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 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 [6], the performance of this concept improves with increasing block length $n$.

On the other hand, we are interested in improving the performance of IEEE Std 802.11 [7] 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 when using shaping is to use only the signals in a set having the smallest possible average energy in $n$ dimensions, with 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

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

(1)

where $\sigma^2$ is the variance of the additive channel noise and where $P$ 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. In particular, $2^m$-PAM operates with values taken from the set $\{\pm 1, \pm 3, \ldots, \pm (2^m - 1)\}$, for $m = 1, 2, \ldots$.

For $2^m$-PAM, the channel input sequences $(x_1, x_2, \cdots, x_n)$ of length $n$ can be regarded as points in an $n$-dimensional space, and the set of all possible signal sequences constitutes an $n$-cube. Moreover, if these sequences are transmitted with equal probability, the channel inputs $X_n$ for all $n$ have a discrete uniform distribution.

For carrier transmission, typically quadrature amplitude modulation (QAM) is used which can be regarded as points in a square with corner points $(\pm \sqrt{P}, \pm \sqrt{P})$ [3, Sec. 4.2.6]). For carrier transmission, typically quadrature amplitude modulation (QAM) is used which can be regarded as points in an $n$-cube. Moreover, if these sequences are transmitted with equal probability, the channel inputs $X_n$ for all $n$ have a discrete uniform distribution.

The mutual information (MI) of a $2^{2m}$-QAM scheme with uniform signalling can be computed as a function of the signal-to-noise ratio $P/\sigma^2$ as in [8]. This mutual information indicates what transmission rates can be realized with diminishing error rates. In Fig. 1, we can find these rates for $m = 1, 2, 3, 4$. In this figure, we also plot the capacity in bits per two dimensions.

![Fig. 1. AWGN capacity, hence input is Gaussian (solid), and MI for a uniformly distributed input (dashed), and the MI between channel input and output for $2^{2m}$-QAM schemes where $m = 1, 2, 3, 4$.](image)

As expected, the mutual information converges to the number of bits corresponding to the constellation size. The gap to capacity includes the effects of having a discrete input set and using a uniform distribution over these inputs.

### C. Shaping Gap

In Fig. 1, we have also plotted the mutual information of the AWGN channel with uniformly distributed input over a square with corner points $(\pm \sqrt{P}, \pm \sqrt{P})$ [3, Sec. 4.2.6]. We observe that there is a gap to capacity which is the consequence of using uniform instead of Gaussian inputs. This gap can be shown to approach 1.53 dB for large SNR’s.

1We do not consider QAM constellations that do not fit into this framework, e.g. 8-QAM.

is equivalent to 0.255 bits per dimension. Another important observation is that this (shaping) gap disappears when the SNR decreases.

### D. Hyperspheres and Gaussian Distribution

In [9], a circle is given as the most energy efficient geometry in 2-D intuitively. Therefore we consider $n$-sphere as the extension of this reasoning and investigate its properties.

Consider an $n$-sphere of radius $\rho$. Let $X$ designate the set of $n$-vectors $(x_1, x_2, \cdots, x_n)$ inside, and assume a continuous uniform probability distribution $p(x) = 1/V_{sph}$ where $V_{sph}$ is the volume. Using the derivation in [10], the joint differential entropy of $X$ is

$$h(X) = -\int_V p(x) \log p(x) \, dx = \log \left( \frac{\rho^n \pi^{n/2}}{(n/2)!} \right),$$

for even $n$ where we used Eq. (4.642) in [11]. 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

$$h(X_i) = \frac{\sum_k h(X_k)}{n} \geq \frac{1}{n} \log \left( \frac{\rho^n \pi^{n/2}}{(n/2)!} \right),$$

where (a) follows from independence bound on entropy, (b) is from Eq. (2), and (c) is an application of the Stirling’s approximation. Finally, we use the fact that Gaussian distribution has the maximum entropy among all continuous distributions for a specified variance and use it to upper bound as

$$h(X_i) = \frac{\sum_k h(X_k)}{n} \leq \frac{1}{n} \log \left( 2\pi e \sqrt{E[X^2]} \right),$$

where (a) follows from independence bound on entropy, (b) is from Eq. (2), and (c) is an application of the Stirling’s approximation. Finally, we use the fact that Gaussian distribution has the maximum entropy among all continuous distributions for a specified variance and use it to upper bound as

$$h(X_i) = \frac{\sum_k h(X_k)}{n} \leq \frac{1}{n} \log \left( 2\pi e \sqrt{E[X^2]} \right),$$

Therefore, as $n$ increases, the distribution of $X$ along any axis converges to a zero-mean Gaussian distribution of variance $\rho^2/n$.

Furthermore, it is shown in [3, Sec. 4.2.3] that the ratio of the average energy of the $n$-sphere to that of the $n$-cube having the same volume is

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

and called the shaping gain in [9]. By using Stirling’s approximation, $G(\infty) \approx \pi e/6 = 1.53$ dB. Note that having the same number of points in discrete situation is equivalent to have the same volume in the continuous situation.
III. ENUMERATIVE AMPLITUDE SHAPING

Simply, we want to find, address, and use all PAM symbol sequences \( \mathbf{x} \) (e.g., signal points) inside an \( n \)-sphere to have a Gaussian distributed channel input. We will utilize an enumerative approach for which similar results can be found in [12] and [13]. Now, we explain how to realize this.

A. Enumerative Amplitude Trellis

To investigate our enumerative sequence finding and addressing approach, we will use an example as in [14]. Consider the one-sided alphabet \( \mathcal{X}_+ = \{1, 3, 5\} \), the block length \( n = 4 \), and the maximum energy \( E_{\text{max}} = 28 \). 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.

```
\begin{align*}
&0 \quad 1 \quad 2 \quad 3 \quad 4 \\
\mid \quad \mid \quad \mid \quad \mid \quad \mid \\
&19 \quad 1 \quad 6 \quad 3 \quad 1
\end{align*}
```

1

\begin{align*}
&1 \quad 25 \quad 26 \quad 27 \quad 28 \\
\mid \quad \mid \quad \mid \quad \mid \\
&5 \quad 18 \quad 2 \quad 20
\end{align*}

\begin{align*}
&1 \quad 1 \quad 1 \quad 1 \quad 1
\end{align*}

Fig. 2. Enumerative shaping trellis for \( n = 4 \) and \( E_{\text{max}} = 28 \).

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 [14]. Basic rules of the structure are the following:

- Each path starts in the zero-energy vertex and ends in one of the final nodes having energy not larger than \( E_{\text{max}} \).
- 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 \( M(i, E_i) \).

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

\[
M(n, e) = \begin{cases} 
1 : & 0 \leq e \leq E_{\text{max}} \\
0 : & \text{otherwise} 
\end{cases}, \quad (10)
\]

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

Consequently, the number all of all sequences \( \mathbf{x} \) of length \( n = 4 \) having an energy \( E(\mathbf{x}) \leq E_{\text{max}} \) 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

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

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 [15], we can write an efficient algorithm to calculate the index of a sequence \( \mathbf{x} \) as

\[
i(\mathbf{x}) = \sum_{t=1}^{T} \sum_{w \in \mathcal{X}_+} M(t, w^2 + \sum_{s=1}^{t-1} x_s^2). \quad (12)
\]

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 [12]. Fig. 3 gives an illustration of the shaper.

![Shaper](image)

Fig. 3. Shaper maps \( k \) uniformly distributed input bits to \( n \) unsigned symbols. The symbol sequences are taken from the \( n \)-sphere specified by block length \( n \) and maximum energy \( E_{\text{max}} \). As \( n \) goes to infinity, the probability distribution of any element of \( \mathbf{S}_n \) converges to a one-sided Gaussian. Shaper addresses these sequences with the enumerative algorithm. Changing \( k \) changes the maximum index that we map to the paths. Note that each sequence \( \mathbf{s}_+ \) is selected with equal probability.

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 [16, 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 \( R_{\text{ccdm}} \) approaches to the target rate \( R \) as \( n \rightarrow \infty \). 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 [6].

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

$$ R = \frac{1}{n} \left\lfloor \log_2 \left( \frac{n!}{n_1^n n_2^n n_3^n n_4^n} \right) \right\rfloor = 1.75, \quad (13) $$

for $n = 96$. This scheme uses only some of the signal points on the surface and $E_{\text{ave}}^{\text{cdcm}} = 13.25$. If we were to use all points on a surface, we would take $E_{\text{max}} = 1160$ to achieve 1.75 bits/dimension, and $E_{\text{ave}}^{\text{surf}} = 12.08$. Finally, if we consider the enumerative scheme with $E_{\text{max}} = 1120$, all points in and on the surface leads to the same rate with $E_{\text{ave}}^{\text{w}} = 11.43$. 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 [17]

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

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 implement the enumerative 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

Combinations of enumerative shaping and coding are investigated in [14] and [18]. In this section, 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 II is over the whole codeword in BICM [8] where it is over an OFDM symbol in [7] to keep latency small. The BICM block diagram is illustrated in [8, Fig. 1].

IEEE 802.11 employs a Gray-coded $2^{m}$-QAM constellation mapping in the form of a Cartesian product of two $2^{m}$-PAM constellations. This mapping is illustrated for 16-QAM in [7, Fig. 17-10]. In such a scheme, every $2m$ bits $(b_1b_2\cdots b_{2m})$ are mapped to a complex signal point in the 2-D plane. First $m$ bits $(b_1 b_2 \cdots b_m)$ form the binary label of the real (in-phase) part, and the last $m$ bits $(b_{m+1} b_{m+2} \cdots b_{2m})$ form the binary label of the imaginary (quadrature) part. We express any signal $X_i$ from the PAM signal set $X$ 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 $b(A)$ is the rest. From this point onwards, we will call labels of the signs $b(S)$ sign bits, and labels of the amplitudes $b(A)$ 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 II which is defined by two concatenated permutations prior to the mapper. The ENC structure can be seen in [7, Fig. 17-8] where $T_b$ 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 $v_0[i]$ and $v_1[i]$. Note that for an input data stream $b[n] (b_1 b_2 \cdots)$, the output bit stream is in the form of $(v_0[1]v_1[1])v_0[2]v_1[2]\cdots$.

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 $i$, we can write the output bit equations as

$$ v_0[i] = g_0 (b_i, s_i), \quad v_1[i] = g_1 (b_i, s_i), \quad (15) $$

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 $(v_0[i]v_1[i])$ is either in the set $V_1 = \{(01), (10)\}$ or in $V_2 = \{(00), (11)\}$. 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 $V_i$. 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](image-url)
which will not be punctured). This idea is illustrated in Fig. 6.

Note that, we can also fix bits of $v_1[i]$, 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 [7, 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 output 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.

3) Effect of Interleaving: The last step is to take the bit-level interleaver $\Pi$ of 802.11 into consideration. $\Pi$ 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 [7].

Due to $\Pi$’s deterministic nature, we can find which output bits of the ENC will be used as $b(A)$ 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 $\Pi$.

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 $b(A_i)$’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.

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.

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 [19]. 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

\[
L(b_k(y_i)) = \log \left( \frac{\sum_{x_i \in X} p_X(x_i)p_{Y|X}(y_i|x_i)}{\sum_{x_i \in X} p_X(x_i)p_{Y|X}(y_i|x_i)} \right),
\]

(16)

where \( b_k(x_i) \) describes the \( k \)th 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 \( b(A) \)'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 \( e \) 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 [7]. Encoding with trellis termination, puncturing, interleaving, subcarrier modulation mapping (with 4 pilots and 12 empty subcarriers), and OFDM modulation are implemented as they are explained in [7, Clause 17]. The modulation-dependent normalization factor \( K_{MOD} \) [7, 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 \( b(A) \)'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 Eq. (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 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 [8, Fig. 6] 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 Eq. (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 [8, Fig. 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 \( E_{\text{max}} \). 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