2007

Error Floors of LDPC Coded BICM

Aditya Ramamoorthy
Iowa State University, adityar@iastate.edu

Nedeljko Varnica
Marvell Semiconductor, Inc.

Error Floors of LDPC Coded BICM

Aditya Ramamoorthy
Department of Electrical and Computer Engineering
Iowa State University
Ames, Iowa 50011
Email: adityar@iastate.edu

Nedeljko Varnica
Marvell Semiconductor Inc.
Santa Clara, CA 95054
Email: nvarnica@marvell.com

***Abstract*— In recent years performance prediction for communication systems utilizing iteratively decodable codes has been of considerable interest. There have been significant breakthroughs as far as the analysis of LDPC code ensembles is concerned but the more practical problem of predicting the FER/BER of a particular code has proved to be much more difficult. In this work we present a technique (based on the work of Richardson '03) for finding lower and upper bounds on the performance of LDPC coded BICM systems for a given code. The insight gained from the prediction technique is used to design interleavers that improve the error floors of these systems.**

I. Introduction

The performance of iterative coding schemes such as turbo codes and LDPC codes is well-known to suffer from the *error floor problem*. The performance curve of such codes typically consists of two different parts. In the low SNR region the frame error rate (FER)/bit error rate (BER) drop very fast with increasing SNR. This part of the curve is usually referred to as the waterfall region. However beyond a certain SNR the slope of the curve changes and the drop in FER/BER is no longer as sharp. In particular the irregular LDPC codes introduced in [1] have an error floor that is much more pronounced than regular codes [2].

For codes that are decoded using a bounded distance decoder (e.g. RS codes) or a maximum likelihood decoder (e.g. convolutional codes) it is possible to predict the performance of the code by using the union bound on the probability of error. This bound is reasonably tight at high SNR. However for iteratively decoded codes no such clear characterization of decision boundaries exists and consequently performance prediction is much harder. It is of great interest to understand the behavior of iterative decoding and characterize its failure mechanisms especially in the error floor region. In the context of data storage applications this problem is of utmost importance because the frame or sector error rate of interest is typically below 10^{-10}.Thus it becomes very important to have a tool for predicting the FER in the high SNR region.

While there is extensive literature on the asymptotic analysis of ensembles of codes ([1] [3] [4] among others) that consider performance bounds in the limit of large block length, the more practical issue of performance analysis for a given code has been much harder. The only channel where the decoding failure has a clear characterization is the binary erasure channel (BEC). As noted by [5] the iterative decoding algorithm hits a fixed point if and only if the set of variable nodes erased by the channel forms a graphical structure called a stopping set. Thus the problem of estimating the FER of the code becomes the problem of finding the stopping set spectrum of the code. For all other channels the problem of FER prediction is much harder. In [6] Richardson presented a numerical technique for the prediction of the error floor of an LDPC code with BPSK modulation over the AWGN channel. Other approaches that we are aware of include [7] [8] and [9].

In this paper we extend Richardson's method of error floor prediction to obtain lower and upper bounds on the frame error rate of LDPC coded bit interleaved coded modulation (BICM) [10]. Furthermore we exploit the insight gained from the prediction technique to design efficient interleavers between the code and the constellation mapping to further reduce the error floors.

Error floor prediction for systems using M-ary modulation and iterative codes is important in storage systems that store data in multiple levels e.g. M-ary optical storage [11] .

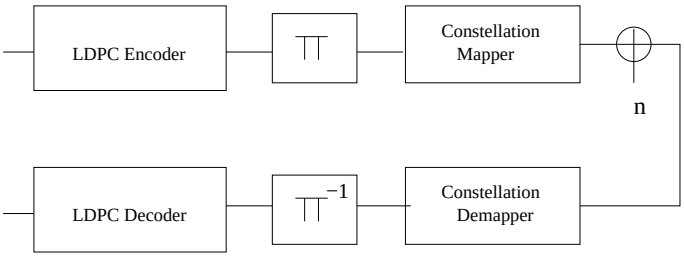

Fig. 1. System Block Diagram

The organization of the paper is as follows. Section II describes our technique for performing the error floor prediction for a LDPC coded BICM system. Results that demonstrate the accuracy of the technique are also included. Section III explains the design of the interleaver based on the prediction technique that is found to have improved error floors and Section IV concludes the paper.

II. Error Floor Prediction Technique

In this section we shall outline our strategy for estimating the error floor for LDPC coded BICM. The system under consideration is shown is Fig. 1. We consider only Gray mapping since this mapping has been found to have the best performance at medium block lengths [12]. Of course other

strategies such as different symbol mappings with turbo equalization and/or multilevel coding may also be of interest. Work is currently under progress to perform error floor prediction for such systems as well.

We would like to begin by pointing out the differences between the prediction problem for the case of BPSK modulation and other higher order modulations. There are two main issues that make the problem more complicated.

1) When we use constellations such as M-PAM the performance of the code in general depends on the transmitted codeword. Thus, we cannot work under the all-zeros codeword assumption.
2) In the BPSK case there is a unique direction and a noise threshold value that causes a bit error at a particular position if the noise pushes the transmitted symbol in that direction. When we have higher-order modulations this does not remain true. A bit error on a particular position can be due to different noise directions and different noise thresholds. For example suppose that constellation

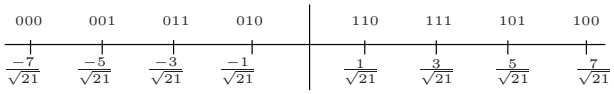

Fig. 2. Normalized 8-PAM constellation

point 010 is transmitted from the power normalized 8-PAM constellation shown in Fig. 2. If the noise affecting the transmission lies in the region $[-3/\sqrt{21}, -1/\sqrt{21})$ then the hard decision corresponding to the received value would cause a LSB error, whereas if the noise acts in the opposite direction (i.e.,lies in the region $(1/\sqrt{21}, 3/\sqrt{21}]$) the hard decision would correspond to a MSB error. Now fix the noise direction to be negative. If the noise lies in the region $[-5/\sqrt{21}, -3/\sqrt{21})$ then the hard decision would result in both an intermediate bit position and a LSB error, whereas if it lies in the region $(-\infty, -5/\sqrt{21})$ it would cause only an intermediate bit position error. Of course, for Gray-labeled constellations, the average probability that the noise results in a LSB error is the highest.

We shall use the terminology of trapping sets introduced in [6] in this paper. Suppose the code has N variable nodes. Let the maximum number of iterations of iterative decoding be denoted by L. Following [6] the iterative decoder is defined as a sequence of maps $D^l(y) : Y \rightarrow \{0,1\}^N, l = 1, 2, ..., L$. Here, l denotes the iteration number. The i^{th} decoded bit in the code shall be denoted by $D^l(y)[i]$. For a given input y, the failure set $T(y)$ is the set of bits that are not eventually correct. If $T(y) \neq \phi$, then $T(y)$ is called a trapping set. We also define a type of trapping set which we call *stable trapping set* [13].

Definition 1: Stable Trapping Set. A trapping set $T(y)$ is said to be a stable trapping set if $T(y) \neq \phi$ and $D^L(y)[i] = D^{L-1}(y)[i] = .. = D^{L-\eta}(y)[i]$ for all $i \in [0, ..., N-1]$ i.e. the hard decision based on the decoding is the same for the last η iterations.

In practice we set $L = 50$ and $\eta = 5$. We have observed experimentally that stable trapping sets are the most dominant failure mechanism in the error floor region. There are also other errors that are not stable. However with increasing SNR we have observed that the fraction of stable errors keeps increasing and eventually almost all the errors correspond to stable errors. Since the error floor prediction technique proceeds by finding the error rate contribution of each trapping set, concentrating only on stable trapping sets reduces the complexity of the procedure without sacrificing much accuracy. We call a stable trapping set dominant if its contribution to the error floor is large. Our strategy for error floor prediction shall have two stages.

- **Stage 1:** In the first stage we attempt to find as many trapping sets as possible. This can be performed by techniques recently presented in the literature, e.g., [8]. (Alternatively, for codes with relatively high error floors, we can simply perform long simulations and collect stable trapping sets.) Experimentally we have observed that the dominant trapping sets for the AWGN channel with BPSK modulation also tend to be the dominant trapping sets for LDPC coded BICM with higher order modulations. However the relative contribution of a given trapping set to the error floor in the two cases tends to be different.
- **Stage 2:** Suppose that a candidate list of trapping sets has been found. The next step is to evaluate the contribution towards the frame error rate of each trapping set. This is explained in detail in the next section.

A. Error Probability evaluation for a given trapping set

For a fixed interleaver each variable node in the code is associated with a particular bit level in the constellation such as the most-significant bit (MSB) or the least-significant bit (LSB) or some intermediate position. For each variable node position v, let $pos(v)$ denote the bit level at which it participates. Our convention shall be to let 0 represent the LSB, $\log_2 M - 1$ represent the MSB and the intermediate numbers represent the intermediate bit levels of the constellation. Let the function $S : \{0, 1, ..., N-1\} \rightarrow \{0, 1, ..., [N/\log_2 M]-1\}$ be the mapping that maps a variable node to its corresponding constellation point position. For a given transmission we let $\Gamma(i), i = 0, 1, ..., [N/\log_2 M] - 1$ denote the particular constellation point that was transmitted at location i. Of course $\Gamma(i)$ depends on the transmitted sequence but we do not make the dependence explicit to keep the notation simple.

Let us denote a given trapping set by T and the subset of decoder inputs that cause the decoder to fail on T by ξ_T. Our objective is to get an estimate of $P(\xi_T)$. This probability depends on

a) the channel characteristics,
b) the modulation, and
c) the constellation labeling.

We shall present a technique to determine a lower bound on $P(\xi_T)$ for the case of a M-PAM, Gray labeled constellation

TABLE I

CLASSIFICATION OF CONSTELLATION POINTS

Bit Level	Bad	Good
0	0,1,2,3,4,5,6,7	·
1	1,3,5,7	0,2,4,6
2	2,6	0,1,3,4,5,7

when the noise is AWGN. Each individual element of T shall be represented by $x_i^T, i = 1, 2, ..., |T|$. Thus $T = \{x_1^T, x_2^T, ..., x_{|T|}^T\}$. We can also identify the bit level position at which each x_i^T participates in as $pos(x_i^T)$.

The first observation we make is that in the high SNR region it is highly probable that the noise is low enough so that the received value lies within the decision boundaries of the neighboring points. e.g. for the normalized 8-PAM constellation the noise is with very high probability in the interval $[-3/\sqrt{21}, +3/\sqrt{21}]$. Next, for a given labeling the different bit levels for a given constellation point see different quality of channels. For example consider the MSB (bit level 2) of point 001 in the Gray mapped 8-PAM constellation shown in Fig. 2. Under the assumption that errors are nearest neighbor this MSB will never be in error. However note that a LSB (bit level - 0) error occurs when the received value is closer to 000 and a bit level - 1 error occurs when the received value is closer to 011.

Analyzing each constellation point in this fashion we can classify each point as *good* or *bad* for a given bit-level k (where $k = 0, 1, 2$ in this case) according to whether a bit-level k error happens when the received point is closer to a neighboring point. This is shown in Table I.

We proceed by fixing a worst case representative for the transmitted constellation point $S(x_i^T)$ for each $x_i^T, i = 1, 2, ..., |T|$ in the transmitted sequence e.g. suppose x_i^T participates as a MSB. Then in the simulations for determining the error rate contribution of T we shall always set $\Gamma(S(x_i^T)) = 010$ or $\Gamma(S(x_i^T)) = 110$. This is intuitively the right choice because these constellation points have the maximum probability of causing a MSB error in the simulations. Therefore the error floor should have a significant contribution from transmitted codewords where the worst case representative occurs. Note that though we can use either 010 or 110, once decided we shall fix the representative to use for $\Gamma(S(x_i^T))$ for a particular simulation. So, given T we first determine $\{pos(x_1^T), pos(x_2^T)..., pos(x_{|T|}^T)\}$ and then fix their representatives to be used for the corresponding constellation point positions $\{S(x_1^T), S(x_2^T), ..., S(x_{|T|}^T)\}$ as well. Note that we assume here that the trapping set size in bits is the same as its size in PAM symbols, i.e. for all dominant trapping sets the gap between the coordinates satisfies $|x_i^T - x_j^T| \geq \log_2 M, i \neq j$, for all $i, j \in \{1, ..., |T|\}$. This is, in fact, a mild assumption as it is not difficult to construct codes that satisfy this constraint. Moreover, most of the well-known codes (e.g., the MacKay (1008,504) code [14]) have this property. In the simulation, the remaining transmitted constellation points were chosen at random and the decoding was performed in the appropriate coset of the transmitted sequence. This was done to avoid the complexity of performing encoding for each transmission.

Consider the noise vector of dimension $|T|$ that acts on the corresponding constellation points. Note that once the representatives have been fixed for each constellation point we can find a $|T|$-dimensional vector that indicates the direction in which the noise should act so that the probability of error on T is maximum. Our strategy shall be to condition on the value of this noise vector in a carefully chosen direction. For example suppose the trapping set is such that its worst case representatives are [010 011 101 010] and we want the noise to cause errors in bit levels $2, 1, 0$ and 2 respectively. Then the normalized direction of the noise should be $[+1\ -1\ +1\ +1]/\sqrt{4}$.

Suppose the noise vector on T is denoted by $(n_1, n_2, ..., n_{|T|})$. As above, based on the bit levels that we want the errors to be at, an orthonormal direction vector can be identified that we denote as $\bar{d} = (d_1, d_2, ..., d_{|T|})$. We can also find $|T|-1$ orthonormal vectors $\bar{b}_2, \bar{b}_3, ..., \bar{b}_{|T|}$ so that along with $\bar{d}$ they complete the $|T|$-dimensional orthonormal basis. Then

$$(n_1, n_2, ..., n_{|T|}) = \gamma\bar{d} + \sum_{i=2}^{|T|} \delta_i \bar{b}_i \tag{1}$$

As in [6] we condition on the value of γ since we expect the component of the noise along the direction $\bar{d}$ to be largely responsible for decoder failure. Let $A_T = \{\Gamma(S(x_1^T)) = r'_1, \Gamma(S(x_2^T)) = r'_2, ..., \Gamma(S(x_{|T|}^T)) = r'_{|T|}\}$ denote the event that the worst case representatives have been chosen for the each of the bits in T. Then we have

$$\begin{aligned} P(\xi_T|A_T) &= \mathbb{E}_\gamma[P(\xi_T|A_T, \gamma)] \\ &= \int_{-\infty}^{\infty} P(\xi_T|A_T, \gamma = x) \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} dx \end{aligned} \tag{2}$$

where σ^2 is the variance of the AWGN noise.

The factor $P(\xi_T|A_T, \gamma = x)$ is determined by simulation whereas the multiplicative factor $\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$ is available analytically. In the simulation we count only those failures that occur on T. While getting an estimate of this integral, one usually also finds more trapping sets that can be added to the list. We refer the reader to [6] for a more detailed explanation of this part of the work.

The previous process enables us to obtain an accurate estimate of $P(\xi_T|A_T)$. However we are actually interested in

$$\begin{aligned} P(\xi_T) = &\\ &\sum_{\{r_1,...,r_{|T|}\} \in \Omega_T} P(\xi_T|\Gamma(S(x_1^T)) = r_1, ..., \Gamma(S(x_{|T|}^T)) = r_{|T|}) \\ &\times P(\Gamma(S(x_1^T)) = r_1, ..., \Gamma(S(x_{|T|}^T)) = r_{|T|}) \end{aligned} \tag{3}$$

where $\Omega_T = \{\{r_1, r_2, ...r_T\} : \Gamma(S(x_1^T)) = r_1, ..., \Gamma(S(x_{|T|}^T)) = r_{|T|}\}$ is the set that contains all

TABLE II

DOMINANT TRAPPING SETS AND THEIR ERROR RATE CONTRIBUTIONS FOR THE MACKAY (1008,504) CODE

Trapping Set	$\sigma^2 = 0.0447$	$\sigma^2 = 0.0398$
70 286 430 516 853 937	1.67e-06	8.09e-07
98 329 418 594 696 884	1.32e-07	6.95e-08
61 199 220 321 435 510 904 971	1.03e-06	4.48e-07
198 222 239 262 294 643 686 792 897 976	2.98e-07	1.01e-07
76 235 239 274 288 332 468 964 979 985	3.7e-08	1.38e-08

possible constellation point choices for the locations $S(x_1^T), S(x_2^T), ..., S(x_{|T|}^T)$. Note that typically $|\Omega_T| = M^{|T|}$.

As explained previously the different constellation points can be grouped into *good* and *bad* categories as shown in Table I for 8-PAM. This categorization enables us to partition the set Ω_T into a total of $2^{|T|}$ disjoint subsets $B_i, i = 1, ..., 2^{|T|}$ such that $\cup_i B_i = \Omega_T$. Let R_i^j denote a member of B_i. We observe that in the high SNR region, based on our assumption of nearest neighbor errors, the error rate contribution

$$P(\xi_T|R_i^j) \approx p_i \quad \forall R_i^j \in B_i \tag{4}$$

i.e. we expect all members of a particular B_i to have approximately the same error rate contribution.

Suppose we let the worst case subset be denoted by B_1. As explained earlier by choosing a representative $A_T \in B_1$ we can determine its error rate contribution. Then we have

$$P(\xi_T) \geq P(\xi_T|A_T) \times \frac{|B_1|}{|\Omega_T|} \tag{5}$$

To improve our lower bound we can also choose subsets other than the worst case subset and find the error rate contribution of a representative from those subsets. This should yield a tighter lower bound. For example instead of choosing all representatives to be bad such as in B_1 we can choose one representative to be good and the remaining to be bad. We expect the error rate contribution of such a subset to be lower than B_1. The tightness of the lower bound will depend in general on the number of different subsets for which the error rate contribution is computed. Thus there is a tradeoff between the tightness of the bound and the complexity of computing it.

1) An Example: We shall now demonstrate the entire process by means of an example. Consider the MacKay $(1008, 504)$ code available at [14] driving a Gray mapped 8-PAM constellation sketched in Fig. 2 with no interleaver between the code and the constellation mapper. Table II contains a list of dominant stable trapping sets and their error rate contributions based on the worst case representatives for two SNR points.

Consider the trapping set $T = \{70\ 286\ 430\ 516\ 853\ 937\}$. The function S defined above is such that $\{x : x = S(t), t \in T\} = \{23\ 95\ 143\ 172\ 284\ 312\}$ and $\{y : y = pos(t), t \in T\} = \{1\ 1\ 1\ 0\ 1\ 1\}$. From Table I we can pick the set of representatives as $\{011\ 011\ 011\ 101\ 011\ 011\}$. Based on the representatives chosen we can choose the direction $\bar{d}$ as $[-1\ -1\ -1\ +1\ -1\ -1]/\sqrt{6}$. Then proceeding as explained above we can estimate $P(\xi_T|A_T)$. Now observe that corresponding to the 6 transmitted constellation points for this trapping set, there are a total of 8^6 possibilities. It is also clear that based on the classification in Table I that there are $4^5 \times 8$ transmit patterns that are equivalent to the one that we performed the simulation with under the nearest neighbor error assumption. Therefore we have

$$P(\xi_T) \geq \frac{4^5 \times 8}{8^6} P(\xi_T|A) \tag{6}$$

In fact intuitively we do not expect this lower bound to be too weak since most of the terms that contribute strongly to the overall failure rate have been captured.

B. Overall Error Probability

The procedure explained in the previous section shall be applied to each trapping set in the list. The number of dominant trapping sets is typically not very large and thus the complexity remains manageable.

1) Lower Bound: Let the trapping sets be denoted $T_1, T_2, ..., T_k$ and $A_{T_1}, A_{T_2}, ..., A_{T_k}$ denote the corresponding worst case representatives and $B_1^{T_1} \subseteq \Omega_{T_1}, B_1^{T_2} \subseteq \Omega_{T_2}, ...B_1^{T_k} \subseteq \Omega_{T_k}$ be the worst case subsets. Then the lower bound on the frame error rate in the error floor region is given by

$$FER \geq \sum_{i=1}^{k} P(\xi_{T_i}|A_{T_i}) \cdot \frac{|B_1^{T_i}|}{|\Omega_{T_i}|} \tag{7}$$

As explained before this lower bound can be improved if for each T_i we also choose representatives that are not worst case.

2) Upper Bound: In the discussion so far we have been concentrating on obtaining lower bounds on the FER contribution of a given trapping set T by ignoring the contribution of subsets of Ω_T that are not expected to contribute significantly to the FER. However our strategy can be used to provide meaningful upper bounds on the error rate contribution of T as well. To see this suppose that $\Omega_T = \cup_i B_i$. Let B_1 be the worst case subset and A_{T_1} be the corresponding worst case representative. Suppose we have β second worst subsets where one representative is chosen to be good and all the others are chosen to be bad. Let these be denoted by $B_2, ..., B_{\beta+1}$ and let $A_T^2, ..., A_T^{\beta+1}$ be the corresponding worst case representatives. Suppose we have computed $P(\xi_T|A_T)$ and $P(\xi_T|A_T^2), ..., P(\xi_T|A_T^{\beta+1})$ and

$$q = \max_{j=2,...,\beta+1} P(\xi_T|A_T^j) \tag{8}$$

Now if we assume that the error rate contribution of all subsets in $\Omega_T - \cup_{i=1}^{\beta+1} B_i$ is at most q, then we have

$$\begin{aligned} P(\xi_T) \leq P(\xi_T|A_T)\frac{|B_1|}{|\Omega_T|} + \sum_{j=2}^{\beta+1} P(\xi_T|A_T^j)\frac{|B_j|}{|\Omega_T|} \\ + q\left(1 - \sum_{k=1}^{\beta+1} \frac{|B_k|}{|\Omega_T|}\right) \end{aligned} \tag{9}$$

Once again this upper bound can be made tighter based on the number of different subsets for which we find the error rate contribution.

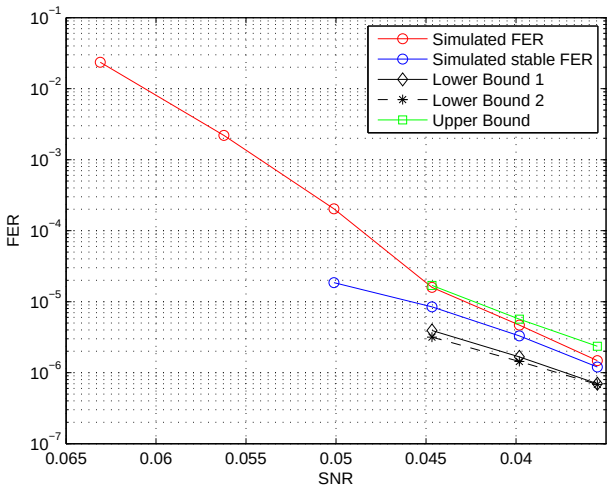

Fig. 3. Simulated and predicted lower and upper bound performance results for the $R = 1/2$ Mackay $(1008, 504)$ code driving an 8-PAM constellation.

C. Results

1) A Regular Code Example: We simulated the performance of the Mackay $(1008, 504)$ code with an 8-PAM constellation. The log-likelihood ratios were computed for each bit based on the received point and the decoding was performed by using the sum-product algorithm. The x-axis is labeled by the variance of the white Gaussian noise that was added to each transmitted point. To avoid the complexity of encoding each transmission, we performed the simulation by generating transmitted codewords at random and decoding in the appropriate coset. The results are plotted in Fig. 3.

The red curve marked "Simulated FER" is the simulated frame error rate of the code. This includes all errors. The blue curve is the error rate of stable trapping sets (see definition 1). We observe that the simulated FER is hitting an asymptote that is determined by the simulated stable FER i.e. as the SNR keeps increasing the fraction of stable errors keeps growing. The two black curves show the lower bound on the error floor based on the procedure we outlined in the previous section. We collected a list of stable trapping sets and computed the contribution of the error rate of each trapping set at the marked SNR points. The dotted black curve is the prediction based only on the worst case representatives for each trapping set. The solid black curve also computes the contribution of a trapping set when one representative is good and all others are bad. The green curve is an upper bound computed by the procedure outlined in the previous subsection. It can be seen that both the lower and the upper bounds track the simulated stable error rate of the system fairly closely.

2) An Irregular Code Example: As another way to demonstrate the validity of the proposed approach, we simulated an irregular rate-1/2 quasi-cyclic code with block length $N = 1008$ driving an 8-PAM constellation. The code was constructed using the degree distribution from [1] with maximum column weight 6. The size of circulants was set to $S_C = 12$. Among the dominant trapping sets of this code are the three

TABLE III

THREE TRAPPING SET CLASSES AND THEIR ERROR RATE CONTRIBUTIONS FOR THE IRREGULAR (1008,504) CODE AT $\sigma^2 = 0.0502$

Class	Trapping Set (Class representative)										lower bound	simulated
Class $T_{\rm I}$	204	273	286	482	515	520	554	597	745	759	2.4e-6	3.5e-6
$pos(t)$	0	0	1	2	2	1	2	0	1	0		
Class $T_{\rm II}$	205	274	287	483	504	521	555	598	746	760	1.2e-6	1.7e-6
$pos(t)$	1	1	2	0	0	2	0	1	2	1		
Class $T_{\rm III}$	206	275	276	484	505	522	556	599	747	761	6.0e-7	8.1e-7
$pos(t)$	2	2	0	1	1	0	1	2	0	2		

sets listed in Table III. These trapping sets are circular shifts of each other. (All S_C circular shifts of a trapping set would have the same error rate contribution for the case of the BPSK modulation, but this is not true for higher order modulations.) Given that the symbol size divides S_C, we can group trapping sets into three classes where each class contains $S_C/3 = 4$ trapping sets. Representatives of these classes are given in Table III. For example, for all sets in class $T_{\rm I}$: 3 variable nodes participate as MSB bits, 3 as intermediate bits and 4 as LSB bits. Thereby, the members of a given class have the same error rate contribution. Noting that the term $P(\xi_T|A_T)$ is the same for all sets in Table III, the ratios between the error rate contributions in the lower bound in (5) are $B^{T_{\rm I}}/B^{T_{\rm II}}/B^{T_{\rm III}} = (2^3 \times 4^3 \times 8^4)/(2^3 \times 4^4 \times 8^3)/(2^4 \times 4^3 \times 8^3) = 4/2/1$. Comparing these expected ratios to the ratios between the actual error rates (computed using Monte-Carlo simulations and counting the number of stable errors on given trapping sets), we see that they are almost exactly the same. Hence, this also validates the assumptions leading to the bound in (5).

III. INTERLEAVER DESIGN

The results of the previous section suggest that the bounds that we have presented earlier are non-trivial lower and upper bounds on the error floor for LDPC coded BICM. This naturally leads us to the question of whether the analysis can suggest a design strategy for codes with low error floors. One technique would be to avoid as many trapping sets as possible in the code. We shall not deal with this issue in this paper. Our design is motivated by the observation that in the case when we use higher order modulations different bit-levels have different reliabilities and we should be able to exploit this by changing the mapping between the constellation mapping and the code bits i.e. by designing the interleaver.

Note that every term in the bound in (7) depends on two quantities.

1) The term $P(\xi_{T_i}|A_{T_i})$ which is the probability of failing on a given trapping set T_i assuming that the worst case constellation symbols were transmitted. This term is dependent upon the structure of the parity check matrix of the LDPC code and the specific iterative decoding algorithm used in the system.

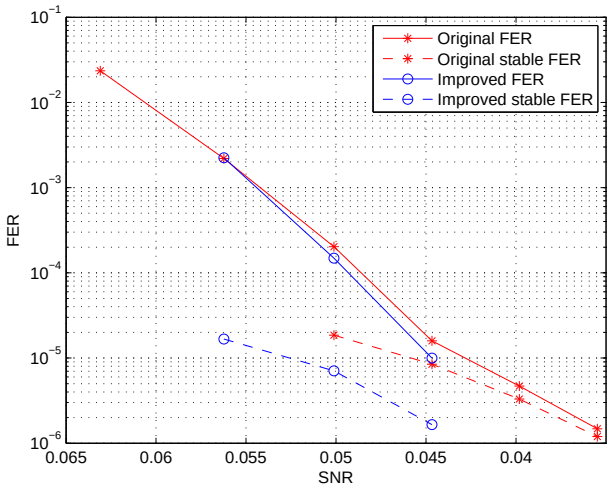

Fig. 4. Performance improvement gained by a proper design of the interleaver between the LDPC code and the constellation mapping.

2) The weighting factor associated with each $P(\xi_{T_i}|A_{T_i})$ term that depends on the bit levels in which a particular trapping set participates. This term depends upon the interleaver under use. For example if T_i has all its bits coming from LSB's then its weighting factor shall be 1. However by changing the interleaver so that the bits in the trapping set are mapped to MSB's we can reduce the weighting factor substantially. This should reduce the contribution of this particular trapping set to the overall frame error rate.

We have then following algorithm to design the interleaver

1) Find the dominant trapping sets in the error floor region. Let these be denoted $T_1, T_2, ..., T_k$.
2) Change the interleaver mapping such that all variable nodes in $\cup_{i=1}^{k} T_i$ are mapped to MSB's in some constellation point.

We have not observed any new trapping sets when a different interleaver is used. However a change in the interleaver could make certain trapping sets whose contribution to the FER in the original interleaver was low to become more dominant. Therefore a prediction of the improved error floor based on simply changing the weighting factors in (7) would be inaccurate. Experimentally we have observed that it is good to target only the most dominant trapping sets, thus not significantly changing the original interleaver.

A. Simulation Results

In this section we present the results obtained by the design of the interleaver between the LDPC code and the constellation mapper as explained in the previous section. All the curves are based on simulation i.e. we did not use the prediction technique presented earlier to generate any of these curves. The curves in red in Figure 4 correspond to the Mackay (1008,504) code driving a 8-PAM constellation with no interleaver between the code and the mapper. The curves in blue correspond to the best interleaver obtained by remapping the bits associated with the five most dominant trapping sets that we identified for the code. At a noise variance $\sigma^2 = 0.0447$ we observe that the stable error rate curve for the interleaved code is about an order of magnitude below the stable error rate curve for the non-interleaved code. Based on our previous observations and discussions this also implies that the error floor asymptote for the interleaved code shall be much lower than the non-interleaved code i.e. with increasing SNR the gap between the two curves shall keep increasing. However at lower SNR's where stable errors are not the dominant mechanism the net improvement in the FER is not much. The interleaver design strategy that we propose can therefore be used to obtain performance gain in the error floor after performance optimization for other metrics such as a steep waterfall.

IV. Conclusion

The problem of error floor prediction for BICM systems using LDPC codes was addressed. We extended the technique introduced by Richardson [6] to compute lower and upper bounds on the performance of these systems when a Gray-mapped constellation is used. An interleaver design strategy based on the prediction technique was developed that improves the error floors of these systems. Future work would involve developing prediction techniques for systems utilizing different labeling with and without turbo-equalization.

References

[1] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes," *IEEE Trans. on Info. Th.*, vol. 47, no. 2, pp. 619–637, 2001.
[2] R. Gallager, "Low Density Parity Check Codes," *PhD Thesis*, 1963.
[3] T. J. Richardson and R. L. Urbanke, "The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding," *IEEE Trans. on Info. Th.*, vol. 47, no. 2, pp. 599–618, 2001.
[4] D. Burshtein and G. Miller, "Asymptotic enumeration methods for analyzing LDPC codes," *IEEE Trans. on Info. Th.*, vol. 50, no. 6, pp. 1115–1131, 2004.
[5] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. on Info. Th.*, vol. 48, no. 6, pp. 1570–1579, 2002.
[6] T. J. Richardson, "Error Floors of LDPC Codes," 41^{st} *Allerton Conference on Communication, Control and Computing*, 2003.
[7] B. Xia and W. Ryan, "On Importance Sampling for Linear Block Codes," in *IEEE Intl. Conf. on Communications*, 2003, pp. 2904–2908.
[8] C. A. Cole, S. G. Wilson, E. K. Hall, and T. R. Giallorenzi, "A General Method for Finding Low Error Rates of LDPC Codes," *Submitted to IEEE Trans. on Info. Th. (available at http://www.arxiv.org/abs/cs.IT/0605051)*.
[9] E. Cavus, C. L. Haymes, and B. Daneshrad, "An IS Simulation Technique for Very Low BER Performance Evaluation of LDPC Codes," in *IEEE Intl. Conf. on Communications*, 2006.
[10] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved Coded Modulation," *IEEE Trans. on Info. Th.*, vol. 44, no. 3, pp. 927–946, 1998.
[11] S. W. Mclaughlin, "Signal Processing and coding for M-ary optical storage," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, vol. 5, 2005, pp. 753–756.
[12] J. Tan and G. Stuber, "Analysis and design of symbol mappers for iteratively decoded BICM," *IEEE Trans. on Wireless Comm.*, vol. 4, no. 2, pp. 662–672, 2005.
[13] N. Varnica, M. Fossorier, and A. Kavcic, "Augmented Belief-Propagation Decoding of Low-Density Parity-Check Codes," *IEEE Trans. on Comm. (to appear)*.
[14] D. J. C. Mackay, "http://www.inference.phy.cam.ac.uk/mackay/codes/."