Design of Unequal Error Protection LDPC Codes for Higher Order Constellations

Neele von Deetzen
School of Engineering and Science
International University Bremen (IUB), Germany
Email: n.vondeetzen@iu-bremen.de

Sara Sandberg
Dept. of Computer Science and Electrical Engineering
Luleå University of Technology, Sweden
Email: sara.sandberg@ltu.se

***Abstract*— We present an optimization method for unequal error protection (UEP)-LDPC codes with higher order constellations. By modifying the density evolution algorithm under the Gaussian approximation, we propose a flexible code design algorithm for a variable number of protection classes and arbitrary modulation schemes with Gray mapping. Our results show that appropriate code design for higher order constellations reduces the overall bit-error rate. Furthermore, the influence on the UEP capability of the code, that is, the difference in bit-error rate between the protection classes, is investigated.**

I. Introduction

Coded modulation is a well-known technique which optimizes the coding scheme given the modulation in order to improve the performance of transmission systems [1], [2], [3], [4]. Usually, the modulation alphabet is successively partitioned into smaller subsets, where each partitioning level is assigned a label. These labels are protected by separate channel codes with certain protection capabilities. The codes have to be designed carefully depending on the modulation scheme and its partitioning or labeling strategy. According to [5], the optimal way of designing the codes is to match the different code rates to the capacities of the partitioning steps. This means that, for a given signal-to-noise ratio (SNR) and given modulation scheme and partitioning, the code rates of the single codes are fixed. However, there are also other design approaches with similar results, [6], [7]. The corresponding channel codes can be block codes, convolutional codes, or concatenated codes.

In our approach, we will use low-density parity-check (LDPC) codes, which were presented by Gallager in [8]. LDPC codes are block codes with a sparse parity-check matrix H that can be conveniently described through a graph commonly called a Tanner graph [9]. Such a graphical representation facilitates a decoding algorithm known as the message-passing algorithm. For more details on message-passing decoding, the reader is referred to an introduction by Kschischang *et al.* [10]. Optimization of LDPC codes as separate codes for each level in multilevel coding has been investigated in [11] amongst others.

In this paper, only one code is used for all levels instead of separate codes. A longer code (with better performance) can then be used while keeping the delay fixed. The task is to design certain local properties of the code to match the higher order constellations and assign bit positions of the modulation scheme to the codeword bits. Achieving local properties in the codeword may be done by designing the variable and/or check node degree distribution of the code in an irregular way [12], [13], [14], [15]. The connection degree of the variable nodes affects the bit-error rate (BER). The message bits are divided into several classes depending on the connection degree and each class has different BER after decoding, that is, the code provides unequal error protection (UEP). In [16], bits from the least protected modulation level are mapped to variable nodes with the best protection, that is, the variable nodes with highest connection degrees. No other optimization is performed. In [7], the different amount of protection for each modulation level is taken into account in the initialization of the density evolution algorithm that is employed to optimize the degree distribution of the code. In our approach, we design UEP-LDPC codes while accounting for the unequal error protection which is already inherent in modulation. Regarding the modulation schemes, any conventional scheme like M-QAM or M-PSK as well as more complex schemes called hierarchical constellations [17] with Gray labeling may be used.

The paper is organized as follows. Section II presents the overall system model, while models for the modulator are given in Section III. Section IV contains the main part of this paper which includes a general description of irregular LDPC codes and the standard code optimization as well as extensions for UEP. We also explain the optimization of the degree distribution for higher order constellations and give an algorithm for the code design. In Section V, some simulation results are discussed.

II. System Model

In this section, we describe the system model of the transmission scheme. Usually, in multilevel coding, the information bits are demultiplexed into l_m parallel streams, where $2^{l_m} = M$ is the constellation size of the modulation scheme. The different bit streams are encoded separately and are assigned to the l_m partitioning steps of a modulation scheme. In our case, the independent and identically distributed (i.i.d.) source bits are not multiplexed but each bit is assigned to one of $N_c - 1$ protection classes which are usually defined by the source coding unit and do not have to be of equal size. We apply only

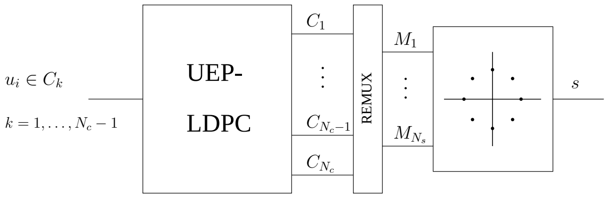

Fig. 1. UEP-LDPC coded modulation scheme.

one code $\mathcal{C}$, providing N_c protection classes at its output (see Fig. 1), where all parity bits correspond to the least protected class C_{N_c}. The bits of the protection classes are remultiplexed and assigned to certain bit positions of the modulator, which correspond to modulation classes $M_1, \ \ldots, \ M_{N_s}$. The bit assignment will be described in Section IV-A. In the following we assume an additive white Gaussian noise (AWGN) channel with noise variance σ^2.

III. Modulation

Let us assume a modulation scheme with $M = 2^{l_m}$ signal points, labeled by binary vectors $\mathbf{d} = (d_{l_m - 1}, \ \ldots d_1, \ d_0)$. In order to design codes for higher order constellations, we investigate the error probabilities of the individual bits. The example of 8-PSK is chosen here, but the scheme can also be designed for any other constellation.

Using the union bound, the approximate symbol-error rate expression for 8-PSK is given as [18],

$$P_{s,8-PSK} = \text{erfc}\left(\sqrt{\frac{3E_b}{N_0}} \sin\frac{\pi}{8}\right) , \tag{1}$$

where erfc is the complementary error function. The individual and average bit-error probabilities depend on the partitioning and labeling strategy. We will only consider Gray labeling, since it leads to the lowest overall bit-error probability. Furthermore, the bits are almost independent of each other which is important for the LDPC decoder performance.

For Gray labeling, a symbol error typically results in only one bit error and, thus, one can assume that the average bit-error rate is $\tilde{P}_b \approx P_s / \log_2(M)$. The expressions for the bit-error probabilities of the individual bits in the symbol are

$$P_{b,d_0} \approx \frac{1}{2} \cdot \text{erfc}\left(\sqrt{\frac{3E_b}{N_0}} \sin\frac{\pi}{8}\right) , \tag{2}$$

$$P_{b,d_1} = P_{b,d_2} \approx \frac{1}{4} \cdot \text{erfc}\left(\sqrt{\frac{3E_b}{N_0}} \sin\frac{\pi}{8}\right) . \tag{3}$$

From these different bit-error probabilities, one can determine equivalent noise variances of the single bit positions corresponding to the case of BPSK. We define the noise vector $\underline{\sigma}^2 = [\sigma_1^2 \ \ldots \ \sigma_{N_s}^2]$ to be a vector that contains the equivalent noise variances for each separate bit-error rate ordered with the lowest variance first. We assume that there are N_s distinct equivalent noise variances, where $N_s \leq l_m$. The equivalent noise variances may be calculated from the individual bit-error rates by

$$\sigma_j^2 = \frac{1}{2\left(\text{erfc}^{-1}(2P_{b,d_j})\right)^2} . \tag{4}$$

Note that these expressions are obtained by applying the union bound. The approximations are assumed to be appropriate for our purposes but can be replaced by more exact formulas. In the following, we assume that N_s equivalent BPSK channels are employed instead of the higher order constellation channel. We claim that this approximation meets our requirements since the system employs Gray mapping.

IV. UEP-LDPC Codes

As a channel code, we choose a UEP-LDPC code. There are different methods for achieving UEP with LDPC codes, the probably most obvious one is puncturing a certain amount of the code bits before modulation. The receiver does not have any knowledge about these bits and assumes all signals of the input alphabet with equal probability. Two other possibilities for obtaining UEP were presented in [14] and [15]. Both approaches use irregular LDPC codes and optimize the irregularities of the code in order to obtain several classes of protection within the codeword. More precisely, the authors in [14] optimize the irregular variable node (also called bit node) degree distribution while keeping the check node degree distribution fixed, whereas in [15], the check node degree distribution is adapted, keeping the variable node degree distribution fixed. We will follow the approach from [14]. The next section gives a general description of UEP-LDPC codes by considering degree distributions.

A. General Description

LDPC codes are block codes with a sparse parity-check matrix H of dimension $(n-k) \times n$, where $R = k/n$ denotes the code rate and k and n are the lengths of the information word and the codeword. The codes can be represented as a bipartite graph, called Tanner graph. The graph consists of two types of nodes, variable nodes and check nodes, which correspond to the bits of the codeword and to the parity-check constraints, respectively. A variable node is connected to a check node if the bit is included in the parity-check constraint. For regular LDPC codes, all variable nodes and check nodes have fixed variable node degree and check node degree, respectively. However, irregular LDPC codes are known to approach capacity closer than regular LDPC codes. The irregular variable node and check node degree distributions may be defined by the polynomials

$$\lambda(x) = \sum_{i=2}^{d_{v_{max}}} \lambda_i x^{i-1} \text{ and } \rho(x) = \sum_{i=2}^{d_{c_{max}}} \rho_i x^{i-1}$$

where $d_{v_{max}}$ and $d_{c_{max}}$ are the maximum variable and check node degree [19]. The degree distributions describe the proportion of edges connected to nodes with a certain degree.

In order to optimize the degree distribution of an irregular LDPC code, the decoding behavior has to be investigated. Using a message-passing algorithm, the messages along the edges of the graph are updated iteratively. The mutual information messages at the input of a variable node and a check node at iteration l can be computed by means of density evolution using the Gaussian approximation [20] to be

$$x_u^{(l-1)} = 1 - \sum_{j=2}^{d_{c_{max}}} \rho_j J((j-1)J^{-1}(1-x_v^{(l-1)})) \ , \quad (5)$$

$$x_v^{(l)} = \sum_{i=2}^{d_{v_{max}}} \lambda_i J(\frac{2}{\sigma^2} + (i-1)J^{-1}(x_u^{(l-1)})) \ , \quad (6)$$

with $J(\cdot)$ computing the mutual information $x = J(m)$ by

$$\begin{aligned} J(m) &= 1 - \mathbb{E}\{log_2(1+e^{-z})\} \quad (7) \\ &= 1 - \frac{1}{\sqrt{4\pi m}} \int_{\mathbb{R}} log_2(1+e^{-z}) \cdot e^{-\frac{(z-m)^2}{4m}} dz \end{aligned}$$

for a consistent Gaussian random variable $z \sim \mathcal{N}(m, 2m)$. These update rules are valid only when all bits belong to one modulation class with noise variance σ^2.

For the case of UEP-LDPC codes, we follow the approach from [14] and define an overall check node degree distribution and different variable node degree distributions for the N_c protection class, i.e.,

$$\lambda^{(C_k)}(x) = \sum_{i=2}^{d_{v_{max}}} \lambda_i^{(C_k)} x^{i-1} \quad \text{for} \quad k = 1 \dots N_c \ . \quad (8)$$

Since the variable node degree distributions give proportions of edges connected to variable nodes of certain degrees, the constraint

$$\sum_{k=1}^{N_c} \sum_{i=2}^{d_{v_{max}}} \lambda_i^{(C_k)} = 1 \quad (9)$$

must be fulfilled. Different variable node degree distributions lead to a modified update rule for the messages from variable nodes to check nodes

$$x_v^{(l)} = \sum_{k=1}^{N_c} \sum_{i=2}^{d_{v_{max}}} \lambda_i^{(C_k)} J(\frac{2}{\sigma^2} + (i-1)J^{-1}(x_u^{(l-1)})) \ . \quad (10)$$

The update rule for the messages from check nodes to variable nodes stays the same since the check node degree distribution is constant.

This paper considers the design of UEP-LDPC codes for higher order constellations, where the individual bits in the symbol may have different error probabilities. The aim of the code design is to reduce the overall BER by taking these different error probabilities into account. The design algorithm should also give the possibility to trade overall BER for UEP capability. The natural way of assigning bits from modulation classes to protection classes to achieve UEP, is to use the best protected bits from the modulation, that is, modulation class M_1, for protection class C_1 and continue like that until all bits have been assigned to a protection class. However, this assignment is not necessarily expected to give a degree distribution with the lowest possible threshold, where the threshold is defined as the lowest E_b/N_0 for which density evolution converges. However, as is discussed later on, there is always a tradeoff between a low threshold and good UEP capability. By introducing different variable node degree distributions also for each modulation class, linear programming may be used to assign bits from the modulation classes to the protection classes.

B. Notations

We consider a UEP-LDPC code with N_c protection classes. The proportions of each class are given by the normalized lengths of each class corresponding to the information bits, $\underline{\alpha} = [\alpha_1, \ \dots, \ \alpha_{N_c-1}]$. The proportion distribution of the bits in the codeword belonging to the protection classes is given by $\underline{p} = [\alpha_1 R, \ \dots, \ \alpha_{N_c-1}R, \ (1-R)]$. N_s is the number of different bit-error rates for the bits in a symbol and we will describe the bits with a distinct bit-error rate as belonging to one modulation class. $\underline{\beta} = [\beta_1, \ \dots, \beta_{N_s}]$ defines the proportion of bits that belongs to each modulation class.

The vector $\underline{\lambda}$ contains the overall variable node degree distribution, both for different protection classes and different modulation classes. Let $\lambda_{j,i}^{(C_k)}$ be the proportion of edges connected to variable nodes of degree i that belong to modulation class M_j and protection class C_k. Define $\underline{\lambda}_j^{(C_k)} = [\lambda_{j,2}^{(C_k)}, \dots, \lambda_{j,d_{v_{max}}}^{(C_k)}]^T$ and $\underline{\lambda} = \left[\underline{\lambda}_1^{(C_1)T}, \dots, \underline{\lambda}_1^{(C_{N_c})T}, \ \dots, \ \underline{\lambda}_{N_s}^{(C_1)T}, \dots, \underline{\lambda}_{N_s}^{(C_{N_c})T}\right]^T$, where $(\cdot)^T$ denotes the transpose. $\underline{\lambda}_j^{(C_k)}$ is a $(d_{v_{max}} - 1 \times 1)$ vector and $\underline{\lambda}$ is a vector of size $((d_{v_{max}} - 1) \cdot N_c \cdot N_s \times 1)$. The vector $\underline{\rho} = [\rho_2, \dots, \rho_{d_{c_{max}}}]^T$ describes the check node degree distribution. For later purposes, we also define $\underline{1/d_v} = [1/2, \ 1/3, \ \dots, \ 1/d_{v_{max}}]^T$, $\underline{1/d_c} = [1/2, \ 1/3, \ \dots, \ 1/d_{c_{max}}]^T$ and $\underline{1}$ to be an all-ones vector of appropriate length.

C. Optimization of the Degree Distribution for HOC

For higher order constellations, the update rule (10) has to be modified to take different noise variances for different variable nodes into account. Similar to [7], the update rule may be written

$$x_v^{(l)} = \sum_{k=1}^{N_c} \sum_{j=1}^{N_s} \sum_{i=2}^{d_{v_{max}}} \lambda_{j,i}^{(C_k)} J(\frac{2}{\sigma_j^2} + (i-1)J^{-1}(x_u^{(l-1)})) \ . \quad (11)$$

Equations (5) and (11) can now be combined to yield the mutual information evolution of the LDPC code

$$x_v^{(l)} = F(\underline{\lambda}, \underline{\rho}, \underline{\sigma}^2, x_v^{(l-1)}) \ . \quad (12)$$

If $x_v^{(l)} > x_v^{(l-1)}$ for any $x_v^{(l-1)}$, then $\underline{\lambda}$ and $\underline{\rho}$ describe a code for which density evolution converges for the noise variance vector $\underline{\sigma}^2$.

UEP capability may be obtained by optimizing each protection class after another by linear programming, starting with the best protected class and fixing the degree distributions of the already optimized classes during the optimization of the following classes, [14]. It is well-known that a higher connectivity of a variable node leads to better protection. Thus, the optimization target is to find a variable node degree distribution for the whole code that maximizes the average variable node degree of the class being optimized. Thus, the target function for protection class C_k can be formulated as

$$\max_{\underline{\lambda}} \sum_{j=1}^{N_s} \sum_{i=2}^{d_{v_{max}}} \lambda_{j,i}^{(C_k)} . \tag{13}$$

This target function results in a degree distribution with UEP capability, but the only requirement on the assignment of the code bits to the modulation classes is that density evolution must converge for the given degree distribution. In order to achieve UEP, one would assign as many bits as possible from better modulation classes to the protection class being optimized. This can be done by introducing a scaling factor k_j for the modulation classes, where the only requirement is $k_1 > k_2 > \ldots > k_{N_s} > 0$. For simplicity, k_j might be chosen as $k_j = N_s - j + 1$. The factor k_j will appear later in the target function (16) of the algorithm where it has the effect that the linear programming algorithm, if possible while fulfilling all constraints, will use modulation classes with low noise variance for the best protected classes.

When designing good LDPC code ensembles, the stability condition which ensures convergence of the density evolution for mutual information close to one should be fulfilled [19]. The stability condition gives an upper bound on the number of degree-2 variable nodes. For a BPSK scheme, where all bits are affected by the same noise variance, we have [19]

$$\frac{1}{\lambda'(0)\rho'(1)} > e^{-r} = \int_{\mathbb{R}} P_0(x) e^{-\frac{x}{2}} dx = e^{-\frac{1}{2\sigma^2}} \tag{14}$$

with $P_0(x)$ being the message density corresponding to the received values and $\lambda'(x)$ and $\rho'(x)$ being the derivatives of the degree polynomials. It is straightforward to see that $\lambda'(0) = \sum_{j=1}^{N_s} \sum_{k=1}^{N_c} \lambda_{j,2}^{(C_k)}$ and $\rho'(1) = \sum_{m=2}^{d_{c_{max}}} \rho_m \cdot (m-1)$. In our case, the bits are affected by channel noise with different variances σ_j^2 (see (4)) and, thus, different densities. We use the average density, which is given by utilizing the modulation class proportions $\underline{\beta}$,

$$e^{-r} = \int_{\mathbb{R}} \sum_{j=1}^{N_s} \beta_j \cdot P_{0,j}(x) e^{-\frac{x}{2}} dx = \sum_{j=1}^{N_s} \beta_j \cdot e^{-\frac{1}{2\sigma_j^2}} . \tag{15}$$

We are very well aware that this is an approximation but assume appropriateness for the ensemble of code constructions with given $\underline{\beta}$.

D. Optimization Algorithm

The optimization algorithm proposed here is a modification of the hierarchical optimization algorithm presented in [14] for higher order constellations. The optimization is performed at $E_b/N_0 = \delta + \epsilon$ (this will be the threshold of the optimized code), where δ is the lowest possible threshold in dB for the given $\underline{\rho}$ and $d_{v_{max}}$, and ϵ is the offset from the lowest threshold that gives freedom in the choice of $\underline{\lambda}$ to enable design of a UEP code.

The algorithm can be divided into two parts, *global optimization* and *local optimization*. In the global optimization, the linear programming is executed class after class for a given E_b/N_0. In the local optimization, $\underline{\lambda}$ is optimized to maximize the scaled average variable node degree of class C_k while using the best possible modulation class, assuming that classes $C_1, \ldots, C_{k-1}$ have already been optimized. In order to find a maximum average degree, the algorithm starts by setting the minimum variable node degree to some maximum value, conveniently the maximum variable node degree of the code, and tries to find a solution. In case of failure, the minimum variable node degree is successively reduced until the algorithm succeeds in finding an overall degree distribution which fulfills the constraints.

The global optimization can be stated as follows.

1) Fix $E_b/N_0 = \delta + \epsilon$ and calculate $\underline{\sigma}^2$.
2) for $k = 1 \ldots N_c$, find $\underline{\lambda}_{opt}^{(C_k)}$ with the local optimization procedure.

$\underline{\lambda}_{opt}^{(C_{N_c})}$ gives the final result.

For the local optimization of class C_k, a linear programming routine is executed, which requires definition of the check node degree distribution $\underline{\rho}$, $E_b/N_0 = \delta + \epsilon$ in dB, and the maximum variable node degree $d_{v_{max}}$.

1) Initialization $\quad d_{v_{min}}^{(k)} = d_{v_{max}}$
2) While optimization failure
   a) Optimize

$$\max_{\underline{\lambda}} \sum_{j=1}^{N_s} k_j \sum_{i=2}^{d_{v_{max}}} \lambda_{j,i}^{(C_k)} \tag{16}$$

under the constraints $[C_1] - [C_6]$.

$[C_1]$ Rate constraint

$$\sum_{j=1}^{N_s} \sum_{k=1}^{N_c} \underline{\lambda}_j^{(C_k)T} \underline{1/d_v} = \frac{1}{1-R} \underline{\rho}^T \underline{1/d_c} \tag{17}$$

$[C_2]$ Proportion distribution constraints

i)

$$\sum_{j=1}^{N_s} \sum_{k=1}^{N_c} \underline{\lambda}_j^{(C_k)T} \underline{1} = 1 \tag{18}$$

ii) $\forall k \in \{1, \ldots, N_c - 1\}$,

$$\sum_{j=1}^{N_s} \underline{\lambda}_j^{(C_k)T} \underline{1/d_v} = \alpha_k \frac{R}{1-R} \underline{\rho}^T \underline{1/d_c} \tag{19}$$

iii) $\forall j \in \{1, \ldots, N_s - 1\}$,

$$\sum_{k=1}^{N_c} \underline{\lambda}_j^{(C_k)T} \underline{1/d_v} = \beta_j \frac{1}{1-R} \underline{\rho}^T \underline{1/d_c} \tag{20}$$

$[C_3]$ Convergence constraints, see (12)

$$F(\underline{\lambda}, \underline{\rho}, \underline{\sigma}^2, x) > x \tag{21}$$

$[C_4]$ Stability condition, see (14) and (15)

$$\sum_{j=1}^{N_s} \sum_{k=1}^{N_c} \lambda_{j,2}^{(C_k)} < \left[\sum_{j=1}^{N_s} \beta_j e^{-1/2\sigma_j^2} \cdot \sum_{m=2}^{d_{c_{max}}} \rho_m (m-1) \right]^{-1} \tag{22}$$

$[C_5]$ Minimum variable node degree constraint

$$\forall i < d_{v_{min}}^{(k)},\ \forall j,\ \lambda_{j,i}^{(C_k)} = 0 \tag{23}$$

$[C_6]$ Previous optimization constraints

$$\forall k' < k,\ \forall j,\ \underline{\lambda}_j^{(C_{k'})} \quad \text{is fixed} \tag{24}$$

b) $d_{v_{min}}^{(k)} = d_{v_{min}}^{(k)} - 1$

End

E. Code Construction

When the optimal degree distribution of the variable nodes is found, a parity-check matrix is constructed by the Approximate Cycle Extrinsic message degree (ACE) algorithm [21]. The ACE algorithm selectively avoids small cycle clusters that are isolated from the rest of the graph and has good performance in the error-floor region for irregular LDPC codes.

V. Simulation Results

In this section, simulation results for an example with 8-PSK are presented. We denote our scheme by higher order constellation UEP ("HOC-UEP"), which is a UEP-LDPC code optimized for the different σ_j^2 from the modulation. The noise vector $\underline{\sigma}^2$ is calculated according to (4), with $N_s = 2$ and $\underline{\beta} = [2/3,\ 1/3]$ for Gray-labeled 8-PSK. The HOC-UEP scheme is compared to a UEP-LDPC code optimized for BPSK [14], but used for 8-PSK. This scheme, that is denoted by "UEP", designs the code for an average σ^2 and assigns the bits following the natural bit assignment. The degree distributions are optimized for $R = 1/2$, $N_c = 3$, $\underline{\alpha} = [0.3,\ 0.7]$, and $d_{v_{max}} = 30$. The check node degree distribution is chosen as $\rho(x) = 0.00749x^7 + 0.99101x^8 + 0.00150x^9$, which is found by numerical optimization in [19] to be a good check node degree distribution for $d_{v_{max}} = 30$.

Table I shows the degree distributions given by the two design algorithms. For the UEP scheme, we arbitrarily choose $\epsilon = 0.1$ dB to allow for some unequal error protection. The resulting degree distributions $\underline{\lambda}^{(C_k)}$ are given for each protection class C_k. The minimum threshold δ of the HOC-UEP code is 0.27 dB lower than of the corresponding UEP code. Thus, we design the HOC-UEP code for $\epsilon = 0.37$ dB in order to have the same thresholds for both schemes. The degree distributions of the HOC-UEP scheme $\underline{\lambda}_j^{(C_k)}$ for protection classes C_k and modulation classes M_j are also given in Table I. For comparison, the degree distributions for both algorithms are also shown for the minimum tresholds, that is, $\epsilon = 0$ dB.

Finite length codeword simulations with $n = 4096$ and 50 decoding iterations are performed using the equivalent BPSK channels. Simulations verify that 8-PSK modulation and demodulation give almost exactly the same results as simulations with the equivalent BPSK channels. We assume that a soft demapper provides the message passing decoder with the channel log-likelihood ratios (LLRs) in any case using higher order constellation modulation and demodulation. Note that the channel LLRs are computed using the appropriate noise variances σ_j^2 of the modulation classes.

Fig. 2 shows the overall BER after 50 decoding iterations. By design, the overall BERs for the codes with $\epsilon \neq 0$ dB are higher than for the corresponding codes with $\epsilon = 0$ dB. This is because the thresholds of the codes are increased in order to allow an increased average variable node degree of the most protected classes. Fig. 2 also shows that for high E_b/N_0, the overall BERs of the HOC-UEP codes are lower than for the UEP codes. The overall BER of the HOC-UEP $\epsilon = 0.37$ dB code is lower than the overall BER of the UEP $\epsilon = 0.1$ dB code, even though they are designed for the same threshold. For an overall BER of 10^{-5}, there is a gain of around 0.7 dB by the HOC-UEP scheme.

The BER performances of the individual protection classes C_1 and C_2 for the UEP scheme are shown in Fig. 3. The UEP capability, that is, the difference in BER between class C_1 and C_2, is increased with increasing ϵ. For $\epsilon = 0$ dB, the UEP capability is accomplished by assignment of high degree variable nodes to the most protected classes.

Fig. 4 shows the BER performance of protection classes C_1 and C_2 for the HOC-UEP scheme. The results show that the

TABLE I
DEGREE DISTRIBUTIONS FOR THE UEP AND HOC-UEP SCHEMES.

	C_1	C_2	C_3
$\epsilon = 0$ dB			
UEP	$\lambda_7 = 0.0799$ $\lambda_8 = 0.0948$ $\lambda_{30} = 0.3029$	$\lambda_3 = 0.1790$ $\lambda_6 = 0.0737$ $\lambda_7 = 0.0414$	$\lambda_2 = 0.2103$ $\lambda_3 = 0.0181$
HOC-UEP M_1	$\lambda_9 = 0.1703$ $\lambda_{10} = 0.0555$ $\lambda_{30} = 0.1811$	$\lambda_3 = 0.1673$	$\lambda_2 = 0.1240$
HOC-UEP M_2	$\lambda_{30} = 0.0854$	$\lambda_4 = 0.0225$ $\lambda_5 = 0.0738$ $\lambda_7 = 0.0117$	$\lambda_2 = 0.0878$ $\lambda_3 = 0.0022$ $\lambda_4 = 0.0183$
$\epsilon = 0.1$ dB			
UEP	$\lambda_{11} = 0.1783$ $\lambda_{12} = 0.1184$ $\lambda_{30} = 0.2183$	$\lambda_3 = 0.2041$ $\lambda_4 = 0.0393$	$\lambda_2 = 0.1841$ $\lambda_3 = 0.0575$
$\epsilon = 0.37$ dB			
HOC-UEP M_1	$\lambda_{16} = 0.5255$ $\lambda_{17} = 0.0088$	$\lambda_3 = 0.0187$	$\lambda_2 = 0.2174$
HOC-UEP M_2		$\lambda_3 = 0.1929$ $\lambda_4 = 0.0293$	$\lambda_3 = 0.0075$

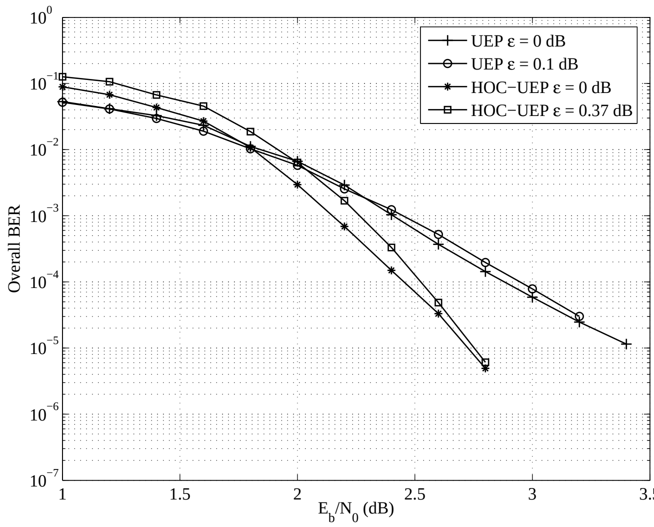

Fig. 2. Overall bit-error rate performance.

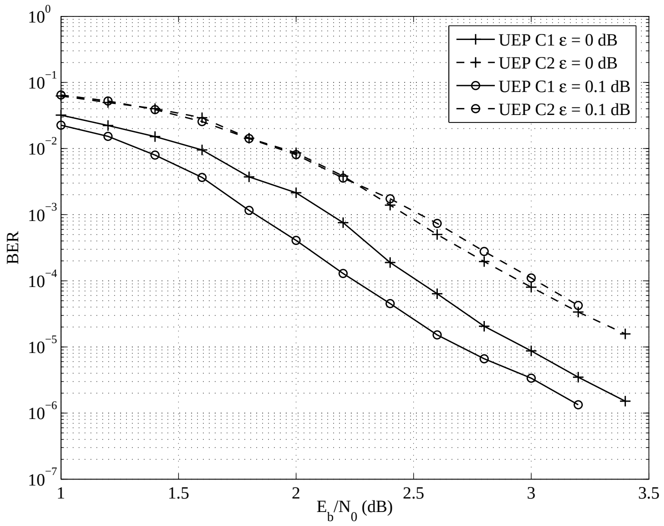

Fig. 3. Bit-error rate performance of protection class C_1 and C_2 for the UEP scheme.

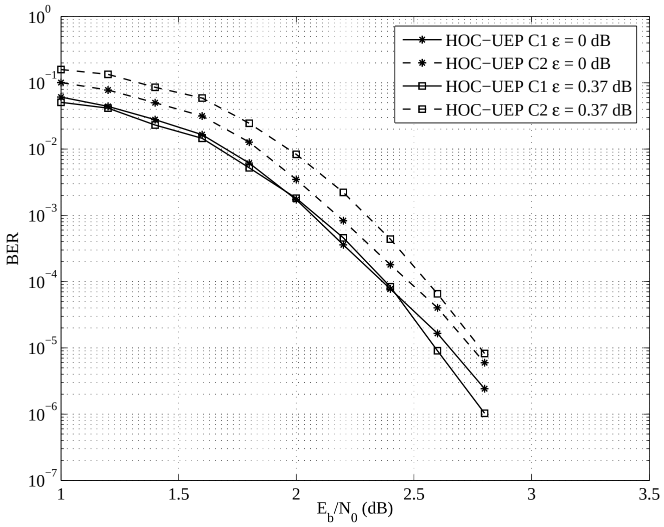

Fig. 4. Bit-error rate performance of protection class C_1 and C_2 for the HOC-UEP scheme.

HOC-UEP $\epsilon = 0.37$ dB code has more UEP capability than the HOC-UEP $\epsilon = 0$ dB code.

A comparison of the UEP capability for the UEP scheme and the HOC-UEP scheme suggests that a high ϵ is needed for the HOC-UEP scheme in order to achieve UEP. However, a high ϵ does not seem to affect the overall BER of the HOC-UEP scheme much for high E_b/N_0, see Fig. 2.

Comparing the individual protection classes of the HOC-UEP $\epsilon = 0.37$ dB and the UEP $\epsilon = 0.1$ dB scheme at BER 10^{-5}, we gain 0.1 dB for protection class C_1 and (expected) 0.7 dB for class C_2. These gains are expected to be even higher for lower BERs.

VI. Conclusions

In this paper, we present a flexible design method for UEP-LDPC codes with higher order constellations which is applicable to arbitrary signal constellations and arbitrary number and proportions of the protection classes. For an example with 8-PSK, it is shown that the overall BER is reduced by the proposed method and there is a gain of 0.7 dB at BER 10^{-5}. The results for the individual protection classes show only slightly reduced UEP capability for the HOC-UEP design method, but lower bit-error rates for all protection classes corresponding to information bits.

Acknowledgment

This work is part of the FP6 / IST project M-Pipe and is co-funded by the European Commission.

References

[1] G. Ungerböck, "Channel Coding with Multilevel/Phase Signals," *IEEE Trans. on Information Theory*, vol. 28, pp. 55–67, Jan. 1982.

[2] H. Imai and S. Hirakawa, "A new multilevel coding method using error correcting codes," *IEEE Trans. on Information Theory*, vol. 23, pp. 371–377, May 1977.

[3] G. Ungerböck, "Trellis-Coded Modulation with Redundant Signal Sets Part I: Introduction," *IEEE Communications Magazine*, vol. 25, pp. 5–11, Feb. 1987.

[4] G. Ungerböck, "Trellis-Coded Modulation with Redundant Signal Sets Part II: State of the Art," *IEEE Communications Magazine*, vol. 25, pp. 12–21, Feb. 1987.

[5] U. Wachsmann, R. Fischer, and J. Huber, "Multilevel Codes: Theoretical Concepts and Practical Design Rules," *IEEE Trans. on Information Theory*, vol. 45, pp. 1361 – 1391, July 1999.

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

[7] H. Sankar, N. Sindhushayana, and K. Narayanan, "Design of low-density parity-check (LDPC) codes for high order constellations," in *Proc. Globecom 2004*, Nov. 2004.

[8] R. Gallager, "Low-Density Parity-Check Codes," *IEEE Trans. on Information Theory*, vol. 8, pp. 21 – 28, Jan. 1962.

[9] M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. on Information Theory*, vol. 27, pp. 533 – 547, Sept. 1981.

[10] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Trans. on Information Theory*, vol. 47, pp. 498 – 519, Feb. 2001.

[11] J. Hou, P. Siegel, L. Milstein, and H. Pfister, "Capacity-Approaching Bandwidth-Efficient Coded Modulation Schemes Based on Low-Density Parity-Check Codes," *IEEE Trans. on Information Theory*, vol. 49, pp. 2141 – 2155, Sept. 2003.

[12] K. Kasai, T. Shibuya, and K. Sakaniwa, "Detailed representation of irregular LDPC code ensembles and density evolution," in *Proc. ISIT 2003*, June 2003.

[13] C. Poulliat, D. Declercq, and I. Fijalkow, "Optimization of LDPC Codes for UEP Channels," in *Proc. ISIT 2004*, June 2004.

[14] C. Poulliat, I. Fijalkow, and D. Declercq, "Scalable image transmission using UEP optimized LDPC codes," in *Proc. ISIVC 2004*, July 2004. http://publi-etis.ensea.fr/2004/PFD04a.

[15] L. Sassatelli, W. Henkel, and D. Declercq, "Check-Irregular LDPC Codes for Unequal Error Protection under Iterative Decoding," in *Proc. 4th International Symposium on Turbo Codes & Related Topics 2006*, Apr. 2006.

[16] Y. Li and W. Ryan, "Bit-reliability mapping in LDPC-coded modulation systems," *IEEE Communications Letters*, vol. 9, pp. 1–3, Jan. 2006.

[17] K. Fazel and M. Ruf, "Combined multilevel coding and multiresolution modulation," in *Proc. ICC 1993*, pp. 1081–1085, May 1993.

[18] J. Proakis, *Digital Communications*. McGraw-Hill, 2001.

[19] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes," *IEEE Trans. on Information Theory*, vol. 47, pp. 619–637, Feb. 2001.

[20] S.-Y. Chung, T. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. on Information Theory*, vol. 47, pp. 657–670, Feb. 2001.

[21] T. Tian, C. Jones, D. Villasenor, and R. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Trans. on Communications*, vol. 52, pp. 1242–1247, Aug. 2004.