques in conjunction with conventional band-width efficient modulation methods, one can achieve an additional shaping gain (refer to [8] for definition) of about one dB with a modest increase in the complexity.

In the work of Wei [9], shaping is a side effect of the method employed to transmit a non-integral number of bits per two dimensions. Forney and Wei generalized this method in [8]. Conway and Sloane in [10] introduced the idea of the Voronoi constellation based on using the Voronoi region of a lattice Λ_s as the shaping region. Voronoi constellations are further studied by Forney in [11]. In [12], Calderbank and Ozarow introduced a shaping method in which the 2-D subspaces are partitioned into equal sized shells of increasing average energy where a shaping code is used to specify the sequence of the 2-D shells. Lang and Longstaff in [13] use an addressing scheme which first divides the final constellation into energy shells. Then, a point in a shell is found by successively decomposing the space into lower-dimensional subspaces via generating function techniques. The addressing scheme of Lang and Longstaff is further discussed and elaborated in [14]-[18]. The idea of the trellis shaping is introduced in [19]. This is based on using an infinite dimensional Voronoi region, determined by a convolutional code, to shape the constellation. In [20], Kschischang and Pasupathy discuss a shaping method which is based on using the 2-D points with non-equal probability. In [16][21][22] some very low complexity addressing schemes are given which achieve (or closely approximate) points on the optimum tradeoff curves.

The method reported in [21] has been the lowest complexity addressing method for spaces of dimensionality up to 32, and the method reported in [22] has been the lowest complexity addressing method for spaces of higher dimensionality (up to 512). In the current article, we present an addressing scheme based on the method proposed in [21]. This reference discusses using a lookup table for non-uniform merging of points into shaping partitions where the final constellation is formed in a hierarchy of stages involving the Cartesian product of the lower dimension subspaces. Reference [21], however, does not provide any method for the indexing of the elements of the table. In the current article, we present a very simple and practical method based on the enumeration of the terminal nodes in a binary Huffman tree for this purpose. This technique is explained by a detailed example. This clarifies the details of the underlying addressing algorithm which offers a lower complexity as compared to all the methods reported in the literature.

C. Turbo-code

In 1993, a new class of channel codes, called Turbo-codes, were announced which have an astonishing performance with a reasonable complexity [23]. The basic idea of Turbo-codes is to make use of some convolutional component codes which are connected in parallel through pseudo-random interleavers. There have been a number of successful research works addressing the problem of band-width efficient modulation in conjunction with Turbo-codes [24]-[58].

One class of methods for combining Turbo-codes with multi-level modulation is based on a direct mapping of the output bits of a Turbo encoder to the constellation points via an interleaver [24]-[35]. In these schemes, Turbo decoding is achieved in two stages. First, the reliability values corresponding to the bits are extracted by adding up the probability of the corresponding constellation points. Then, these bit reliability values are passed to a conventional Turbo-decoder for iterative decoding. The complexity of these methods grows with the size of the constellation (spectral efficiency) due to the step required in extracting the reliability values. In addition, the extra interleaving stage required between the binary encoder and the constellation (to reduce the dependency between the adjacent bits mapped to the same constellation) adds to the overall system complexity. These methods perform well for small values of spectral efficiency, however, it is well known that the corresponding coding gain drops as the spectral efficiency increases (this drop in the coding gain is due to the dependency between bits) [24][28].

References [36]-[55] present structures based on parallel concatenation of trellis-coded modulation (TCM) schemes. References [36]-[53] are based on binary component codes, while references [54][55] are based on codes constructed from rings. In [36][37][51]-[53], the interleaver operates on bits, while in [38]-[50][54][55] the interleaver operates on symbols. The use of symbol interleaving has the dis-advantage of reducing the effective interleaving gain, however, it offers some distinct advantages in terms of performance [46][47][60]-[62]. This usually results in a net gain for symbol vs. bit interleaving. In addition, in [46][47], it is shown that using a symbol-based interleaver with two bits interleaving also results in a reduction in the size of the memory required for Turbo decoding.

A class of methods for combining Turbo-codes with multi-level modulation has initiated from the work of [39]-[41] and continued in [43]-[45]. In these methods, to produce a systematic encoder, the interleaver maps even symbol positions to even symbol positions and odd ones to odd. The output of the second encoder is de-interleaved and the output symbols from each encoder are

punctured alternatively. This odd-to-odd and even-to-even interleaving (first introduced in [59]) is equivalent to using two independent interleavers of half the original length, operating on the even and odd positions. This structure further reduces the effective length and consequently the interleaving gain of these schemes, as is also observed in [43]. A solution to remove this constraint is proposed in [43].

Note that the trellis branching factor in [38]-[40][42][46]-[50][54][55] depends on the number of constellation points and consequently the complexity of the soft output decoder for the underlying trellis can be substantially higher than the standard binary Turbo code. This problem is solved in [41][52][53] and in the current article by allowing for some uncoded bits. In this case, the underlying binary component codes operate on the cosets (and not on the individual constellation points), and consequently, the decoding complexity does not grow with the size of the constellation.

Reference [57] presents a multi-level Turbo-coded modulation scheme where different labeling bits of a signal constellation are assigned to different Turbo-codes of appropriate code rates. The code rates are selected using information theoretic arguments involving channel cutoff rate and are implemented by a puncturing of a basic binary Turbo-code structure. This scheme requires binary codes of different rates to be implemented over different labeling bits, and consequently, it is not practically easy to implement. This configuration also results in a reduction in the effective code-word length, and consequently, a reduction in the achievable interleaver gain.

The above references discuss a variety of successful techniques to combine Turbo-codes with multi-level modulation. Many of these schemes perform fairly close to the maximum rate achievable for their underlying signaling scheme². However, some of these schemes have certain shortcomings, including: (i) can not be easily scaled (i.e., their complexity increases with an increase in the spectral efficiency)[24]-[35][38]-[40][46]-[51][54][55], and/or (ii) performance drops with an increase in the spectral efficiency[24]-[35], and/or (iii) show an early error floor [48][49] (and to some extent [39]-[41]), and/or (iv) impose certain restrictions on the structure of the interleaver reducing the effective interleaver length and the corresponding interleaving gain [39]-[41][57].

This article presents methods for the combined Turbo-code and QAM modulation using coset coding where the underlying binary code is a symbol-based Turbo-code which operates on the basis of two-bits by two-bits interleaving (regardless of the size of the constellation). The use of two-bits

²This means that the effect of shaping is neglected and the maximum achievable rate is computed assuming that the constellation points are used with equal probability.

by two-bits interleaving results in reducing the size of the required memory as discussed in [46][47]. The proposed method satisfies the conditions for the separability of shaping and coding in [8], and consequently, can be easily integrated with a shaping algorithm. The only prior work which discusses the application of shaping to Turbo-codes is reference [50] which is based on a direct mapping of bits (same category of work as reported in [24]-[35]) to the points of a nonuniform constellation. However, the shaping gain achievable using the method proposed in [50] is limited to about 0.2 dB. In addition, the signal constellations with non-uniform positioning of points may not be easy to implement. The problem of shaping in conjunction with Turbo-codes is also briefly mentioned in [4].

II. PROPOSED METHOD

A common approach to Turbo-coded modulation is based on a direct mapping of the output bits of a binary Turbo-code to the points of a base constellation via an interleaver (the role of the interleaver is to reduce the dependency between the bits mapped to the points of the base constellation) [24]-[35]. In this configuration, the channel can be considered as m binary input, continuous output channels (the output of each such binary channel is the corresponding bit LLR), where 2^m is the number of points in the base constellation. These binary channels are not necessarily identical and also their outputs are dependent on each other. As we will see in the following, this dependency between the bits plays a crucial role in the achievable performance.

The value of the achievable rate over the equivalent binary channels, namely the value of the mutual information associated with different bits, can be easily computed as explained in [4]. Figure 1 shows an example of such calculation for a 4-point one-dimensional constellation using both Gray and Natural labeling. Referring to Fig. 1, we observe that there is a substantial gap of about 1.5dB between the overall achievable rate of the underlying binary channels using the two methods of labeling. This may appear in contradiction with the result of [4] which using chain rule of mutual information shows that the overall achievable rate is independent of the labeling and is indeed equal to the mutual information associated with the 4-point constellation. Note that the method explained in [4] requires that the underlying binary channels are decoded sequentially where each decoder takes advantage of the decoded bit value produced by its earlier stages. It is also observed that this gap becomes smaller for higher values of signal-to-noise ratio. Referring to Fig. 1, we also observe that there exists a noticeable difference between the achievable rate associated with different bits.

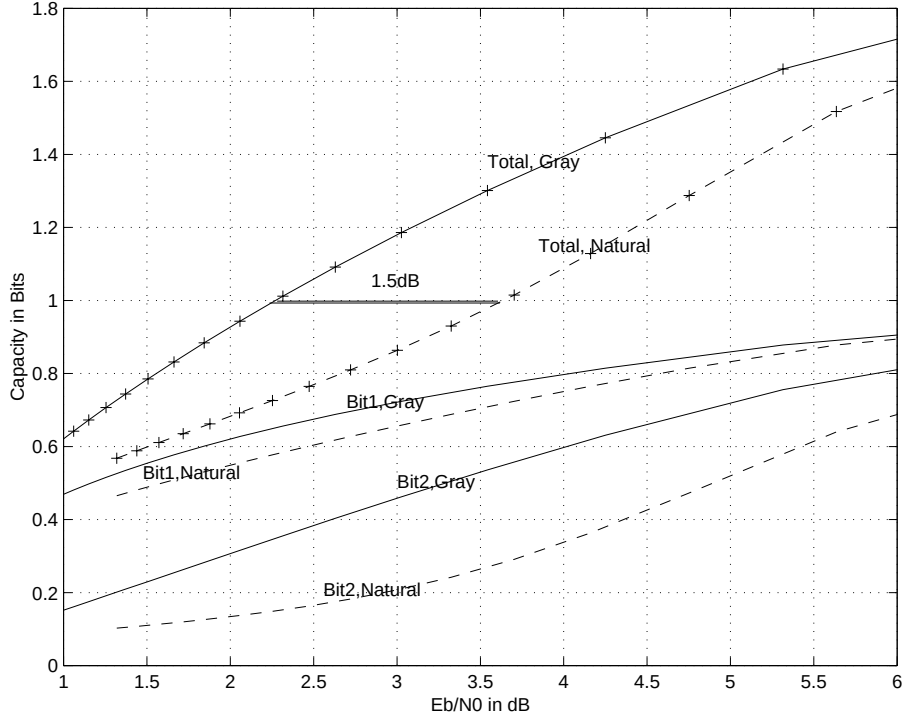


Fig. 1. Achievable rate over the binary channels associated with the bit labels in a one-D constellation with four points (assuming independent decoding).

Figure 2 shows different configurations for the overall achievable rate of a 4-point one-dimensional constellation. This figure also contains the capacity of an AWGN channel for the sake of comparison. The two curves labeled as “binary independent channels” are simply obtained by adding the mutual information associated with the corresponding equivalent binary channels. This indicates the achievable rate if the underlying binary channels are *decoded* independent of each other. The curve labeled as “4-point constellation with equal probability of points” is the mutual information associated the original constellation when the constellation points are used with equal probability. This is indeed the sum of the achievable rates of the underlying binary channels if these are coded independently (using different code rates for the underlying binary channels), but decoded sequentially where each decoder takes advantage of the decoded bit value produced by its previous decoders. The difference between this curve and the two curves labeled as “binary independent channels” accounts for the effect of the memory between underlying binary channels (note that this difference is substantially higher for the Natural labeling). We intent to improve the performance by finding a method to exploit this dependency between bits in the decoding process. The curve labeled as “4-point constellation with optimum probability of points” is the mutual information

associated with the 4-point constellation maximized over the input probability assignment. The difference between this curve and those labeled as “4-point constellation with equal probability of points” accounts for the effect of shaping (using the constellation points with non-equal probability). Note that the shaping gain here is limited to about 0.5dB because the number of constellation points is small.

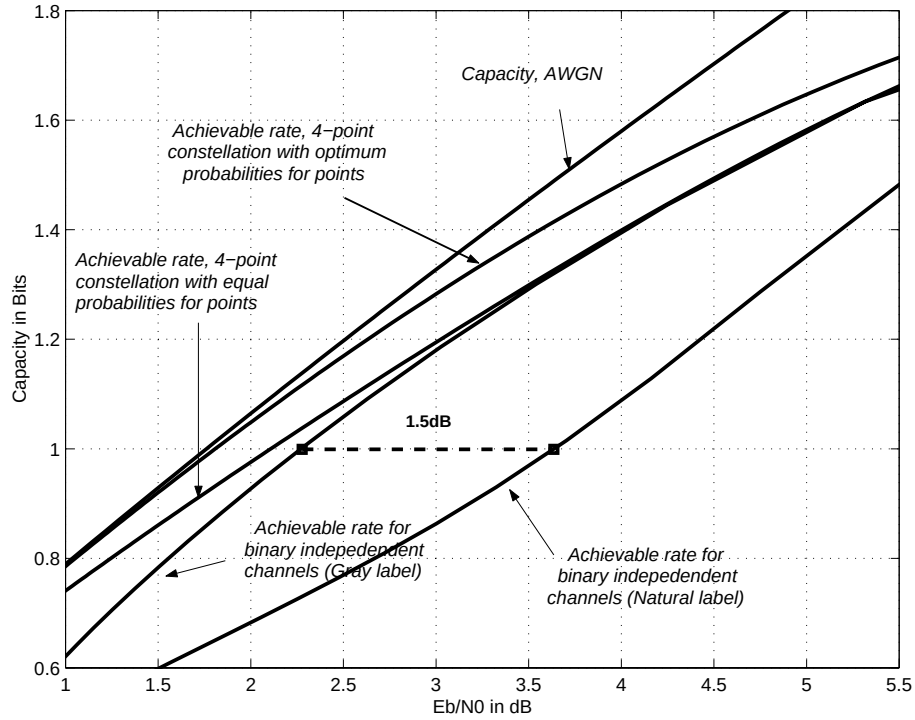


Fig. 2. Achievable rates in a one-D constellation with four points using different configurations, as well as the maximum achievable rate in an AWGN channel.

In general, to apply shaping, the points in the base sub-constellations are partitioned into a set of shaping partitions of increasing average energy where the shaping algorithm uses part of the input bits to select a sequence of shaping partitions in the Cartesian product of the base sub-constellations. Similarly, to apply channel coding, the set of the points in the base sub-constellations are divided into a set of coding partitions where the channel coding algorithm selects a sequence of the coding partitions in the Cartesian product of the base sub-constellations. It is well known that the shaping and the coding algorithms can operate independent of each other under the condition that each shaping partition contains an equal number of points from each coding partition [8]. In this case, the shaping block will use a subset of input bits to select a shaping partition along each base sub-

constellation and then the channel coding part will use the rest of the bits to select a point within each of the selected shaping partitions. Note that a major assumption to have independence between shaping and coding algorithms is that the shaping bits are not channel coded.

The case of two bit interleaving (used in the current article) is especially attractive because the number of branches starting from each state will be equal to four, resulting in two branches per bit. As a result, the computational complexity (per bit) of the forward-backward algorithm on the resulting trellis will be the same as the standard case of single bit interleaving, while the effective length of the trellis reduces by a factor of two, reducing the memory required to store the state probabilities in the forward-backward algorithm by a factor of two [47]. As an example, in Fig. 4, a 16-point constellation is decomposed into 4 cosets using the partition chain $Z/4Z$. These cosets are labeled by two bits using Gray labeling and are selected by the output bits of a symbol-based Turbo-code with $m = 2$. For larger values of m , the complexity of the resulting trellis for the symbol-based interleaving increases exponentially with m , making it infeasible.

In conventional shaping algorithms used with TCM, the shaping partitions are created by using a set of concentric circles in 2-D sub-constellations. In this case, the coset decomposition required for channel coding is also applied in 2-D by partitioning of Z^2 using its subsets obtained by scaling and/or 45 degrees rotation. However, in our case, the shaping and coding partitions are applied to the one-D sub-constellations. The reason is that we would like to have a 4-way partition to apply a symbol-based Turbo-code with four symbols. However, it turns out that using a 4-way partition in 2-D results in a small minimum distance within cosets which will be problematic in our case. Figure 4 shows such a structure for the 16 points constellation satisfying the separability condition. The achievable shaping gain for this constellation can be computed using the technique given in [16]. The result for a constellation of dimension 32 as a function of Constellation Expansion Ratio (refer to [8] for definition) and for 4 one-D shaping partitions is shown in Fig. 3. The same graph also contains the shaping gain achievable in 32 dimensions, as well as the shaping gain in an infinite dimensional space assuming continuous approximation [15]. It is observed that by using 4 one-D shaping partitions, one can achieve most of the available shaping gain in 32 dimensions. Note that the curves corresponding to optimum shaping in 32 or ∞ dimensions are based on circular 2-D sub-constellations (conventional shaping) and consequently start from a shaping gain of 0.2 dB (gain of a circle in 2-D), while the curve corresponding to the method proposed here starts from zero dB. It should be added that in all three curves the CER is measured in two dimensions.

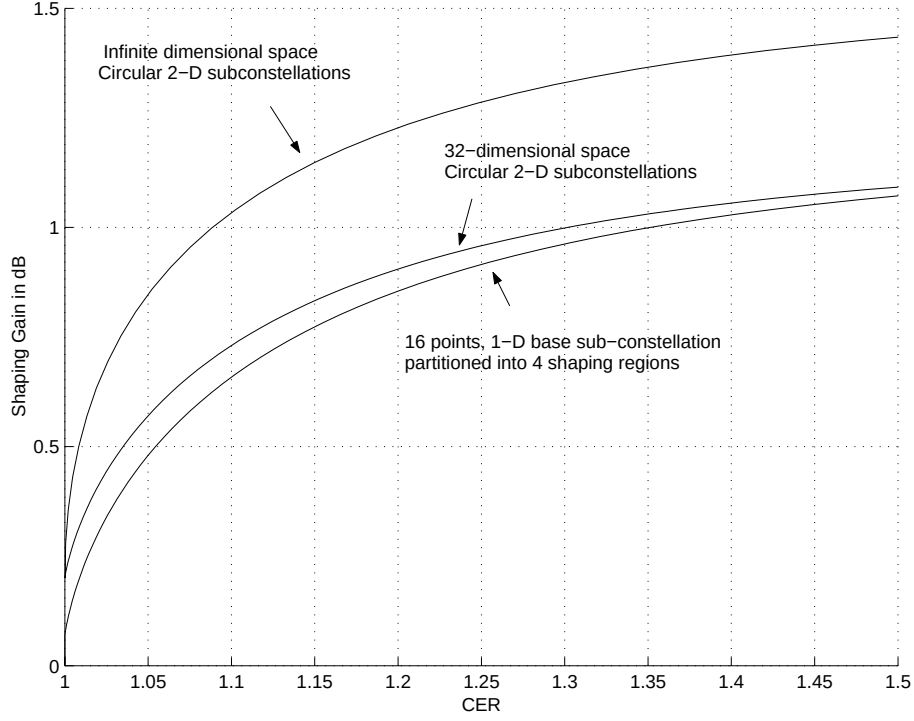


Fig. 3. Shaping gain in dB vs. CER for a 2-D sub-constellation with 256 points in a 32-D space.

Such a coset decomposition of a large constellation results in the complexity of the symbol-based decoder to be independent of the number of constellation points (i.e., determined by the number of cosets). The decoding proceeds by: (i) detecting the sequence of cosets by decoding the Turbo-code (recovering part of the input bits), then (ii) detecting a point within the selected cosets, and finally, (iii) recovering the second sequence of bits using the inverse of the addressing scheme used for shaping. The probability values required to initialize the Turbo decoder are computed by adding the probability of points within each coset, and then the conventional iterative decoding procedure follows.

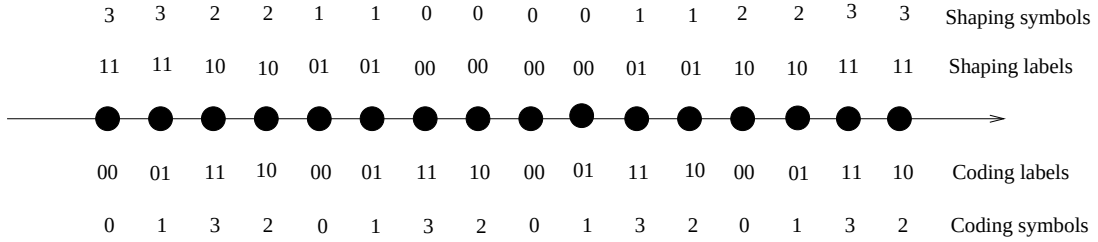


Fig. 4. A one-D constellation of 16 points divided into 4 cosets with related coding and shaping labels.

As far as coding within a coset is concerned, in many situations, one does not need any extra level of error protection because the distance between the points is already very large. For example, in the constellation in Fig. 4, the square distance with a coset is 16 times larger than the distance of the original constellation. However, in the case of using Turbo-code with large coding redundancy, due to the large gain of these codes, the distance within a coset may not be sufficient to guarantee the desired probability of error (requiring a second level of error protection within the cosets). The corresponding coding scheme is composed of two independent coders, one symbol-based Turbo-code (using four symbols) of an appropriate rate selecting the sequence of cosets and a second coder selecting a point within the coset. Decoding proceeds by first detecting the sequence of cosets by decoding the Turbo-code (recovering part of the input bits). Then, knowing the sequence of cosets, the decoder for the second channel code will find the final point within the coset and recovers the sequence of shaping bits using the inverse of the addressing scheme. We refer to this method as the sequential decoding. Using the results of [4], we know that using the method of sequential decoding, one can exploit the full achievable rate over the corresponding constellation. Figure 5 shows the achievable rate in a 16-point one-D constellation divided into 4 cosets for sequential decoding. It is observed that for a redundancy of about one bit per one-D (or lower) in the coding of cosets (point specified by a circle on Fig. 5), the achievable rate for the signaling within cosets is already well in its saturation region (flat portion of the curve), meaning that the underlying channel requires negligible coding redundancy to achieve reliable transmission. This means that the coding within cosets is in most cases not needed. Note that coding redundancies in excess of 1 bit per one-D are very unlikely to be used in practice due to requiring a signal constellation of a large size. The above claim is supported by the fact the minimum square distance with the cosets is equal to 16 indicating a gain of 12 dB in square distance.

In general, one can use any lattice or Trellis code for coding within cosets. It is usually sufficient to use a simple lattice code, for example based on the lattice D_n , for coding within cosets. Lattice D_n is based on the single parity check code and results in doubling the minimum square distance within a coset. Lattice D_n also enjoys a very simple soft decision decoding algorithm using Wagner rule. This decoding rule is based on performing a hard decision along each co-ordinate and accepting the resulting point if the parity is satisfied, or selecting the next closets point (of a different parity) along the co-ordinate for which the initial hard decision is least reliable. This operation will flip the overall parity, resulting in a valid element of the D_n lattice.

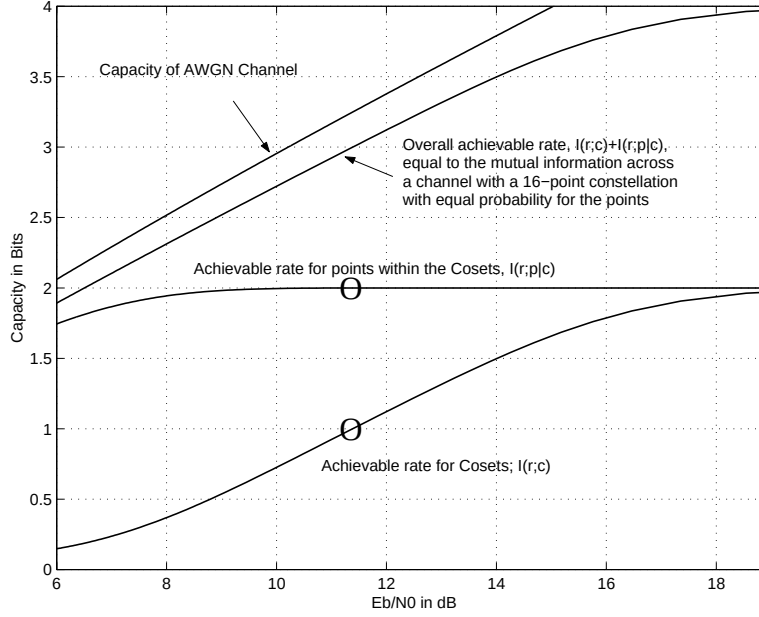


Fig. 5. Achievable rate in a one-D constellation with 16 points divided into 4 cosets (assuming sequential decoding).

The major drawback for using an extra level of channel coding within cosets is that it can not be easily integrated with a shaping algorithm. This means that the desirable situation of having independence between the channel coding and the shaping which results in some simplification is not valid any more. In the following, we explain a solutions for this problem corresponding to the configuration shown in Fig. 6.

The proposed solution corresponding to the configuration in Fig. 6 will add the channel coding redundancy to the sequence of points at the output of the shaping block (without applying shaping to the symbols corresponding to the channel coding redundancy). Note that the proposed configuration based on using a lattice code is compatible with this configuration. In this case, the receiver side will first recover the sequence of points and pass the result to the block inverting the addressing operation to recover the shaping bits. This means that the operation of the corresponding channel decoder for decoding within cosets will not be affected by the presence of shaping. This configuration will obviously result in some loss in the achievable shaping gain, but as the redundancy is usually quite small, the degradation will be negligible. For example, if the shaping gain is equal to 1dB, having a code of rate $63/64$ (corresponding to lattice D_{64}) within cosets will reduce the shaping gain to 0.98dB. Note that this configuration for encoding/decoding within cosets has no

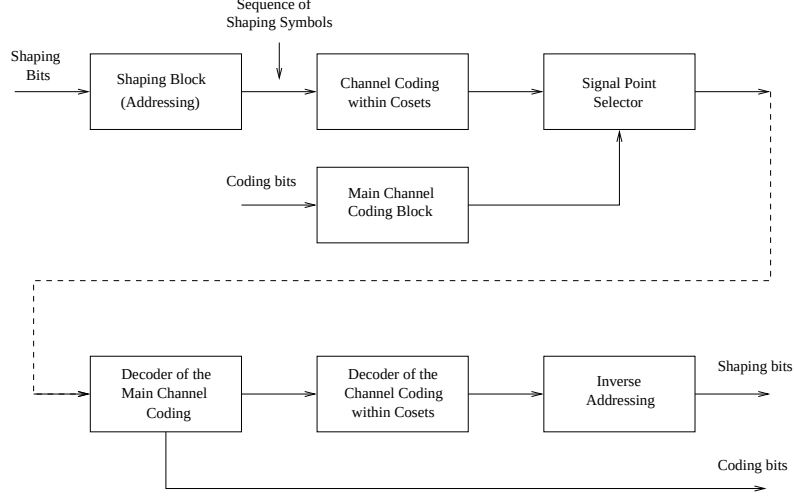


Fig. 6. Block diagram for coding/shaping.

impact on the operation of the main channel coding component and different coding/shaping parts will function in harmony.

A second method for exploiting the dependency between the binary channels (appropriate for methods based on a direct mapping of coded bits to the constellation points [24]–[35]) is to incorporate the constellation as a component in the iterative loop. We have implemented such an algorithm and the experiment was not very successful (resulted in a marginal improvement in BER performance).

III. A SIMPLE AND EFFECTIVE SHAPING ALGORITHM USING A LOOKUP TABLE

In the following, we present an efficient addressing scheme for shaping in a 32-dimensional space. First, the one-D sub-constellations are partitioned into 4 shaping partitions as shown in Fig. 4. This results in 16 shaping partitions (using Cartesian product) in the 2-D sub-constellations. We assume that addressing is achieved in a hierarchy of stages which each stage involves the Cartesian product of two (similar) lower dimensional sub-spaces. This results in 5 stages to reach to dimension 32 starting from 2-D sub-spaces. This hierarchy is shown in Fig. 7. This number of one-D shaping partitions is enough to realize most of the achievable shaping gain in 32-D (refer to Fig. 3). Note that in all our discussions, addressing refers to selecting a 2-D shaping partition within each 2-D sub-constellation. Obviously, if each such 2-D shaping partition contains 2^r signal points, then one will need another r bits per 2-D to select the final point with the selected 2-D shaping partition

(in each sub-constellation). We do not discuss the effect of these extra bits because these are simply extracted from the input bit stream.

Assuming 16 shaping partitions in 2-D, we obtain $16 \times 16 = 256$ elements in the Cartesian product of the 2-D sub-constellations in 4-D sub-spaces. In general, we refer to the elements formed in the Cartesian product of the lower dimensional shaping partitions as *Clusters*. The 4-D clusters are ordered according to their average energy, and 16 subsequent clusters are merged into a 4-D shaping partition. This results in 16 shaping partitions of equal cardinality in the 4-D sub-spaces. Again, the Cartesian product of the 4-D shaping partitions (resulting in 256 clusters in each 8-D sub-space) are ordered according to their average energy and subsequent clusters are merged into 8-D shaping partitions. To reduce the addressing complexity, the merging at this stage is achieved using non-equal number of clusters in subsequent 8-D shaping partitions. In specific, the number of clusters (in the order of increasing average energy) merged into subsequent 8-D shaping partitions are equal to $\{16, 16, 32, 32, 32, 32, 32, 64\}$, respectively. This results in 8 shaping partitions in 8-D sub-spaces. Following that, we obtain 64 clusters in the 16-D sub-spaces (which are not merged) and 4096 clusters in the 32-D space (which are not merged). Note that the cardinalities of the 16-D and 32-D clusters are not equal (these cardinalities are all an integer power of two). The final constellation is selected from the 32-D clusters of least energy such that the overall bit rate is 54. Note that without shaping, the bit rate would be $16 \times 4 = 64$. This means that the shaping redundancy is equal to 7 bits/32-D, resulting in,

$$\text{CER} = \frac{16}{2^{57/16}} \simeq 1.35$$

The addressing at the highest level of the hierarchy is achieved using a Huffman tree as shown in Fig. 8. The first step is to decide if the selected 32-D cluster belongs to Set I, Set II or Set III. This is achieved by assuming that the 194 clusters are labeled by the binary number obtained by assigning zero to the left branch and one to the right branch at each node of the tree. Then, the label of each final node (corresponding to a cluster) is obtained by concatenating the binary labels of its branches (where the most significant bits correspond to values closer to the top of the tree). This will result in the labels of the final nodes to be ordered increasingly from left to right. Note that the label of the final nodes are composed of 7 bits (for Set I), 8 bits (for Set II), and 9 bits (for Set III). In this case, to select a 32-D cluster, we extract 7 bits from the input stream and compare its numerical value with the threshold T_1 which is the label of the last cluster in Set I. If the label

is smaller or equal to T_1 , then we are within Set I, and have another $57 - 7 = 50$ bits to proceed with the addressing within the selected cluster. Otherwise, we extract one more bit from the input stream, resulting in an 8 bit label. We compare the numerical value of the resulting 8-bit label with the threshold value T_2 which is the label of the last node in Set II. Again, if the label turns out to be smaller or equal, we proceed with the addressing within Set II using the remaining bits (in this case, 49 bits are left). If the label turns out to be larger than T_2 , then we extract one more it from the input stream and use the resulting 9 bits to select an element within Set III (in this case, we are left with 48 bits to select an element within Set III).

After selecting one of the 194 clusters in 32-D (as explained above), to proceed with the addressing within the selected cluster, we assume that there exists a look-up table with 194 memory locations each of $4 \times 3 = 12$ bits, where the 3-bit address sections point to the 8-D shaping partitions building a given 32-D cluster (note that each 32-D cluster is simply the Cartesian product of four 8-D shaping partitions). This results in $1294 \times 12 = 2328$ bits of ROM.

To proceed with the addressing within 8-D shaping partitions, we note that the selected 8-D shaping partition is composed of 16, 32, or 64 of 8-D clusters (corresponding to 4, 5, 6 bits, respectively). In this case, we extract another 4, 3 or 2 bits from the input (for each 8-D sub-space) to bring the total number of bits per 8-D to 8 bits. We use these 8-bit addresses to select an 8-D cluster in each 8-D sub-space. To do this, we assume that there exists a look-up table with 256 memory locations each of $4 \times 2 = 8$ bits, where each 4-bit section of these 8-bit addresses points to the 4-D shaping partitions construing a given 8-D cluster. These bits are used to select one 4-D shaping partition in each 4-D sub-space.

To proceed with the addressing within 4-D shaping partitions, we assume that there exists a look up table of 256 memory locations each of $4 \times 2 = 8$ bits, where each 4-bit section of these 8-bit addresses points to the 2-D shaping partitions construing a given 4-D cluster. These bits are used to select one 2-D shaping partition in each 2-D sub-space. As mentioned before, there will be another group of input bits (depending on the number of signal points within each 2-D shaping partition) which will be used to select the final point within each 2-D sub-constellation. The total memory requirement for the proposed addressing scheme is equal to,

$$M_{total} = 2328 + 2048 + 2048 = 6424 \text{ bits} \simeq 0.73 \text{ K}$$

This means the overall complexity of this addressing scheme is 0.73 K bytes (8 bits) of ROM and a

negligible number of comparisons. This scheme has been simulated and the shaping gain achieved is about 1 dB.

Note that the clusters always happen in pair (of equal energy), say $A \times B$ and $B \times A$, unless the two constituent lower dimensional components are the same, say $A \times A$. This property can be used to reduce the size of the required memory by a factor of close to two at the price of a very small number of comparisons (comparing the labels to some anchor points corresponding to the label of the clusters with identical components).

Example: Assume that each 1-D sub-constellation is composed of 16 points (divided into 4 shaping partitions). The input is composed of a total of 57 shaping bits plus another 64 bits. We are only concerned with the operation of the selection of the shaping partitions using the 57 shaping bits. The selection of the final points within the shaping partitions is a trivial task. The same procedure will apply to any other scenario where the number of 1-D points is different from 16 while the number of 1-D shaping partitions is 4. Let us consider the following cases for the 57 shaping bits:

Case I: 0010010... The first 7 bits are 0010010 with the value of 18 that is less than $T_1 = 69$. In this case, the selection will be within Set I in Fig. 2. This value of 18 will point to a location in the corresponding look up table which contains 4 pointers each of 3 bits specifying the related 8-D clusters (there are 4 of these 8-D clusters). After knowing the 8-D clusters, we proceed with the addressing within each of those 8-D clusters by extracting an appropriate number of bits from the remaining input bits in a sequential manner and using those bits to select their related elements within the 4-D and subsequently 2-D sub-spaces by referring to the related look-up tables used for the addressing within 4-D and 2-D sub-spaces.

Case II: 1010010 0 100100... The first 7 bits are 1010010 which is equal to 82 and is greater than $T_1 = 69$. To proceed, we extract one more bit from the input which is equal to 0. Then, the first 8 bits, namely, 10100100 with the value of 164 is compared to $T_2 = 247$. As the label is smaller than T_2 , then we are within the Set II. In this case, we are left with 49 shaping bits to proceed with the selection of the 8-D clusters.

Case III: 1111011 1 100100... The first 7 bits are 1111011 with the value 123 which is larger than 69 and the first 8 bits, namely 11110111 are of value 247, so we are within Set II and we use the first 8 bits for the addressing in Set II.

Case IV: 1111110 1 100100... The value of the first 7 bits is larger than $T_1=69$ and the value of

the first 8 bits is larger than $T_2 = 247$, so we are with Set III and we extract the first 9 bits for the addressing with Set III

For the sake of comparison, in [63][64] (adopted for TIU-T telephone line modem standard) an addressing scheme based on direct shell mapping [13] is given where the overall complexity in a 16-D space is about 42 multiply-adds/2-D together with a few divisions and a memory of about 0.5 kilo-bytes and achieves nearly optimum tradeoff with a CER up to 1.5. In [18], an example for $N = 64$ is given which needs 1440 multiply-adds (assuming a 16 bit processor) and a memory of 1.5 kilo-bytes to achieve a tradeoff point with $\gamma_s = 1.15$ dB, CER = 1.5. For $N = 64$, an algorithm is given in [22] which can achieve $\gamma_s = 1.18$ dB, CER = 1.5 using 160 additions, 4 multiplications and 2.6 kilo-bytes of memory. As an alternative, reference [22] can achieve the same tradeoff point using about one multiply-add per 2-D and a memory of 1.0 kilo-byte. As another example, reference [22] can achieve $\gamma_s = 1.0$ dB for $N = 64$, CER = 1.25 using 100 additions, 2 multiplications and 1.1 kilo-bytes of memory. It is observed that the algorithm presented in the current article is substantially less complex as compared to the above methods.

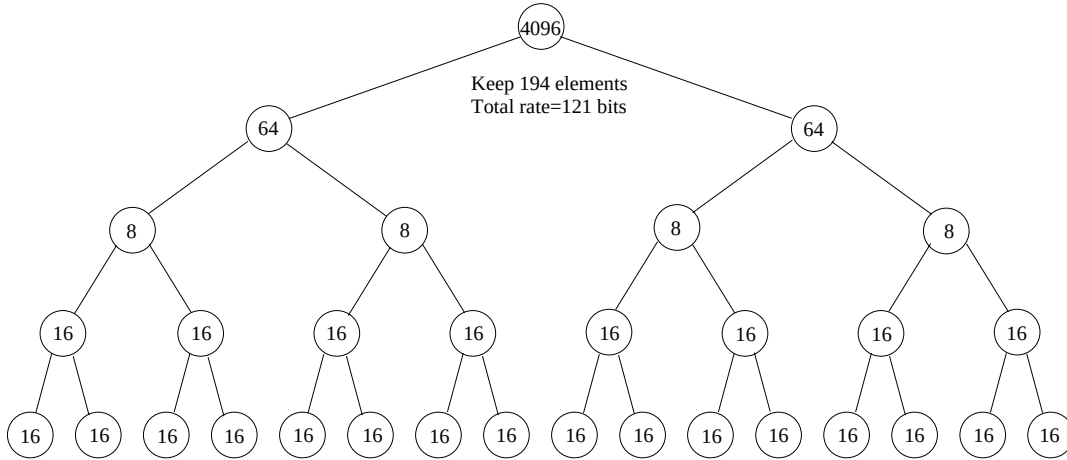


Fig. 7. Tree structure of the hierarchy used for addressing 32 dimensions.

IV. NUMERICAL RESULTS

Figure 9 shows the BER performance of the proposed scheme for a spectral efficiency of 4,6 bits/sec/Hz where the coding redundancy is equal to 2 bits per two dimensions. Block length is 2048 symbols and the code has 4 memory elements, the interleaver (of size 2048 symbols) is S-

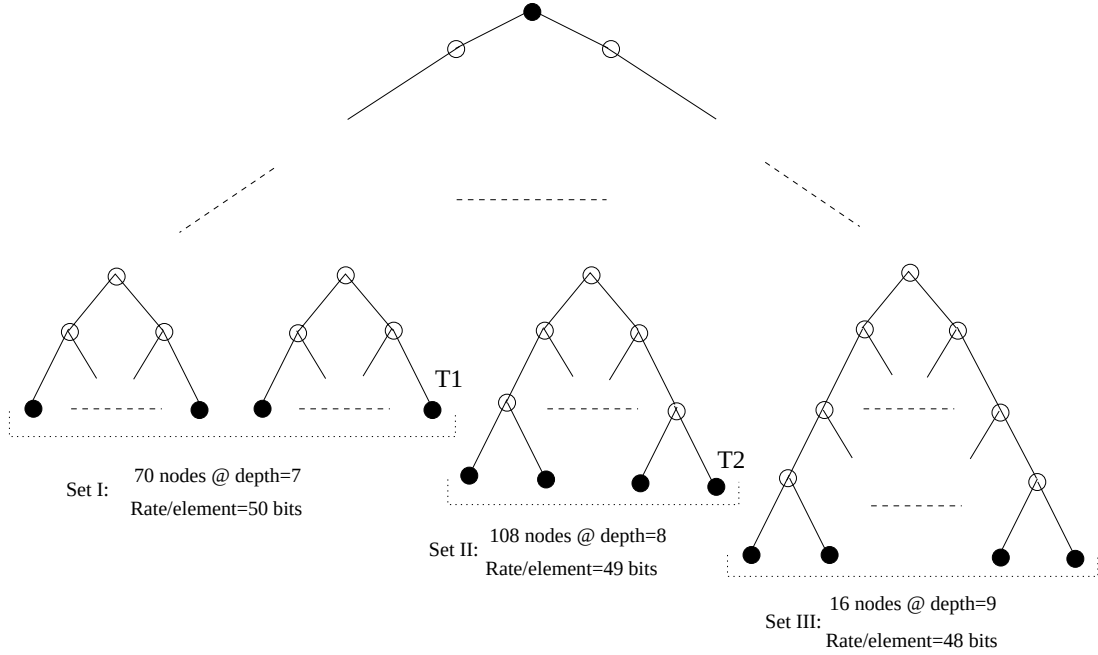


Fig. 8. Structure of the Huffman tree used for addressing.

random with $S = 19$ [65] (all the interleavers used in different numerical simulations are selected to maximize the value of S). In these curves, the system is simulated based on a partitioning of a one-D constellation of 8,16 points, respectively, into 4 cosets. An appropriate puncturing is applied to the parity streams of the symbol-based Turbo-code acting on the cosets to adjust the rate at $1/2$. The puncturing patterns for the two parity sequences is of the form $1010 \dots$ and $0101 \dots$ where 1 means transmission and zero means puncturing. The simulation is conducted without shaping, then a gain of 0.95 dB is included to reflect the effect of shaping.

Reference [53] is also based on using a set of uncoded bits to avoid the growth of the complexity with the number of constellation points. This reference is one of the few works reported in the literature which studies constellations with higher values of spectral efficiency appropriate for ADSL applications. Note that reference [53] takes advantage of bit interleaving (unlike the current work) in conjunction with a special labeling method. Reference [53] is motivated by the observation that “the very large constellation sizes used in ADSL systems make both conventional bit-level Turbo-coded QAM and symbol-level Turbo TCM very complicated to decode in the receiver side”. The best result of [53] (without shaping) for a scheme with a spectral efficiency of 4 bits/sec/Hz (using a 64 QAM) requires an E_b/N_0 of about 8.4 dB (refer to Fig. 9 for comparison). In [53], another

example for a spectral efficiency of 12 bits/sec/Hz is given which is about 2.7 dB away from the Shannon limit, while the distance o Shannon limit in our case is mostly about 1 dB. Note that the basic structure of the encoder in [53] (use of uncoded bits and bit interleaving) is similar to [52].

As another example of schemes with high spectral efficiency, reference [28] reports results (without shaping) for a scheme with a spectral efficiency of 6 bits/sec/Hz (using a 256 QAM) which requires an E_b/N_0 of about 14 dB [28] (refer to Fig. 9 for comparison). Note that reference [28] is based on a direct mapping of bits to the constellation points (same category of work as reported in [24]-[35]).

To study the effect of coding redundancy, Fig. 11 shows the BER performance of the proposed scheme for a spectral efficiency 5,7 bits/sec/Hz when the coding redundancy is equal to 1 bits per two dimensions. All system parameters are the same as in Fig. 9 except that the puncturing patterns for the two parity sequences is of the form 10000010000010 . . . and 00010000010 . . . resulting in a code rate of 3/4 for the symbol-based Turbo-code acting on cosets.

Figure 11 provides comparison with the method of [24]-[35] for 2 bits/sec/Hz (without shaping). The block length is 4096 bits and the code rate is 1/2 with 16 QAM. The symbol-based interleaver of size 2048 is S-random with $S = 19$. The other method requires two interleavers of size 4096 and 8192 (the second interleaver is used to shuffle the bits prior to mapping to the constellation points). These two interleavers are also S-random with $S = 23$ and $S = 27$, respectively. Note that the difference in energy between these curves is very close to the difference between the curves labeled as “4-point constellation with equal probability of points” and the one labeled as “binary independent channels” with Gray labeling (for a spectral efficiency of 1 bit/dimension) in Fig. 2.

REFERENCES

- [1] J. L. Massey, “Coding and modulation in digital communication,” *Proc. 1974 International Zürich Seminar on Digital Communication*, Zürich, Switzerland, pp. E2(1)-E2(4), Mar. 1974.
- [2] G. Ungerboeck and I. Csajka, “On improving data-link performance by increasing channel alphabet and introducing sequence coding,” *Proc. IEEE Int Symp. Information Theory (ISIT)* (Ronneby, Sweden), June 1976.
- [3] H. Imai, and S. Hirakawa, “A new multilevel coding method using error-correcting codes,” *IEEE Trans. Inform. Theory*, vol.IT-23, pp. 371-377, May 1977
- [4] U. Wachsmann, R. F. H. Fischer and J. B. Huber, “Multilevel codes: theoretical concepts and practical design rules,” *IEEE Transactions on Information Theory*, vol.45, no. 5, July 1999, pp. 1361-1391
- [5] A. R. Calderbank, “Multilevel codes and multistage decoding,” *IEEE Trans Commun.*, vol. 37, pp. 222-229, Mar. 1989
- [6] G. J. Pottie and D. P. Taylor, “Multilevel Codes based on Partitioning,” *IEEE Transactions on Information Theory*, vol.IT-35, pp.87-98, Jan. 1989

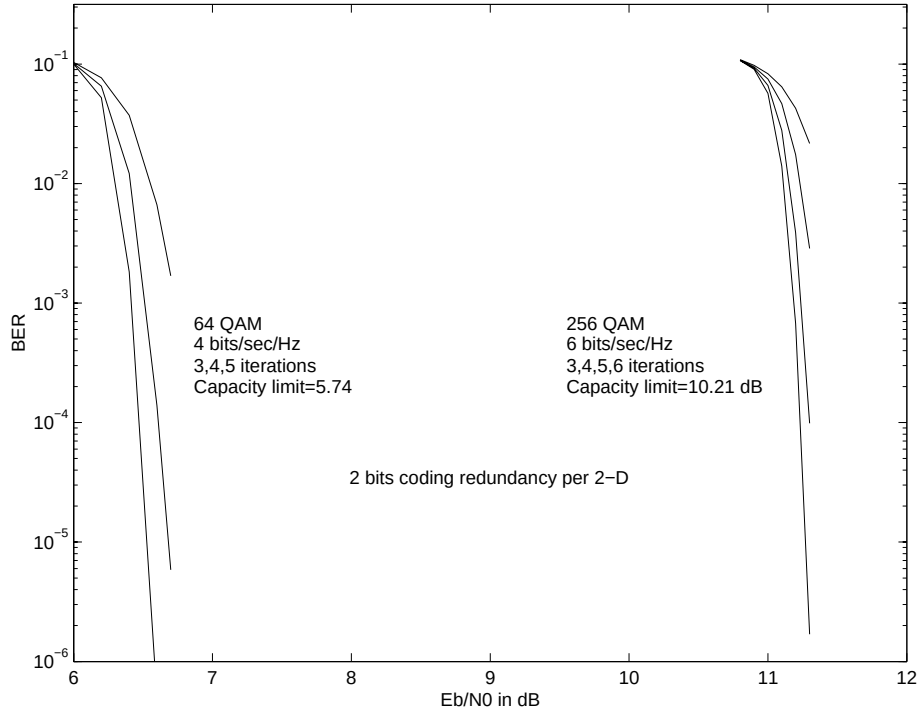


Fig. 9. BER performance of the proposed scheme for a spectral efficiency of 4,6 bits/sec/Hz with two bits of coding redundancy per two-D. Shannon limit corresponds to the capacity of a continuous AWGN channel.

- [7] G. D. Forney, M. D. Trott and C. Sae-Young, "Sphere-bound-achieving coset codes and multilevel coset codes," *IEEE Transactions on Information Theory*, vol.46 no.3, May 2000, pp. 820-850
- [8] 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.
- [9] L. F. Wei, "Trellis coded modulation with multidimensional constellations," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 483–501, July 1987.
- [10] 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.
- [11] G. D. Forney, Jr., "Multidimensional constellations—Part II: Voronoi constellations," *IEEE J. Select. Areas Commun.*, vol. SAC-7, pp. 941–958, August 1989.
- [12] 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.
- [13] G. R. Lang and F. M. Longstaff, "A leech lattice modem," *IEEE J. Select. Areas Commun.*, vol. SAC-7, pp. 968–973, August 1989
- [14] P. Fortier, A. Ruiz and J. M. Cioffi, "Multidimensional Signal Sets through the Shell Construction for Parallel Channels", *IEEE Transactions on Communications*, vol. 40, no.3, pp. 500-512, March 1992
- [15] A. K. Khandani and P. Kabal, "Shaping multi-dimensional signal spaces-Part I: optimum shaping, shell mapping ," *IEEE Trans. Inform. Theory*, vol.IT-39, pp.1799-1808, Nov. 1993.
- [16] A. K. Khandani and P. Kabal, "Shaping multi-dimensional signal spaces-Part II: shell-addressed constellations," *IEEE Trans. Inform. Theory*, vol.IT-39, pp.1809-1819, Nov. 1993.

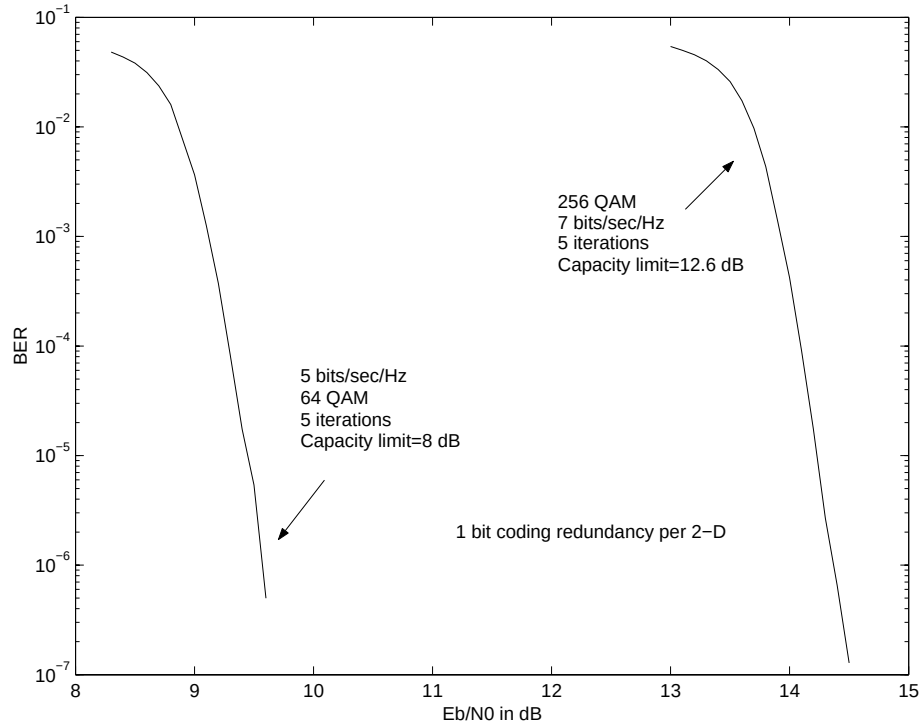


Fig. 10. BER performance of the proposed scheme for a spectral efficiency of 5,7 bits/sec/Hz with one bit of coding redundancy per 2-D. Shannon limit corresponds to the capacity of a continuous AWGN channel.

- [17] F. R. Kschischang and S. Pasupathy, "Optimal shaping properties of the truncated polydisc," *IEEE Trans. Inform. Theory*, vol. IT-40, pp. 892–903, May 1994
- [18] R. Laroia, N. Farvardin and S. A. Tretter "On optimal shaping of multi-dimensional constellations," *IEEE Trans. Inform. Theory*, vol. IT-40, pp. 1044–1056, July 1994
- [19] G. D. Forney, "Trellis shaping," *IEEE Trans. Inform. Theory*, vol. IT-38, pp. 281–300, March 1992.
- [20] F. R. Kschischang and S. Pasupathy, "Optimal nonuniform signaling for gaussian channels," *IEEE Trans. Inform. Theory*, vol. IT-39, pp. 892–903, May 1993
- [21] A. K. Khandani and P. Kabal, "Shaping of multi-dimensional signal constellations using a lookup table," *IEEE Trans. Inform. Theory*, vol. IT-40, pp. 2058–2062, Nov. 94.
- [22] A. K. Khandani and P. Kabal, "An efficient block-based addressing scheme for the nearly optimum shaping of multi-dimensional signal spaces," *IEEE Trans. Inform. Theory*, vol. IT-41, pp. 2026–2031, Nov. 1995.
- [23] C. Berrou, A. Glavieux, and P. Thitimajshima, *Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-codes*, *IEEE Int. Conf. on Comm. 1993 (ICC'93)* (Geneva, Switzerland), pp. 1064–1070, May 1993.
- [24] S. Le Goff, A. Glavieux, and C. Berrou, "Turbo-codes and high spectral efficiency modulation," in *Proc. IEEE Int. Conf. Commun.*, 1994, pp. 645–649
- [25] L. Sylla and P. Fortier, "A study of Turbo codes for multilevel modulations in gaussian and mobile channels", *Congres Canadian de genie electrique et de genie informatique*, Waterloo, May 24–28, 1997
- [26] M.B. Shoemake, C. Heegard and E. Rossin, "Turbo codes for high order constellations," *Proceedings of Information Theory Workshop, 1998*, Page(s): 6–7
- [27] J. P. Lauer and J. M Cioffi, "Turbo coding for discrete multitone transmission systems," *IEEE Global Telecommunications*

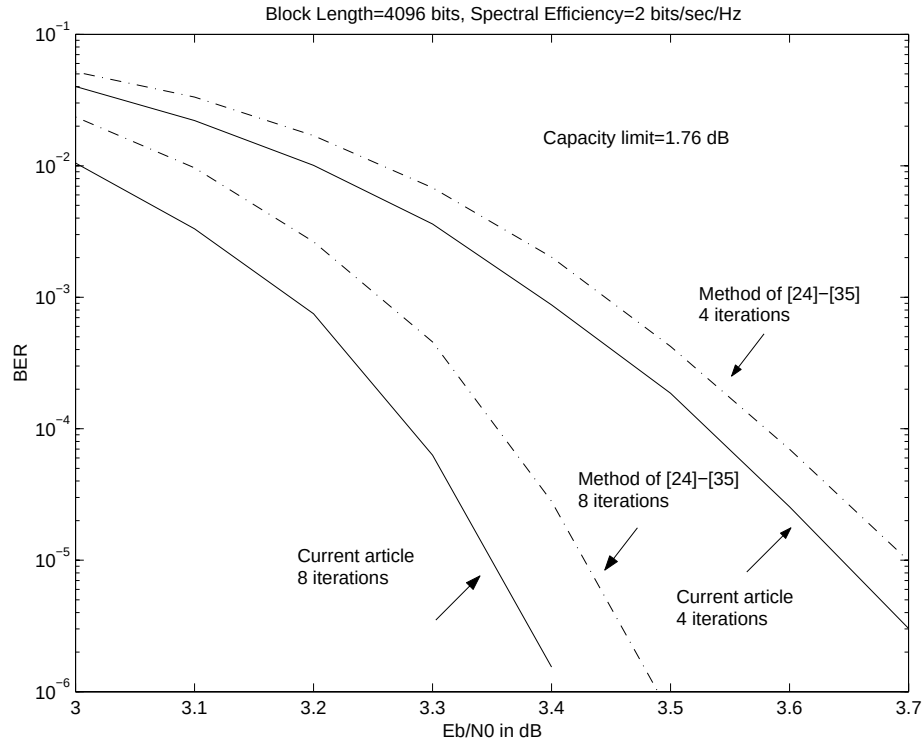


Fig. 11. BER performance comparison for a spectral efficiency of 2 bits/sec/Hz (without shaping). Shannon limit corresponds to the capacity of a continuous AWGN channel.

Conference 1998 (GLOBECOM 1998), Page(s): 3256 -3260 vol.6

- [28] H. R. Sadjadpour, "Application of Turbo codes for discrete multi-tone modulation schemes," *IEEE International Conference on Communications*, 1999 (ICC99) Page(s): 1022-1027 vol.2
- [29] L. Weixiao Liu and S. G. Wilson, "Rotationally-invariant concatenated(turbo) TCM codes," *Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, 1999, Volume: 1, Page(s): 32-36 vol.1
- [30] T. M. Duman and M. Salehi, "Performance bounds for turbo-coded modulation systems," *IEEE Transactions on Communications*, vol.47 no.4, April 1999, pp. 511-521
- [31] J. Yuan and W. Feng and B. Vucetic, "Turbo trellis coded modulation for fading channels," *Proceedings of Vehicular Technology Conference (VTC 2000)-Spring*, Tokyo, Page(s): 2059 -2063 vol.3
- [32] Y. Jinhong, B. Vucetic and F. Wen, "Turbo-coded M-QAM for fading channels," *Electronics Letters*, Volume: 36 Issue: 18 , 31 Aug. 2000 Page(s): 1562-1563
- [33] P.D. Papadimitriou and P. Varshney, "Turbo coded modulation for high-throughput TDMA systems," *Vehicular Technology Conference*, 2001. VTC 2001 Spring, 2001, Page(s): 1390-1394 vol.2
- [34] S. Le Goff and F. O. Al-Ayyan, "Design of bit-interleaved turbo-coded modulations," *Electronics Letters*, Volume: 37 Issue: 16 , 2 Aug. 2001 Page(s): 1030-1031
- [35] S. Y. Le Goff and F. O. Al-Ayyan "On the design of bit-interleaved turbo-coded modulation," *Proceedings of Information Theory Workshop*, 2001, Page(s): 73-75
- [36] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Bandwidth efficient parallel concatenated coding schemes," *Electron Lett*, vol.31, no.24, pp. 2067-2069, 1995
- [37] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Parallel concatenated trellis coded modulation," *Proc. IEEE Int. Conf.*

- Communications*, (Dallas, TX), pp. 974-978, June 1996
- [38] C. Yuwei and V. K. Wei, "Q-ary turbo codes with QAM modulations," *IEEE International Conference on Universal Personal Communications*, 1996, Page(s): 814 -817 vol.2
 - [39] P. Robertson and T. Worz, "Coded modulation scheme employing turbo codes," *Electronics Letters*, Volume: 31, Issue: 18, Aug. 1995, Page(s): 1546 -1547
 - [40] P. Robertson and T. Worz, "A novel bandwidth efficient coding scheme employing turbo codes," *Proc. IEEE Int. Conf. Communications*, vol.2, Dallas, TX, June 1996, pp.962-967
 - [41] P. Robertson and T. Worz, "Extensions of turbo trellis coded modulation to high bandwidth efficiencies," *IEEE International Conference on Communications, 1997 (ICC'97)*, Montreal, Page(s): 1251-1255 vol.3
 - [42] P. Robertson and T. Worz, "Bandwidth-efficient turbo trellis-coded modulation using punctured component codes," *IEEE J. Select Areas Commun*, vol. 16, pp.206-218, Feb. 1998.
 - [43] H. Ogiwara and M. Yano "Improvement of Turbo Trellis-Coded Modulation System," *IEICE Trans. Fundamentals*, vol. E81-A, No.10, pp. 2040-2046, Oct. 1998
 - [44] J. P. Lauer and J. M. Cioffi, "A turbo trellis coded discrete multitone transmission system," *IEEE Global Telecommunications Conference 1999 (GLOBECOM '99)*, Page(s): 2581 -2585 vol.5
 - [45] C. Zhaohui; K. R. Subramanian and Liren Zhang "DMT scheme with multidimensional turbo trellis code," *Electronics Letters*, Volume: 36 Issue: 4 , 17 Feb. 2000 Page(s): 334 -335
 - [46] Mark Bingeman and Amir K. Khandani, *Design, Performance Analysis and Decoding of Combined Turbo-code and Modulation*, Proceedings of 19th Biennial Symposium on Communications, Queens University, Kingston, Ontario, Canada, pp. 36—40, June 1998.
 - [47] M. Bingeman and A. K. Khandani, "Symbol-based Turbo-codes," *IEEE Communication Letters*, Vol. 3, No. 10, pp. 285—287, Oct. 1999.
 - [48] C. Fragouli and R.D. Wesel, "Bit vs. symbol interleaving for parallel concatenated trellis coded modulation," *IEEE Global Telecommunications Conference 2001 (GLOBECOM'01)*, Page(s): 931 -935, vol 2
 - [49] C. Fragouli and R. D. Wesel, "Turbo-encoder design for symbol-interleaved parallel concatenated trellis-coded modulation," *IEEE Transactions on Communications*, vol. 49, no. 3, March 2001, pp. 425-435
 - [50] C. Fragouli, R. D. Wesel, D. Sommer and G.P. Fettweis, "Turbo codes with non-uniform constellations," *IEEE International Conference on Communications (ICC 2001)*, pp.70—73 vol.1
 - [51] B.E. Wahlen and C.Y. Mai, "Turbo coding applied to pragmatic trellis-coded modulation," *IEEE Communications Letters*, Volume: 4 Issue: 2 , Feb. 2000 Page(s): 65 -67
 - [52] L. Zhang and A. Yongacoglu, "Turbo coding in ADSL DMT systems," *IEEE International Conference on Communications, 2001 (ICC 2001)*, Page(s): 151-155 vol.1
 - [53] B. Li, A. Deczky and A. Ginesi, "A new turbo coded QAM scheme with very low decoding complexity for ADSL system," *Global Telecommunications Conference, 2001 (GLOBECOM 01)*, Page(s): 349 -353 vol. 1
 - [54] A. Ruscitto and E.M. Biglieri, "Joint source and channel coding using turbo codes over rings," *IEEE Transactions on Communications*, Volume: 46 Issue: 8, Aug. 1998, Page(s): 981 -984
 - [55] Z. Cai, K. R. Subramanian and L. Zhang "Source-controlled channel decoding using non-binary turbo codes," *Electronics Letters*, Volume: 37 Issue: 1 , 4 Jan. 2001 Page(s): 39 -40
 - [56] P. Robertson "An Overview of Bandwidth Efficient Turbo Coding Schemes," *Proc. The International Symposium on Turbo Codes and Related Topics*, pages 103-110, September 1997.
 - [57] U. Wachsmann and J. Huber, "Power and bandwidth efficient digital communication using Turbo codes in multilevel codes," *Europ Tran., Telecommun (ETT)*, vol. 6, pp. 557-567, Sept.–Oct. 1995

- [58] R. Pyndiah, A. Picart and A. Glavieux, "Performance of block turbo coded 16-QAM and 64-QAM modulations," *Proc. IEEE Global Telecommunications Conference, 1995 (GLOBECOM'95)*, vol. 2, pp. 1039-1043
- [59] A. S. Barbulescu and S. S. Pietrobon, "Interleaver design for turbo codes," *Electron Let*, vol 30, pp. 2107-2 108, Dec 1994
- [60] P. Sauvé and F. R. Kschischang, *Decoding Turbocodes by Multibit Probability Propagation, Proceedings of 19th Biennial Symposium on Communications*, Queens University, Kingston, Ontario, Canada, pp. 89—93, June 1998,
- [61] C. Berrou and M. Jezequel, "Non binary convolutional codes for turbo coding", *Elect. Letters*, vol.35, no.1, pp. 39-40, Jan. 1999
- [62] C. Berrou, M. Jezequel, C. Douillard and S. Kerouedan, "The Advantages of Non-Binary Turbo Codes," *Proc. 2001 IEEE Information Theory Workshop*, Cairns, Australia 2-7 Sept. 2001
- [63] Motorola Information Systems Group, "Signal mapping and shaping for V. fast." Contribution D196, CCITT Study Group XVII, June 1992.
- [64] G. D. Forney, Jr., "Advances in modem technology since V.32/V .32bis," *Int. Conf. on Data Transmission—Advances in Modem and ISDN Technology and Applications*, (London, U.K.), pp. 1–6, I.E.E., Sept 1992.
- [65] S. Dolinar and D. Divsalar, "Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations", *The Telecommunications and Data Acquisition Progress Report 42-122, April-June 1995*, Jet Propulsion Laboratory, Pasadena, California, pp. 56-65, August 15 1995.