

A Low-Density Parity-Check Code Tutorial

Part I - Introduction and Overview

William Ryan, Associate Professor
Electrical and Computer Engineering Department
The University of Arizona

Box 210104
Tucson, AZ 85721

ryan@ece.arizona.edu

April 2002

Block Code Fundamentals

- we will consider only (n,k) linear block codes over the binary field $\mathbb{F}_2 \triangleq (\{0, 1\}, +, \cdot)$
- $\mathbb{F}_2^n \triangleq$ the n -dimensional vector space over \mathbb{F}_2
- the elements of \mathbb{F}_2^n are the 2^n n -tuples $\bar{v} = [v_0, v_1, \dots, v_{n-1}]$ which we consider to be row vectors
- **Definition** An (n, k) linear block code \mathcal{C} with data word length k and codeword length n is a k -dimensional subspace of \mathbb{F}_2^n
- there are 2^k datawords $\bar{u} = [u_0, u_1, \dots, u_{k-1}]$ and 2^k corresponding codewords $\bar{c} = [c_0, c_1, \dots, c_{n-1}]$ in the code \mathcal{C} .

2

- since \mathcal{C} is a subspace of dimension k , \exists k linearly independent vectors $\bar{g}_0, \bar{g}_1, \dots, \bar{g}_{k-1}$ which span \mathcal{C}
- the correspondence (mapping) $\bar{u} \rightarrow \bar{c}$ is thus naturally written as

$$\bar{c} = u_0 \bar{g}_0 + \dots + u_{k-1} \bar{g}_{k-1}$$

- in matrix form, this is

$$\bar{c} = \bar{u}G$$

where

$$G = \begin{bmatrix} - & \bar{g}_0 & - \\ - & \bar{g}_1 & - \\ & \vdots & \\ - & \bar{g}_{k-1} & - \end{bmatrix}_{k=n}$$

is the so-called generator matrix for \mathcal{C}

- $\{\bar{g}_i\}$ being linearly independent
 $\Rightarrow G$ has rank k
 $\Rightarrow G$ may be row reduced and put in the form

$$G = [I : P]$$

(after possible column swapping which permutes the order of the bits in the code words)

- the null space \mathcal{C}^\perp of the subspace \mathcal{C} has dimension $n-k$ and is spanned by $n-k$ (linearly independent vectors $\bar{h}_0, \bar{h}_1, \dots, \bar{h}_{n-k-1}$
- since each $\bar{h}_i \in \mathcal{C}^\perp$, we must have for any $\bar{c} \in \mathcal{C}$ that
$$\bar{c} \bar{h}_i^T = 0, \forall i$$
- further, if $\bar{x} \in \mathbb{F}_2^n$, but $\bar{x} \notin \mathcal{C}$, then $\bar{x} \bar{h}_i^T \neq 0, \forall i$

- we may put this in a more compact matrix form by defining a so-called parity-check matrix H ,

$$H \Delta = \begin{bmatrix} - & \bar{h}_0 & - \\ - & \bar{h}_1 & - \\ & \vdots & \\ - & \bar{h}_{n-k-1} & - \end{bmatrix}_{(n-k) \times n},$$

so that

$$\bar{c} H^T = \bar{0}$$

if and only if $\bar{c} \in \mathcal{C}$

- suppose \bar{c} has w 1's (i.e., the Hamming weight of \bar{c} , $W_H(\bar{c})=w$) and the locations of those 1's are P_1, P_2, \dots, P_w
- then the computation $\bar{c} H^T = \bar{0}$ effectively adds w rows of H^T , rows P_1, P_2, \dots, P_w , to obtain the vector $\bar{0}$ one important consequence of this fact is that the minimum distance d_{\min} (= minimum weight W_{\min}) of \mathcal{C} is exactly the minimum number of rows of H^T which can be added together to obtain $\bar{0}$

5

Example (7,4) Hamming Code

$$H^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ & \dots & \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- we can see that no two rows sum to $\bar{0}$, but row 0 + row 1 + row 6 = $\bar{0}$

$$\Rightarrow d_{\min} = 3$$

=====

6

Low-Density Parity-Check Codes

- note that the parity-check matrix H is so called because it performs $m := n-k$ separate parity checks on a received word $\bar{y} = \bar{c} + \bar{e}$

Example with H^T as given above, the $n-k=3$ parity checks implied by $\bar{y} H^T = \bar{0}$ are

$$\begin{array}{cccccc} y_0 & + & y_1 & + & y_2 & + & y_4 & \stackrel{?}{=} & 0 \\ y_0 & + & y_1 & + & y_3 & + & y_5 & \stackrel{?}{=} & 0 \\ y_0 & + & y_2 & + & y_3 & + & y_6 & \stackrel{?}{=} & 0 \end{array}$$

=====

- a low-density parity-check (LDPC) code is a linear block code for which the parity-check matrix H has a low density of 1's

7

Definition a regular (n,k) LDPC code is a linear block code whose parity-check matrix H contains exactly $W_r = W_c(n/m)$ 1's per row, where $W_c \ll m$.

Remarks

- note multiplying both sides of $W_c \ll m$ by n/m implies $W_r \ll n$.

- the code rate $r = k/n$ can be computed from

$$r = \frac{W_r - W_c}{W_r} = 1 - \frac{W_c}{W_r}$$

- $W_c \geq 3$ is a necessity for good codes (Gallager) if H is low density, but if the number of 1's per column or row is not constant, the code is an irregular LDPC code

- LDPC codes were invented by Robert Gallager of MIT in his PhD dissertation (1960). They received virtually no attention from the coding community until the mid-1990's.

=====

8

- note from the result developed earlier,

$$d_{\min} = \min \{W_H(\bar{c}), \bar{c} \neq \mathbf{0} : \bar{c}H^T = \mathbf{0}\},$$

we should expect reasonably designed LDPC codes to have large d_{\min}

- this is because the operation $\bar{c}H^T$ adds selected rows of H^T (columns of H) and it would take a large number of such columns to sum to $\mathbf{0}$ if H is sparsely populated with 1's.

Example $W_c=3$

$$H = \begin{bmatrix} 1 & 1 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & & \\ 0 & 1 & 1 & 0 & & \\ 0 & 0 & 0 & 0 & \dots & \\ 1 & 0 & 1 & 1 & & \\ 1 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 1 & \dots & \\ 0 & 0 & 0 & 0 & & \\ 0 & 1 & 0 & 1 & & \\ 0 & 0 & 1 & 0 & & \\ & & & & & \\ & & & & & \\ & & & & & \end{bmatrix}$$

- note any two columns have an overlap of at most one 1; also the sparse property allows us to minimize such overlap
- a consequence of this is that the sum of the columns shown is nonzero

9

10

Representation of Linear Block Codes via Tanner Graphs

- one of the very few researchers who studied LDPC codes prior to the recent resurgence is Michael Tanner of UC Santa Cruz
- Tanner considered LDPC codes (and a generalization) and showed how they may be represented effectively by a so-called bipartite graph, now call a Tanner graph

Definition a bipartite graph is a graph (nodes or vertices connected by undirected edges) whose nodes may be separated into two classes, and where edges may only connect two nodes not residing in the same class

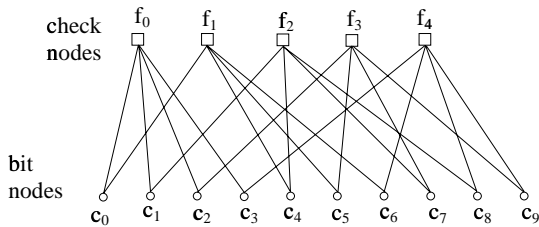
- the two classes of nodes in a Tanner graph are the (code bit nodes (or variable nodes) and the check nodes (or function nodes)
- the Tanner graph of a code is drawn according to the following rule:

check node j is connected to bit node i
whenever element H_{ji} in H is a 1

- one may deduce from this that there are $m = n-k$ check nodes and n bit nodes
- further, the m rows of H specify the m check node connections, and the n columns of H specify the n bit node connections

Example (10, 5) block code with $W_c = 2$ and $W_r = W_c(n/m) = 4$.

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$



- observe that nodes c_0, c_1, c_2 , and c_3 are connected to node f_0 in accordance with the fact that in the first row of H , $h_{00} = h_{01} = h_{02} = h_{03} = 1$ (all others equal zero)

- (for convenience, the first row and first column of H are assigned an index of 0)

- observe an analogous situation for f_1, f_2, f_3 , and f_4 .

- thus, as follows from the fact that $\bar{c}H^T = \bar{0}$, the bit values connected to the same check node must sum to zero

- note that the Tanner graph in this example is regular: each bit node is of degree 2 (has 2 edge connections and each check node is of degree 4)

- this is in accordance with the fact that $W_c = 2$ and $W_r = 4$

- we also see from this why $W_r = W_c(n/m)$ for regular LDPC codes:

$$(\text{\# bit nodes}) (\text{bit node degree}) = nW_c$$

must equal

$$(\text{\#check nodes}) (\text{check node degree}) = mW_r$$

13

14

Definition a cycle of length l in a Tanner graph is a path comprising l edges which closes back on itself

- the Tanner graph in the above example possesses a length-6 cycle as made evident by the 6 bold edges in the figure

Definition the girth of a Tanner graph is the minimum cycle length of the graph

- the shortest possible cycle in a bipartite graph is clearly a length-4 cycle
- length-4 cycles manifest themselves in the H matrix as four 1's that lie on the corners of a submatrix of H :

$$H = \begin{matrix} & \begin{matrix} a & b \end{matrix} \\ \begin{matrix} r \\ s \end{matrix} & \begin{bmatrix} \dots & \dots \\ 1 & 1 \\ \dots & \dots \\ 1 & 1 \\ \dots & \dots \end{bmatrix} \end{matrix}$$

- length-6 cycles are not quite as easily found in an H matrix:

$$H = \begin{matrix} & \begin{matrix} a & b & c \end{matrix} \\ \begin{matrix} r \\ s \\ t \end{matrix} & \begin{bmatrix} 1 & 1 & \\ 1 & & 1 \\ & 1 & 1 \end{bmatrix} \end{matrix}$$

- we are interested in cycles, particularly short cycles, because they have a negative impact on the decoding algorithm for LDPC codes as will be made evident below

15

16

Encoding

- as indicated above, once H is generated, it may be put in the form $\tilde{H} = [\tilde{P}^T : I]$ from which the systematic form of the generator matrix is obtained:

$$G = [I : P]$$

- encoding is performed via

$$\bar{c} = \bar{u}G = [\bar{u} : \bar{u}P],$$

although this is more complex than it appears for capacity-approaching LDPC codes (n large)

Example Consider a (10000, 5000) linear block code. Then $G = [I : P]$ is 5000 \times 10000 and P is 5000 \times 5000. We may assume that the density of ones in P is ~ 0.5 .

\Rightarrow there are $\sim 0.5(5000)^2 = 12.5 \times 10^6$ ones in P

$\Rightarrow \sim 12.5 \times 10^6$ addition (XOR) operations are required to encode one codeword

=====

Selected Results

- we present here selected performance curves from the literature to demonstrate the efficacy of LDPC codes
- the papers from which these plots were taken are listed in the reference section at the end of the note set
- we indicate the paper each plot is taken from to ensure proper credit is given

- Richard and Urbanke (2001) have proposed a lower complexity encoding technique based on the H matrix (not to be discussed here)
- an alternative approach to simplified encoding is to design the LDPC code via algebraic, geometric, or combinatoric methods
- such “structured” codes lend themselves to simple encoders based on shift-register circuits
- since they are simultaneously LDPC codes, the same decoding algorithms apply
- LDPC codes based on cyclic codes will be briefly discussed later

17

MacKay (March 1999, Trans IT)

- MacKay (and others) re-invented LDPC codes in the late 1990's
- here are selected figures from his paper (see his paper for code construction details; his codes are regular or nearly regular)

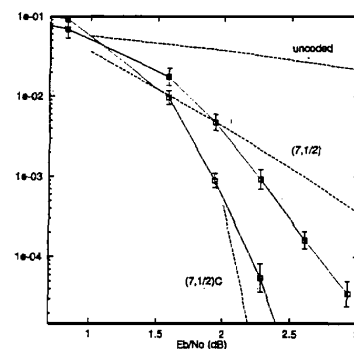


Fig. 11. Short-blocklength Gallager codes' performance over Gaussian channel (solid curves) compared with that of standard textbook codes (dotted curves). Vertical axis shows empirical bit error probability. It should be emphasised that *all* the block errors in the experiments with Gallager codes were *detected* errors: the decoding algorithm reported the fact that it had failed. Textbook codes: as in Fig. 9. Gallager codes: From left to right the codes had the following parameters (N, K, R) : (1008, 504, 0.5) (Construction 1A); (504, 252, 0.5) (1A).

18

MacKay (cont'd)

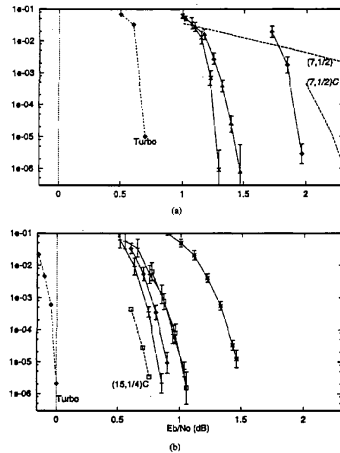


Fig. 9. Gallager codes' performance over Gaussian channel (solid curves) compared with that of standard textbook codes and state-of-the-art codes (dotted curves). Vertical axis shows empirical bit error probability. It should be emphasized that all the block errors in the experiments with Gallager codes were detected errors; the decoding algorithm reported the fact that it had failed. Panel (a) shows codes with rates between about 1/2 and 2/3; panel (b) shows codes with rates between 1/4 and 1/3. Textbook codes: The curve labeled (7,1/2) shows the performance of a rate 1/2 convolutional code with constraint length 7, known as the *de facto* standard for satellite communications [29]. The curve (7,1/2)C shows the performance of the concatenated code composed of the same convolutional code and a Reed-Solomon code. State of the art: The curve (15,1/4)C shows the performance of an extremely expensive and computer intensive concatenated code developed at JPL, based on a constraint length 15, rate 1/4 convolutional code (data courtesy of R. J. McEliece.) The curves labeled Turbo show the performance of the rate 1/2 Turbo code described in [12], [11] and the rate 1/4 code reported in [21]. Gallager codes: From left to right the codes had the following parameters (N, K, R) . Panel (a): (65389, 32621, 0.499) (1B); (19532, 9839, 0.496) (1B); (29331, 15331, 0.659) (1B). Panel (b): (40000, 10000, 0.25) (Construction 2A); (29507, 9507, 0.322) (2B); (14971, 4971, 0.332) (2B); (19900, 5000, 0.333) (2A); (13298, 3298, 0.248) (1B).

Irregular LDPC Codes

- our discussions above favored regular LDPC codes for their simplicity, although we gave examples of irregular LDPC codes
- recall an LDPC code is irregular if the number of 1's per column of H and/or the number of 1's per row of H is allowed to vary
- in terms of the Tanner graph, this means that the bit node degree and/or the check node degree is allowed to vary (the degree of a node is the number of edges connected to it)
- a number of researchers have examined the optimal degree distribution among nodes:

- MacKay, Trans. Comm., October 1999
- Luby, et al., Trans. IT, February 2001
- Richardson, et al., Trans. IT, February 2001
- Chung, et al., Comm. Letters, February 2001

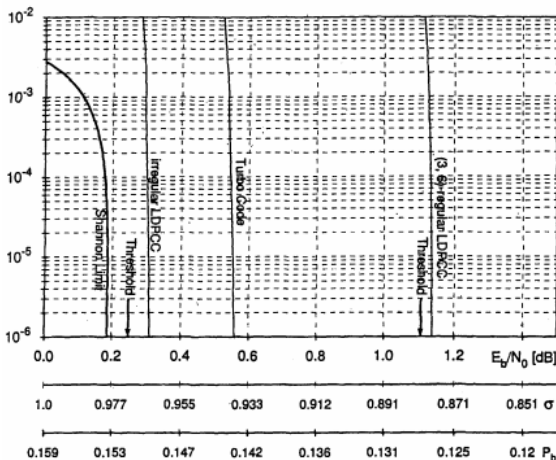
21

22

- the results have been spectacular, with performance surpassing the best turbo codes

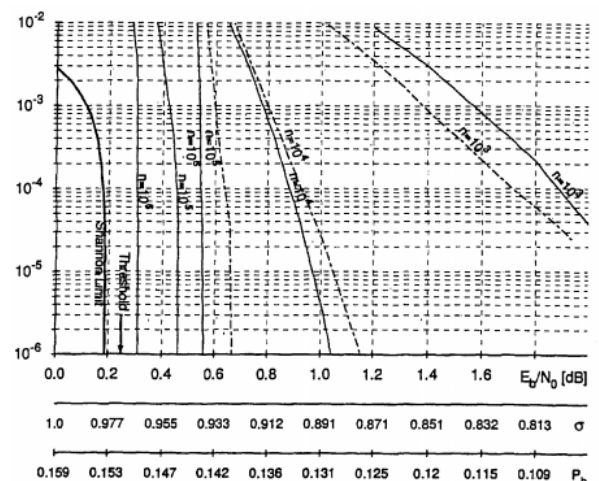
Richardson et al. Irregular Codes

- the plots below are for a (3, 6) – regular LDPC code, an optimized irregular LDPC code, and a turbo code
- the code parameters are $\frac{1}{2}$ (10^6 , $\frac{1}{2}$ 10^6) in all cases



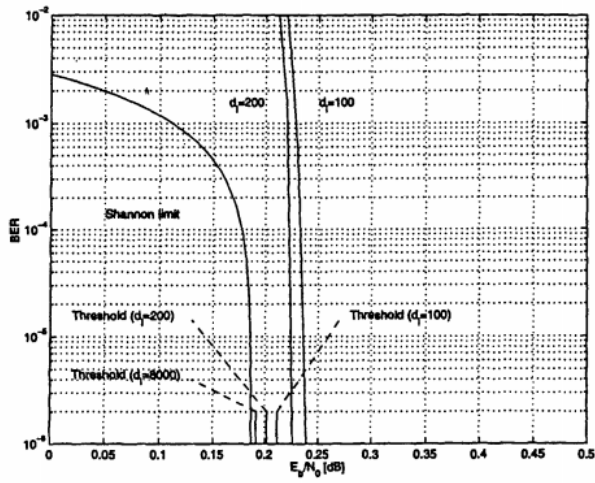
23

- plot below: turbo codes (dashed) and irregular LDPC codes (solid); for block lengths of $n=10^3$, 10^4 , 10^5 , and 10^6 ; all rates are $\frac{1}{2}$



24

- the plot below is of two separate $\frac{1}{2}(10^7, 12/10^7)$ irregular LDPC codes



25

- various LDPC codes based on Euclidean geometries (EG) and Projective geometries (PG)

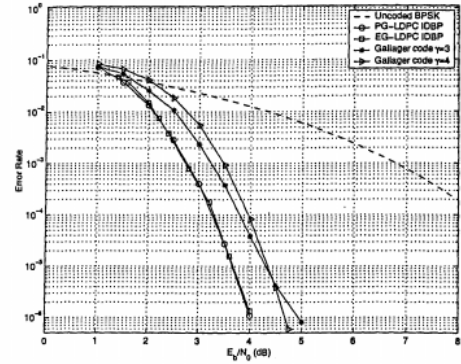


Figure 4: Bit-error probabilities of the (255, 175) EG-LDPC code, (273,191) PG-LDPC code and two computed searched (273,191) Gallager codes.

26

Kou et al. (cont'd)

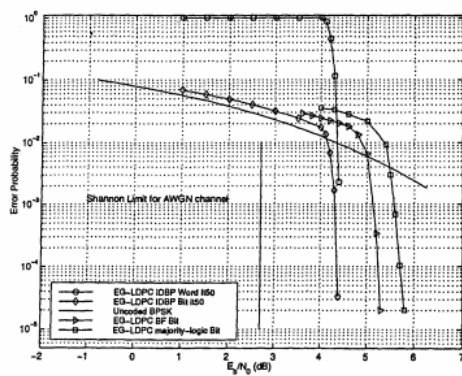


Figure 8: Bit- and block-error probabilities of the (16383,14179) EG-LDPC code.

27

Kou et al. (cont'd)

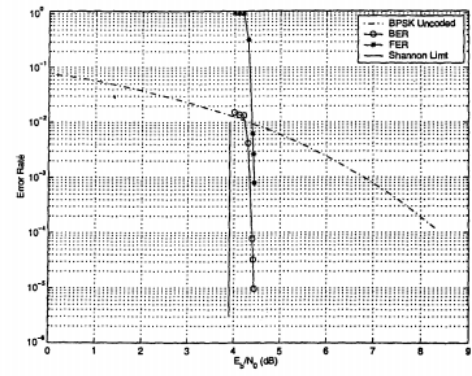


Figure 10: Bit- and block-error probabilities of the (65520,61425) EG-LDPC code based on IDBP.

28