On the Performance of a Multi-Edge Type LDPC Code for Coded Modulation

Harm S. Cronie

Department of Electrical Engineering, Mathematics and Computer Science (EEMCS),
Signals and Systems Group, University of Twente, Enschede, The Netherlands,
e-mail: h.s.cronie@utwente.nl

***Abstract*— We present an error-correcting code which merges error-correction and modulation. The code is an extension of a Low-Density Parity-Check (LDPC) code, and can be viewed as a multi-edge type LDPC code. The symbols of the codewords are from a ternary alphabet, and have a different probability of occurrence. When the code is used on the complex Additive White Gaussian Noise (AWGN) channel, the spectral efficiency is 2 bit/s/Hz. Therefore, the code is suitable for bandwidth-efficient communication. Simulations on the AWGN channel show that the code outperforms several other coded modulation schemes proposed in literature.**

I. Introduction

The use of iterative decoding techniques enables near capacity performance by acceptable computational effort [1], [2], [3]. Low-density parity-check (LDPC) codes are decoded by iterative techniques and can be optimized for performance and complexity [4], [5], [6].

For high spectral efficiencies, coding has to be combined with modulation. Several approaches are reported in literature of which examples are trellis-coded modulation (TCM) [7], bit-interleaved coded modulation (BICM) [8], BICM with iterative decoding [9] and multilevel coding [10]. LDPC codes can also be combined with conventional modulation methods such as quadrature amplitude modulation (QAM). In this case, modulation and coding are more or less considered separately, but there are methods to match the code and detector in order to improve the performance [11].

We use another approach in which we define a code which combines error-correction and modulation. The code can be viewed as an extension of the graph of a LDPC code to include modulation. Hence it can be described by the multi-edge type LDPC code formalism [12]. The two problems of modulation and coding are not solved separately, but jointly. This gives additional performance/complexity trade-offs and may eventually lead to better performance with less computational complexity.

This paper is organized as follows. In section II we describe how encoding of the proposed code is performed. The derivation of bitwise MAP decoding and an iterative message-passing algorithm for decoding are given in section III. The performance of two specific graph structures is simulated for transmission over the additive white Gaussian noise (AWGN) channel and the results are shown in section IV. We end the paper with conclusions in section V.

II. Encoding

The family of codes we consider are encoded as follows. Let s denote a vector of K source bits, $s \in \mathrm{GF}(2)^K$. The encoding operation proceeds in three stages. First, $\mathbf{s}$ is multiplied by a generator matrix $\mathbf{G}$ of a LDPC code:

$$\mathbf{x} = \mathbf{G}^T\mathbf{s}, \tag{1}$$

where $\mathbf{G} \in \mathrm{GF}(2)^{K\times N}$. The design rate of the LDPC code is denoted by r and the block length by N. A parity-check matrix associated to $\mathbf{G}$ is denoted by $\mathbf{H}$.

Second, each of the elements of $\mathbf{x}$ is mapped to the real numbers $\mathcal{R}$ by a binary phase-shift keying (BPSK) mapping:

$$\mathbf{t} = \phi(\mathbf{x}) = \phi\left(\mathbf{G}^T\mathbf{s}\right), \tag{2}$$

where $\phi(\ldots)$ denotes the BPSK mapping and acts on the components of $\mathbf{x}$. For a bit x_i the BPSK mapping is defined as:

$$\phi(x_i) = \begin{cases} 1 & x_i = 0 \\ -1 & x_i = 1. \end{cases} \tag{3}$$

In the final stage of encoding, the codeword $\mathbf{z}$ is obtained by a linear transformation:

$$\mathbf{z} = \mathbf{A}\mathbf{t}, \tag{4}$$

where $\mathbf{A} \in \mathcal{R}^{N_z\times N}$ is a sparse matrix consisting of 0s and 1s. We denote the length of the codewords by N_z and the set of all codewords by $\mathcal{C}$. We limit ourselves to $\mathbf{A}$ matrices with d_z ones in each row and in this paper $d_z = 2$. Hence the ith coordinate of $\mathbf{z}$, which is denoted by z_i, is equal to the sum of two BPSK mapped bits:

$$z_i = \phi(x_{j1}) + \phi(x_{j2}), \tag{5}$$

where x_{j1} and x_{j2} are two different elements of $\mathbf{x}$. Thus, z_i is equal to an element of the following set:

$$\mathcal{X} = \{-2, 0, 2\}. \tag{6}$$

Since a LDPC code is used and the prior distribution of the source bits is assumed to be uniform, two different elements of $\mathbf{x}$ are approximately independent of each other. Consequently, we can assign the following probabilities to the elements of $\mathcal{X}$:

x_{j1}	x_{j2}	z_i	$P(z_i)$
0	0	2	0.25
0/1	1/0	0	0.5
1	1	-2	0.25

Note that two combinations of x_{j1} and x_{j2} map to the same z_i, which implies that it is possible that two different source vectors map to the same codeword. However, to guarantee a good performance, it is sufficient that the probability that this happens decays rapidly for increasing N_z. The distance properties of the LDPC code will play a role in this. In [13] we show that equation 5 can be interpreted in a different way, which shows that the problem does not have to be relevant for transmission over the AWGN channel.

A signal constellation for the complex AWGN channel can be generated by a summation of different pairs of BPSK mapped bits for each dimension. As mentioned before, we limit ourselves to $d_z = 2$ and the resulting constellation is shown in figure 1. Each constellation symbol is labeled with its probability.

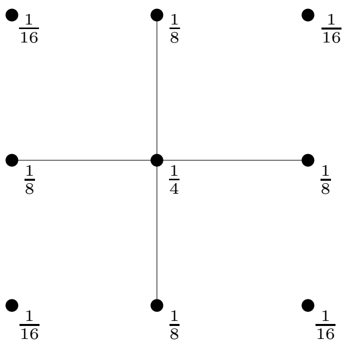

Fig. 1. The signal constellation for $d_z = 2$.

We conclude this section with a summary of the parameters of the code. The rate R of the code is defined as:

$$R = \frac{K}{N_z} = \frac{rN}{N_z}, \tag{7}$$

where r and N are the design rate and block length of the LDPC code, respectively. The energy expended per channel use is defined as:

$$E_s = \mathcal{E}\{z_i^2\}, \tag{8}$$

where $\mathcal{E}$ denotes the expectancy operator. Finally, the energy expended per information bit is given by:

$$E_b = \frac{N_z E_s}{K}. \tag{9}$$

III. Decoding

We consider the transmission of a codeword $\mathbf{z}$ over the extended AWGN channel:

$$\mathbf{y} = \mathbf{z} + \mathbf{n}, \tag{10}$$

where $\mathbf{y}$ is the channel output corresponding to N_z consecutive channel uses and $\mathbf{n}$ is a Gaussian noise vector.

A. Bitwise MAP decoding

The bitwise maximum a posteriori (MAP) decision rule of a bit x_i associated to a codeword is given by:

$$\hat{x}_i = \underset{x' \in 0,1}{\text{argmax}}\ P(x_i = x'|\mathbf{y}). \tag{11}$$

$P(x_i|\mathbf{y})$ can be obtained by marginalization:

$$P(x_i|\mathbf{y}) = \sum_{\substack{\mathbf{z},\sim x_i: \\ \mathbf{z}\in\mathcal{C},\mathbf{Hx}=\mathbf{0}}} P(\mathbf{x},\mathbf{z}|\mathbf{y}), \tag{12}$$

where $\sim x_i$ denotes the summary or not-sum operator. With Bayes law, we rewrite this equation as:

$$\begin{aligned} P(x_i|\mathbf{y}) &= \sum_{\substack{\mathbf{z},\sim x_i: \\ \mathbf{z}\in\mathcal{C},\mathbf{Hx}=\mathbf{0}}} P(\mathbf{x}|\mathbf{y},\mathbf{z})P(\mathbf{z}|\mathbf{y}) \\ &= \sum_{\substack{\mathbf{z},\sim x_i: \\ \mathbf{z}\in\mathcal{C},\mathbf{Hx}=\mathbf{0}}} P(\mathbf{x}|\mathbf{y},\mathbf{z})\frac{P(\mathbf{y}|\mathbf{z})P(\mathbf{z})}{P(\mathbf{y})} \\ &= \frac{1}{P(\mathbf{y})}\sum_{\substack{\mathbf{z},\sim x_i: \\ \mathbf{z}\in\mathcal{C},\mathbf{Hx}=\mathbf{0}}} P(\mathbf{x}|\mathbf{z})P(\mathbf{y}|\mathbf{z})P(\mathbf{z}), \end{aligned} \tag{13}$$

where we have used that $P(\mathbf{x}|\mathbf{y},\mathbf{z}) = P(\mathbf{x}|\mathbf{z})$. $P(\mathbf{x}|\mathbf{z})$ is equal to 0 in case $\mathbf{A}\phi(\mathbf{x}) \neq \mathbf{z}$. Next, consider a particular $\mathbf{z}$ and assume that there is only one $\mathbf{x}$ for which $\mathbf{A}\phi(\mathbf{x}) = \mathbf{z}$. In this case $P(\mathbf{x}|\mathbf{z}) = 1$ which means that $P(\mathbf{x}|\mathbf{z})$ can be dropped in equation 13. Under the summation sign there are two conditions that have to be satisfied. First, $\mathbf{z} \in \mathcal{C}$, which implies that for a vector $\mathbf{x}$, $\mathbf{z} = \mathbf{A}\phi(\mathbf{x})$. Second, $\mathbf{x}$ must be a codeword of the LDPC code, which implies that $\mathbf{Hx} = \mathbf{0}$. These two conditions can be dropped under the summation sign by use of the so-called truth function:

$$\mathbb{1}(A) = \begin{cases} 0 & A \text{ is false} \\ 1 & A \text{ is true.} \end{cases} \tag{14}$$

We rewrite equation 13 as:

$$P(x_i|\mathbf{y}) = \frac{1}{P(\mathbf{y})}\sum_{\mathbf{z},\sim x_i} P(\mathbf{y}|\mathbf{z})P(\mathbf{z})\mathbb{1}\Big(\mathbf{A}\phi(\mathbf{x}) = \mathbf{z}\Big)\mathbb{1}\Big(\mathbf{Hx} = \mathbf{0}\Big). \tag{15}$$

Since the AWGN channel is memoryless, $P(\mathbf{y}|\mathbf{z})$ is a separable function. Furthermore, the remaining terms in equation 15 are also separable functions. We rewrite:

$$P(x_i|\mathbf{y}) = \frac{1}{P(\mathbf{y})}\sum_{\mathbf{z},\sim x_i}\Bigg(\prod_{k=1}^{N_z} P(y_k|z_k)\prod_{k=1}^{N_z} P(z_k) \cdot \prod_{m=1}^{N-K}\mathbb{1}\Big(\mathbf{h}_m^T\mathbf{x} = 0\Big)\prod_{m=1}^{N_z}\mathbb{1}\Big(\mathbf{a}_m^T\phi(\mathbf{x}) = z_m\Big)\Bigg), \tag{16}$$

where $\mathbf{h}_m^T$ and $\mathbf{a}_m^T$ denote the mth row in $\mathbf{H}$ and $\mathbf{A}$, respectively.

B. Factor graphs and the sum-product algorithm

A factor graph [14] is a convenient graphical representation of an equation which is factored in several parts. In our case, the variable nodes correspond to the elements of $\mathbf{x}$ and $\mathbf{z}$. We refer to the variable nodes associated to $\mathbf{x}$ and $\mathbf{z}$ as *bit nodes* and *z nodes*, respectively. Each factor node corresponds to one of the terms in equation 16. An example of a factor graph of the proposed code is shown in figure 2. The parameters of the code are $N_z = 8$, $N = 16$ and $K = 8$. In the figure we have drawn a single factor node for $P(y_k|z_k)$ and $P(z_k)$. Note that the part on the right corresponds to the Tanner graph [15] of the LDPC code.

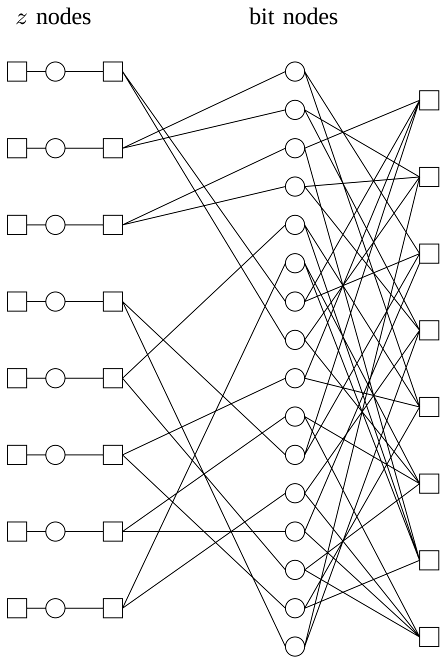

Fig. 2. An example of a factor graph of the proposed code.

In case the factor graph has a tree structure, the sum-product algorithm can be applied to compute exact bitwise MAP probabilities in a finite number of iterations. The principle behind the sum-product algorithm is a massive application of the distributive law [16].

In our case, the factor graph is not a tree, which also applies to the factor graph of LDPC codes. We can still apply an iterative version of the sum-product algorithm and obtain approximate bitwise MAP probabilities. The algorithm is defined on the factor graph and proceeds by an exchange of messages similar to the decoding of LDPC codes. We refer to [17], [5] and [18] for an overview of the decoding of LDPC codes.

C. Message-passing algorithm

The message-passing algorithm we use to decode consists of three phases: an initialization, a factor node update and a variable node update. The latter two are repeated until convergence of the algorithm. In the next subsections, we describe these three phases. We use the following notation for the messages passed. Each message to or from a variable node in the lth iteration consists of a set defining a probability mass function (p.m.f.) associated to the variable node. The messages that are sent from variable nodes to factor nodes are denoted by $m_{vc}^{(l)}$ and messages that are sent from factor nodes to variable nodes are denoted by $m_{cv}^{(l)}$. We use $[\cdot]_k$ to index the kth element in a set. For a bit node x_k we have:

$$\begin{aligned}[m_{vc}^{(l)}]_1 &= P(x_k = 0|\ldots) \\ [m_{vc}^{(l)}]_2 &= P(x_k = 1|\ldots), \end{aligned} \tag{17}$$

and for a z node z_k we have:

$$\begin{aligned}[m_{vc}^{(l)}]_1 &= P(z_k = -2|\ldots) \\ [m_{vc}^{(l)}]_2 &= P(z_k = 0|\ldots) \\ [m_{vc}^{(l)}]_3 &= P(z_k = 2|\ldots). \end{aligned} \tag{18}$$

The following messages are passed during the three phases of the algorithm:

1) Initialization: The algorithm is initialized by a message pass from variable nodes to factor nodes. A z node z_k computes the *a posteriori* p.m.f. associated to the z node conditioned on the channel output:

$$P(z_k|y_k) \propto P(y_k|z_k)P(z_k), \tag{19}$$

where $P(y_k|z_k)$ is the likelihood of z_k and $P(z_k)$ is the prior on z_k. For $d_z = 2$, we have shown in section II that $P(z_k)$ is given by:

$$P(z_k) = \begin{cases} 0.25 & z_k = -2 \\ 0.5 & z_k = 0 \\ 0.25 & z_k = +2. \end{cases} \tag{20}$$

For transmission over the AWGN channel $P(y_k|z_k)$ is given by:

$$P(y_k|z_k) = \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(y_k - z_k)^2}{2\sigma_n^2}}, \tag{21}$$

where σ_n^2 is the noise variance. Summarized, the z nodes send the following messages during initialization:

$$[m_{vc}^{(0)}]_i = P(z_k = [\mathcal{X}]_i|y_k), \quad \text{for } i = 1, 2 \text{ and } 3 \tag{22}$$

where $\mathcal{X}$ was defined in section II.

Since the prior bit probabilities are assumed to be uniform, the bit nodes send the following messages during initialization:

$$[m_{vc}^{(0)}]_1 = P(x_k = 0) = 0.5 \qquad [m_{vc}^{(0)}]_2 = P(x_k = 1) = 0.5. \tag{23}$$

2) Factor node update: At a z factor node, we send messages to the bit nodes based on the other connected bit nodes and the connected z node. Since we are only interested in the bit nodes, the messages send to the z nodes are not described. Let x_i be a bit node for which we compute a message, x_i' the other connected bit node and z_i the connected z node. The message is computed as:

$$[m_{cv}^{(l)}]_1 = \frac{1}{Z} P(x_i' = 0)P(z_i = 2) + P(x_i' = 1)P(z_i = 0)$$

$$[m_{cv}^{(l)}]_2 = \frac{1}{Z} P(x_i' = 0)P(z_i = 0) + P(x_i' = 1)P(z_i = -2), \tag{24}$$

where the probabilities on the right handside are obtained from the messages received from variable nodes x_i' and z_i. Furthermore, Z is a normalization constant which is chosen in such a way that:

$$[m_{cv}^{(l)}]_1 + [m_{cv}^{(l)}]_2 = 1. \tag{25}$$

We note that at the z factor nodes, there is an efficient way to compute the messages for $d \geq 2$. Details are described in [13].

At the bit factor nodes we use Gallager's form to compute the outgoing messages [18]. The outgoing message send to a bit node x_i is computed as:

$$[m_{cv}^{(l)}]_1 = \frac{1 + \prod_{j \neq i}(1 - 2[m_{vc}^{l-1}]_2)}{2}$$
$$[m_{cv}^{(l)}]_2 = \frac{1 - \prod_{j \neq i}(1 - 2[m_{vc}^{l-1}]_2)}{2}, \tag{26}$$

where j loops over each connected bit node excluding x_i.

3) Variable node update: The outgoing message send to a bit node x_i is computed as:

$$[m_{vc}^{(l)}]_1 = \frac{1}{Z} \prod_{j \neq i} [m_{cv}^{(l-1)}]_1$$
$$[m_{vc}^{(l)}]_2 = \frac{1}{Z} \prod_{j \neq i} [m_{cv}^{(l-1)}]_2, \tag{27}$$

where Z is a normalization constant which is chosen in such a way that:

$$[m_{vc}^{(l)}]_1 + [m_{vc}^{(l)}]_2 = 1. \tag{28}$$

Again j loops over each connected bit node excluding x_i.

IV. RESULTS

In this section we present simulation results for two code structures. In the first subsection, we give some specific details about the construction of the codes and in the second subsection we present simulation results for transmission over the AWGN channel.

A. Code construction and parameters

So far, we have not specified any details about the structure of the graph. We define a family of graphs by specification of the component LDPC code. Furthermore, each bitnode is connected to one z factor node and each z factor node has degree $d_z + 1 = 3$. Degree distributions are a convenient way to specify an ensemble of LDPC codes and we use the normalized degree distributions from a node perspective [19]:

$$L(x) = \sum_i L_i x^i \qquad R(x) = \sum_i R_i x^i \tag{29}$$

where L_i and R_i are the fraction of variable nodes of degree i and check nodes of degree i, respectively. In this paper we use one regular degree distribution and one irregular degree distribution. We have not used an optimization technique such as

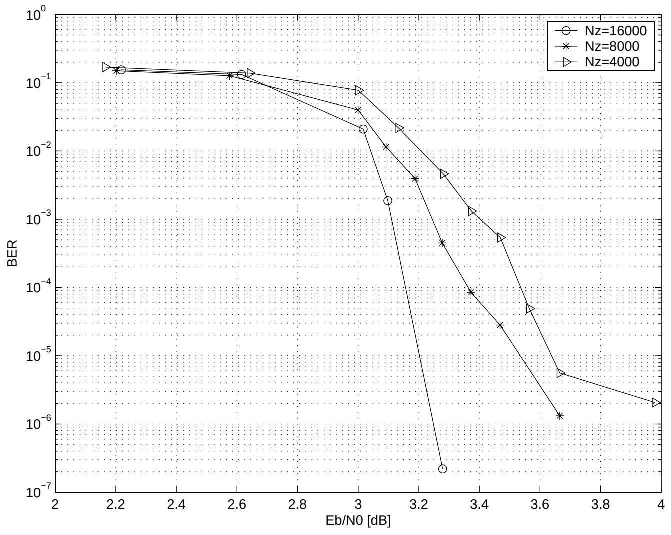

Fig. 3. The BER of code I for transmission over the AWGN channel

density evolution, but selected two pairs of distributions with low variable node degrees by trial and error. The polynomials and other code parameters are given in the following table:

Code	r	R	$L(x)$	$R(x)$
I	0.5	1	$0.35x^2 + 0.65x^3$	$0.82x^4 + 0.18x^{11}$
II	0.5	1	x^2	x^4

The $\mathbf{A}$ matrix is constructed at random, but we made sure that each row has two ones and all rows are different.

B. Results for transmission over the AWGN channel

We consider transmission over the AWGN channel for code I first. Transmission takes place at a rate of 1 bit/channel use. Figure 3 shows the bit error rate (BER) for several signal to noise ratios (E_b/N_0) with N_z equal to 16000, 8000 and 4000. We have set the maximum number of iterations to 800, but stop whenever a word $\mathbf{x}$ is found which satisfies $\mathbf{Hx} = \mathbf{0}$. For BERs $\leq 10^{-3}$, convergence usually takes place within 40 iterations. Code I achieves a BER of 10^{-5} at an E_b/N_0 of 3.2 dB for $N_z = 16000$. The Shannon limit for the unconstrained AWGN channel is at 1.76 dB and if we consider a BER of 10^{-5} reliable, the distance to the Shannon limit is 1.4 dB. When we compare the results to other coded modulation schemes we observe the following. In [9] a BICM scheme with iterative decoding and a 8-PSK constellation is used. For 4000 information bits a BER of 10^{-5} is achieved at 4.5 dB. From figure 3 we see that code I with $N_z = 4000$ requires an E_b/N_0 of 3.65 dB to achieve the same BER. Our signal constellation contains one extra symbol in comparison to a 8-PSK constellation, but the results are better. In [20] a turbo trellis-coded modulation scheme is used with a 8-PSK constellation. This method has a similar performance as the coded modulation method described in this paper.

The simulation results for code II are shown in figure 4. Since the degree of all bit nodes is two, the code has many low-weight codewords. This causes the message-passing

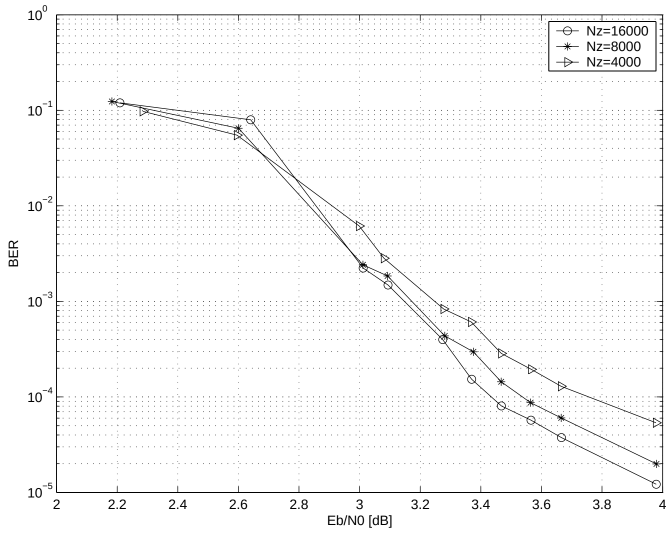

Fig. 4. The BER of code II for transmission over the AWGN channel

algorithm to converge to low-weight codewords, which results in undetected errors. Code I performs much better. However, due to the low degrees of the variable nodes and check nodes, the complexity of code II is lower. Furthermore, the threshold of code II is lower (2.6 dB).

V. Conclusions and future research

We have proposed a multi-edge type LDPC code which combines modulation and error-correction. A message-passing algorithm is derived, which computes approximate bit probabilities. We have constructed codes with different block lengths from two different pairs of degree polynomials. The empirical performance for transmission over the AWGN channel was simulated and results compare well against other schemes described in literature. We have not used any optimized degree distributions, but plan to do this in future research. Finally, we would like to point out that the use of larger d_z could be beneficial to obtain a signal shaping gain.

References

[1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[2] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar. 1996.

[3] S.-Y. Chung, J. Forney, G. D., T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, pp. 58–60, Feb. 2001.

[4] R. G. Gallager, "Low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[5] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, pp. 457–458, Mar. 1997.

[6] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.

[7] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. 28, pp. 55–67, Jan. 1982.

[8] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, pp. 927–946, May 1998.

[9] X. Li, A. Chindapol, and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding and 8 PSK signaling," *IEEE Trans. Commun.*, vol. 50, pp. 1250–1257, Aug. 2002.

[10] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, "Multilevel codes: theoretical concepts and practical design rules," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1361–1391, July 1999.

[11] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, pp. 670–678, Apr. 2004.

[12] T. J. Richardson and R. L. Urbanke, "Multi-edge type LDPC codes." [Online]. Available: http://lthcwww.epfl.ch/papers/multiedge.ps

[13] H. S. Cronie, "Sparse graph codes for multilevel modulation with signal shaping," submitted to ISIT 2005.

[14] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[15] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, Sept. 1981.

[16] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Trans. Inform. Theory*, vol. 46, pp. 325–343, Mar. 2000.

[17] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.

[18] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[19] T. J. Richardson and R. L. Urbanke, "modern coding theory." [Online]. Available: http://lthcwww.epfl.ch/mct/index.php

[20] S. Y. Le Goff, "Signal constellations for bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 49, pp. 307–313, Jan. 2003.