Constellation Shaping for IEEE 802.11

Yunus Can Gültekin*, W.J. van Houtum†, Semih Şerbetli‡ and Frans M.J Willems*
*Eindhoven University of Technology, Eindhoven, The Netherlands
Email: {y.c.g.gultekin, f.m.j.willems}@tue.nl
†Catena Radio Design, Eindhoven, The Netherlands
Email: wvhoutum@catena.nl
‡NXP Semiconductors, Eindhoven, The Netherlands
Email: semih.serbetli@nxp.com

***Abstract*—A constellation shaping scheme is proposed. The motivation is to decrease the required transmit power for a specified spectral efficiency. Instead of imposing a non-uniform distribution on the constellation or using non-uniformly spaced symbols, a sphere constraint is employed on the n-dimensional signal space. An efficient algorithm called *enumerative amplitude shaping* is given to find and index all signal points in the sphere. A comparison with a prominent probabilistic shaping algorithm is provided. The enumerative approach achieves the target rate more efficiently for small block lengths. To introduce error correction, the convolutional encoder used in IEEE Std 802.11 is combined with the shaper in a novel way. Instead of utilizing a larger constellation in combination with shaping, a higher code rate is used by puncturing. Gains up to 1.61 dB are observed for the rates 3 to 6 bits/2-D with 64- and 256-QAM schemes in AWGN channels. The contribution of puncturing in these gains is discussed.**

I. Introduction

Constellation shaping is widely investigated in the literature ever since Shannon proved that the capacity of additive white Gaussian noise (AWGN) channels can only be achieved by using a continuous Gaussian input distribution [1]. Both probabilistic approaches (e.g., see [2]) where the constellation points are employed with non-uniform probabilities, and geometric methods (e.g., see [3, Sec. 4.5]) in which they are positioned with non-uniform spacing are considered.

Recently, an implementation of probabilistic shaping is introduced in [4] to impose desired distribution on the constellation using arithmetic coding. Employing this idea with a low-density parity-check (LDPC) encoder (ENC) which operates within 2.5 dB of capacity for various code rates, it is shown that the system can operate within 1.1 dB of capacity in [5].

However, their constant composition distribution matching (ccdm [4]) idea is designed to find and address signal vectors having the same composition. Therefore, the multi-dimensional signal constellation consists of only a portion of the sequences on the surface of a hypersphere, without the ones inside and the ones on the surface but with a different composition. As a consequence of this approach and the sphere hardening argument, the performance of this concept improves with increasing block length n.

978-1-5386-3531-5/17/$31.00

On the other hand, we are interested in improving the performance of IEEE Std 802.11 [6] with an emphasis on the 802.11p amendment which is designed for vehicular networks. In these networks, the density of the users will be high which motivates us to work on shaping to decrease the transmit power levels (equivalently the interference), and the packet sizes are planned to be relatively small. Considering also the variability of packet sizes and the nature of the interleaving (e.g., block-wise over orthogonal frequency-division multiplexing -OFDM- symbols), shaping over OFDM symbols (where $n = 96$) is reasonable. Therefore we need a shaping recipe which is effective even for small n.

In this work, we will investigate the shaping question from a different perspective. Unlike the probabilistic approaches, we will not start with the desired probability distribution. We will constrain our signal space in a special way that it induces Gaussian distribution on the constituent constellation. The paper is organized as follows. In Sec. II we summarize the necessary background. We introduce enumerative amplitude shaping in Sec. III. In Sec. IV we show our novel way of combining shaping with the convolutional coding and interleaving defined in IEEE Std 802.11.

II. Background

Our objective is to use the signals having the smallest possible average energy for a prescribed spectral efficiency. First, we give the story behind this goal.

A. Channel Capacity

In [1], Shannon showed that the capacity of an AWGN channel is given by the expression

$$C(P) = \frac{1}{2}\log_2\left(1+\frac{P}{\sigma^2}\right) \text{ bits/dimension}, \qquad (1)$$

where σ^2 is the variance of the additive channel noise and P is the threshold that constrains the channel input variance. Shannon showed that this capacity can be achieved using a channel input having a zero-mean Gaussian density with variance P.

B. Pulse Amplitude and Quadrature Modulation

Although Gaussian inputs are required to achieve capacity, signaling methods which do not lead to Gaussian inputs, but to inputs that are discrete and uniform are used in practice. For example in pulse amplitude modulation (PAM), equally spaced values are used as channel inputs. For 2^m-PAM, the channel input sequences of length n can be regarded as points in an n-dimensional space and constitute an n-cube. Moreover, if these sequences are transmitted with equal probability, the inputs X_i for $i = 1, 2, \cdots, n$ are discrete and uniform.

For carrier transmission, typically rectangular quadrature amplitude modulation (QAM) is used which can be regarded as transmission of a pair of PAM symbols. The mutual information (MI) achieved with equiprobable 2^{2m}-QAM can be computed as a function of the signal-to-noise ratio (SNR) P/σ^2 as in [7], and is shown for $m = 1, 2, 3, 4$ in Fig. 1 along with the capacity in bits per two dimensions.

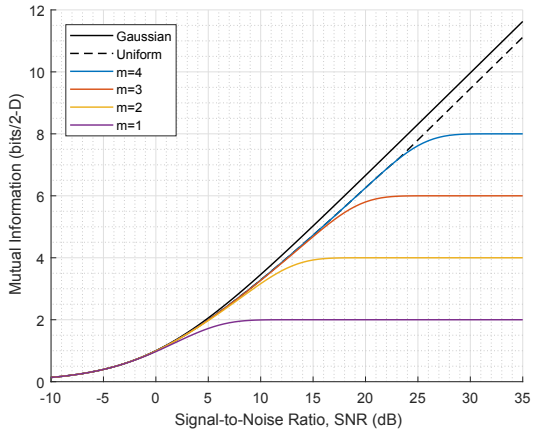

Fig. 1. Capacity of the AWGN channel with Gaussian, uniform, and equiprobable 2^{2m}-QAM inputs for $m = \{1, 2, 3, 4\}$.

Fig. 1 also shows the capacity of the AWGN channel with uniformly distributed input over a square with corner points $(\pm\sqrt{3P}, \pm\sqrt{3P})$ [3]. We observe that there is a gap to capacity which is the result of using uniform instead of Gaussian inputs. This gap can be shown to approach 1.53 dB (i.e., 0.255 bits per dimension) and to disappear in the high- and low-SNR regimes, respectively.

C. Hyperspheres and Gaussian Distribution

Consider an n-sphere of radius ρ. Let $\boldsymbol{X}$ designate the set of n-vectors $(X_1, X_2, \cdots, X_n)$ inside, and assume a continuous uniform probability distribution $p(\boldsymbol{x}) = 1/V_{\circ,n}$ where $V_{\circ,n}$ is the volume. Using the derivation in [8], the joint differential entropy of $\boldsymbol{X}$ is

$$h(\boldsymbol{X}) = -\int_{V_{\circ,n}} p(\boldsymbol{x}) \log\left(p(\boldsymbol{x})\right) d\boldsymbol{x} = \log\left(\frac{\rho^n \pi^{(n/2)}}{(n/2)!}\right), \tag{2}$$

for even n. Then using the independence of the dimensions and symmetry of the n-sphere, we can lower bound the average entropy along any dimension i as

$$h(X_i) \overset{(a)}{\geq} \frac{h(X_1, X_2, \cdots, X_n)}{n}, \tag{3}$$

$$\overset{(b)}{=} \frac{1}{n} \log\left(\frac{\rho^n \pi^{(n/2)}}{(n/2)!}\right), \tag{4}$$

$$\overset{(c)}{\geq} \frac{1}{2} \log\left(2\pi e \frac{\rho^2}{n}\right) - \frac{1}{2n} \log\left(\frac{e^2 n}{2}\right), \tag{5}$$

where (a) follows from independence bound on entropy, (b) is from (2), and (c) is an application of the Stirling's approximation. Finally, we use the maximum entropy property (for a specified variance) of the Gaussian distribution and use it to upper bound $h(X_i)$ as

$$h(X_i) = \frac{\sum_k h(X_k)}{n} \leq \frac{\sum_k \log\left(2\pi e E[X_k^2]\right)}{2n}, \tag{6}$$

$$\leq \frac{1}{2} \log\left(2\pi e \frac{\sum_k E[X_k^2]}{n}\right), \tag{7}$$

$$= \frac{1}{2} \log\left(2\pi e \frac{\rho^2}{n}\right). \tag{8}$$

Therefore, as n increases, the distribution of $\boldsymbol{X}$ along any axis converges to a zero-mean Gaussian distribution of variance ρ^2/n. Furthermore, it is shown in [3] that the ratio of the average energy of the n-sphere to that of the n-cube having the same volume is

$$G(n) = \frac{\pi(n+2)}{12} [(n/2)!]^{-2/n}, \text{ for even } n, \tag{9}$$

and called the shaping gain. Using Stirling's approximation, $G(\infty) \approx \pi e/6 = 1.53$ dB. Note that the number of points in the discrete situation is equivalent to the volume in the continuous situation assuming that the continuous approximation holds [9].

III. Enumerative Amplitude Shaping

We need to find and index all PAM amplitude sequences $\mathbf{x}$ (e.g., signal points) inside an n-sphere to have a Gaussian channel input. We will utilize an enumerative approach for which similar results can be found in [10] and [11]. Now, we explain how to realize this.

A. Enumerative Amplitude Trellis

To investigate our enumerative sequence finding and indexing approach, we will use an example as in [12]. Consider the one-sided alphabet $\mathcal{X}_+ = \{1, 3, 5\}$, the block length $n = 4$, and the maximum energy $E_{max} = 28$. The corresponding exemplary trellis is given in Fig. 2. Note that we will utilize channel coding later to add signs to the amplitudes in Sec. IV.

We construct the trellis by representing every sequence with a unique path, every energy level by a unique node, and every element of $\mathcal{X}_+$ by a link between nodes in the trellis as in [12]. Basic rules of the structure are the following:

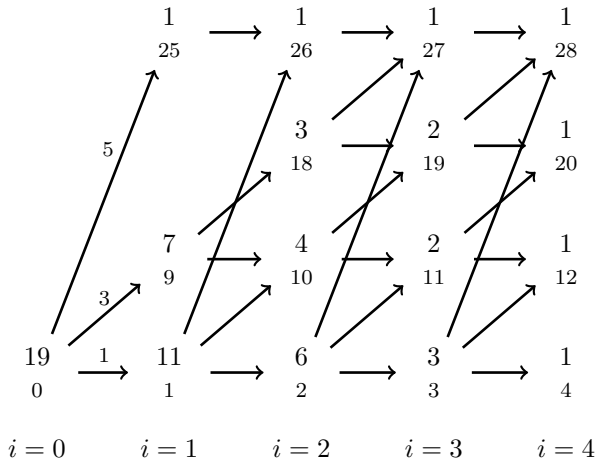

Fig. 2. Enumerative shaping trellis for $n = 4$ and $E_{max} = 28$.

- Each path starts in the zero-energy vertex and ends in one of the final nodes having energy not larger than E_{max}.
- Each edge represents a symbol x_i at time i from $\mathcal{X}_+$.
- The energy E_i of a node is denoted by a small number.
- The number of total paths advancing from a vertex of energy E_i at position i to one of the final states is denoted by a large number and notated by $M\left(i, E_i\right)$.

Note that numbers $M\left(i, E_i\right)$ can be calculated in a recursive manner as

$$M\left(n, e\right) = \begin{cases} 1: & 0 \le e \le E_{max} \\ 0: & \text{otherwise} . \end{cases} \tag{10}$$

$$M\left(t, e\right) \triangleq \sum_{a \in \mathcal{X}_+} M\left(t+1, e+a^2\right), \text{ for } t = n-1, \cdots, 0.$$

Consequently, the number of sequences $\mathbf{x}$ of length $n = 4$ having an energy $E(\mathbf{x}) \le E_{max}$ is given by $M(0,0) = 19$ for our example. Note also that possible energy values at final vertices are 4, 12, 20, or 28, and are always separated by 8.

Lastly, the rate of an enumerative amplitude shaping trellis is

$$R = \frac{\log_2\left(M(0,0) \times 2^n\right)}{n}, \text{ bits/dimension}, \tag{11}$$

where we also adjusted for the combinations with the signs.

B. Enumeration Algorithm

To address all paths in the trellis, we use a lexicographical approach. The elements of $\mathcal{X}_+$ have a lexicographical order as $1 < 3 < \cdots < (2|\mathcal{X}_+| - 1)$. Using this, we can order all sequences in a similar way as the dictionaries do, and give an index to them. Using the approach in [13], we can write an efficient algorithm to calculate the index of a sequence $\mathbf{x}$ as

$$i(\mathbf{x}) = \sum_{t=1}^{T} \sum_{w \in \mathcal{X}_+}^{w < x_t} M\left(t, w^2 + \sum_{s=1}^{t-1} x_s^2\right). \tag{12}$$

In words, to calculate the index of a sequence, we are counting the number of all paths which are lexicographically prior to that sequence. As an example, the sequence $\mathbf{x} = (3131)$ has a path traveling through the nodes of 0, 9, 10, 19, and 20 energy. Therefore, we calculate its index as $i(3131) = M(1,1) + M(3,11) = 11 + 2 = 13$. The inverse (index-to-sequence) mapping algorithm can also be implemented in a similar manner. The complexity of the algorithms is comparable to that of the algorithm 1 given in [10]. Fig. 3 gives an illustration of the shaper.

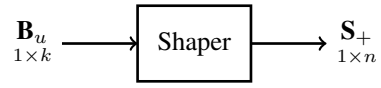

Fig. 3. Shaper maps k uniform bits to n amplitudes. The amplitude sequences are taken from an n-sphere. As $n \to \infty$, the distribution of the amplitudes converges to a one-sided Gaussian. Note that each sequence is selected with equal probability. Shaper addresses these sequences with the enumerative algorithm.

C. A Comparison: Probabilistic Amplitude Shaping

The ccdm [4] converts equally likely input bits into a sequence of amplitudes having the desired distribution [4, Fig. 1]. First, the distribution P_A on the one-sided PAM alphabet $\mathcal{A}_+$ which has the entropy equal to $H(P_A) = R - 1$ (where R is the target rate) is found. Note that the remaining 1 bit/dimension will be added when the signs are introduced. Then the corresponding composition is found by using the concept of n-type sequences, and either rounding nP_A to the nearest integer or using the quantization rule in [14, Algorithm 2]. Last, amplitude sequences having the same composition are produced by the matcher using arithmetic coding. Therefore, only some of the signal points on the surface of an n-sphere are addressed. As it is shown in [4, Fig. 2], the rate of ccdm R_{ccdm} approaches to the target rate R as $n \to \infty$. However for relatively small block lengths (e.g., $n < 200$) there is a substantial gap to R due to the fact that some of the possible points in and on the surface of the n-sphere are not utilized. This can also be justified using the sphere hardening argument.

To quantify this loss, consider an example where we target $R = 2.75$ bits/dimension using 8-PAM. The composition $\{37, 30, 19, 10\}$ achieves

$$R = \frac{1}{n}\left\lfloor \log_2\left(\frac{n!}{n_1!n_3!n_5!n_7!}\right)\right\rfloor = 1.75, \tag{13}$$

for $n = 96$. This scheme uses only some of the signal points on the surface and $E_{ccdm}^{av} = 13.25$. If we were to use all points on a surface, we would take $E_{max} = 1160$

to achieve 1.75 bits/dimension, and $E^{av}_{surface} = 12.08$. Finally, if we consider the enumerative scheme with $E_{max} = 1120$, all points in and on the surface leads to the same rate with $E^{av}_{sphere} = 11.43$. Therefore, the enumerative approach is 0.64 dB more energy efficient than the ccdm idea for $n = 96$ and $R = 2.75$. As a benchmark, observe that the ultimate shaping gain including the effect of the discretization [15]

$$G_\infty(R) \approx \frac{\pi e}{6}\left(1 - 2^{-2R}\right), \tag{14}$$

is 1.44 dB for this case. Therefore, for small n, enumerative approach performs more efficiently than probabilistic approach but requires increased complexity. There are several methods (which won't be discussed in this paper) inspired from source coding to make a trade-off between the memory and the number of calculations needed to compute the trellis. There may be additional advantages provided by constant composition based approaches in nonlinear fiber channels since there n is large.

IV. Shaping in a Wireless Standard

Combining enumerative shaping and coding is investigated in [12]. Now, we will apply enumerative shaping to IEEE 802.11.

A. Fundamentals of IEEE Std 802.11

The physical layer (PHY) of IEEE Std 802.11 is based on bit-interleaved coded modulation (BICM) with a little difference that is the interleaver Π is over the whole codeword in BICM [7] where it is over an OFDM symbol in [6] to keep latency small. The BICM block diagram is illustrated in [7, Fig. 1].

IEEE 802.11 employs a Gray-coded 2^{2m}-QAM constellation mapping in the form of a Cartesian product of two 2^m-PAM constellations. This mapping is illustrated for 16-QAM in [6, Fig. 17-10]. In such a scheme, every $2m$ bits ($\mathrm{b}_1\mathrm{b}_2\cdots\mathrm{b}_{2m}$) are mapped to a complex signal point in the 2-D plane. First m bits ($\mathrm{b}_1\mathrm{b}_2\cdots\mathrm{b}_m$) form the binary label of the real (in-phase) part, and the last m bits ($\mathrm{b}_{m+1}\mathrm{b}_{m+2}\cdots\mathrm{b}_{2m}$) form the binary label of the imaginary (quadrature) part. We express any signal X_i from the PAM signal set $\mathcal{X}$ as a multiplication of its amplitude A_i and its sign S_i as in [5, Sec. IV]. With this approach, the binary label of the sign $b(S)$ is the first of those m bits (b_1 and b_{m+1}), and the binary label of the amplitude $\boldsymbol{b}(A)$ is the rest. From this point onwards, we will call labels of the signs $b(S)$ sign bits, and labels of the amplitudes $\boldsymbol{b}(A)$ amplitude bits.

B. NASA Standard Code Structure

IEEE 802.11 employs the rate 1/2 non-systematic 64-state convolutional code (NASA standard code), followed by a bit-level block interleaver Π which is defined by two concatenated permutations prior to the mapper. The ENC structure can be seen in [6, Fig. 17-8] where $\mathrm{T_b}$ represents a delay element, output data A and B represent the outputs corresponding to the generator polynomials g_0=133_8 and g_1=171_8 respectively. We denote them by v_0[n] and v_1[n]. Note that for an input data stream b[n] ($\mathrm{b}_1\mathrm{b}_2\cdots$), the output bit stream is in the form of ($\mathrm{v}_0[1]\mathrm{v}_1[1]\mathrm{v}_0[2]\mathrm{v}_1[2]\cdots$).

C. Combining Shaping with a Non-Systematic Code

We will now explain our method to combine enumerative amplitude shaping with a non-systematic ENC.

1) Mother Encoder: The finite-state machine (FSM) of the NASA standard code has a special state transition mechanism. For a state s_i at time i, the output bit equations are $\mathrm{v}_0[i] = \mathrm{g}_0\left(\mathrm{b}_i, s_i\right)$ and $\mathrm{v}_1[i] = \mathrm{g}_1\left(\mathrm{b}_i, s_i\right)$ where b_i is the i^{th} input bit, g_0 and g_1 are the output functions. For a transition, the output bit pair $(\mathrm{v}_0[i]\mathrm{v}_1[i])$ is either in the set $\mathcal{V}_1 = \{(01), (10)\}$ or in $\mathcal{V}_2 = \{(00), (11)\}$. Consequently, if we have the freedom of selecting b_i, we can produce one of the output bits as we desire by switching between the elements of $\mathcal{V}_i$. In the long run, we can fix half of the output bits (one of two at a time) to the values we selected, by using the appropriate input bit stream considering the FSM. In other words, half of the output bits of the ENC above are similar to the don't-care (DC) terms in digital logic circuit minimization, and they give us degrees of freedom to fix the other half. An illustration of this idea is given in Fig. 4.

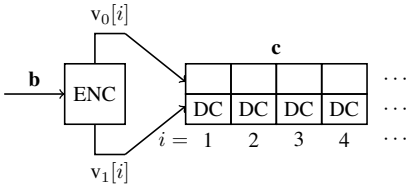

Fig. 4. The structure of the output codeword **c** of the mother ENC of rate $r = 1/2$. DCs represent the bits which we don't care about. This freedom enables us to determine one of the output bits at a time. We do that by altering the input bit at that time and choosing the correct element of $\mathcal{V}_i$ with respect to the output bit which will be fixed.

We call this method, where we follow the state transitions and select the correct input bit looking at the output, *input-select* with the functional diagram given in Fig. 5.

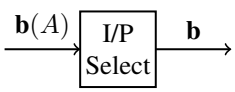

Fig. 5. When fed with $\mathbf{b}(A)$, the input-select block outputs the bit stream **b** such that the subsequent ENC will output $\mathbf{b}(A)$ at the specified positions of **c**. During the select procedure, FSM of the code and the fixed bit locations at the output is known to the input-select.

As an example, assume that we start from the zero state, need a 7-bit bit stream **b** which will lead to the $\mathrm{v}_0[i]$ having all-ones, and don't care what $\mathrm{v}_1[i]$ will be. Note that this is equivalent to saying that $\mathbf{b}(A)$=(1111111).

Then, by using the methodology explained above, *input-select* computes **b**=(1101010) which should be used at the ENC input. For this input combination $\mathrm{v}_0[i]$ and $\mathrm{v}_1[i]$ become (1111111) and (1001001) respectively, where the latter is not important for us. Therefore, we selected half of the output bits by using the degrees of freedom provided by the structure of the code. Note that, we can also fix bits of $\mathrm{v}_1[i]$, the necessary condition being that only one of the outputs bits at a given time can be fixed prior to encoding.

2) Effect of Puncturing: We know that puncturing is provided in 802.11 to reach higher coding rates such as 2/3, 3/4, and 5/6, by disregarding some of the encoded bits according to the puncturing pattern [6, Fig. 18.9]. Considering encoding and puncturing operations as two serially concatenated steps, it can be observed that puncturing adds more DC terms at the output of the ENC by increasing the number of output bits (adding the ones which will be omitted) whose values do not matter. Equivalent to say, puncturing increases the fraction of coded bits (after puncturing) that we can fix in advance.

As a different example, consider the rate 2/3 ENC. The puncturing pattern for this code is $[1\ 1\ 1\ 0]^T$. By using the same approach, we can fix half of the bits at the output of the mother ENC again. However, this corresponds to two-thirds of the coded bits after puncturing (of course, if we fix the bits which will not be punctured). This idea is illustrated in Fig. 6.

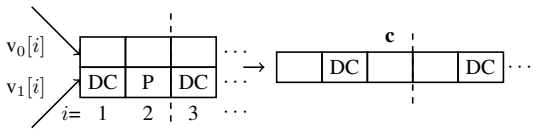

Fig. 6. The structure of the output codeword **c** of mother ENC of rate $r = 1/2$ followed by a puncturing operation of pattern $[1\ 1\ 1\ 0]^T$ to rate 2/3. DCs represent the bits which we don't care about. Ps represent the bits which will be omitted. These degrees of freedom enable us to determine the other bits. Note that due to puncturing, the fraction of bits in **c** that we can decide beforehand increases. The dashed partitioning shows the blocks before and after puncturing.

From a different perspective, if we set some of the output bits **c** to the input bits **b**, the effective ENC can be considered as systematic *virtually* (i.e., without changing its structure hardware-wise). Then, fixed output bits are the systematic outputs, and the remaining are the parity. The role of ENC here is to add redundancy by deciding the values of the output bits that we treat as DC terms (which will specify the signs). This practically systematic ENC uses only a subset of the main codeword space, therefore the performance should not degrade.

3) Effect of Interleaving: The last step is to take the bit-level interleaver Π of 802.11 into consideration. Π permutes coded bits in a pre-determined way prior to the mapping. The goal is to be sure that adjacent coded bits will be mapped onto nonadjacent subcarriers and that they will not be mapped onto low reliable bits of the constellation in a row [6].

Due to Π's deterministic nature, we can find which output bits of the ENC will be used as $\mathbf{b}(A)$ and $b(S)$ by the mapper considering the type of mapping explained in Sec. IV-A. As an example, an illustration of **c** is given in Fig. 7 for the rate $r = 2/3$ code with 64-QAM constellation mapping considering the effect of Π.

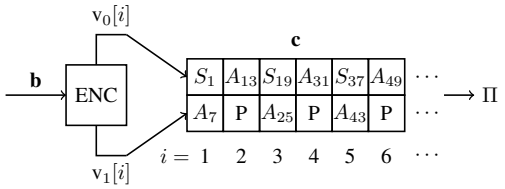

Fig. 7. The functions of the bits at the output of the mother ENC which will be punctured to rate $r = 2/3$ and mapped onto 64-QAM symbols. A_i and S_i represent that the bits in their positions will be used as the binary labels of the i^{th} sign or amplitude (e.g., as $b(S_i)$ or one of $\mathbf{b}(A_i)$) by the mapper (after puncturing and interleaving). P represents the bits which will be punctured. In this structure, $b(S_i)$'s and Ps are considered as DC terms giving the freedom to choose $\mathbf{b}(A_i)$'s.

D. Overall Structure

In simple terms, we let the enumerative shaper specify A_i's and ENC produce the S_i's to generate signal points X_i's by using the following procedure:

1) Prescribe n amplitudes A_i's corresponding to k input bits by using *enumerative amplitude shaping* idea given in Fig. 3.
2) Select the proper input to ENC such that it will place the binary amplitude labels $\mathbf{b}(A_i)$'s to the specified positions in the codeword **c**, by using the *input-select* idea given in Fig. 5.
3) Use the other bits produced by ENC (e.g., effectively parity output bits) as the binary sign labels $b(S_i)$.
4) Signal points $X_i = S_i \times A_i$ will have an approximately Gaussian distribution.

The functional diagram of the block which should be put in front of an IEEE Std 802.11 transmitter chain to realize this procedure is given in Fig. 8. We call this block *Shaped Input Selection* (SIS) since it first addresses the shaped amplitudes and then finds the correct input to the ENC.

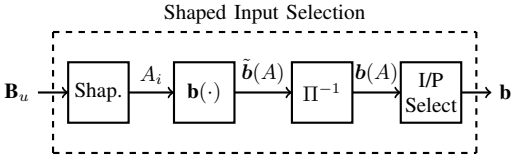

Fig. 8. Shaped Input Selection. Note that $\tilde{b}$ indicates that the bits in b are interleaved.

For the sake of completeness, an enumeratively shaped IEEE 802.11 transmitter is illustrated in Fig. 9. In this respect, SIS can be seen as a black box which maps uniform data to specific bit streams (in a one-to-one manner), and makes ENC address certain codewords: The ones in an n-sphere.

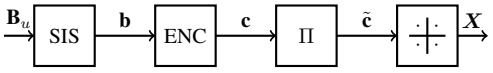

Fig. 9. Block diagram of a shaped 802.11 transmitter. SIS produces the correct bit stream $\mathbf{b}$ which will make the following blocks produce the shaped signal points X_i's.

Finally, the reverse operations of the blocks inside SIS should be performed at the end of the PHY receiver. The inverse of *input-select* can be implemented in the same manner since ENC terminates the trellis also at state zero. For the optimal decoding, the Viterbi algorithm should operate on the product of the enumerative trellis and coding trellis. However, we can use only the coding trellis which leads to a suboptimal decoding without losing from the performance too much as in [16]. Note that, a-priori information about the symbols should be taken into account when calculating the bit-metrics (log-likelihood ratios, L-values) to be used as the branch metrics by the Viterbi engine. These a-priori symbol probabilities $p_X(x_i)$ can be calculated using $M(i, E_i)$. For the AWGN channel model $Y_i = X_i + N_i$, the metrics are calculated as

$$\mathrm{L}\left(b_k\left(y_i\right)\right) = \log\left(\frac{\sum\limits_{x_i \in X | b_k(x_i)=0} p_X(x_i) p_{Y|X}(y_i|x_i)}{\sum\limits_{x_i \in X | b_k(x_i)=1} p_X(x_i) p_{Y|X}(y_i|x_i)}\right), \tag{15}$$

where $b_k(x_i)$ describes the kth binary label of x_i.

E. Shaping for Higher Code Rates

For 2^{2m}-QAM constellations, a small modification of SIS structure is necessary to combine shaping with codes of rates larger than $(m-1)/m$. In such cases, a number of DC terms and equivalently the fraction of coded bits that we can decide prior to ENC increases as explained in Sec. IV-C2. Then in addition to the $\boldsymbol{b}(A)$'s, we can prescribe some of the $b(S)$'s. By modifying the *input-select* block, we put some of the uniformly distributed data bits to the positions of some sign bits in $\mathbf{c}$ in Fig. 7. The block diagram of this modified SIS is given in Fig. 10.

V. Results and Discussion

We simulated the packet error performance of the proposed enumeratively shaped 802.11 system. 64- and 256-QAM constellations with code rates r={1/2, 2/3, 3/4, 5/6} are considered. Note that we also simulated some combinations of constellations and code rates which are not allowed in the standard [6]. Encoding with trellis

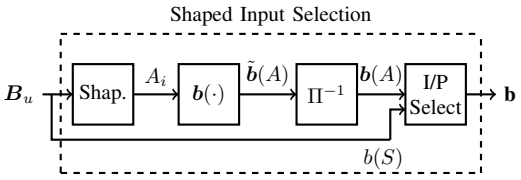

Fig. 10. Modified Shaped Input Selection. For code rates higher than $(m-1)/m$ with 2^{2m}-QAM constellation mappings, we have more DC terms than the $\boldsymbol{b}(A)$'s that we should fix in $\mathbf{c}$. We use these additional degrees of freedom to fix some of the $b(S)$'s.

termination, puncturing, interleaving, subcarrier modulation mapping (with 4 pilots and 12 empty subcarriers), and OFDM modulation are implemented as they are explained in [6, Clause 17]. The modulation-dependent normalization factor $\mathrm{K_{MOD}}$ [6, Table 17-11] is modified using the average energy of the effective constellations after shaping which can be calculated from $M(i, E_i)$ easily.

Circularly symmetric complex Gaussian noise is added in the time domain. Fading is not considered. The signal-to-noise ratio (SNR) is also calculated in the time domain (see Fig. 11 for a remark on this). Noise power is assumed to be estimated perfectly. Bit-metrics are calculated using max-log approximation considering the non-uniform a priori probabilities. The Viterbi algorithm is used for decoding as recommended by the standard. As the reverse operation of *input-select*, the estimates are encoded again to find $\boldsymbol{b}(A)$'s and $b(S)$'s since ENC will be practically available in the hardware. Finally, we say that a packet error occurs whenever at least one of the bits in a packet prescribed by the shaper is decoded erroneously.

The payload size is taken to be 200 bytes. n is taken to be 96 (since there are 48 QAM symbols in an OFDM symbol) implying that shaping is done over OFDM symbols. E_{max} is selected for every R using (11). Observe that instead of using a larger constellation and shaping its probability distribution to have a rate smaller than its maximum entropy, we used a higher code rate and shaping. This approach brings no constellation expansion which is one of the main drawbacks of shaping. For each of the code rate and modulation type combinations, different rates are simulated by gradually decreasing E_{max} (e.g., making the n-sphere smaller). The results of the simulations along with the Shannon capacity and the standard operating points of 802.11 are given in Fig. 11.

Rates R in the range of 3 to 6 bits/2-D are simulated. In this range, enumerative amplitude shaping (along with puncturing) brings an SNR gain of 1 dB (at $R = 4$ with 256-QAM) to 1.61 dB (at $R = 3$ with 64-QAM). As we shape a constellation in combination with a higher code rate, the gain increases. For example at 5.33 bits/2-D, the gain increases from 1.11 dB to 1.5 dB when we increase the code rate from 3/4 to 5/6 while the constellation is

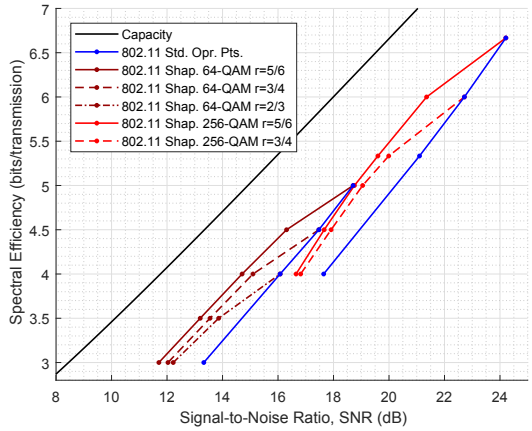

Fig. 11. Packet error probability = 10^{-2}. For each point, 4 different simulations are completed, and for each of them at least 200 packet errors are collected. r={2/3, 3/4, 5/6} with 64-QAM, and r={3/4, 5/6} with 256-QAM are allowed combinations in 802.11. The others are simulated to provide references. Note that there is a $10\log_{10}(52/64) \approx -0.9$ dB difference between the observed and the actual SNR values since 12 out of 64 subcarriers are empty in [6].

256-QAM. We see that for $R = 4$, 64-QAM with rate 2/3 performs better than any 256-QAM scheme with or without shaping. Therefore, the gain there, is calculated considering only 64-QAM schemes and is 1.37 dB.

Notice that the gains are larger than the ultimate shaping gain in some cases. This is because of the fact that as we increase the code rate r, the gap to capacity decreases in AWGN channels for a range of r, which introduces a *puncturing gain*. It is indicated in [7] that there is a performance gap between coded modulation (CM) and BICM. Over AWGN channels, CM performs better than BICM and the gap between them increases with constellation size and decreases with code rate. This can be seen as a kind of puncturing gain in AWGN channels, and is the reason why we are seeing gains larger than (14) when we use a larger code rate and shape the distribution back to the target rate. Note that the ultimate shaping gain only represents the effect of geometry, and implies nothing about coding. This characteristic of CM and BICM is reversed for Rayleigh fading channels [7]. Therefore the performance of our shaping algorithm should also be investigated in different channel conditions. Lastly, observe that it is easy to introduce rate granularity by using the flexibility of choosing E_{max}. For instance, 802.11 cannot operate at the rates 3.5, and 5.33 bits/2-D where shaped scheme can.

VI. Conclusion

A comparison showing that the enumerative amplitude shaping performs more efficiently than the probabilistic approach is given. Then, a shaped transmission scheme based on the enumerative approach is proposed for IEEE 802.11. A shaper is used to find and index all signal points inside an hypersphere. A shaped input selection block produces the inputs that are processed by the encoder. The encoder generates the outputs in such a way that they will be mapped to the symbols having the amplitudes prescribed by the shaper. Overall, this scheme transmits uniform information bits using approximately Gaussian distributed channel inputs without requiring a constellation expansion, and introduces rate granularity. Performance improvements up to 1.61 dB are observed for rates 3 to 6 bits/2-D. Puncturing contributes to these gains by decreasing the coding gap in AWGN channels.

Acknowledgment

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

References

[1] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul 1948.

[2] G. Ungerboeck, *Huffman Shaping*. Boston, MA: Springer US, 2002, pp. 299–313.

[3] R. F. H. Fischer, *Precoding and signal shaping for digital transmission*. J. Wiley-Interscience, 2002.

[4] P. Schulte and G. Böcherer, "Constant Composition Distribution Matching," *IEEE Transactions on Information Theory*, vol. 62, no. 1, pp. 430–434, Jan 2016.

[5] G. Böcherer, F. Steiner, and P. Schulte, "Bandwidth Efficient and Rate-Matched Low-Density Parity-Check Coded Modulation," *IEEE Transactions on Communications*, vol. 63, no. 12, pp. 4651–4665, Dec 2015.

[6] "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534, Dec 2016.

[7] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 927–946, May 1998.

[8] J. Wuijts, "Shaping and Coding for n-dimensional Pulse Amplitude Modulation," Master's Thesis, Eindhoven University of technology, 1993.

[9] G. Forney, R. Gallager, G. Lang, F. Longstaff, and S. Qureshi, "Efficient Modulation for Band-Limited Channels," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 5, pp. 632–647, Sep 1984.

[10] R. Laroia, N. Farvardin, and S. A. Tretter, "On optimal shaping of multidimensional constellations," *IEEE Transactions on Information Theory*, vol. 40, no. 4, pp. 1044–1056, Jul 1994.

[11] G. D. Forney, "Trellis shaping," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 281–300, March 1992.

[12] F. Willems and J. Wuijts, "A pragmatic approach to shaped coded modulation," in *IEEE 1st Symposium on Communications and Vehicular Technology in the Benelux*, 1993.

[13] J. Schalkwijk, "An algorithm for source coding," *IEEE Transactions on Information Theory*, vol. 18, no. 3, pp. 395–399, May 1972.

[14] G. Böcherer and B. C. Geiger, "Optimal Quantization for Distribution Synthesis," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6162–6172, Nov 2016.

[15] F. R. Kschischang and S. Pasupathy, "Optimal nonuniform signaling for Gaussian channels," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 913–929, May 1993.

[16] A. R. Calderbank, T. A. Lee, and J. E. Mazo, "Baseband trellis codes with a spectral null at zero," *IEEE Transactions on Information Theory*, vol. 34, no. 3, pp. 425–434, May 1988.