Approximate Enumerative Sphere Shaping

Yunus Can Gültekin, Frans M.J. Willems
Eindhoven University of Technology
Eindhoven, The Netherlands
Email: {y.c.g.gultekin, f.m.j.willems}@tue.nl

W.J. van Houtum
Catena Radio Design
Eindhoven, The Netherlands
Email: wim.van.houtum@catena.tech

Semih Şerbetli
NXP Semiconductors
Eindhoven, The Netherlands
Email: semih.serbetli@nxp.com

***Abstract*—Enumerative sphere shaping of N-dimensional constellations is discussed. It is proven that a finite-precision number representation is suitable for use in two enumerative indexing algorithms: Enumerative sphere shaping and Divide & Conquer (D&C) shaping. This representation decreases the storage complexities of these methods significantly. D&C is the basis of the well-known shell mapping algorithm and thus our approximations also apply there.**

I. Introduction

Consider transmission of N-dimensional vectors x^N over an additive white Gaussian noise channel for which we know that the capacity-achieving input distribution is Gaussian. The loss of mutual information of channel input X and output Y arising from using a uniform input distribution is called shaping gap and approaches 0.255 bits per dimension, asymptotically in block length N. This gap can also be seen as the increase in the average energy resulting from using a cubical signal structure instead of a spherical one, which is 1.53 dB asymptotically.

There is extensive literature on shaping which can be defined as optimization of the channel input with the purpose of closing the shaping gap, see [1, Chapter 4]. Since most of the signal shaping methods require selection of points from an N-dimensional space, the fundamental bottleneck is the addressing complexity. Attempting to solve the addressing problem, enumerative techniques are used in source coding [2], and also in shaping, see [3] and [4]. As another example, shell mapping [5] is a well-known enumerative indexing method which is applied in the V.34 modem standard for $N = 16$ [6]. Here approximate enumerative algorithms will be introduced to decrease computational and storage complexities.

In this paper, shaping will be treated as a procedure to select the amplitudes of the channel inputs where the signs can be selected in any way that leads to a symmetrical input distribution, i.e., the signs must be uniform. In particular, channel coding can be used for this purpose and completes the system which can be called shaped coded modulation of which a block diagram is given in [7, Fig. 3]. See [8] for a similar construction which does not require systematic encoders and employs convolutional codes of the IEEE 802.11 system in a special manner.

There are several shaping philosophies by which the shaper outputting the amplitudes is implemented. As an example, constant composition distribution matching (CCDM) [9] is employed as the shaping technique in [7]. CCDM represents amplitude sequences by intervals inspired by arithmetic data compression techniques. These sequences realize a fixed composition, i.e., the frequencies of the amplitudes are constant for all sequences leading to a constant composition. As proved in [9], this method attains the capacity asymptotically in block length N. However as analyzed in [8], using all the points inside an N-sphere is more efficient for short block lengths. As an example, motivated by the number of subcarriers in the IEEE 802.11 system, $N = 96$ was used in [8] and it was shown for rate $R = 1.75$ bits per dimension that there is 0.64 dB gain of N-sphere shaping over CCDM. Therefore our focus here is on sphere shaping.

This paper is organized as follows. In Sec. II, a base-2 number representation is proposed for enumerative sphere shaping and it is proved that the reproducibility of the indexes is maintained. In Sec. III, the same is accomplished for D&C algorithm which is the basis of shell mapping. In Sec. IV, the rate loss induced by the approximate method is analyzed.

II. Approximate Enumerative Sphere Shaping

Let $\mathcal{S}$ be a set of bounded-energy amplitude-sequences $a^N = (a_1, a_2, \cdots, a_N) \in \mathcal{A}^N$ of length N and let $S = |\mathcal{S}|$ denote the cardinality of the set. Here $\mathcal{A} = \{+1, +3, \cdots, 2|\mathcal{A}| - 1\}$, and the energy constraint is $e(a^N) = \sum_{i=1}^{N} a_i^2 \leq E_{\max}$ for all $a^N \in \mathcal{S}$. Our problem is to find a one-to-one mapping from the index set $\mathcal{I} = [0, S)$ to $\mathcal{S}$. Assuming lexicographical ordering, the enumerative approach is to derive (in an efficient way) from a sequence its index, i.e., the number of sequences in $\mathcal{S}$, which are "smaller" in the lexicographical ordering. Moreover an efficient reverse operation should be specified. Since S is quite large in general, using a lookup table is infeasible. Enumerative sphere shaping [3] provides efficient recursive shaping and deshaping algorithms to solve this problem.

A. Bounded Energy Trellis

To find amplitude sequences a^N with $e(a^N) \leq E_{\max}$, a bounded-energy trellis is constructed. As an example, consider Fig. 1 where $\mathcal{A} = \{1, 3, 5\}$, $N = 4$ and $E_{\max} = 28$. This is equivalent to say that the 4-dimensional amplitude-sequence space is bounded by a sphere of radius $\sqrt{28}$.

In this trellis, nodes in column n, represent the accumulated energy $e = \sum_{i=1}^{n} a_n^2$, which is indicated in black. Therefore we will use (n, e) to pinpoint a specific node where n is the corresponding column (dimension) and e the energy. The nodes in the rightmost column, column N, are called final

978-1-5386-4780-6/18/$31.00©2018 IEEE

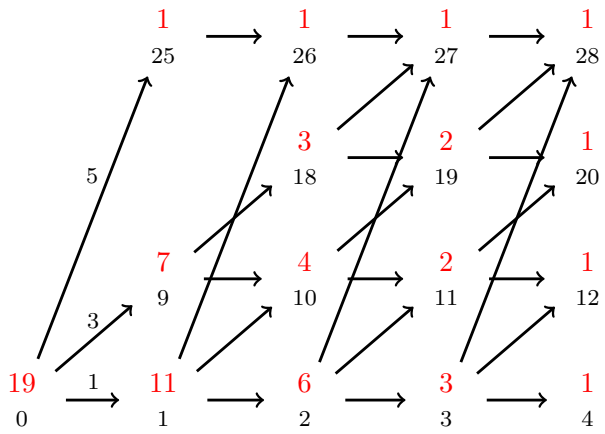

Fig. 1. Enumerative trellis for $N = 4$, $A = 3$, and $E_{\max} = 28$.

states. Branches between states are labeled with amplitudes $a \in \mathcal{A}$. Each path starting in the zero-energy node (in column 0) and ending in a final node represents an energy-bounded N-sequence.

The number written in red in a node (n, e) is the number of possible ways to reach a final node starting from that node, and is denoted by T_n^e. Thus T_0^0 indicates the total number of sequences represented in the trellis. The information rate R corresponding to the trellis can be computed as $R = \log_2\left(T_0^0\right)/N$ in bits per dimension. For our example $R = 1.06$.

The numbers T_n^e in the trellis for $n = 0, 1, \cdots, N-1$ and $e \le E_{\max}$ can be computed in a recursive manner as

$$T_n^e \triangleq \sum_{a \in \mathcal{A}: e+a^2 \le E_{\max}} T_{n+1}^{e+a^2}, \tag{1}$$

where the initialization is

$$T_N^e = \begin{cases} 1: & e \le E_{\max} \\ 0: & \text{otherwise} \end{cases}. \tag{2}$$

Note that we only consider states with energy levels that are possible. Possible states in column n have energy level n plus a multiple of 8, not exceeding $E_{\max}$. The maximum energy can be written as $E_{\max} = N + 8 \times (L-1)$ where L is the number of possible energy values in each column of the trellis. Therefore the effective number of states is $L(N+1)$. Since the numbers in the trellis can be up to $\lceil NR \rceil$-bit long, the required memory to store the trellis is upper bounded by $L(N+1)\lceil NR \rceil$ bits which is similar to what was found for Algorithm 1 in [4].

B. Shaping (Encoding)

The shaping operation maps $\lfloor NR \rfloor$ bits (i.e., the base-2 representation of index I) to amplitude-sequences of length N which are ordered lexicographically assuming that $1 < 3 < \cdots < 2|\mathcal{A}| - 1$. An efficient way of implementing this, see [3], is formulated in Algorithm 1.

The shaping algorithm requires at most $(|\mathcal{A}| - 1)$ subtractions per dimension which upper bounds the number of required computations by $(|\mathcal{A}| - 1)\lceil NR \rceil$ bit operations per dimension.

Algorithm 1 Enumerative Shaping

Given that $0 \le I < T_0^0$, initialize the algorithm by setting the *local index* $I_1 = I$. Then for $n = 1, 2, \cdots, N$:

1) Take a_n be such that

$$\sum_{a < a_n} T_n^{e\left(a^{n-1}, a\right)} \le I_n < \sum_{a \le a_n} T_n^{e\left(a^{n-1}, a\right)}, \tag{3}$$

2) and (for $n < N$)

$$I_{n+1} = I_n - \sum_{a < a_n} T_n^{e\left(a^{n-1}, a\right)}. \tag{4}$$

Finally output a^N.

C. Deshaping (Decoding)

Deshaping finds the lexicographical index J of an amplitude-sequence a^N. An efficient way of implementing this, see [3], is formulated in Algorithm 2.

Algorithm 2 Enumerative Deshaping

1) Initialize the algorithm by setting the *local index* $J_{N+1} = 0$.
2) For $n = N, N-1, \cdots, 1$, update the local index as

$$J_n = \sum_{a < a_n} T_n^{e\left(a^{n-1}, a\right)} + J_{n+1}. \tag{5}$$

3) Finally output $J = J_1$.

Similar to shaping, at most $(|\mathcal{A}|-1)$ additions per dimension are necessary for deshaping which upper bounds the number of required computations by $(|\mathcal{A}|-1)\lceil NR \rceil$ bit operations per dimension.

Note that both for shaping and for deshaping, the trellis T_n^e must be computed and stored for $n = 0, 1, \cdots, N$ and relevant values of e.

D. Bounded Precision Trellis

To decrease the amount of memory needed to store the trellis, we use the base-2 representation $T_n^e = m \cdot 2^p$. This is a finite-precision notation where m and p are called the mantissa and power respectively, and are represented using n_m and n_p bits as in [10].

Based on this representation, the enumerative trellis is now computed as

$$T_n^e \triangleq \left\lfloor \sum_{a \in \mathcal{A}: e+a^2 \le E_{\max}} T_{n+1}^{e+a^2} \right\rfloor_{n_m}, \tag{6}$$

where $\lfloor x \rfloor_{n_m}$ means rounding x down to n_m bits, so that x can be represented with a mantissa of n_m bits. The result is again stored in the form (m, p). In this way, the required memory is decreased from $L(N+1)\lceil NR \rceil$ bits to $L(N+1)(n_m + n_p)$ which is now linear in N.

Obviously, the numbers in the trellis will become smaller by the rounding. It is not obvious however that the index that is

input to the shaping algorithm will lead to an energy-bounded sequence from which the deshaping algorithm can reconstruct the original index, using the rounded trellis.

We will show next that reproducibility based on Algorithms 1 and 2 is guaranteed if the trellis condition (1) is relaxed to

$$T_n^e \le \sum_{a\in\mathcal{A}: e+a^2\le E_{\max}} T_{n+1}^{e+a^2}. \tag{7}$$

The proof will consist of two steps, a lemma, and a theorem.

Lemma 1. *If* $0 \le I_n < T_{n-1}^{e(a^{n-1})}$*, then Algorithm 1 guarantees that* $0 \le I_{n+1} < T_n^{e(a^n)}$*. This implies that if* $0 \le I < T_0^0$*, then all* I_n *for* $n = 1, 2, \cdots, N$ *satisfy* $0 \le I_n < T_{n-1}^{e(a^{n-1})}$.

Proof. Note that[1]

$$0 \le I_n < T_{n-1}^{e(a^{n-1})} \overset{(7)}{\le} \sum_{a\in\mathcal{A}} T_n^{e(a^{n-1},a_k)},$$

therefore Algoritm 1 will always find an a_n that satisfies (3). From (3) and (4) we then find that

$$\begin{aligned} I_{n+1} &= I_n - \sum_{a<a_n} T_n^{e\left(a^{n-1},a\right)} \\ &< \sum_{a\le a_n} T_n^{e\left(a^{n-1},a\right)} - \sum_{a<a_n} T_n^{e\left(a^{n-1},a\right)} = T_n^{e\left(x^{n-1},a_n\right)}. \end{aligned}$$

$$\begin{aligned} I_{n+1} &= I_n - \sum_{a<a_n} T_n^{e\left(a^{n-1},a\right)}, \\ &\ge \sum_{a<a_n} T_n^{e\left(a^{n-1},a\right)} - \sum_{a<a_n} T_n^{e\left(a^{n-1},a\right)} = 0. \end{aligned}$$

□

Theorem 1. *Algorithms 1 and 2 guarantee that a local index* $0 \le I_n < T_{n-1}^{e\left(a^{n-1}\right)}$ *in state* $\left(n-1, e\left(a^{n-1}\right)\right)$ *for* $n = 1, 2, \cdots, N$ *results in a sequence* $a_n, a_{n+1}, \cdots, a_N$ *that has local index* $J_n = I_n$*. Note that we are interested in* $n = 1$ *in the end.*

Proof. The proof is by induction.

- First consider the state $\left(N-1, e\left(a^{N-1}\right)\right)$ at depth $N-1$. The states at depth N to which this state is connected are final states. Observe that there are at least $T_{N-1}^{e(a^{N-1})}$ such final states since (7) holds. Note that since $I_N < T_{N-1}^{e(a^{N-1})}$ there exist I_N final states below the state that corresponds to a_N that was chosen during shaping. These final states will lead to local index $J_N = I_N$ during deshaping, see (5).
- Next focus on the state $\left(n-1, e\left(a^{n-1}\right)\right)$ at depth $n-1$, for $n < N$. During shaping, based on local index I_n, an a_n was chosen and this resulted in next local index I_{n+1}, see (4). The induction hypothesis now tells that in state $(n, e(a^n))$, the corresponding sequence $a_{n+1}, a_{n+2}, \cdots, a_N$ will lead to an local index $J_{n+1} = I_{n+1}$. Therefore the sequence $a_n, (a_{n+1}, a_{n+2}, \cdots, a_N)$, by (5), and then by (4), leads to

$$\begin{aligned} J_n &= \sum_{a<a_n} T_n^{e\left(a^{n-1},a\right)} + J_{n+1} \\ &= \sum_{a<a_n} T_n^{e\left(a^{n-1},a\right)} + I_{n+1} = I_n. \end{aligned}$$

□

[1]Notation is simplified by replacing $a \in \mathcal{A} : e + a^2 \le E_{\max}$ into $a \in \mathcal{A}$.

We have shown now that reproducibility is guaranteed as long as (7) holds. Note that summations and subtractions in (3), (4) and (5) are assumed to be exact.

III. Approximate Divide and Conquer

Optimum shaping of multidimensional constellations is discussed also in [4] where N-sequences are ordered based on their energy. Sequences of the same energy, i.e., on the same N-dimensional shell, can then be addressed in two different enumerative manners.

The first manner is similar to what we discussed before, but now we constrain ourselves on fixed energy sequences. We can use the trellis structure in Fig. 1, but only allow for the final state corresponding to the shell energy E. Only for this state $T_N^E = 1$, all other final states have $T_N^e = 0$, $e \ne E$.

In the second manner, sequences (having the same energy) are sorted with respect to the index of their first half, and the ones having identical first halves with respect to the index of their second half. This principle is applied recursively. An example of this ordering can be found in [11]. Assuming that N is a power of 2, this second algorithm [4], which is called Divide & Conquer (D&C) uses a table with only $\log_2(N) + 1$ columns. In this way, the storage complexity is decreased relative to the first manner, but at the expense of doing multiplications. The columns contain the numbers M_n^e for $n \in \{1, 2, 4, \cdots, N\}$ and the relevant values of e, that are used in the enumeration processes. These are the number of n-vectors having energy e and can be computed recursively

$$M_n^e = \sum_{k\le e} M_{n/2}^k M_{n/2}^{e-k}, \tag{8}$$

where M_1^e can be determined from $\mathcal{A}$. Note that for $n = N$ the relevant e-value is E, and this leads to M_N^E.

A. D&C Shaping

The shaping algorithm successively divides an n-dimensional problem into two $n/2$-dimensional problems. At the end, the 2-dimensional mapping can be realized easily. An efficient way of implementing this, see [4] and [11], is formulated in Algorithm 3. We start from the index $I_N(a^N)$, the index pointing to the desired sequence in the selected shell.

B. D&C Deshaping

The deshaping algorithm implements the inverse mapping by successively concatenating $n/2$-tuples to get n-tuples and computing their offsets. At $n = N$, the index $J_N(a^N)$ is computed using two depth $N/2$ offsets. An efficient way of implementing this, see [4] and [11], is formulated in Algorithm 4.

Algorithm 3 D&C Shaping

For $n = N, N/2, \cdots, 4$:

1) The energy e_1 of the first half a_1^n of a^n (and consequently the energy e_2 of the second half a_2^n) is determined by taking

$$\sum_{k<e_1} M_{n/2}^k M_{n/2}^{e(a^n)-k} \le I_n(a^n) < \sum_{k\le e_1} M_{n/2}^k M_{n/2}^{e(a^n)-k}, \tag{9}$$

and then setting $e_2 = e(a^n) - e_1$.

2) First, residual offset D_s follows from

$$D_s = I_n\left(a^n\right) - \sum_{k<e_1} M_{n/2}^k M_{n/2}^{e(a^n)-k}, \tag{10}$$

and then the local offsets $I_{n/2}\left(a_1^n\right)$ and $I_{n/2}\left(a_2^n\right)$

$$I_{n/2}\left(a_1^n\right) = \left\lfloor \frac{D_s}{M_{n/2}^{e_2}} \right\rfloor, \tag{11a}$$

$$I_{n/2}\left(a_2^n\right) = D_s - I_{n/2}\left(a_1^n\right) M_{n/2}^{e_2} \tag{11b}$$

are computed.

Finally, mapping from depth-2 offsets to symbols is straightforward as can be seen in [11].

Algorithm 4 D&C Deshaping

Note that $J_1\left(a_1^2\right) = J_1\left(a_2^2\right) = 0$. Now, for $n = 2, 4, \cdots, N$

$$D_d = J_{n/2}\left(a_1^n\right) M_{n/2}^{e(a_2^n)} + J_{n/2}\left(a_2^n\right), \tag{12a}$$

$$J_n\left(a^n\right) = \sum_{k<e\left(a_1^n\right)} M_{n/2}^k M_{n/2}^{e(a^n)-k} + D_d. \tag{12b}$$

C. Bounded Precision Implementation

We will prove now that reproducibility based on Algorithms 3 and 4 is guaranteed if the trellis condition (8) is relaxed to

$$M_n^e = \left\lfloor \sum_{k\le e} M_{n/2}^k M_{n/2}^{e-k} \right\rfloor_{n_m} \le \sum_{k\le e} M_{n/2}^k M_{n/2}^{e-k}, \tag{13}$$

which will be the case when the D&C trellis is computed with bounded precision. In this way, the required memory will decrease from $L(\log_2(N)+1)\lceil NR\rceil$ to $L(\log_2(N)+1)(n_m + n_p)$ bits.

The proof will again consist of two steps, a lemma, and a theorem.

Lemma 2. *If $0 \le I_n(a^n) < M_n^{e(a^n)}$, then the D&C Shaping Algorithm guarantees that $0 \le I_{n/2}(a_i^n) < M_{n/2}^{e(a_i^n)}$ for $i = 1, 2$. This implies that if $0 \le I_N(a^N) < M_N^{e(a^N)}$, then all $I_n(a^n)$ for $n = N/2, N/4, \cdots, 2$ satisfy $0 \le I_n(a^n) < M_n^{e(a^n)}$. Note that there are two $I_{N/2}$, four $I_{N/4}$, etc.*

Proof. Note that

$$0 \le I_n\left(a^n\right) < M_n^{e(a^n)} \overset{(13)}{\le} \sum_{k\le e(a^n)} M_{n/2}^k M_{n/2}^{e(a^n)-k},$$

therefore Algorithm 3 will always find an e_1 that satisfies (9). From (9) and (10) we then find that

$$0 \le D < M_{n/2}^{e_1} M_{n/2}^{e(a^n)-e_1}. \tag{14}$$

From (11a) and (11b), using $e_2 = e(a^n) - e_1$, we find that

$$0 \le I_{n/2}\left(a_1^n\right) < M_{n/2}^{e_1},$$
$$0 \le I_{n/2}\left(a_2^n\right) < M_{n/2}^{e_2}.$$

□

Theorem 2. *The D&C Shaping and Deshaping algorithms guarantee that a local offset $0 \le I_n(a^n) < M_n^{e(a^n)}$ for $n = 2, 4, \cdots, N$, results in a sequence a^n that has a local offset $J_n(a^n) = I_n(a^n)$. We are interested in $n = N$ in the end.*

Proof. The proof is by induction.

- First consider depth 2. Observe that there are at least $M_2^{e(a^2)}$ possible symbol pairs since (13) holds. Note that since $I_2(a^2) < M_2^{e(a^2)}$ there exists $I_2(a^2)$ pairs below the a^2 which was chosen during shaping. These pairs will lead to a local offset $J_2(a^2) = I_2(a^2)$ during deshaping, see (12b).
- Next focus on depth n for $n > 2$. During shaping, based on local offset $I_n(a^n)$, an e_1 was chosen and this resulted in next local offsets $I_{n/2}(a_1^n)$ and $I_{n/2}(a_2^n)$, see (11). The induction hypothesis now tells that in depth $n/2$, the corresponding sequences a_1^n and a_2^n will lead to local offsets $J_{n/2}(a_1^n) = I_{n/2}(a_1^n)$ and $J_{n/2}(a_2^n) = I_{n/2}(a_2^n)$. Therefore the sequence $a^n = (a_1^n, a_2^n)$ by (12), and then by (10) and (11) leads to

$$\begin{aligned} J_n\left(a^n\right) &= \sum_{k<e_1} M_{n/2}^k M_{n/2}^{e(a^n)-k} + D_d, \\ &= \sum_{k<e_1} M_{n/2}^k M_{n/2}^{e(a^n)-k} + \mathcal{D}_s = I_n\left(a^n\right). \end{aligned}$$

□

We have shown now that reproducibility within a shell is guaranteed as long as (13) holds. Along the same lines we can show that (13) eventually implies that $J = I$. Here I is the index from which first the shell is chosen and then local index $I_N(a^N)$ that enters the D&C procedure. Now J is the corresponding output index.

IV. Rate Loss

Numbers in a bounded precision trellis are smaller than their full precision counterparts which translates to a decrease in rate. To quantify this rate loss, let $\tilde{\alpha}$ be defined as $\tilde{\alpha} = \lfloor \alpha \rfloor_{n_m}$. In the worst case, i.e., the case in which the largest possible relative error due to rounding occurs, $\tilde{\alpha}$ can be lower bounded as $\tilde{\alpha} \ge (1-\delta)\alpha$ where $\delta = 2^{-(n_m-1)}$. Using this, the loss in rate of an enumerative or D&C trellis can be upper bounded.

Proposition 1. *In bounded precision enumerative trellises,* $\tilde{T}_n^e \geq T_n^e(1-\delta)^{(N-n)}$ *for* $n = 0, 1, \cdots, N$ *where* $\tilde{T}_n^e$ *denotes the trellis computed with bounded precision.*

Proof. The proof is by induction.

- First consider $n = N$. Since $T_N^e = 1$ for $e \leq E_{\max}$, see (2), $\tilde{T}_N^e = T_N^e$.
- Next focus on depth n for $n < N$. The induction hypothesis tells that $\tilde{T}_{n+1}^e \geq T_{n+1}^e(1-\delta)^{(N-(n+1))}$. From (6) we obtain

$$\begin{aligned}\tilde{T}_n^e = \left\lfloor \sum \tilde{T}_{n+1}^{e+a^2} \right\rfloor_{n_m} &\geq (1-\delta) \sum \tilde{T}_{n+1}^{e+a^2} \\ &\geq (1-\delta) \sum (1-\delta)^{(N-n-1)} T_{n+1}^{e+a^2} \\ &= (1-\delta)^{(N-n)} T_n^e.\end{aligned}$$

□

Then the rate loss of an enumerative trellis can be upper bounded by $\log_2(T_0^0/\tilde{T}_0^0)/N \leq -\log_2(1-\delta)$ bits per dimension.

Proposition 2. *In bounded precision D&C trellises,* $\tilde{M}_n^e \geq M_n^e(1-\delta)^{(n-1)}$ *for* $n = 1, 2, 4, \cdots, N$ *where* $\tilde{M}_n^e$ *denotes the trellis computed with bounded precision.*

Proof. The proof is by induction.

- First consider $n = 1$. By definition, $M_1^e = 1$ for $e \in \{1, 9, \cdots, (2|\mathcal{A}|-1)^2\}$. Thus $\tilde{M}_1^e = M_1^e$.
- Next focus on depth n for $n \in \{2, 4, \cdots, N\}$. The induction hypothesis tells that $\tilde{M}_{n/2}^e \geq M_{n/2}^e(1-\delta)^{(n/2-1)}$. Consider from (13) that

$$\begin{aligned}\tilde{M}_n^e &\geq (1-\delta) \sum_{k \leq e} \tilde{M}_{n/2}^k \tilde{M}_{n/2}^{e-k} \\ &\geq (1-\delta) \sum_{k \leq e} (1-\delta)^{(n-2)} M_{n/2}^k M_{n/2}^{e-k} \\ &= (1-\delta)^{(n-1)} M_n^e.\end{aligned}$$

□

Then the rate loss of a D&C trellis can be upper bounded by $\log_2(\sum M_N^e / \sum \tilde{M}_N^e)/N \leq -\log_2(1-\delta)$ bits per dimension.

Rate losses induced by bounded precision and the upper bound are shown in Fig. 2 as a function of n_m. Here $N = 64$, $L = 59$ and $\mathcal{A} = \{1, 3, 5, 7\}$ for which the full precision rate is $R = 1.509$. It can be deduced from Fig. 2 that a small number of bits (e.g., $n_m \approx 8$) can be used to store mantissas while the rate loss is kept smaller than 10^{-2} bits per symbol. Since D&C computes the index of a sequence by concatenating multiple shorter sequences successively, rounding error accumulation during recursion starts later than that of enumerative shaping. Therefore the rate loss of D&C is smaller for the same n_m.

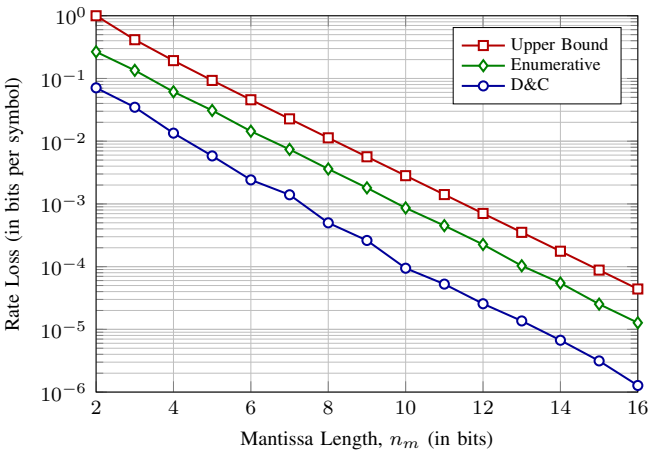

Fig. 2. Rate loss due to bounded precision ($N = 64$, $L = 59$, $|\mathcal{A}| = 4$).

V. Conclusion

In this paper, enumerative sphere shaping of multidimensional constellations is investigated. A finite-precision number representation is proposed and proven to be suitable for use in enumerative sphere shaping and in Divide & Conquer. It is shown that the storage complexities of these algorithms are decreased considerably. Note that Divide & Conquer is the basis of shell mapping algorithm to which our approximations are directly applicable.

As a final remark, we mention that the approximate realization of enumerative sphere shaping enables sliding window shaping and deshaping implementations. This reduces the computational complexities significantly.

Acknowledgment

The authors would like to acknowledge NXP-Research Eindhoven for their support to accomplish this work.

References

[1] R. F. H. Fischer, *Precoding and signal shaping for digital transmission*. J. Wiley-Interscience, 2002.
[2] T. Cover, "Enumerative source encoding," *IEEE Trans. on Inf. Theory*, vol. 19, no. 1, pp. 73–77, January 1973.
[3] F. Willems and J. Wuijts, "A pragmatic approach to shaped coded modulation," in *IEEE 1st Symp. on Commun. and Veh. Technol. in the Benelux*, 1993.
[4] R. Laroia, N. Farvardin, and S. A. Tretter, "On optimal shaping of multidimensional constellations," *IEEE Trans. on Inf. Theory*, vol. 40, no. 4, pp. 1044–1056, Jul 1994.
[5] G. R. Lang and F. M. Longstaff, "A leech lattice modem," *IEEE J. on Sel. Areas in Commun.*, vol. 7, no. 6, 1989.
[6] G. D. Forney, L. Brown, M. V. Eyuboglu, and J. L. Moran, "The v.34 high speed modem standard," *IEEE Commun. Mag.*, vol. 34, no. 12, pp. 28–33, Dec 1996.
[7] G. Böcherer, F. Steiner, and P. Schulte, "Bandwidth Efficient and Rate-Matched Low-Density Parity-Check Coded Modulation," *IEEE Trans. on Commun.*, vol. 63, no. 12, 2015.
[8] Y. C. Gültekin, W. van Houtum, S. Şerbetli, and F. M. J. Willems, "Constellation Shaping for IEEE 802.11," in *2017 IEEE 28th Int. Symp. on Personal, Indoor, and Mobile Radio Commun. (PIMRC)*, Oct 2017.
[9] P. Schulte and G. Böcherer, "Constant Composition Distribution Matching," *IEEE Trans. on Inf. Theory*, vol. 62, no. 1, 2016.
[10] K. A. S. Immink, "A practical method for approaching the channel capacity of constrained channels," *IEEE Trans. on Inf. Theory*, vol. 43, no. 5, pp. 1389–1399, Sep 1997.
[11] S. Tretter, *Constellation Shaping, Nonlinear Precoding, and Trellis Coding for Voiceband Telephone Channel Modems: with Emphasis on ITU-T Recommendation*. Springer US, 2012, no. v. 34.