EXIT Charts for Non-Binary LDPC Codes Over Arbitrary Discrete-Memoryless Channels

Amir Bennatan and David Burshtein
Department of Electrical Engineering–Systems, Fleischman Faculty of Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel
Email: abn@eng.tau.ac.il, burstyn@eng.tau.ac.il

Abstract—We consider coset LDPC codes over GF(q), designed for use over arbitrary channels (particularly nonbinary and asymmetric channels). We show that the random selection of the nonzero elements of the GF(q) parity-check matrix induces a permutation-invariance property on the densities of the messages produced by the decoder. We use this property to show that under a Gaussian approximation, the entire q − 1 dimensional distribution of the vector messages is described by a single scalar parameter. We apply this result to develop EXIT charts for our codes. We use appropriately designed signal constellations to obtain substantial shaping gains. Simulation results indicate that our codes outperform multilevel codes at short block lengths. We also present results for the AWGN channel at 0.56 dB of the unconstrained Shannon limit (i.e., not restricted to any signal constellation) at a spectral efficiency of 6 bits/s/Hz.

I. INTRODUCTION

With binary LDPC codes, Chung et al. [4] observed that the message densities tracked by density-evolution are well approximated by Gaussian densities. Furthermore, the symmetry of the messages in binary LDPC decoding implies that the mean m and variance σ2 of each density are related by σ2 = 2m. Thus, a symmetric Gaussian random variable may be described by a single parameter. This property was also observed by ten Brink et al. [15] and is essential to their development of EXIT charts.

Nonbinary LDPC decoders use messages that are q − 1 dimensional vectors, rather than scalar values. Li et al. [10] proved that under a Gaussian assumption, the distributions of the messages are completely characterized by q − 1 parameters. In the full version of this paper [2], we show that the distributions are in fact characterized by a single scalar parameter, like the distributions of binary LDPC codes. This result enables us to extend the EXIT chart method of [15] to coset GF(q) LDPC codes. We use EXIT charts to design nonbinary LDPC codes and present simulation results that demonstrate the effectiveness of our approach.

Note that permutation-invariance, which was the key property that enabled these results, has also been instrumental in extending the stability condition to arbitrary channels [2], [3].

This research was supported by the Israel Science Foundation, grant no. 22/01–1 and by an equipment grant from the Israel Science Foundation to the school of Computer Science at Tel Aviv University.

II. BACKGROUND

A. Coset GF(q) LDPC Codes

Our definition starts with GF(q) LDPC codes, which were first considered in [5]. A GF(q) LDPC code is defined in a way similar to binary LDPC codes, using a bipartite Tanner graph, with the following important addition: At each edge (i, j) of the graph, a label gi,j ∈ GF(q) \{0\} is defined. Each label is randomly selected with uniform probability in GF(q) \{0\}. A word x with components from GF(q) is a codeword if at each check-node j, the following equation holds:

\[ \sum_{i \in \mathcal{N}(j)} g_{i,j} x_i = 0 \]

where \( \mathcal{N}(j) \) is the set of variable nodes adjacent to j. In the code’s sparse parity-check matrix, the labels \( g_{i,j} \) correspond to the nonzero elements.

Rather than use plain GF(q) LDPC codes, it is useful instead to consider coset codes (an approach first considered by Kavčič et al. [9]). A coset code is obtained from a linear code by adding a fixed vector v (called the coset vector) to each of the codewords.

In the sequel, we examine our codes in a random coset setting, where the underlying LDPC code remains fixed and performance is averaged over all possible realization of the coset vector. This produces an effect that is similar to that of output-symmetry in the analysis of standard LDPC codes [13].

With binary LDPC codes, the BPSK signals ±1 are typically used instead of the \{0, 1\} symbols of the code alphabet. With nonbinary codes, we let \( \delta : GF(q) \rightarrow A \) denote a mapping from the code alphabet to the signal alphabet \( A \).

B. Plain-Likelihood and LLR Vectors

As noted above, nonbinary LDPC decoders use messages that are multidimensional vectors. Like binary decoders, there are two useful representations for the messages: plain-likelihood vectors and log-likelihood-ratio (LLR) vectors.

A q-dimensional plain-likelihood probability vector is a vector \( x = (x_0, ..., x_{q-1}) \) of real numbers such that \( x_i \geq 0 \) for all i and \( \sum_{i=0}^{q-1} x_i = 1 \). We let the indices \( i = 0, ..., q-1 \) be interpreted as elements of GF(q). That is, each index i is taken to mean the ith element of GF(q), given some enumeration of the field elements (we assume that indices 0 and 1 correspond to the zero and one elements of the field, respectively).
The LLR values associated with $x$ are $w_i \triangleq \log(x_i/x_i')$, $i = 0, ..., q - 1$ (a definition borrowed from [10]). Notice that for all $x$, $w_0 = 0$. We define the LLR-vector representation of $x$ as the $q - 1$ dimensional vector $w = (w_1, ..., w_{q-1})$. For convenience, although $w_0$ is not defined as belonging to this vector, we will allow ourselves to refer to it with the implicit understanding that it is always equal to zero.

C. Decoding

The coset $GF(q)$ LDPC belief-propagation decoder is based on Gallager [7]. The decoder attempts to recover $c$, the codeword of the underlying $GF(q)$ LDPC code. The decoding process is modelled on the standard binary LDPC decoder, with the important difference that messages are vector-valued rather than scalar-valued. In this paper, we consider only decoders that employ LLR-vector messages.

At the initial rightbound iteration, the elements of the rightbound message at a variable node $i$, are given by,

$$r_i^{(0)} = \log \frac{\Pr[y_i | \delta(v_i)]}{\Pr[y_i | \delta(k + v_i)]}$$

where $y_i$ and $v_i$ are the corresponding channel output and coset vector component, respectively. At later rightbound iterations, the outgoing message across an edge $(i, j)$ is given by,

$$r = r^{(0)} + \sum_{n=1}^{d_i-1} l_i^{(n)}$$

where $d_i$ is the degree of the node $i$ and $l_i^{(1)}, ..., l_i^{(d_i-1)}$ are the incoming (leftbound) LLR messages across the edges $\{(i, j') : j' \in N(i) \setminus j\}$. $r^{(0)}$ is the message that was sent at the initial iteration, and addition of vectors is performed componentwise.

Now consider the leftbound iteration. Our expression below assumes that all messages are in plain-likelihood form. We therefore convert the rightbound messages from LLR, apply the expression, and convert the result back to LLR form. Element $l_k$ of the leftbound message across an edge $(i, j)$ is given by,

$$l_k = \sum_{a_1, ..., a_{d_i-1} \in GF(q)} \sum_{g_a = -g_{d_i} k}^{d_j-1} c^{(n)}_a$$

where $d_j$ is the degree of node $j$, $r^{(1)}, ..., r^{(d_j-1)}$ denote the rightbound messages across the edges $\{(i', j) : i' \in N(j) \setminus i\}$ and $g_1, ..., g_{d_j-1}$ are the labels on those edges. $g_{d_j}$ denotes the label on the edge $(i, j)$. The summations and multiplications of the indices $a_n$ and the labels $g_a$ are performed over $GF(q)$.

The complexity of computing the leftbound messages can be dramatically reduced using a method due to Richardson and Urbanke [13][Section V] employing the multidimensional DFT, coupled with an efficient DFT algorithm [6][page 65].

D. The All-Zero Codeword Assumption

An important property of standard binary LDPC decoders [13] is that the probability of decoding error is equal for any transmitted codeword. With coset $GF(q)$ LDPC codes, we show in [2], [3] that in a random-coset setting, the probability of error is independent of the transmitted codeword of the underlying LDPC code (that is, prior to the addition of the coset vector). This enables us to condition our analysis results on the assumption that the transmitted codeword corresponds to the all-zero codeword of the underlying LDPC code.

E. Symmetry of Message Distributions

The symmetry property, introduced by Richardson and Urbanke [12] is a major tool in the analysis of standard binary LDPC codes. In [2], [3], we produced the following generalization to multidimensional random variables.

**Definition 1:** An LLR-vector random variable $W$ is symmetric if $Pr[W = w] = e^{w} Pr[W = w^*]$ for all LLR-vectors $w$, indices $i \in GF(q) \setminus \{0\}$, and where $w^*$ is an LLR-vector whose components are given by $w^*_j = w_{j+i} - w_j$, $j = 1, ..., q - 1$ (i+j being evaluated over $GF(q)$).

It is easy to verify that when $q = 2$, this definition coincides with the standard definition for symmetry. In [2] we show that in a random-coset setting, under the independence assumption and the all-zero codeword assumption, the messages of coset $GF(q)$ LDPC codes are symmetric.

III. PERMUTATION-INVARANCE

We now discuss permutation-invariance, the key property that simplifies the analysis of coset $GF(q)$ LDPC codes.

Consider expression (2) for computing the leftbound messages in the process of belief-propagation decoding. Given an LLR-vector $w$ and an element $i \in GF(q) \setminus \{0\}$, we define $w^{x_i}$ to be an LLR-vector whose components are given by $w^{x_i}_j = w_{j+i}$, $j = 1, ..., q - 1$, and where $i \cdot j$ is evaluated over $GF(q)$ (the same definition applies to plain-likelihood vectors). We also define the $GF(q)$ convolution operator $\oplus$ in the following way,

$$x^{(1)} \oplus x^{(2)} = \sum_{a \in GF(q)} x^{(1)}_a \cdot x^{(2)}_{k-a}$$

Using these definitions, we can rewrite (2) as

$$1 = \left[ \bigoplus_{n=1}^{d_j-1} (r^{(n)})^{\times g_a} \right]^{\times (-g_{d_j})}$$

In an ensemble analysis, the label $g_{d_j}$ is a random variable, independent of the other labels, of the rightbound messages $\{r^{(n)}\}$ and consequently of $m = \bigoplus_{n=1}^{d_j-1} (r^{(n)})^{\times g_a}$. For any fixed $h \in GF(q) \setminus \{0\}$, consider $I^h$. From the above discussion, we have $I = (m^{x_i})^h = m^{x_i(-g_{d_j}h)}$, and obtain that $I$ is distributed identically with $I$. This leads us to the following definition.

**Definition 2:** An LLR-vector random variable $W$ is permutation-invariant if for any fixed $h \in GF(q) \setminus \{0\}$, the random variable $W^h \triangleq W^{x_i h}$ is distributed identically with $W$. 

Although (3) assumes plain-likelihood representation, the result applies to LLR representation as well.
Thus the leftbound messages $I$ in LDPC decoding are permutation-invariant. The rightbound messages, $r^{(n)}$, in general are not permutation-invariant. However, it is easy to show [2] that letting $\bar{r}^{(n)} \triangleq r^{(n)} \times q_{n}$, $n = 1, \ldots, d - 1$, $\bar{r}^{(n)}$ are permutation-invariant, symmetric and $P_{r}(\bar{r}^{(n)}) = P_{r}(r^{(n)})$ ($P_{r}(\cdot)$ being the probability of error under the all-zero codeword assumption).

IV. GAUSSIAN APPROXIMATION

Using symmetry and permutation-invariance, we can now state the following theorem, which considers the messages of belief-propagation decoding under a Gaussian assumption. The proof of the theorem is provided in [2].

Theorem 1: Let $W$ be an LLR-vector random-variable, Gaussian distributed with a mean $m$ and covariance matrix $\Sigma$. Assume that the probability density function $f(w)$ of $W$ exists and that $\Sigma$ is nonsingular. Then $W$ is both symmetric and permutation-invariant if and only if there exists $\sigma > 0$ such that,

$$m = \begin{bmatrix} \sigma^2/2 \\ \sigma^2/2 \\ \vdots \\ \sigma^2/2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \sigma^2 & \sigma^2 & \sigma^2/2 \\ \sigma^2 & \sigma^2 & \vdots \\ \sigma^2/2 & \vdots & \sigma^2 \end{bmatrix}$$ (4)

That is, $\Sigma_{i,j} = \sigma^2$ if $i = j$ and $\Sigma_{i,j} = \sigma^2/2$ otherwise.

A Gaussian symmetric and permutation-invariant random variable, is thus described by a single parameter $\sigma$. In Section V-A we discuss the validity of the Gaussian assumption with nonbinary LDPC codes.

V. EXIT CHARTS

Formally, EXIT charts (as defined in [15]) track the mutual information $I(C; W)$ between the message $W$ (rightbound or leftbound) transmitted across a randomly selected edge, and the code symbol $C$ at the corresponding variable node. Note that we assume that the base of the log function in the mutual information is $q$, and thus $0 \leq I(C; W) \leq 1$.

Two curves (functions) are computed: The VND (variable node decoder) curve and the CND (check node decoder) curve. The argument to each curve is denoted $I_{E}$ and the value of the curve is denoted $I_{E}$. With the VND curve, $I_{A}$ is interpreted as equal to the functional $I(C; L_{t})$ when applied to the distribution of the leftbound messages $L_{t}$ at a given iteration $t$. The output $I_{E}$ is interpreted as equal to $I(C; R_{t})$ where $R_{t}$ is the rightbound message produced at the following rightbound iteration. With the CND curve, the opposite occurs.

As in [15], the decoding process is predicted to converge if for all $I_{A} \in [0, 1]$, $I_{E,VND}(I_{A}) > I_{E,CND}^{-1}(I_{A})$. In an EXIT chart, the CND curve is plotted with its $I_{A}$ and $I_{E}$ axes reversed (see, for example, Fig. 2). Thus, the condition for convergence is that the VND curve be strictly greater than the reversed-axes CND curve.

3More precisely, we have proven that the ensemble average distribution of messages, under the independence assumption, is permutation-invariant.

4$C$ is the code symbol of the underlying LDPC code (that is, prior to the addition of the corresponding coset vector component).

Strictly speaking, an approximation of $I(C; W)$ requires not only the knowledge of the distribution of $W$ but primarily the knowledge of the conditional distribution $Pr(W | C = i)$ for all $i = 0, \ldots, q - 1$ (we assume that $C$ is uniformly distributed). However, in [2] we show that the messages of the coset $GF(q)$ LDPC decoder satisfy $Pr(W = w | C = i) = Pr(W = w^{+i} | C = 0)$, where $w^{+i}$ is defined as in Section II-E. Thus, we may restrict ourselves to an analysis of the marginal distribution $Pr(W | C = 0)$. We continue in [2] to show

$$I(C; W) = 1 - E \left[ \log_{q} \left( 1 + \sum_{i=1}^{q-1} e^{-W_{i}} \right) | C = 0 \right]$$ (5)

In [2], we further show that we may replace the conditioning on $C = 0$ by a conditioning on the transmission of the all-zero codeword. In the sequel, we assume that all distributions are conditioned on the all-zero codeword assumption.

In their development of EXIT charts for binary LDPC codes, ten Brink et al. [15] confine their attention to LLR message distributions that are Gaussian and symmetric. Under these assumptions, a message distribution is uniquely described by its variance $\sigma^{2}$. For every value of $\sigma$, they evaluate (5) (with $q = 2$) when applied to the corresponding Gaussian distribution. The result is denoted $J(\sigma)$ and is used in the computation of the EXIT curves.

The Gaussian assumption is not strictly true. With binary irregular LDPC codes, the distributions of rightbound messages are approximately Gaussian mixtures. The distributions of the leftbound messages resemble “spikes”. The EXIT method in [15] nonetheless continues to model these distributions as Gaussian. Simulation results are provided, that indicate that this approach still produces a very close prediction of the performance of binary LDPC codes.

The modelling of the message distributions of coset $GF(q)$ LDPC as Gaussian is even less accurate than it is with binary LDPC codes. This results from the distribution of the initial messages, which is not Gaussian even on an AWGN channel. In Section V-A below, we therefore provide a more accurate model for the distributions. Our method differs from [15] in its modelling of the initial and rightbound message distributions. We continue, however, to model the leftbound messages as Gaussian.

A. Modelling of Rightbound Messages

Consider, for simplicity, regular LDPC codes. Observe that the computation of the rightbound message using (1) involves the summation of i.i.d leftbound messages, $I^{(n)}$. This sum is typically well-approximated by a Gaussian random variable. To this sum, the initial message $r^{(0)}$ is added. With binary LDPC codes, transmission over an AWGN channel results in an initial message $r^{(0)}$ that is also Gaussian distributed (assuming the all-zero codeword was transmitted). Thus, the rightbound messages are very closely approximated by a Gaussian random variable.

With coset $GF(q)$ LDPC codes, the distribution of the initial messages may be determined using the following lemma.
Lemma 1: Assume transmission over an AWGN channel with noise variance \( \sigma^2 \) and with a mapping \( \delta \) to the channel signal alphabet, under the all-zero codeword assumption. Then the initial message satisfies \( v(0) = \alpha(v) + \beta(v) \cdot z \), where \( z \) is the noise produced by the channel, \( v \) is the coset symbol and \( \alpha(v) \) and \( \beta(v) \) are \( q - 1 \) dimensional vectors, dependent on \( v \), whose components are given by, \( \alpha(v)_i = \frac{1}{\sigma^2}((\delta(v) - \delta(v + i))^2) \), \( \beta(v)_i = \frac{1}{\sigma^2}((\delta(v) - \delta(v + i))) \).

We assume that the coset symbol \( V \) is a random variable uniformly distributed in \( GF(q) \). Thus, \( \alpha(V) \) and \( \beta(V) \) are random variables, whose values are determined by the mapping \( \delta(\cdot) \) and by the noise variance \( \sigma^2 \). The distribution of the channel noise \( Z \) is determined by \( \sigma^2 \). The distribution of the initial messages is therefore determined by \( \delta(\cdot) \) and \( \sigma^2 \). Following the above discussion, we model the distribution of the rightbound messages as the sum of two random vectors. The first is distributed as the initial messages above, and the second (the intermediate sum of leftbound messages) is modelled as Gaussian. Empirical distributions of the random variables, as observed in simulations, are provided in Fig. 1.

In [2], we show that the intermediate value (the second random variable) is symmetric and permutation-invariant. Thus, by Theorem 1, it is characterized by a single parameter \( \sigma \).

Summarizing, the approximate distribution of rightbound messages is determined by three parameters: \( \sigma^2 \) and \( \delta(\cdot) \), which determine the distribution of the initial message, and \( \sigma \), which determines the distribution of the intermediate value.

B. Computation of EXIT Curves

For every value of \( \sigma \), we define \( J_R(\sigma; \sigma_2, \delta) \) to equal (5) when applied to the rightbound distribution corresponding to \( \sigma, \sigma_2^2 \) and \( \delta(\cdot) \). In an EXIT chart, \( \sigma_2 \) and \( \delta(\cdot) \) are fixed. The remaining parameter that determines the rightbound distribution is \( \sigma \), and \( \sigma = J_R^{-1}(I; \sigma_2, \delta) \) is well-defined. We also define \( J(\sigma) \) in lines completely analogous to [15]. That is, \( J(\sigma) \) equals (5) when applied to a Gaussian distribution as defined in Theorem 1.

Direct computation of \( J \) and \( J_R \) involves the evaluation of a multidimensional integral. However, they can be approximated by empirically evaluating the expectation (5).

The EXIT curves may now be computed as follows.

1) The VND curve. For each left-degree \( i \), we let \( I_{E,VND}(I_A; i, \sigma_2, \delta) \) denote the value of the VND curve when confined to the distribution of rightbound messages across edges whose left-degree is \( i \).

2) The CND curve. For each right-degree \( j \), we similarly define \( I_{E,CND}(I_A; j, \sigma_2, \delta) \). The parameters \( \sigma_2 \) and \( \delta(\cdot) \) are used in conjunction with \( \sigma = J_R^{-1}(I_A; \sigma_2, \delta) \) to characterize the distribution of the rightbound messages at the input of the check-nodes. Computation of \( I_{E,CND}(I_A; j, \sigma_2, \delta) \) is then performed empirically.

Note that \( J_R(\sigma; \sigma_2, \delta) \) needs to be computed once for each choice of \( \sigma_2 \) and \( \delta(\cdot) \). \( I_{E,CND}(\sigma; j, \sigma_2, \delta) \) needs to be computed also for each value of \( j \). \( J(\sigma) \) needs to be computed once for each choice of \( q \).

Given an edge distribution pair \( (\lambda, \rho) \) we evaluate,

\[
I_{E,VND}(I_A; \sigma_2, \delta) = \sum_{j=2} \lambda_j I_{E,VND}(I_A; j, \sigma_2, \delta)
\]

\[
I_{E,CND}(I_A; \sigma_2, \delta) = \sum_{j=2} \rho_j I_{E,CND}(I_A; j, \sigma_2, \delta)
\]

To define \( \lambda \), we fix \( p \) and maximize \( \sum \lambda_i/\epsilon \) (which is equivalent to maximizing the design rate of the code) subject to the following constraints (borrowed from [15]),

1) \( \lambda \) is required to be a valid probability vector. That is \( \lambda_i \geq 0 \forall i \), and \( \sum \lambda_i = 1 \).

2) To ensure decoding convergence, we require \( I_{E,VND}(I, I(0)) > I_{E,CND}(I) \) (as explained above) for all \( I \) belonging to a discrete, fine grid over \((0, 1)^q \).

A similar process can be used to design \( p \) with \( \lambda \) fixed.

VI. DESIGN EXAMPLE

We design codes for a spectral efficiency of 6 bits/s/Hz (3 bits per dimension) over the AWGN channel. We set the alphabet size at \( q = 32 \). Rather than use a 32-PAM constellation, we used a constellation \( A \) whose signals are nonuniformly spaced, designed to approximate a Gaussian random variable. The constellation was obtained by applying a variation of a method suggested by Sun and van Tilborg [14]. This produced a substantial shaping gain in comparison with 32-PAM. The mapping \( \delta \) from the code alphabet is given below, with its elements listed in ascending order using the representation of \( GF(32) \) elements as binary numbers (e.g. \( \delta(00000) = -2.0701, \delta(00001) = -1.7096 \)). Note, however, that our simulations indicate that for a given \( A \), different mappings \( \delta \) typically render the same performance.

\[
\begin{align*}
\lambda &= [0.07144, 0.12592, 0.21075, 0.29689, 0.38474, 0.47523, 0.569, 0.66697, 0.7096, 0.7685, 0.81867, 0.86697, 0.90022, 0.93621, 0.97061, 0.99926] \\
\rho &= [0.0022, 0.0701, 0.1362, 0.21075, 0.29689, 0.38474, 0.47523, 0.569, 0.66697, 0.7096, 0.7685, 0.81867, 0.86697, 0.90022, 0.93621, 0.97061, 0.99926] \\
\end{align*}
\]

We fixed \( \rho = 1 \) and iteratively applied linear programming, first to obtain \( \lambda \) and then, fixing \( \lambda \), to obtain a better \( \rho \). After a few such iterations, we obtained \( \lambda(2, 5, 6, 16, 30) = (0.5768, 0.1498, 0.07144, 0.1045, 0.09752), \rho(5, 6, 7, 8, 20) = (0.09973, 0.02331, 0.5885, 0.1833, 0.1051) \). Interestingly, this code is right-irregular, unlike typical binary LDPC codes. Fig. 2 presents the EXIT chart for the code. Note that the CND curve does not begin at \( IA = 0 \). This is discussed in [2].

Simulation results indicate successful decoding at an SNR of 18.55 dB. The block length was 1.8 \( 10^6 \) symbols, and decoding typically converged after approximately 150–200 iterations. The symbol error rate, after 50 simulations, was approximately \( 10^{-6} \). The Shannon limit at 6 bits/s/Hz for our signal constellation is 18.25 dB, and thus the gap from this limit is 0.3 dB. The unconstrained Shannon limit (i.e. not restricted to any signal constellation) at this rate is 17.99 dB, and thus our gap from this limit is 0.56 dB. This result is well beyond the shaping gap, which at 6 bits/s/Hz is 1.1 dB.

---

3Our Matlab source code for computing EXIT charts is provided at [http://www.eng.tau.ac.il/~bursyn/publications.htm](http://www.eng.tau.ac.il/~bursyn/publications.htm).

4An improvement is sometimes obtained by further requiring \( I_{E,VND}(I, I(0)) > I_{E,CND}(I) + \epsilon \) for \( I \in (0, L) \) where \( L, \epsilon > 0 \).
(a) Initial messages  
(b) Intermediate sum of leftbound messages  
(c) Rightbound messages.

Fig. 1. Empirical distributions of the messages of a (3,6) coset GF(3) LDPC code.

![Graph showing empirical distributions of messages](image)

Fig. 2. The EXIT chart for a code at a spectral efficiency of 6 bits/s/Hz and an SNR of 18.5 dB.

VII. COMPARISON WITH MULTILEVEL CODING (MLC)

We did not find examples in the literature of MLC LDPC codes at a spectral efficiency of 6 bits/s/Hz. Hou et al. [8] presented simulations for MLC over the AWGN channel at a spectral efficiency of 2 bits/s/Hz, using a 4-PAM constellation. The Shannon limit for 4-PAM and at this rate is 5.12 dB (SNR). At a block length of $10^4$ symbols, their best code is capable of transmission at 1 dB of the Shannon limit with a BER of about $10^{-5}$. It is composed of binary LDPC component codes with a maximum left-degree of 15.

We designed edge-distributions for two coset GF(4) LDPC codes at the same spectral efficiency, signal constellation, block-length and BER as [8]. Our best code is capable of transmission within 0.55 dB of the Shannon limit and has a maximum left-degree of 21. Our second code is capable of transmission within 0.8 dB of the Shannon limit. Its maximum left-degree is 6, and is thus lower than the MLC code of [8]. Consequently, it has a lower level of connectivity in its Tanner graph, implying that its slightly better performance was achieved at a comparable decoding complexity.

Note that the increasing the maximum left-degree of the best MLC code of [8] does not guarantee an improvement at short block lengths, because it would also result in an increase in graph connectivity.

REFERENCES